

4°
edizione

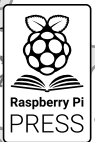
COMPLETAMENTE AGGIORNATA PER RASPBERRY PI 400

LA GUIDA
INTRODUTTIVA
ufficiale
Raspberry Pi

Come utilizzare il tuo
nuovo computer



di Gareth Halfacree



**LA GUIDA
INTRODUTTIVA**
ufficiale
Raspberry Pi

**Come utilizzare il tuo
nuovo computer**



Publicata per la prima volta nel 2020 da Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS

Direttore editoriale: Russell Barnes • Editore: Phil King
Design: Critical Media • Illustrazioni: Sam Alder
CEO: Eben Upton

ISBN: 978-1-912047-92-5

L'editore e i collaboratori non si assumono alcuna responsabilità per eventuali omissioni o errori relativi a beni, prodotti o servizi a cui si fa riferimento o che sono pubblicizzati in questa guida. Salvo diversa indicazione, il contenuto di questa guida è concesso in licenza a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)

Benvenuto nella guida introduttiva ufficiale di Raspberry Pi

Siamo sicuri che Raspberry Pi ti piacerà. Questo economico computer dal design compatto costa meno della maggior parte dei videogiochi, ma puoi utilizzarlo per imparare a programmare, costruire robot e creare tutti i progetti strani e incredibili che vuoi.

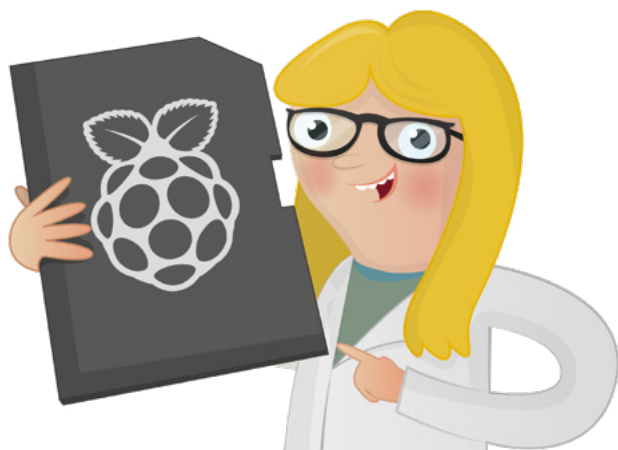
Con Raspberry Pi puoi fare tutto quello che faresti con un computer normale: navigare in Internet, giocare, guardare film e ascoltare musica. Ma Raspberry Pi è molto più di un moderno computer,

è il vero e proprio cuore di un computer. Imparerai a configurare il tuo sistema operativo e a collegare fili e circuiti direttamente ai pin della scheda. È stato progettato per insegnare ai più giovani come programmare utilizzando linguaggi come Scratch e Python, due dei principali linguaggi di programmazione inclusi nel sistema operativo ufficiale.

Mai come oggi il mondo ha bisogno di programmatori e Raspberry Pi è riuscito a suscitare nelle nuove generazioni l'interesse per l'informatica e la tecnologia.

Persone di tutte le età usano Raspberry Pi per creare progetti straordinari: dalle console per il retrogaming alle stazioni meteorologiche collegate a Internet.

Quindi, se anche tu vuoi progettare giochi, costruire robot o realizzare tanti progetti sorprendenti, sfoglia questa guida e preparati a iniziare.



Informazioni sull'autore

Gareth Halfacree è un giornalista freelance esperto in tecnologia, scrittore ed ex amministratore di sistema nel settore dell'istruzione con la passione del software e dell'hardware open-source. È stato uno dei primi ad adottare la piattaforma Raspberry Pi e a descriverne capacità e flessibilità in diverse pubblicazioni. Puoi trovarlo su Twitter come **@ghalfacree** o tramite il suo sito Web all'indirizzo **freelance.halfacree.co.uk**.



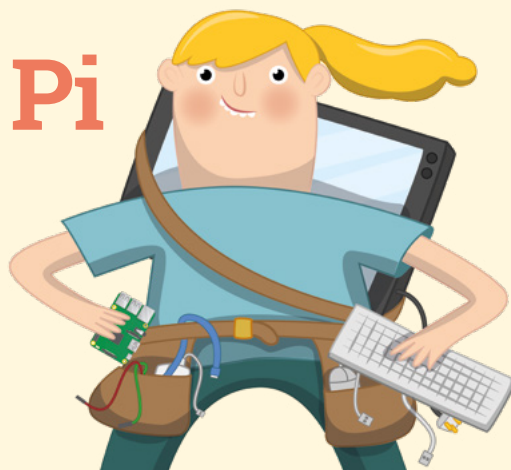
Indice

Capitolo 1 - Cos'è Raspberry Pi	008
Panoramica del tuo nuovo computer compatto	
Capitolo 2 - Introduzione a Raspberry Pi	022
Collega tutto il necessario per far funzionare il tuo Raspberry Pi	
Capitolo 3 - Come utilizzare Raspberry Pi	036
Scopri tutto sul sistema operativo Raspberry Pi	
Capitolo 4 - Come programmare con Scratch 3	054
Inizia a programmare con questo linguaggio a blocchi facile da imparare	
Capitolo 5 - Come programmare con Python	092
Impara a gestire la codifica basata sul testo utilizzando Python	
Capitolo 6 - Physical computing con Scratch e Python	120
Controlla i componenti elettronici collegati ai pin GPIO del tuo Raspberry Pi	
Capitolo 7 - Physical computing con Sense HAT	152
Utilizza i sensori e il display a matrice LED di questa scheda aggiuntiva	
Capitolo 8 - Fotocamera di Raspberry Pi	196
Scatta foto e video ad alta risoluzione con questa piccola macchina fotografica	
APPENDICI	
Appendice A - Come installare un sistema operativo su una scheda microSD	214
Appendice B - Come installare e disinstallare il software	216
Appendice C - Interfaccia a riga di comando	222
Appendice D - Ulteriori letture	228
Appendice E - Strumento Configurazione di Raspberry Pi	234
Appendice F - Impostazione della High Quality Camera	240
Appendice G - Specifiche tecniche di Raspberry Pi	244
Appendice H - Guida all'uso e avvertenze di sicurezza di Raspberry Pi	247

Capitolo 1

Cos'è Raspberry Pi

Scopri tutti i componenti e le funzionalità racchiuse nel tuo piccolo Raspberry Pi con una breve presentazione introduttiva



Raspberry Pi è un dispositivo straordinario: un computer perfettamente funzionante dal design compatto ed economico. Che tu stia cercando un computer da utilizzare per navigare nel Web o per giocare, per imparare a scrivere programmi o per creare circuiti e dispositivi progettati da te, Raspberry Pi e la sua incredibile community, ti supporteranno in ogni fase.

Raspberry Pi è noto come *computer a scheda singola*, che significa esattamente quello che sembra: è un computer a tutti gli effetti, proprio come un PC desktop, un laptop o uno smartphone, ma costituito da un'unica *scheda a circuito stampato*. Come la maggior parte dei computer a scheda singola, Raspberry Pi è grande più o meno come una carta di credito, ma ciò non significa che non sia potente: un Raspberry Pi può fare esattamente le stesse cose che fa un computer più grande, anche se non necessariamente alla stessa velocità.

La gamma di prodotti Raspberry Pi nasce dal desiderio di promuovere un'educazione informatica più pratica e accessibile in tutto il mondo. I creatori, che si sono uniti per formare l'organizzazione Raspberry Pi Foundation senza scopo di lucro, non avevano idea che sarebbe diventata così popolare. Dalle poche migliaia di unità costruite nel 2012 per sondare il terreno e andate immediatamente esaurite, sono passati a spedirne milioni in tutto il mondo. Queste schede si sono rivelate la soluzione ideale per abitazioni, aule, uffici, data centre, fabbriche e persino barche a pilotaggio automatico e palloni spaziali.

Dopo il Modello B originario, si sono susseguiti diversi modelli, ciascuno con specifiche migliorate o caratteristiche dedicate in base all'uso. La gamma Raspberry Pi Zero, ad esempio, è la versione ridotta dell'originale, che però manca di alcune caratteristiche, come ad esempio le varie porte USB e la porta di rete cablata, a favore di un design significativamente più piccolo e un ridotto fabbisogno di potenza.

Tutti i modelli Raspberry Pi però hanno una cosa in comune: sono *compatibili* fra loro, nel senso che il software scritto per un modello funzionerà perfettamente anche sugli altri. È anche possibile prendere l'ultima versione del sistema operativo di Raspberry Pi ed eseguirla sul prototipo del Modello B originario. Funzionerà più lentamente, ma funzionerà.

In questa guida introduttiva imparerai a conoscere Raspberry Pi 4 Modello B e Raspberry Pi 400, le versioni più recenti e potenti di Raspberry Pi, ma ciò che apprenderai potrai applicarlo facilmente anche agli altri modelli della gamma Raspberry Pi, quindi non preoccuparti se stai utilizzando una versione diversa.



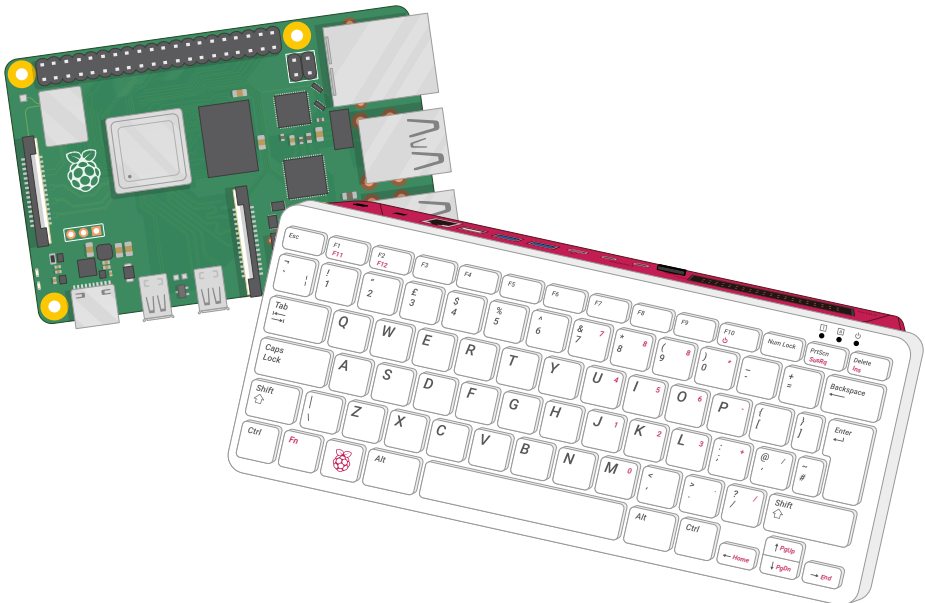
RASPBERRY PI 400

Se disponi di un Raspberry Pi 400, il circuito stampato è integrato nella custodia della tastiera. Continua a leggere per conoscere tutti i componenti di Raspberry Pi oppure vai a pagina 20 per una panoramica delle caratteristiche del dispositivo.

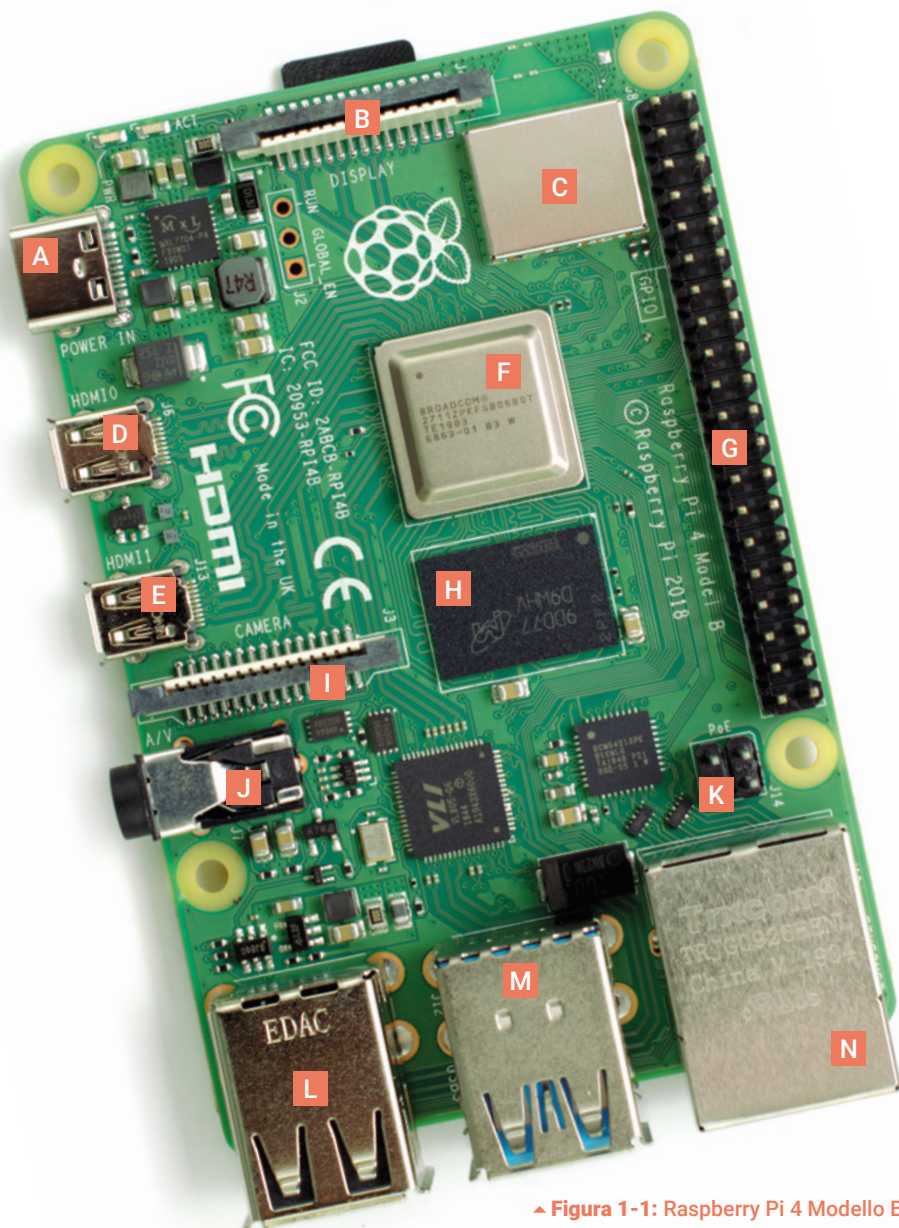


Presentazione introduttiva di Raspberry Pi

A differenza di un computer tradizionale dove i componenti sono racchiusi in un case e non si vedono, in un Raspberry Pi tutti i componenti, le porte e le altre caratteristiche sono visibili. Se preferisci una maggiore protezione, potrai acquistare un apposito case in un secondo momento. Il fatto di poter vedere i diversi componenti lo rende un ottimo strumento per imparare a capire e approfondire il funzionamento di un computer e delle varie parti extra, dette *periferiche*.



- | | | |
|--------------------------------------|-------------------------------|-------------------------|
| A Alimentazione USB di tipo C | F System-on-chip | K PoE |
| B Porta display DSI | G GPIO | L USB 2.0 |
| C Wireless / Bluetooth | H RAM | M USB 3.0 |
| D Micro-HDMI 0 | I porta fotocamera CSI | N porta Ethernet |
| E Micro-HDMI 1 | J AV da 3,5 mm | |



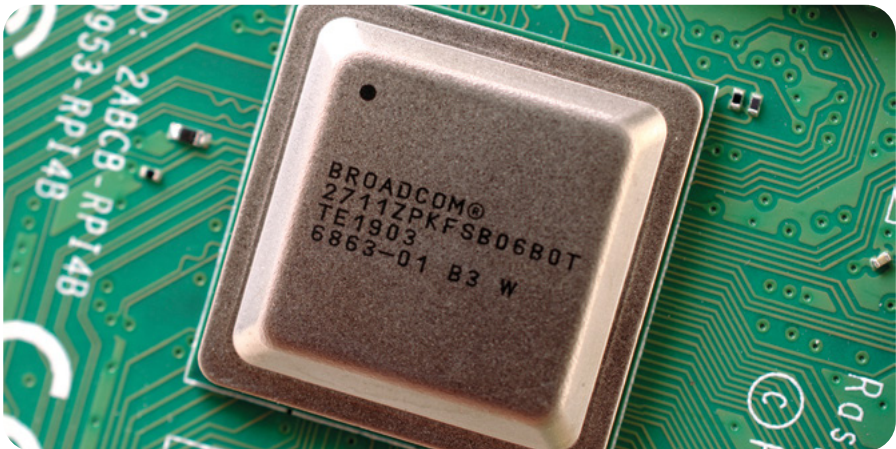
▲ Figura 1-1: Raspberry Pi 4 Modello B

Figura 1-1, un Raspberry Pi 4 Modello B visto dall'alto. Durante la lettura di questa guida, cerca di tenere sempre il tuo Raspberry Pi orientato come nella foto per evitare di confonderti quando dovrai maneggiare i vari componenti come, ad esempio, il terminale GPIO (maggiori dettagli nel **Capitolo 6, Physical computing con Scratch e Python**).

Anche se sembra che la scheda contenga troppi elementi, il Raspberry Pi è in realtà un dispositivo piuttosto semplice da capire, a partire dai *componenti*, i meccanismi interni che lo fanno funzionare.

I componenti di Raspberry Pi

Come ogni computer, Raspberry Pi è costituito da vari componenti, ognuno dei quali svolge un ruolo ben preciso. Il primo, e probabilmente il più importante, si trova appena al di sopra del punto centrale, sul lato superiore della scheda (**Figura 1-2**), ed è coperto da una protezione metallica: è il *system-on-chip*(SoC).



▲ **Figura 1-2:**Il system-on-chip di Raspberry Pi (SoC)

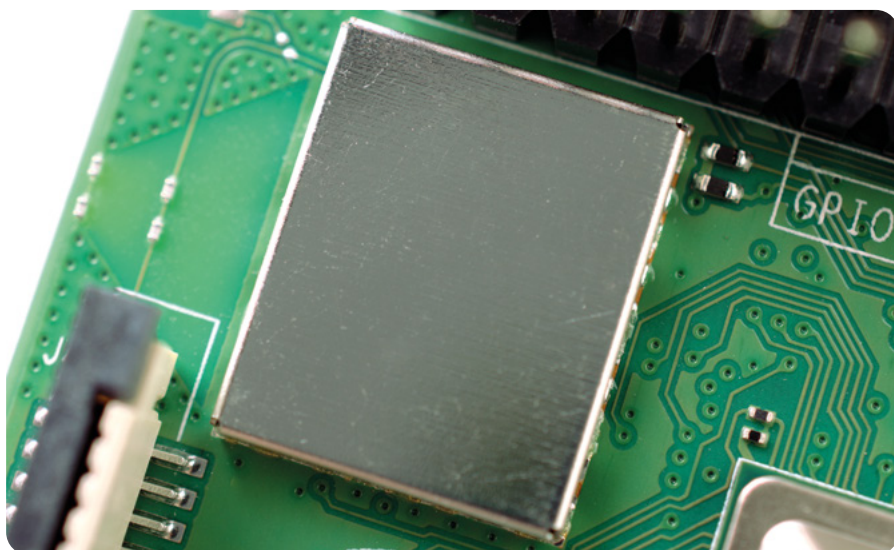
Il nome "system-on-chip" è un ottimo indizio di ciò che troverai sotto la protezione metallica: un chip di silicio, meglio conosciuto come *circuito integrato*, che contiene la maggior parte del sistema di Raspberry Pi. È composto infatti da un'*unità centrale di elaborazione*(CPU), comunemente considerata come il "cervello" di un computer, e dall'*unità di elaborazione grafica*(GPU), che gestisce tutta la parte grafica.

Un cervello senza memoria, però, non funziona bene ed è per questo che a fianco del SoC troverai un altro chip che sembra un quadratino nero di plastica (**Figura 1-3**, sul retro). Si tratta della *memoria ad accesso casuale* (RAM) di Raspberry Pi. Quando utilizzi Raspberry Pi, tutte le operazioni che esegui vengono salvate nella RAM e solo quando le salvi vengono scritte e memorizzate sulla scheda microSD. Sono le cosiddette memorie volatili e non volatili: la RAM volatile perde il suo contenuto ogni volta che Raspberry Pi viene spento, mentre la scheda microSD non volatile lo memorizza.



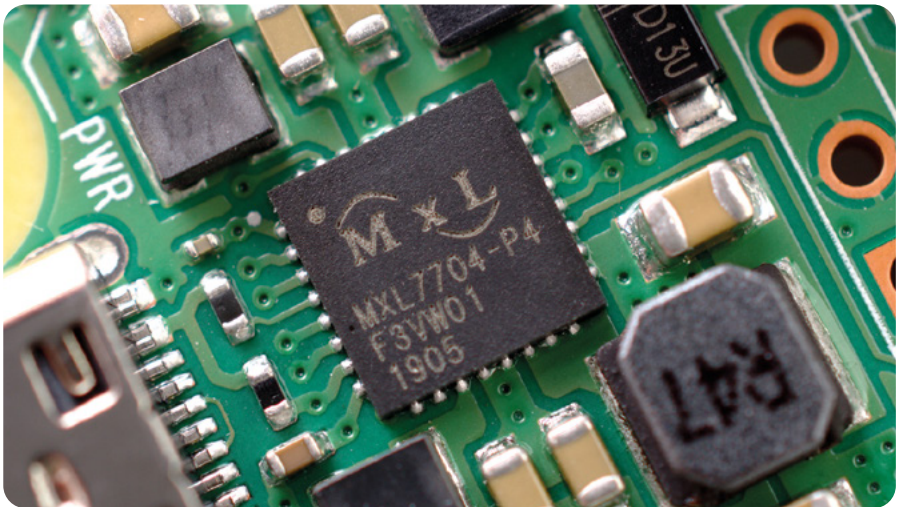
▲ **Figura 1-3:** memoria ad accesso casuale (RAM) di Raspberry Pi

Nella parte in alto a destra della scheda c'è un'altra copertura metallica (**Figura 1-4**) sotto la quale si trova la *radio*, il componente che consente a Raspberry Pi di comunicare con i dispositivi in modalità wireless. La radio svolge infatti la funzione di due componenti principali: *radio Wi-Fi*, per collegarsi alle reti informatiche, e *radio Bluetooth*, per connettersi a periferiche come i mouse e per inviare o ricevere dati da dispositivi intelligenti vicini come sensori o smartphone.



▲ **Figura 1-4:** il modulo radio di Raspberry Pi

Sul bordo inferiore della scheda, appena dietro il set centrale di porte USB, troviamo un altro chip con copertura in plastica nera. Si tratta del *controller USB* ed è responsabile del funzionamento delle quattro porte USB. Accanto, c'è un chip ancora più piccolo, il *controller di rete*, che gestisce la porta di rete Ethernet di Raspberry Pi. Leggermente al di sopra del connettore di alimentazione USB di tipo C nella parte superiore sinistra della scheda troviamo un ultimo chip nero, più piccolo degli altri (**Figura 1-5**). È il *circuito integrato di gestione dell'alimentazione (PMIC)*, che trasforma l'energia che entra dalla porta micro USB nella potenza necessaria a Raspberry Pi per funzionare.



▲ **Figura 1-5:** circuito integrato di gestione dell'alimentazione di Raspberry Pi (PMIC)

Se ti sembra di doverti ricordare troppe cose, non preoccuparti: per poter utilizzare Raspberry Pi non è necessario conoscere ogni componente o sapere dove si trova sulla scheda.

Le porte di Raspberry Pi

Raspberry Pi ha diverse porte, a cominciare dalle quattro *porte USB (Universal Serial Bus)* (Figura 1-6) al centro e a destra del bordo inferiore. Queste porte consentono di collegare a Raspberry Pi qualsiasi periferica compatibile con USB, da tastiere e mouse a fotocamere digitali e flash drive. Tecnicamente parlando, ci sono due tipi di porte USB: quelle con parti nere all'interno sono porte USB 2.0, basate sulla versione 2 dello standard Universal Serial Bus, mentre quelle con parti blu sono porte USB 3.0, più veloci, basate sulla più recente versione 3.



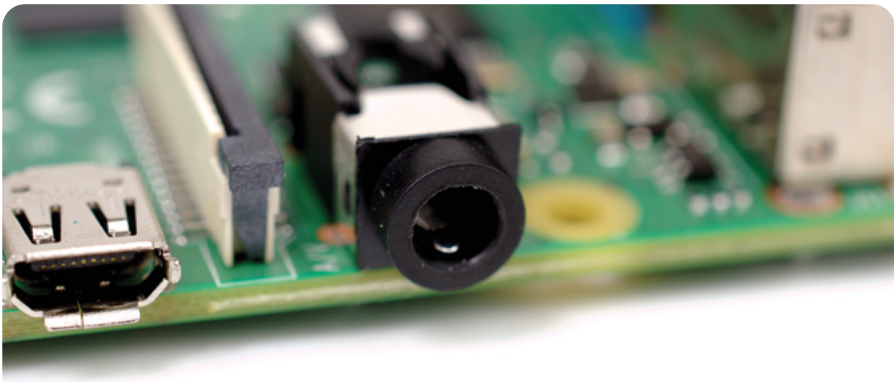
▲ Figura 1-6: le porte USB di Raspberry Pi

A destra delle porte USB c'è una *porta Ethernet*, nota anche come *porta di rete* (**Figura 1-7**), che serve a collegare Raspberry Pi a una rete di computer cablata utilizzando un cavo con il cosiddetto connettore RJ45. Se guardi attentamente nella parte inferiore della porta Ethernet, noterai due diodi luminosi (LED) che fungono da indicatori di stato che segnalano il corretto funzionamento della connessione.



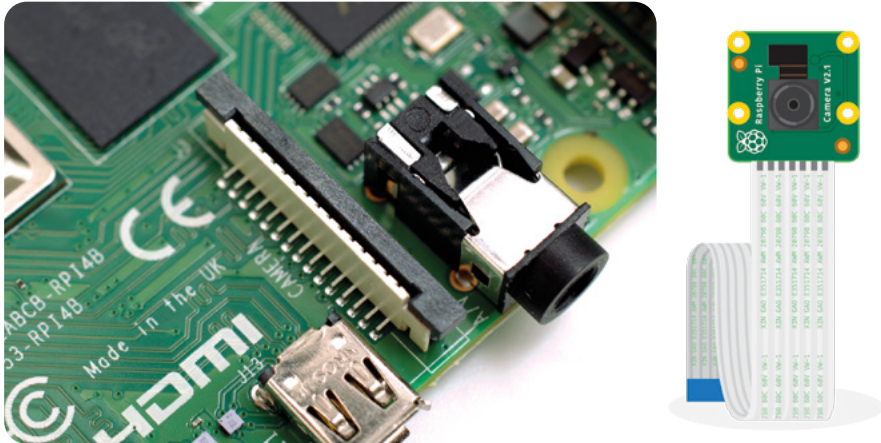
▲ **Figura 1-7:** la porta Ethernet di Raspberry Pi

Appena sopra la porta Ethernet, in corrispondenza del margine sinistro del Raspberry Pi, si trova un *connettore jack audio/video (AV) da 3,5 mm* (**Figura 1-8**), detto anche *jack per cuffie*, che può appunto essere collegato alle cuffie, anche se, per ottenere un suono di alta qualità, consigliamo di collegarlo ad altoparlanti amplificati. Questo componente presenta una caratteristica extra nascosta: oltre all'audio, il jack AV da 3,5 mm consente di inviare anche un segnale video che può essere collegato a TV, proiettori e altri display che supportano il *segnale video composito* utilizzando un cavo speciale chiamato adattatore *tip-ring-ring-sleeve (TRRS)*.



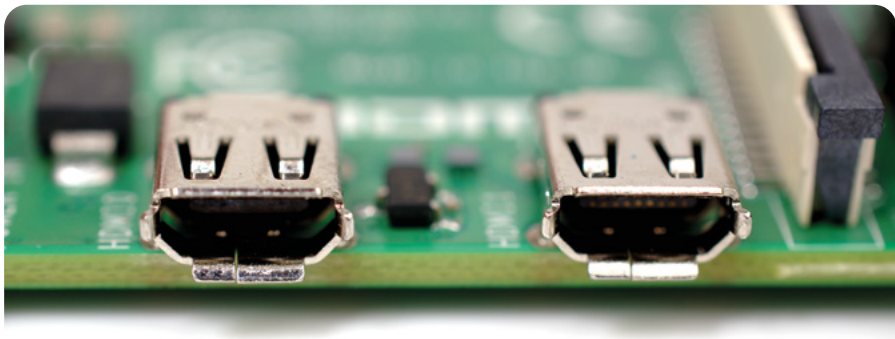
▲ **Figura 1-8:** connettore jack AV da 3,5 mm Raspberry Pi

Subito sopra il jack AV da 3,5 mm c'è un connettore dall'aspetto strano con una linguetta di plastica che può essere tirata verso l'alto. È il *connettore per fotocamera*, conosciuto anche come *Interfaccia seriale della fotocamera (CSI)* (**Figura 1-9**). Questo componente ti consente di utilizzare il Raspberry Pi Camera Module (modulo per fotocamera) appositamente progettato (maggiori informazioni al **Capitolo 8, Fotocamera di Raspberry Pi**.)



▲ **Figura 1-9:** il connettore per fotocamera di Raspberry Pi

Sopra, sempre in corrispondenza del margine sinistro della scheda, si trovano le *porte micro-HDMI (High Definition Multimedia Interface)*, che sono una versione più piccola dei connettori che troviamo su console di gioco, set-top box e televisori (**Figura 1-10**). L "M" di Multimedia indica che la porta è in grado di inviare sia segnali audio che video, mentre "High Definition" indica che le prestazioni sono di alta qualità. Potrai utilizzare queste porte per collegare Raspberry Pi a uno o due dispositivi di visualizzazione: un monitor per computer, un televisore o un proiettore.



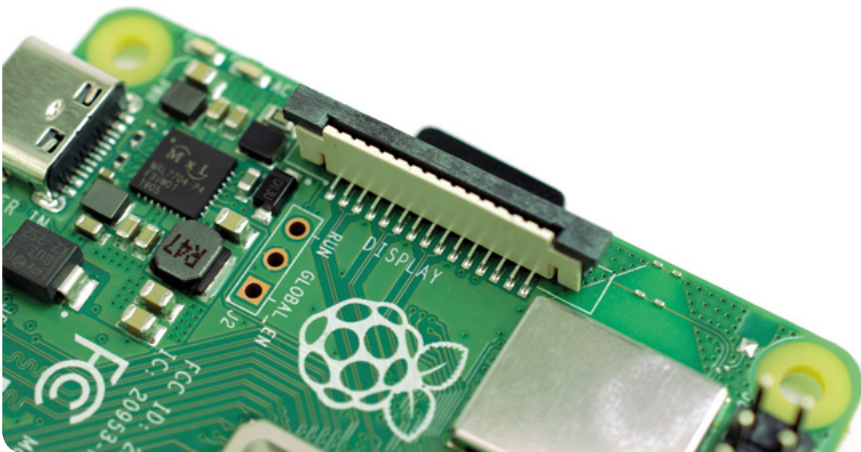
▲ **Figura 1-10:** le porte micro-HDMI di Raspberry Pi

Sopra le porte HDMI si trova una *porta di alimentazione USB di tipo C* (**Figura 1-11**) che ti servirà per collegare il Raspberry Pi a una fonte di alimentazione. La porta USB di tipo C è molto comune su smartphone, tablet e altri dispositivi portatili. Puoi alimentare Raspberry Pi con un tradizionale caricabatterie per cellulare, ma per migliori prestazioni ti consigliamo di utilizzare l'alimentatore ufficiale Raspberry Pi USB di tipo C.

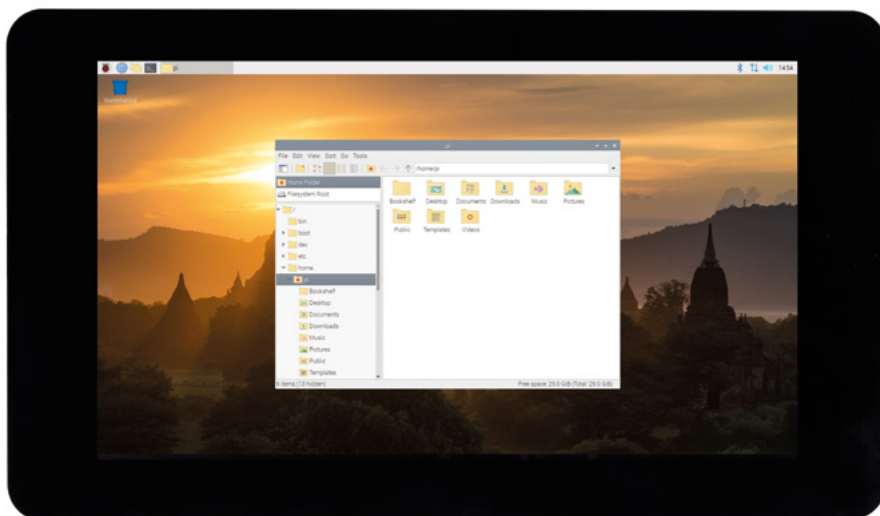


▲ **Figura 1-11:** la porta di alimentazione USB di tipo C di Raspberry Pi

Sul bordo superiore della scheda c'è un altro connettore dall'aspetto curioso (**Figura 1-12**) che a prima vista potrebbe sembrare identico a quello della fotocamera, ma non lo è. Si tratta del *connettore del display* o *Interfaccia seriale del display (DSI)*, progettato per essere utilizzato con un display Raspberry Pi Touch (**Figura 1-13**, sul retro)

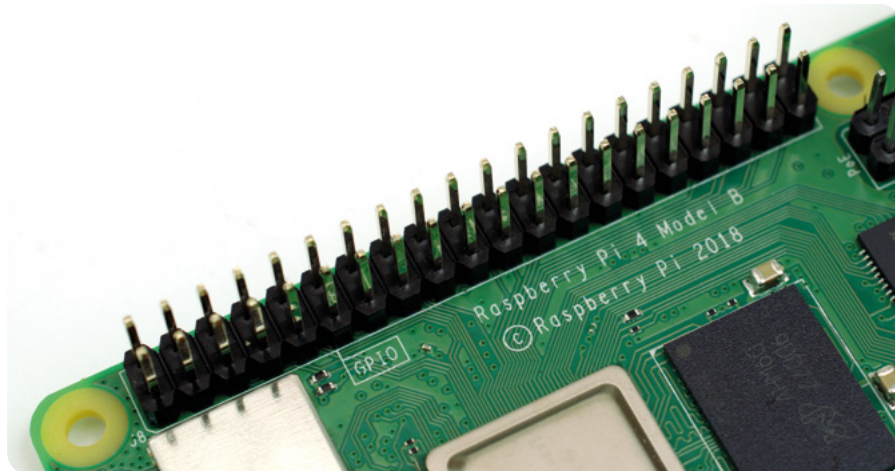


▲ **Figura 1-12:** il connettore del display di Raspberry Pi (DSI)



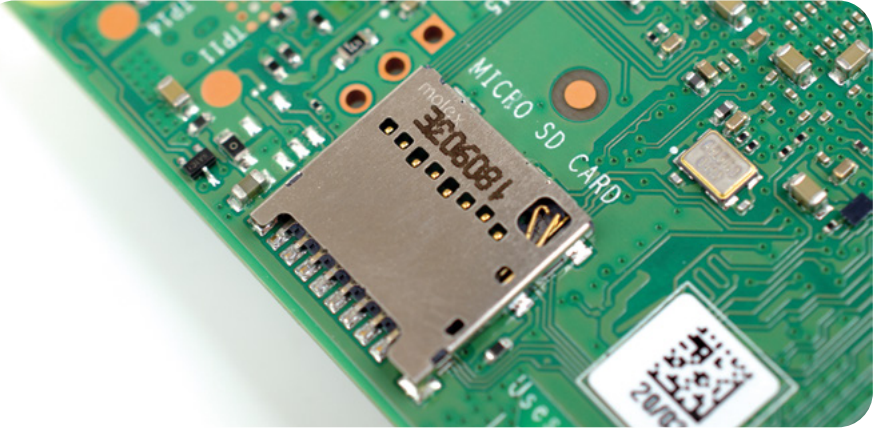
▲ **Figura 1-13:** il display Raspberry Pi Touch

Sul bordo destro della scheda si trovano 40 pin in metallo, suddivisi in due file da 20 pin (**Figura 1-14**). Si tratta del *terminale GPIO* (*general-purpose input/output*), un componente del Raspberry Pi utilizzato per comunicare con l'hardware aggiuntivo, da LED e pulsanti fino a sensori di temperatura, joystick e monitor della frequenza cardiaca. Scoprirai di più sul terminale GPIO nel **Capitolo 6, Physical computing con Scratch e Python**. Appena sotto, a sinistra del terminale, c'è n'è un altro più piccolo con quattro pin che serve a collegare l'HAT Power over Ethernet (PoE), un piccolo add-on opzionale che permette al Raspberry Pi di ricevere l'alimentazione da una connessione di rete piuttosto che dalla porta USB di tipo C.



▲ **Figura 1-14:** il terminale GPIO di Raspberry Pi

C'è un'ultima porta, ma non si trova su questo lato della scheda. Capovolgi la scheda e troverai un *connettore per scheda microSD* sul lato opposto rispetto al connettore del display (**Figura 1-15**). Si tratta della memoria di Raspberry Pi: la scheda microSD qui inserita contiene tutti i file salvati, tutti i software installati e il sistema operativo che fa funzionare Raspberry Pi.



▲ **Figura 1-15:** il connettore per scheda microSD di Raspberry Pi



▲ **Figura 1-16:** Raspberry Pi 400 ha una tastiera integrata

Raspberry Pi 400

Nel Raspberry Pi 400 troviamo gli stessi componenti del Raspberry Pi 4 collocati all'interno di un alloggiamento per tastiera. Oltre a proteggere i componenti, l'alloggiamento occupa meno spazio sulla scrivania e ti aiuta a mantenere i cavi più ordinati.

Il modello 400 è dotato degli stessi componenti principali del modello 4, compresi il sistema su chip e la memoria. Non li vedi, ma ci sono. Quello che puoi vedere sono le parti esterne, a cominciare dalla tastiera (**Figura 1-16**). In prossimità dell'angolo destro ci sono tre diodi luminosi (LED): il primo si illumina quando si preme il tasto Num Lock, che commuta alcuni tasti in tastierino numerico su una tastiera standard, il secondo si illumina quando si preme Caps Lock, che consente di usare lettere maiuscole anziché minuscole, mentre l'ultimo si illumina quando il Raspberry Pi 400 è acceso e lampeggia per indicare che è in esecuzione un'operazione.

Nel Raspberry Pi 400 (**Figura 1-17**) le porte sono sul retro della scheda. La porta più a sinistra, vista dal retro, è il terminale GPIO (general-purpose input/output). È lo stesso terminale descritto a pagina 17, ma capovolto: il primo pin, il pin 1, è in alto a destra, mentre l'ultimo pin, il pin 40, è in basso a sinistra. Scopri di più sul terminale GPIO nel **Capitolo 6, Physical computing con Scratch e Python**.



▲ **Figura 1-17:** le porte sul retro del Raspberry Pi 400

A fianco del terminale GPIO si trova lo slot per la scheda microSD che funge da memoria di Raspberry Pi 400 per il sistema operativo, le applicazioni e altri dati. La scheda microSD è preinstallata, ma puoi rimuoverla esercitando una leggera pressione su di essa finché non si sblocca ed esce dallo slot. Quando la rimetti in posizione, assicurati che i contatti metallici siano rivolti verso il basso; la scheda dovrebbe bloccarsi in posizione con un leggero clic.

Le due porte successive sono le porte micro-HDMI, per collegare un monitor, una TV o un altro display. Come Raspberry Pi 4, anche Raspberry Pi 400 supporta fino a due display. Accanto a esse c'è la porta di alimentazione USB di tipo C per il collegamento all'alimentatore Raspberry Pi o all'alimentatore USB compatibile.

Le due porte blu sono porte USB 3.0, che forniscono una connessione ad alta velocità a dispositivi come unità di memoria a stato solido (SSD), memory stick, stampanti e altro ancora. La porta bianca a destra, invece, è una porta USB 2.0 a bassa velocità, alla quale è possibile collegare il mouse Raspberry Pi preinstallato.

L'ultima è una porta di rete Gigabit Ethernet, che consente di collegare il Raspberry Pi 400 tramite un cavo di rete invece che tramite rete wireless Wi-Fi integrata. Per saperne di più su come collegare il Raspberry Pi 400 ad una rete, consulta il Capitolo 2, Introduzione a Raspberry Pi.

Capitolo 2

Introduzione a Raspberry Pi

Scopri quali sono i componenti esterni essenziali per il tuo Raspberry Pi e come collegarli per farlo funzionare in modo ottimale



Raspberry Pi è stato progettato per garantire la massima semplicità e velocità in termini di configurazione e utilizzo, ma come ogni computer si basa su una serie di componenti esterni, chiamati *periferiche*. È facile lasciarsi impressionare dal circuito stampato di Raspberry Pi e temere di non riuscirne a capire il funzionamento a causa del suo aspetto così diverso da quello a cui siamo abituati con i classici computer con case, ma non devi preoccuparti. Seguendo le procedure dettagliate di questa guida, imparerai a utilizzarlo in meno di dieci minuti.

Se questa guida ti è stata fornita con un Kit Desktop di Raspberry Pi o con un Raspberry Pi 400, hai già quasi tutto quello che ti serve per iniziare. Procurati un monitor o un televisore con presa HDMI, lo stesso tipo di presa utilizzata da un set-top-box, lettori Blu-ray e console, in modo da vedere cosa sta facendo il tuo Raspberry Pi.

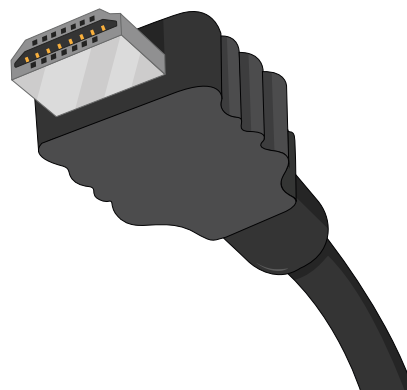
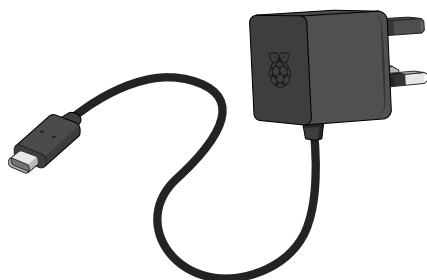
Se invece hai acquistato Raspberry Pi senza accessori, allora avrai bisogno anche di:

- **Alimentatore USB** - Un alimentatore da 5 V con potenza nominale pari a 3 ampere (3 A) e connettore USB di tipo C. Consigliamo di utilizzare l'alimentatore originale di Raspberry Pi, poiché è in grado di gestire ottimamente le richieste di potenza a commutazione rapida di Raspberry Pi.

- **Scheda microSD con NOOBS** - La scheda microSD funge da memoria permanente di Raspberry Pi. Tutti i file creati e i software installati, oltre al sistema operativo stesso, vengono memorizzati sulla scheda. Per iniziare, puoi usare una scheda da 8 GB e successivamente passare a quella da 16 GB, per avere più spazio. Utilizzare una scheda con NOOBS (New Out-Of-Box Software) preinstallato ti farà risparmiare tempo. In alternativa, ti consigliamo di consultare l'**Appendice A** per sapere come installare un sistema operativo (OS) su una scheda vuota.

- **Tastiera e mouse USB** - La tastiera e il mouse consentono di controllare il tuo Raspberry Pi. Quasi tutte le tastiere e i mouse cablati o wireless con connettore USB sono compatibili con Raspberry Pi, anche se alcune tastiere per gaming con luci colorate assorbono troppa energia per potere essere utilizzate in modo affidabile.

- **Cavo Micro-HDMI** - Questo cavo è fondamentale per inviare l'audio e le immagini da Raspberry Pi al tuo televisore o monitor. Un'estremità del cavo presenta un connettore micro-HDMI per Raspberry Pi mentre all'altra estremità c'è un connettore HDMI standard da inserire nel monitor. In alternativa, puoi utilizzare un adattatore da micro-HDMI a HDMI e un cavo HDMI standard. Se utilizzi un monitor senza presa HDMI, puoi acquistare gli adattatori da micro-HDMI a DVI-D, DisplayPort o VGA. Per collegare eventuali televisori vecchi che utilizzano il video composito o che hanno una presa SCART, ti consigliamo un cavo audio/video TRRS (tip-ring-ring-sleeve) con punta da 3,5 mm.



Raspberry Pi può essere utilizzato in tutta sicurezza anche senza case, a condizione che non lo si posizioni su una superficie metallica che potrebbe condurre elettricità e causare un cortocircuito. Tuttavia, l'utilizzo di un case garantisce una maggiore protezione. Il Kit Desktop include il case originale di Raspberry Pi, mentre i case di terze parti sono disponibili presso tutti i migliori rivenditori.

Se hai intenzione di utilizzare Raspberry Pi su rete cablata anziché su rete wireless (Wi-Fi), avrai bisogno anche di un cavo di rete, che dovrai collegare allo switch o al router della rete che vuoi utilizzare. Se stai pensando di utilizzare la radio wireless integrata di Raspberry Pi, invece, non avrai bisogno di un cavo, ma dovrai comunque sapere il nome e la chiave o la passphrase della tua rete wireless.



CONFIGURAZIONE DEL RASPBERRY PI 400

Le seguenti istruzioni consentono di configurare il Raspberry Pi 4 o un altro dispositivo della famiglia Raspberry Pi. Le istruzioni su come impostare Raspberry Pi 400 si trovano a pagina 32.



Come impostare l'hardware

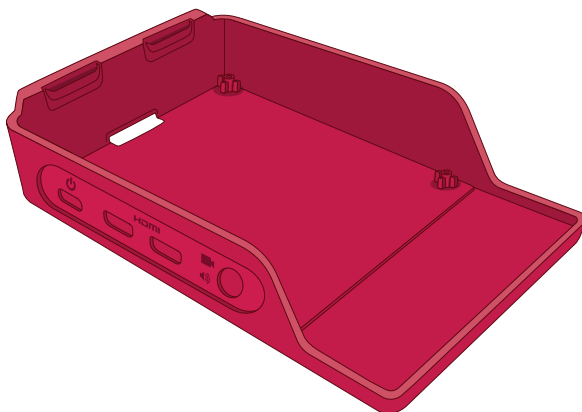
Estrai il tuo Raspberry Pi dalla confezione. Come vedi, si tratta di un componente hardware robusto, ma ciò non significa che sia indistruttibile. Cerca di abituarti a non tenere la scheda di piatto ma dai bordi e fai molta attenzione ai pin metallici rialzati. Se dovessero piegarsi, infatti, potrebbe essere più difficile utilizzare schede aggiuntive o altri componenti hardware e, nel peggiore dei casi, potrebbe verificarsi un cortocircuito che danneggerebbe il tuo Raspberry Pi.

Se non l'hai già fatto, dai un'occhiata al **Capitolo 1, Cos'è Raspberry Pi** per sapere dove si trovano esattamente le varie porte e la loro funzione.

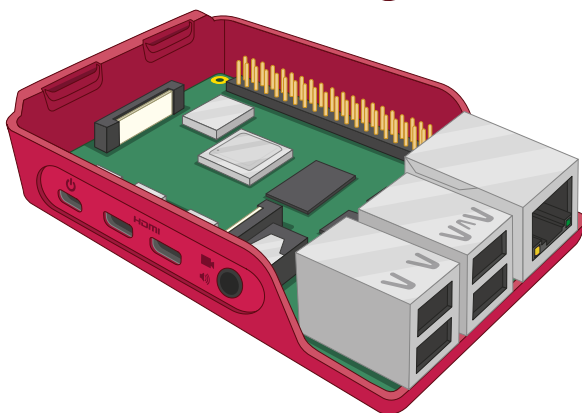
Come montare il case

Per installare Raspberry Pi in un case, la prima cosa da fare è montarlo. Se utilizzi il case originale di Raspberry Pi, separa la base rossa dal coperchio bianco.

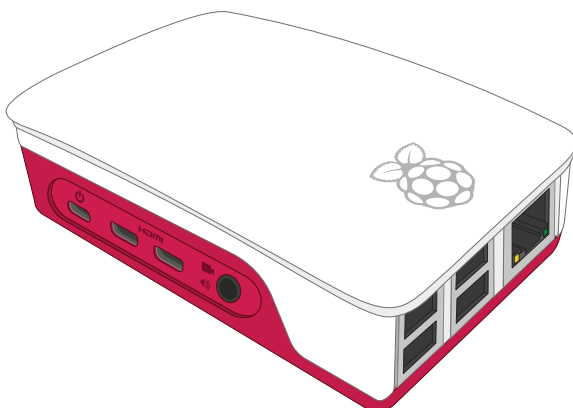
- 1** Prendi la base e tienila in modo che l'estremità più alta sia a sinistra e quella più bassa a destra.



- 2** Tenendo il Raspberry Pi (senza scheda microSD inserita) per le porte USB ed Ethernet, leggermente inclinato, inserisci i connettori (USB di tipo C, 2 micro-HDMI e 3,5 mm) negli appositi alloggiamenti della base, quindi abbassa delicatamente l'altra estremità in modo da posizionarlo in orizzontale.

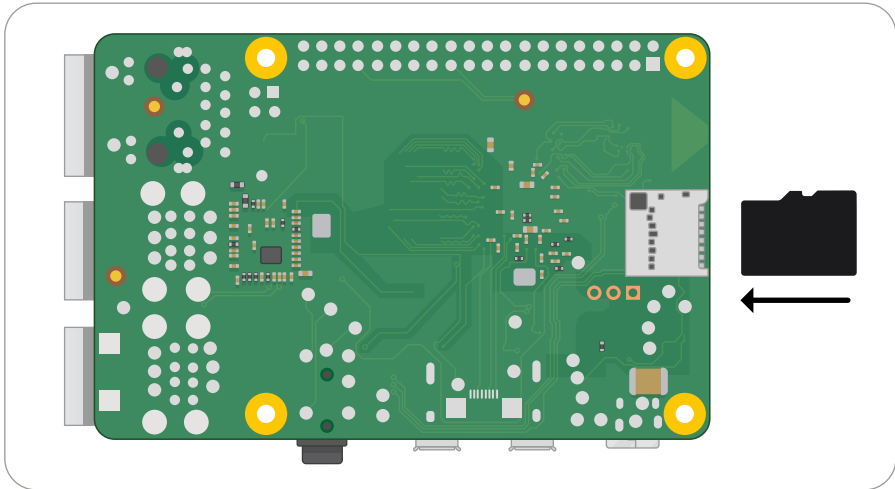


- 3** Prendi il coperchio bianco e inserisci le due clip a sinistra nei corrispondenti alloggiamenti della base, sopra lo slot per schede microSD. Quando sono in posizione, spingi il lato destro (sopra le porte USB) verso il basso fino a sentire un clic.

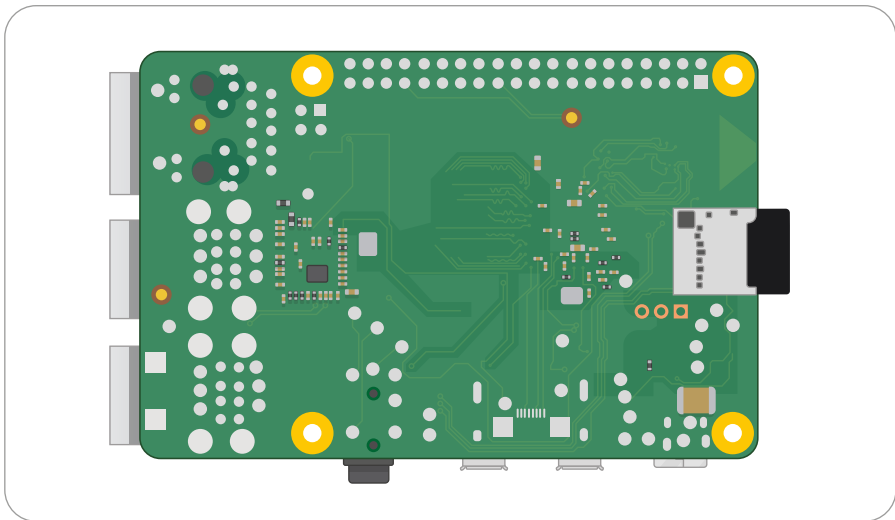


Come installare la scheda microSD

Per installare la scheda microSD, la *memoria* di Raspberry Pi, capovolgila il Raspberry Pi (nel case, se utilizzato) e inserisci la scheda nello slot per microSD con l'etichetta rivolta verso l'alto. Può essere inserita in un solo modo e dovresti riuscirci senza premere troppo.



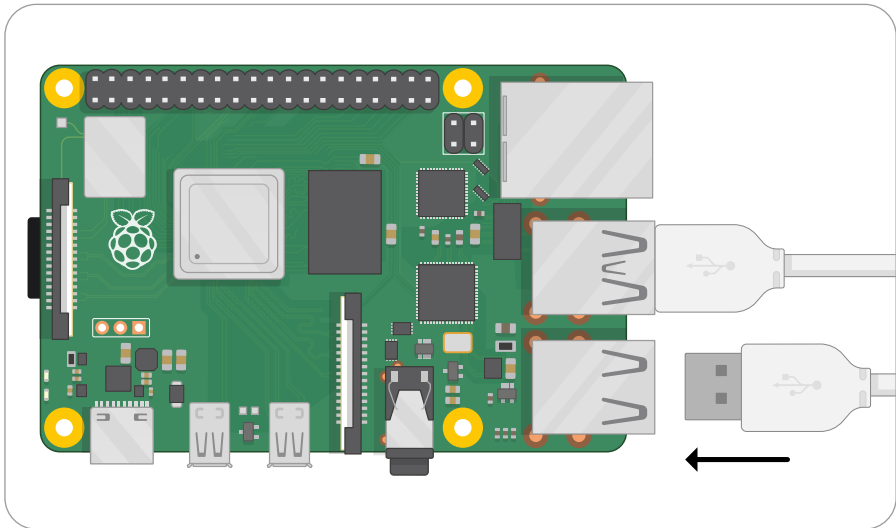
La scheda microSD si inserirà nel connettore e si bloccherà senza fare clic.



Se in futuro vorrai rimuoverla, ti basterà estrarla delicatamente prendendola per l'estremità. Se usi un modello vecchio di Raspberry Pi, dovrai prima esercitare una leggera pressione sulla scheda per sbloccarla (nei Raspberry Pi 3 o 4 non è necessario).

Come collegare una tastiera e un mouse

Collega il cavo USB della tastiera a una delle quattro porte USB (2.0 o 3.0) di Raspberry Pi. Se stai utilizzando la tastiera originale di Raspberry Pi, sul retro troverai una porta USB per il mouse; in caso contrario, dovrai solo collegare il cavo USB del mouse a un'altra porta USB su Raspberry Pi.



I connettori USB per la tastiera e il mouse dovrebbero inserirsi facilmente; se devi forzare il connettore, allora qualcosa non va. Controlla che il connettore USB sia rivolto verso l'alto.

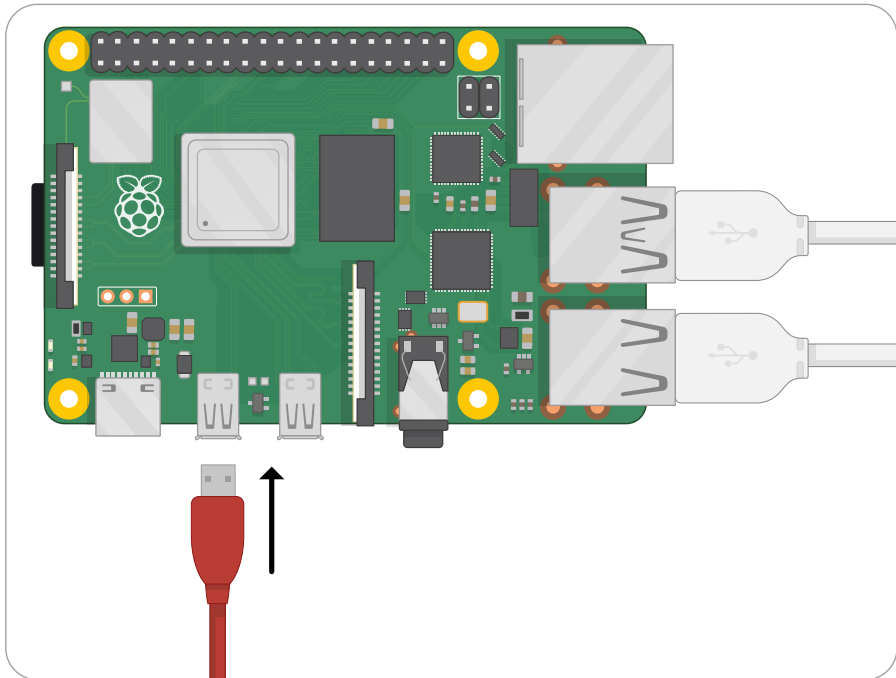


TASTIERA E MOUSE

La tastiera e il mouse sono il mezzo principale per dire a Raspberry Pi cosa fare. In ambito informatico, vengono chiamati *dispositivi di input*, al contrario del monitor che è un *dispositivo di output*.

Come collegare un monitor

Prendi il cavo micro-HDMI e collega l'estremità più piccola alla porta micro-HDMI più vicina alla porta USB di tipo C del Raspberry Pi e l'altra estremità al tuo monitor. Se il monitor ha più di una porta HDMI, cerca un numero di porta accanto al connettore: per vedere il monitor di Raspberry Pi dovrai commutare il televisore su questo ingresso. Se non trovi il numero di porta, non preoccuparti: prova ogni ingresso finché non trovi quello del Raspberry Pi.



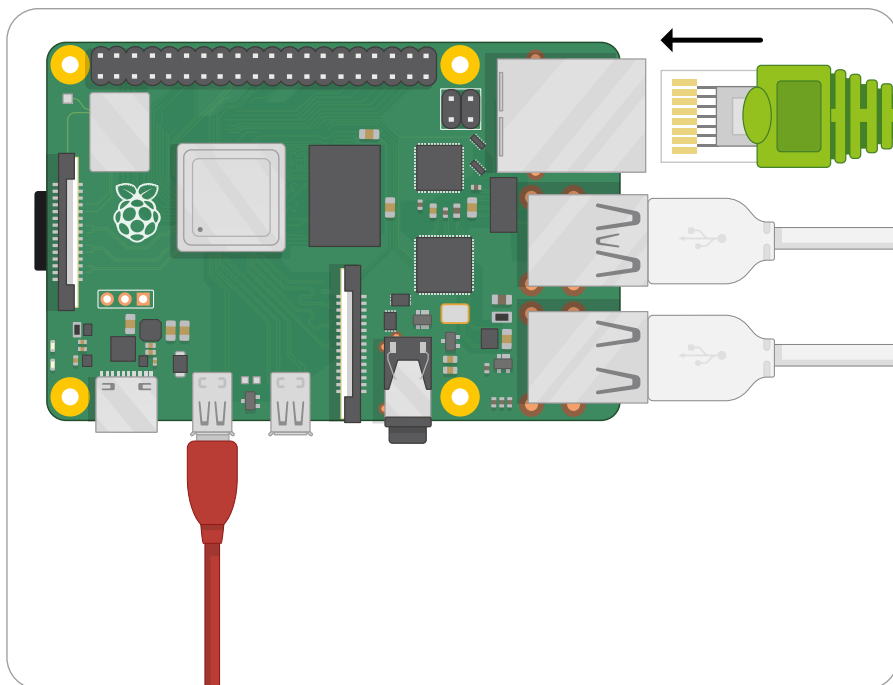
COLLEGAMENTO TV

Se il tuo televisore o monitor non ha un connettore HDMI, non significa che tu non possa usare Raspberry Pi. I cavi adattatori, disponibili presso qualsiasi rivenditore di elettronica, consentono di convertire la porta micro-HDMI di Raspberry Pi in DVI-D, DisplayPort o VGA per l'uso con monitor di vecchi computer: basta collegare un'estremità del cavo adattatore alla porta micro-HDMI di Raspberry Pi e l'altra estremità a un apposito cavo da collegare al monitor. Se il tuo televisore ha solo un ingresso video composito o SCART, puoi acquistare cavi adattatori da 3,5 mm tip-ring ring-sleeve (TRRS) e adattatori composite-to-SCART che si collegano al jack AV da 3,5 mm.



Come collegare un cavo di rete (opzionale)

Per collegare il Raspberry Pi a una rete cablata, prendi un cavo Ethernet e inseriscilo nella porta Ethernet del Raspberry Pi, con la linguetta rivolta verso il basso, fino a sentire un clic. Per rimuovere il cavo, premi la linguetta verso l'interno in direzione della presa ed estrailo delicatamente.

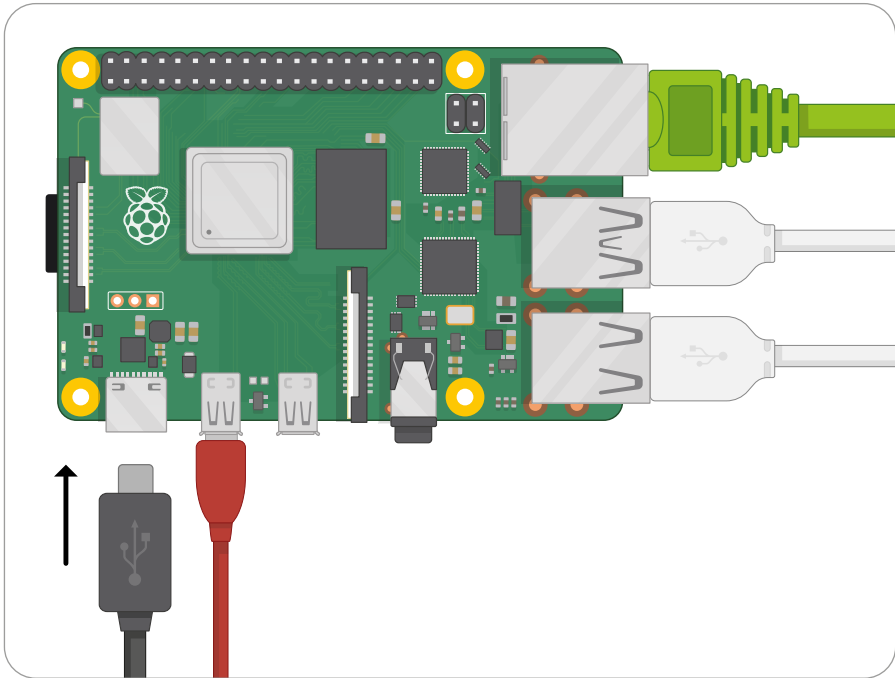


L'altra estremità del cavo di rete deve essere collegata a una porta libera dell'hub di rete, dello switch o del router nello stesso modo.

Come collegare l'alimentazione

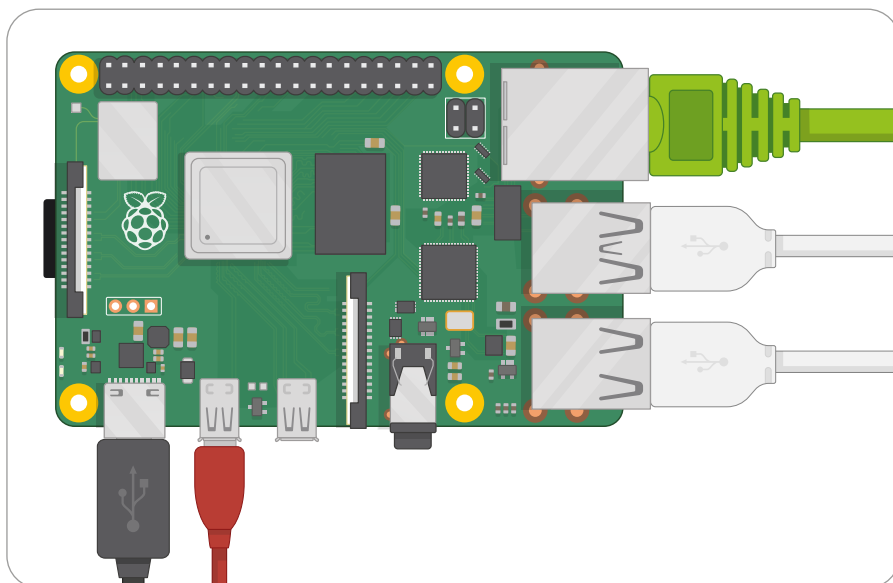
Collegare il Raspberry Pi a un alimentatore è l'ultimo passo del processo di configurazione dell'hardware e dovresti farlo solo quando sei pronto a configurare il software. Raspberry Pi non ha un interruttore e si accende non appena viene collegato a un alimentatore sotto tensione.

Per prima cosa, collega l'estremità USB di tipo C del cavo di alimentazione al connettore di alimentazione USB di tipo C di Raspberry Pi. Può essere inserito in entrambi i sensi e dovrebbe entrare facilmente. Se il tuo alimentatore ha un cavo scollegabile, assicurati che l'altra estremità sia inserita nel corpo dell'alimentatore.

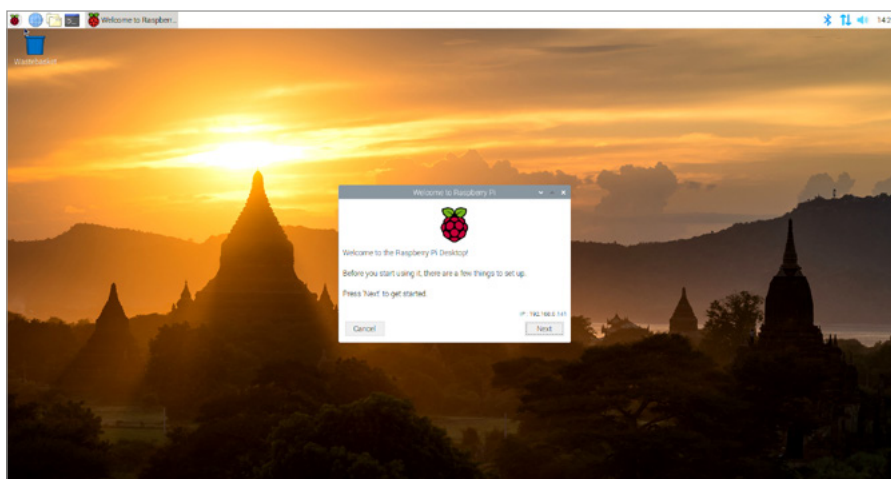


Infine, collega l'alimentatore a una presa di corrente e accendila: il Raspberry Pi inizierà a funzionare immediatamente.

Complimenti, hai montato correttamente il tuo Raspberry Pi!



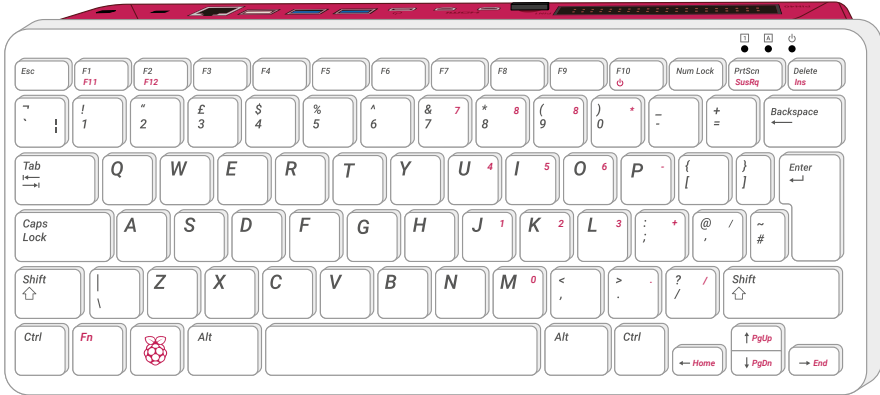
Verranno visualizzati per pochi istanti quattro loghi Raspberry Pi in alto a sinistra di una schermata nera e potresti visualizzare una schermata blu quando il software si ridimensionerà per consentirti di utilizzare al meglio la scheda microSD. Se vedi una schermata nera, aspetta qualche minuto: al primo avvio, Raspberry Pi deve fare un po' di pulizie in background. Dopo qualche minuto visualizzerai il desktop del sistema operativo Raspberry Pi OS e la procedura di configurazione guidata, come in **Figura 2-1**. Il tuo sistema operativo è ora pronto per essere configurato, cosa che imparerai a fare nel **Capitolo 3, Come utilizzare Raspberry Pi**.



▲ **Figura2-1:** il desktop e la procedura guidata di configurazione di Raspberry Pi OS

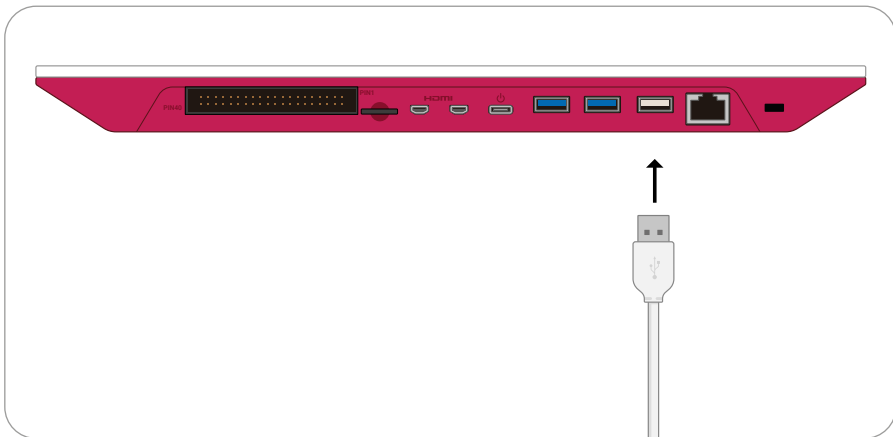
Configurazione di Raspberry Pi 400

A differenza di Raspberry Pi 4, Raspberry Pi 400 è dotato di una tastiera integrata e della scheda microSD già installata. Dovrai comunque collegare alcuni cavi all'inizio, ma ci vorranno solo pochi minuti.



Come collegare un mouse

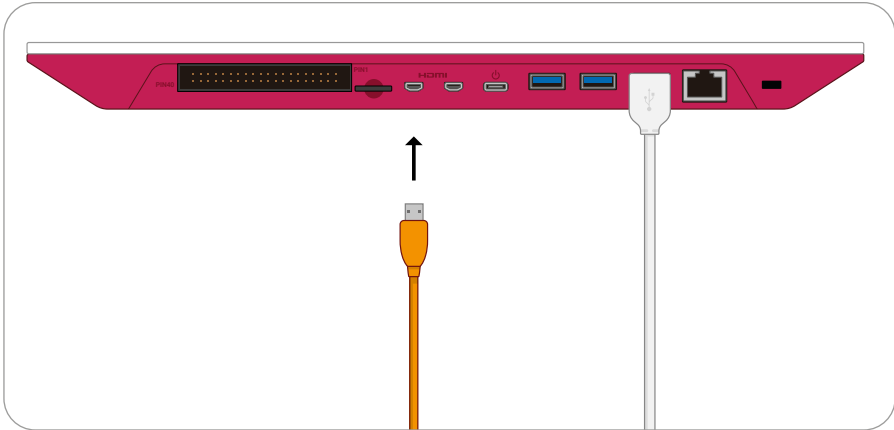
La tastiera del Raspberry Pi 400 è già collegata, quindi ti resta da collegare solo il mouse. Prendi il cavo USB all'estremità del mouse e inseriscilo in una delle tre porte USB (2.0 o 3.0) sul retro del Raspberry Pi 400. Se desideri lasciare libere le due porte USB 3.0 ad alta velocità per altri accessori, utilizza la porta bianca.



Il connettore USB dovrebbe inserirsi facilmente; se devi forzarlo, allora qualcosa non va. Controlla che il connettore USB sia rivolto verso l'alto.

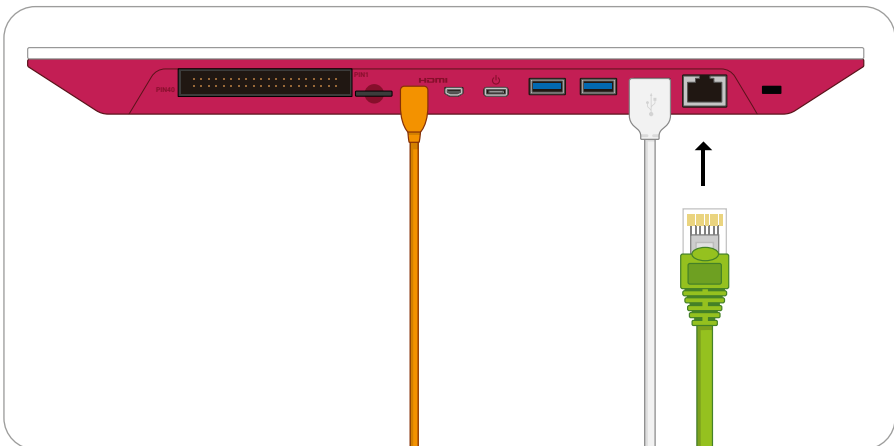
Come collegare un monitor

Prendi il cavo micro-HDMI e collega l'estremità più piccola alla porta micro-HDMI più vicina allo slot microSD del Raspberry Pi 400 e l'altra estremità al tuo monitor. Se il monitor ha più di una porta HDMI, cerca un numero di porta accanto al connettore: per vedere il monitor di Raspberry Pi dovrai commutare il televisore su questo ingresso. Se non trovi il numero di porta, non preoccuparti: prova ogni ingresso finché non trovi quello del Raspberry Pi.



Come collegare un cavo di rete (opzionale)

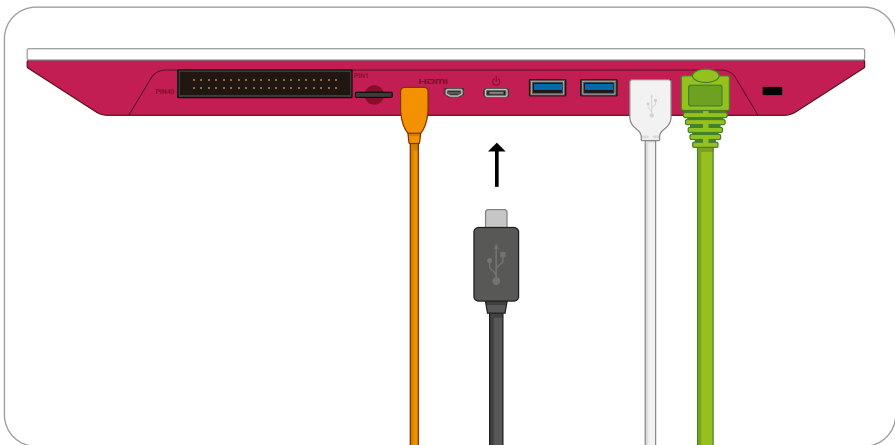
Per collegare il Raspberry Pi 400 a una rete cablata, prendi un cavo Ethernet e inseriscilo nella porta corrispondente del Raspberry Pi 400, con la linguetta rivolta verso l'alto, fino a sentire un clic. Per rimuovere il cavo, premi la linguetta verso l'interno in direzione della presa ed estrailo delicatamente.



L'altra estremità del cavo di rete deve essere collegata a una porta libera dell'hub di rete, dello switch o del router nello stesso modo.

Come collegare l'alimentazione

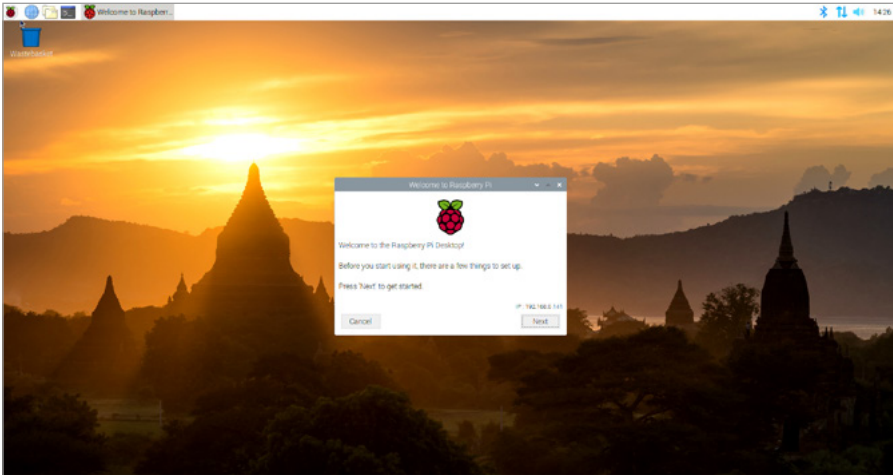
Collegare il Raspberry Pi 400 a un alimentatore è l'ultimo passo del processo di configurazione dell'hardware e dovresti farlo solo quando sei pronto a configurare il software. Raspberry Pi 400 non ha un interruttore e si accende non appena viene collegato a un alimentatore sotto tensione. Per prima cosa, collega l'estremità USB di tipo C del cavo di alimentazione al connettore di alimentazione USB di tipo C di Raspberry Pi. Può essere inserito in entrambi i sensi e dovrebbe entrare facilmente. Se il tuo alimentatore ha un cavo scollegabile, assicurati che l'altra estremità sia inserita nel corpo dell'alimentatore.



Infine, collega l'alimentatore a una presa di corrente e accendila: il Raspberry Pi 400 inizierà a funzionare immediatamente. Complimenti, hai montato correttamente il tuo Raspberry Pi 400!

Verranno visualizzati per pochi istanti quattro loghi Raspberry Pi in alto a sinistra di una

schermata nera e potresti visualizzare una schermata blu quando il software si ridimensionerà per consentirti di utilizzare al meglio la scheda microSD. Se vedi una schermata nera, aspetta qualche minuto: al primo avvio, Raspberry Pi deve fare un po' di pulizie in background. Dopo qualche minuto visualizzerai il desktop del sistema operativo Raspberry Pi OS e la procedura di configurazione guidata, come in **Figura 2-2**. Il tuo sistema operativo è ora pronto per essere configurato, cosa che imparerai a fare nel **Capitolo 3, Come utilizzare Raspberry Pi**.



▲ **Figura 2-2:** il desktop e la procedura guidata di configurazione di Raspberry Pi OS

Capitolo 3

Come utilizzare Raspberry Pi

Scopri il sistema operativo

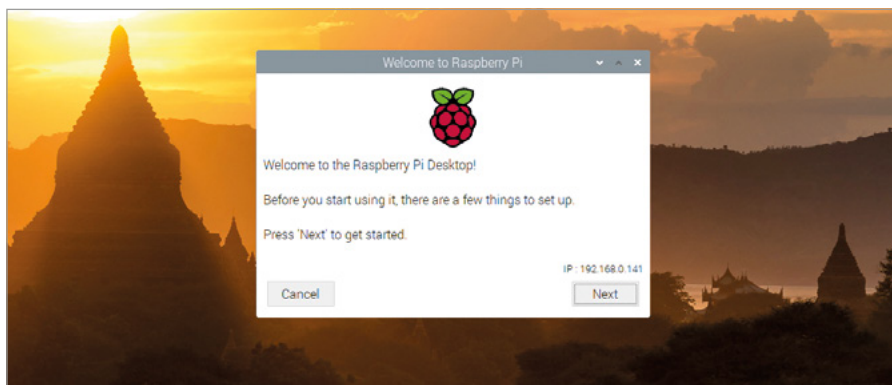


Raspberry Pi consente di utilizzare un'ampia gamma di software e diversi sistemi operativi, ossia il software principale che fa funzionare un computer. Il sistema operativo più rinomato, nonché ufficiale della Raspberry Pi Foundation, è Raspberry Pi OS. Basato su Debian Linux, è pensato appositamente per Raspberry Pi e viene fornito con una serie di software extra preinstallati e pronti all'uso.

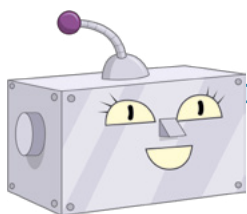
Se finora hai utilizzato solo Microsoft Windows o macOS Apple, non preoccuparti: Raspberry Pi OS è basato infatti sugli stessi elementi di finestre, icone, menu e puntatore (WIMP), così potrai acquisire rapidamente familiarità. Nel capitolo seguente apprendrai le basi del sistema e scoprirai alcuni dei software inclusi.

Welcome Wizard

La prima volta che esegui Raspberry Pi OS, viene avviata una procedura guidata chiamata Welcome Wizard (**Figura 3-1**). Questo strumento ti guiderà attraverso la modifica di alcune impostazioni in Raspberry Pi OS, per *configurare* Raspberry Pi in modo tale che sia adatto all'utilizzo che ne vuoi fare.



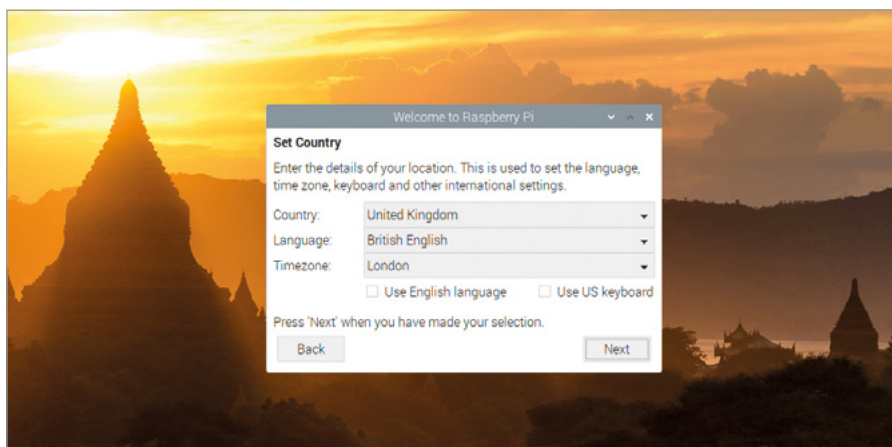
▲ **Figura 3-1: Welcome Wizard**



CHIUSURA DELLA PROCEDURA GUIDATA

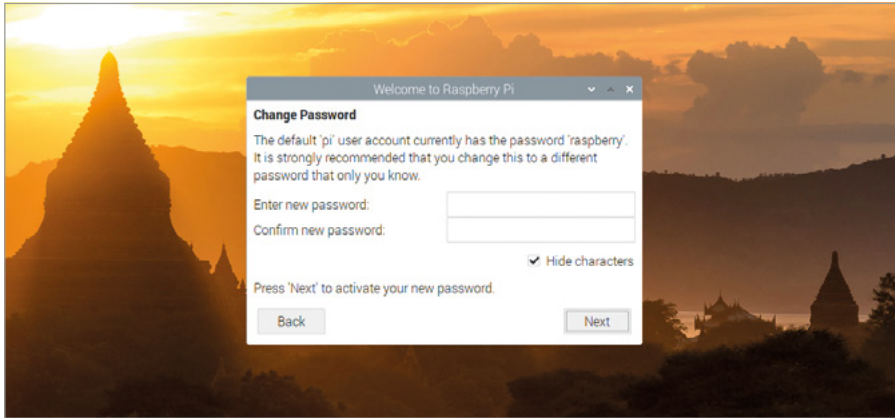
Puoi scegliere di interrompere la procedura guidata Welcome Wizard facendo clic sul pulsante Annulla, tuttavia alcune funzionalità Raspberry Pi (come la rete wireless) non funzioneranno a meno che tu non risponda alle prime domande.

Fai clic sul pulsante Next, quindi scegli il paese, la lingua e il fuso orario selezionandoli dagli elenchi a discesa (**Figura 3-2**). Se utilizzi una tastiera con layout statunitense, fai clic sulla casella per assicurarti che il layout utilizzato in Raspberry Pi OS sia quello corretto. Se vuoi che il desktop e i programmi vengano visualizzati in inglese, indipendentemente dalla lingua del tuo paese, fai clic per spuntare la casella "Use English language". Una volta terminato, fai clic su Next.



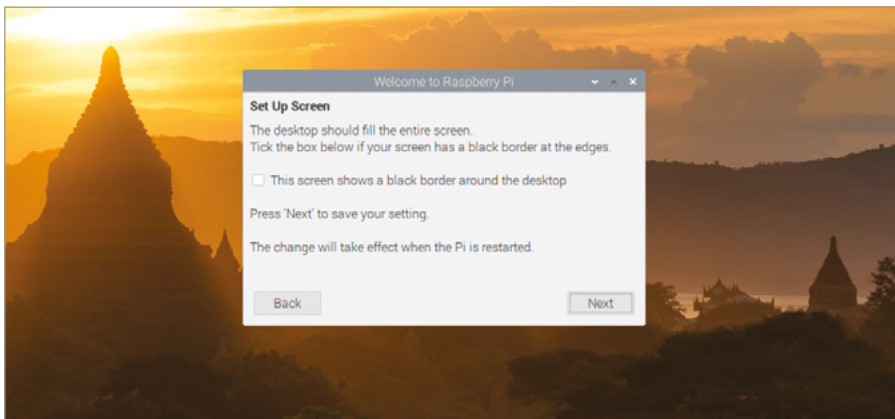
▲ **Figura 3-2: selezione della lingua e altre opzioni**

La schermata successiva ti chiederà di cambiare la password predefinita per l'utente "pi" ("raspberry"), ti consigliamo di crearne una nuova per motivi di sicurezza. Immettila nelle caselle (**Figura 3-3**). Una volta terminato, fai clic su Next.



▲ **Figura 3-3:** impostazione di una nuova password

Nella schermata successiva ti viene chiesto se è presente un bordo nero intorno alla schermata (**Figura 3-4**). Se il desktop Raspberry Pi occupa tutto il televisore o il monitor, lascia la casella non spuntata, se invece presenta un bordo nero intorno, spunta la casella. Al termine della procedura, fai clic su Next.



▲ **Figura 3-4:** controllo presenza di un bordo nero

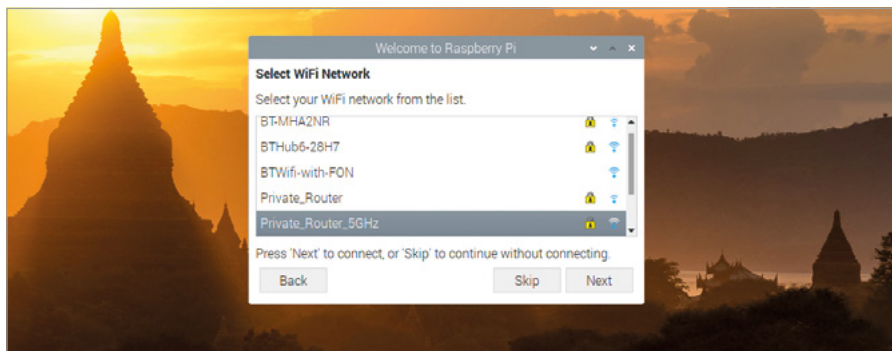
Nella schermata seguente potrai scegliere la rete Wi-Fi da un elenco (**Figura 3-5**). Scorri l'elenco delle reti con il mouse o la tastiera finché non trovi la tua, quindi selezionala e fai clic su Next. Se la tua rete wireless è sicura, come dovrebbe essere, ti verrà chiesta la password (nota anche come chiave precondivisa), che generalmente è scritta sulla scheda che



RETE WIRELESS

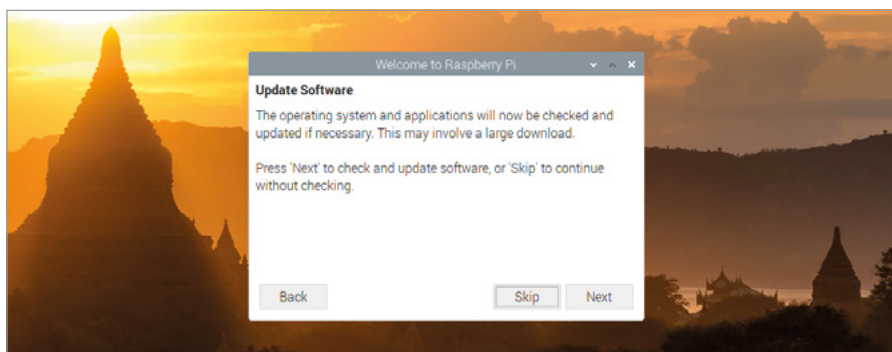
La rete wireless integrata è disponibile solo per la gamma Raspberry Pi 3, Pi 4 e Pi Zero W. Se vuoi utilizzare un altro modello di Raspberry Pi con una rete wireless, dovrai utilizzare un adattatore USB Wi-Fi.

accompagna il router o sulla parte inferiore del router. Fai clic su Next per collegarti alla rete. Se non desideri connetterti a una rete wireless, fai clic su Skip.



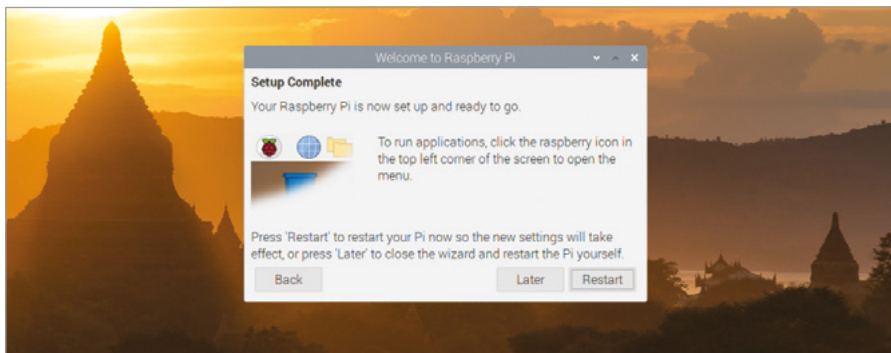
▲ **Figura 3-5:** selezione della rete wireless

Nella schermata successiva potrai controllare e installare gli aggiornamenti per Raspberry Pi OS e gli altri software presenti su Raspberry Pi (**Figura 3-6**). Raspberry Pi OS viene regolarmente aggiornato per correggere bug, aggiungere nuove funzionalità e migliorare le prestazioni. Per installare gli aggiornamenti fai clic su Next, altrimenti fai clic su Skip. Il download degli aggiornamenti può richiedere diversi minuti, quindi ti chiediamo di avere un po' di pazienza. Al termine dell'installazione, visualizzerai una finestra che indica l'avvenuto aggiornamento, quindi fai clic sul pulsante OK.



▲ **Figura 3-6:** controllo degli aggiornamenti

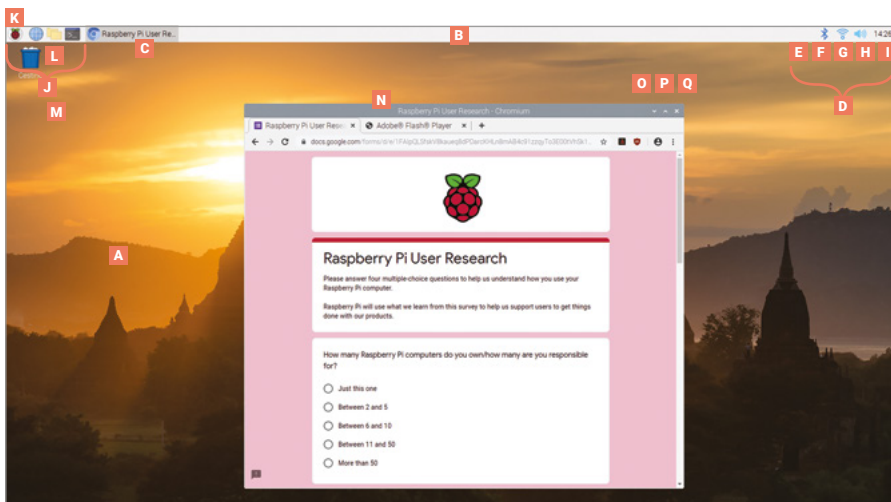
L'ultima schermata del Welcome Wizard (**Figura 3-7**) ha un compito semplice: alcune modifiche apportate verranno applicate solo al riavvio di Raspberry Pi (processo noto anche come riavvio). Se richiesto, fai clic sul pulsante Restart e Raspberry Pi si riavvierà. Al termine della procedura non visualizzerai nuovamente il Welcome Wizard e potrai iniziare a utilizzare il tuo Raspberry Pi.



▲ **Figura 3-7:** riavvio di Raspberry Pi

Il desktop

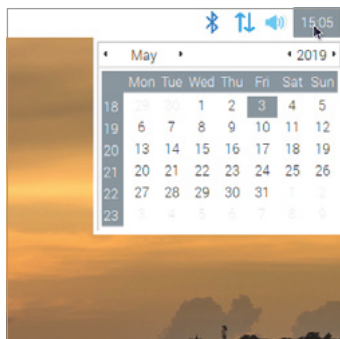
La versione di Raspberry Pi OS installata sulla maggior parte delle schede Raspberry Pi è conosciuta come "Raspberry Pi OS con desktop" relativamente all'interfaccia grafica utente (**Figura 3-8**). La maggior parte del desktop è occupata da una foto, ossia uno sfondo (**A** nella **Figura 3-8**), sulla quale visualizzerai i programmi che esegui. Nella parte superiore del desktop è presente una barra delle applicazioni (**B**), da qui potrai caricare i programmi, che verranno quindi indicati da attività (**C**) nella barra.



▲ **Figura 3-8:** il desktop di Raspberry Pi OS

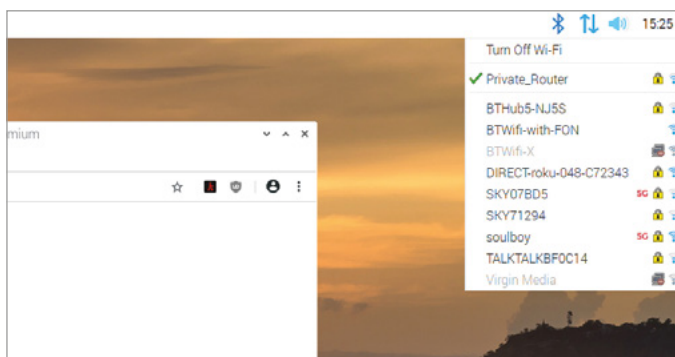
A Sfondo	G Icona di rete	M Icona unità rimovibile
B Barra delle applicazioni	H Icona volume	N Barra del titolo della finestra
C Attività	I Orologio	O Riduci a icona
D Area di notifica	J Launcher	P Ingrandisci
E Espulsione media	K Icona menu (o lampone)	Q Chiudi
F Icona Bluetooth	L Icona cestino	

Il lato destro della barra dei menu ospita l'area di notifica (D). Quando colleghi un dispositivo di archiviazione rimovibile (come ad esempio una chiavetta USB) a Raspberry Pi, vedrai un simbolo di espulsione (E): fai clic su questo simbolo per espellere e rimuovere il dispositivo in modo sicuro. L'orologio si trova a destra (I), facendovi clic sopra potrai visualizzare un calendario digitale (Figura 3-9).



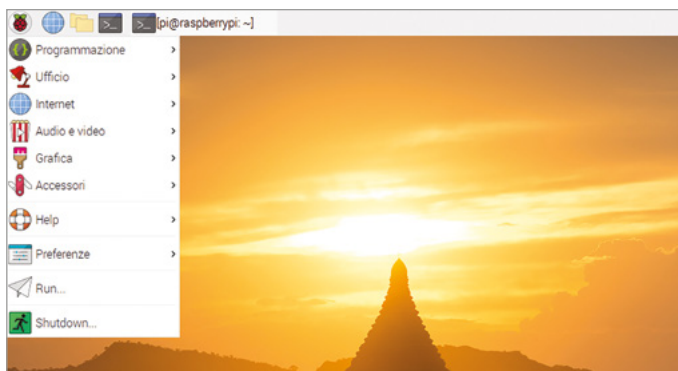
► **Figura 3-9: calendario digitale**

Accanto all'orologio è presente l'icona dell'altoparlante (H): fai clic con il tasto sinistro del mouse per regolare il volume di Raspberry Pi, oppure con il tasto destro per scegliere l'uscita audio. Accanto c'è un'icona di rete (G): se sei collegato a una rete wireless visualizzerai la potenza del segnale come una serie di barre, mentre se sei collegato a una rete cablata vedrai solo due frecce. Facendo clic sull'icona di rete visualizzerai un elenco di reti wireless presenti nelle vicinanze (Figura 3-10), mentre facendo clic sull'icona Bluetooth (F) accanto potrai connetterti a un dispositivo Bluetooth nelle vicinanze.



◀ **Figura 3-10: elenco delle reti wireless vicine**

Il menu *launcher* (J) si trova nella parte sinistra della barra dei menu, qui troverai i programmi installati con Raspberry Pi OS. Alcuni di questi sono visibili come icone di collegamento, mentre altri si trovano nel menu, che potrai richiamare facendo clic sull'icona a forma di lampone (K) a sinistra (Figura 3-11, sul retro).

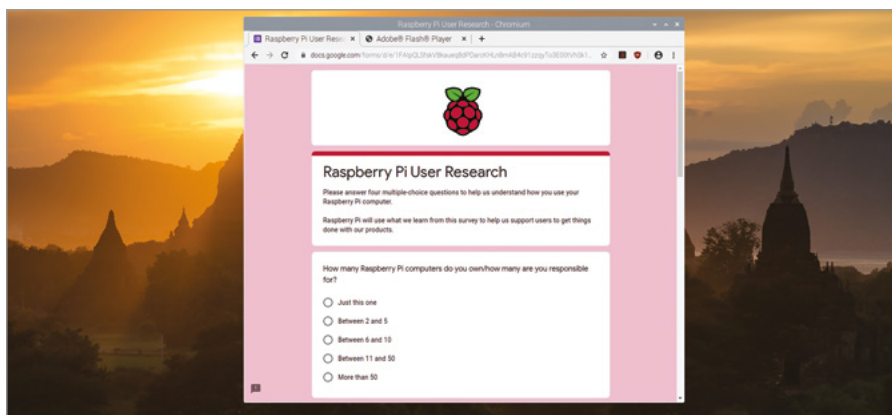


▲ **Figura 3-11:** il menu di Raspberry Pi OS

I programmi nel menu sono suddivisi in categorie descritte dai nomi. La categoria Programmazione, ad esempio, contiene software per creare i tuoi programmi, come spiegato nel **Capitolo 4, Come programmare con Scratch**, mentre i Giochi ti faranno passare il tempo. In questa guida non verranno illustrati tutti i programmi: sperimenta per conoscerli di più.

Il browser Chromium

Per fare pratica con il tuo Raspberry Pi, inizia aprendo il browser Web Chromium: fai clic sull'icona del lampone in alto a sinistra per aprire il menu, sposta il puntatore del mouse per selezionare la categoria Internet e fai clic su Chromium Web Browser per aprirlo (**Figura 3-12**).



▲ **Figura 3-12:** il browser Chromium

Se hai già usato il browser Chrome di Google su un altro computer, Chromium ti sarà immediatamente familiare. Come browser Web, Chromium ti consente di visitare siti Web, guardare video, giocare e persino di comunicare con persone in tutto il mondo su forum e siti di chat.

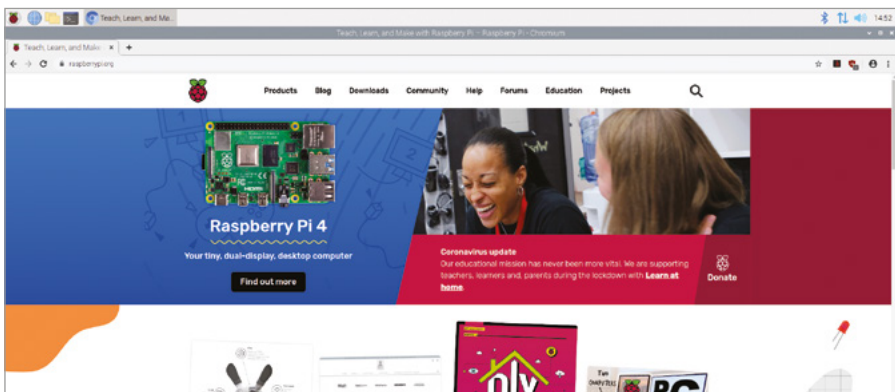
Inizia a usare Chromium ingrandendo la finestra in modo che occupi più spazio sullo schermo: cerca le tre icone in alto a destra nella barra del titolo della finestra Chromium (**N**) e fai clic sull'icona della freccia in alto al centro (**P**). Questo è il pulsante di *ingrandimento*, che consentirà alla finestra di occupare tutto lo schermo. A sinistra del pulsante per ingrandire è presente quello per *ridurre a icona*(**O**), con cui potrai nascondere la finestra fino a che non farai clic su di essa nella barra delle applicazioni nella parte superiore dello schermo. La crocetta a destra del pulsante per ingrandire serve per *chiudere*(**Q**) e fa esattamente quello che ci si aspetta: chiude la finestra.



CHIUDI E SALVA

Chiudere una finestra prima di aver salvato il lavoro svolto non è una buona idea, infatti non tutti i programmi avvisano che è necessario salvare quando fai clic sul pulsante di chiusura.

Fai clic nella barra degli indirizzi nella parte superiore della finestra di Chromium (la grande barra bianca con una lente d'ingrandimento sul lato sinistro) e digita **www.raspberrypi.org**, quindi premi il tasto **INVIO** sulla tastiera. Verrà caricato il sito Web Raspberry Pi (**Figura 3-13**). Nella barra puoi digitare anche termini di ricerca: prova a cercare "Raspberry Pi", "Raspberry Pi OS" o "Educational Computing".



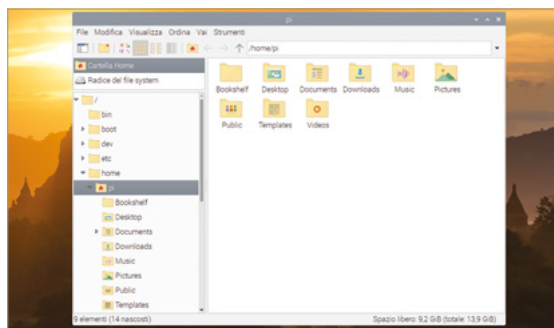
▲ **Figura 3-13:** caricamento del sito Web Raspberry Pi in Chromium

Al primo avvio di Chromium potrebbero caricarsi alcune *schede* nella parte superiore della finestra. Per passare a un'altra scheda, fai clic su di essa; per chiudere una scheda senza chiudere Chromium, fai clic sulla crocetta in alto a destra nella scheda che vuoi chiudere. Per aprire più siti Web senza dover passare da una finestra di Chromium all'altra, fai clic sul pulsante della scheda a destra dell'ultima scheda aperta oppure tieni premuto il tasto **CTRL** sulla tastiera, quindi premi il tasto **T** prima di rilasciare **CTRL**.

Quando hai finito di usare Chromium, fai clic sul pulsante di chiusura nell'angolo in alto a destra della finestra.

Il File Manager

I file che salvi, ad esempio programmi, video, immagini, vengono memorizzati nella *home directory*. Per visualizzare la home directory, fai clic sull'icona a forma di lampone per aprire il menu, sposta il puntatore del mouse per selezionare Accessori, quindi fai clic sul File Manager per aprirlo (**Figura 3-14**).



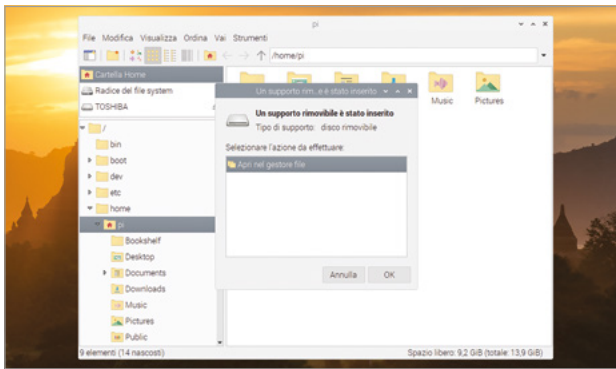
◀ **Figura 3-14: il File Manager**

Con il File Manager potrai sfogliare i file e le cartelle (*directory*) presenti nella scheda microSD di Raspberry Pi, nonché in qualsiasi dispositivo di archiviazione rimovibile (come chiavette USB) collegato alle porte USB di Raspberry Pi. Alla prima apertura, accede automaticamente alla home directory. Qui troverai una serie di cartelle, o *sottodirectory*, organizzate in categorie, proprio come il menu. Le sottodirectory principali sono:

- **Bookshelf:** contiene le copie digitali di libri, riviste e altre pubblicazioni della Raspberry Pi Press, inclusa una copia della *Guida introduttiva* (versione inglese). Puoi leggerli e scaricarne altri con l'applicazione Bookshelf nella sezione Guida del menu.
- **Desktop:** questa cartella contiene ciò che vedi quando carichi Raspberry Pi OS. Se salvi un file qui, lo vedrai sul desktop e sarà facile trovarlo e aprirlo.
- **Documents:** la cartella in cui memorizzerai la maggior parte dei file che creerai, dai racconti alle ricette.
- **Downloads:** quando scarichi un file da Internet con il browser Web Chromium, viene automaticamente salvato in Downloads.
- **Music:** qui potrai memorizzare i brani musicali creati o importati su Raspberry Pi.
- **Pictures:** questa cartella è dedicata alle immagini o, in termini tecnici, *file di immagini*.
- **Public:** mentre gli altri file sono privati, tutto ciò che salverai in Public sarà disponibile per altri utenti Raspberry Pi, anche se hanno il loro nome utente e password.

- **Templates:** questa cartella contiene tutti i modelli e i documenti vuoti con un layout o una struttura di base predefiniti creati da te o installati dalle tue applicazioni.
- **Videos:** una cartella per i video, nonché la prima cartella in cui cercheranno i programmi di riproduzione video.

La finestra File Manager è divisa in due riquadri: quello di sinistra mostra le directory presenti sul tuo Raspberry Pi, mentre quello destra i file e le sottodirectory della directory selezionata a sinistra. Quando colleghi un dispositivo di archiviazione rimovibile alla porta USB di Raspberry Pi, si aprirà una finestra che chiede se desideri aprirlo in File Manager (**Figura 3-15**). Fai clic su OK per visualizzarne file e directory.



◀ **Figura 3-15:** inserimento di un dispositivo di archiviazione rimovibile

I file possono essere facilmente copiati tra la scheda microSD di Raspberry Pi e un dispositivo rimovibile: con l'home directory e il dispositivo rimovibile aperti in finestre File Manager separate, sposta il puntatore del mouse sul file che vuoi copiare, fai clic e tieni premuto il tasto sinistro del mouse, quindi trascina il puntatore sull'altra finestra e rilascia il tasto del mouse (**Figura 3-16**, sul retro). Questa operazione è chiamata *trascinamento*.

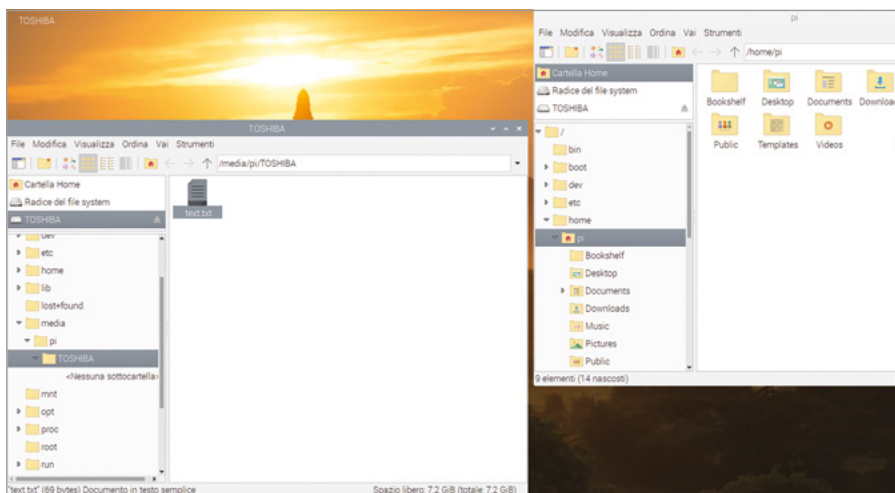
In alternativa fai clic una volta sul file, quindi sul menu Modifica e su Copia, successivamente fai clic sull'altra finestra, sul menu Modifica e su Incolla.

L'opzione Sposta, disponibile anche nel menu Modifica, è simile, tranne per il fatto che cancella il file dalla sua posizione originale dopo averlo copiato. Entrambe le opzioni possono essere utilizzate anche attraverso le seguenti combinazioni di tasti di scelta rapida: **CTRL+C** per copiare o **CTRL+X** per tagliare e **CTRL+V** per incollare.



TASTI DI SCELTA RAPIDA

Quando vedi un tasto di scelta rapida come **CTRL+C**, significa che devi tenere premuto il primo tasto sulla tastiera (**CTRL**), quindi premere il secondo (**C**) e rilasciarli.



▲ **Figura 3-16:** trascinamento di un file

Quando hai finito di sperimentare, chiudi il File Manager facendo clic sul pulsante di chiusura della finestra, in alto a sinistra. Se hai più di una finestra aperta, chiudile tutte. Se hai collegato un dispositivo di archiviazione rimovibile a Raspberry Pi, espellilo facendo clic sul pulsante di espulsione in alto a destra dello schermo, quindi cerca il dispositivo nell'elenco e selezionalo prima di scollegarlo.



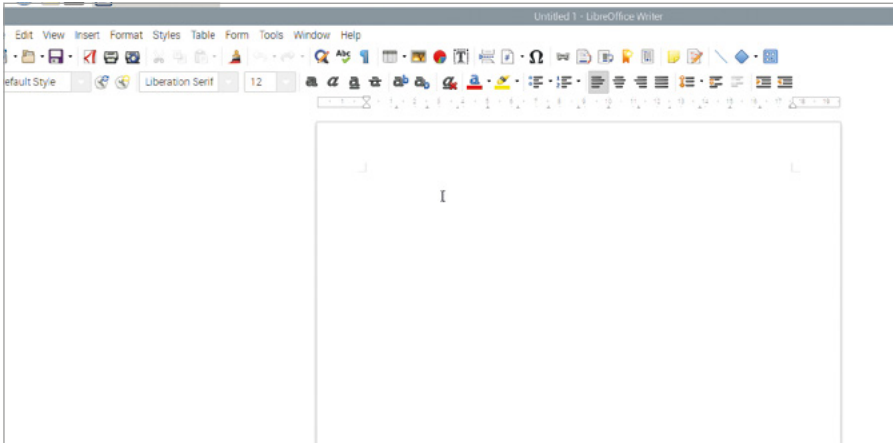
ESPULSIONE DEI DISPOSITIVI

Utilizza sempre il pulsante di espulsione prima di scollegare un dispositivo di archiviazione esterno, altrimenti i file presenti su di esso potrebbero corrompersi e diventare inutilizzabili.

Suite di produttività LibreOffice

Per avere un altro assaggio di ciò che Raspberry Pi può fare, fai clic sull'icona del menu a forma di lampone, sposta il puntatore del mouse su Ufficio e fai clic su LibreOffice Writer. In questo modo verrà aperto l'elaboratore di testi di LibreOffice (**Figura 3-17**), una popolare suite di produttività simile a Microsoft Office o Google Docs.

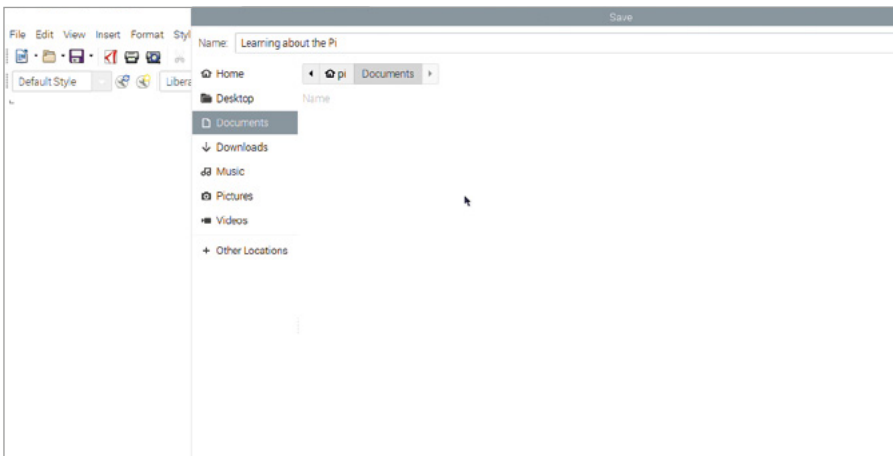
Nota: la suite LibreOffice potrebbe non essere installata su tutte le immagini del sistema operativo Raspberry Pi; qualora non fosse presente, utilizza lo strumento Software consigliato (vedi pagina 49) per installarla.



▲ **Figura 3-17:** programma LibreOffice Writer

Con un elaboratore di testi non solo potrai scrivere documenti, ma anche formattarli modificandone carattere, colore, dimensioni, aggiungendo effetti e persino inserendo immagini, grafici, tabelle e altri contenuti. Potrai anche controllare il tuo lavoro per verificare la presenza di errori: l'elaboratore di testi, infatti, rileva gli errori ortografici e grammaticali rispettivamente in rosso e verde mentre digiti.

Inizia a scrivere un paragrafo su ciò che hai imparato finora su Raspberry Pi e il suo software. Sperimenta con le diverse icone nella parte superiore della finestra per vedere cosa fanno: vedi se puoi ingrandire i caratteri e cambiarne il colore. Se non sei sicuro di come fare, sposta il puntatore del mouse su ogni icona, vedrai una descrizione che ti indica le sue funzioni. Quando sei soddisfatto, fai clic sul menu File e sull'opzione Save per salvare il tuo lavoro (**Figura 3-18**). Assegna un nome e fai clic sul pulsante Save.



▲ **Figura 3-18:** salvataggio di un documento



SALVA IL TUO LAVORO

Prendi l'abitudine di salvare il tuo lavoro, anche se non l'hai ancora finito, così avrai meno problemi qualora si verifici un'interruzione di corrente nel bel mezzo di un lavoro.

LibreOffice Writer è solo una parte della suite di produttività LibreOffice. Nella categoria del menu Ufficio, oltre a LibreOffice Writer troverai:

- **LibreOffice Base:** un database, ossia uno strumento per memorizzare le informazioni, cercarle velocemente e analizzarle.
- **LibreOffice Calc:** un foglio di calcolo, ovvero uno strumento per la gestione dei numeri e la creazione di grafici e diagrammi.
- **LibreOffice Draw:** un programma di illustrazione, ossia uno strumento per la creazione di immagini e diagrammi.
- **LibreOffice Impress:** un programma di presentazione per la creazione di slide e presentazioni.
- **LibreOffice Math:** un editor di formule, ovvero uno strumento per creare formule matematiche correttamente formattate da utilizzare in altri documenti.

LibreOffice è disponibile anche per altri computer e sistemi operativi. Se ti piace usarlo su Raspberry Pi, puoi scaricarlo gratuitamente da libreoffice.org e installarlo su qualsiasi computer Microsoft Windows, macOS Apple o Linux.

Per saperne di più sull'utilizzo di LibreOffice, fai clic sul menu Guida. Altrimenti chiudi LibreOffice Writer, facendo clic sul pulsante di chiusura nell'angolo in alto a destra della finestra.



ISTRUZIONI PER DUBBI E PROBLEMI

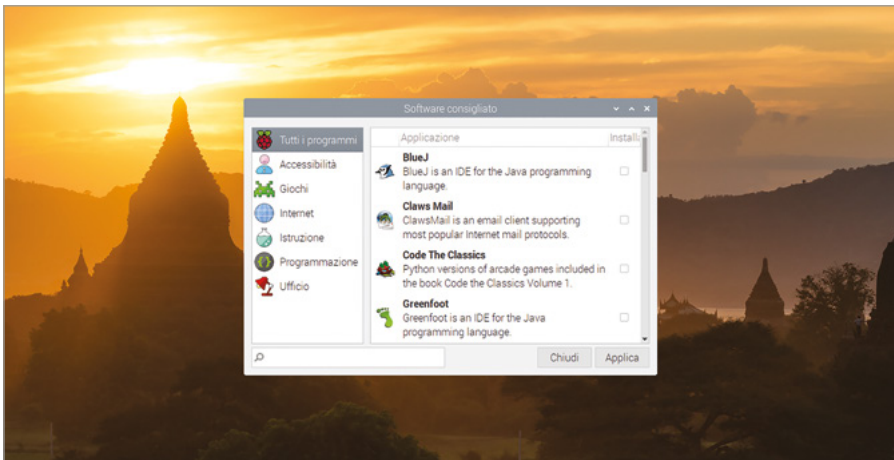
La maggior parte dei programmi include un menu di guida con tutte le informazioni necessarie: dalle spiegazioni sul programma alle guide su come usarlo. Se durante l'uso di un programma ti senti disorientato o riscontri dei problemi, apri il menu Guida per riorientarti.

Lo strumento Software consigliato

Raspberry Pi OS viene fornito ed è compatibile con un'ampia gamma di software. Nello strumento Software consigliato è disponibile una selezione dei migliori programmi.

Per utilizzarlo è necessaria una connessione Internet. Se il tuo Raspberry Pi è collegato, fai clic sull'icona a forma di lampone del menu, sposta il puntatore del mouse su Preferenze e fai clic su Software consigliato. Lo strumento si carica, quindi inizia a scaricare le informazioni sul software disponibile.

Dopo qualche secondo visualizzerai un elenco di pacchetti software compatibili (**Figura 3-19**). Sono organizzati in varie categorie, proprio come il software nel menu. Fai clic su una categoria nel riquadro a sinistra per visualizzare i software di quella categoria, oppure fai clic su Tutti i programmi per visualizzare tutto.

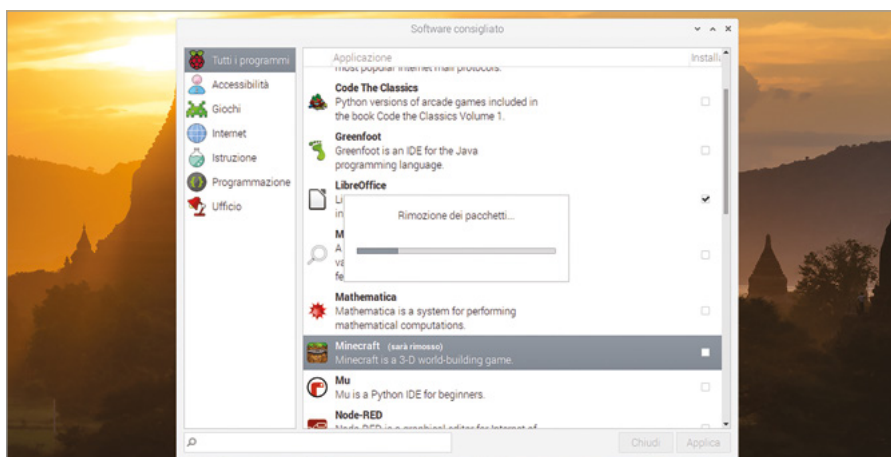


▲ **Figura 3-19:** lo strumento Software consigliato

Se accanto a un software è presente un segno di spunta, è già installato su Raspberry Pi. Altrimenti fai clic sulla casella di controllo accanto a esso per aggiungere un segno di spunta e contrassegnarlo per l'installazione. Puoi contrassegnare tutti i software che desideri prima di installarli in una volta sola, tuttavia se utilizzi una scheda microSD con meno memoria rispetto a quanto consigliato, potrebbe non esserci abbastanza spazio per installare tutto.

La procedura per disinstallare il software è simile: ti basterà rimuovere il segno di spunta accanto al software. Se hai fatto un errore o cambiato idea, ti basterà fare clic per aggiungere di nuovo il segno di spunta.

Una volta selezionato il software, fai clic sul pulsante OK per avviare il processo di installazione o disinstallazione (**Figura 3-20**, sul retro). Dopo aver scaricato e installato il nuovo software scelto, visualizzerai una finestra di dialogo: fai clic su OK per chiudere lo strumento Software consigliato.

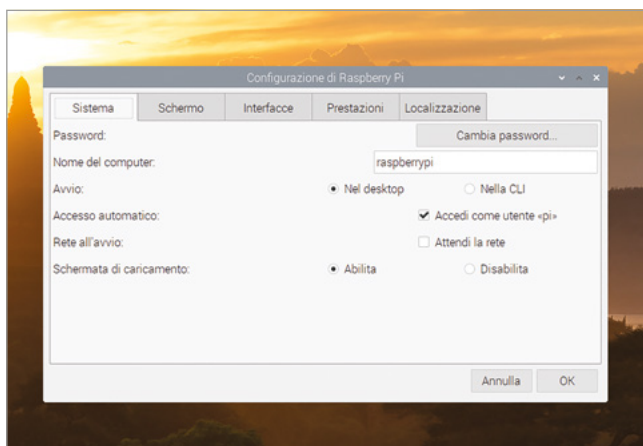


▲ **Figura 3-20:** disinstallazione del software

Nella categoria Preferenze del menu Raspberry Pi OS potrai trovare lo strumento di aggiunta/rimozione software, uno strumento alternativo per installare o disinstallare software. Offre una selezione più ampia di software, ma non è stato controllato dalla Raspberry Pi Foundation.

Strumento Configurazione di Raspberry Pi

L'ultimo programma che prenderemo in considerazione in questo capitolo è lo strumento Configurazione di Raspberry Pi. Questo strumento è simile alla procedura guidata Welcome Wizard utilizzata all'inizio, infatti ti consente di modificare alcune impostazioni in Raspberry Pi OS. Fai clic sull'icona a forma di lampone, sposta il puntatore del mouse per selezionare la categoria Preferenze, quindi fai clic su Configurazione di Raspberry Pi per aprire (**Figura 3-21**).



◀ **Figura 3-21:** strumento Configurazione di Raspberry Pi

Lo strumento è suddiviso in cinque schede. La prima è Sistema: ti consente di modificare la password del tuo account, impostare un nome host (ossia il nome usato da Raspberry Pi nella rete locale cablata o wireless) e modificare altre impostazioni. Tuttavia non dovrebbero essere necessarie modifiche per la maggior parte delle impostazioni in questa scheda. Fai clic sulla scheda Schermo per passare alla categoria successiva. Qui, se necessario, potrai modificare le impostazioni di visualizzazione dello schermo per adattarle al televisore o al monitor.



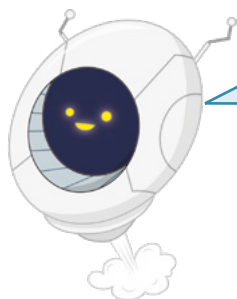
ALTRI DETTAGLI

Questa breve panoramica ti consente di abituarti allo strumento. Per maggiori informazioni sulle configurazioni, consulta l'**Appendice E, strumento Configurazione di Raspberry Pi**.

Nella scheda Interfaccia sono presenti diverse impostazioni disattivate. Modificalle solo se aggiungi un nuovo hardware, come il modulo fotocamera Raspberry Pi e solamente se indicato dal produttore dell'hardware. Fanno eccezione: SSH, che attiva un "Secure Shell" e ti consente di collegarti a Raspberry Pi da un altro computer presente sulla tua rete tramite un client SSH; VNC, che abilita un "Virtual Network Computer" per consentirti di visualizzare e controllare il desktop di Raspberry Pi OS da un altro computer tramite un VNC e Remote GPIO, che ti consente di utilizzare pin GPIO Raspberry Pi (che imparerai a conoscere nel **Capitolo 6, Physical computing con Scratch e Python**) da un altro computer nella tua rete.

Fai clic sulla scheda Prestazioni per visualizzare la quarta categoria. Qui potrai impostare la memoria utilizzata dall'unità di elaborazione grafica (GPU) Raspberry Pi e, in alcuni modelli, migliorare le prestazioni di Raspberry Pi tramite il processo di *overclocking*. Non modificare queste impostazioni se non sai come fare.

Infine, fai clic sulla scheda Localizzazione per visualizzare l'ultima categoria. Qui potrai modificare le impostazioni locali, che controllano, fra le varie impostazioni, la lingua usata in Raspberry Pi OS, la modalità di visualizzazione dei numeri, il fuso orario, il layout della tastiera e le impostazioni del Paese per il Wi-Fi. Per ora ti basta fare clic su Annulla per chiudere lo strumento senza effettuare modifiche.



ATTENZIONE

Le frequenze radio Wi-Fi consentite possono variare a seconda del Paese. Se nello strumento Configurazione di Raspberry Pi imposti un Paese diverso da quello impostato per il Wi-Fi, potresti riscontrare problemi nella connessione alle tue reti e potrebbe anche essere illegale in base alle leggi di licenza delle frequenze radio, quindi seleziona il tuo Paese.

Spegnimento

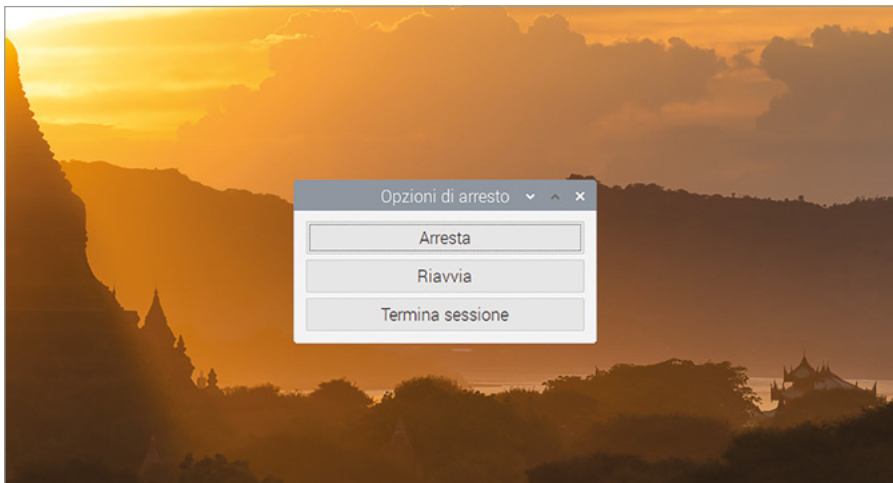
Ora che hai scoperto alcune funzioni del desktop di Raspberry Pi OS è arrivato il momento di apprendere una funzione importante: come spegnere correttamente il tuo Raspberry Pi. Come in ogni computer, Raspberry Pi mantiene i file su cui stai lavorando nella *memoria volatile*, ossia una memoria che viene svuotata una volta spento il sistema. Salva i documenti che stai creando nella scheda microSD, così non perderai nulla, perché verranno spostati dalla memoria volatile a quella *non volatile*.

I documenti su cui stai lavorando non sono gli unici file aperti. Il sistema operativo Raspberry Pi OS stesso mantiene una serie di file aperti mentre è in esecuzione: staccando il cavo di alimentazione dal tuo Raspberry Pi mentre questi sono ancora aperti potrebbe causarne la corruzione e la necessità di reinstallare il sistema operativo.

Per evitare che questo accada, assicurati che Raspberry Pi OS salvi tutti i suoi file e sia pronto per essere spento, un processo noto come *spegnimento* del sistema operativo.

Fai clic sull'icona a forma di lampone nell'angolo in alto a sinistra del desktop, quindi su Arresta. Visualizzerai una finestra con tre opzioni (**Figura 3-22**): Arresta, Riavvia, Termina sessione. Arresta sarà l'opzione che userai di più: facendovi clic sopra Raspberry Pi OS chiuderà tutti i software e file aperti, quindi spegnerà il sistema. Quando il display diventa nero, attendi qualche secondo lo spegnimento della luce verde lampeggiante su Raspberry Pi, successivamente potrai spegnere l'alimentazione.

Per riaccendere Raspberry Pi, scollega e ricollega il cavo di alimentazione, o attiva la corrente della presa a muro.



▲ **Figura 3-22:** spegnimento di Raspberry Pi

Il riavvio è simile allo spegnimento, infatti chiude tutto, ma invece di spegnere Raspberry Pi lo riavvia, è quasi come se avessi scelto lo spegnimento, poi scollegato e ricollegato il cavo di alimentazione. Dovrai riavviare quando apporti alcune modifiche che richiedono un riavvio del sistema operativo (come l'installazione di alcuni aggiornamenti del software principale) o se si è verificato qualche problema con un software (*crash*) che ha reso inutilizzabile il sistema.

Infine, Termina sessione è utile solo se hai più di un account utente sul tuo Raspberry Pi: chiude tutti i programmi aperti e ti porta a una schermata di accesso in cui ti vengono richiesti nome utente e password. Se hai premuto accidentalmente Termina sessione e vuoi riaccedere, digita "pi" come nome utente e la password scelta nella procedura guidata Welcome Wizard illustrata all'inizio del capitolo.



ATTENZIONE

Non staccare mai il cavo di alimentazione da Raspberry Pi senza averlo prima spento. Il sistema operativo potrebbe corrompersi e potresti perdere i file creati o scaricati.

Capitolo 4

Come programmare con Scratch 3

Scopri come iniziare a programmare con Scratch, il linguaggio di programmazione a blocchi

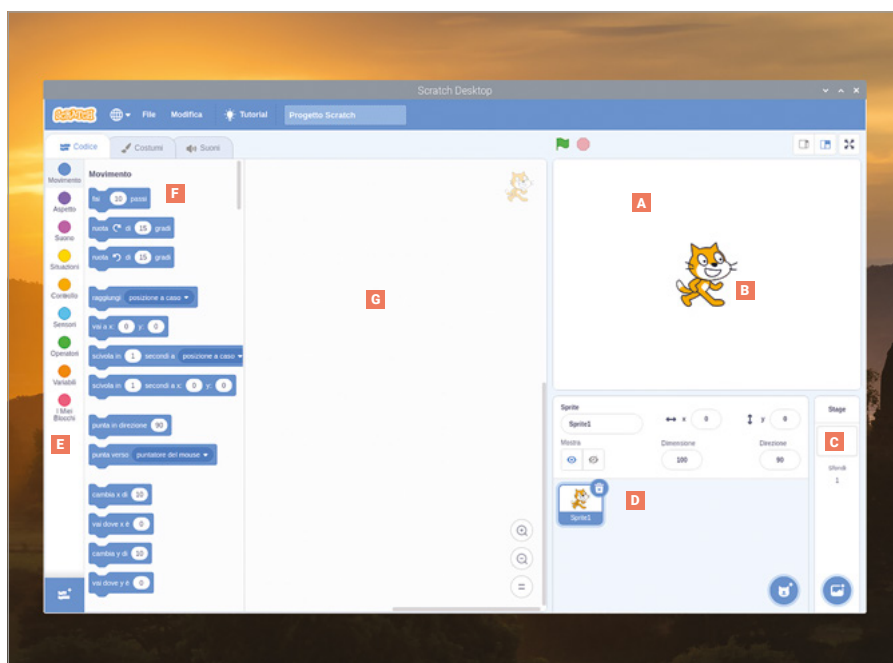


Con Raspberry Pi non solo potrai utilizzare i software creati da altre persone, ma potrai creare il tuo software seguendo la tua immaginazione. Raspberry Pi è la piattaforma perfetta per creare e sperimentare, sia che tu abbia già esperienza nella programmazione sia che ti avvicini per la prima volta a questa disciplina.

Scratch, un linguaggio di programmazione di tipo grafico sviluppato dal Massachusetts Institute of Technology (MIT), è ciò che rende la programmazione su Raspberry Pi così intuitiva. Mentre con i linguaggi di programmazione tradizionali è necessario scrivere istruzioni testuali che dovranno essere eseguite dal computer, all'incirca come scrivere la ricetta da seguire per preparare una torta, con Scratch potrai costruire il tuo programma passo dopo passo mediante blocchi: parti di codice già scritte nascoste in pezzi di puzzle definiti da colori.

Scratch è un ottimo primo linguaggio per codificatori in erba grandi e piccoli, ma non farti ingannare dalla grafica: infatti è un ambiente di programmazione potente e pienamente funzionale che ti consentirà di creare qualsiasi cosa, da semplici giochi e animazioni fino a più complessi progetti di robotica interattiva.

L'interfaccia Scratch 3



- A Area Stage:** i tuoi sprite si muovono in quest'area in base alle istruzioni del programma.
- B Sprite:** sono i personaggi o oggetti che controlli in Scratch e che collocherai nell'area stage.
- C Controlli Stage:** ti consentono di cambiare lo stage, aggiungendo anche immagini come sfondo.
- D Elenco di Sprite:** in questa sezione della finestra vedrai tutti gli sprite che hai creato o caricato in Scratch.
- E Palette di blocchi:** qui vengono visualizzati tutti i blocchi disponibili, suddivisi in categorie contraddistinte per colore.

VERSIONI DI SCRATCH

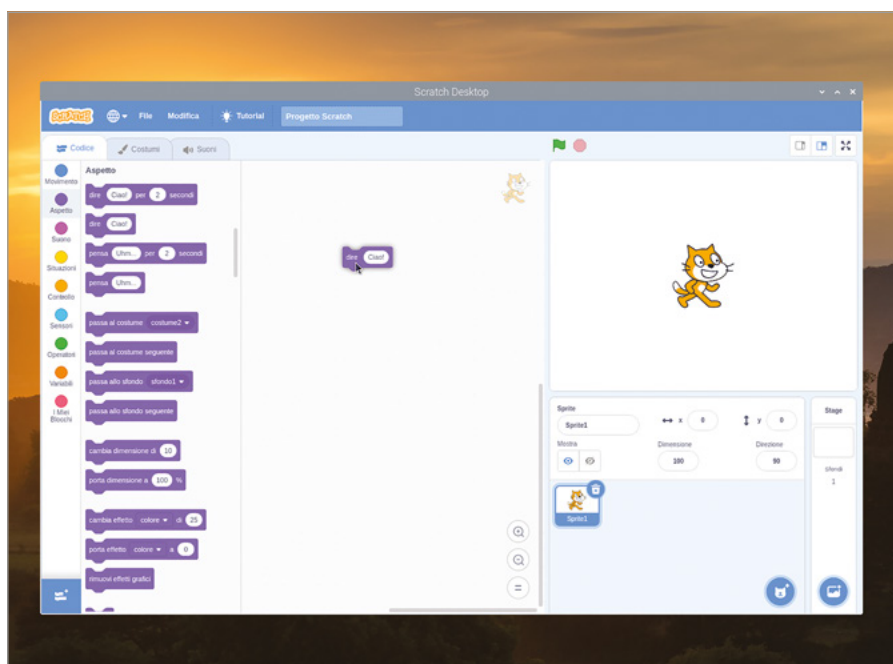
Al momento della stesura di questo libro, Raspberry Pi OS comprendeva tre versioni di Scratch: (Scratch 1, 2 e 3), che si trovano nella sezione Programmazione del menu Raspberry Pi OS. Questo capitolo illustra Scratch 3. Scratch 3 funziona solo su Raspberry Pi 4. Se desideri utilizzare Scratch 2, tieni presente che non è compatibile con Raspberry Pi Zero, modello A, A+, B o B+.

- F Blocchi:** parti di codice già scritte che ti consentiranno di creare il tuo programma.
- G Area codice:** qui potrai creare il tuo programma trascinando i blocchi dalle palette per formare degli script.

Il tuo primo programma Scratch: ciao a tutti!


Scratch 3 si apre come qualsiasi altro programma su Raspberry Pi: fai clic sull'icona a forma di lampone per aprire il menu Raspberry Pi OS, sposta il cursore sulla sezione Programmazione e fai clic su Scratch 3. L'interfaccia utente di Scratch 3 si aprirà dopo pochi secondi.

Scratch è diverso dalla maggior parte dei programmi di programmazione, infatti non dovrai dire al computer cosa fare tramite istruzioni scritte. Inizia facendo clic sulla categoria Aspetto nella palette dei blocchi a sinistra della finestra Scratch. Vedrai tutti i blocchi disponibili nella categoria (di colore viola). Trova il blocco **dire Ciao!**, fai clic su di esso e tieni premuto il pulsante sinistro del mouse, quindi trascinalo nell'area codice al centro della finestra Scratch prima di rilasciarlo (**Figura 4-1**).



▲ **Figura 4-1:** trascinamento del blocco nell'area codice

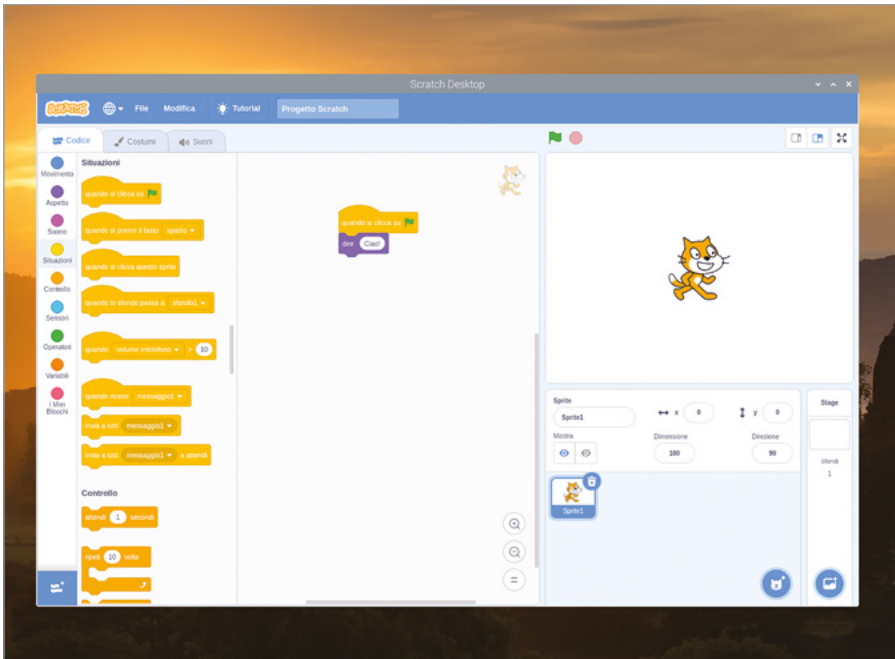
Osserva la forma del blocco che hai appena trascinato: nella parte superiore presenta un incavo e una sporgenza in quella inferiore. Questo significa che sopra e sotto al blocco che hai trascinato andranno collocati altri blocchi, proprio come in un puzzle. In questo programma i blocchi sopra sono *trigger*.

Fai clic sulla categoria Situazioni della palette dei blocchi, color oro, quindi fai clic e trascina il blocco **quando si clicca su** , o blocco *cappello*, nell'area codice. Posizionalo in modo che la parte sporgente combaci con l'incavo nella parte superiore del blocco **dire Ciao!** fino a quando non vedi un contorno bianco, poi rilascia il pulsante del mouse. È sufficiente che i

blocchi siano abbastanza vicini e si incastreranno come due pezzi di un puzzle. Qualora non si incastrino, fai clic e tieni premuto per modificare la posizione finché non si incastrano.

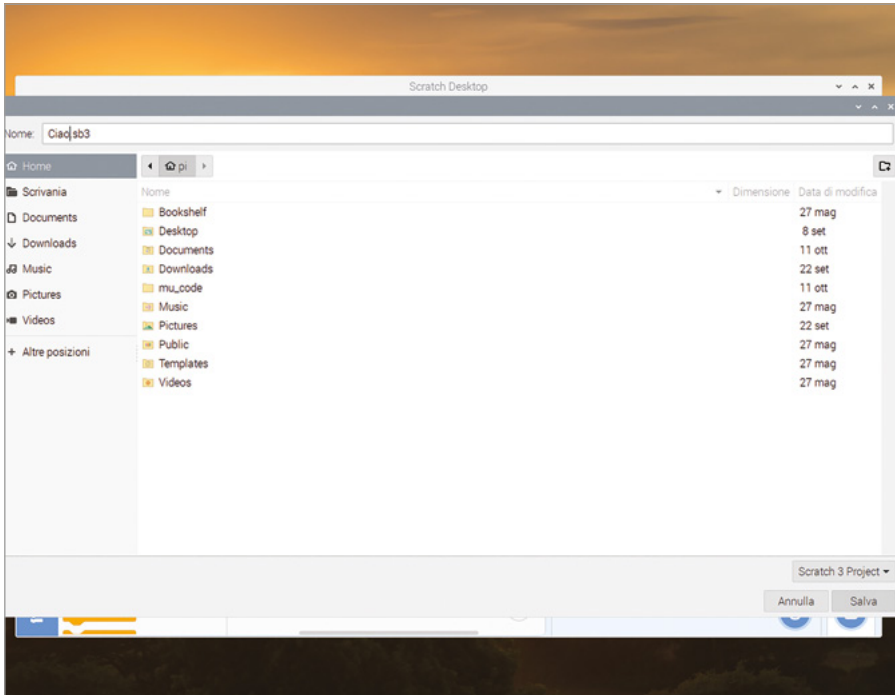


Il tuo programma è completo. Per farlo funzionare o eseguirlo, fai clic sulla bandiera verde in alto a sinistra nell'area stage. Se hai programmato correttamente, lo sprite del gatto nell'area stage ti saluterà dicendoti "Ciao!" (**Figura 4-2**). Complimenti per la tua prima animazione!



▲ **Figura 4-2:** fai clic sulla bandiera verde sullo stage e il gatto ti dirà "Ciao!"

Prima di andare avanti, assegna un nome al tuo progetto e salvalo. Fai clic sul menu File, quindi su "Salva sul tuo computer". Digita un nome e fai clic sul pulsante Salva (**Figura 4-3**).



▲ **Figura 4-3:** salva il tuo progetto con un nome inconfondibile




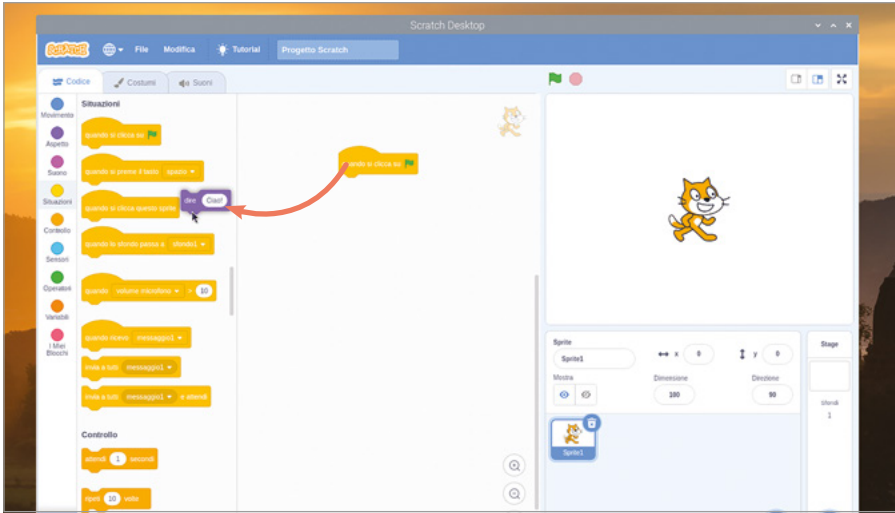
COSA PUÒ DIRE?

Alcuni blocchi in Scratch possono essere cambiati. Prova a fare clic sulla parola "Ciao!" e digita qualcos'altro, quindi fai di nuovo clic sulla bandiera verde. Cosa succede sullo stage?

Passaggi successivi: creazione di sequenze

Il tuo progetto ha due blocchi, ma comprende una sola istruzione, ossia dire "Ciao!" ogni volta che viene fatto clic sulla bandiera e il programma viene eseguito. Per fare di più, bisogna approfondire la *creazione di sequenze*. I programmi per computer, nella loro semplicità, sono un elenco di istruzioni, proprio come una ricetta. Ogni istruzione prosegue dall'ultima in una progressione logica nota come *sequenza lineare*.

Inizia facendo clic e trascinando il blocco **dire Ciao!** dall'area codice alla palette dei blocchi (**Figura 4-4**). In questo modo il blocco verrà rimosso dal tuo programma e rimarrà solo il blocco di trigger **quando si clicca su** .



▲ **Figura 4-4:** per eliminare un blocco trascinabile fuori dall'area codice

Fai clic sulla categoria Movimento nella palette dei blocchi, quindi fai clic e trascina il blocco **fai 10 passi** in modo che si incastrino sotto il blocco trigger nell'area codice. Come suggerisce il nome, fa muovere lo sprite (il gatto) di un certo numero di passi nella direzione verso cui è rivolto.



Aggiungi altre istruzioni al tuo programma per realizzare una sequenza. Fai clic sulla categoria Suono (rosa) nella palette dei blocchi, quindi fai clic e trascina il blocco **riproduci suono Mia e attendi la fine** in modo che si incastrino sotto il blocco **fai 10 passi**. Continua: fai di nuovo clic sulla categoria Movimento e trascina un altro blocco **fai 10 passi** sotto il blocco Suono, ma questa volta fai clic su "10" per selezionarlo e digita "-10" per creare un blocco **fai -10 passi**.



Fai clic sulla bandiera verde sullo stage per eseguire il programma. Vedrai il gatto muoversi verso destra, fare un miagolio (per sentirlo assicurati di aver collegato gli altoparlanti o le cuffie) e tornare all'inizio. Fai di nuovo clic sulla bandiera e il gatto ripeterà le azioni.

Complimenti! Hai creato una sequenza di istruzioni che viene eseguita da Scratch dall'inizio alla fine. Scratch eseguirà un'istruzione della sequenza alla volta, tuttavia lo fa molto velocemente: prova a cancellare il blocco **riproduci suono Miao e attendi la fine** facendo clic e trascinando il blocco successivo **fai -10 passi** per staccarlo, trascina quindi il blocco **riproduci suono Miao e attendi la fine** nella palette dei blocchi poi sostituiscilo con **riproduci suono Miao** prima di trascinare il blocco **fai -10 passi** in fondo al programma.



Fai clic sulla bandiera verde per eseguire di nuovo il programma e lo sprite gatto sembrerà non muoversi. In realtà si sta muovendo, ma torna alla posizione di partenza talmente velocemente che sembra fermo. Il blocco **riproduci suono Miao** infatti non attende che il suono termini prima di proseguire con il passaggio successivo. Siccome Raspberry Pi elabora velocemente, l'istruzione successiva viene eseguita prima che sia possibile vedere il movimento dello sprite gatto. C'è un altro modo per ovviare a questo fatto, oltre a utilizzare il blocco **riproduci suono Miao e attendi la fine**: fai clic sulla categoria gialla Controllo della palette dei blocchi, quindi fai clic e trascina il blocco **attendi 1 secondi** tra il blocco **riproduci suono Miao** e il blocco successivo **fai -10 passi**.



Fai clic sulla bandiera verde per eseguire il tuo programma un'ultima volta, e vedrai che il gatto attenderà un secondo dopo essersi spostato a destra e prima di tornare a sinistra. Questo viene chiamato *ritardo* ed è importante per controllare la durata della sequenza di istruzioni.



SFIDA: AGGIUNGI ALTRI PASSAGGI

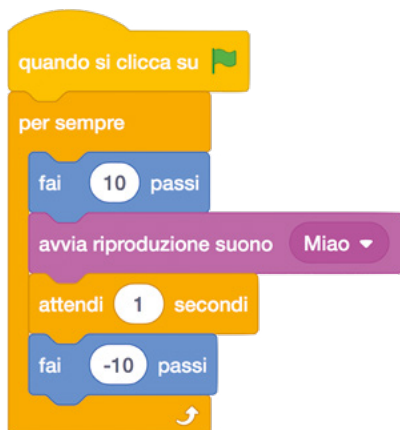


Prova ad aggiungere altri passaggi nella sequenza e a modificare i valori in quelli già presenti. Cosa succede quando il numero di passaggi in un blocco di mosse non corrisponde a quello di un altro? Cosa succede se si prova a riprodurre un suono mentre un altro suono è ancora in esecuzione?

Ripetere la sequenza

La sequenza che hai creato finora viene eseguita una sola volta: quando fai clic sulla bandiera verde, lo sprite gatto si muove e miagola e il programma si ferma finché non fai di nuovo clic sulla bandierina verde. Tuttavia non deve fermarsi, infatti Scratch presenta anche un blocco Controllo *ripeti*.

Fai clic sulla categoria Controllo nella palette blocchi e cerca il blocco **per sempre**. Fai clic e trascina questo blocco nell'area codice, quindi rilascialo sotto a **quando si clicca su** e sopra al primo blocco **fai 10 passi**.



Il blocco per sempre è a forma di C e si ingrandisce automaticamente per racchiudere gli altri blocchi della tua sequenza. Fai clic sulla bandiera verde e vedrai subito che cosa fa il blocco **per sempre**: il programma, anziché essere eseguito una sola volta, si ripeterà, praticamente per sempre. In programmazione, questo è noto come *loop infinita*, ossia un ciclo che non finisce mai.

Per fermare il suono del miagolio costante fai clic sull'ottagono rosso accanto alla bandiera verde sopra l'area stage, in questo modo il programma si arresterà. Per cambiare il tipo di loop fai clic e trascina il primo blocco **fai 10 passi** e trascinalo insieme ai blocchi successivi fuori dal blocco **per sempre**, quindi rilasciali sotto al blocco **quando si clicca su**. Fai clic e trascina il blocco **per sempre** nella palette di blocchi per eliminarlo, quindi fai clic e trascina il blocco **ripeti 10 volte** sotto a **quando si clicca su** in modo che circonda gli altri blocchi.



Fai clic sulla bandiera verde per eseguire il nuovo programma. All'inizio sembrerà uguale alla versione originale, che ripeteva in continuazione la sequenza di istruzioni. Questa volta invece, anziché continuare per sempre, il loop cesserà dopo dieci volte. Viene definita *loop definita*: sarai tu a decidere quando finisce. I loop sono uno strumento essenziale e la maggior parte dei programmi, specialmente giochi e programmi di rilevamento, usa spesso loop sia infinite sia definite.



COSA SUCCEDDE ORA?

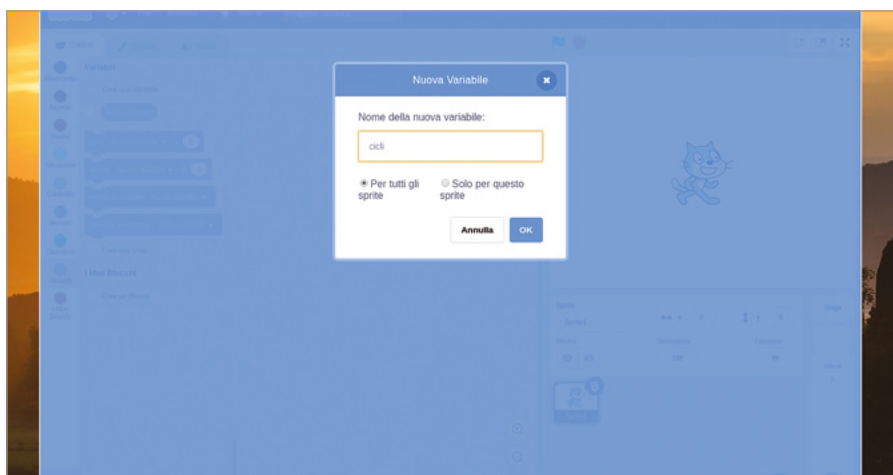
Cosa succede se cambi il numero nel blocco di loop perché sia superiore? Cosa succede se è inferiore? Cosa succede se metti il numero 0 nel blocco di loop?

Variabili e condizionali

Gli ultimi concetti da comprendere prima di iniziare a programmare con Scratch sono strettamente legati tra loro: *variabili* e *condizionali*. Una variabile è, come suggerisce il nome, un valore che può variare o cambiare nel tempo e sotto il controllo del programma. Le variabili hanno due proprietà principali: il nome e il valore che memorizzano. Questo valore non deve essere necessariamente un numero: può essere un numero, un testo, vero o falso oppure può essere vuota, ossia avere un *valore nullo*.

Le variabili sono strumenti essenziali. Pensa alle cose che devi tenere sotto controllo in un gioco: la salute di un personaggio, la velocità di un oggetto in movimento, il livello a cui si sta giocando e il punteggio. Tutti questi aspetti vengono monitorati come variabili.

Per prima cosa, fai clic sul menu File e salva il tuo programma facendo clic su "Salva sul tuo computer". Se hai già salvato il programma ti verrà chiesto se vuoi sovrascriverlo, sostituendo così la vecchia copia salvata con la nuova versione aggiornata. Successivamente fai clic su File, quindi su Nuovo per aprire un nuovo progetto vuoto (fai clic su OK quando viene chiesto se vuoi sostituire il contenuto del progetto corrente). Fai clic sulla categoria Variabili di colore arancione scuro nella palette dei blocchi, quindi sul pulsante "Crea una variabile". Digita "cicli" come nome della variabile (**Figura 4-5**), quindi fai clic su OK per visualizzare una serie di blocchi nella palette di blocchi.



▲ **Figura 4-5:** assegna un nome alla nuova variabile

Fai clic e trascina il blocco **porta cicli a 0** nell'area codice. Questo dice al programma di *inizializzare* la variabile con valore 0. Successivamente fai clic sulla categoria Aspetto della palette dei blocchi e trascina il blocco **dire Ciao! per 2 secondi** sotto il blocco **porta cicli a 0**.

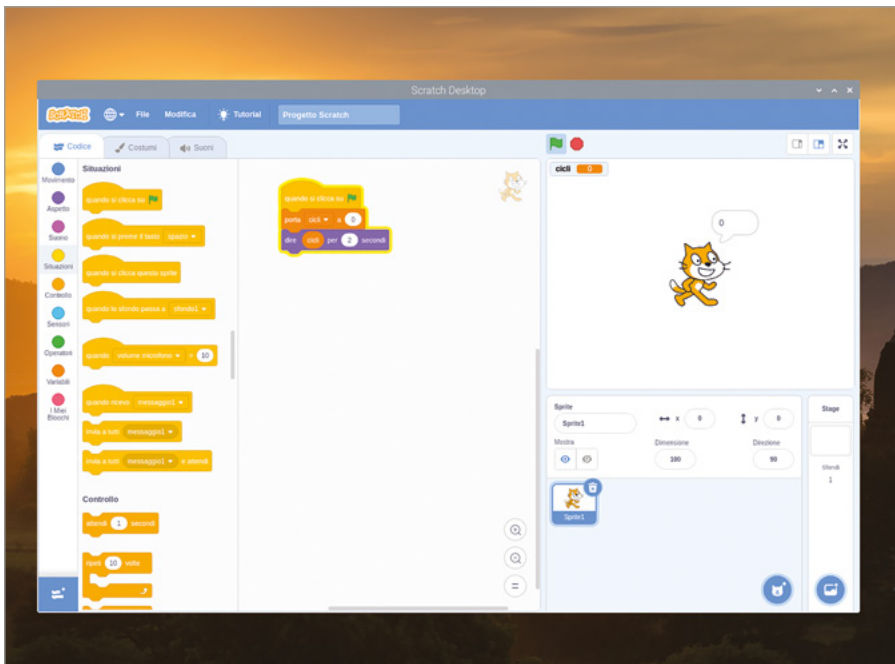


Come abbiamo visto prima, il blocco **dire Ciao!** fa dire allo sprite qualsiasi cosa vi sia scritto dentro. Anziché scrivere il messaggio nel blocco puoi utilizzare una variabile. Fai nuovamente clic sulla categoria Variabili nella palette dei blocchi, quindi fai clic e trascina il

blocco arrotondato di **cicli** (detto anche *bloccoreporter*, all'inizio della lista e con una casella di spunta accanto) sopra la parola 'Ciao!' nel blocco **dire Ciao! per 2 secondi**. In questo modo creerai un nuovo blocco combinato: **dire cicli per 2 secondi**.



Fai clic sulla categoria Situazioni della palette dei blocchi, fai clic e trascina il blocco **quando si clicca su**, quindi posizionalo all'inizio della sequenza di blocchi. Fai clic sulla bandiera verde sopra l'area stage e lo sprite gatto dirà "0" (**Figura 4-6**), ossia il valore che hai assegnato alla variabile "cicli".

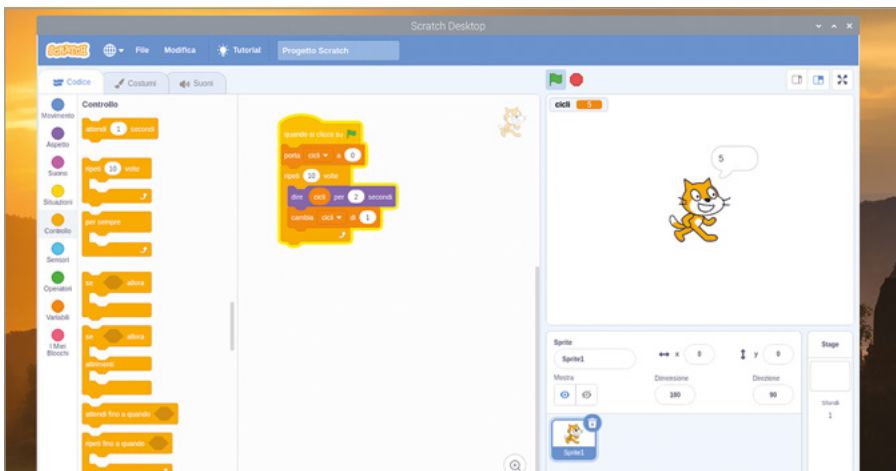


▲ **Figura 4-6:** questa volta il gatto dirà il valore della variabile

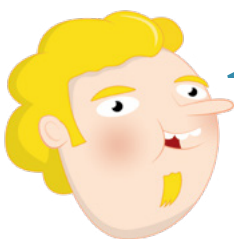
Le variabili non sono immutabili. Fai clic sulla categoria Variabili della palette dei blocchi, fai clic e trascina il blocco **cambia cicli di 1** in fondo alla sequenza. Successivamente fai clic sulla categoria Controllo, quindi fai clic e trascina un blocco **ripeti 10 volte** in modo che inizi direttamente sotto a **porta cicli a 0** e sia attorno ai blocchi rimanenti nella tua sequenza.



Fai di nuovo clic sulla bandiera verde. Questa volta vedrai il gatto contare da 0 a 9. Questo succede perché il programma sta cambiando o *modificando* la variabile stessa: ogni volta che il loop viene eseguita, il programma aggiunge uno al valore della variabile "cicli" (**Figura 4-7**).



▲ **Figura 4-7:** Grazie al loop il gatto conta



CONTARE DA ZERO

Anche se il loop che hai creato si ripete dieci volte, il gatto sprite conta solo fino a nove. Questo perché partiamo da zero per la nostra variabile. Includendo lo zero ci sono dieci numeri tra lo zero e il nove, così il programma si ferma prima che il gatto dica "10". Per modificarlo puoi impostare il valore iniziale della variabile su 1 anziché 0.

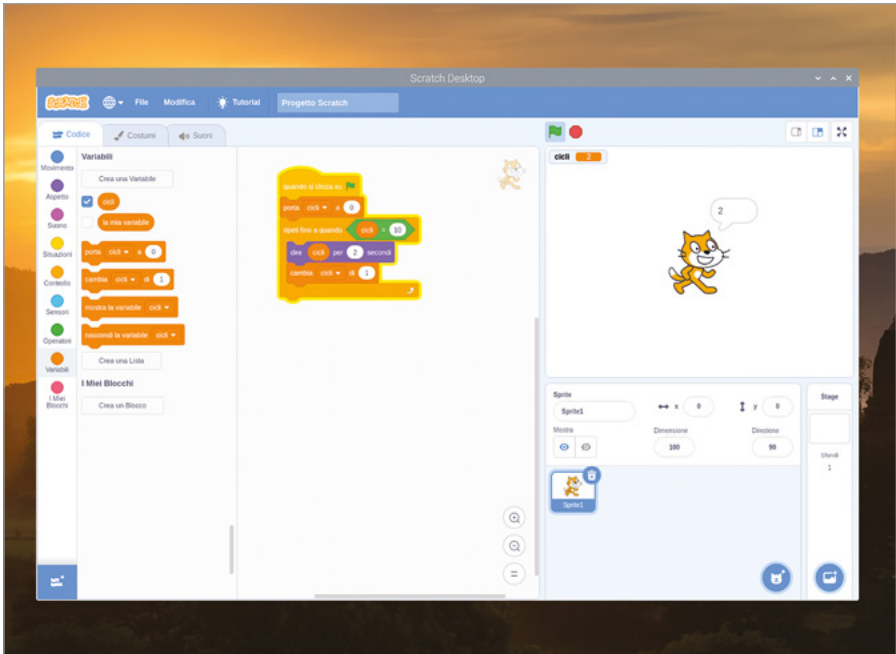
Puoi fare tantissime cose con le variabili. Fai clic e trascina il blocco **dire cicli per 2 secondi** per staccarlo da **ripeti 10 volte** e posizionalo sotto al blocco **ripeti 10 volte**. Fai clic e trascina il blocco **ripeti 10 volte** nella palette dei blocchi per eliminarlo, quindi sostituiscilo con **ripeti fino a quando**, assicurandoti che il blocco sia collegato sotto a **porta cicli a 0** e circonda gli altri due blocchi nella sequenza. Fai clic sulla categoria Operatori, di colore verde, nella palette dei blocchi, quindi fai clic e trascina il blocco a forma di diamante **=** e rilascialo sullo spazio a forma di diamante presente in **ripeti fino a quando**.



Questo blocco Operatori ti consente di confrontare due valori, comprese le variabili. Fai clic sulla categoria Variabili, trascina il blocco reporter **cicli** nello spazio vuoto del blocco Operatori **=**, poi fai clic sullo spazio con "50" e digita il numero "10".



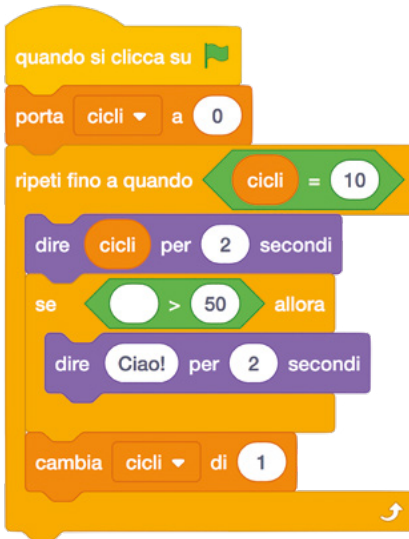
Fai clic sulla bandiera verde sopra l'area stage e vedrai che il programma funzionerà come prima: lo sprite gatto conta da 0 a 9 (**Figura 4-8**) e il programma si ferma. Questo perché il blocco **ripeti fino a quando** funziona esattamente come il blocco **ripeti 10 volte**, ma invece di contare il numero di loop confronta il valore della variabile "cicli" con quello digitato a destra del blocco. Quando la variabile "cicli" raggiunge il valore 10 il programma si ferma.



▲ **Figura 4-8:** utilizzo di un blocco "ripeti fino a quando" con un operatore comparativo

Questo si chiama *operatore comparativo* perché confronta due valori. Fai clic sulla categoria Operatori della palette dei blocchi e cerca gli altri due blocchi a forma di diamante sopra e sotto quelli con il simbolo "=". Sono operatori comparativi anche "<", che confronta due valori e si attiva quando il valore di sinistra è inferiore a quello di destra, e ">", che si attiva quando il valore di sinistra è superiore a quello di destra.

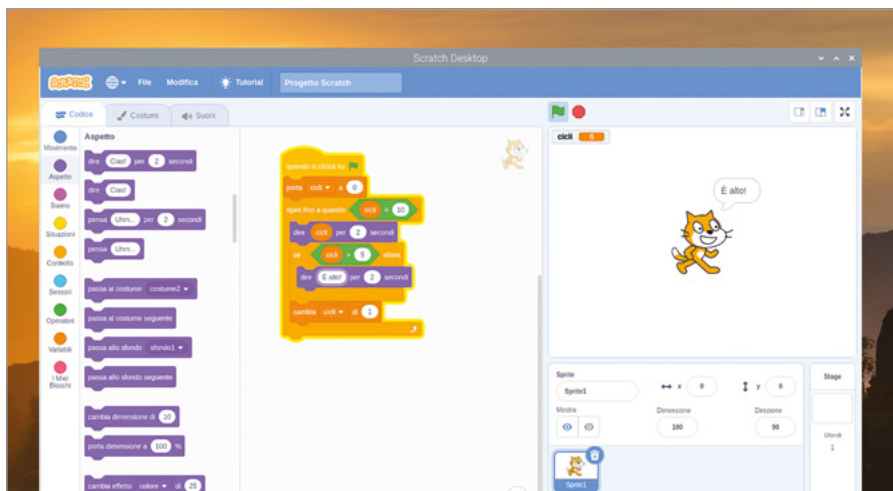
Fai clic sulla categoria Controllo della palette dei blocchi, trova il blocco **se allora**, quindi fai clic e trascinalo nell'area del codice prima di collocarlo direttamente sotto al blocco **dire cicli per 2 secondi**. Circonderà automaticamente il blocco **cambia cicli di 1**, quindi fai clic e trascinalo per spostarlo in modo che si colleghi alla parte inferiore del blocco **se allora**. Fai clic sulla categoria Aspetto della palette dei blocchi e trascina il blocco **dire Ciao! per 2 secondi**, quindi rilascialo nel blocco **se allora**. Fai clic sulla categoria Operatori della palette dei blocchi e trascina il blocco **>** nello spazio a forma di diamante nel blocco **se allora**.



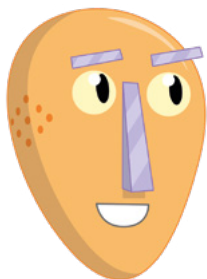
Il blocco **se allora** è un blocco condizionale, il che significa che i blocchi al suo interno funzioneranno solo se viene soddisfatta una certa condizione. Fai clic sulla categoria Variabili nella palette, trascina il blocco reporter **cicli** nello spazio vuoto del blocco **>**, poi fai clic sullo spazio con "50" e digita il numero "5". Infine, fai clic sulla parola "Ciao!" nel blocco **dire Ciao! per 2 secondi** e digita "È alto!".



Fai clic sulla bandiera verde. In un primo momento il programma funzionerà come prima, con lo sprite gatto che conta da 0 in poi. Quando raggiungerà 6, il primo numero maggiore di 5, il blocco **se allora** verrà attivato e il gatto commenterà quanto sono alti i numeri (**Figura 4-9**). Complimenti, ora puoi lavorare con variabili e condizionali!



▲ **Figura 4-9:** il gatto fa un commento quando il numero raggiunge il 6



SFIDA: ALTO E BASSO

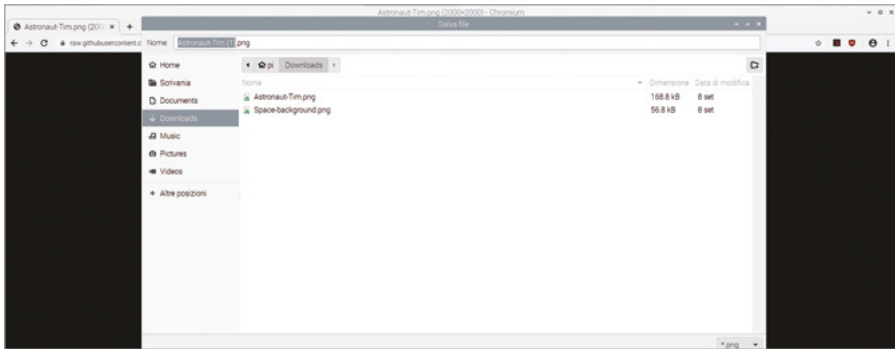
Come si può cambiare il programma in modo che lo sprite gatto commenti quanto sono bassi i numeri sotto il 5? Puoi cambiarlo in modo che il gatto commenti sia i numeri alti che quelli bassi? Sperimenta con il blocco **se allora altrimenti** per rendere tutto più semplice!

Progetto 1: misuratore della velocità di reazione con astronauta

Ora che hai appreso come funziona Scratch, è il momento di creare qualcosa di più interattivo: un misuratore della velocità di reazione per onorare l'astronauta britannico dell'ESA Tim Peake e il suo tempo a bordo della Stazione Spaziale Internazionale.

Salva il programma appena creato, se desideri tenerlo, quindi apri un nuovo progetto facendo clic su File e Nuovo. Prima di iniziare, assegnagli un nome facendo clic su File e "Salva sul tuo computer" e chiamalo "Misuratore della velocità di reazione con astronauta".

Questo progetto si basa su due immagini non incluse nelle risorse integrate di Scratch: lo sfondo dello stage e lo sprite. Per scaricarle, fai clic sull'icona a forma di lampone per caricare il menu Raspberry Pi OS, sposta il puntatore del mouse su Internet e fai clic sul browser Web Chromium. Una volta aperto il browser, digita **rpf.io/astronaut-backdrop** nella barra degli indirizzi, e premi il tasto **INVIO**. Fai clic con il tasto destro del mouse sull'immagine dello spazio, fai clic su "Salva immagine con nome...", quindi fai clic sul pulsante Salva (**Figura 4-10**). Fai nuovamente clic sulla barra degli indirizzi e digita **rpf.io/astronaut-sprite** e premi il tasto **INVIO**.





▲ **Figura 4-10: salvataggio dell'immagine di sfondo**

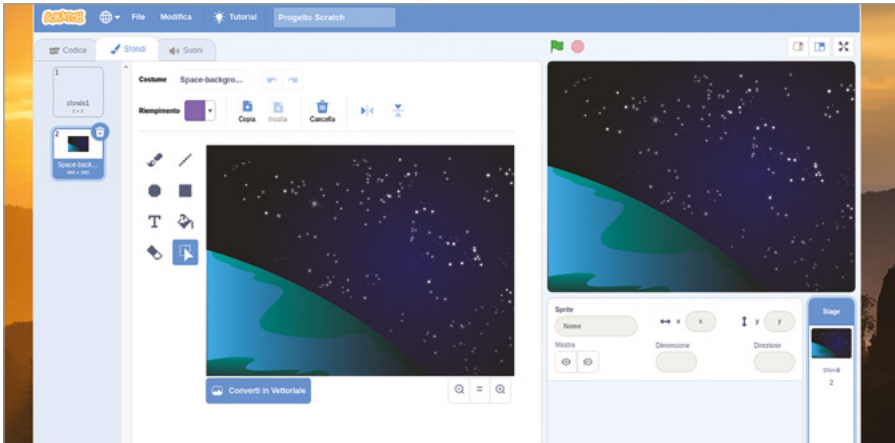
Fai clic con il tasto destro del mouse sull'immagine di Tim Peake, fai clic su "Salva immagine con nome...", quindi seleziona la cartella Download e fai clic sul pulsante Salva. Una volta salvate le immagini, potrai chiudere Chromium o lasciarlo aperto e utilizzare la barra delle applicazioni per tornare a Scratch 3.



INTERFACCIA UTENTE

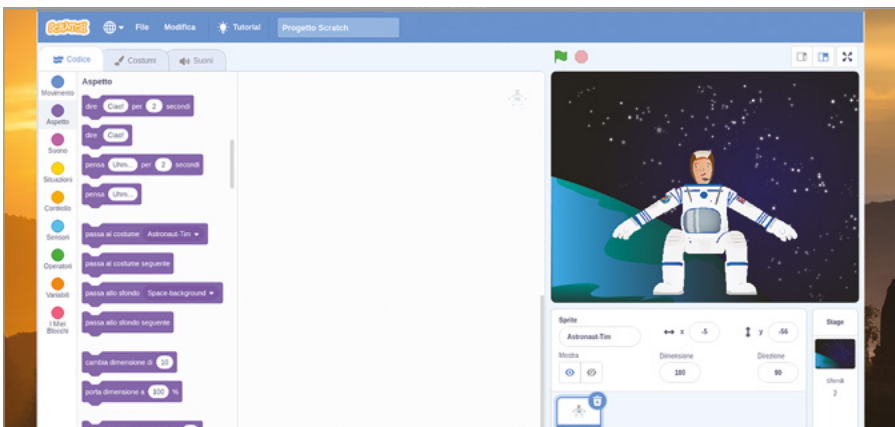
Se hai seguito questo capitolo fin dall'inizio, dovresti avere familiarità con l'interfaccia utente di Scratch 3. Le seguenti istruzioni presuppongono che tu conosca un po' la struttura del programma, se dimentichi dove trovare qualcosa, guarda di nuovo l'immagine dell'interfaccia utente all'inizio del capitolo.

Fai clic con il tasto destro del mouse sullo sprite gatto nell'elenco e fai clic su "cancella". Passa il puntatore del mouse sull'icona Scegli uno sfondo , quindi fai clic sull'icona di caricamento dello sfondo  dall'elenco visualizzato. Cerca il file **Space-background.png** nella cartella Downloads, fai clic per selezionarlo, quindi fai clic su OK. Lo stage avrà quindi come sfondo l'immagine dello spazio, e l'area codice sarà sostituita dall'area dello sfondo (**Figura 4-11**). Qui puoi disegnare sullo sfondo, ma per ora fai clic sulla scheda Codice nella parte superiore della finestra di Scratch 3



▲ **Figura 4-11:** sfondo nell'area stage

Carica il tuo nuovo sprite passando il puntatore del mouse sull'icona **Scegli uno Sprite**, quindi facendo clic sull'icona **Importa Sprite** all'inizio dell'elenco visualizzato. Cerca il file **Astronaut-Tim.png** nella cartella Downloads, fai clic per selezionare, quindi fai clic su OK. Lo sprite verrà aperto immediatamente nello stage, ma potrebbe non essere al centro: fai clic e trascinalo con il mouse e rilascialo in modo che sia vicino al centro inferiore (**Figura 4-12**).



▲ **Figura 4-12:** trascinamento dello sprite astronauta nella parte inferiore centrale dello stage

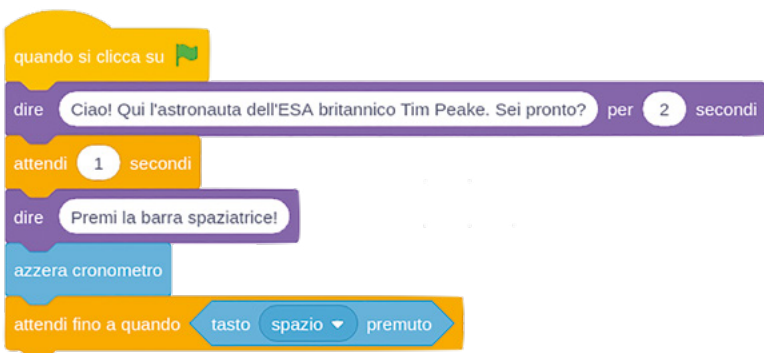
Una volta aperti sfondo e sprite puoi iniziare a creare il tuo programma. Inizia creando una nuova variabile chiamata "tempo", accertandoti di aver selezionato "Per tutti gli sprite" prima di fare clic su OK. Fai clic sul tuo sprite (sullo stage o nel pannello sprite) per selezionarlo, quindi aggiungi un blocco **quando si clicca su** dalla categoria Situazioni nell'area del codice. Quindi aggiungi un blocco **dire Ciao! per 2 secondi** dalla categoria Aspetto e fai clic su di esso per modificare in "Ciao! Qui l'astronauta dell'ESA britannico Tim Peake. Sei pronto?"



Aggiungi un blocco **attendi 1 secondi** dalla categoria Controllo, quindi un blocco **dire Ciao!**. Cambia questo blocco perché dica "Premi la barra spaziatrice!", quindi aggiungi un blocco della categoria Sensori **azzerà cronometro**. Questo blocco controlla una variabile speciale di Scratch che ti consente di cronometrare alcune azioni e servirà per capire quanto velocemente reagisci nel gioco.

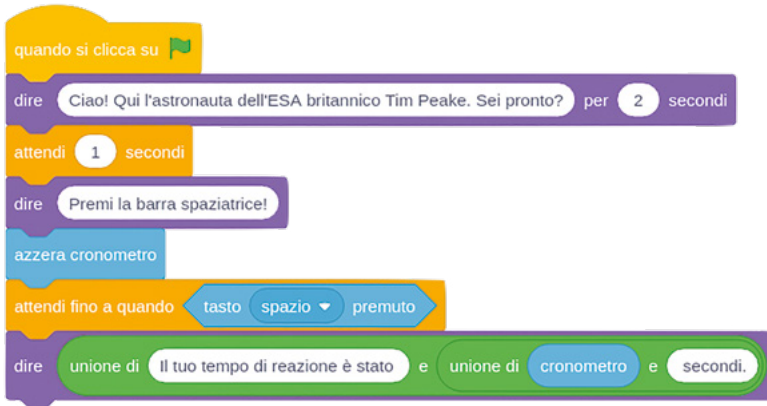


Aggiungi un blocco di Controllo **attendi fino a quando**, quindi trascina il blocco della categoria Sensori **tasto spazio premuto** nello spazio bianco. Il programma rimarrà in pausa finché non premi il tasto **SPAZIO** sulla tastiera, ma il cronometro continuerà ad andare avanti e cronometrerà il tempo intercorso tra il messaggio che richiede di premere il tasto spazio e il momento in cui l'hai premuto.

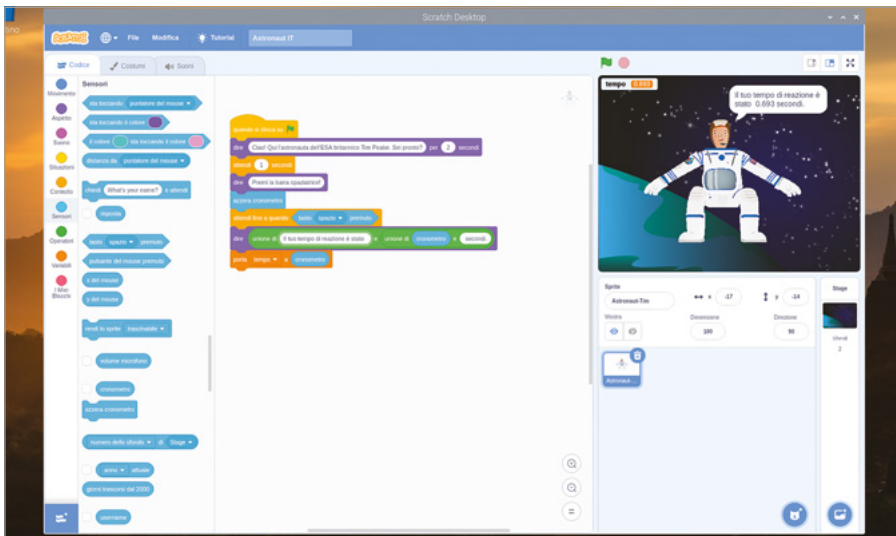


Ora hai bisogno che Tim ti dica quanto tempo hai impiegato a premere il tasto **SPAZIO** in un modo facile da leggere. Per farlo dovrai utilizzare un blocco Operatori **unione**. Questo prenderà due valori, comprese le variabili, e li unirà uno dopo l'altro, un'azione nota come *concatenazione*.

Inizia con un blocco **dire Ciao!** quindi trascina un blocco Operatori **unione** sulla parola "Ciao!". Fai clic sulla "apple" e digita "Il tuo tempo di reazione è stato", accertandoti di aggiungere uno spazio alla fine, quindi trascina un altro blocco di unione sulla "banana" nella seconda casella. Trascina dalla categoria Sensori un blocco di reporting **cronometro** nella casella centrale e digita "secondi" nell'ultima casella, accertandoti di includere uno spazio all'inizio.



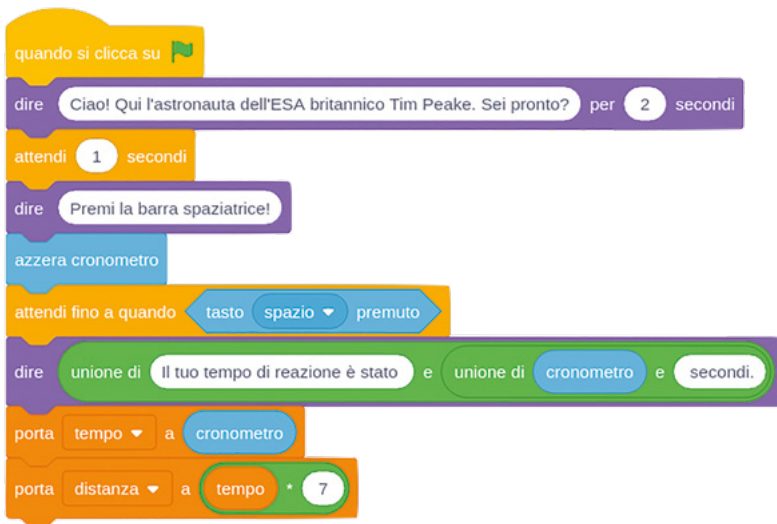
Infine trascina un blocco Variabili **porta variabile a 0** alla fine della sequenza. Fai clic sulla freccia verso il basso accanto alla variabile e fai clic su "tempo" dall'elenco, quindi sostituisci lo "0" con un blocco di reporting **cronometro** dalla categoria Sensori. Puoi testare il tuo gioco facendo clic sulla bandiera verde sopra allo stage. Preparati e, non appena vedrai il messaggio "Premi la barra spaziatrice!", premi il tasto **SPAZIO** il più velocemente possibile (**Figura 4-13**) e cerca di battere il nostro punteggio più alto!



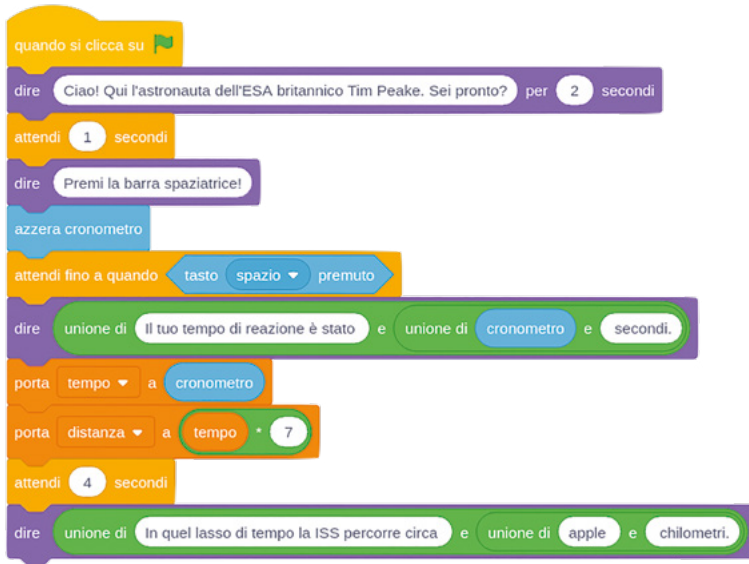
▲ **Figura 4-13:** è arrivato il momento di giocare!

Puoi estendere ulteriormente questo progetto facendo calcolare la distanza percorsa all'incirca dalla Stazione Spaziale Internazionale nel tempo che ti è occorso per premere il tasto **SPAZIO** in base alla velocità di sette chilometri al secondo pubblicata dalla stazione. Per prima cosa, crea una nuova variabile chiamata "distanza". Osserva come i blocchi nella categoria Variabili cambiano automaticamente per mostrare la nuova variabile, ma i blocchi di variabili **tempo** già presenti nel programma rimangono invariati.

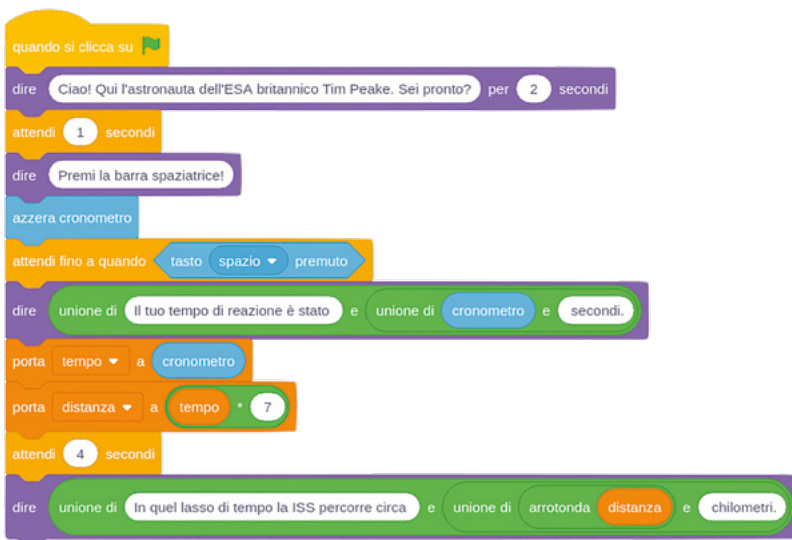
Aggiungi un blocco **porta distanza a 0**, quindi trascina un blocco Operatori ***** (ossia moltiplicazione) sullo "0". Trascina un blocco di reporting **tempo** sul primo spazio bianco, quindi digita il numero "7" nel secondo spazio. Una volta terminato, il blocco combinato leggerà **porta distanza a tempo * 7**. Il tempo che ti è occorso per premere il tasto **SPAZIO** verrà quindi moltiplicato per sette per ottenere la distanza percorsa in chilometri dalla stazione spaziale.



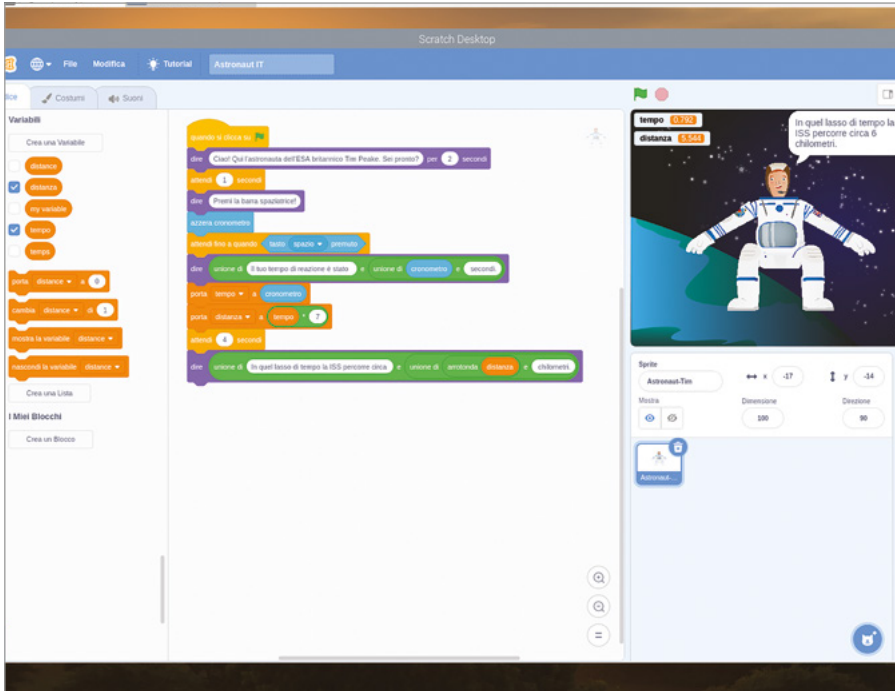
Aggiungi un blocco **attendi 1 secondi** e portalo a "4". Infine, trascina un altro blocco **dire Ciao!** alla fine della sequenza e aggiungi due blocchi **unione** proprio come hai fatto prima. Nel primo spazio (su "apple") digita "In quel lasso di tempo la ISS percorre circa", ricordandoti di includere lo spazio alla fine, mentre nello spazio "banana" digita 'chilometri', ricordandoti di inserire uno spazio all'inizio.



Infine, trascina un blocco Operatori **arrotonda** nello spazio vuoto centrale, quindi trascina un blocco di reporting della **distanza** nel nuovo spazio vuoto creato. Il blocco **arrotonda** fa quello che dice: arrotonda i numeri per eccesso o per difetto al numero intero più vicino, così si ottiene un numero di facile lettura dei chilometri, anziché un numero preciso ma difficile da leggere.



Fai clic sulla bandiera verde per eseguire il programma e vedere quanto si sposta la stazione spaziale nel tempo che ti è occorso a premere il tasto **SPAZIO**. Ricordati di salvare il programma quando hai finito in modo da poterlo caricare facilmente in futuro senza dover ricominciare da capo.



▲ Figura 4-14: Tim ti dice quanta distanza ha percorso la stazione spaziale



SFIDA: CHI È VELOCE?

Oltre all'astronauta, quali altre professioni richiedono riflessi rapidi? Puoi disegnare sprite e sfondi per mostrare una di queste professioni?


Progetto 2: nuoto sincronizzato

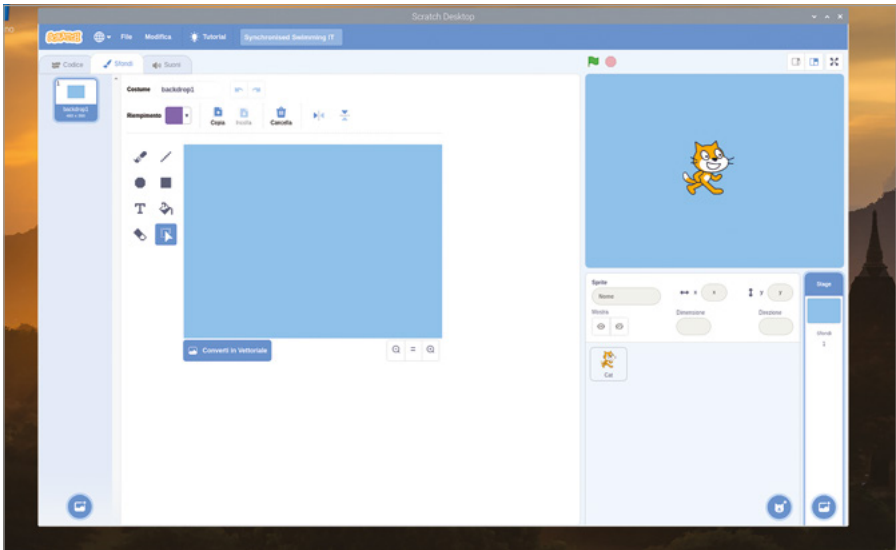
La maggior parte dei giochi usa più di un tasto, con questo progetto vedremo come usare il controllo con i tasti ← e → della tastiera.

PROGETTO ONLINE


Questo progetto è disponibile anche online all'indirizzo rpf.io/synchro-swimming



Crea un nuovo progetto e salvalo come "Nuoto sincronizzato". Fai clic su Stage nella relativa sezione, quindi fai clic sulla scheda Sfondi in alto a sinistra. Fai clic sul pulsante per convertire in bitmap sotto lo sfondo. Scegli un colore blu simile all'acqua dalla tavolozza di riempimento, fai clic sull'icona di riempimento , quindi fai clic sullo sfondo a scacchiera per colorare di blu (Figura 4-15).



▲ **Figura 4-15:** riempire lo sfondo con un colore blu

Fai clic con il tasto destro del mouse sullo sprite gatto nell'elenco e fai clic su "cancella". Fai clic sull'icona "Scegli uno Sprite"  per aprire una lista di sprite predefiniti. Fai clic sulla categoria Animali, quindi su "Cat flying" (Figura 4-16). Questo sprite è molto adatto ai progetti che prevedono il nuoto.



▲ **Figura 4-16:** Scegli uno sprite dalla libreria

Fai clic sul nuovo sprite, quindi trascina due blocchi Situazioni **quando si preme il tasto spazio** nell'area codice. Fai clic sulla piccola freccia verso il basso accanto alla parola "spazio" nel primo blocco e scegli "freccia sinistra" dall'elenco delle opzioni possibili. Trascina un blocco di movimento **ruota 15 gradi** sotto


il blocco **quando si preme il tasto freccia sinistra**, quindi fai lo stesso con il secondo blocco Situazioni, ma scegli la "freccia destra" dall'elenco e usa un blocco movimento **ruota ↻15 gradi**.



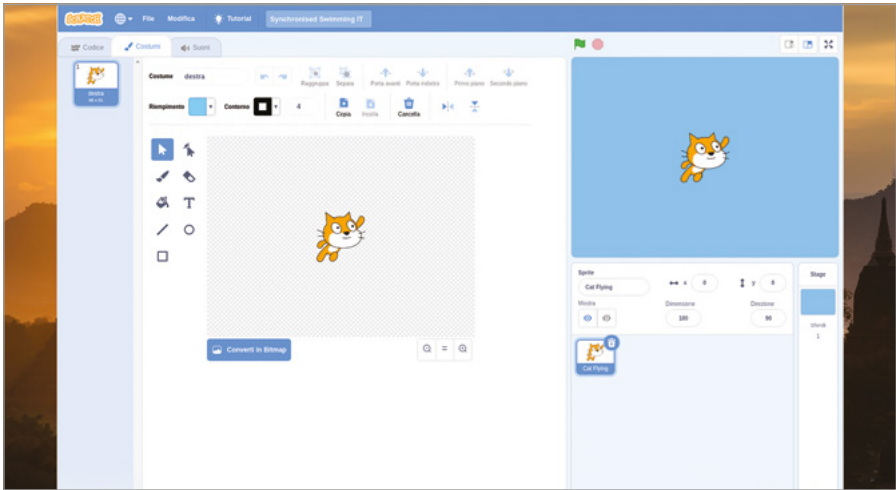
Premi il tasto ← o → per testare il programma. Vedrai lo sprite gatto girare in corrispondenza della direzione scelta sulla tastiera. Questa volta non è stato necessario cliccare sulla bandiera verde, questo perché i blocchi di trigger di Situazioni sono sempre attivi, anche quando il programma non è effettivamente "in esecuzione".

Ripeti due volte gli stessi passaggi, ma questa volta scegliendo "freccia su" e "freccia giù" per i blocchi di trigger di Situazioni, quindi **fai 10 passi** e **fai -10 passi** per i blocchi Movimento. Prem i tasti freccia ora e vedrai che il gatto può girarsi e anche nuotare in avanti e indietro!





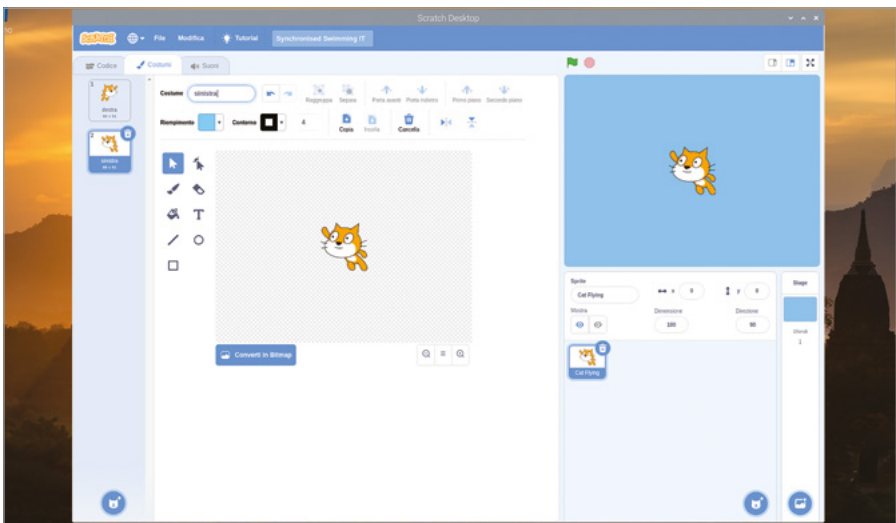
Per rendere più realistico il movimento dello sprite gatto, è possibile modificarne il suo aspetto, ossia usare *costumi*. Fai clic sullo sprite gatto, quindi sulla scheda Costumi sopra la palette dei blocchi. Fai clic sul costume "cat flying-a" e sull'icona con la X sul cestino 

nell'angolo in alto a destra per cancellarlo. Successivamente fai clic sul costume 'cat flying-b' e usa la casella del nome in alto per rinominarlo "destra" (**Figura 4-17**).



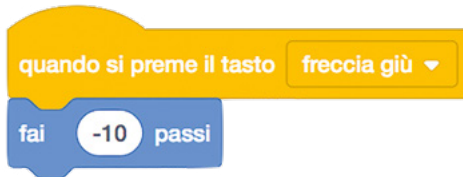
▲ **Figura 4-17: rinominare il costume come "destra"**


Fai clic con il tasto destro del mouse sul costume appena rinominato "destra" e fai clic su "copia" per crearne una copia. Fai clic su questa copia per selezionarla, quindi sull'icona Seleziona  e fai clic sul pulsante per riflettere in orizzontale , quindi rinominala "sinistra" (**Figura 4-18**). Avrai due "costumi" per il tuo sprite, che sono immagini speculari esatte: uno chiamato "destra" con il gatto rivolto a destra, e uno chiamato "sinistra" con il gatto rivolto a sinistra

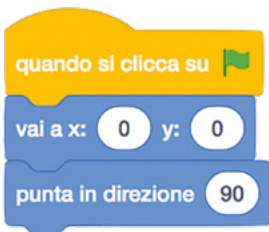


▲ **Figura 4-18: duplica il costume, capovolgilo e chiamalo "sinistra"**

Fai clic sulla scheda Codice sopra l'area del costume, quindi trascina due blocchi Aspetto **passa al costume sinistra** sotto i blocchi Situazioni freccia sinistra e destra, cambiando quello sotto il blocco freccia destra perché legga **passa al costume destra**. Prova di nuovo i tasti freccia, il gatto ora gira nella direzione in cui sta nuotando.



Per il nuoto sincronizzato in stile olimpico, però, abbiamo bisogno di più nuotatori, quindi di un modo per ripristinare la posizione dello sprite gatto. Aggiungi un blocco Situazioni **quando si clicca su** , quindi sotto aggiungi un blocco Movimento **vai a x: 0 y: 0**. Blocco Movimento cambiando i valori, se necessario, e un blocco Movimento **punta in direzione 90**. Ora, quando fai clic sulla bandiera verde il gatto verrà spostato al centro dello stage e punterà a destra.



Per creare più nuotatori, aggiungi un blocco **ripeti 6 volte** modificando il valore predefinito "10" e aggiungi un blocco **Controllo crea clone di me stesso** al suo interno. Per fare in modo che i nuotatori non nuotino tutti nella stessa direzione, aggiungi un blocco **ruota 60 gradi** sopra il blocco **crea clone**, ma all'interno del blocco **ripeti 6 volte**. Fai clic sulla bandiera verde e prova i tasti freccia per vedere i tuoi nuotatori muoversi!



```
quando si preme il tasto freccia sinistra
  passa al costume sinistra
  ruota di 15 gradi
```



```
quando si preme il tasto freccia destra
  passa al costume destra
  ruota di 15 gradi
```




```
quando si preme il tasto freccia su
  fai 10 passi
```

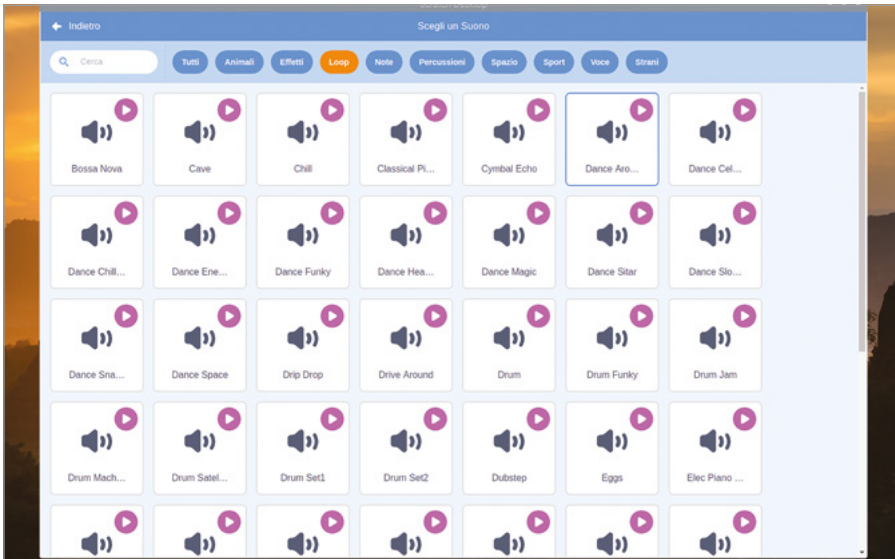


```
quando si preme il tasto freccia giù
  fai -10 passi
```




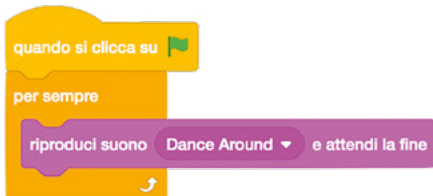
```
quando si clicca su
  vai a x: 0 y: 0
  punta in direzione 90
  ripeti 6 volte
    ruota di 60 gradi
    crea clone di me stesso
```

Per un'atmosfera più olimpica, aggiungi un po' di musica. Fai clic sulla scheda Suoni, quindi sull'icona  per scegliere un suono. Fai clic sulla categoria Loop, quindi sfoglia l'elenco (**Figura 4-19**) fino a quando non trovi la musica che ti piace: abbiamo scelto "Dance Around". Fai clic sulla musica che vuoi, quindi fai clic sulla scheda Codice per aprire nuovamente l'area del codice.

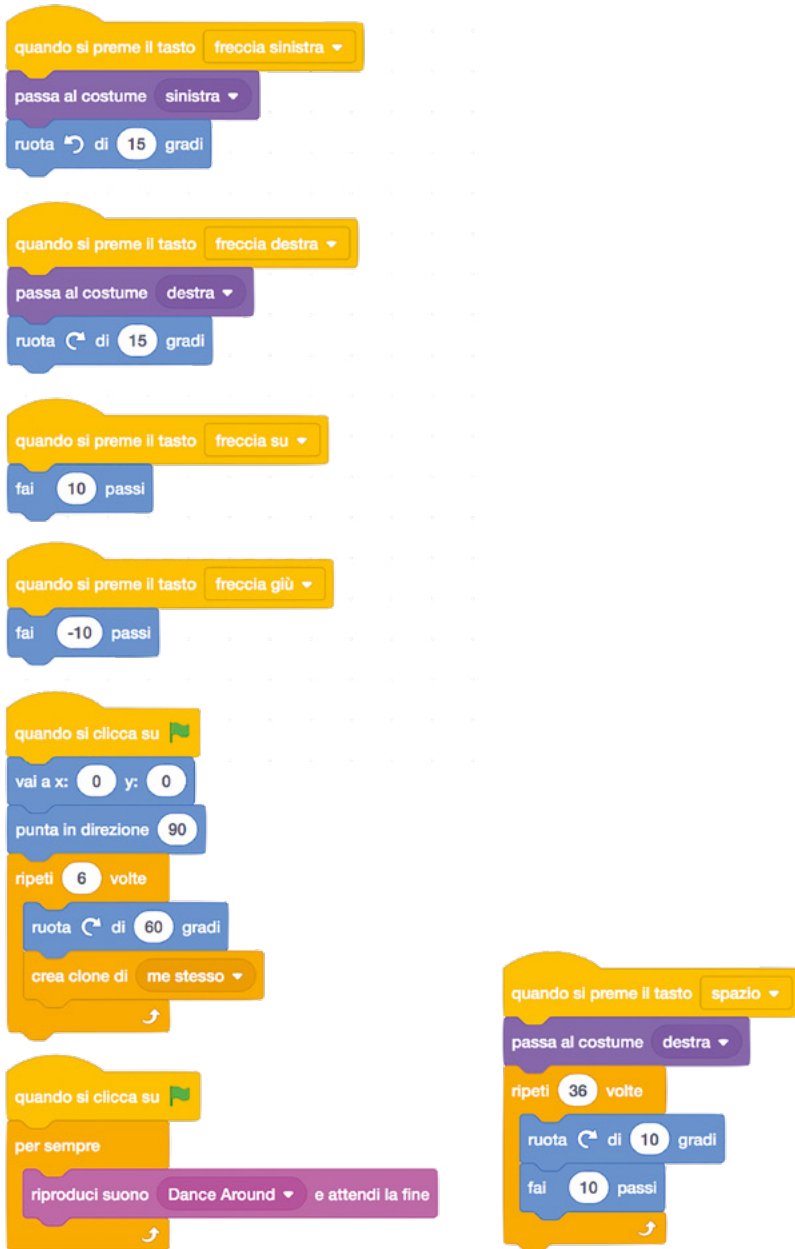


▲ **Figura 4-19:** seleziona un loop musicale dalla libreria di suoni

Aggiungi un altro blocco Situazioni **quando si clicca su**  nell'area codice, quindi aggiungi un blocco di Controllo **per sempre**. Nel blocco Controllo, aggiungi un blocco **riproduci suono Dance Around e attendi la fine** ricordandoti di cercare il nome del brano musicale scelto, quindi fai clic sulla bandiera verde per testare il nuovo programma. Se vuoi fermare la musica, fai clic sull'ottagono rosso per fermare il programma e la musica.

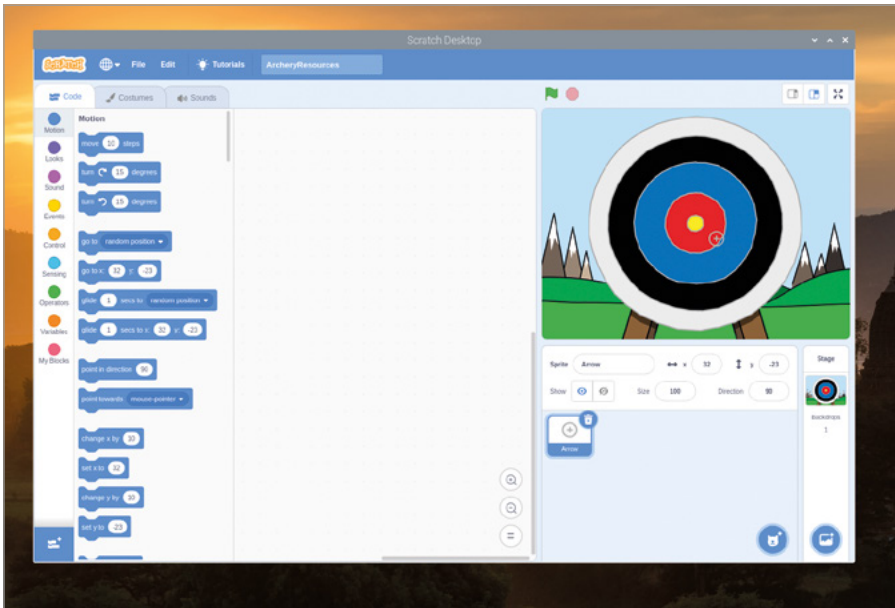


Infine, è possibile simulare una routine di ballo completa aggiungendo un nuovo trigger di eventi al programma. Aggiungi un blocco Situazioni **quando si preme il tasto spazio**, poi un blocco **passa al costume destra**. Sotto questo blocco aggiungi un blocco **ripeti 36 volte** (ricordandoti di cambiare il valore predefinito) e all'interno di questo un blocco **ruota 10 gradi** e **fai 10 passi**.




Fai clic sulla bandiera verde per avviare il programma, quindi premi il tasto **SPAZIO** per provare la nuova routine (**Figura 4-20**, sul retro). Non dimenticare di salvare il programma alla fine.

seguito da "Carica dal computer". Fai clic su **ArcheryResources.sb3**, quindi sul pulsante Apri. Ti verrà chiesto se vuoi sostituire il contenuto del tuo progetto: se non hai salvato le modifiche, fai clic su Annulla e salvale, altrimenti fai clic su OK.



▲ **Figura 4-21:** risorse del progetto caricate per il gioco del tiro con l'arco

Il progetto appena caricato contiene uno sfondo e uno sprite (**Figura 4-21**), ma dovrai aggiungere tu il codice del gioco. Inizia aggiungendo un blocco **quando si clicca su** , poi un blocco **invia a tutti messaggio1**. Fai clic sulla freccia in basso alla fine del blocco, poi "Nuovo messaggio" e digita "nuova freccia" prima di fare clic sul pulsante OK. Il blocco sarà quindi **invia a tutti nuova freccia**.



Un messaggio inviato a tutti (o broadcast) è un messaggio di una parte del programma che può essere ricevuto da qualsiasi altra parte del programma. Per fare in modo che faccia qualcosa, aggiungi un blocco **quando ricevo messaggio1** e cambiarlo di nuovo per leggere **quando ricevo nuova freccia**. Questa volta dovrai solo fare clic sulla freccia verso il basso e scegliere "nuova freccia" dall'elenco, non è necessario creare di nuovo il messaggio.

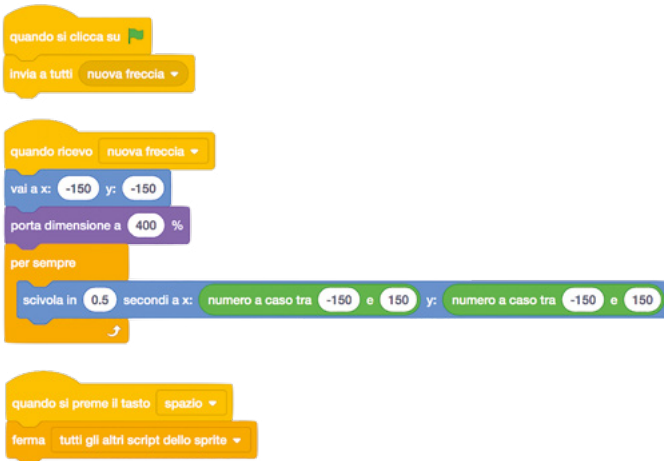
Sotto al blocco **quando ricevo nuova freccia**, aggiungi un blocco **vai a x: -150 y: -150** e **porta dimensione a 400 %**. Ricorda che questi non sono i valori predefiniti per quei blocchi, quindi dovrai cambiarli una volta trascinati nell'area codice. Fai clic sulla bandiera verde per vedere quanto fatto finora: lo sprite freccia, che il giocatore usa per mirare al bersaglio, si sposterà nell'angolo inferiore sinistro dello stage e avrà dimensioni quadruplicate.



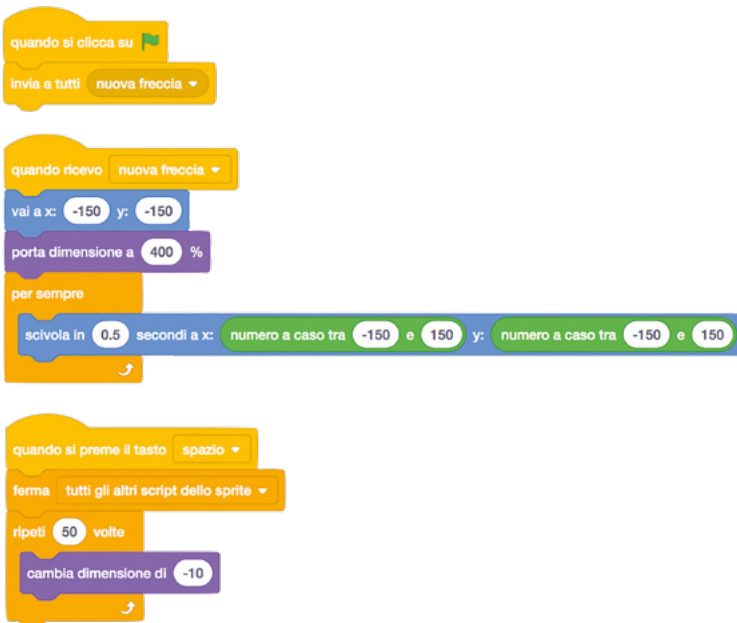
Per fare in modo che il giocatore si senta sfidato, aggiungi un movimento che simula l'oscillazione mentre viene tirato l'arco e l'arciere prende la mira. Trascina il blocco **per sempre**, seguito da **scivola in 1 secondi a x: -150 y: -150**. Modifica la prima casella bianca a "0.5" anziché "1" poi colloca un blocco Operatori **numero a caso tra -150 e 150** in ognuna delle altre due caselle bianche. Questo significa che la freccia si sposterà sullo stage in una direzione casuale per una distanza casuale, rendendo così più difficile colpire il bersaglio.




Fai clic di nuovo sulla bandiera verde e vedrai cosa fa quel blocco: il tuo sprite freccia ora si sta spostando nello stage, coprendo diverse parti del bersaglio. Al momento, però, non c'è modo di scagliare la freccia verso il bersaglio. Trascina un blocco **quando si preme il tasto spazio** nell'area codice, seguito da un blocco di controllo **ferma tutto**. Fai clic sulla freccia verso il basso alla fine del blocco e seleziona **tutti gli altri script dello sprite**.

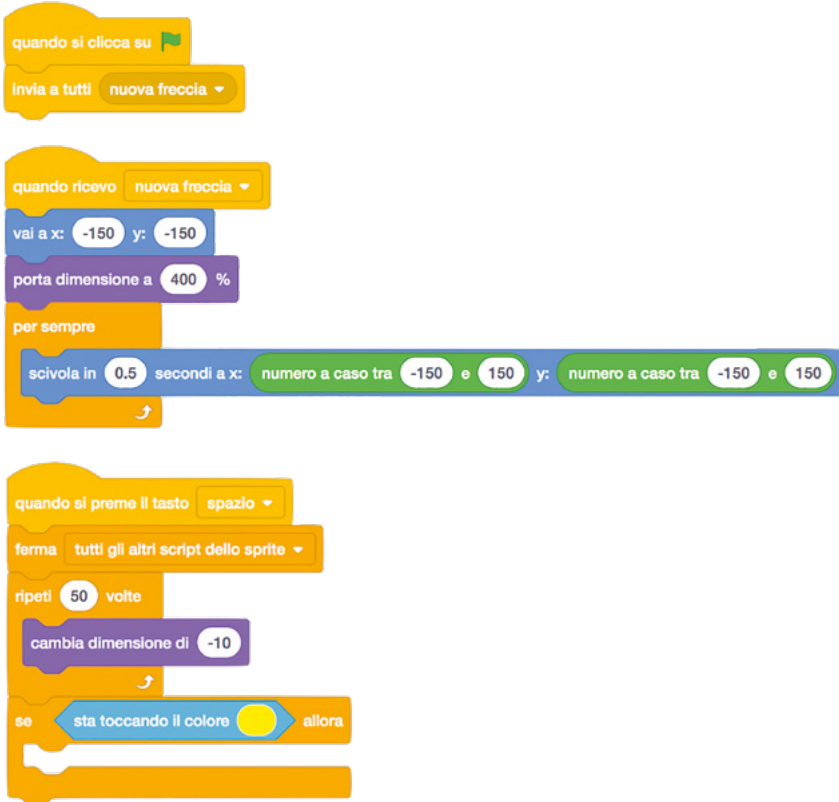


Se hai fermato il tuo programma per aggiungere i nuovi blocchi, fai clic sulla bandiera verde per farlo ripartire e premi il tasto **SPAZIO**: vedrai lo sprite freccia smettere di muoversi. È un buon inizio, ma devi fare in modo che la freccia sembri volare verso il bersaglio. Aggiungi un blocco **ripeti 50 volte** seguito da **cambia dimensione di -10**, quindi fai clic sulla bandiera verde per testare di nuovo il tuo gioco. Questa volta la freccia sembra volare via verso il bersaglio.

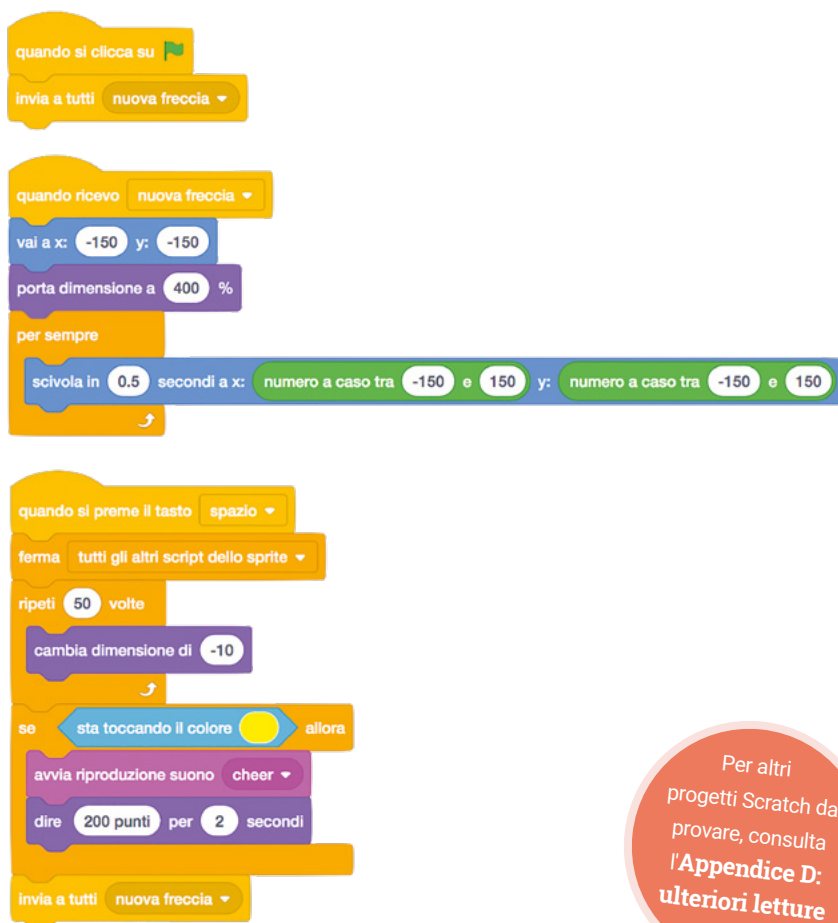


Per rendere il gioco divertente, è necessario aggiungere un modo per tenere il punteggio. Nel stesso insieme di blocchi, aggiungi un blocco **se allora**, assicurandoti che sia al di sotto

di **ripeti 50 volte** e non all'interno, con un blocco Sensori **sta toccando il colore**. Blocco Sensori nello spazio a forma di diamante. Per scegliere il colore corretto, fai clic sulla casella colorata alla fine del blocco Sensori, poi sull'icona del contagocce , quindi fai clic sul centro giallo del bersaglio sullo stage.



Per far in modo che il giocatore sappia di aver centrato il bersaglio, aggiungi un blocco **riproduzione suono cheer** e un blocco **dire 200 punti per 2 secondi** all'interno del blocco **se allora**. Infine, aggiungi un blocco **invia a tutti nuova freccia** in fondo alla serie di blocchi, sotto e fuori dal blocco **se allora**, per dare al giocatore un'altra freccia ogni volta che ne scaglia una. Fai clic sulla bandiera verde per avviare la partita e cerca di colpire il centro giallo: quando lo farai, sarai ricompensato con un applauso dalla folla e un punteggio di 200 punti!



Per altri
progetti Scratch da
provare, consulta
l'Appendice D:
ulteriori letture

Il gioco funziona, ma è un po' impegnativo. Usa quello che hai imparato in questo capitolo per aggiungere punteggi per i vari cerchi del bersaglio oltre al centro: 100 punti per il rosso, 50 punti per il blu e così via.



SFIDA: COME PUOI MIGLIORARE IL GIOCO?

Come puoi renderlo più semplice? Come puoi renderlo più difficile? Puoi usare le variabili per far aumentare il punteggio del giocatore quando lancia più frecce? Puoi aggiungere un timer per il conto alla rovescia per mettere più pressione al giocatore?

Capitolo 5

Come programmare con Python

Ora che hai imparato a usare Scratch, affronteremo la programmazione basata su testo con Python

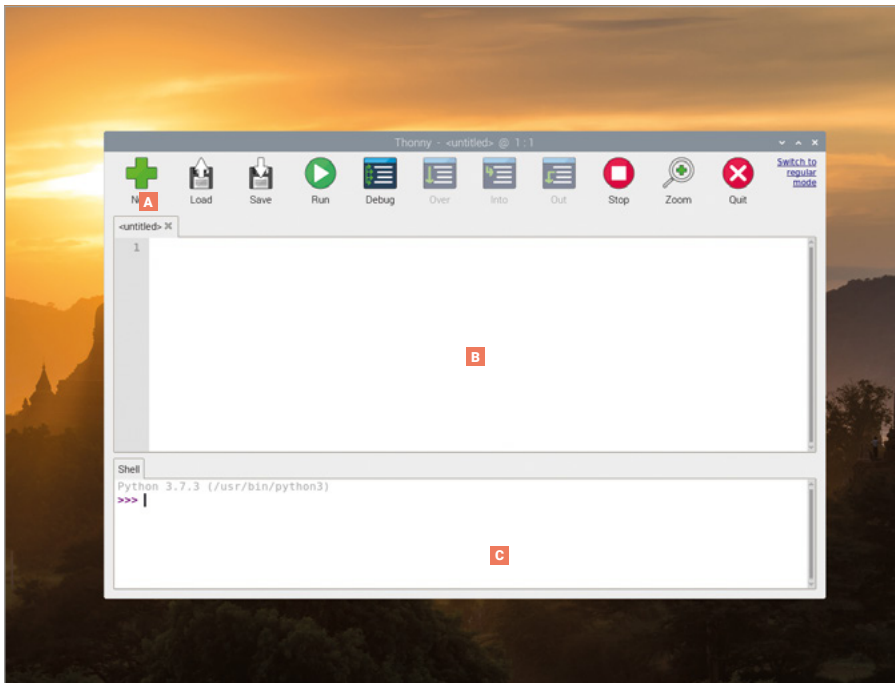


Python, che prende il nome dal gruppo comico Monty Python, dal 1991 ad oggi è passato dall'essere un progetto creato per hobby da Guido van Rossum a un linguaggio di programmazione molto apprezzato alla base di numerosi progetti. A differenza dell'ambiente visivo di Scratch, Python si basa su testo: si scrivono istruzioni in un linguaggio semplificato e con un formato specifico, che poi verranno eseguite dal computer.

Grazie alla flessibilità e a un ambiente di programmazione più tradizionale, è un ottimo programma per fare un passo in più rispetto a Scratch. Non è difficile da apprendere, infatti tutti con un po' di pratica possono imparare a scrivere programmi Python per realizzare i progetti più disparati: da semplici calcoli a giochi più complessi.

In questo capitolo svilupperemo ulteriormente i termini e concetti introdotti nel **Capitolo 4 - Come programmare con Scratch 3**. Ti consigliamo quindi di provare le esercitazioni introdotte nel capitolo precedente, così ti sarà più facile seguire questo capitolo.

Panoramica di Thonny Python IDE



- A Barra degli strumenti:** l'interfaccia semplice di Thonny utilizza come menu una barra di icone intuitive che permettono di creare, salvare, caricare ed eseguire i programmi Python, oltre che testarli in vari modi.
- B Area script:** l'area script è il luogo in cui vengono scritti i programmi Python, ed è suddivisa in un'area principale per il programma e un piccolo margine laterale che mostra i numeri di riga.
- C Python Shell:** consente di digitare singole istruzioni che vengono poi eseguite non appena si preme il tasto **INVIO** e fornisce anche informazioni sull'esecuzione dei programmi.

VERSIONI DI THONNY

Sono disponibili due interfacce per Thonny: normale e semplice, quest'ultima è la soluzione migliore per i principianti. In questo capitolo viene utilizzata la modalità semplice, che viene caricata di default quando si apre Thonny dalla sezione Programmazione del menu a forma di lampone.



Il tuo primo programma Python: ciao a tutti!

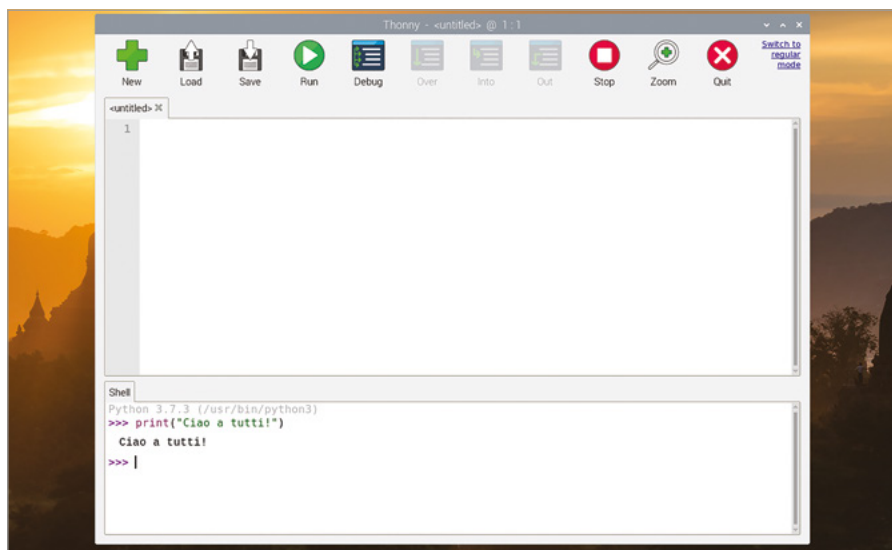
Thonny si apre come qualsiasi altro programma su Raspberry Pi: fai clic sull'icona a forma di lampone per aprire il menu Raspberry Pi OS, sposta il cursore sulla sezione Programmazione e fai clic su Thonny Python IDE. L'interfaccia utente di Thonny si aprirà dopo pochi secondi.

Thonny è un pacchetto che rientra negli *ambienti di sviluppo integrato (integrated development environment, IDE)*, un nome dal suono complicato con una semplice spiegazione: riunisce, o *integra*, tutti i vari strumenti necessari per scrivere, oppure *sviluppare*, software in un'unica interfaccia utente, detta anche *ambiente*. Ci sono numerosi IDE, alcuni di questi supportano molti linguaggi di programmazione diversi, mentre altri, come Thonny, si concentrano su un singolo linguaggio.

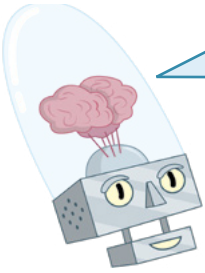
A differenza di Scratch, che consente di creare programmi partendo da blocchi visivi, Python è un linguaggio di programmazione più tradizionale dove è necessario scrivere il codice. Crea il tuo primo programma facendo clic sull'area Python shell in basso a sinistra nella finestra Thonny, quindi digita le seguenti istruzioni prima di premere il tasto **INVIO**:

```
print("Ciao a tutti!")
```

Quando premi **INVIO**, vedrai che il programma viene immediatamente eseguito: Python risponderà nella stessa area shell con il messaggio "Hello, World!" (**Figura 5-1**), proprio come hai chiesto. La shell infatti è collegata direttamente all'*interprete* Python, che ha il compito di esaminare le istruzioni scritte e *interpretarne* il significato. Questa modalità è detta *interattiva* e possiamo paragonarla a una conversazione faccia a faccia con qualcuno: quando hai finito di parlare, l'altra persona risponderà, quindi aspetterà che a tua volta risponda.



▲ **Figura 5-1:** Python stampa il messaggio "Hello, World!" nell'area shell




ERRORE DI SINTASSI

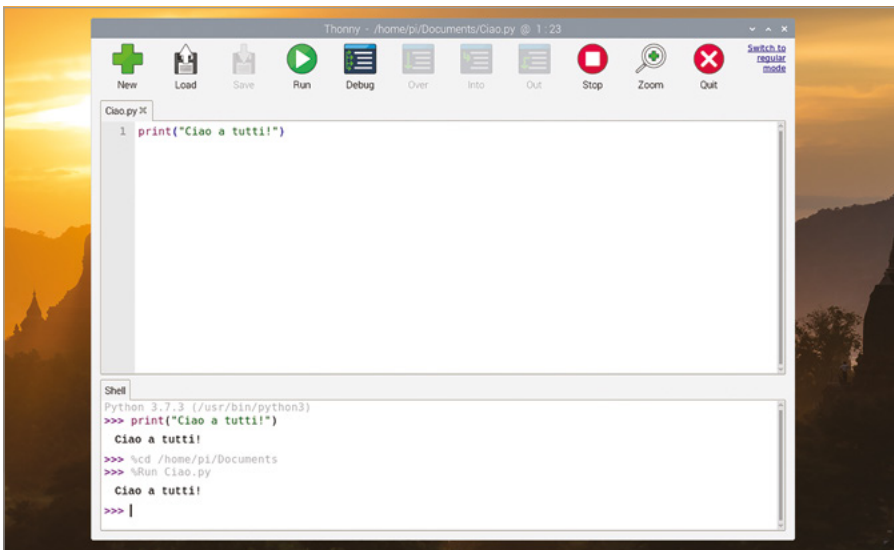
Se il programma non viene eseguito, ma stampa invece un messaggio di errore di sintassi nell'area shell, significa che c'è un errore da qualche parte in quello che hai scritto. In Python è necessario scrivere le istruzioni in un modo molto specifico: se mancano una parentesi o una virgoletta, si scrive "print" in modo sbagliato o con una P maiuscola oppure si aggiungono altri simboli da qualche parte nelle istruzioni, non funzionerà. Prova a digitare di nuovo le istruzioni e assicurati che corrispondano a quanto indicato in questo manuale prima di premere il tasto **INVIO**.

Non è necessario utilizzare sempre Python in modalità interattiva. Fai clic sull'area script sul lato sinistro della finestra Thonny, quindi digita di nuovo il programma:

```
print("Ciao a tutti!")
```

Una volta premuto il tasto **INVIO** questa volta non succederà nulla, ma passerai a una riga vuota nell'area dello script. Per far funzionare questa versione del programma, dovrai fare clic sull'icona  Run nella barra degli strumenti Thonny. Una volta fatto clic sull'icona ti verrà chiesto di salvare il programma, assegnargli un nome, come "Ciao" e fare clic sul pulsante Save. Una volta salvato il programma, nell'area shell Python visualizzerai due messaggi (**Figura 5-2**):

```
>>> %Run 'Ciao.py'
Ciao a tutti!
```



▲ **Figura 5-2:** eseguire un semplice programma

La prima di queste righe è un'istruzione di Thonny che dice all'interprete Python di eseguire il programma appena salvato, mentre la seconda è l'output del programma, ossia il messaggio che hai detto di stampare a Python. Complimenti, hai scritto ed eseguito il tuo primo programma Python sia in modalità interattiva, sia script!




SFIDA: NUOVO MESSAGGIO



È possibile modificare il messaggio che il programma Python stampa come output? Useresti la modalità interattiva o la modalità script per aggiungere altri messaggi? Cosa succede se si rimuovono le parentesi o le virgolette dal programma e poi si cerca di eseguirlo di nuovo?

Prossimi passaggi: loop e rientro del codice

Per controllare il collegamento tra le varie parti del programma Scratch si serve di blocchi simili a puzzle, mentre Python controlla la sequenza in cui vengono eseguiti i programmi tramite il *rientro*. Crea un nuovo programma facendo clic sull'icona  New nella barra degli strumenti Thonny. Non perderai il programma che hai già scritto, Thonny infatti creerà una nuova scheda sopra l'area dello script. Inizia digitando quanto segue:

```
print("Il loop sta iniziando!")  
for i in range(10):
```

La prima riga stampa un semplice messaggio in shell, proprio come il programma Ciao. La seconda avvia un loop *definita* che funziona proprio come in Scratch: al loop viene assegnato un contatore **i** e una serie di numeri (l'istruzione **range**), che indica di contare 10 numeri partendo da 0 (quindi fino a 9). Il simbolo dei due punti (:) indica a Python di includere l'istruzione successiva nel loop.

In Scratch le istruzioni da includere nel loop sono incluse nel blocco a forma di C. Python utilizza un approccio diverso, ossia il rientro del codice. La riga successiva inizia con quattro spazi vuoti aggiunti da Thonny una volta premuto **INVIO** dopo la riga 2:

```
    print("Numero di loop", i)
```

La riga risulterà quindi rientrante rispetto alle altre. Con questo rientro Python distinguerà le istruzioni fuori dal loop da quelle all'interno, il codice rientrante è definito annidato.

Noterai che quando hai premuto **INVIO** alla fine della terza riga, anche quella successiva risulta rientrante, in quanto Thonny ha continuato con il loop. Per rimuovere il rientro, premi semplicemente il tasto **BACKSPACE** prima di digitare la quarta riga:


```
print("Loop ha finito!!")
```

Il tuo programma è completo. La prima riga si trova all'esterno del loop e verrà eseguita solo una volta, la seconda riga avvia il loop, la terza si trova all'interno del loop e verrà eseguita una volta per loop, mentre la quarta si trova all'esterno del loop.

```
print("Il loop sta iniziando!")
for i in range (10):
    print("Numero di loop", i)
print("Loop ha finito!")
```

Fai clic sull'icona Run, salva il programma come **Rientro** e guarda l'output nell'area shell (Figura 5-3):

Il loop sta iniziando!

Numero di loop 0

Numero di loop 1

Numero di loop 2

Numero di loop 3

Numero di loop 4

Numero di loop 5

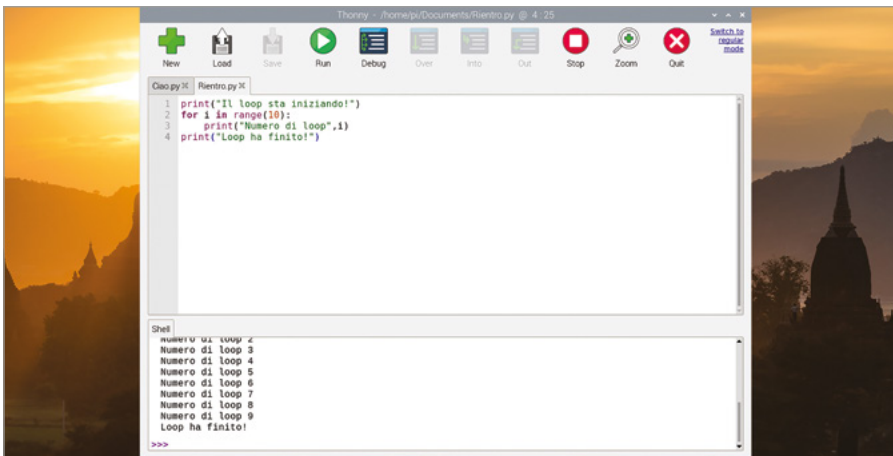
Numero di loop 6

Numero di loop 7

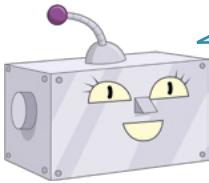
Numero di loop 8

Numero di loop 9

Loop ha finito!



▲ Figura 5-3: esecuzione di un loop



CONTARE DA ZERO

Python è un linguaggio a indice zero, il che significa che inizia a contare da 0 e non da 1, ed è per questo che il programma stampa i numeri da 0 a 9 anziché da 1 a 10. Puoi modificare questo comportamento, se lo desideri, passando dall'istruzione `range(10)` all'istruzione `range(1, 11)` o a qualsiasi altro numero.

Il rientro è una parte fondamentale di Python ed è uno dei motivi più comuni per cui un programma non funziona come ci si aspetta. Quando si cercano problemi in un programma, un processo noto come *debugging*, controlla sempre i rientri soprattutto quando inizi ad annidare le ripetizioni in altre ripetizioni.

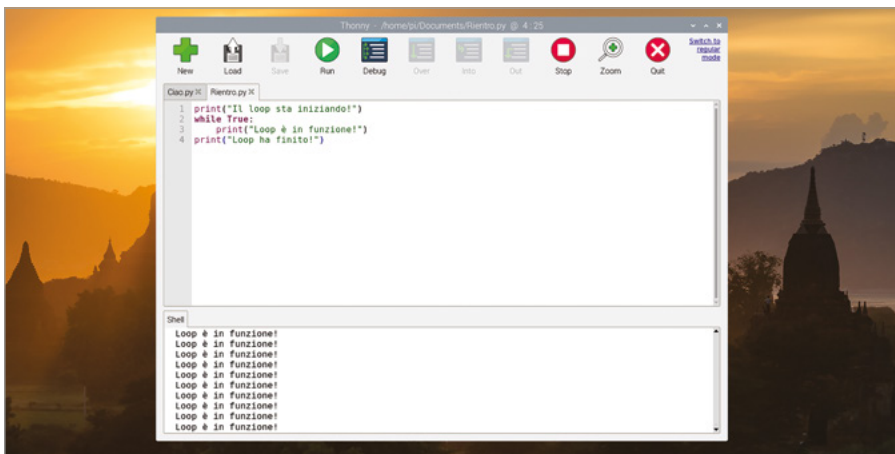
Python supporta anche loop *infinite*. Per passare da un ciclo definito a uno infinito, modifica la riga 2 affinché legga:

```
while True:
```


Se fai clic sull'icona Run, visualizzerai l'errore **name 'i' is not defined** perché hai eliminato la riga che creava e assegnava un valore per la variabile `i`. Per risolvere questo problema, modifica la riga 3 in modo che non utilizzi più la variabile:

```
print("Loop è in funzione!")
```

Fai clic sull'icona Run e, se sei veloce, vedrai il messaggio "Il loop sta iniziando!" seguito da una stringa infinita di messaggi "Loop è in funzione!" (Figura 5-4). Il messaggio "Loop ha finito!" non verrà mai stampato perché il ciclo non ha fine: ogni volta che Python termina di stampare il messaggio "Loop è in funzione!" tornerà all'inizio della ripetizione e stamperà nuovamente.



▲ Figura 5-4: un loop infinito continua fino a quando non si interrompe il programma

Fai clic sull'icona Stop  sulla barra degli strumenti Thonny per dire al programma di interrompere ciò che sta facendo (interruzione del programma). Vedrai un messaggio nell'area shell Python e il programma si arresterà senza raggiungere la riga 4.




SFIDA: LOOPING DEL LOOP



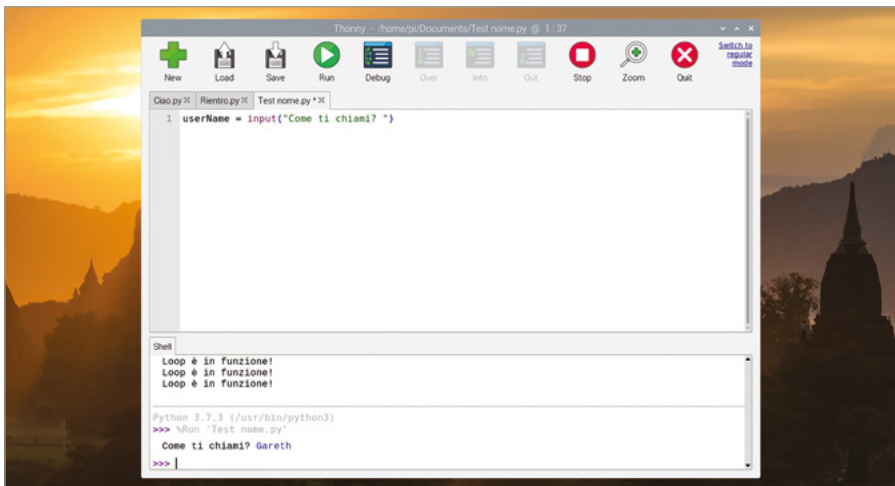
È possibile cambiare il loop e renderla definita? Puoi aggiungere un secondo loop definita al programma? Come aggiungeresti un loop all'interno di un loop e quale comportamento ti aspetteresti?

Condizionali e variabili

Le variabili, come in tutti i linguaggi di programmazione, non servono solamente a controllare le ripetizioni. Crea un nuovo programma facendo clic sull'icona New  nella barra degli strumenti Thonny, quindi nell'area script digita quanto segue:

```
userName = input ("Come ti chiami? ")
```

Fai clic sull'icona Run, salva il programma come **Test nome** e guarda cosa succede nell'area shell: ti verrà chiesto il nome. Digita il tuo nome nell'area shell, seguito da **INVIO**. Poiché questa è l'unica istruzione del programma, non accadrà nient'altro (**Figura 5-5**). Se vuoi fare qualcosa con i dati inseriti nella variabile, avrai bisogno di più righe nel programma.



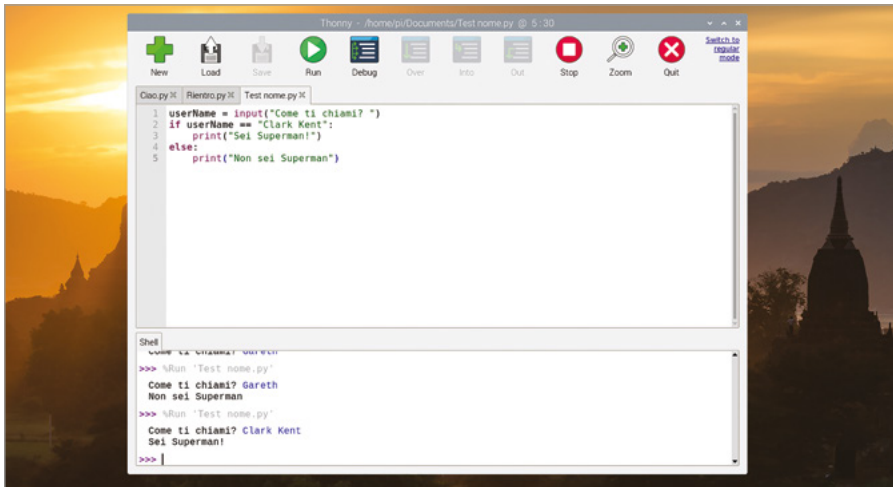
▲ **Figura 5-5:** la funzione `input` ti consente di chiedere a un utente di inserire testo

Per far sì che il programma faccia qualcosa con il nome, aggiungi un'istruzione condizionale digitando quanto segue:

```
if userName == "Clark Kent":
    print("Sei Superman!")
else:
    print("Non sei Superman!")
```

Ricorda che quando Thonny riconosce che il codice deve essere rientrante inserirà automaticamente un rientro, tuttavia non sa quando il codice non deve più essere rientrante, quindi dovrai cancellare gli spazi.

Fai clic sull'icona Run e digita il tuo nome nell'area shell. A meno che il tuo nome non sia Clark Kent, visualizzerai il messaggio "Non sei Superman! Fai di nuovo clic su Run e questa volta digita il nome "Clark Kent" assicurandoti di scriverlo esattamente come nel programma, con C e K maiuscole (Figura 5-6).



▲ Figura 5-6: non dovresti essere in giro a salvare il mondo?

I simboli == indicano a Python di fare un confronto diretto e quindi individuare se la variabile **userName** corrisponde al testo (*stringa*) nel programma. Se lavori con i numeri puoi fare altri confronti: > per vedere se un numero è maggiore di un altro numero, < per vedere invece è minore, ==> per vedere se è uguale o superiore, ==< per vedere se è uguale o inferiore. != invece significa che non è uguale, ossia l'esatto opposto di ==. Questi simboli sono tecnicamente noti come *operatori di confronto*.



SFIDA: AGGIUNGI ALTRE DOMANDE



Puoi modificare il programma per porre più di una domanda, memorizzando le risposte in diverse variabili? Puoi realizzare un programma che utilizza condizionali e operatori di confronto per stampare se un numero digitato dall'utente è superiore o inferiore a 5, come il programma creato nel **Capitolo 4 - Come programmare con Scratch?**

Progetto 1: fiocchi di neve con turtle


Ora che hai capito come funziona Python, è il momento di giocare con la grafica e creare un fiocco di neve utilizzando uno strumento noto come *turtle* (tartaruga).

PROGETTO ONLINE

Questo progetto è disponibile anche online all'indirizzo rpf.io/turtle-snowflakes



Originariamente robot fisici simili all'animale omonimo, le tartarughe sono progettate per muoversi in linea retta, girare e sollevare e abbassare una penna, nella versione digitale questo significa semplicemente iniziare o smettere di disegnare una linea mentre si muove. A differenza di altri linguaggi, come Logo e le sue numerose varianti, Python non ha uno strumento turtle incorporato, ma viene fornito con una *libreria* di codice aggiuntivo che consente di operare similmente. Le librerie sono bundle di codice che aggiungono nuove istruzioni per espandere le capacità di Python e puoi importarle nei tuoi programmi utilizzando un comando di importazione.

Crea un nuovo programma facendo clic sull'icona New  e digita quanto segue:

```
import turtle
```

Quando si utilizzano le istruzioni incluse in una libreria è necessario utilizzare il nome della libreria seguito da un punto, quindi il nome dell'istruzione. Per non doverlo digitare ogni volta puoi assegnare un nome di variabile più corto, anche solo una lettera, ma abbiamo pensato di dare un nome alla "tartaruga". Digita quanto segue:

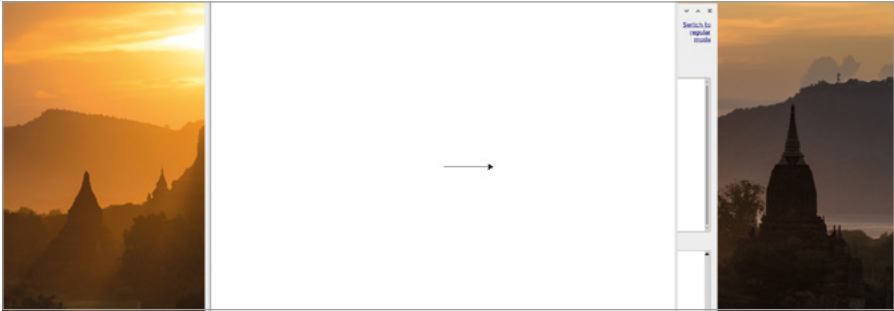
```
pat = turtle.Turtle()
```

Per testare il programma, dovrai assegnare un compito alla tartaruga. Digita:

```
pat.forward(100)
```

Fai clic sull'icona Run e salva il tuo programma come **Fiocchi di neve tartaruga**. Una volta salvato il programma, visualizzerai una nuova finestra chiamata "Turtle Graphics" e vedrai il

risultato del programma: la tartaruga Pat si sposterà in avanti di 100 unità disegnando una linea retta (**Figura 5-8**).



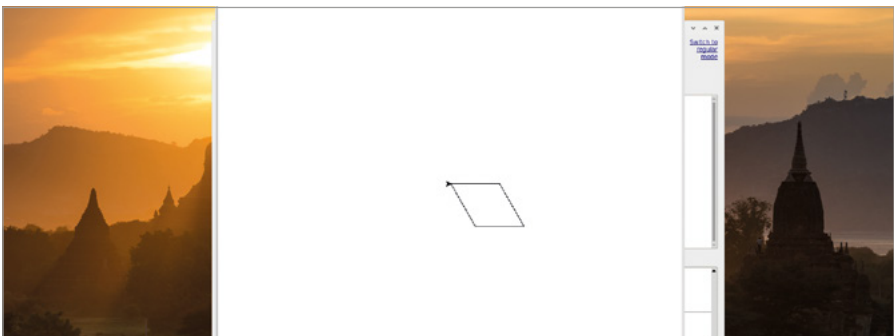
▲ **Figura 5-8:** la tartaruga si sposta in avanti per tracciare una linea retta

Torna alla finestra principale Thonny e fai clic sul pulsante Stop per chiudere la finestra Turtle Graphics, qualora questa nasconda la finestra principale, fai clic sul pulsante minimizza nella finestra Turtle Graphics o fai clic sulla voce Thonny nella barra delle applicazioni nella parte superiore dello schermo.

Per non dover digitare ogni singola istruzione di movimento, cancella la riga 3 e crea una ripetizione per creare le forme:

```
for i in range(2):
    pat.forward(100)
    pat.right(60)
    pat.forward(100)
    pat.right(120)
```

Esegui il programma e Pat disegnerà un parallelogramma (**Figura 5-9**).



▲ **Figura 5-9:** combinando curve e movimenti, puoi disegnare forme

Per creare una forma simile a un fiocco di neve, fai clic sull'icona Stop nella finestra principale di Thonny e inserisci un loop attorno al loop aggiungendo la seguente riga come riga 3:

```
for i in range(10):
```

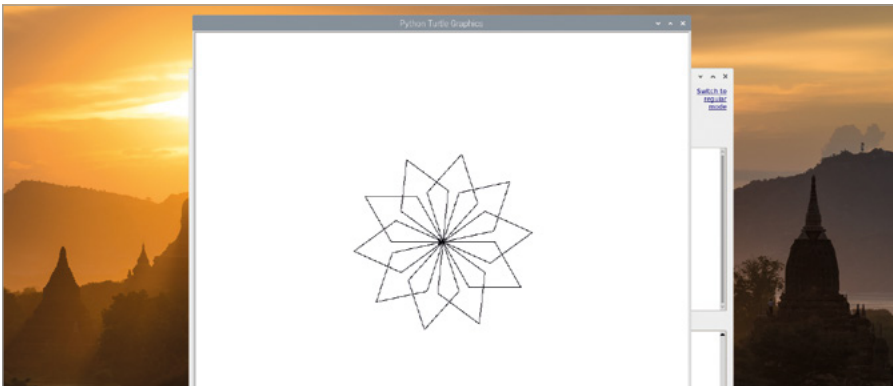
... e quanto segue alla fine del programma:

```
pat.right(36)
```

Il programma non funzionerà così com'è scritto, perché la ripetizione esistente non presenta un rientro corretto. Per risolvere il problema, fai clic sull'inizio di ogni linea della ripetizione (linee da 4 a 8) e premi il tasto **SPAZIO** quattro volte per correggere il rientro. Il programma dovrebbe avere questa struttura:

```
import turtle
pat = turtle.Turtle()
for i in range(10):
    for i in range(2):
        pat.forward(100)
        pat.right(60)
        pat.forward(100)
        pat.right(120)
    pat.right(36)
```

Fai clic sull'icona Run, e guarda la tartaruga: disegnerà un parallelogramma, come prima, ma quando avrà finito girerà di 36 gradi e ne disegnerà un altro, poi un altro, e così via fino a quando ci saranno dieci parallelogrammi sovrapposti sullo schermo, uno schema per realizzare il fiocco di neve (**Figura 5-10**).

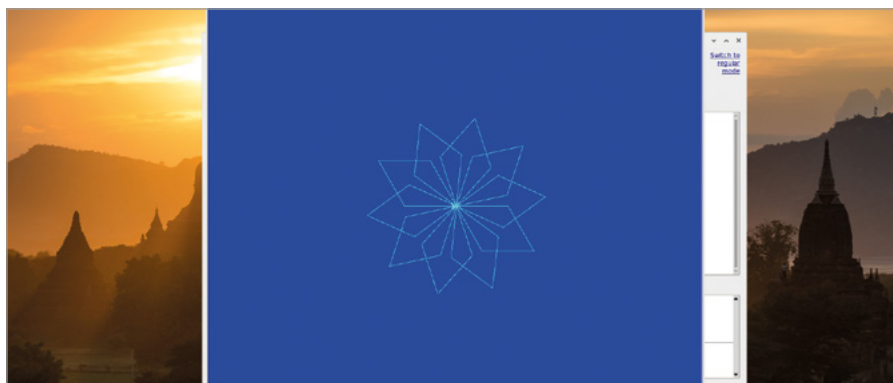


▲ **Figura 5-10:** ripetere la forma per realizzarne una più complessa

Mentre una tartaruga robot disegna in un unico colore su un grande pezzo di carta, la tartaruga simulata di Python può utilizzare vari colori. Aggiungi una nuova linea 3 e 4, facendo andare verso il basso le linee già presenti:

```
turtle.Screen().bgcolor("blue")
pat.color("cyan")
```

Esegui di nuovo il programma e vedrai l'effetto del nuovo codice: il colore di sfondo della finestra Turtle Graphics è diventato blu, mentre il fiocco di neve ora è ciano (**Figura 5-11**).



▲ **Figura 5-11: modifica del colore di sfondo e fiocchi di neve**

Puoi anche far scegliere i colori in modo casuale da una lista utilizzando la libreria **random**. Torna all'inizio del programma e inserisci quanto segue come riga 2:

```
import random
```

Cambia il colore di sfondo in quella che ora è la riga 4 da **"blue"** (blu) a **"grey"** (grigio), quindi crea una nuova variabile chiamata "colours" inserendo una nuova riga 5:

```
colours = ["cyan", "purple", "white", "blue"]
```



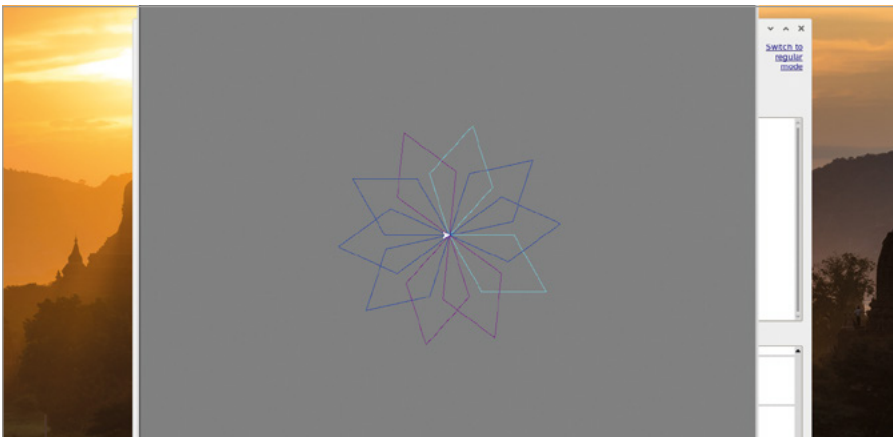
INGLESE STATUNITENSE

Molti linguaggi di programmazione usano l'inglese americano e Python non fa eccezione: il comando per cambiare il colore della penna turtle si scrive *color* e se lo si scrive con l'ortografia inglese britannica, ossia *colour*, non funzionerà. Tuttavia per le variabili puoi usare l'ortografia che preferisci, ecco perché puoi chiamare la nuova variabile *colours* e Python comprenderà ugualmente.

Questo tipo di variabile è noto come elenco ed è contrassegnato da parentesi quadre. In questo caso, la lista è piena di possibili colori per i segmenti del fiocco di neve, ma è comunque necessario indicare a Python di sceglierne uno ogni volta che il ciclo si ripete. Alla fine del programma inserisci quanto segue, assicurandoti che ogni riga abbia quattro spazi di rientranza all'inizio in modo che faccia parte della ripetizione esterna, proprio come la linea sopra:

```
pat.color(random.choice(colours))
```

Fai di nuovo clic sull'icona Run e verrà ancora disegnato il fiocco di neve. Questa volta però Python sceglierà casualmente un colore dalla lista mentre disegna ogni petalo, dando al fiocco di neve un aspetto multicolore (Figura 5-12).



▲ Figura 5-12: utilizzo di colori casuali per i "petali"

Per far sembrare il fiocco di neve meno simile a una stella ninja e più simile a un vero e proprio fiocco di neve, aggiungi una nuova riga 6, direttamente sotto alla lista **colours** e digita quanto segue:

```
pat.penup()  
pat.forward(90)  
pat.left(45)  
pat.pendown()
```

Le istruzioni **penup** e **pendown** farebbero muovere la penna sulla carta in modo da creare dei tratti con il robot tartaruga, ma nel mondo virtuale dicono semplicemente alla tartaruga di iniziare e smettere di disegnare linee. Questa volta, però, invece di usare una ripetizione, creerai una *funzione*, ossia un segmento di codice richiamabile in qualsiasi momento, così avrai creato un'istruzione Python.

Inizia cancellando il codice per disegnare i fiocchi di neve basati su parallelogrammi, ossia tutto ciò che è compreso tra l'istruzione `pat.color("cyan")` sulla linea 10 fino a `pat.right(36)` sulla linea 17 incluse. Lascia `pat.color(random.choice(colours))`, ma aggiungi il simbolo cancelletto (`#`) all'inizio della linea. Questo è un *commentofuori* da un'istruzione, e significa che Python lo ignorerà. Puoi usare i commenti per aggiungere spiegazioni al codice, così sarà molto più facile da capire quando tornerai a lavorarci dopo un po' di tempo o lo invierai a qualcun altro.

Crea la funzione, che sarà chiamata "branch" (linee), digitando la seguente istruzione sulla riga 10 sotto a `pat.pendown()`:

```
def branch():
```

Questo *definisce* la funzione `branch`. Quando premi il tasto **INVIO** Thonny aggiungerà automaticamente una rientranza per le istruzioni della funzione. Digita quanto segue facendo attenzione ai rientri perché a un certo punto dovrai annidare il codice con tre livelli di rientro.

```
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
```

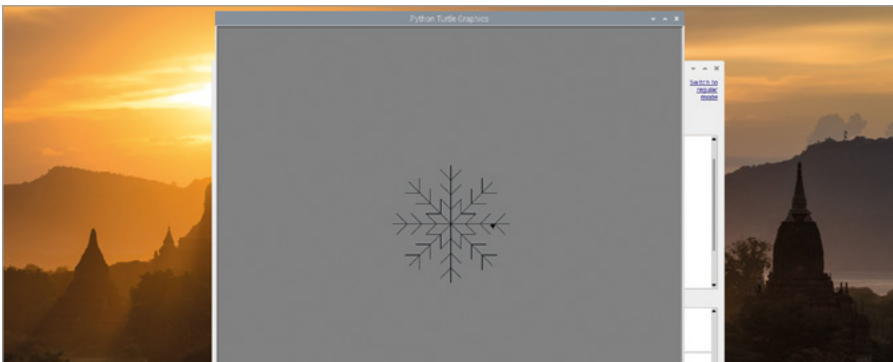
Infine, crea una nuova ripetizione da eseguire in fondo al programma, ma sopra la linea del colore commentata, oppure *richiama* la nuova funzione:

```
for i in range(8):
    branch()
    pat.left(45)
```

Il programma finito dovrebbe avere questa struttura:

```
import turtle
import random
pat = turtle.Turtle()
turtle.Screen().bgcolor("grey")
colours = ["cyan", "purple", "white", "blue"]
pat.penup()
pat.forward(90)
pat.left(45)
pat.pendown()
def branch():
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
for i in range(8):
    branch()
    pat.left(45)
# pat.color(random.choice(colours))
```

Fai clic su Run e guarda la finestra grafica mentre Pat disegna seguendo le istruzioni. Complimenti, adesso sembra un vero un fiocco di neve (**Figura 5-13**)!



▲ **Figura 5-13:** le linee fanno sembrare la figura un fiocco di neve



SFIDA: E ORA?

Puoi usare le istruzioni commentate per far disegnare le linee del fiocco di neve in diversi colori? Puoi creare una funzione "fiocco di neve" e utilizzarla per disegnare molti fiocchi di neve sullo schermo? Puoi far cambiare al programma la dimensione e il colore dei fiocchi di neve in modo casuale?



Progetto 2: uno spaventoso trova le differenze

Python può anche gestire immagini e suoni nonché la grafica basata su turtle, così potrai fare divertenti scherzi ai tuoi amici con uno spaventoso gioco in cui trovare le differenze, perfetto per Halloween.

PROGETTO ONLINE

Questo progetto è disponibile anche online all'indirizzo rpf.io/scary-spot



Per questo progetto sono necessari un file audio e due immagini: una per trovare le differenze e una "spaventosa" immagine a sorpresa. Fai clic sull'icona a forma di lampone per caricare il menu Raspberry Pi OS, seleziona la categoria Internet e fai clic sul browser Web Chromium. Una volta aperto il browser, digita **rpf.io/spot-pic** nella barra degli indirizzi e premi il tasto **INVIO**. Fai clic con il tasto destro del mouse sull'immagine dello spazio, fai clic su "Salva immagine con nome...", seleziona la cartella Home dall'elenco a sinistra, quindi fai clic su Salva. Fai clic nuovamente sulla barra degli indirizzi di Chromium, quindi digita **rpf.io/scary-pic** e premi il tasto **INVIO**. Come prima, fai clic con il tasto destro del mouse sull'immagine, quindi su "Salva immagine con nome...", seleziona la cartella Home e fai clic su Salva.

Fai nuovamente clic sulla barra degli indirizzi e digita **rpf.io/scream** e premi il tasto **INVIO** per scaricare il file audio. Il file (il suono di un urlo) verrà riprodotto automaticamente durante il gioco, sorprendendo i giocatori, tuttavia lo dovrai salvare prima di utilizzarlo. Fai clic con il tasto destro del mouse sul lettore audio, quindi su "Salva con nome...", seleziona la cartella Home e fai clic su Salva. Puoi chiudere la finestra Chromium.

Crea un nuovo progetto facendo clic sull'icona New nella barra degli strumenti Thonny. Come in precedenza, utilizzerai una libreria per estendere le capacità di Python: la libreria Pygame, che come suggerisce il nome è stata creata pensando ai giochi. Digita quanto segue:

```
import pygame
```

Necessiterai anche di parti di altre librerie e di una sottosezione della biblioteca Pygame. Importale digitando quanto segue:

```
from pygame.locals import *
from time import sleep
from random import randrange
```

L'istruzione **from** è diversa da **import**, infatti ti consente di importare solo le parti di una libreria che ti occorrono, anziché una libreria intera. Successivamente dovrai impostare Pygame, questo processo è noto come *inizializzazione*. Pygame deve conoscere la larghezza e l'altezza del monitor o del televisore del giocatore, ossia la *risoluzione*. Digita quanto segue:

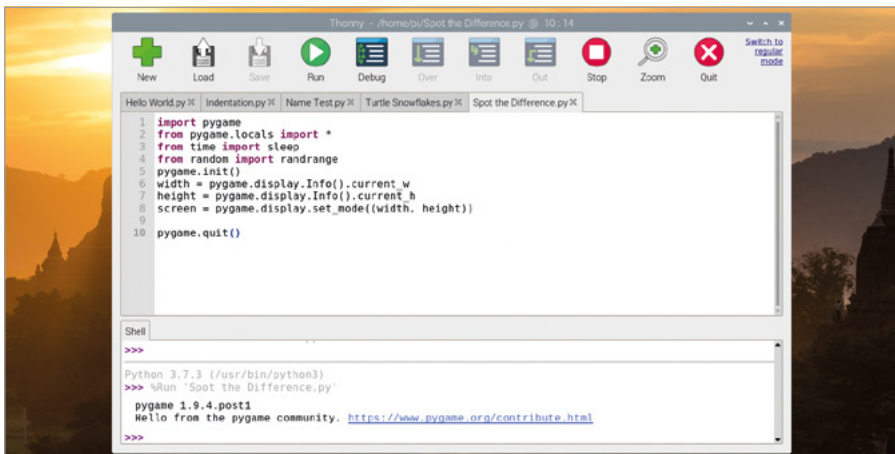
```
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
```

L'ultimo passaggio per impostare Pygame è quello di creare la sua finestra, ossia lo "screen" di Pygame. Digita quanto segue:

```
screen = pygame.display.set_mode((width, height))
```

```
pygame.quit()
```

Il tuo programma andrà nella riga bianca al centro. Per ora, però, fai clic sull'icona Run, salva il programma come **Trova le differenze** e guarda: Pygame creerà una finestra con uno sfondo nero, che scomparirà quasi immediatamente non appena raggiungerà l'istruzione di uscire. A parte un breve messaggio nell'area shell (**Figura 5-14**), non abbiamo fatto molto con il programma.



▲ **Figura 5-14:** il programma è funzionale, ma non fa ancora molto

Per visualizzare l'immagine in cui trovare le differenze, digita la seguente riga nello spazio soprastante `pygame.quit()`:

```
difference = pygame.image.load('spot_the_diff.png')
```

Per assicurarti che l'immagine riempia lo schermo, dovrai ridimensionarla in modo che corrisponda alla risoluzione del monitor o della TV. Digita quanto segue:

```
difference = pygame.transform.scale(difference, (width, height))
```

Ora che l'immagine è stata memorizzata, è necessario dire a Pygame di mostrarla sullo schermo, un processo noto come *blitting* o *trasferimento a blocchi di bit*. Digita quanto segue:

```
screen.blit(difference, (0, 0))  
pygame.display.update()
```

La prima di queste righe copia l'immagine sullo schermo, partendo dall'angolo in alto a sinistra, mentre la seconda indica a Pygame di ridisegnare lo schermo. Senza questa seconda riga, l'immagine sarà memorizzata correttamente, ma non la vedrai.

Fai di nuovo clic sull'icona Run e l'immagine verrà brevemente visualizzata sullo schermo (Figura 5-15).



▲ **Figura 5-15: la tua immagine per il gioco trova le differenze**

Per vedere l'immagine più a lungo, aggiungi la seguente riga appena sopra a `pygame.quit()`:

```
sleep(3)
```

Fai di nuovo clic su Run e l'immagine rimarrà sullo schermo più a lungo. Aggiungi l'immagine a sorpresa digitando quanto segue appena sotto la riga `pygame.display.update()`:

```
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
```

Aggiungi un ritardo, in modo che l'immagine dello zombie non appaia subito:

```
sleep(3)
```

Esegui il blit dell'immagine sullo schermo e aggiornala in modo da mostrarla al giocatore:

```
screen.blit(zombie, (0,0))
pygame.display.update()
```

Fai clic sull'icona Run e guarda cosa succede: Pygame caricherà l'immagine per il gioco trova le differenze, ma dopo tre secondi sarà sostituita da uno spaventoso zombie (Figura 5-16)!



▲ Figura 5-16: uno scherzetto spaventoso

Il ritardo di tre secondi rende il tutto abbastanza prevedibile. Cambia la riga `sleep(3)` sopra a `screen.blit(zombie, (0,0))` in:

```
sleep(randrange(5, 15))
```

Verrà scelto un numero casuale tra 5 e 15, che corrisponderà al tempo di ritardo. Successivamente aggiungi la seguente riga appena sopra all'istruzione `sleep` per caricare il file del suono dell'urlo:

```
scream = pygame.mixer.Sound('scream.wav')
```

Spostati sotto l'istruzione `sleep` e digita le seguenti istruzioni su una nuova riga per avviare la riproduzione del suono, in modo che si attivi appena prima dell'immagine spaventosa che viene mostrata al giocatore:


```
scream.play()
```

Infine indica a Pygame di smettere di riprodurre il suono digitando la seguente riga appena sopra a `pygame.quit()`:

```
scream.stop()
```

Fai clic sull'icona Run e ammira il tuo lavoro: dopo qualche secondo di innocente divertimento con il gioco per trovare le differenze, il tuo terribile zombie apparirà dopo un urlo che sicuramente spaventerà i tuoi amici. Se l'immagine dello zombie appare prima che il suono venga riprodotto, puoi aggiungere un piccolo ritardo subito dopo l'istruzione `scream.play()` e prima di `screen.blit`:

```
sleep(0.4)
```

Il programma finito dovrebbe avere questa struttura:

```
import pygame
from pygame.locals import *
from time import sleep
from random import randrange
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
screen = pygame.display.set_mode((width, height))
difference = pygame.image.load('spot_the_diff.png')
difference = pygame.transform.scale(difference, (width, height))
screen.blit(difference, (0, 0))
pygame.display.update()
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
scream = pygame.mixer.Sound('scream.wav')
sleep(randrange(5, 15))
scream.play()
screen.blit(zombie, (0,0))
pygame.display.update()
sleep(3)
scream.stop()
pygame.quit()
```

Ora non ti resta che invitare i tuoi amici a giocare a questo gioco e ovviamente assicurarti che il volume sia attivo.



SFIDA: CAMBIA L'ASPETTO



Puoi cambiare le immagini per rendere lo scherzo più appropriato per altri eventi, come il Natale? Puoi disegnare le immagini per il gioco in cui trovare le differenze e le immagini con cui spaventare i tuoi amici usando ad esempio un software grafico come GIMP? Puoi tenere traccia delle azioni dell'utente quando fa clic su una differenza, per rendere il tutto più convincente?

Progetto 3: labirinto per gioco di ruolo


Ora che stai imparando a usare Python, è il momento di usare Pygame per realizzare qualcosa di un po' più complicato: un gioco funzionante basato su testo con un labirinto, ispirato ai classici giochi di ruolo. Noti come avventure testuali o narrativa interattiva, questi giochi risalgono a quando i computer non erano in grado di gestire la grafica e hanno ancora appassionati che sostengono che nessuna grafica sarà mai vivida quanto l'immaginazione.

PROGETTO ONLINE

Questo progetto è disponibile anche online all'indirizzo rpf.io/python-rpg

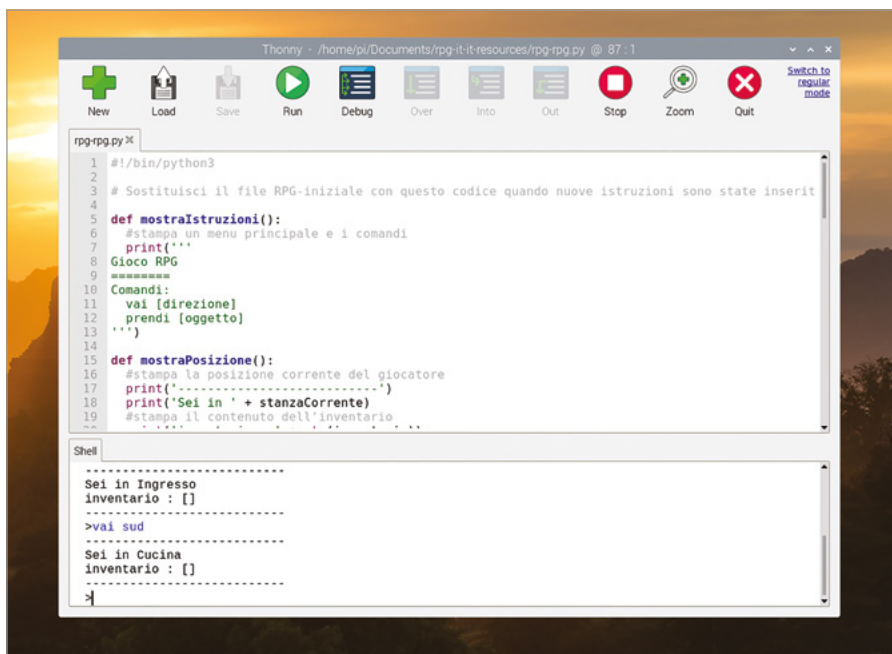


Questo programma è un po' più complesso rispetto a quelli presentati prima, quindi per rendere le cose più semplici inizierai con una versione già parzialmente scritta. Apri il browser Chromium e accedi al seguente indirizzo: rpf.io/rp/it-IT/rpg-go.

Il browser Chromium scaricherà automaticamente il codice del programma nella cartella Download, ma ti avvertirà che il tipo di file (un programma Python) potrebbe danneggiare il computer. Hai scaricato il file da Raspberry Pi Foundation, una fonte affidabile, quindi nel messaggio visualizzato in basso nello schermo fai clic sul pulsante per mantenere il file. Torna a Thonny e fai clic sull'icona Load.  Cerca il file, **rpg-rpg.py**, nella cartella Downloads e fai clic sul pulsante Load.

Inizia facendo clic sull'icona Run per prendere familiarità con il funzionamento di un'avventura testuale. L'output del gioco viene visualizzato nell'area shell nella parte inferiore della finestra Thonny, ingrandisci la finestra Thonny facendo clic sul pulsante per ingrandirla e renderla più facile da leggere.

Il gioco, così com'è ora, è molto semplice: ci sono due stanze e nessun oggetto. Il giocatore inizia nell'ingresso, la prima delle due stanze. Per andare in cucina, digita semplicemente "vai sud" seguito dal tasto **INVIO (Figura 5-17)**. Quando sei in cucina, puoi digitare "vai nord" per tornare all'ingresso. Puoi anche provare a digitare "vai ovest" e "vai est", ma dato che non ci sono stanze in quelle direzioni il gioco ti mostrerà un messaggio di errore.



▲ Figura 5-17: finora ci sono solo due stanze

Scorri fino alla riga 29 del programma nell'area script per trovare una variabile chiamata **stanze**. Questo tipo di variabile è noto come *dizionario* e indica al gioco le stanze, le loro uscite e a quale stanza porta una determinata uscita.

Per rendere il gioco più interessante, aggiungi un'altra stanza: una sala da pranzo, a est dell'ingresso. Trova la variabile **stanze** nell'area degli script ed estendila aggiungendo un simbolo di virgola (,) dopo } sulla riga 38, digitando poi quanto segue (in un dizionario non è necessario inserire rientri precisi):

```

'Sala da pranzo' : {
    'ovest' : 'Ingresso'
}

```

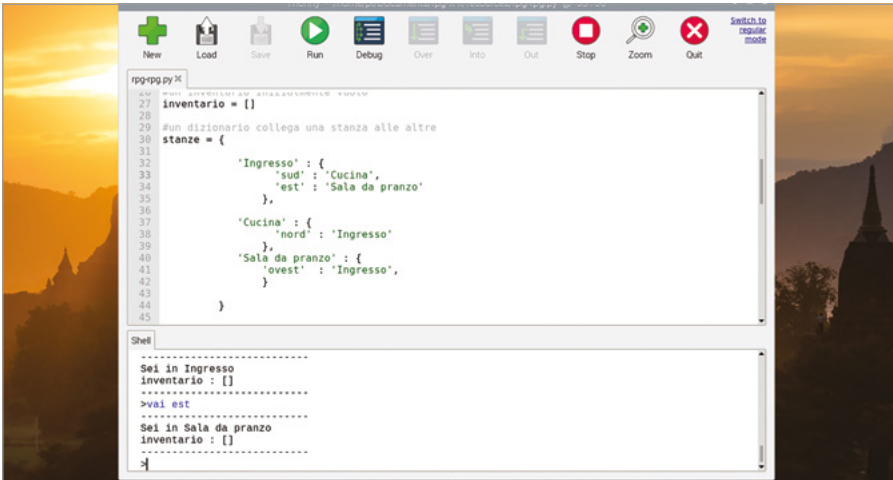
Avrai anche bisogno di una nuova uscita nell'ingresso, in quanto non viene creata automaticamente. Vai alla fine della riga 33, aggiungi una virgola, quindi aggiungi la riga seguente:

```

'est' : 'Sala da pranzo'

```

Fai clic sull'icona Run, e prova la nuova stanza: digita "vai est" mentre sei nell'ingresso per entrare nella sala da pranzo (**Figura 5-18**, sul retro), e digita "vai ovest" mentre sei nella sala da pranzo per accedere all'ingresso. Complimenti, hai creato la tua stanza!

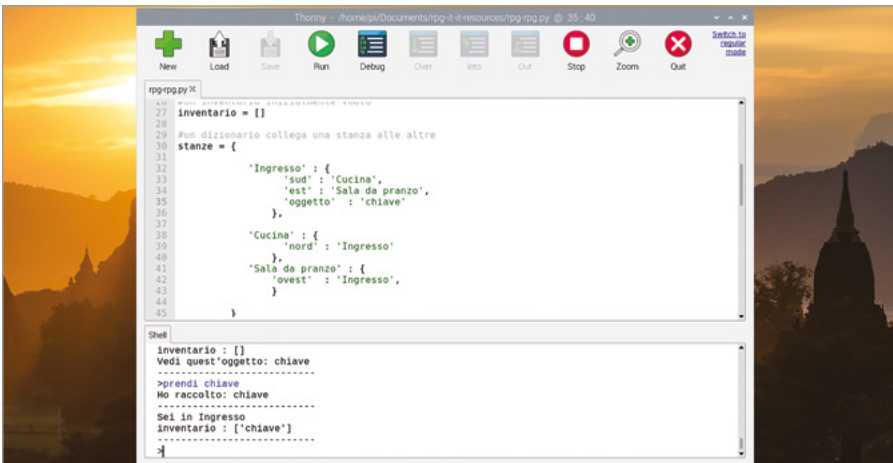


▲ Figura 5-18: hai aggiunto un'altra stanza


Le stanze vuote non sono molto divertenti. Per aggiungere un elemento a una stanza, dovrai modificare il dizionario della stanza. Ferma il programma facendo clic sull'icona Stop. Trova il dizionario **Hall** nell'area degli script, quindi aggiungi una virgola alla fine della riga **'est'** : **'Sala da pranzo'** prima di premere **INVIO** e digita la seguente riga:

'oggetto' : 'chiave'

Fai di nuovo clic su Run. Questa volta il gioco ti dirà che puoi vedere il nuovo oggetto: una chiave. Digita "get key" (**Figura 5-19**) per ritrarlo, aggiungendolo alla lista degli oggetti che stai trasportando, il tuo *inventario*. L'inventario rimane con te mentre passi da una stanza all'altra.



▲ Figura 5-19: la chiave raccolta viene aggiunta all'inventario

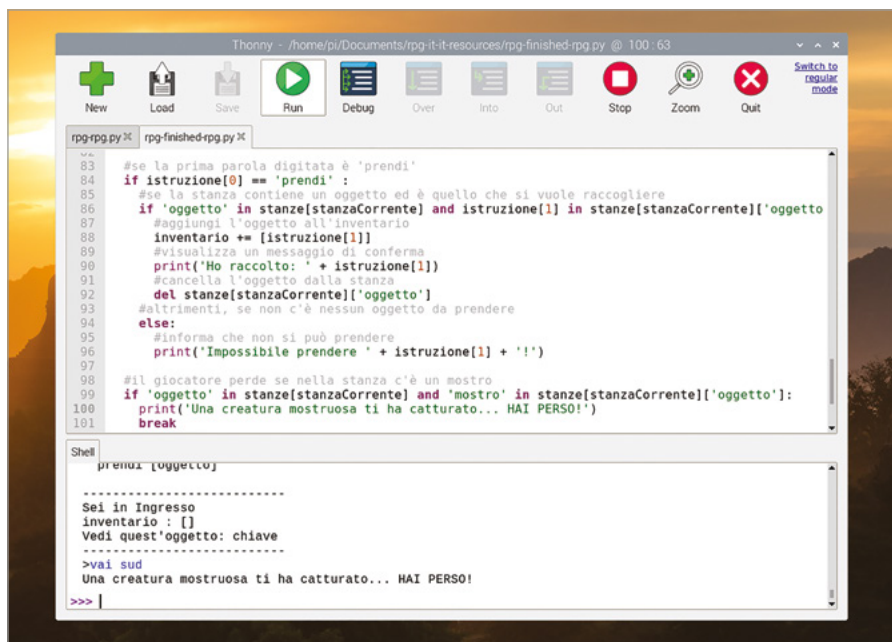
Fai clic sull'icona Stop  e rendi il gioco più interessante aggiungendo un mostro da evitare. Trova il dizionario **Cucina** e aggiungi una voce "mostro" nello stesso modo in cui hai aggiunto la voce "chiave", ricordandoti di aggiungere una virgola alla fine della riga precedente:

```
'oggetto' : 'mostro'
```

Per far sì che il mostro sia in grado di attaccare il giocatore, è necessario aggiungere elementi di logica al gioco. Scorri fino in fondo al programma nell'area dello script e aggiungi le seguenti righe (compreso il commento, contrassegnato da un cancelletto, che ti aiuterà a capire il programma se ci lavorerai un altro giorno) e assicurati di far rientrare le righe:

```
# il giocatore perde se nella stanza c'è un mostro
if 'oggetto' in stanze[stanzaCorrente] and 'mostro' in
stanze[stanzaCorrente]['oggetto']:
    print('Una creatura mostruosa ti ha catturato... HAI
PERSO!')
    break
```

Fai clic su Run e prova ad andare nella stanza della cucina (**Figura 5-20**), il mostro non sarà sorpreso.



```
Thonny - /home/pi/Documents/rpg-it-resources/rpg-finished-rpg.py @ 100 63
New Load Save Run Debug Over Info Out Stop Zoom Quit Switch to regular mode

rpg-rpg.py x rpg-finished-rpg.py x
83 #se la prima parola digitata è 'prendi'
84 if istruzione[0] == 'prendi' :
85     #se la stanza contiene un oggetto ed è quello che si vuole raccogliere
86     if 'oggetto' in stanze[stanzaCorrente] and istruzione[1] in stanze[stanzaCorrente]['oggetto']
87     #aggiungi l'oggetto all'inventario
88     inventario += [istruzione[1]]
89     #visualizza un messaggio di conferma
90     print('Ho raccolto: ' + istruzione[1])
91     #cancella l'oggetto dalla stanza
92     del stanze[stanzaCorrente]['oggetto']
93     #altrimenti, se non c'è nessun oggetto da prendere
94     else:
95     #informa che non si può prendere
96     print('Impossibile prendere ' + istruzione[1] + '!')
97
98 #il giocatore perde se nella stanza c'è un mostro
99 if 'oggetto' in stanze[stanzaCorrente] and 'mostro' in stanze[stanzaCorrente]['oggetto']:
100     print('Una creatura mostruosa ti ha catturato... HAI PERSO!')
101     break

Shell
prende [oggetto]
-----
Sei in Ingresso
inventario : []
Vedi quest'oggetto: chiave
-----
>vai sud
Una creatura mostruosa ti ha catturato... HAI PERSO!
>>>
```

▲ **Figura 5-20:** che paura fa un topo quando c'è un mostro in cucina

Per trasformare quest'avventura in un vero e proprio gioco avrai bisogno di più oggetti, di un'altra stanza e della possibilità di vincere lasciando la casa con tutti gli oggetti al sicuro nel tuo inventario. Inizia aggiungendo un'altra stanza, proprio come hai fatto per la sala da pranzo, solo che questa volta si tratta di un giardino. Aggiungi un'uscita dal dizionario della sala da pranzo, ricordandoti di aggiungere una virgola alla fine della riga precedente:

```
'sud' : 'Giardino'
```

Quindi aggiungi la tua nuova stanza al dizionario **stanze** principale, ricordandoti sempre di aggiungere una virgola dopo `}` sulla riga precedente:

```
'Giardino' : {  
    'nord' : 'Sala da pranzo'  
}
```

Aggiungi un oggetto "pozione" al dizionario della sala da pranzo, ricordandoti di aggiungere una virgola nella riga precedente:

```
'oggetto' : 'pozione'
```

Infine, scorri in fondo al programma e aggiungi la logica necessaria per controllare se il giocatore ha tutti gli oggetti e, in caso affermativo, digli che ha vinto la partita:

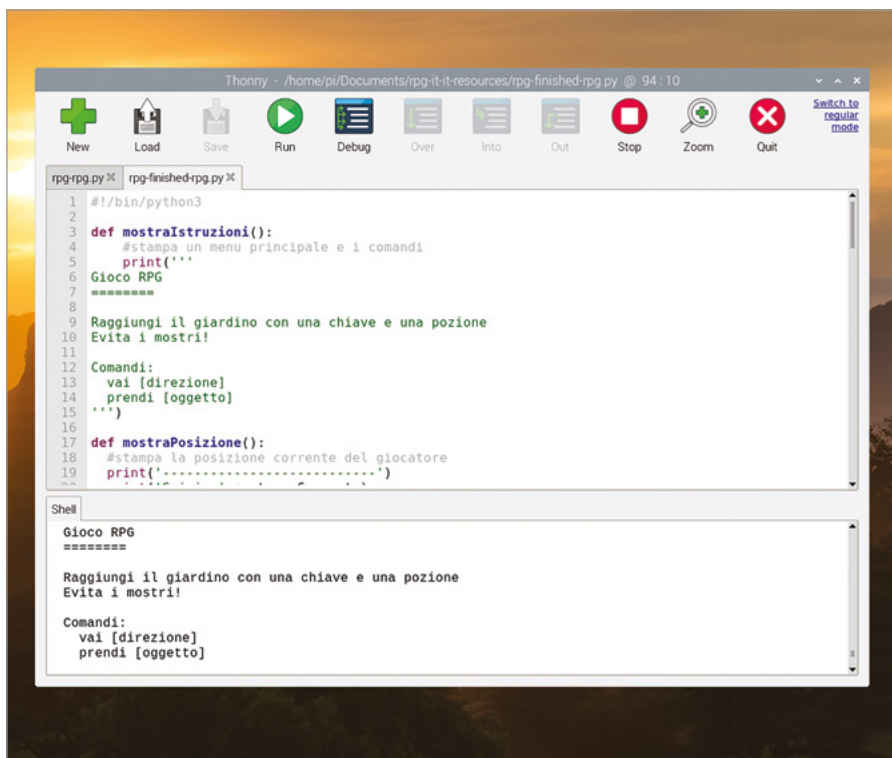
```
# il giocatore vince se raggiunge il giardino con una chiave e  
un pozione  
if stanzaCorrente == 'Giardino' and 'chiave' in inventario  
and 'pozione' in inventario:  
    print('Sei scappato dalla casa... HAI VINTO!')  
    break
```

Fai clic su Run e cerca di finire il gioco raccogliendo la chiave e la pozione prima di andare in giardino. Ricordati di non entrare in cucina, perché è lì che si trova il mostro!

Come ultima modifica per il gioco, aggiungi alcune istruzioni che spiegano al giocatore come completare il gioco. Scorri fino all'inizio del programma, dov'è definita la funzione `mostraIstruzioni()`, e aggiungi quanto segue:

```
Raggiungi il giardino con una chiave e una pozione  
Evita i mostri!
```

Esegui il gioco un'ultima volta e visualizzerai le nuove istruzioni all'inizio (**Figura 5-21**). Complimenti, hai realizzato un gioco interattivo basato su testo con un labirinto!



```

Thonny - /home/pi/Documents/rpg-it-resources/rpg-finished-rpg.py @ 94 : 10
New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode

rpg-rpg.py x rpg-finished-rpg.py x
1 #!/bin/python3
2
3 def mostraIstruzioni():
4     #stampa un menu principale e i comandi
5     print("""
6     Gioco RPG
7     =====
8
9     Raggiungi il giardino con una chiave e una pozione
10    Evita i mostri!
11
12    Comandi:
13    vai [direzione]
14    prendi [oggetto]
15    """)
16
17 def mostraPosizione():
18     #stampa la posizione corrente del giocatore
19     print('.....')
20
Shell
Gioco RPG
=====

Raggiungi il giardino con una chiave e una pozione
Evita i mostri!

Comandi:
vai [direzione]
prendi [oggetto]

```

▲ Figura 5-21: ora il giocatore sa cosa deve fare



SFIDA: SVILUPPA IL GIOCO

Puoi aggiungere altre stanze per far durare il gioco più a lungo? Puoi aggiungere un oggetto per proteggerti dal mostro? Come aggiungere un'arma per uccidere il mostro? Puoi aggiungere stanze sopra e sotto le camere esistenti a cui accedere tramite scale?



Capitolo 6

Physical computing con Scratch e Python

Programmare non vuol dire solo eseguire azioni su uno schermo, ma anche controllare i componenti elettronici collegati ai pin GPIO del Raspberry Pi



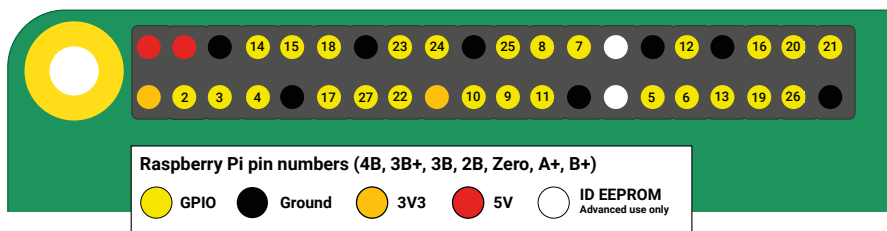
Quando si parla di "programmazione" o "scrittura di codice" generalmente si pensa al software. Tuttavia la programmazione va oltre il software: essa può essere usata anche nel mondo reale per gestire hardware. In questo caso viene chiamata *physical computing*.

Come suggerisce il nome, il *physical computing* (letteralmente: calcolo fisico) consente di controllare tramite programmi elementi del mondo reale, ossia l'hardware anziché il software. Quando imposti il programma della lavatrice, cambi la temperatura del termostato o premi un pulsante al semaforo per attraversare la strada in sicurezza, stai utilizzando il *physical computing*.

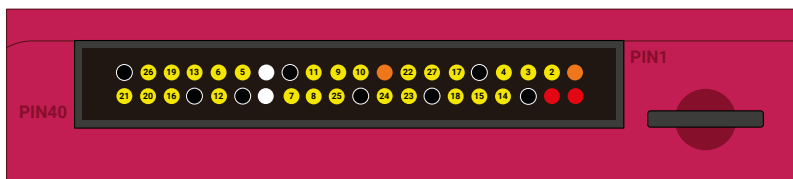
Raspberry Pi è un ottimo strumento per imparare a conoscere il *physical computing* grazie a una funzione chiave: il terminale *general-purpose input/output* (GPIO).

Il terminale GPIO

Il terminale GPIO (general-purpose input/output) si trova sul bordo superiore del circuito di Raspberry Pi o sul retro di Raspberry Pi 400, presenta due lunghe file di pin in metallo e consente di collegare hardware come LED e interruttori a Raspberry Pi per controllare i programmi che crei. I pin possono essere utilizzati sia in ingresso che in uscita.



Il terminale GPIO di Raspberry Pi presenta 40 pin maschi. Alcuni pin possono essere usati nei progetti di physical computing, altri forniscono l'alimentazione, mentre altri ancora sono dedicati alla comunicazione con l'hardware aggiuntivo come Sense HAT (vedi **Capitolo 7**).



Raspberry Pi 400 presenta lo stesso terminale GPIO e pin, ma è capovolto rispetto agli altri modelli Raspberry Pi. Il diagramma rappresenta il terminale GPIO visto dalla parte posteriore di Raspberry Pi 400. Controlla sempre il cablaggio quando colleghi qualcosa al terminale GPIO di Raspberry Pi 400: è facile dimenticarsi, nonostante le etichette "Pin 40" e "Pin 1" sul case.

ESTENSIONI GPIO

È possibile utilizzare il terminale GPIO di Raspberry Pi 400 così com'è, ma potrebbe essere più facile usare un'estensione. Con un'estensione, i pin vengono portati sul lato di Raspberry Pi 400, il che significa che è possibile controllare e modificare il cablaggio senza dover andare sulla parte posteriore.

Le estensioni compatibili includono la gamma Black HAT Hack3r di pimoroni.com e Pi T-Cobbler Plus di adafruit.com.

Se acquisti un'estensione, controlla sempre com'è cablata, alcune, come Pi T-Cobbler Plus, hanno un layout dei pin GPIO diverso. In caso di dubbi, consulta sempre le istruzioni del produttore.

Esistono diverse tipologie di pin, ognuna delle quali ha una funzione particolare:

3V3	Tensione 3,3 volt	Una fonte di alimentazione permanentemente accesa da 3,3 V, la stessa tensione interna di Raspberry Pi
5V	Tensione 5 volt	Una fonte di alimentazione permanentemente accesa da 5 V, la stessa tensione di Raspberry Pi a livello del connettore micro USB
Terra (GND)	Messa a terra da 0 volt	Un collegamento a terra, utilizzato per completare un circuito collegato alla fonte di alimentazione
GPIO XX	Pin di ingresso/uscita per uso generale numero "XX"	I pin GPIO disponibili per i programmi, identificati da un numero da 2 a 27
ID EEPROM	Perni speciali riservati	Pin riservati all'uso con Hardware Attached on Top (HAT) e altri accessori

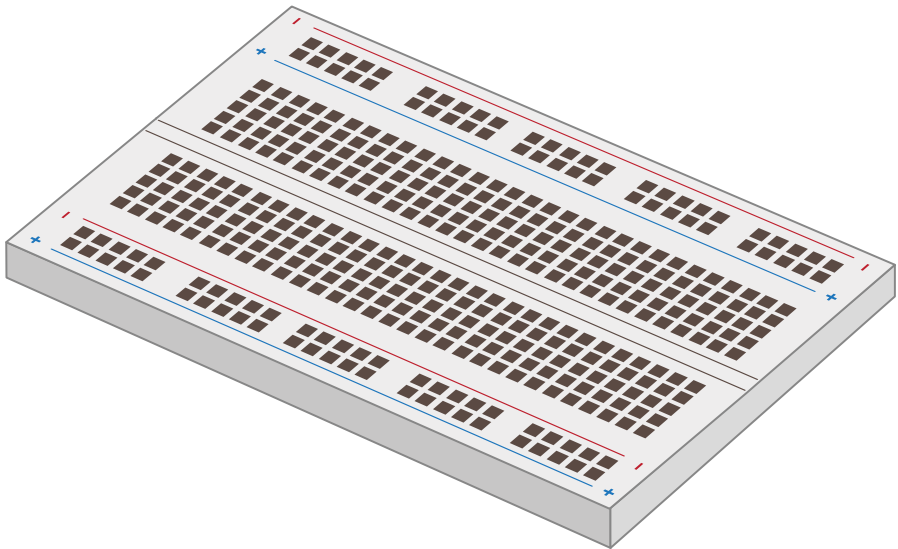
ATTENZIONE

Il terminale GPIO di Raspberry Pi è un modo divertente e sicuro di sperimentare con il physical computing, ma deve essere trattato con cura. Fai attenzione a non piegare i pin quando colleghi e scollegi l'hardware. Non collegare mai due pin direttamente insieme, accidentalmente o deliberatamente, a meno che non sia espressamente indicato nelle istruzioni di un progetto: si tratta di un cosiddetto corto circuito e, a seconda dei pin, potresti danneggiare permanentemente Raspberry Pi.



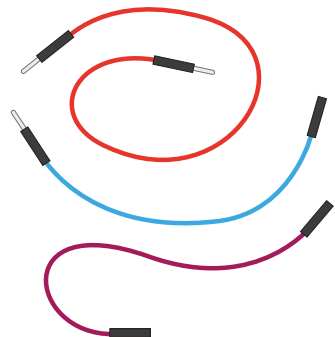
Componenti elettronici

Il terminale GPIO è solo una parte di ciò di cui hai bisogno per iniziare a lavorare con il physical computing; ti serviranno infatti anche i componenti elettrici, ossia i dispositivi che controllerai dal terminale GPIO. Sono migliaia i componenti disponibili, ma la maggior parte dei progetti GPIO vengono realizzati con quelli indicati di seguito.

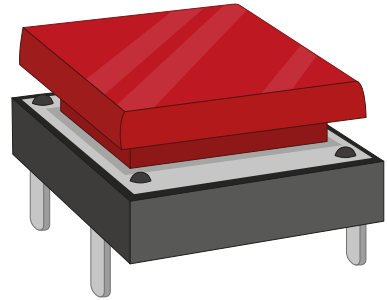


Una *breadboard*, nota anche come *breadboard senza saldature* può rendere i progetti di physical computing molto più semplici. Aniché avere un gruppo di componenti separati da collegare con fili, una breadboard consente di inserire i componenti e collegarli attraverso le guide metalliche nascoste. Molte breadboard includono anche sezioni per distribuire l'alimentazione, rendendo più facile realizzare circuiti. Non è necessario avere una breadboard per iniziare a lavorare con il physical computing, ma sicuramente aiuta.

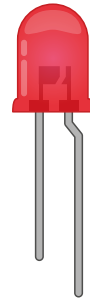
I cavi jumper collegano i componenti a Raspberry Pi o tra loro, se non usi una breadboard. Sono disponibili in tre versioni: maschio-femmina (M2F), per collegare una breadboard ai pin GPIO, femmina-femmina (F2F), che può essere utilizzata per collegare insieme i singoli componenti se non si utilizza una breadboard, e maschio-maschio (M2M) per effettuare collegamenti da una parte all'altra della breadboard. A seconda del progetto, potresti aver bisogno di tutti e tre i tipi di cavi jumper, mentre se stai usando una breadboard, possono bastare i cavi jumper M2F e M2M.



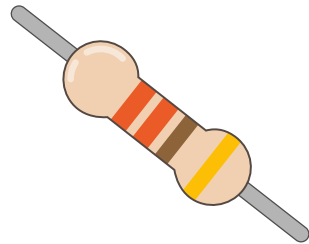
L'interruttore a pulsante, noto anche come *interruttore momentaneo*, è il tipo di interruttore che si usa per controllare una console di gioco. Generalmente disponibile con due o quattro connettori (entrambi compatibili con Raspberry Pi), il pulsante è un dispositivo di input: puoi indicare al programma di attendere che venga premuto per eseguire un compito. Un altro tipo di interruttore comune è l'*interruttore basculante*: mentre il pulsante è attivo solo quando lo si tiene premuto, l'interruttore basculante (come ad esempio l'interruttore della luce) si attiva quando lo si aziona e rimane attivo fino a quando non lo si aziona di nuovo.



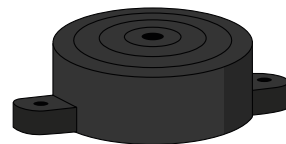
Un *diodo a emissione di luce (LED)* è un *dispositivo di output*: lo si controlla direttamente dal programma. Un LED si illumina quando è acceso e li puoi trovare in tutta la casa: da quelli piccoli che ti fanno sapere quando hai lasciato la lavatrice accesa, a quelli più grandi che illuminano le stanze. I LED possono avere varie forme, colori e dimensioni, ma non tutti possono essere usati con Raspberry Pi: non utilizzare quelli indicati per alimentazione a 5 V o 12 V.



I *resistori* sono componenti che controllano il flusso di *corrente elettrica* e possono presentare diversi valori misurati utilizzando un'unità chiamata *ohm* (Ω). Più elevato è il numero di ohm, maggiore è la resistenza. Nei progetti di physical computing Raspberry Pi il loro uso più comune è quello di impedire ai LED di assorbire troppa corrente e danneggiarsi o danneggiare Raspberry Pi: per questo ti consigliamo resistori da circa 330 Ω , anche se molti fornitori vendono pratiche confezioni contenenti resistori con diversi valori di resistenza per dare maggiore flessibilità.



Il *cicalino piezoelettrico*, di solito chiamato semplicemente cicalino o buzzer, è un altro dispositivo di output. Mentre un LED produce luce, un cicalino produce un rumore, un ronzio. All'interno dell'alloggiamento in plastica del cicalino sono presenti un paio di piastre metalliche: quando si attivano vibrano l'una contro l'altra per produrre il ronzio. Ci sono due tipi di cicalino: *attivo* e *passivo*. Accertati di procurarti un cicalino attivo, in quanto è più semplice da usare.



Altri componenti elettrici comuni sono i motori, che necessitano di una speciale scheda di controllo prima di poter essere collegati a Raspberry Pi, i sensori a infrarossi, che rilevano il movimento, i sensori di temperatura e umidità, che possono essere utilizzati per prevedere le condizioni atmosferiche e infine i fotoresistori (LDR), ossia dispositivi di ingresso che funzionano all'opposto dei LED rilevando la luce.

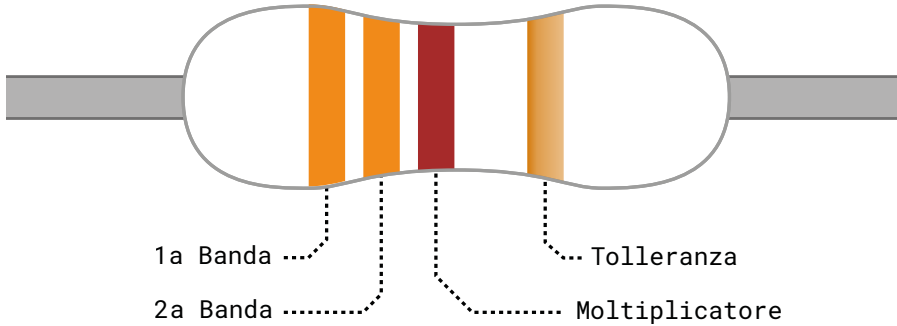
Rivenditori di tutto il mondo forniscono componenti per il physical computing con Raspberry Pi, sia come parti singole che in kit con tutto il necessario per iniziare. Per trovare i venditori, visitate il sito rpf.io/products, cliccate su Raspberry Pi 4 e vedrete un elenco di negozi online partner Raspberry Pi (approved resellers) per il vostro paese o regione.

Per completare i progetti di questo capitolo, dovrete disporre di almeno:

- 3 LED: rosso, verde, giallo o ambra
- 2 interruttori a pulsante
- 1 cicalino attivo
- Cavi jumper maschio-femmina (M2F) e femmina-femmina (F2F)
- Facoltativamente, una breadboard e cavi jumper maschio-maschio (M2M)

Letture dei codici colore del resistore

I resistori sono disponibili in un'ampia gamma di valori: dalle versioni a resistenza zero, che sono sostanzialmente pezzi di filo, alle versioni ad alta resistenza dalle dimensioni di una gamba. Sono pochi però i resistori a presentare i valori numerici stampati: viene invece utilizzato un codice speciale stampato come strisce o bande colorate attorno al corpo del resistore.



	1a/2a Banda	Moltiplicatore	Tolleranza
Nero	0	$\times 10^0$	-
Marrone	1	$\times 10^1$	$\pm 1\%$
Rosso	2	$\times 10^2$	$\pm 2\%$
Arancione	3	$\times 10^3$	-
Giallo	4	$\times 10^4$	-
Verde	5	$\times 10^5$	$\pm 0.5\%$
Blu	6	$\times 10^6$	$\pm 0.25\%$
Viola	7	$\times 10^7$	$\pm 0.1\%$
Grigio	8	$\times 10^8$	$\pm 0.05\%$
Bianco	9	$\times 10^9$	-
Oro	-	$\times 10^{-1}$	$\pm 5\%$
Argento	-	$\times 10^{-2}$	$\pm 10\%$
Nessuno	-	-	$\pm 20\%$

Per leggere il valore di un resistore, posizionalo in modo che il gruppo di bande sia a sinistra e la banda isolata a destra. Partendo dalla prima banda, cerca il colore nella colonna "1^a/2^a banda" della tabella per ottenere la prima e la seconda cifra. In questo esempio sono presenti due bande arancioni, che significano entrambe un valore di "3" per un totale di "33". Se il resistore presenta quattro bande raggruppate invece di tre, annota anche il valore della terza banda (per le resistenze a cinque/sei bande, vedi rpf.io/5-6band).

Passando all'ultima banda nel gruppo (la terza o la quarta) cerca il colore nella colonna "Moltiplicatore". Indica per quale numero devi moltiplicare il numero per ottenere il valore effettivo del resistore. In questo esempio è presente una banda marrone, che significa "x10¹". Può sembrare complesso, ma non è altro che la *notazione scientifica*: "x10¹" significa semplicemente "aggiungere uno zero alla fine del numero". Se fosse blu, per "x10⁶", significherebbe "aggiungere sei zeri alla fine del numero".

33, dalle bande arancioni, più lo zero aggiunto dalla banda marrone dà 330, che è il valore del resistore misurato in ohm. L'ultima banda sulla destra è la *tolleranza* della resistenza, ossia quanto si avvicina al valore nominale. I resistori più economici possono presentare una banda d'argento, che indica che il valore può essere superiore o inferiore del 10% rispetto al valore nominale, oppure essere privi dell'ultima banda, a indicare che il valore può essere superiore o inferiore del 20%, mentre i resistori più costosi presentano una banda grigia, che indica che il valore sarà entro lo 0,05% del valore nominale. Per i progetti amatoriali, la precisione non è così importante: generalmente va bene qualsiasi tolleranza.

Se il valore del resistore supera i 1000 ohm (1000 Ω), è solitamente valutato in kilohm (k Ω), se invece supera un milione di ohm, si tratta di megaohm (M Ω). Un resistore da 2200 Ω verrebbe indicato come 2,2 k Ω , mentre da 2200000 Ω come 2,2 M Ω .



RIESCI A RISOLVERLO?



Quali bande di colore dovrebbe avere un resistore da 100 Ω ? Quali bande di colore dovrebbe avere un resistore da 2,2 M Ω ? Per trovare i resistori più economici che bande di tolleranza devi individuare?

Il tuo primo programma di physical computing: Ciao LED!

Proprio come la stampa di "Hello, World" sullo schermo è un primo passo nell'apprendimento di un linguaggio di programmazione, riuscire a illuminare un LED è la tradizionale introduzione all'apprendimento del physical computing. Per questo progetto avrai bisogno di un LED e di una resistenza da 330 ohm (330 Ω), o con un valore simile, oltre che di cavi jumper femmina-femmina (F2F).



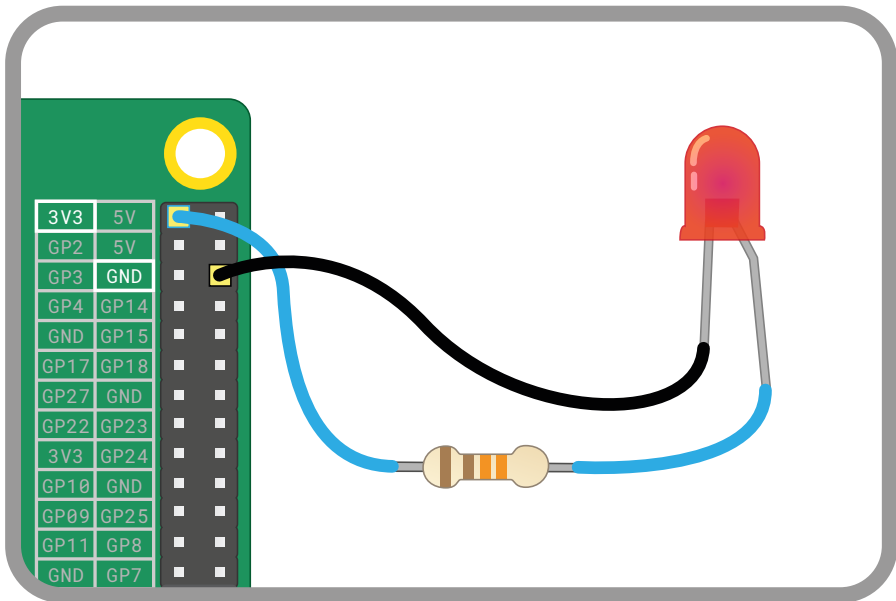
LA RESISTENZA È ESSENZIALE



Il resistore è un componente essenziale in questo circuito: protegge il Raspberry Pi e il LED limitando la quantità di corrente elettrica che il LED può assorbire. Senza di esso, il LED potrebbe assorbire troppa corrente e fulminarsi (o fulminare Raspberry Pi). Quando viene usato in questo modo, il resistore è noto come *resistenza di limitazione della corrente*. Il valore esatto del resistore necessario dipende dal LED che stai usando, tuttavia generalmente 330 Ω è adatto per la maggior parte dei LED più comuni. Più elevato è il valore, più il LED si attenua, più è basso, maggiore è la luminosità.

Non collegare mai un LED a Raspberry Pi senza una resistenza di limitazione di corrente, a meno che il LED non abbia un resistore incorporato di valore appropriato.

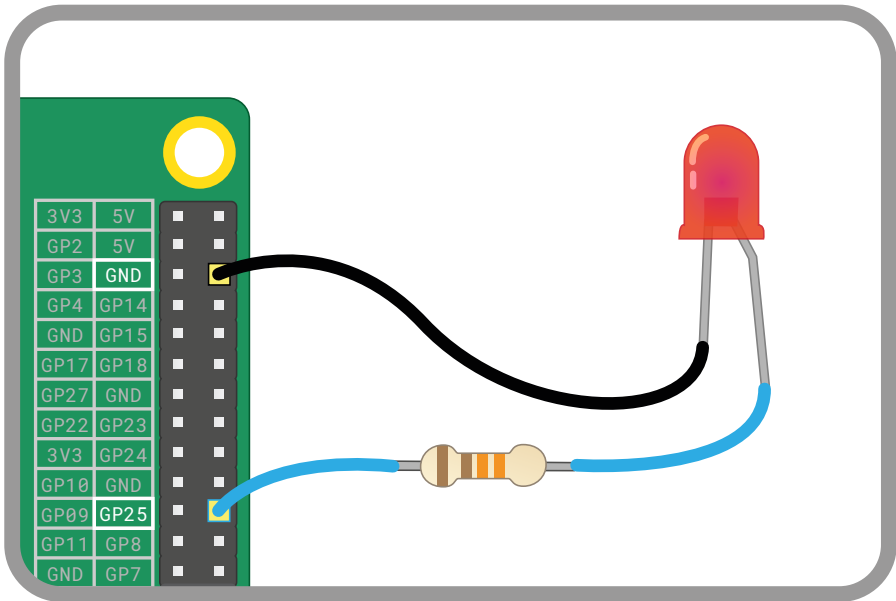
Inizia controllando che il LED funzioni. Gira Raspberry Pi in modo che il terminale GPIO sia posizionato come due strisce verticali sul lato destro. Collega un'estremità del resistore da 330Ω al primo pin da 3,3 V (contrassegnato con 3V3 nella **Figura 6-1**) utilizzando un cavo jumper femmina-femmina, quindi collega l'altra estremità al terminale lungo (positivo o anodo) del LED con un altro cavo jumper femmina-femmina. Prendi un ultimo cavo jumper femmina-femmina e collega il terminale corto (negativo o catodo) del LED al primo polo di messa a terra (contrassegnato come GND nella **Figura 6-1**).



▲ **Figura 6-1:** collega il LED a questi pin senza dimenticare il resistore!

Finché Raspberry Pi è acceso, il LED dovrebbe accendersi. In caso contrario, controlla il circuito: assicurati di non aver usato una resistenza troppo elevata, che tutti i fili siano collegati correttamente e di aver scelto i pin GPIO giusti corrispondenti allo schema. Controlla anche i terminali del LED, in quanto funzionano in un solo modo: con il terminale più lungo collegato al lato positivo del circuito e quello più corto al negativo.

Quando il LED è funzionante è il momento di programmarlo. Scollega il cavo jumper dal pin 3,3 V (contrassegnato con 3V3 nella **Figura 6-2**) e collegalo al pin GPIO 25 (contrassegnato GP25 nella **Figura 6-2**). Il LED si spegne, ma non preoccuparti, è normale.



▲ **Figura 6-2:** scollega il cavo da 3V3 e collegalo al pin GPIO 25

Ora sei pronto per creare un programma Scratch o Python per accendere e spegnere il LED.

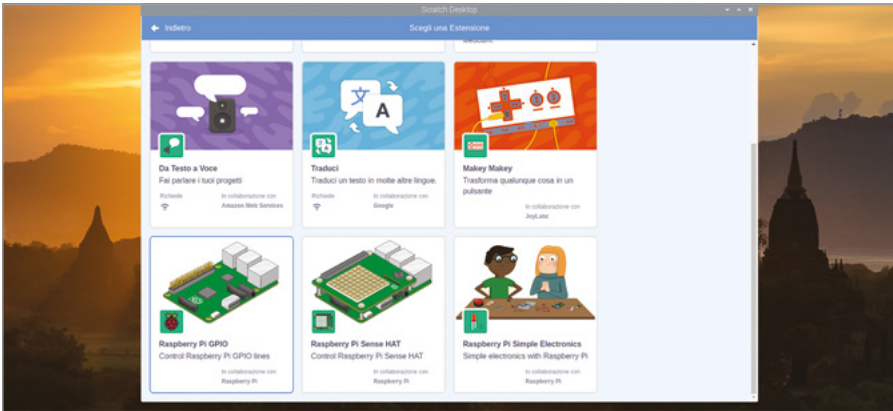


CONOSCENZE DI PROGRAMMAZIONE

Per eseguire i progetti di questo capitolo è necessario che tu sia a tuo agio con l'utilizzo di Scratch 3 e l'ambiente di sviluppo integrato Thonny Python (IDE). Se non l'hai già fatto, leggi il **Capitolo 4, Come programmare con Scratch 3** e il **Capitolo 5, Come programmare con Python** e prova a eseguire i progetti indicati.

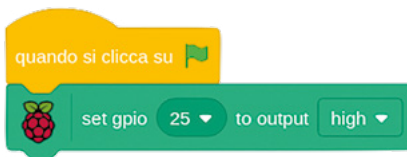
Controllo LED in Scratch

Carica Scratch 3 e fai clic sull'icona **Aggiungi un'estensione**. Scorri verso il basso per trovare l'estensione "Raspberry Pi GPIO" (**Figura 6-3**, sul retro) e fai clic su di essa. Si caricheranno i blocchi necessari per controllare il terminale GPIO di Raspberry Pi da Scratch 3. Visualizzerai nuovi blocchi nella categoria Raspberry Pi GPIO della palette dei blocchi.

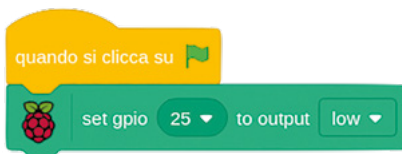


▲ **Figura 6-3:** aggiunta dell'estensione Raspberry Pi GPIO a Scratch 3

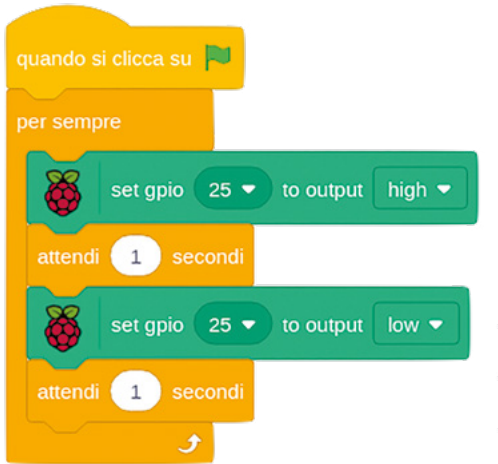
Inizia trascinando un blocco Situazioni **quando si clicca su** nell'area codice, quindi trascina il blocco **set gpio to output high** direttamente sotto di esso. Dovrai scegliere il numero del pin che stai usando: fai clic sulla freccia piccola per aprire la selezione a tendina e fai clic su "25" per dire a Scratch che stai controllando il pin GPIO 25.



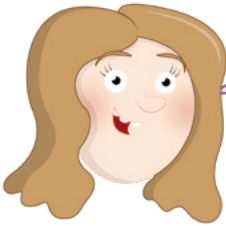
Fai clic sulla bandiera verde per eseguire il programma. Vedrai il LED accendersi: hai programmato il tuo primo progetto di physical computing! Fai clic sull'ottagono rosso per fermare il programma: vedrai che il LED rimane acceso. Questo perché il programma ha indicato a Raspberry Pi solamente di accendere il LED: questo è il significato della parte "output high" del blocco **set gpio 25 to output high**. Fai clic sulla freccia in basso alla fine del blocco, quindi seleziona "low" dall'elenco.



Fai di nuovo clic sulla bandiera verde, questa volta il programma spegnerà il LED. Per rendere le cose più interessanti, aggiungi un blocco di controllo **per sempre** e un paio di blocchi **attendi 1 secondi** per creare unprogramma per accendere e spegnere il LED ogni secondo.



Fai clic sulla bandiera verde e osserva il LED: si accenderà per un secondo, si spegnerà per un secondo, si accenderà di nuovo per un secondo e continuerà così fino a quando non farai clic sull'ottagono rosso per fermarlo. Guarda cosa succede quando fai clic sull'ottagono mentre il LED è in stato di accensione o spegnimento.



SFIDA: PUOI MODIFICARLO?

Come cambieresti il programma per far sì che il LED rimanga acceso più a lungo? Oppure spento più a lungo? Qual è il ritardo più piccolo che puoi usare per vedere comunque il LED accendersi e spegnersi?

Controllo LED in Python

Carica Thonny dalla sezione Programmazione del menu a forma di lampone, quindi fai clic sul pulsante New per avviare un nuovo progetto e salvalo come **Ciao LED**. Per utilizzare i pin GPIO di Python, è necessaria una libreria chiamata GPIO Zero. Per questo progetto occorre solo la parte della libreria per lavorare con i LED. Importa solo questa sezione della libreria digitando quanto segue nell'area shell di Python:

```
from gpiozero import LED
```

Successivamente, è necessario indicare a GPIO Zero a quale pin GPIO è collegato il LED. Digita quanto segue:

```
led = LED(25)
```

Insieme, queste due righe consentono a Python di controllare i LED collegati ai pin GPIO di Raspberry Pi e indicano quale o quali pin (se è presente più di un LED nel circuito) controllare. Per controllare il LED digita quanto segue:

```
led.on()
```

Per spegnerlo di nuovo, digita:

```
led.off()
```

Complimenti, ora sai come controllare i pin GPIO di Raspberry Pi in Python! Prova a digitare nuovamente le due istruzioni. Se il LED è già spento, digitando **led.off()** non succederà nulla, lo stesso vale se il LED è già acceso e si digita **led.on()**.

Per creare un vero e proprio programma, digita quanto segue nell'area dello script:

```
from gpiozero import LED
from time import sleep
```

```
led = LED(25)
```

```
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Questo programma importa la funzione LED dalla libreria **gpiozero** (GPIO Zero) e **sleep** dalla libreria **time**, quindi realizza una ripetizione infinita per accendere il LED per un secondo, spegnerlo per un secondo e ripetere. Fai clic sul pulsante Run per vederlo in azione: il LED inizierà a lampeggiare. Come per il programma Scratch, prendi nota del comportamento quando fai clic sul pulsante Stop mentre il LED è acceso rispetto a quando il LED è spento.



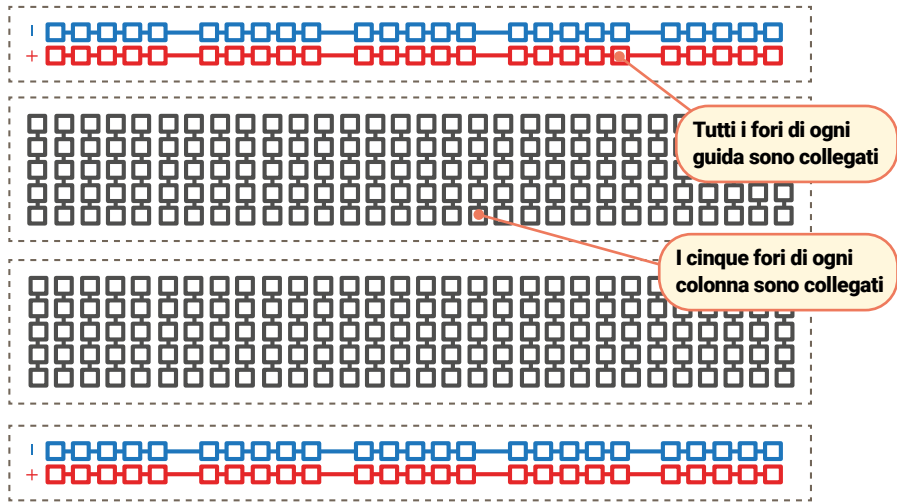
SFIDA: ILLUMINAZIONE PIÙ LUNGA



Come cambieresti il programma per far sì che il LED rimanga acceso più a lungo? Oppure spento più a lungo? Qual è il ritardo più piccolo che puoi usare per vedere comunque il LED accendersi e spegnersi?

Utilizzo di una breadboard

I prossimi progetti di questo capitolo saranno molto più facili da completare se hai a disposizione una breadboard per contenere i componenti e realizzare i collegamenti elettrici.



Una breadboard presenta numerosi fori distanziati di 2,54 mm l'uno dall'altro per corrispondere ai componenti. Sotto a questi fori sono presenti strisce in metallo che si comportano come i cavi jumper usati finora. Queste sono disposte in file in tutta la breadboard. La maggior parte delle breadboard presenta uno spazio al centro che le divide a metà nonché lettere in alto e numeri ai lati per trovare più facilmente determinati fori: A1 è l'angolo in alto a sinistra, B1 è il foro immediatamente a destra, mentre B2 è il foro in basso. A1 è collegato a B1 tramite le strisce metalliche nascoste, tuttavia nessun foro è mai collegato a due fori, a meno che non si aggiunga un cavo jumper.

Le breadboard più grandi presentano anche linee di fori lungo i lati, tipicamente contrassegnate con linee rosse e nere o rosse e blu. Queste linee sono le *linee di alimentazione* e sono progettate per facilitare il cablaggio: è possibile collegare un singolo cavo dal polo di messa terra di Raspberry Pi a una delle linee di alimentazione (tipicamente contrassegnate con una striscia blu o nera e un simbolo meno) per fornire una *messa a terra comune* per molti componenti sulla breadboard e si può fare lo stesso se il circuito necessita di alimentazione a 3,3 V o 5 V.

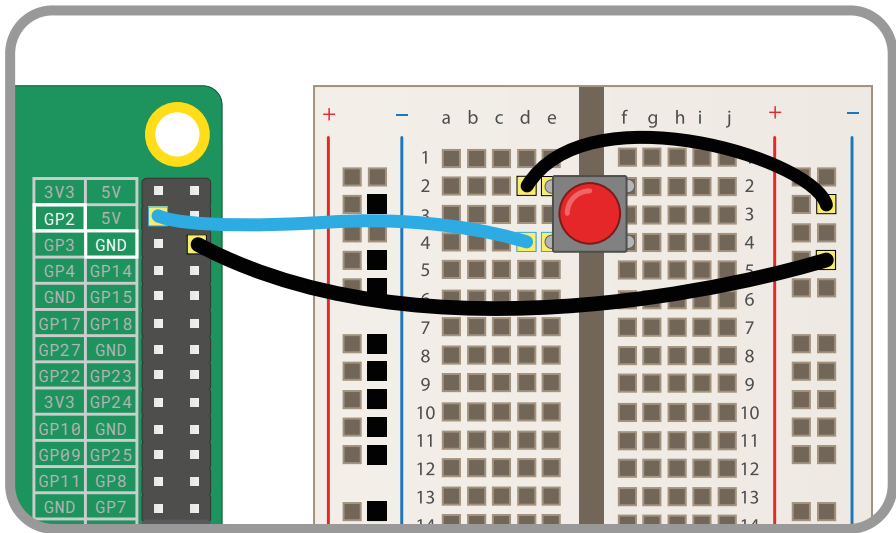
L'aggiunta di componenti elettronici a una breadboard è semplice: basta allineare i conduttori (le parti metalliche sporgenti) con i fori e spingere delicatamente fino a quando il componente non è in posizione. Per le connessioni da effettuare oltre a quelle che la breadboard ti consente, puoi usare cavi jumper maschio-maschio (M2M), mentre per le connessioni dalla breadboard al Raspberry Pi, usa cavi jumper maschio-femmina (M2F).

Non cercare di inserire in un foro più di un connettore o cavo jumper del componente. Ricorda: i fori sono collegati tra loro in colonne, a parte nella suddivisione centrale, quindi un connettore del componente in A1 è collegato elettricamente a qualsiasi cosa tu aggiunga a B1, C1, D1 ed E1.

Passaggi successivi: lettura di un pulsante

Gli output come i LED sono una cosa, ma la parte "input/output" di "GPIO" significa che si possono usare anche pin come ingressi. Per questo progetto avrai bisogno di una breadboard, un cavo jumper maschio-maschio (M2M) e maschio-femmina (M2F), nonché di un interruttore a pulsante. Se non disponi di una breadboard puoi utilizzare cavi jumper femmina-femmina (F2F), ma il pulsante sarà molto più difficile da premere senza rischiare di rompere il circuito.

Inizia aggiungendo il pulsante alla breadboard. Se il pulsante ha solo due connettori, assicurati che siano in diverse file numerate della breadboard, mentre se ne ha quattro, giralo in modo che i lati da cui escono i connettori siano lungo le file della breadboard e i lati piatti senza connettori in alto e in basso. Collega il conduttore di terra della breadboard a un polo di messa a terra su Raspberry Pi (contrassegnato con GND nella **Figura 6-4**) con un cavo jumper maschio-femmina, quindi collega un connettore del pulsante al conduttore di terra con un cavo jumper maschio-maschio. Infine, collega l'altro connettore (quello sullo stesso lato del connettore appena collegato se si utilizza un interruttore a quattro connettori) al pin GPIO 2 (contrassegnato con GP2 nella **Figura 6-4**) di Raspberry Pi con un cavo jumper maschio-femmina.



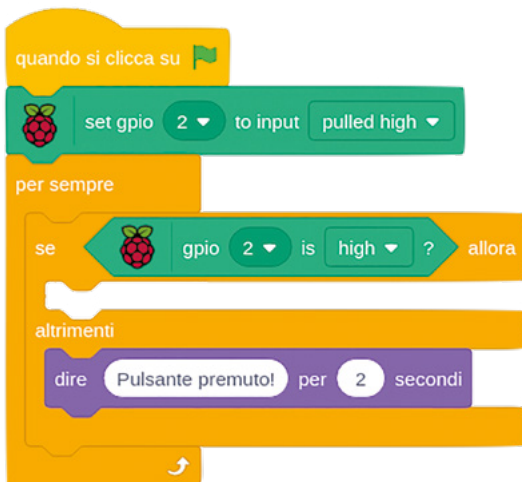
▲ **Figura 6-4:** cablaggio del pulsante ai pin GPIO

Letture di un pulsante in Scratch

Avvia un programma Scratch e trascina un blocco **quando si clicca su** nell'area codice. Collega un blocco **set gpio to input pulled high** e seleziona il numero 2 dal menu a tendina in modo che corrisponda al pin GPIO utilizzato per il pulsante.



Se fai clic sulla bandiera verde non accadrà nulla. Questo perché hai detto a Scratch di usare il pin come input, ma non cosa fare con quell'input. Trascina un blocco **per sempre** alla fine della sequenza, quindi trascina un blocco **se allora altrimenti** al suo interno. Cerca il blocco **gpio is high?**, trascinalo nello spazio a forma di diamante nella parte **se allora** del blocco e usa il menu a tendina per selezionare il numero 2, che indica quale pin GPIO controllare. Trascina un blocco **dire Ciao! per 2 secondi** nella parte **altrimenti** del blocco e modificalo affinché ci sia scritto "Pulsante premuto!". Per il momento lascia la parte "se allora" del blocco vuota.



C'è molto da fare, ma inizia a provare: fai clic sulla bandiera verde, poi premi il pulsante sulla breadboard. Lo sprite dovrebbe dirti che il pulsante è stato premuto: hai letto correttamente un input dal pin GPIO!

Avrai notato che la parte **se gpio 2 is high? allora** del blocco è vuota. Il codice che viene eseguito quando il pulsante viene effettivamente premuto si trova nella parte **altrimenti** del blocco. Sembra contraddittorio: premendo il pulsante non andrà verso l'alto? In realtà, è il contrario: i pin GPIO di Raspberry Pi sono normalmente alzati, o accesi, quando sono impostati come input e premendo il pulsante vengono portati verso il basso.

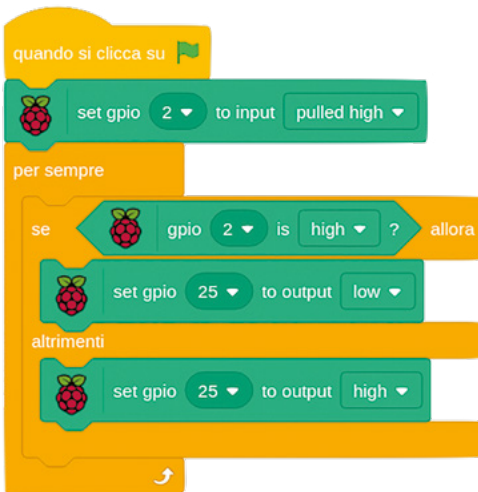
Guarda di nuovo il circuito: il pulsante è collegato al pin 2 GPIO, che fornisce la parte positiva del circuito e al polo di messa a terra. Quando si preme il pulsante, la tensione sul pin GPIO viene

abbassata tramite il polo di messa a terra e il programma Scratch smette di eseguire il codice nel blocco **se gpio 2 is high? allora** ed esegue invece il codice nella parte **altrimenti** del blocco.

Se tutto ciò può sembrare strano, ricorda che il pulsante sul pin GPIO Raspberry Pi viene premuto quando il pulsante va verso il basso e non il contrario.

Per estendere ulteriormente il programma, aggiungi il LED e il resistore nel circuito: ricorda di collegare il resistore al pin GPIO 25 e al terminale lungo del LED, mentre quello corto del LED al conduttore di terra sulla breadboard.

Trascina il blocco **dire Pulsante premuto! per 2 secondi** dall'area del codice alla palette dei blocchi per cancellarlo, quindi sostituiscilo con un blocco **set gpio 25 to output high**, ricordandoti che dovrai cambiare il numero GPIO usando la freccia verso il basso. Aggiungi un blocco **set gpio 25 to output low**, ricordandoti di modificarne i valori, nella parte vuota del blocco **se gpio 2 is high? allora**.



Fai clic sulla bandiera verde e premi il pulsante. Il LED resterà acceso finché tieni premuto il pulsante; rilasciandolo, si spegnerà di nuovo. Complimenti: stai controllando un pin GPIO sulla base di un input di un altro!



SFIDA: FAR RIMANERE ACCESO IL LED

Come cambieresti il programma per far rimanere acceso il LED per qualche secondo, anche dopo aver lasciato il pulsante? Cosa bisogna cambiare per avere il LED acceso mentre non premi il pulsante e spento quando lo premi?



Letture di un pulsante in Python

Fai clic sul pulsante New in Thonny per avviare un nuovo progetto e salvalo come **Input pulsante**. L'uso di un pin GPIO come ingresso per un pulsante è molto simile all'uso di un pin come uscita per un LED, ma è necessario importare una parte diversa della libreria GPIO Zero. Digita quanto segue nell'area dello script:

```
from gpiozero import Button
button = Button(2)
```

Per far eseguire il codice quando il pulsante viene premuto, GPIO Zero fornisce l'opzione `wait_for_press`. Digita quanto segue:

```
button.wait_for_press()
print("Mi hai premuto!")
```

Fai clic sul pulsante Run, quindi premi l'interruttore a pulsante. Il messaggio verrà stampato in Python shell in fondo alla finestra Thonny: hai letto correttamente un input dal pin GPIO! Se vuoi provare di nuovo il programma, dovrai fare clic di nuovo sul pulsante Run. Poiché non c'è una ripetizione, il programma chiude non appena ha finito di stampare il messaggio in shell.

Per estendere ulteriormente il programma, aggiungi il LED e il resistore nel circuito se non l'hai già fatto: ricorda di collegare il resistore al pin GPIO 25 e al terminale lungo del LED, mentre quello corto del LED al conduttore di terra sulla breadboard.

Per controllare un LED, oltre a leggere un pulsante, è necessario importare sia l'opzione **Buttons** sia **LED** della libreria GPIO Zero. Avrai anche bisogno della funzione `sleep` della biblioteca `time`. Torna all'inizio del programma e inserisci quanto segue come prime due righe:

```
from gpiozero import LED
from time import sleep
```

Sotto alla riga `button = Button(2)`, digita:

```
led = LED(25)
```

Cancella la riga `print("Mi hai premuto!")` e sostituiscila con:

```
led.on()
sleep(3)
led.off()
```

Il programma finito dovrebbe avere questa struttura:

```
from gpiozero import LED
from time import sleep
from gpiozero import Button

button = Button(2)
led = LED(25)
button.wait_for_press()
led.on()
sleep(3)
led.off()
```

Fai clic sul pulsante Run, quindi premi l'interruttore a pulsante, il LED si accenderà per tre secondi, quindi si spegnerà e il programma verrà chiuso. Complimenti: ora sai come controllare un LED utilizzando un input del pulsante in Python!



SFIDA: AGGIUNGERE UNA RIPETIZIONE



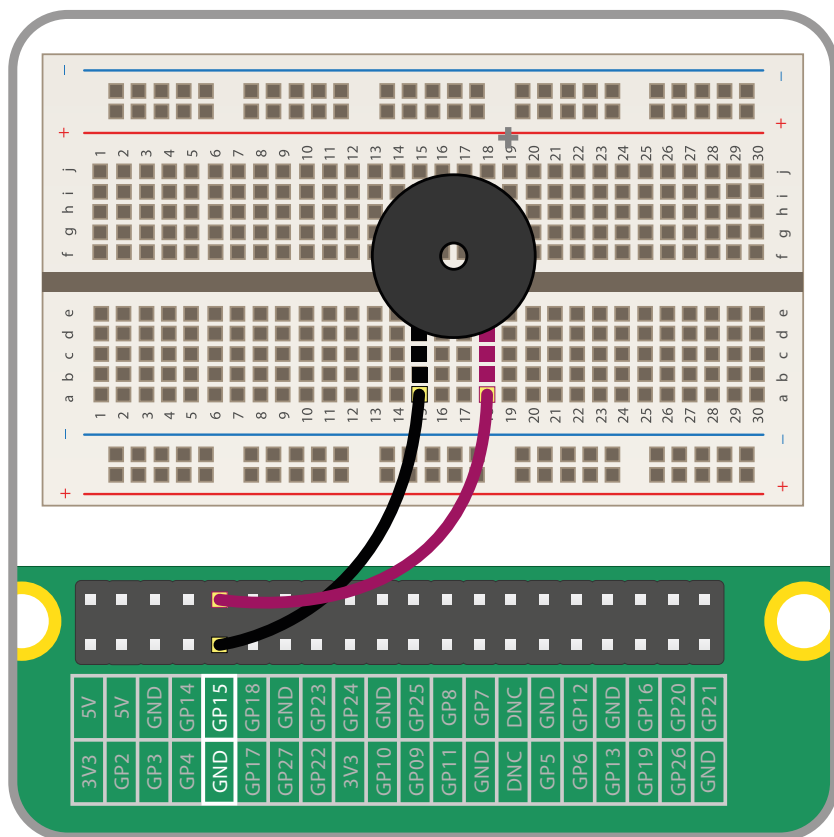
Come si fa ad aggiungere una ripetizione per far ripetere il programma invece di terminare dopo la pressione di un tasto? Cosa bisogna cambiare per avere il LED acceso mentre non premi il pulsante e spento quando lo premi?

Facciamo un po' di rumore: controllare un cicalino

I LED sono un ottimo dispositivo di output, ma non sono molto utili se si guarda in un'altra direzione. La soluzione sono i cicalini, che producono un rumore udibile in qualsiasi punto della stanza. Per questo progetto avrai bisogno di una breadboard, di cavi jumper maschio-femmina (M2F) e di un cicalino attivo. Se non disponi di una breadboard puoi collegare il cicalino utilizzando cavi femmina-femmina (F2F).

Un cicalino attivo può essere trattato esattamente come un LED, in termini di circuiteria e programmazione. Ripeti la procedura del circuito che hai realizzato per il LED, ma sostituisci il LED con il cicalino attivo e ometti il resistore, poiché il cicalino avrà bisogno di più corrente per funzionare. Collega un terminale del cicalino al pin GPIO 15 (contrassegnato GP15 nella **Figura 6-5**) e l'altro al polo di terra (contrassegnato GND nello schema) utilizzando la breadboard e i cavi jumper maschio-femmina.

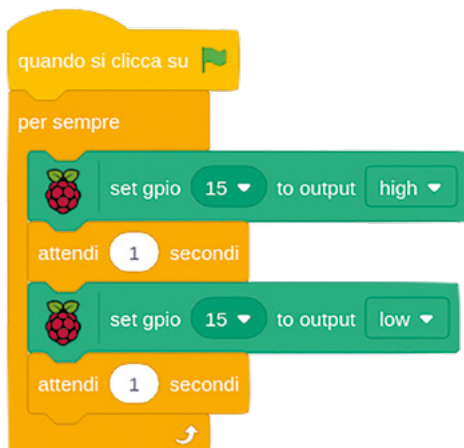
Se il cicalino ha tre terminali, assicurati che il terminale contrassegnato con il simbolo meno (-) sia collegato al polo di terra e quello contrassegnato con "S" o "SIGNAL" a GPIO 15, quindi collega il terminale rimanente, generalmente quello centrale, al pin da 3,3 V (contrassegnato con 3V3).



▲ **Figura 6-5:** collegamento di un cicalino ai pin GPIO

Controllo di un cicalino in Scratch

Ricrea o, se lo hai salvato prima di creare il progetto del pulsante, carica lo stesso programma usato per far lampeggiare il LED. Utilizza il menu a tendina nei blocchi **set gpio to output high** per selezionare il numero 15, in modo che Scratch controlli il pin GPIO corretto.



Fai clic sulla bandiera verde e il cicalino inizierà a produrre un ronzio: un secondo sì e uno no. Se senti il cicalino fare clic solo una volta al secondo, stai utilizzando un cicalino passivo anziché attivo. Il cicalino attivo genera un segnale che cambia rapidamente, ossia un'oscillazione, per far vibrare le piastre metalliche, mentre un cicalino passivo necessita di un segnale oscillante. Quando lo si accende semplicemente con Scratch, le piastre si muovono una sola volta e si fermano, producendo il "clic" fino alla prossima accensione o allo spegnimento del pin da parte del programma.

Fai clic sull'ottagono rosso per fermare il cicalino, ma assicurati di farlo quando non emette alcun suono, altrimenti il cicalino continuerà a suonare fino a quando non eseguirai di nuovo il programma!



SFIDA: CAMBIARE IL SUONO



Come si può cambiare il programma per far suonare il cicalino per un tempo più breve? Si può costruire un circuito in modo che il cicalino sia controllato da un pulsante?

Controllo di un cicalino in Python

Il controllo di un cicalino attivo attraverso la libreria GPIO Zero è quasi identico al controllo di un LED, in quanto ha stati di accensione e spegnimento. Hai bisogno di una funzione diversa, però: **buzzer**. Crea un nuovo progetto in Thonny e salvalo come **Cicalino**.

```
from gpiozero import Buzzer
from time import sleep
```

Come per i LED, GPIO Zero ha bisogno di sapere a quale pin è collegato il cicalino per poterlo controllare. Digita quanto segue:

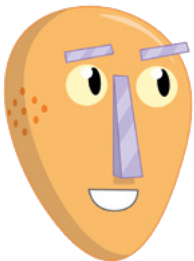
```
buzzer = Buzzer(15)
```

Da qui, il programma è quasi identico a quello che hai scritto per controllare il LED: l'unica differenza (a parte un diverso numero di pin GPIO) è che stai usando **buzzer** al posto di **led**. Digita quanto segue:

```
while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

Fai clic sul pulsante Run e il cicalino inizierà a produrre un ronzio: un secondo sì e uno no. Se utilizzi un cicalino passivo anziché attivo, si sentirà solo un breve clic ogni secondo anziché un suono continuo: questo perché il cicalino passivo è privo dell'*oscillatore* che genera il segnale che cambia rapidamente e fa vibrare le piastre all'interno del cicalino.

Fai clic sul pulsante Stop per uscire dal programma, ma assicurati che il cicalino non emetta alcun suono in quel momento, altrimenti continuerà a suonare fino a quando non eseguirai di nuovo il programma.



SFIDA: UN SUONO MIGLIORE

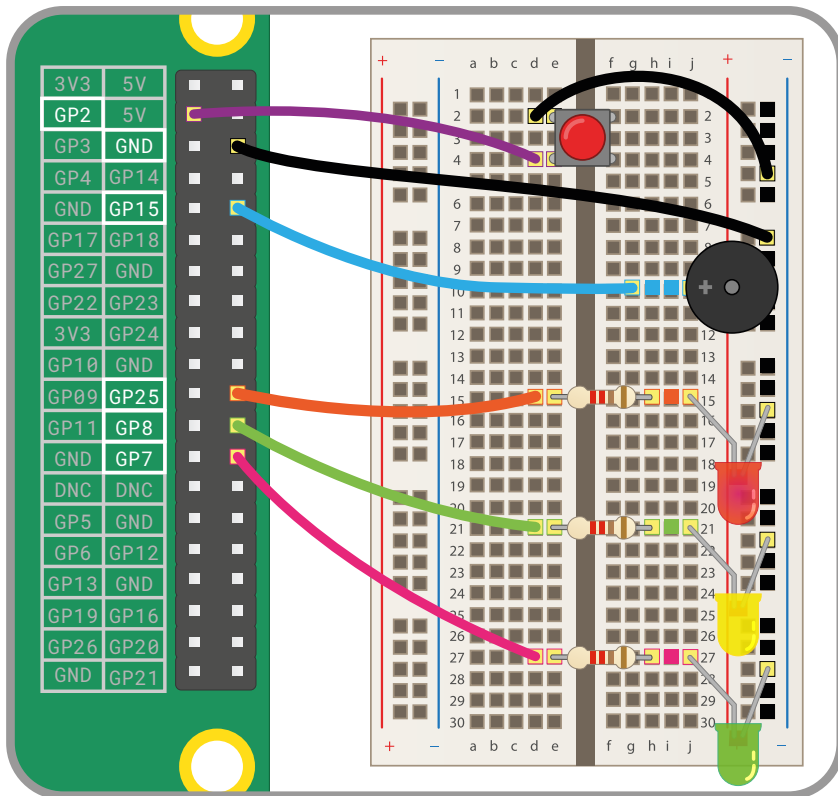
Come si può cambiare il programma per far suonare il cicalino per un tempo più breve? Si può costruire un circuito in modo che il cicalino sia controllato da un pulsante?



Progetto Scratch: semafori

Ora che sai come usare pulsanti, cicalini e LED come input e output, sei pronto per costruire un esempio di calcolo fisico reale, ossia dei semafori, completi di pulsante da premere per attraversare la strada. Per questo progetto avrai bisogno di una breadboard, un LED rosso, giallo e verde, tre resistori da 330 Ω , un cicalino, un interruttore a pulsante e infine una selezione di cavi jumper maschio-maschio (M2M) e maschio-femmina (M2F).

Inizia realizzando il circuito (Figura 6-6), collegando il cicalino al pin GPIO 15 (contrassegnato GP15 nella Figura 6-6), il LED rosso al pin GPIO 25 (contrassegnato GP25), il LED giallo al GPIO 8 (GP8), il LED verde al GPIO 7 (GP7) e l'interruttore al GPIO 2 (GP2). Ricordati di collegare i resistori da 330 Ω tra i pin GPIO e i terminali lunghi dei LED e di collegare i secondi terminali di tutti i componenti al conduttore di terra della breadboard. Infine, collega il conduttore di terra a un polo di messa a terra (contrassegnato GND) su Raspberry Pi per completare il circuito.



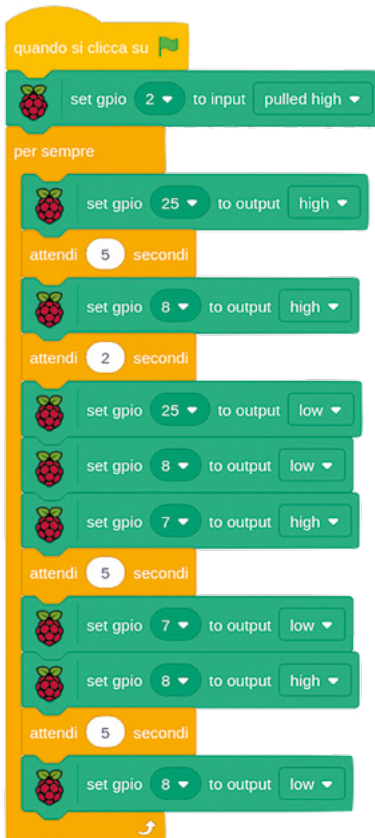
▲ Figura 6-6: schema di cablaggio per il progetto Semaforo

Crea un nuovo progetto Scratch 3 e trascina un blocco [quando si clicca su](#) nell'area codice. Successivamente, dovrai indicare a Scratch che il pin GPIO 2 collegato

all'interruttore a pulsante nel circuito è un ingresso anziché un'uscita: trascina un blocco **set gpio to input pulled high** dalla categoria Raspberry Pi GPIO della palette dei blocchi sotto il blocco **quando si clicca su**. Fai clic sulla freccia verso il basso accanto a "0" e seleziona il numero 2 dall'elenco a discesa.



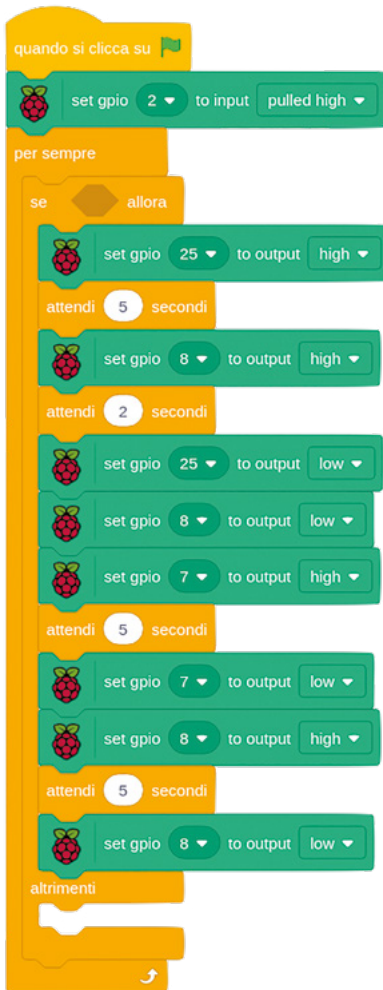
Successivamente, è necessario creare la sequenza del semaforo. Trascina un blocco **per sempre** nel programma, quindi aggiungi blocchi per accendere e spegnere i LED del semaforo secondo uno schema. Ricordati quali pin GPIO hanno ciascun componente collegato: con il pin 25 si usa il LED rosso, con il pin 8 il LED giallo e con il 7 il LED verde.



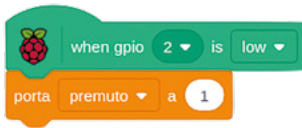
Fai clic sulla bandiera verde e osserva i LED: prima si accenderà il rosso, poi sia il rosso sia il giallo, poi il verde, poi il giallo e infine la sequenza si ripete con la luce rossa. Questo

schema corrisponde a quello utilizzato dai semafori nel Regno Unito: se lo desideri, è possibile modificare la sequenza in modo che corrisponda agli schemi di altri paesi.

Per simulare un attraversamento pedonale, è necessario che il programma attenda che venga premuto il pulsante. Fai clic sull'ottagono rosso per fermare il programma se è in funzione. Trascina un blocco **se allora altrimenti** nell'area script e collegalo in modo che si trovi direttamente sotto il blocco **per sempre**, con la sequenza del semaforo nella sezione "se allora". Lascia vuoto lo spazio a forma di diamante per ora.

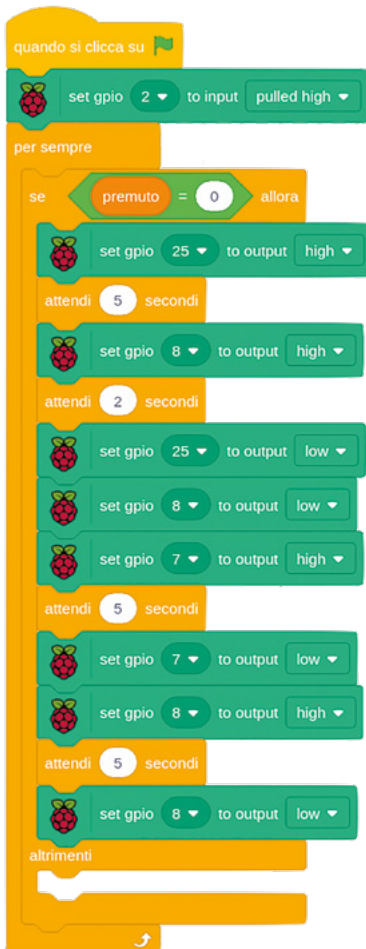


Un vero e proprio attraversamento pedonale non attiva il rosso non appena si preme il pulsante, ma attende il semaforo rosso successivo nella sequenza. Per inserirlo nel programma, trascina un blocco **when gpio is low** nell'area del codice e seleziona "2" dall'elenco a discesa. Trascina un blocco **porta premuto a 1** di seguito.



Questi blocchi attendono che il pulsante venga premuto, quindi impostano la variabile "premuto" su 1. Impostando la variabile in questo modo è possibile memorizzare il fatto che il pulsante è stato premuto, anche se non lo utilizzerai subito.

Torna alla tua sequenza di blocchi originale e cerca il blocco **se allora**. Trascina un blocco Operatori **=** nello spazio vuoto a forma di diamante del blocco **se allora**, quindi trascina un blocco reporter **premuto** nel primo spazio vuoto. Digita "0" sopra a "50" sul lato destro del blocco.



Fai clic sulla bandiera verde e guarda la sequenza di luci del semaforo. Premi il pulsante: all'inizio sembrerà che non stia succedendo nulla, ma una volta che la sequenza è giunta al termine (con il solo LED giallo acceso) le luci del semaforo si spegneranno e rimarranno spente, grazie alla variabile "premuto".

Non resta che fare in modo che il pulsante di attraversamento pedonale non spenga solo le luci. Nella sequenza di blocchi principale, cerca il blocco **altrimenti** e trascinavi dentro un blocco **set gpio 25 to output high** ricordandoti di cambiare il numero di pin GPIO predefinito in modo che corrisponda al pin a cui è collegato il LED rosso.

Sotto di esso, sempre nel blocco **altrimenti**, crea uno schema per il cicalino: trascina un blocco **ripeti 10 volte**, poi inseriscivi i blocchi **set gpio 15 to output high**, **attendi 0.2 secondi**, **set gpio 15 to output low** e **attendi 0.2 secondi**, cambiando i valori del pin GPIO per farli corrispondere al pin del componente cicalino.

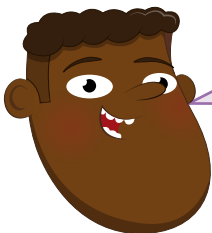
Infine, sotto al blocco **ripeti 10 volte**, ma sempre nel blocco **altrimenti**, aggiungi un blocco **set gpio 25 to output low** e un blocco **porta premuto a 0**; l'ultimo blocco reimposta la variabile che memorizza la pressione del pulsante, in modo che la sequenza del cicalino non si ripeta all'infinito.

Fai clic sulla bandiera verde, poi premi l'interruttore sulla breadboard. Al termine della sequenza, vedrai la luce rossa accendersi e sentirai il cicalino suonare per far sapere ai pedoni che è sicuro attraversare. Dopo un paio di secondi il cicalino si ferma e la sequenza del semaforo ricomincia e continua fino alla successiva pressione del tasto.

Complimenti: hai programmato il tuo semaforo perfettamente funzionale, completo di attraversamento pedonale!

```

quando si clicca su
  set gpio 2 to input pulled high
  per sempre
    se premuto = 0 allora
      set gpio 25 to output high
      attendi 5 secondi
      set gpio 8 to output high
      attendi 2 secondi
      set gpio 25 to output low
      set gpio 8 to output low
      set gpio 7 to output high
      attendi 5 secondi
      set gpio 7 to output low
      set gpio 8 to output high
      attendi 5 secondi
      set gpio 8 to output low
    altrimenti
      set gpio 25 to output high
      ripeti 10 volte
        set gpio 15 to output high
        attendi 0.2 secondi
        set gpio 15 to output low
        attendi 0.2 secondi
      set gpio 25 to output low
      porta premuto a 0
  porta premuto a 1
  
```



SFIDA: PUOI MIGLIORARE?

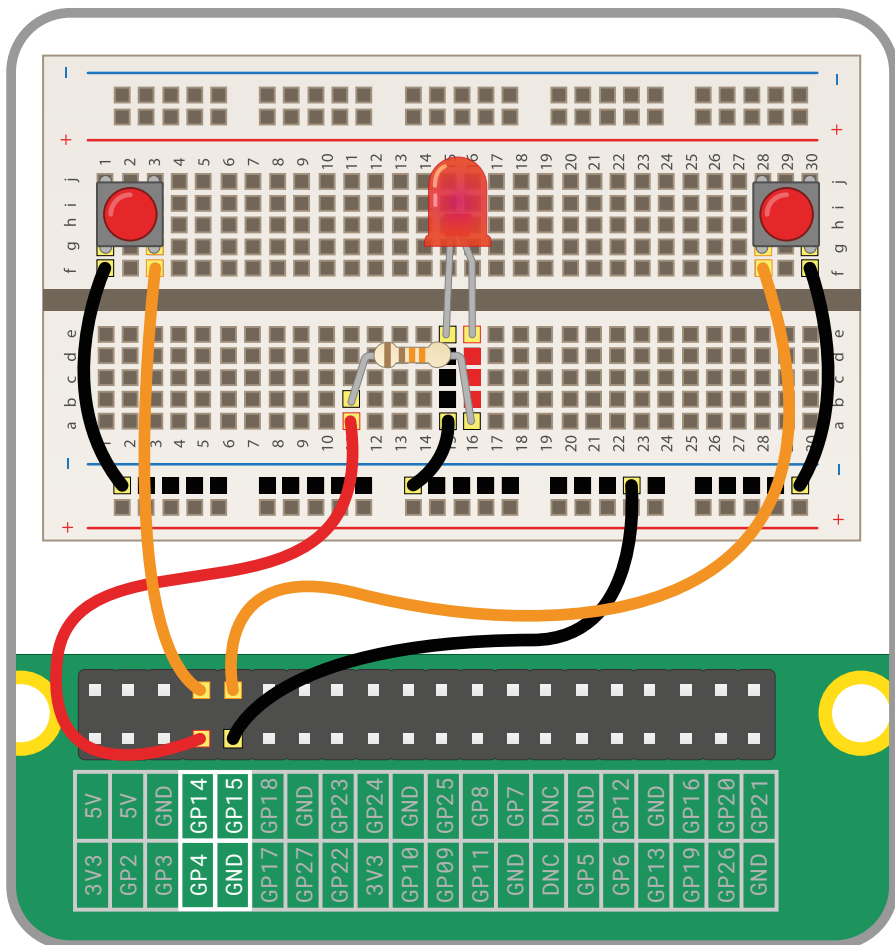


Puoi cambiare il programma per dare al pedone più tempo per attraversare? Riesci a trovare informazioni sulle sequenze semaforiche di altri paesi e riprogrammare le luci in modo che corrispondano? Come potresti rendere i LED meno luminosi?

Progetto Python: gioco di reazione

Ora che sai come usare pulsanti e LED come input e output, sei pronto per costruire un esempio di calcolo fisico reale, ossia un gioco per mettere alla prova la tua velocità di reazione pensato per due giocatori. Per questo progetto, avrai bisogno di una breadboard, un LED, un resistore da 330 Ω , due interruttori a pulsante, alcuni cavi jumper maschio-femmina (M2F) e maschio-maschio (M2M).

Inizia costruendo il circuito (**Figura 6-7**): collega il primo interruttore sul lato sinistro della breadboard al pin GPIO 14 (contrassegnato GP14 nella **Figura 6-7**), il secondo interruttore sul lato destro della breadboard al pin GPIO 15 (contrassegnato come GP15), il terminale piú lungo del LED al resistore da 330 Ω che poi si collega al pin GPIO 4 (contrassegnato GP4) di Raspberry Pi e il secondo terminale a tutti i componenti al conduttore di terra della BREADBOARD. Infine, collega il conduttore di terra al polo di messa a terra di Raspberry Pi (contrassegnato GND).



▲ **Figura 6-7:** schema di cablaggio per il gioco per la velocità di reazione

Crea un nuovo progetto in Thonny e salvalo come **Gioco per la velocità di reazione**. Utilizzerai le funzioni **LED** e **button** della libreria GPIO Zero e la funzione **sleep** dalla libreria `time`. Invece di importare le due funzioni GPIO Zero su due righe separate puoi risparmiare tempo e importarle insieme usando il simbolo di una virgola (,) per separarle. Digita quanto segue nell'area dello script:

```
from gpiozero import LED, Button
from time import sleep
```

Come prima, dovrai indicare a GPIO Zero a quali pin sono collegati i due pulsanti e il LED. Digita quanto segue:

```
led = LED(4)
right_button = Button(15)
left_button = Button(14)
```

Ora aggiungi le istruzioni per accendere e spegnere il LED, in modo da poter controllare che funzioni correttamente:

```
led.on()
sleep(5)
led.off()
```

Fai clic sul pulsante Run: il LED si accende per cinque secondi, poi si spegne e il programma si chiude. Ai fini di un gioco di reazione, però, se il LED si spegne sempre dopo esattamente 5 secondi è un po' prevedibile. Aggiungi questo comando sotto la riga **from time import sleep**:

```
from random import uniform
```

La libreria `random`, come suggerisce il nome, consente di generare numeri casuali (qui con una distribuzione uniforme, vedi rpf.io/uniform). Trova la riga **sleep(5)** e modificala come segue:

```
sleep(uniform(5, 10))
```

Fai nuovamente clic sul pulsante Run: questa volta il LED rimarrà acceso per un numero di secondi casuale compreso tra 5 e 10. Conta per vedere quanto tempo ci vuole perché il LED si spenga, poi fai clic un paio di volte sul pulsante Run: vedrai che il tempo è diverso per ogni esecuzione, rendendo il programma meno prevedibile.

Per trasformare i pulsanti in trigger per ogni giocatore, è necessario aggiungere una funzione. Al termine del programma digita quanto segue:

```
def pressed(button):
    print(str(button.pin.number) + " ha vinto la partita")
```

Ricorda che Python usa il rientro per sapere quali righe fanno parte della funzione: Thonny farà rientrare automaticamente la seconda riga. Infine aggiungi le seguenti due righe per individuare quando i giocatori premono i pulsanti, ricordati che non devono essere rientranti, altrimenti Python le tratterà come parte della funzione.

```
right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Esegui il programma, questa volta prova a premere uno dei due pulsanti non appena il LED si spegne. Vedrai un messaggio stampato in Python shell, in fondo alla finestra Thonny, per il primo pulsante premuto. Purtroppo visualizzerai messaggi ogni volta che uno dei due pulsanti viene premuto, inoltre per il pulsante viene usato il numero di pin anziché un nome intuitivo.

Per risolvere il problema, chiedi ai giocatori i loro nomi. Sotto la riga **from random import uniform**, digita quanto segue:

```
left_name = input("Il nome del giocatore a sinistra è ")
right_name = input("Il nome del giocatore a destra è ")
```

Torna alla funzione e sostituisci la riga **print(str(button.pin.number) + " ha vinto la partita")** con:

```
if button.pin.number == 14:
    print(left_name + "ha vinto la partita")
else:
    print(right_name + " ha vinto la partita")
```

Fai clic sul pulsante Run, quindi digita i nomi di entrambi i giocatori nell'area shell di Python. Quando premerete il pulsante, ricordandovi di farlo il più velocemente possibile dopo lo spegnimento del LED, vedrai che al posto del numero del pin viene stampato il nome del giocatore.

Per fare in modo che non vengano segnalate come vincenti tutte le pressioni dei pulsanti, è necessario aggiungere la funzione **exit** dalla libreria **sys** (abbreviazione di **sistema**). Sotto l'ultima riga **import**, digita quanto segue:

```
from os import _exit
```

Ai termine della funzione, sotto la riga **print(right_name + "ha vinto la partita")**, digita quanto segue:

```
    _exit(0)
```

In questo caso il rientro è importante: `_exit(0)` dovrebbe essere preceduto da quattro spazi, allineato con `else`: due righe sopra di esso e `if` due righe sopra. Questa istruzione indica a Python di fermare il programma dopo la prima pressione del pulsante, il che significa che il giocatore che ha premuto il pulsante per secondo non ha vinto!

Il programma finito dovrebbe avere questa struttura:

```
from gpiozero import LED, Button
from time import sleep
from random import uniform
from os import _exit

left_name = input("Il nome del giocatore a sinistra è ")
right_name = input ("Il nome del giocatore a destra è ")
led = LED(4)
right_button = Button(15)
left_button = Button(14)

led.on()
sleep(uniform(5, 10))
led.off()

def pressed(button):
    if button.pin.number == 14:
        print(left_name + " ha vinto la partita")
    else:
        print(right_name + " ha vinto la partita")
    _exit(0)

right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Fai clic sul pulsante Run, inserisci i nomi dei giocatori, aspetta che il LED si spenga e vedrai il nome del giocatore vincente. Vedrai anche un messaggio da Python: **Backend terminated or disconnected . Use 'Stop/Restart' to restart ...**Questo significa semplicemente che Python ha ricevuto il comando `_exit(0)` e fermato il programma, ma sarà necessario fare clic sull'icona Stop per uscire completamente e preparare il programma per un altro round (Figura 6-8).

The screenshot shows the Thonny IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The main editor window displays the following Python code:

```

9 right_button = Button(15)
10 left_button = Button(14)
11
12 led.on()
13 sleep(uniform(5, 10))
14 led.off()
15
16 def pressed(button):
17     if button.pin_number == 14:
18         print(left_name + " won the game")
19     else:
20         print(right_name + " won the game")
21     _exit(0)
22
23 right_button.when_pressed = pressed
24 left_button.when_pressed = pressed

```

Below the code editor is a Shell window showing the execution output:

```

>>> %Run 'Reaction Game.py'
Left player name is Gareth
Right player name is Eben
>>> Gareth won the game

Backend terminated or disconnected. Use 'Stop/Restart' to restart ...

```

▲ **Figura 6-8:** Una volta deciso il vincitore, è necessario interrompere il programma

Complimenti, hai creato il tuo gioco fisico!



SFIDA: MIGLIORA IL GIOCO

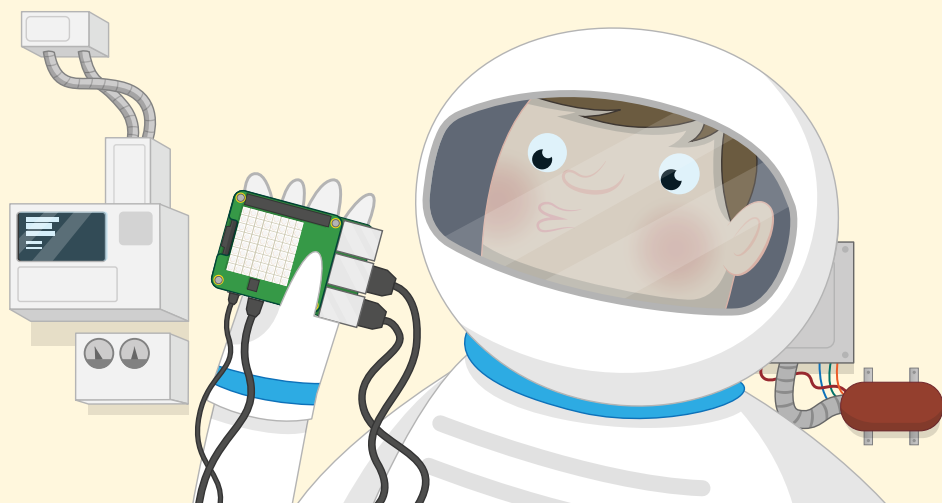


Riesci ad aggiungere una ripetizione, in modo che il gioco funzioni continuamente? Ricordati di rimuovere l'istruzione `_exit(0)` prima! Riesci ad aggiungere un conteggio per il punteggio in modo da poter vedere chi sta vincendo nei vari round? Che ne dici di un timer, in modo da vedere quanto tempo ci è voluto per reagire allo spegnimento della luce?

Capitolo 7

Physical computing con Sense HAT

Sense HAT è una scheda aggiuntiva multifunzione Raspberry Pi dotata di sensori e display a matrice LED, la stessa usata nella Stazione spaziale internazionale



Raspberry Pi supporta un tipo speciale di scheda aggiuntiva chiamata *HAT (Hardware Attached on Top)*. Con i componenti HAT è possibile aggiungere a Raspberry Pi microfoni, luci, relè elettronici, schermi e molto altro ancora, ma un HAT in particolare è molto speciale: Sense HAT.

Sense HAT è stato progettato appositamente per la missione spaziale Astro Pi. Con questo progetto, nato dalla collaborazione tra Raspberry Pi Foundation, UK Space Agency e l'Agenzia spaziale europea, le schede e i Sense HAT di Raspberry Pi sono stati portati dal veicolo spaziale Cygnus Orbital Science sulla Stazione spaziale internazionale. Da quando sono stati portati in orbita, i Sense HAT (soprannominati Ed e Izzy dagli astronauti) sono stati usati per eseguire codici ed esperimenti scientifici con il contributo degli studenti di tutta Europa.

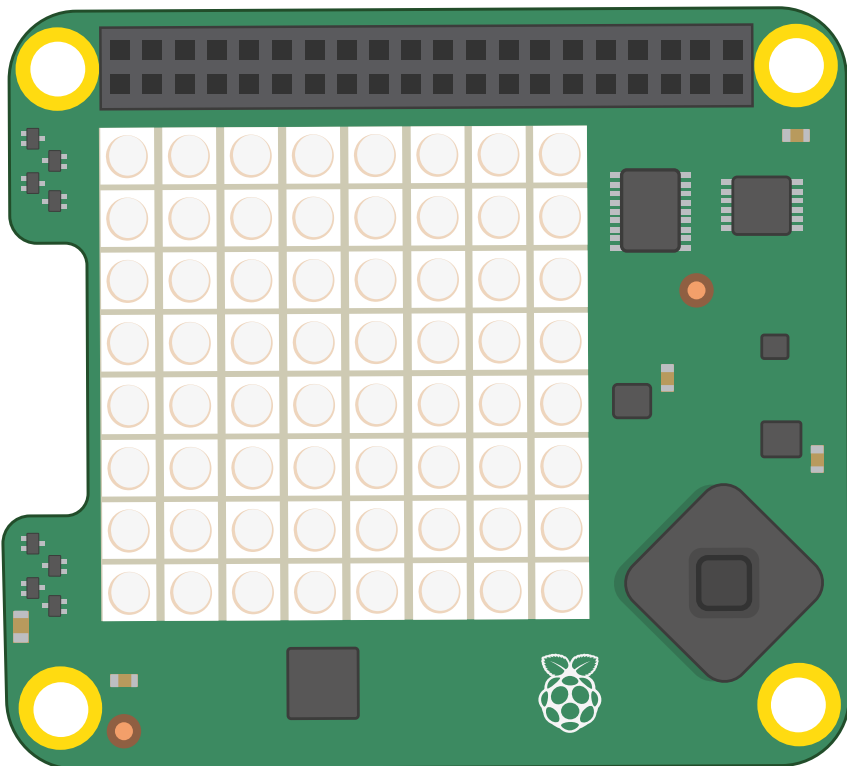
Anche se Ed e Izzy sono un po' lontani per poterli usare, puoi trovare lo stesso hardware Sense HAT anche qui sulla Terra presso tutti i rivenditori Raspberry Pi e, se in questo momento non vuoi comprare un Sense HAT, puoi simularlo tramite software.

REALE O SIMULATO

Questo capitolo indica come utilizzare un Sense HAT reale collegato al terminale GPIO di Raspberry Pi, tuttavia se non disponi di un Sense HAT puoi semplicemente saltare la sezione "Installazione di Sense HAT" e provare a realizzare i progetti nell'emulatore Sense HAT, funzioneranno altrettanto bene!

Cos'è Sense HAT

Sense HAT è un potente componente aggiuntivo per Raspberry Pi, che racchiude diverse funzioni. Oltre a presentare una matrice 8X8 con 64 LED programmabili rossi, verdi e blu (RGB) in grado di produrre milioni di colori, Sense HAT è dotato di un controller joystick a cinque posizioni e sei sensori.



Sensore giroscopico: consente di rilevare i cambiamenti di angolazione nel tempo, ossia la *velocità angolare*, tenendo traccia della direzione del campo di gravità della Terra, ossia la forza che trascina le cose verso il centro del pianeta. In poche parole, il sensore giroscopico ti dice quando stai ruotando Sense HAT rispetto alla superficie della Terra e la velocità di rotazione.

Accelerometro: è simile al sensore giroscopico, ma anziché monitorare un angolo rispetto alla gravità terrestre misura la forza di accelerazione in più direzioni. Le letture (dati) combinate dei due sensori possono aiutarti a comprendere dove sta puntando Sense HAT e come viene mosso.

Magnetometro: misura la forza di un campo magnetico ed è un altro sensore che consente di tenere traccia dei movimenti di Sense HAT. Misurando il campo magnetico naturale della Terra il magnetometro individua la direzione del nord magnetico. Lo stesso sensore può essere usato anche per rilevare oggetti metallici e persino campi elettrici. Questi tre sensori si trovano in un unico chip, denominato "ACCEL/GYRO/MAG", sul circuito di Sense HAT.

Sensore di umidità: misura la quantità di vapore acqueo presente nell'aria, ossia l'*umidità relativa*. L'umidità relativa può variare dallo 0% (quando non c'è acqua) al 100% (quando l'aria è completamente satura). I dati relativi all'umidità possono essere utilizzati per capire quando sta per piovere.

Sensore di pressione barometrica: noto anche come *barometro*, misura la pressione dell'aria. La maggior parte delle persone ha familiarità con la pressione barometrica di cui si parla nelle previsioni del tempo, tuttavia il barometro ha un secondo uso segreto: consente di tenere traccia di quando si sale o si scende da una collina o una montagna, poiché più ci si allontana dal livello del mare più l'aria diventa rarefatta e la pressione si abbassa.

Sensore di temperatura: misura quanto l'ambiente circostante sia caldo o freddo, tuttavia è influenzato dalla temperatura di Sense HAT, infatti se utilizzi una custodia, i valori potrebbero essere più elevati del previsto. Sense HAT non ha un sensore di temperatura separato, ma utilizza sensori di temperatura integrati nei sensori di umidità e di pressione barometrica. Un programma può utilizzare uno o entrambi i sensori, dipende da te.



SENSE HAT SU RASPBERRY PI 400

Sense HAT è compatibile con Raspberry Pi 400 e può essere inserito direttamente nel terminale GPIO dalla parte posteriore. Tuttavia in questo modo i LED non saranno rivolti verso di te e la scheda sarà rivolta verso il basso.

Per risolvere questo problema, è necessario un cavo di prolunga GPIO o una scheda. Tra le estensioni compatibili c'è Black HAT Hack3r range di Pimoroni, puoi usare Sense HAT con la scheda Black HAT Hack3r oppure utilizzare semplicemente il cavo a nastro a 40 pin come estensione. Controlla sempre le istruzioni del produttore per essere sicuro di collegare il cavo e Sense HAT nel modo giusto!



Installazione di Sense HAT

Se disponi di un Sense HAT fisico, inizia rimuovendolo dalla confezione e assicurandoti di avere tutti i pezzi: dovrebbero essere presenti il Sense HAT, quattro perni in metallo o plastica, ossia i *distanziatori*, e otto viti. Potrebbero essere presenti anche otto pin in metallo inseriti in una striscia di plastica nera, come i pin GPIO di Raspberry Pi; se sono presenti, spingi questa striscia con i pin rivolti verso l'alto in direzione della parte inferiore di Sense HAT finché non senti un clic.

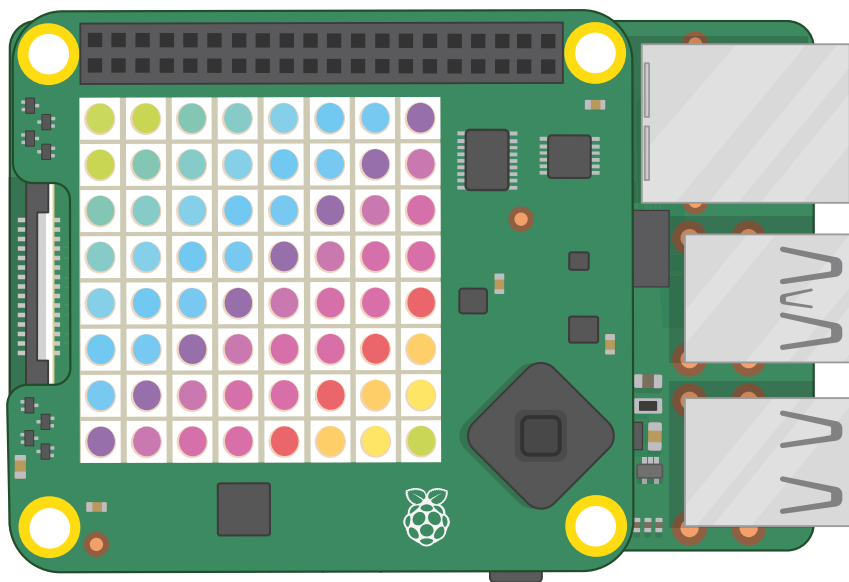
I distanziatori sono progettati per impedire a Sense Hat di piegarsi quando usi il joystick. Sense HAT funzionerà anche se non li installi, tuttavia ti consentono di proteggere Sense HAT, Raspberry Pi e il terminale GPIO da eventuali danni.

ATTENZIONE

I moduli HAT (Hardware Attached on Top) devono essere collegati e rimossi dal terminale GPIO solo quando Raspberry Pi è spento e scollegato dall'alimentazione. Mantieni sempre il modulo HAT piatto durante l'installazione e verifica che sia allineato con i pin del terminale GPIO prima di spingerlo verso il basso.

Installa i distanziatori spingendo quattro delle viti da sotto la parte inferiore di Raspberry Pi attraverso i quattro fori di montaggio in ogni angolo, quindi avvita i distanziatori sulle viti. Spingi Sense HAT sul terminale GPIO di Raspberry Pi, assicurandoti di allinearli correttamente con i pin sottostanti e di mantenerlo il più piatto possibile. Infine, avvita le rimanenti quattro viti attraverso i fori di montaggio su Sense HAT e nei distanziatori installati in precedenza. Se installato correttamente, Sense HAT dovrebbe risultare perfettamente orizzontale e non dovrebbe piegarsi o traballare quando si preme sul joystick.

Collega alla corrente Raspberry Pi e vedrai i LED di Sense HAT accendersi come un arcobaleno (**Figure 7-1**), quindi spegnersi. Hai installato il tuo Sense HAT!



▲ **Figura 7-1:** alla prima accensione si illumina con una sequenza arcobaleno

Se vuoi rimuovere nuovamente Sense HAT, allenta le viti superiori, solleva HAT facendo attenzione a non piegare i perni sul terminale GPIO, in quanto HAT è fissato saldamente (potrebbe essere necessario staccarlo con un piccolo cacciavite), quindi rimuovi i distanziatori da Raspberry Pi.

Ciao Sense HAT!

Come in ogni progetto di programmazione che si rispetti, anche in Sense HAT è consigliato iniziare con un messaggio di benvenuto sul display LED. Se utilizzi l'emulatore Sense HAT, caricalo facendo clic sull'icona del menu Raspberry Pi OS, selezionando la categoria Programmazione e facendo clic su Sense HAT Emulator.

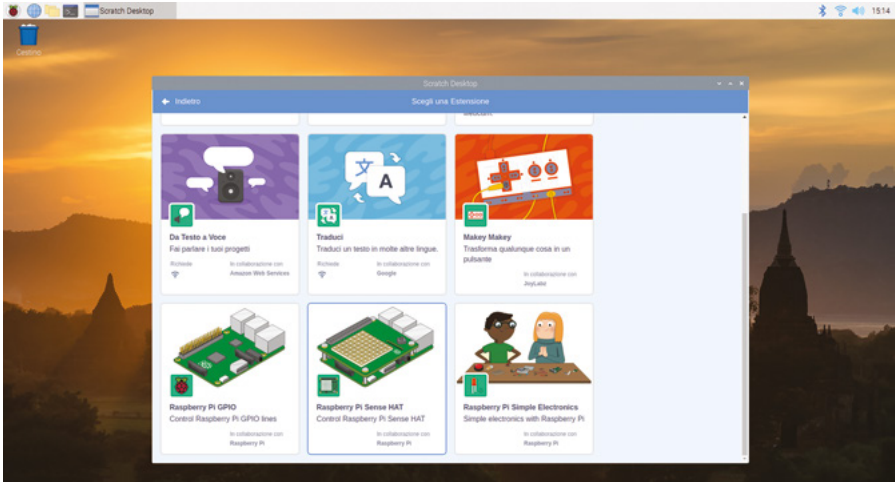


ESPERIENZA DI PROGRAMMAZIONE


Questo capitolo presuppone conoscenze di base di Scratch 3 o Python e dell'ambiente di sviluppo integrato (IDE) Thonny, a seconda del linguaggio di programmazione scelto. Se non l'hai già fatto, leggi il **Capitolo 4, Come programmare con Scratch** oppure il **Capitolo 5, Come programmare con Python** e prova a eseguire i progetti indicati.

Benvenuto da Scratch

Carica Scratch 3 dal menu Raspberry Pi OS. Fai clic sul pulsante di aggiunta estensione in basso a sinistra della finestra Scratch. Fai clic sull'estensione Raspberry Pi Sense HAT (**Figura 7-2**). Si caricherà il blocco necessario per controllare le varie funzioni di Sense HAT, incluso il display LED. Quando ne avrai bisogno, le troverai nella categoria Raspberry Pi Sense HAT.

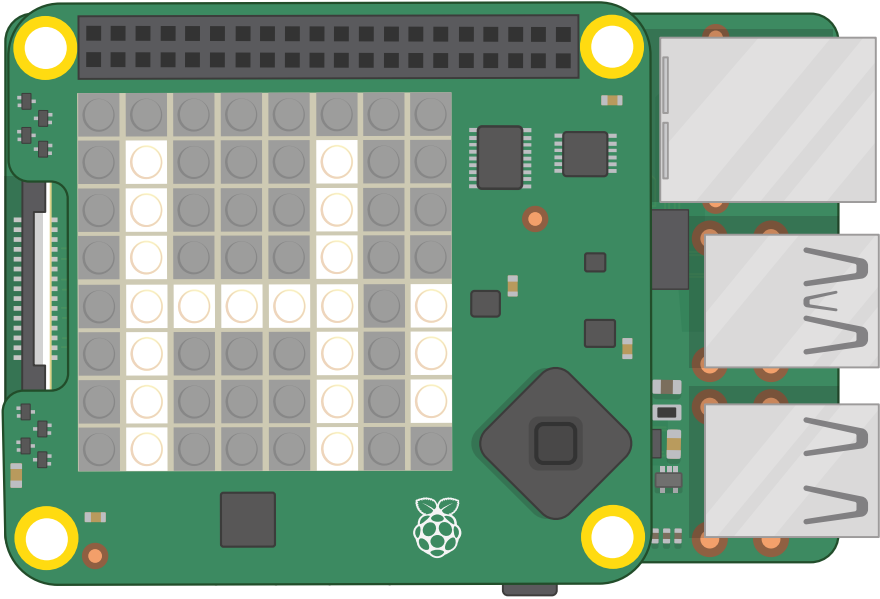


▲ **Figura 7-2:** aggiunta dell'estensione Raspberry Pi Sense HAT a Scratch 3

Inizia trascinando un blocco Situazioni **quando si clicca su**  nell'area script, quindi trascina un blocco **display text Ciao a tutti!** direttamente sotto di esso. Modifica il testo in modo che vi sia scritto **display text Ciao a tutti!**.

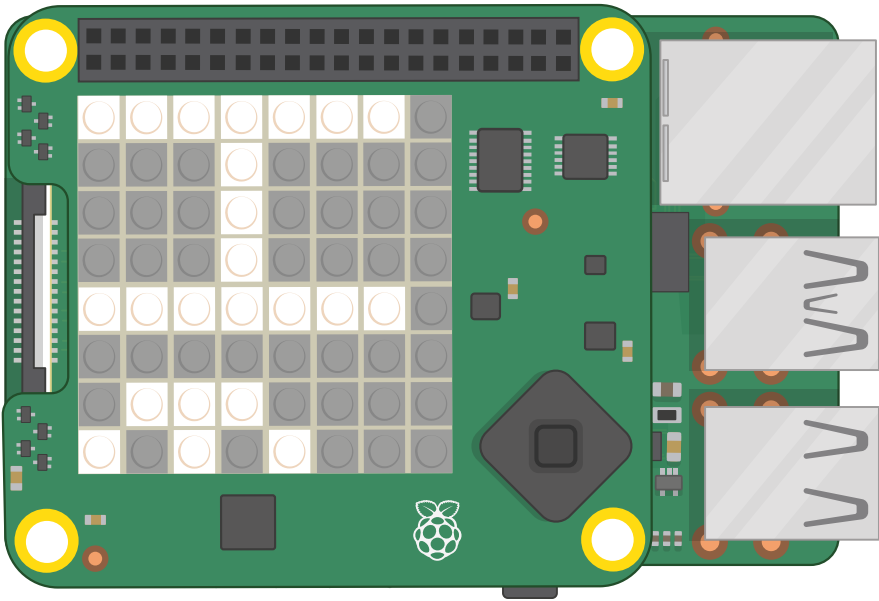


Fai clic sulla bandiera verde nell'area stage e guarda il tuo Sense HAT o l'emulatore: vedrai scorrere lentamente il messaggio nella matrice LED Sense HAT, i pixel LED si accenderanno a turno per formare ciascuna lettera (**Figura 7-3**, sul retro). Complimenti, il tuo programma è un successo!



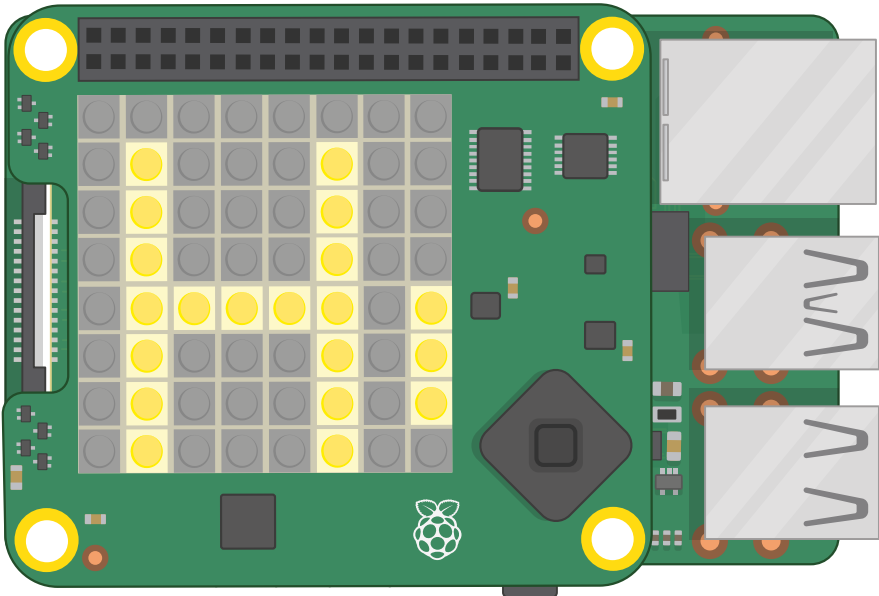
▲ **Figura 7-3:** il tuo messaggio scorre nella matrice LED

Ora che sai come fare scorrere un messaggio, scopriamo come controllarne la modalità di visualizzazione. Oltre a modificare il messaggio, puoi alterarne la rotazione (ossia la direzione in cui viene visualizzato in Sense HAT). Trascina un blocco **ruota di 0 gradi** dalla palette e inseriscilo sotto a **quando si clicca su** e sopra a **display text Ciao a tutti!**, quindi fai clic sulla freccia verso il basso accanto a 0 e porta il valore a 90. Fai clic sulla bandiera verde e vedrai lo stesso messaggio di prima, ma anziché scorrere da sinistra a destra, scorrerà dal basso verso l'alto (**Figura 7-4**): dovrai girare la testa o Sense HAT per leggerlo.



▲ **Figura 7-4:** questa volta il messaggio scorre verticalmente

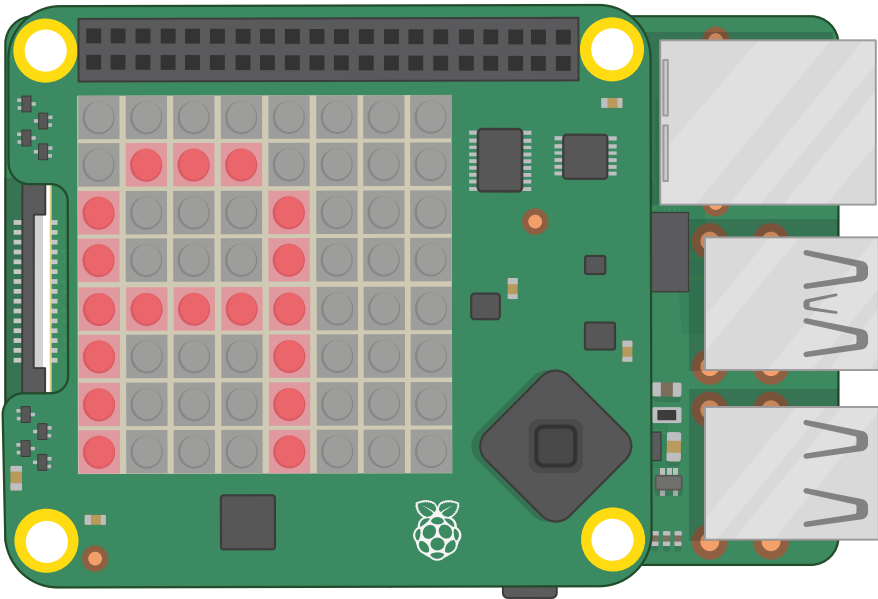
Ora porta nuovamente la rotazione a 0, quindi trascina un blocco **set colour** tra **ruota di 0 gradi** e **display text Ciao a tutti!**. Fai clic sul colore alla fine del blocco per visualizzare il selettore di colori di Scratch e trovare un bel colore giallo brillante, poi fai clic sulla bandiera verde per vedere come è cambiato l'output del programma (**Figura 7-5**).



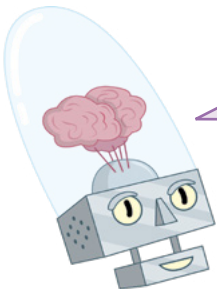
▲ **Figura 7-5:** modifica del colore del testo

Infine, trascina un blocco `set background` tra `set colour to yellow` e `display text Ciao a tutti!`, quindi fai clic sul colore per visualizzare di nuovo il selettore di colori. In questo caso la scelta del colore non influisce sui LED che compongono il messaggio, ma sugli altri, ossia quelli di sfondo. Trova un bel colore blu, poi fai di nuovo clic sulla bandiera verde: questa volta il tuo messaggio sarà giallo brillante su sfondo blu. Prova a cambiare i colori per trovare la tua combinazione preferita, non tutti i colori si abbinano bene tra loro.

Oltre a poter far scorrere interi messaggi, puoi mostrare singole lettere. Trascina il blocco `display text` fuori dall'area script per cancellarlo e sostitiscilo trascinando un blocco `display character A` nell'area script. Fai clic sulla bandiera verde e vedrai la differenza: questo blocco mostra solo una lettera alla volta e la lettera rimane su Sense HAT senza scorrere o scomparire finché non inserisci altre istruzioni. A questo blocco si applicano gli stessi blocchi di controllo del colore del blocco `display text`: prova a cambiare il colore della lettera selezionando il rosso (Figura 7-6).



▲ Figura 7-6: visualizzazione di una sola lettera



SFIDA: RIPETERE IL MESSAGGIO

Riesci a usare le tue nozioni sulle ripetizioni per ottenere un messaggio che scorre e si ripete? Puoi creare un programma che scrive una parola lettera per lettera con diversi colori?



Benvenuto da Python

Carica Thonny facendo clic sull'icona del menu Raspbian, selezionando Programmazione e facendo clic su Thonny. Se utilizzi l'emulatore Sense HAT e viene nascosto dalla finestra Thonny, fai clic e tieni premuto il pulsante del mouse sulla barra del titolo di una delle due finestre (in alto, blu) e trascina per spostarla sul desktop fino a visualizzare entrambe le finestre.



CAMBIO DI RIGA IN PYTHON

Il codice Python scritto per un Sense HAT fisico viene eseguito anche sull'emulatore Sense HAT e viceversa, è necessario apportare solo una modifica. Se utilizzi l'emulatore Sense HAT con Python dovrai modificare la riga `sense_hat import SenseHat` in tutti i programmi di questo capitolo in `from sense_emu import SenseHat`. Se poi vuoi eseguirli su un Sense HAT fisico, ripristina la riga precedente.

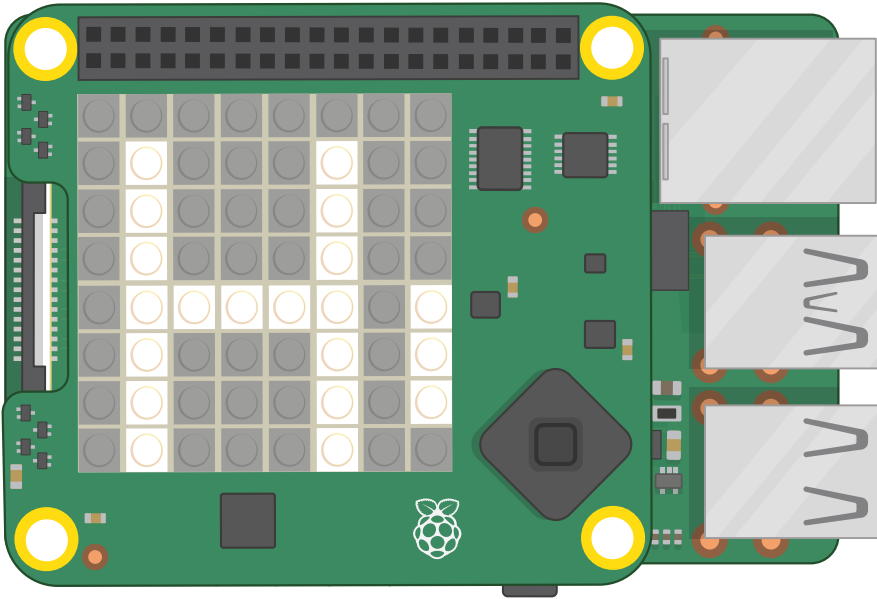
Per utilizzare Sense HAT o l'emulatore in un programma Python dovrai importare la libreria Sense HAT. Digita quanto segue nell'area script, ricordandoti di utilizzare `sense_emu` (al posto di `sense_hat`) se utilizzi un emulatore Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
```

La libreria Sense HAT presenta una funzione semplice per scrivere messaggi, formattarli in modo che possano essere visualizzati sul display LED e fatti scorrere senza problemi. Digita quanto segue:

```
sense.show_message("Ciao a tutti!")
```

Salva il programma come **Ciao Sense HAT**, quindi fai clic sul pulsante Run. Vedrai scorrere lentamente il messaggio nella matrice LED Sense HAT, i pixel LED si accenderanno a turno per formare ciascuna lettera (**Figura 7-7**, sul retro). Complimenti, il tuo programma è un successo!



▲ Figura 7-7: far scorrere un messaggio nella matrice LED

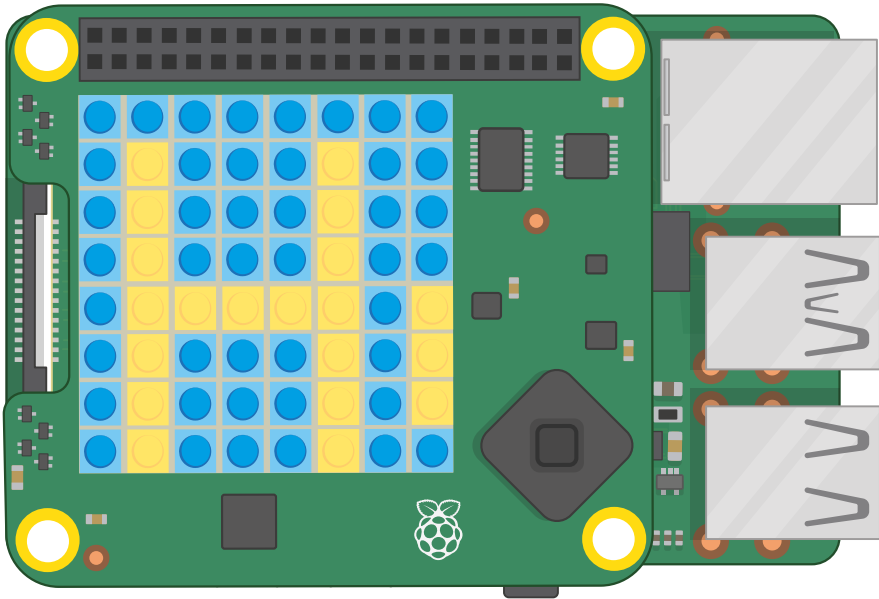
La funzione `show_message()` consente di effettuare altre azioni. Torna al tuo programma e modifica l'ultima riga come segue:

```
sense.show_message("Ciao a tutti!", text_colour=(255, 255, 0),  
back_colour=(0, 0, 255), scroll_speed=(0.05))
```

Queste istruzioni aggiuntive, separate da virgole, sono note come *parametri* e controllano vari aspetti della funzione `show_message()`. Il più semplice è `scroll_speed=()`, che cambia la velocità con cui il messaggio scorre sullo schermo. Inserendo come valore 0.05 la velocità di scorrimento è raddoppiata. Più elevato è il numero, minore è la velocità.

I parametri `text_colour=()` e `back_colour=()` (con ortografia inglese britannico, a differenza della maggior parte delle istruzioni Python) consentono di impostare rispettivamente il colore della scritta e dello sfondo. Tuttavia non accettano nomi di colori: dovrai quindi indicare il colore desiderato con un numero a tre cifre. Il primo numero rappresenta la quantità di rosso nel colore (dove 0 indica l'assenza di rosso e 255 il più rosso possibile), il secondo la quantità di verde, mentre il terzo la quantità di blu. Insieme, questi valori sono noti come *RGB*, che sta per red, green, blue (rosso, verde e blu).

Fai clic sull'icona Run e guarda Sense HAT: questa volta, il messaggio scorrerà molto più velocemente e sarà giallo brillante su sfondo blu (Figura 7-8). Prova a modificare i parametri per trovare la perfetta combinazione di velocità e colore.



▲ **Figura 7-8:** modifica del colore del messaggio e dello sfondo

Se vuoi usare nomi più comprensibili al posto dei valori RGB per impostare i colori, dovrai creare delle variabili. Sopra alla riga `sense.show_message()` aggiungi quanto segue:

```
giallo = (255, 255, 0)
blu = (0, 0, 255)
```

Torna alla riga `sense.show_message()` e modificala come segue:

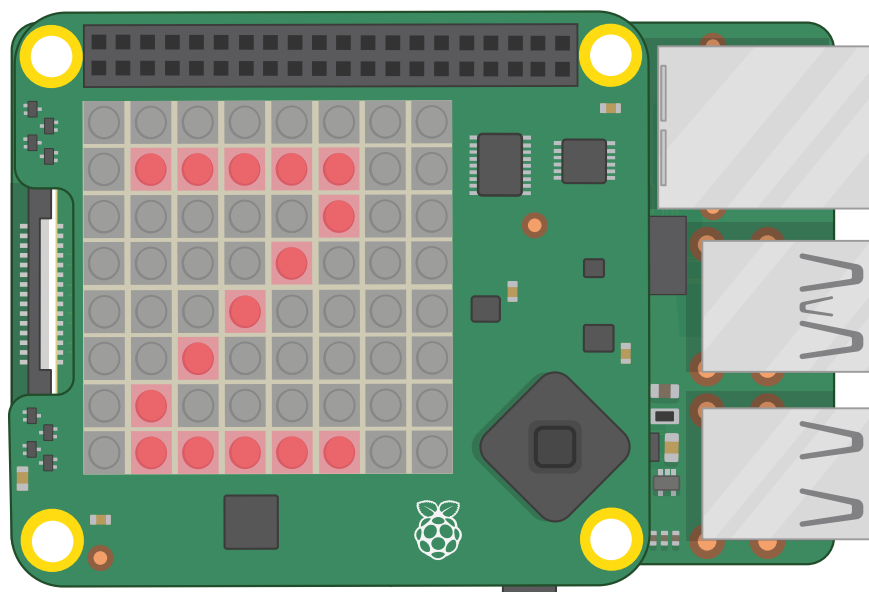
```
sense.show_message("Ciao a tutti!", text_colour=(giallo), back_
colour=(blu), scroll_speed=(0.05))
```

Fai di nuovo clic sull'icona Run e vedrai che non è cambiato nulla: il messaggio è ancora in giallo su sfondo blu. Questa volta, però, avrai usato i nomi delle variabili per rendere il codice più leggibile: invece di una stringa di numeri, il codice indica il colore che sta impostando. Puoi definire tutti i colori che vuoi: prova ad aggiungere una variabile chiamata "rosso" con i valori 255, 0 e 0; una variabile chiamata "bianco" con i valori 255, 255, 255; infine una variabile chiamata "nero" con i valori 0, 0 e 0.

Oltre a poter far scorrere interi messaggi, puoi mostrare singole lettere. Cancella la riga `sense.show_message()` e digita al suo posto la seguente riga:

```
sense.show_letter("Z")
```

Fai clic su Run e vedrai la lettera "Z" apparire sul display Sense HAT. Questa volta rimarrà ferma, infatti le singole lettere, a differenza dei messaggi, non scorrono automaticamente. Puoi controllare `sense.show_letter()` con gli stessi parametri di colore di `sense.show_message()`: prova a cambiare il colore della lettera passando al rosso (Figura 7-9).



▲ Figura 7-9: visualizzazione di una sola lettera



SFIDA: RIPETERE IL MESSAGGIO

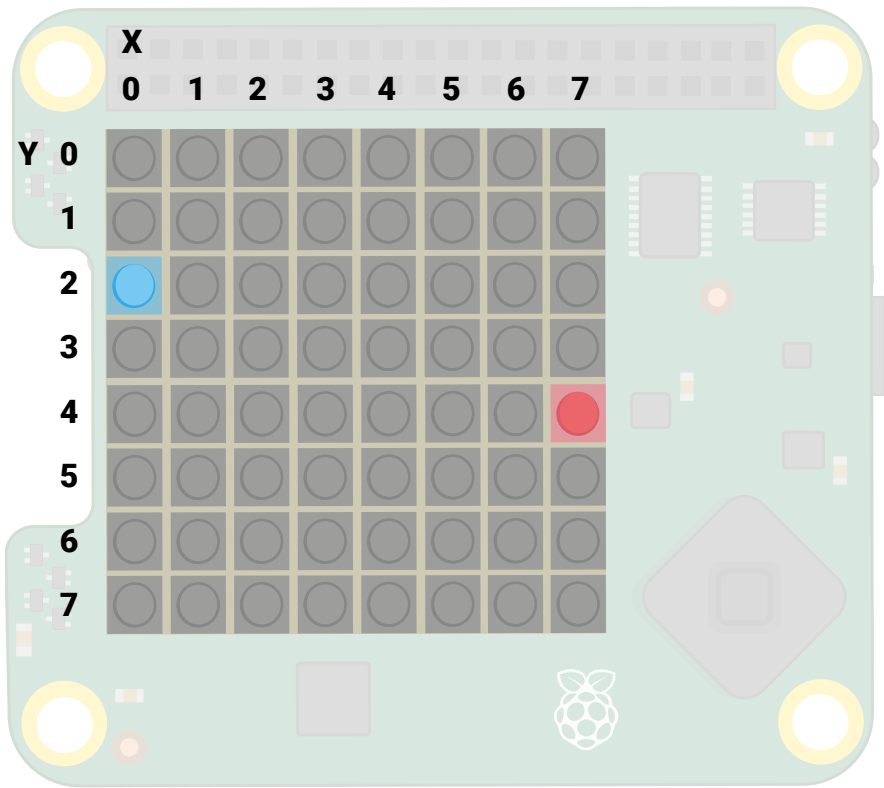
Riesci a usare le tue nozioni sulle ripetizioni per ottenere un messaggio che scorre e si ripete? Puoi creare un programma che scrive una parola lettera per lettera con diversi colori? Quanto velocemente si può far scorrere un messaggio?



Prossimi passaggi: disegnare con le luci

Il display LED Sense HAT non è solo per i messaggi: puoi usarlo anche per rappresentare immagini. Ogni LED può essere trattato come un singolo pixel, abbreviazione che sta per *elemento immagine*, in un'immagine a tua scelta, consentendoti di ravvivare i tuoi programmi con immagini e persino animazioni.

Per creare disegni, tuttavia, dovrai cambiare i singoli LED. Per fare questo, dovrai apprendere come è disposta la matrice LED di Sense HAT per scrivere un programma che accende o spegne i LED corretti.



▲ **Figura 7-10:** la matrice LED coordina il sistema

Ci sono otto LED in ogni riga del display e otto in ogni colonna (**Figura 7-10**). Tuttavia quando si contano i LED, come nella maggior parte dei linguaggi di programmazione, si dovrebbe iniziare da 0 e finire con 7. Il primo LED è in alto a sinistra, l'ultimo in basso a destra. Utilizzando i numeri delle righe e delle colonne, puoi trovare le *coordinate* di qualsiasi LED sulla matrice. Nella matrice raffigurata, il LED blu si trova in corrispondenza delle coordinate 0, 2, mentre il LED rosso alle coordinate 7, 4. Dovrai considerare prima l'asse orizzontale X, poi l'asse verticale Y.

Quando pianifichi le immagini da disegnare su Sense HAT, ti consigliamo di disegnarle a mano su carta millimetrata o a quadretti oppure in un foglio di calcolo come LibreOffice Calc.

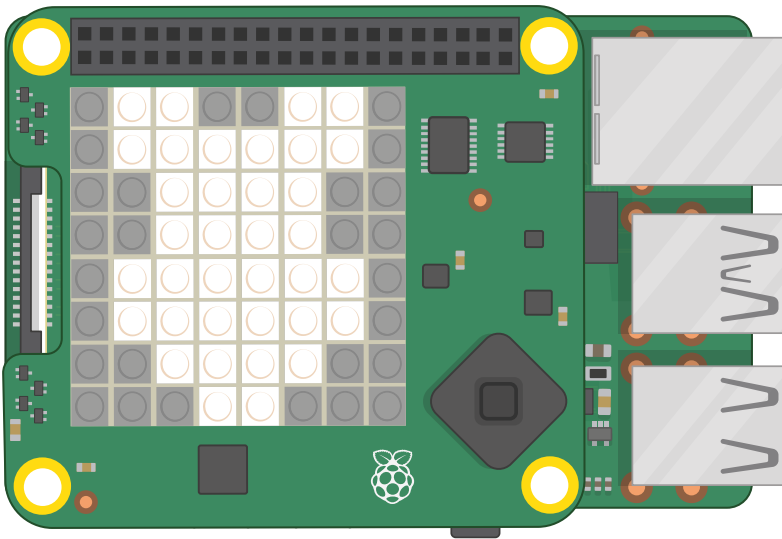
Immagini in Scratch

Salva il programma appena creato, quindi crea un nuovo progetto su Scratch. Se hai lavorato sui progetti di questo capitolo, Scratch 3 manterrà caricata l'estensione Raspberry Pi Sense HAT; se invece hai chiuso e riaperto Scratch 3 dall'ultimo progetto, carica l'estensione usando il pulsante per aggiungerla. Inizia trascinando un blocco Situazioni **quando si clicca su** nell'area codice, quindi trascina i blocchi **set background** e **set colour** direttamente sotto di esso. Modifica entrambi per impostare il colore di sfondo su nero e il colore su bianco: puoi

ottenere il nero facendo scorrere i cursori di luminosità e saturazione fino a raggiungere lo 0 e il bianco facendo scorrere il cursore di luminosità fino a 100 e saturazione fino a 0. Dovrai eseguire quest'operazione all'avvio di ogni programma Sense HAT, altrimenti Scratch utilizzerà gli ultimi colori che hai scelto, anche se li hai selezionati in un programma diverso. Infine trascina un blocco **display lampone** in fondo al tuo programma.



Fai clic sulla bandiera verde: vedrai i LED Sense HAT accendersi formando un lampone (Figura 7-11).

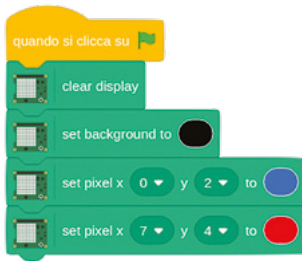


▲ **Figura 7-11: non guardare direttamente i LED quando sono di colore bianco brillante**

Potrai ottenere molte forme oltre a quella del lampone. Fai clic sulla freccia in basso accanto al lampone per attivare la modalità di disegno: è possibile fare clic su qualsiasi LED per accenderlo o spegnerlo, mentre i due pulsanti in basso accendono o spengono tutti i LED. Prova a realizzare il tuo disegno, poi fai clic sulla freccia verde per vederlo su Sense HAT. Prova anche a cambiare il colore del disegno e dello sfondo utilizzando i blocchi indicati in precedenza.

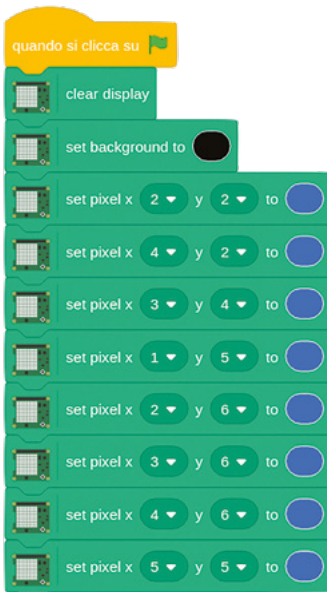
Quando hai finito, trascina i tre blocchi nella palette per cancellarli e metti un blocco **clear display** sotto a **quando si clicca su**, quindi fai clic sulla bandiera verde e tutti i LED si spegneranno.

Per realizzare un'immagine devi essere in grado di controllare i singoli pixel e assegnargli colori diversi. Puoi farlo concatenando i blocchi **display lampone** modificati con blocchi **set colour** oppure puoi gestire ogni singolo pixel. Per creare la tua versione della matrice LED mostrata nell'esempio all'inizio di questa sezione, con due LED specifici illuminati in rosso e blu, lascia il blocco **clear display** all'inizio del programma e trascina un blocco **set background** al di sotto. Modifica il colore del blocco **set background** su nero, poi trascina due blocchi **set pixel x 0 y 0** sotto di esso. Infine modifica questi blocchi come segue:

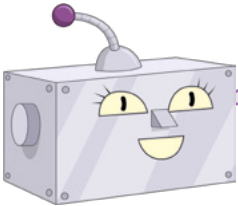


Fai clic sulla bandiera verde, vedrai i LED accendersi in corrispondenza dell'immagine della matrice (**Figura 7-10**) a pagina 165. Complimenti, ora puoi controllare i singoli LED!

Modifica i blocchi di pixel come nel prossimo esempio e trascina altri di seguito fino a ottenere questa sequenza:



Prima di fare clic sulla bandiera verde, vedi se riesci a indovinare quale immagine apparirà in base alle coordinate della matrice di LED che hai usato, poi esegui il tuo programma e vedi se hai ragione!



SFIDA: NUOVE IMMAGINI



Puoi creare altre immagini? Procurati carta millimetrata o a quadretti per pianificare l'immagine che vuoi ottenere disegnandola a mano. Puoi realizzare un'immagine e far cambiare i colori?

Immagini in Python

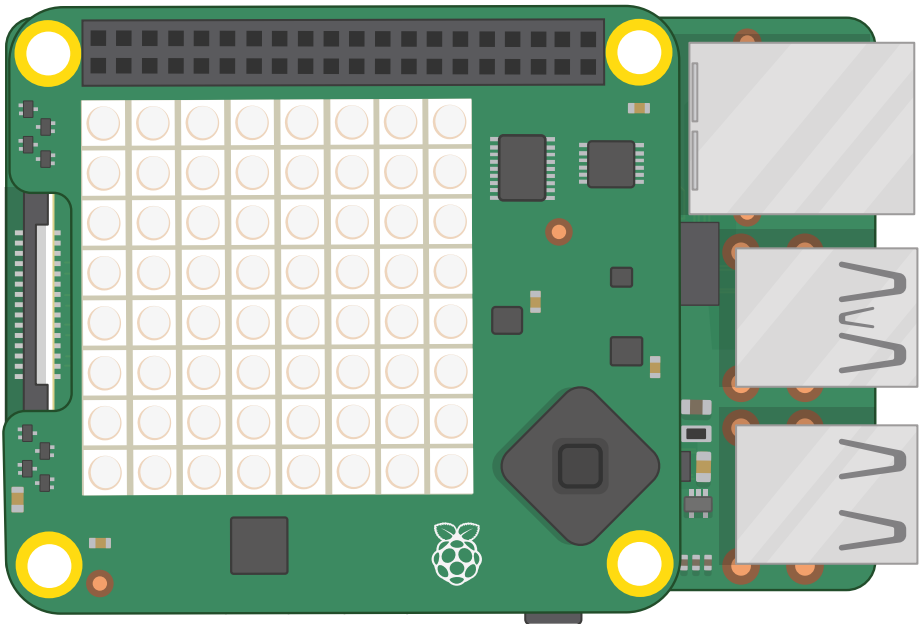
Avvia un nuovo programma e digita quanto segue in Thonny, quindi salva come disegno Sense HAT, ricordandoti di utilizzare `sense_emu` (al posto di `sense_hat`) se utilizzi un emulatore:

```
from sense_hat import SenseHat  
sense = SenseHat()
```

Per utilizzare Sense HAT devi inserire entrambe le righe nel tuo programma. Quindi digita:

```
sense.clear(255, 255, 255)
```

Fai clic sull'icona Run senza guardare direttamente i LED Sense HAT: dovresti comunque vederli diventare tutti di un bianco brillante (Figura 7-12), per questo non devi guardarli direttamente quando esegui il programma.

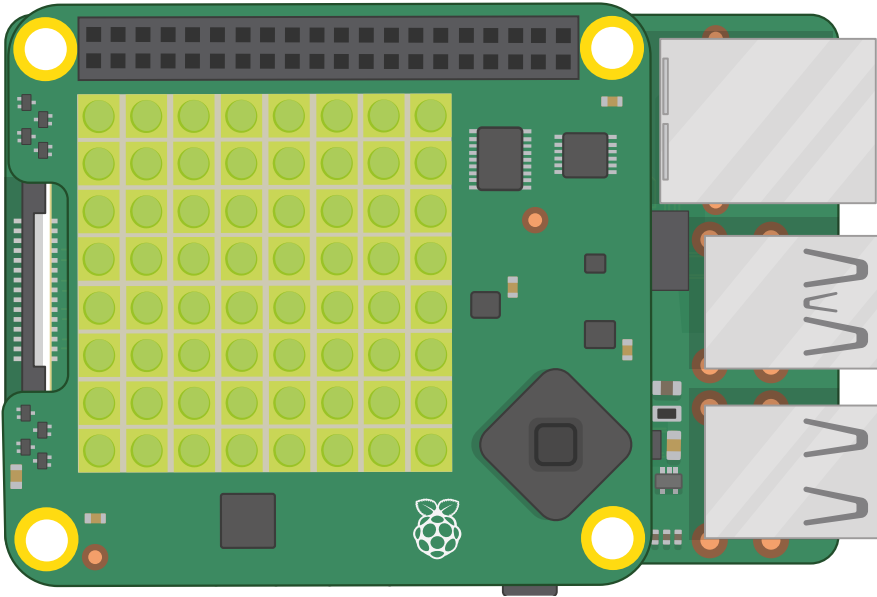


▲ **Figura 7-12:** non guardare direttamente la matrice quando si illumina di colore bianco brillante

`sense.clear()` è progettato per cancellare i LED di qualsiasi programmazione precedente, ma accetta parametri di colore RGB, il che significa che si può ottenere qualsiasi colore. Modifica la riga così:

```
sense.clear(0, 255, 0)
```

Fai clic su Run e Sense HAT diventerà verde brillante (**Figura 7-13**, sul retro). Sperimenta con colori diversi oppure aggiungi le variabili dei nomi dei colori che hai creato per il tuo programma Hello World per agevolare la lettura.

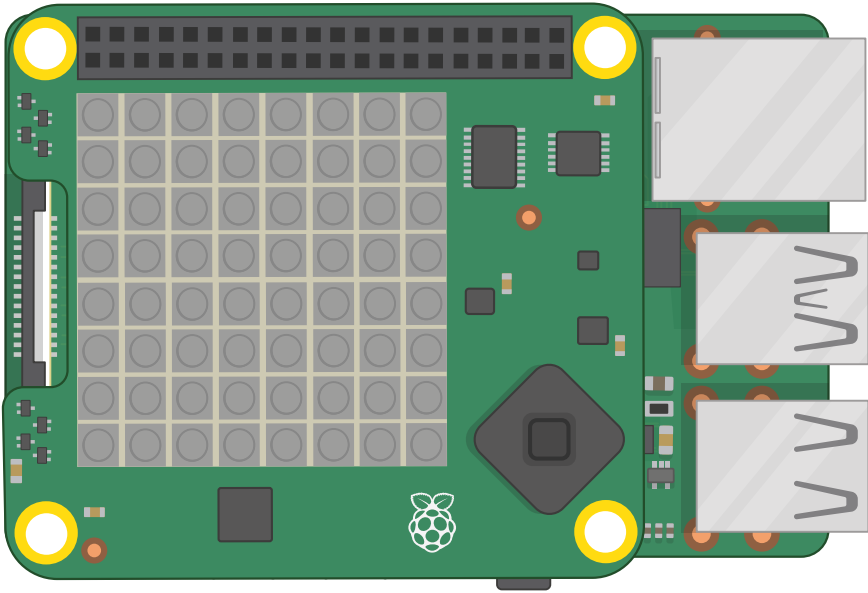


▲ **Figura 7-13:** la matrice LED si illumina di verde brillante

Per cancellare i LED, è necessario utilizzare i valori RGB per il nero: 0 rosso, 0 blu e 0 verde. Ma esiste un modo ancora più semplice. Modifica la riga del programma così:

```
sense.clear()
```

Sense HAT diventerà nero in quanto se non si inserisce nulla tra le parentesi della funzione **sense.clear()** i LED diventano neri, ossia vengono spenti (**Figura 7-14**). Questa è la funzione da utilizzare quando è necessario spegnere completamente i LED nei programmi.



▲ **Figura 7-14:** utilizza la funzione `sense.clear()` per spegnere tutti i LED

Per creare la tua versione della matrice LED mostrata nell'esempio precedente in questo capitolo, con due LED specifici illuminati in rosso e blu, aggiungi le seguenti righe al programma dopo `sense.clear()`:

```
sense.set_pixel(0, 2, (0, 0, 255))
sense.set_pixel(7, 4, (255, 0, 0))
```

La prima coppia di numeri è la posizione del pixel sulla matrice, l'asse X (orizzontale) seguito dall'asse Y (verticale). La seconda, tra parentesi tonde, sono i valori RGB che indicano il colore dei pixel. Fai clic sul pulsante Run e vedrai l'effetto: due LED del Sense HAT si accenderanno, proprio come nella **Figura 7-10** a pagina 165.

Elimina queste due righe e digita quanto segue:

```
sense.set_pixel(2, 2, (0, 0, 255))
sense.set_pixel(4, 2, (0, 0, 255))
sense.set_pixel(3, 4, (100, 0, 0))
sense.set_pixel(1, 5, (255, 0, 0))
sense.set_pixel(2, 6, (255, 0, 0))
sense.set_pixel(3, 6, (255, 0, 0))
sense.set_pixel(4, 6, (255, 0, 0))
sense.set_pixel(5, 5, (255, 0, 0))
```

Prima di fare clic su Run, guarda le coordinate e confrontale con la matrice: riesci a indovinare l'immagine che otterrai? Fai clic su Run per scoprire se hai ragione!

Disegnare un'immagine dettagliata utilizzando la funzione `set_pixel()` per gestire ogni singolo pixel richiede molto tempo. Per lavorare con più velocità, modifica più pixel contemporaneamente. Cancella tutte le righe `set_pixel()` e digita quanto segue:

```
g = (0, 255, 0)
b = (0, 0, 0)
```

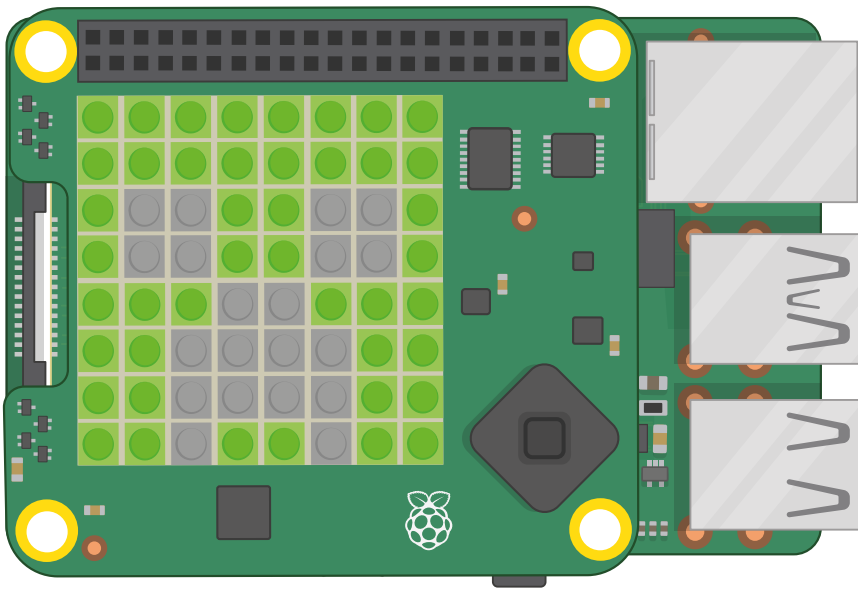
```
creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, b, b, g, g, b, b, g,
    g, b, b, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g
]
```

```
sense.set_pixels(creeper_pixels)
```

C'è molto da fare, ma fai clic su Run per vedere se riconosci un piccolo creeper. Le prime due linee creano due variabili per i colori: verde e nero. Per rendere il codice più facile da scrivere e leggere, le variabili sono lettere singole: "g" per il verde e "b" per il nero.

Il blocco di codice successivo crea una variabile che contiene i valori di colore per i 64 pixel della matrice LED, separati da virgole e racchiusi tra parentesi quadre. Anziché i numeri usa le variabili di colore create prima: guarda bene, tenendo a mente che "g" indica il verde e "b" il nero, e vedrai l'immagine apparire (Figura 7-15).

Infine `sense.set_pixels(creeper_pixels)` prende la variabile e usa la funzione `sense.set_pixels()` per disegnare su tutta la matrice. È molto più semplice che cercare di disegnare pixel per pixel.



▲ **Figura 7-15:** visualizzazione di un'immagine sulla matrice

Puoi anche ruotare e capovolgere le immagini, sia per mostrarle nel modo giusto quando Sense HAT è girato, sia per creare semplici animazioni da un'unica immagine asimmetrica.

Inizia modificando la variabile `creeper_pixels` per chiudere l'occhio sinistro sostituendo i quattro pixel "b" con "g", iniziando dai primi due sulla terza riga e poi passando ai primi due sulla quarta:

```
creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, b, b, g,
    g, g, g, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g
]
```

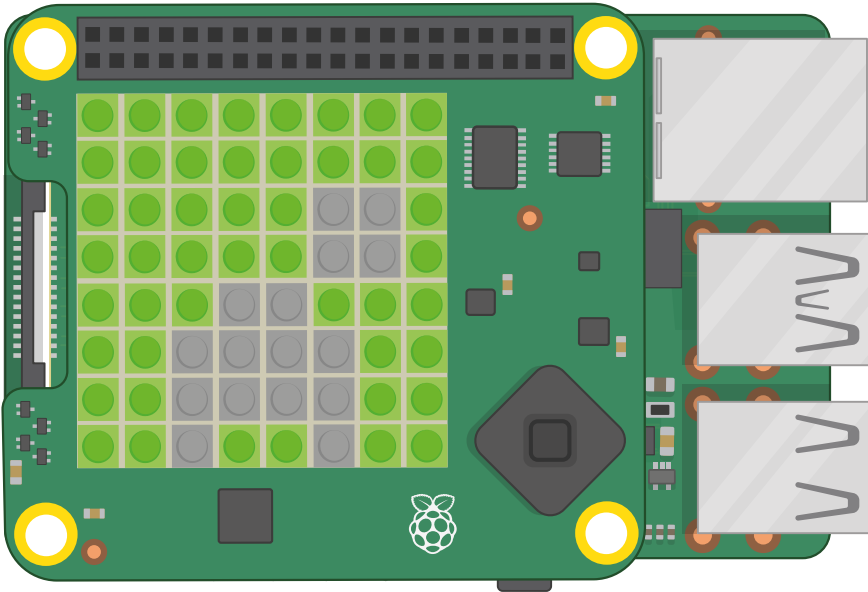
Fai clic su Run e vedrai l'occhio sinistro del creeper chiudersi (**Figura 7-16**, sul retro). Per realizzare un'animazione torna all'inizio del programma e inserisci la riga:

```
from time import sleep
```

Quindi vai alla fine e digita:

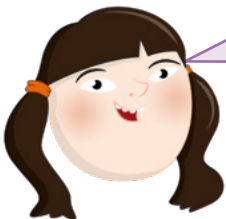
```
while True:  
    sleep(1)  
    sense.flip_h()
```

Fai clic su Run e guarda il creeper mentre chiude e apre gli occhi uno alla volta!



▲ Figura 7-16: una semplice animazione a due fotogrammi

La funzione `flip_h()` capovolge un'immagine sull'asse orizzontale; se invece vuoi capovolgere un'immagine sul suo asse verticale, sostituisci `sense.flip_h()` con `sense.flip_v()`. Puoi anche ruotare un'immagine di 0, 90, 180 o 270 gradi utilizzando `sense.set_rotation(90)` e modificando il numero in base a quanti gradi desideri ruotare l'immagine. Prova questa funzione per vedere il creeper girare anziché chiudere e aprire gli occhi.



SFIDA: NUOVE IMMAGINI

Puoi creare altre immagini e animazioni? Procurati carta millimetrata o a quadretti per pianificare l'immagine che vuoi ottenere disegnandola a mano per rendere più semplice la scrittura della variabile. Puoi realizzare un'immagine e far cambiare i colori? Suggerimento: puoi modificare le variabili dopo averle già utilizzate.



Percepire il mondo che ti circonda

La caratteristica principale di Sense HAT sono i sensori. Questi consentono di rilevare valori che vanno dalla temperatura all'accelerazione e di utilizzarli nei programmi.



EMULAZIONE DEI SENSORI

Se utilizzi l'emulatore Sense HAT dovrai abilitare la simulazione dei sensori inerziali e ambientali: nell'emulatore, fai clic su Edit, poi su Preferences, quindi spunta le opzioni. Nello stesso menu scegli "180°..360°|0°..180°" sotto 'Orientation Scale' per assicurarti che i numeri nell'Emulatore corrispondano a quelli riportati da Scratch e Python, quindi fai clic sul pulsante di chiusura.

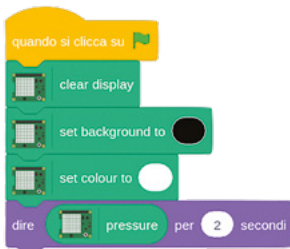
Rilevamento ambientale

I sensori di pressione barometrica, umidità e temperatura sono sensori ambientali che rilevano i valori dell'ambiente in cui si trova Sense HAT.

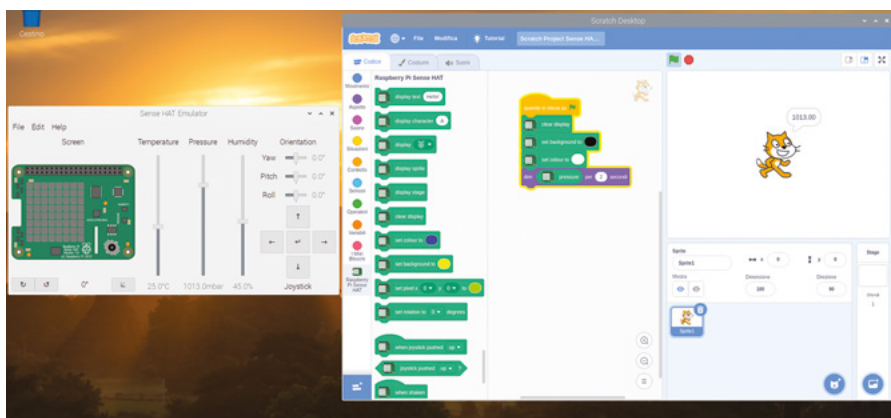
Rilevamento ambientale in Scratch

Avvia un nuovo programma in Scratch, salvando il precedente se lo desideri, e aggiungi l'estensione Sense HAT Raspberry Pi se non è già presente. Inizia trascinando un blocco Situazioni **quando si clicca su** nell'area codice, quindi trascina un blocco **clear display** sotto di esso e poi un blocco **set background to nero** sotto il blocco precedente. Quindi aggiungi un blocco **set colour to bianco** e utilizza i cursori Luminosità e Saturazione per scegliere il colore corretto. Ti consigliamo di farlo all'inizio della programmazione, così sarai certo che Sense HAT non mostri pattern di vecchi programmi e potrai essere sicuro dei colori scelti.

Trascina un blocco Aspetto **dire Ciao! per 2 secondi** direttamente sotto i blocchi già presenti. Per effettuare una lettura dal sensore di pressione, cerca il blocco **pressure** nella categoria Sense HAT Raspberry Pi e trascinarlo sopra la parola "Ciao!" nel tuo blocco **dire Ciao! per 2 secondi**.



Fai clic sulla bandiera verde e il gatto di Scratch ti dirà la pressione in *millibar* rilevata dal sensore. Il messaggio scomparirà dopo due secondi: prova a soffiare su Sense HAT (o a spostare verso l'alto il cursore della pressione nell'emulatore) e a fare clic sulla bandiera verde per eseguire nuovamente il programma; questa volta dovresti vedere un valore più alto (**Figura 7-17**, sul retro).



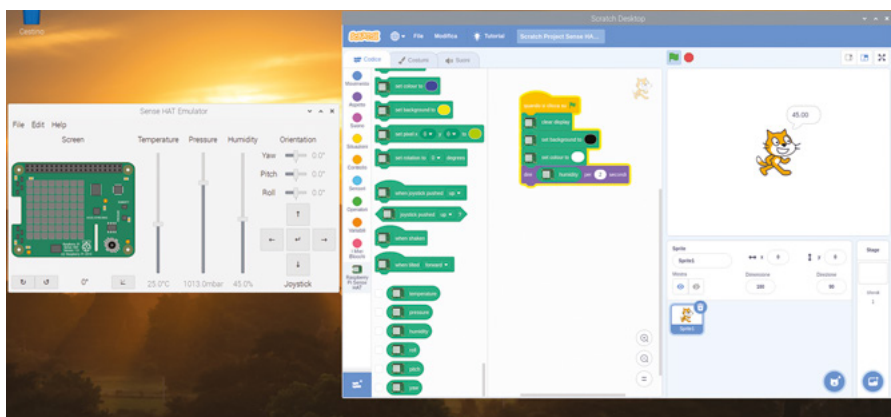
▲ **Figura 7-17:** visualizzazione della lettura del sensore di pressione



MODIFICA DEI VALORI

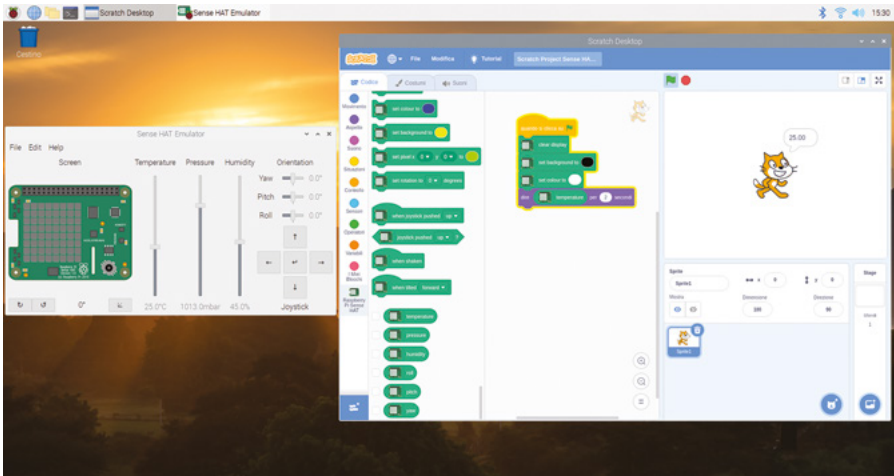
Se utilizzi l'emulatore Sense HAT puoi modificare i valori indicati da ciascun sensore mediante cursori e pulsanti. Prova a far scorrere il sensore di pressione verso il basso e fai clic nuovamente sulla bandiera verde.

Per passare al sensore di umidità, cancella il blocco **pressure** e sostituisilo con **humidity**. Esegui nuovamente il programma e vedrai l'umidità relativa attuale della stanza. Anche in questo caso puoi provare ad eseguirlo di nuovo mentre soffi su Sense HAT (o spostando il cursore dell'umidità dell'emulatore verso l'alto) per modificare la lettura (**Figura 7-18**); il respiro è infatti particolarmente umido.

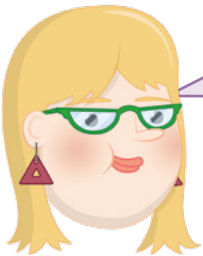


▲ **Figura 7-18:** visualizzazione della lettura del sensore di umidità

Per utilizzare il sensore di temperatura, dovrai solamente cancellare il blocco **humidity** e sostituirlo con **temperature**, quindi eseguire nuovamente il programma. Visualizzerai la temperatura in gradi Celsius (**Figura 7-19**). Tuttavia, questa potrebbe non essere la temperatura esatta della tua stanza: Raspberry Pi genera calore mentre è in funzione e così anche Sense HAT e i suoi sensori si riscaldano.



▲ **Figura 7-19:** Visualizzazione della lettura del sensore di temperatura



SFIDA: SCORRIMENTO E RIPETIZIONE

Riesci a modificare il programma per effettuare una lettura da ciascuno dei sensori a turno, quindi farli scorrere attraverso la matrice LED piuttosto che stamparli nell'area stage? Puoi far eseguire in modo continuo il programma in modo che stampi sempre le condizioni ambientali di ogni momento?

Rilevamento ambientale in Python

Per iniziare a rilevare le letture dei sensori, crea un nuovo programma in Thonny e salvalo come **Sensori Sense HAT**. Digita quanto segue nell'area script, ricordandoti di utilizzare **sense_emu** se utilizzi un emulatore Sense HAT:

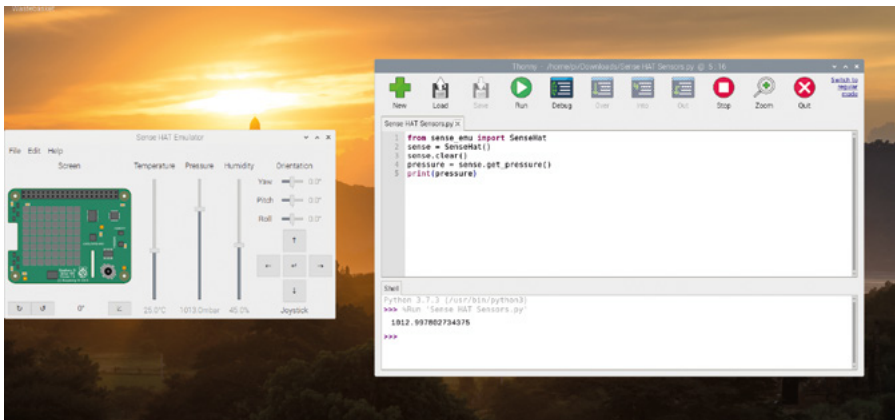
```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

È sempre una buona idea includere **sense.clear()** all'inizio dei programmi nel caso in cui Sense HAT mostri ancora qualcosa dell'ultimo programma che ha eseguito.

Per effettuare una lettura dal sensore di pressione, digita:

```
pressure = sense.get_pressure()
print(pressure)
```

Fai clic su Run e vedrai un numero in Python shell in fondo alla finestra Thonny. Questa è la lettura della pressione dell'aria in *millibar* rilevata dal sensore di pressione barometrica (**Figura 7-20**). Prova a soffiare su Sense HAT (o a spostare verso l'alto il cursore della pressione nell'emulatore) e a fare nuovamente clic sull'icona Run: dovresti vedere un valore più alto.



▲ **Figura 7-20:** stampa della lettura della pressione da Sense HAT



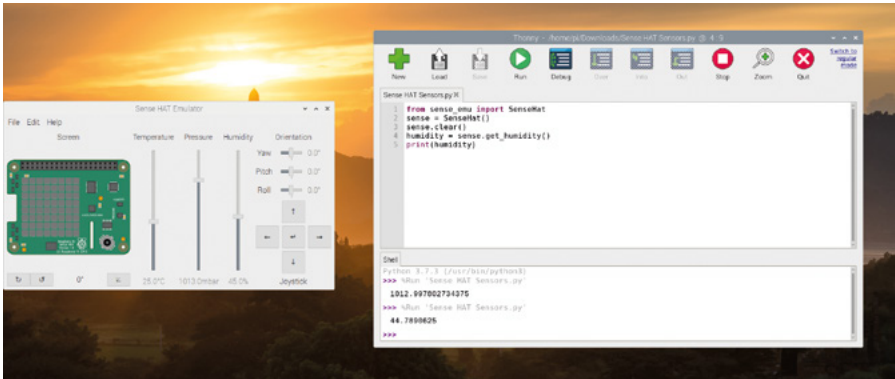
MODIFICA DEI VALORI

Se utilizzi l'emulatore Sense HAT puoi modificare i valori indicati da ciascun sensore mediante cursori e pulsanti. Prova a far scorrere il sensore di pressione verso il basso e fai clic nuovamente su Run.

Per passare al sensore di umidità, cancella le ultime due righe di codice e sostituiscile con:

```
humidity = sense.get_humidity()
print(humidity)
```

Fai clic su Run e vedrai un numero in Python shell: questa volta è l'umidità relativa della tua stanza indicata con una percentuale. Anche in questo caso puoi soffiare su Sense HAT (o spostare il cursore dell'umidità dell'emulatore verso l'alto) per modificare la lettura (**Figura 7-21**); il respiro è infatti particolarmente umido.

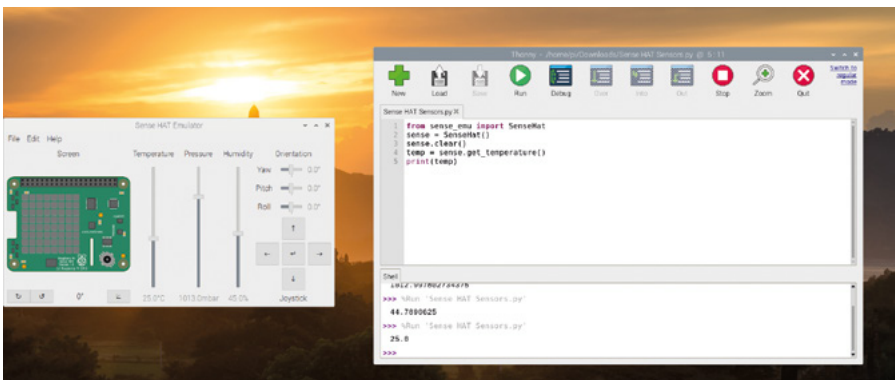


▲ **Figura 7-21:** visualizzazione della lettura del sensore di umidità

Per passare al sensore di temperatura, cancella le ultime due righe di codice e sostituiscele con:

```
temp = sense.get_temperature()
print(temp)
```

Fai di nuovo clic su Run e visualizzerai la temperatura in gradi Celsius (**Figura 7-22**). Tuttavia, questa potrebbe non essere la temperatura esatta della tua stanza: Raspberry Pi genera calore mentre è in funzione e così anche Sense HAT e i suoi sensori si riscaldano.



▲ **Figura 7-22:** visualizzazione della lettura della temperatura

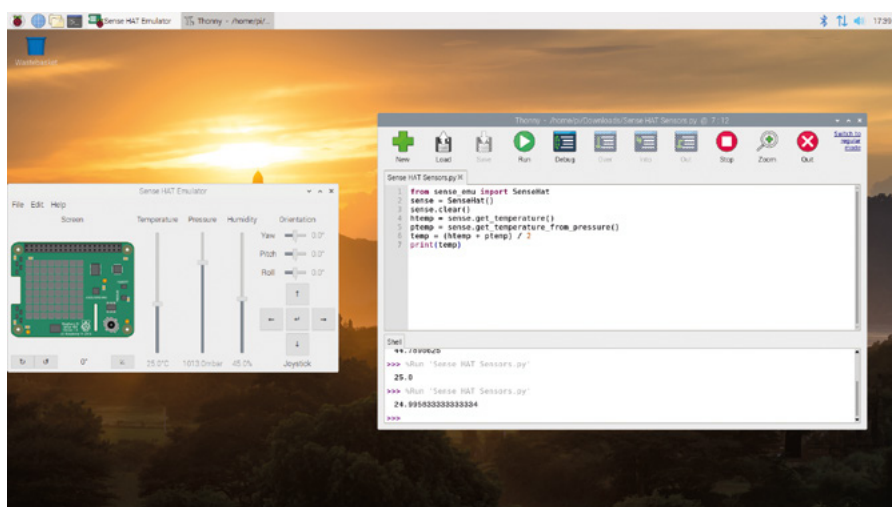
Generalmente Sense HAT riporta la temperatura sulla base di una lettura del sensore di temperatura integrato nel sensore di umidità; se invece vuoi utilizzare la lettura dal sensore di pressione, utilizza **sense.get_temperature_from_pressure()**. È possibile unire le due letture per ottenere una media, che è generalmente più precisa rispetto all'utilizzo dei singoli sensori. Cancella le ultime due righe di codice e sostituiscele con:

```

htemp = sense.get_temperature()
ptemp = sense.get_temperature_from_pressure()
temp = (htemp + ptemp) / 2
print(temp)

```

Fai clic sull'icona Run e vedrai un numero nella console Python (Figura 7-23). In questo caso si basa sulle letture di entrambi i sensori sommate e divise per due (il numero di letture) per ottenere una media. Se utilizzi l'emulatore, tutti e tre i metodi (umidità, pressione e media) mostreranno lo stesso numero.



▲ Figura 7-23: temperatura basata sulle letture di entrambi i sensori



SFIDA: SCORRIMENTO E RIPETIZIONE

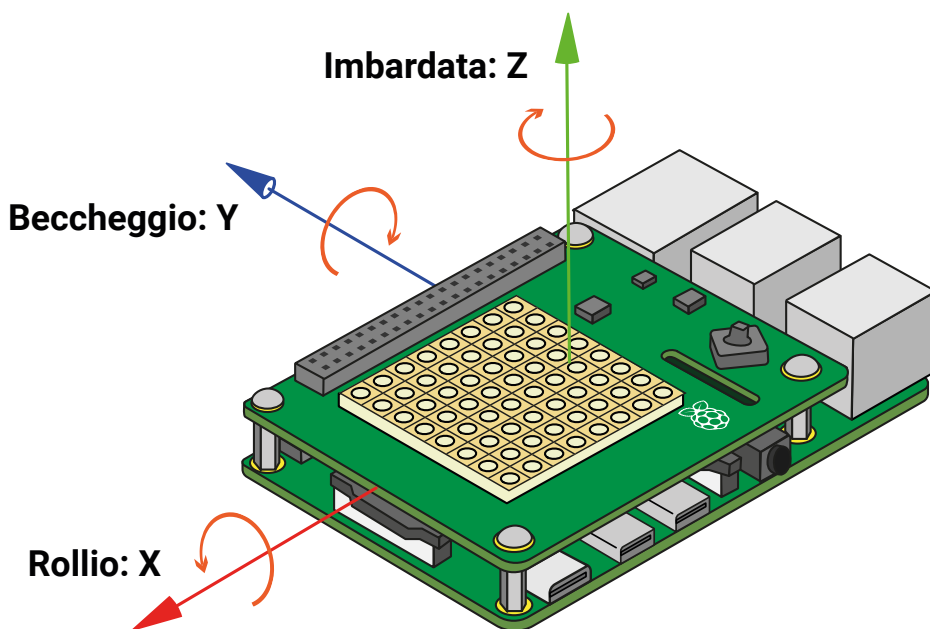


Riesci a modificare il programma per effettuare una lettura da ciascuno dei sensori a turno, quindi farli scorrere attraverso la matrice LED piuttosto che stamparli in shell? Puoi far eseguire in modo continuo il programma in modo che stampi sempre le condizioni ambientali di ogni momento?

Rilevamento inerziale

Il sensore giroscopico, l'accelerometro e il magnetometro insieme formano la cosiddetta *unità di misura inerziale (IMU)*. Tecnicamente parlando, questi sensori rilevano i valori dall'ambiente circostante proprio come i sensori ambientali (il magnetometro, per esempio, misura l'intensità del campo magnetico), tuttavia sono usati per raccogliere i dati sul movimento del Sense HAT stesso. L'IMU è la somma di più sensori: mentre alcuni linguaggi di programmazione permettono di effettuare letture da ogni sensore in modo indipendente, altri forniscono solo una lettura combinata.

Per comprendere l'IMU è necessario apprendere come si muovono gli oggetti. Sense HAT e il Raspberry Pi a cui è collegato possono muoversi lungo tre assi spaziali: da un lato all'altro sull'asse X, in avanti e indietro sull'asse Y e su e giù sull'asse Z (**Figura 7-24**). Può anche ruotare lungo questi tre assi, ma i nomi cambiano: la rotazione sull'asse X è chiamata *rollio*, sull'asse Y *beccheggio* e sull'asse Z *imbardata*. Quando ruoti Sense HAT sull'asse corta modifichi il suo beccheggio, sull'asse lunga rollio e se lo ruoti mantenendolo pari sul tavolo modifichi la sua imbardata. È come se fosse un aereo: quando sta decollando, aumenta il suo beccheggio per salire, quando ruota su se stesso, sta ruotando attorno all'asse di rollio, mentre quando usa il timone per virare senza rollio, è l'imbardata.

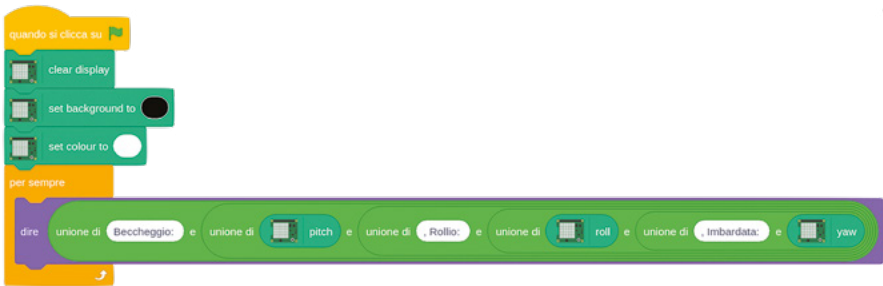


▲ **Figura 7-24:** gli assi spaziali dell'IMU di Sense HAT

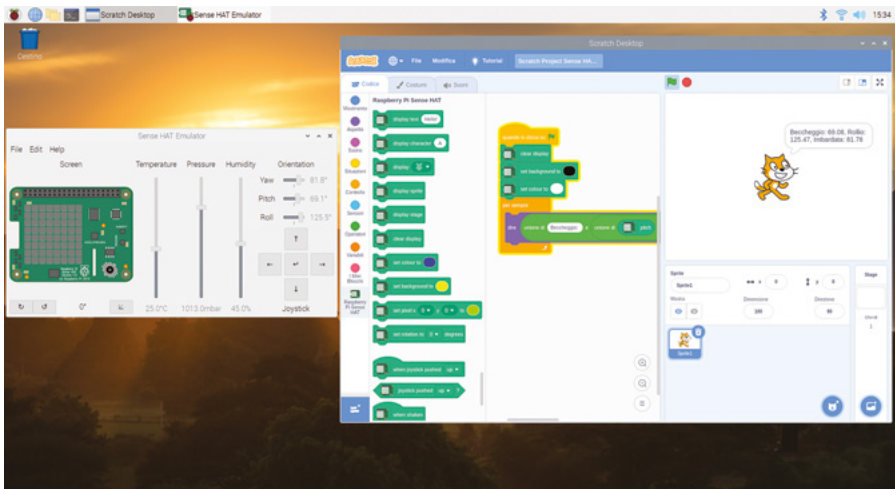
Rilevamento inerziale in Scratch

Avvia un nuovo programma in Scratch e aggiungi l'estensione Sense HAT Raspberry Pi se non è già presente. Avvia il programma come prima: inizia trascinando un blocco Situazioni **quando si clicca su** nell'area codice, trascina un blocco **clear display** sotto di esso, quindi trascina e modifica un blocco **set background to** e **set colour to**.

Infine trascina un blocco **per sempre** in fondo al tuo programma e inserisci un blocco **dire Ciao!**. Per mostrare una lettura per le tre assi IMU (beccheggio, rollio e imbardata), dovrai aggiungere dei blocchi Operatori **unione di** oltre ai blocchi Sense HAT corrispondenti. Ricordati di includere spazi e virgole, in modo che l'output sia di facile lettura.



Fai clic sulla bandiera verde per eseguire il programma e provare a spostare Sense HAT e Raspberry facendo attenzione a non staccare i cavi. Mentre inclini Sense HAT lungo i suoi tre assi, vedrai valori di beccheggio, rollio e imbardata cambiare (**Figura 7-25**).



▲ **Figura 7-25:** visualizzazione dei valori di beccheggio, rollio e imbardata

Rilevamento inerziale in Python

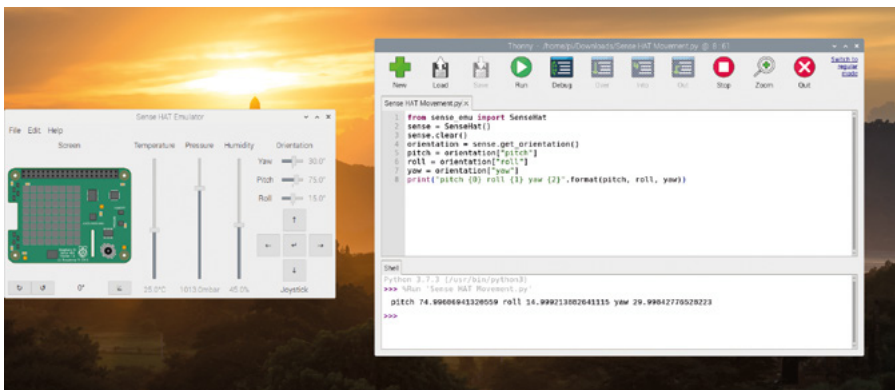
Crea un nuovo programma in Thonny e salvalo come **Movimento Sense HAT**. Digita le righe iniziali usate anche in precedenza, ricordandoti di utilizzare `sense_emu` se utilizzi un emulatore Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Per utilizzare le informazioni IMU per elaborare l'orientamento attuale di Sense HAT sui suoi tre assi, digita quanto segue:

```
orientation = sense.get_orientation()
pitch = orientation["pitch"]
roll = orientation["roll"]
yaw = orientation["yaw"]
print("beccheggio {0} rollio {1} imbardata {2}".format(pitch,
roll, yaw))
```

Fai clic su Run e vedrai le letture relative all'orientamento di Sense HAT suddivise sui tre assi (**Figura 7-26**). Prova a ruotare Sense HAT e fai nuovamente clic su Run: dovresti vedere i numeri cambiare in base all'orientamento.



▲ **Figura 7-26:** visualizzazione dei valori di beccheggio, rollio e imbardata

Oltre all'orientamento, con l'unità di misura inerziale è possibile rilevare anche il movimento. In questo caso l'IMU deve essere letto frequentemente in ripetizione: a differenza dell'orientamento, infatti, il rilevamento di una sola lettura non fornisce informazioni utili quando si tratta di individuare il movimento. Elimina tutte le righe dopo `sense.clear()` e digita il seguente codice:

while True:

```
    acceleration = sense.get_accelerometer_raw()
    x = acceleration["x"]
    y = acceleration["y"]
    z = acceleration["z"]
```

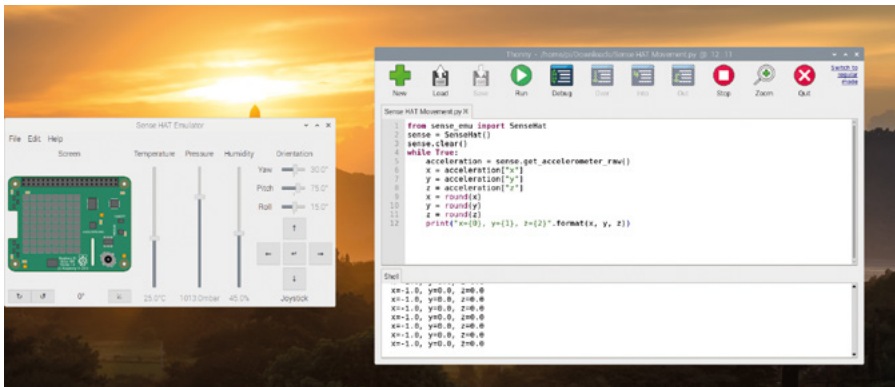
Ora disponi delle variabili che contengono le letture dell'accelerometro per i tre assi spaziali: X (sinistra e destra), Y (avanti e indietro) e Z (su o giù). I numeri del sensore dell'accelerometro potrebbero essere difficili da leggere, quindi digita quanto segue per renderli più comprensibili arrotondandoli al numero intero più vicino:

```
x = round(x)
y = round(y)
z = round(z)
```

Infine stampa i tre valori digitando quanto segue:

```
print("x={0}, y={1}, z={2}".format(x, y, z))
```

Fai clic su Run e vedrai i valori dell'accelerometro stampati nell'area Python shell (**Figura 7-27**). A differenza del programma precedente, verranno stampati in modo continuo. Per interrompere la stampa, fai clic sul pulsante rosso di arresto.



▲ **Figura 7-27:** letture dell'accelerometro arrotondate al numero intero più vicino

Avrai notato che l'accelerometro ti sta dicendo che uno degli assi (l'asse Z, se Raspberry Pi è piatto sul tavolo) ha un valore di accelerazione di gravità pari a 1.0 (1G) anche se Sense HAT non si muove. Questo perché sta rilevando l'attrazione gravitazionale della Terra, ossia la forza che attrae Sense HAT verso il centro della Terra e la ragione per cui se si butta giù qualcosa dalla scrivania cadrà a terra.

Con il programma in esecuzione, prova a prendere con delicatezza Sense HAT e Raspberry Pi e a farli ruotare, assicurandoti di non staccare i cavi. Con la rete di Raspberry Pi e le porte USB che puntano verso il pavimento, i valori saranno i seguenti: 0G per l'asse Z e 1G per l'asse X; se giri di nuovo in modo che le porte HDMI e dell'alimentazione puntino verso il pavimento, sarà l'asse Y a leggere 1G. Se invece rivolgi la porta HDMI verso il soffitto, il valore dell'asse Y sarà -1G.

Usando le conoscenze che hai sulla gravità della Terra (all'incirca intorno a 1G) e sugli assi spaziali, potrai usare le letture dell'accelerometro per capire quando il dispositivo è rivolto verso il basso e quando verso l'alto. Puoi anche usarlo per rilevare il movimento: prova a scuotere con cautela Sense HAT e Raspberry Pi e osserva i numeri: più forte viene scosso, maggiore è l'accelerazione.

Quando utilizzi `sense.get_accelerometer_raw()` stai dicendo a Sense HAT di spegnere gli altri due sensori dell'IMU (giroscopio e magnetometro) e di indicare esclusivamente i dati dall'accelerometro. Naturalmente si può fare la stessa cosa anche con gli altri sensori.

Trova la riga `acceleration = sense.get_accelerometer_raw()` e modificala come segue:

```
orientation = sense.get_gyroscope_raw()
```

Sostituisci la parola **acceleration** sulle tre righe con **orientation**. Fai clic su Run e vedrai l'orientamento di Sense HAT per tutti e tre gli assi, arrotondato al numero intero più vicino. A differenza dell'ultimo valore dell'orientamento, questa volta i dati provengono esclusivamente dal giroscopio ed escludono i valori di accelerometro o magnetometro. Questo può essere utile ad esempio per conoscere l'orientamento di un Sense HAT in movimento sul retro di un robot, senza che il movimento crei confusione, oppure se stai usando Sense HAT vicino a un forte campo magnetico.

Arresta il programma facendo clic sul pulsante rosso di arresto. Per utilizzare il magnetometro, cancella tutte le righe del programma eccetto le prime quattro, quindi digita quanto segue sotto la riga **while True**:

```
north = sense.get_compass()  
print(north)
```

Esegui il programma e vedrai la direzione del nord magnetico stampata ripetutamente nell'area di Python shell. Ruota con attenzione Sense HAT e vedrai il cambiamento di direzione mentre l'orientamento di Sense HAT relativo al nord si sposta: hai costruito una bussola! Se hai un magnete (anche una semplice calamita da frigo) prova a spostarlo intorno a Sense HAT per vedere come incide sulle letture del magnetometro.



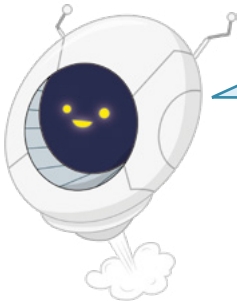
SFIDA: ROTAZIONE AUTOMATICA



Usando quello che hai imparato sulla matrice LED e sui sensori dell'unità di misura inerziale, riesci a scrivere un programma che fa ruotare un'immagine a seconda della posizione di Sense HAT?

Controllo joystick

Il joystick di Sense HAT, nell'angolo in basso a destra, è piccolo ma sorprendentemente potente: oltre ad essere in grado di riconoscere gli input da quattro direzioni (su, giù, sinistra e destra) ha anche un quinto input che si ottiene premendolo dall'alto come un interruttore a pulsante.



ATTENZIONE

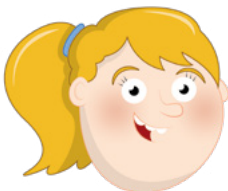


Il joystick Sense HAT deve essere utilizzato solo se sono stati montati i distanziatori, come descritto all'inizio di questo capitolo. Senza i distanziatori, spingendo verso il basso il joystick, la scheda Sense HAT potrebbe flettersi e danneggiarsi, danneggiando anche il terminale GPIO di Raspberry Pi.

Controllo joystick in Scratch

Avvia un nuovo programma in Scratch con l'estensione Sense HAT Raspberry Pi caricata. Inizia trascinando un blocco Situazioni **quando si clicca su** nell'area codice, trascina un blocco **clear display** sotto di esso, quindi trascina e modifica un blocco **set background to nero** e **set colour to bianco**.

In Scratch, il joystick Sense HAT mappa i tasti cursore della tastiera: spostare il joystick verso l'alto equivale a premere il tasto freccia su, verso il basso il tasto freccia giù, a sinistra il tasto freccia sinistra e a destra il tasto freccia destra, se invece lo premi verso l'interno come un interruttore a pulsante equivale a premere il tasto **INVIO**.

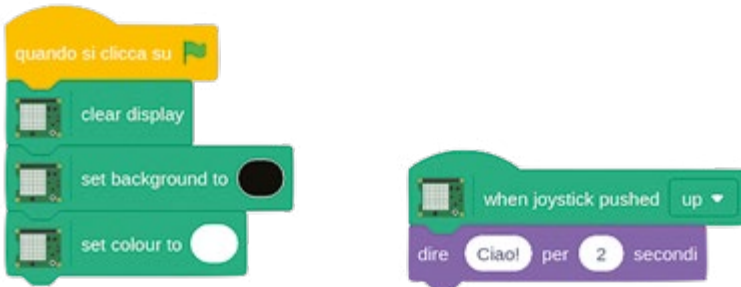


ATTENZIONE



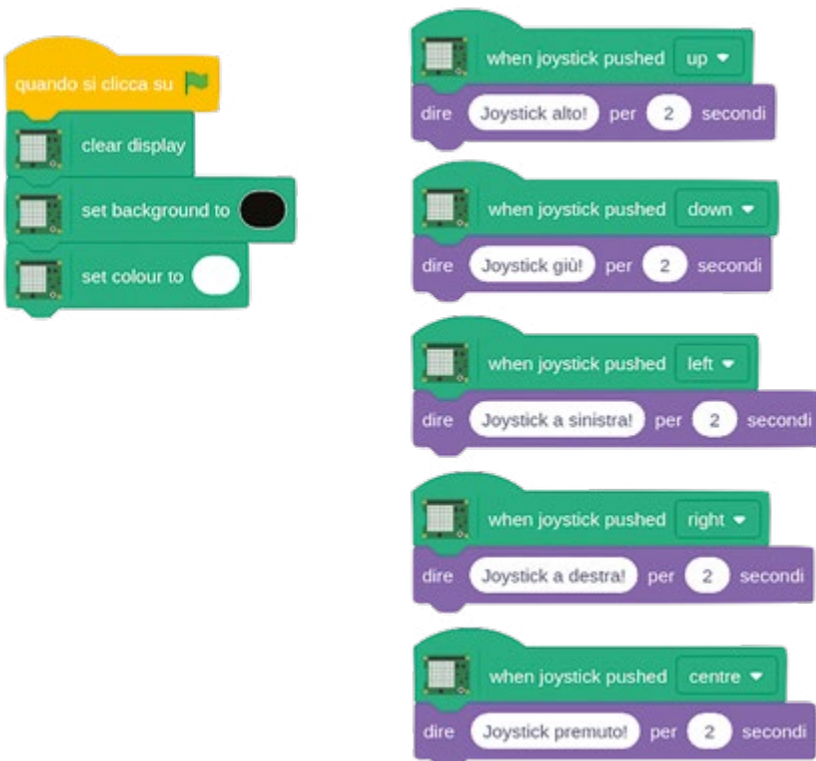
Il controllo del joystick è disponibile solo sul Sense HAT fisico. Quando utilizzi l'emulatore Sense HAT, usa i tasti corrispondenti sulla tastiera per simulare le pressioni del joystick.

Trascina un blocco **when joystick pushed up** nell'area codice. Quindi trascina un blocco **dire Ciao! per 2 secondi** sotto di esso.

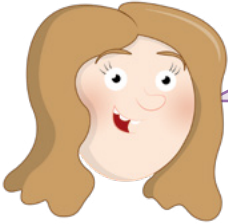


Porta il joystick verso l'alto e vedrai il gatto Scratch dire un allegro "Ciao!".

Successivamente modifica il blocco **dire Ciao! per 2 secondi** in un blocco **dire Joystick verso l'alto! per 2 secondi** e continua ad aggiungere i blocchi Situazioni e Aspetto fino a quando non hai aggiunto azioni per tutti e cinque modi in cui può essere mosso il joystick.



Prova a spostare il joystick nelle varie direzioni per visualizzare i tuoi messaggi.



SFIDA FINALE



Puoi usare il joystick Sense HAT per controllare uno sprite Scratch nell'area stage? Puoi fare in modo che se lo sprite raccoglie un altro sprite che rappresenta un oggetto il display LED Sense HAT mostri un simpatico messaggio?

Controllo joystick in Python

Avvia un nuovo programma in Thonny e salvalo come Joystick Sense HAT. Inizia con le consuete tre righe che consentono di impostare Sense HAT e cancellare eventuali pattern dalla matrice LED:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Quindi imposta un loop infinito:

```
while True:
```

Indica a Python di rilevare gli input dal joystick Sense HAT con la seguente riga, che verrà fatta rientrare automaticamente da Thonny:

```
for event in sense.stick.get_events():
```

Infine, aggiungi la seguente riga (verrà fatta rientrare da Thonny) per indicare le azioni da eseguire quando viene rilevata una pressione del joystick:

```
print(event.direction, event.action)
```

Fai clic su Run e prova a spostare il joystick nelle varie direzioni. Vedrai la direzione che hai scelto stampata nell'area Python shell: su, giù, sinistra, destra e al centro per quando hai premuto il joystick come un interruttore a pulsante.

Per ogni volta che premi il joystick vedrai due eventi: un evento **pressed** per quando si sposta la prima volta in una direzione e uno **released** per quando il joystick torna al centro. Puoi usarlo nei programmi: pensa a un personaggio in un gioco, che potrebbe iniziare a muoversi quando il joystick viene premuto in una direzione per fermarsi non appena viene rilasciato.

Puoi anche utilizzare il joystick per attivare funzioni, anziché limitarti a utilizzarlo per loop. Elimina le righe sotto a `sense.clear()` e digita quanto segue:

```
def rosso():
    sense.clear(255, 0, 0)

def blu():
    sense.clear(0, 0, 255)

def verde():
    sense.clear(0, 255, 0)

def giallo():
    sense.clear(255, 255, 0)
```

Queste funzioni modificano l'intera matrice di LED Sense HAT in un unico colore, rosso, blu, verde o giallo, così riuscirai a capire con facilità che il programma funziona. Per attivarle è necessario indicare a Python le funzioni correlate ai vari input del joystick. Digita quanto segue:

```
sense.stick.direction_up = rosso
sense.stick.direction_down = blu
sense.stick.direction_left = verde
sense.stick.direction_right = giallo
sense.stick.direction_middle = sense.clear
```

Infine, il programma ha bisogno di una ripetizione infinita (nota come ripetizione *principale*) per continuare a funzionare e controllare gli input dei joystick, anziché limitarsi a eseguire una volta il codice che si è scritto e terminare. Digita quanto segue:

```
while True:
    pass
```

Fai clic su Run, prova a muovere il joystick e vedrai i LED illuminarsi del colore scelto. Per spegnere i LED, premi il joystick come un pulsante: la direzione **middle** è impostata per attivare `sense.clear()` e spegnerli tutti. Complimenti ora puoi acquisire input dal joystick.



SFIDA FINALE



Puoi usare quello che hai imparato per disegnare un'immagine sullo schermo e farla ruotare in qualsiasi direzione il joystick venga spostato? Puoi usare l'input centrale per passare da un'immagine all'altra?

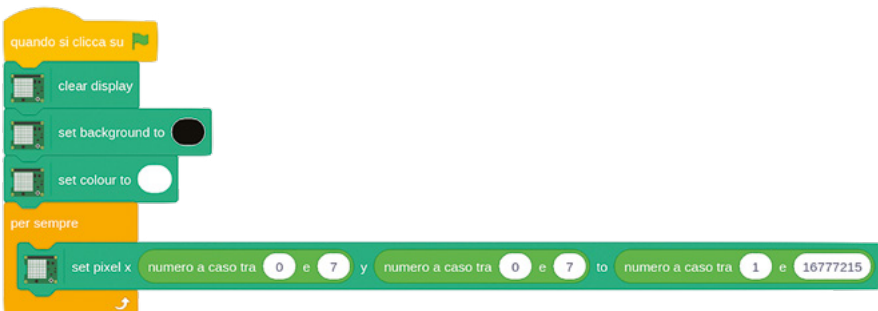
Progetto Scratch: luce scintillante con Sense HAT

Ora che conosci il funzionamento di Sense HAT, è il momento di creare un programma che emetta una luce scintillante in base alla temperatura: con una pulsazione rapida quando fa freddo e gradualmente più lenta mano a mano che diventa più caldo.

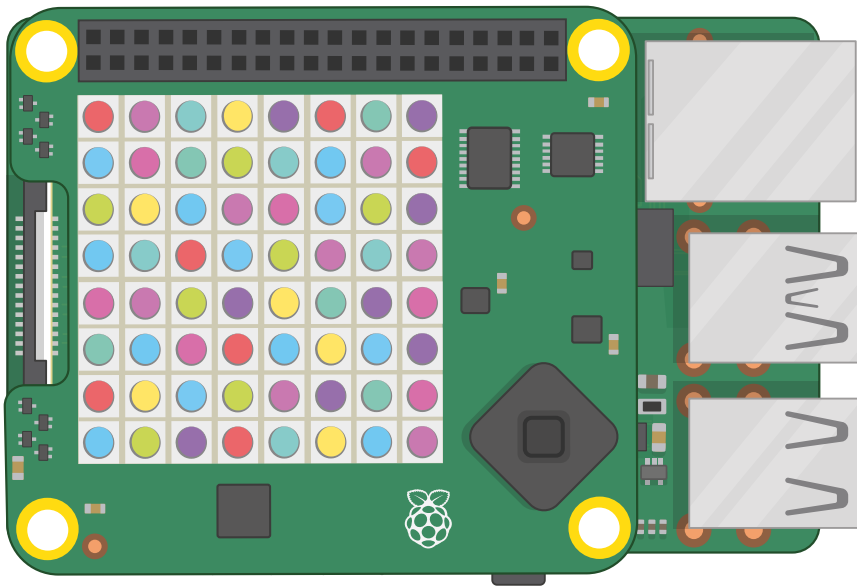
Crea un nuovo progetto in Scratch e aggiungi l'estensione Sense HAT Raspberry Pi se non è già presente. Come sempre, inizia con quattro blocchi: **quando si clicca su**, **clear display**, **set background to nero** e **set colour to bianco**, ricordandoti che dovrai cambiare i colori predefiniti.

Inizia creando uno scintillio semplice, ma artistico. Inizia trascinando un blocco **per sempre** nell'area codice, quindi completa con un blocco **set pixel x 0 y 0 to colour**. Anziché utilizzare i numeri impostati, completa ciascuna delle sezioni x, y e colore di quel blocco con un blocco Operatori **numero a caso tra 1 e 10**.

I valori da 1 a 10 non sono molto utili in questo caso, quindi è necessario fare qualche modifica. I primi due numeri del blocco **set pixel** sono le coordinate X e Y del pixel sulla matrice LED, il che significa che dovrebbero essere numeri compresi tra 0 e 7, quindi modifica i primi due blocchi in **numero a caso tra 0 e 7**. La sezione successiva è il colore con cui impostare il pixel. Quando utilizzi il selettore dei colori, il colore scelto viene mostrato direttamente nell'area dello script, internamente però i colori sono rappresentati da un numero, quindi puoi utilizzare quest'ultimo. Modifica l'ultimo blocco da scegliere casualmente in **numero a caso tra 0 e 16777215**.

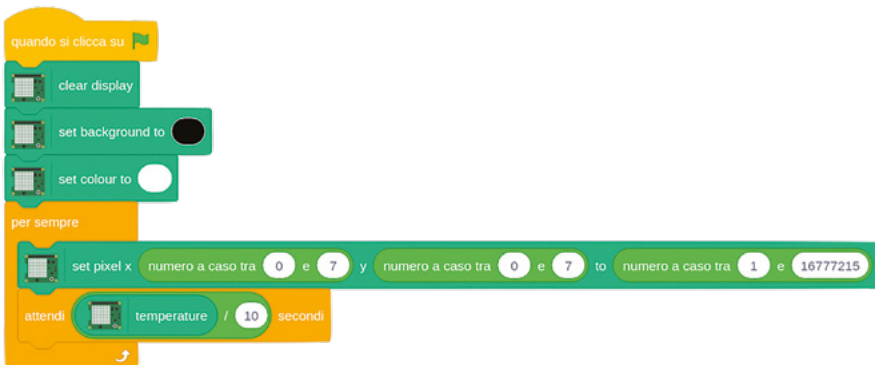


Fai clic sulla bandiera verde e vedrai i LED di Sense HAT cominciare a illuminarsi con colori casuali (**Figura 7-28**). Complimenti, hai realizzato una luce scintillante!



▲ Figura 7-28: accendere i pixel con colori casuali

Il programma ora non è molto interattivo. Per renderlo interattivo, trascina un blocco **attendi 1 secondi** in modo che sia sotto il blocco **set pixel**, ma all'interno del blocco **per sempre**. Trascina un blocco Operatore **● / ●** sopra l'1, quindi digita 10 nel secondo spazio. Infine, trascina un blocco **temperature** sul primo spazio del blocco Operatore di divisione.



Fai clic sulla bandiera verde e, a meno che tu non viva in un posto molto freddo, noterai che la scintilla è più lenta di prima. Questo perché hai creato un ritardo basato sulla temperatura: il programma ora aspetta *la temperatura attuale divisa per 10* secondi prima di ogni ciclo. Se la temperatura nella stanza è di 20 °C, il programma aspetterà 2 secondi prima di eseguire la ripetizione, mentre se la temperatura è di 10 °C, aspetterà 1 secondo e sotto i 10 °C meno di un secondo.

Se Sense HAT rileva una temperatura negativa (ossia al di sotto di 0 °C, il punto di congelamento dell'acqua) cercherà di aspettare meno di 0 secondi, ma siccome questo è impossibile senza viaggiare nel tempo, vedrai comunque lo stesso effetto, come se stesse aspettando 0 secondi. Complimenti: ora puoi integrare le varie funzioni di Sense HAT nei tuoi programmi!

Progetto Python: Tricorder con Sense HAT

Ora che conosci il funzionamento di Sense HAT, è il momento di creare un tricorder, un dispositivo dotato di sensori che i fan di un certo franchise fantascientifico riconosceranno subito.

Crea un nuovo progetto in Thonny e salvalo come **Tricorder**, quindi inserisci le righe per creare un programma Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Dovrai quindi avviare le funzioni per definire ciascun sensore Sense HAT. Inizia con l'unità di misura inerziale digitando:

```
def orientation():
    orientation = sense.get_orientation()
    pitch = orientation["pitch"]
    roll = orientation["roll"]
    yaw = orientation["yaw"]
```

Poiché i risultati del sensore vengono fatti scorrere attraverso i LED, arrotonda i risultati in modo da non dover attendere per visualizzare decine di decimali. Invece di visualizzare solo numeri interi, arrotonda a un decimale digitando quanto segue:

```
pitch = round(pitch, 1)
roll = round(roll, 1)
yaw = round(yaw, 1)
```

Infine dovrai indicare a Python di mostrare i risultati nei LED, in modo che il tricorder funzioni come un dispositivo portatile senza bisogno di essere collegato a un monitor o televisore:

```
sense.show_message("Beccheggio {0}, Rollio {1}, Imbardata {2}".format(pitch, roll, yaw))
```

Ora che disponi di una funzione completa per la lettura e la visualizzazione dell'orientamento dall'IMU, crea funzioni simili per ciascuno degli altri sensori. Inizia con il sensore di temperatura:


```
def temperature():
    temp = sense.get_temperature()
    temp = round(temp, 1)
    sense.show_message("Temperatura: %s gradi Celsius" % temp)
```

Guarda attentamente la riga che stampa il risultato sui LED: `%s` è un segnaposto e viene sostituito con il contenuto della variabile `temp`. In questo modo è possibile formattare l'output in modo comprensibile con un'etichetta, "Temperatura.", e un'unità di misura, "gradi Celsius".

Quindi definisci una funzione per il sensore di umidità:

```
def humidity():
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    sense.show_message("Umidità: %s percento" % humidity)
```

Poi il sensore di pressione:

```
def pressure():
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)
    sense.show_message("Pressione: %s millibar" % pressure)
```

Infine la lettura della bussola dal magnetometro:

```
def compass():
    for i in range(0, 10):
        north = sense.get_compass()
        north = round(north, 1)
        sense.show_message("Nord: %s gradi" % north)
```

La ripetizione breve `for` in questa funzione richiede dieci letture del magnetometro per avere dati sufficienti al fine di fornire un risultato accurato. Se ritieni che il valore riportato continui a cambiare, prova a estenderlo a 20, 30 o anche 100 ripetizioni per migliorare ulteriormente la precisione.

Il tuo programma ora ha cinque funzioni, ognuna delle quali riporta una lettura di uno dei sensori Sense HAT facendola scorrere sui LED. È però necessario un modo per scegliere il sensore da usare e il joystick è perfetto a questo scopo.

Digita quanto segue:

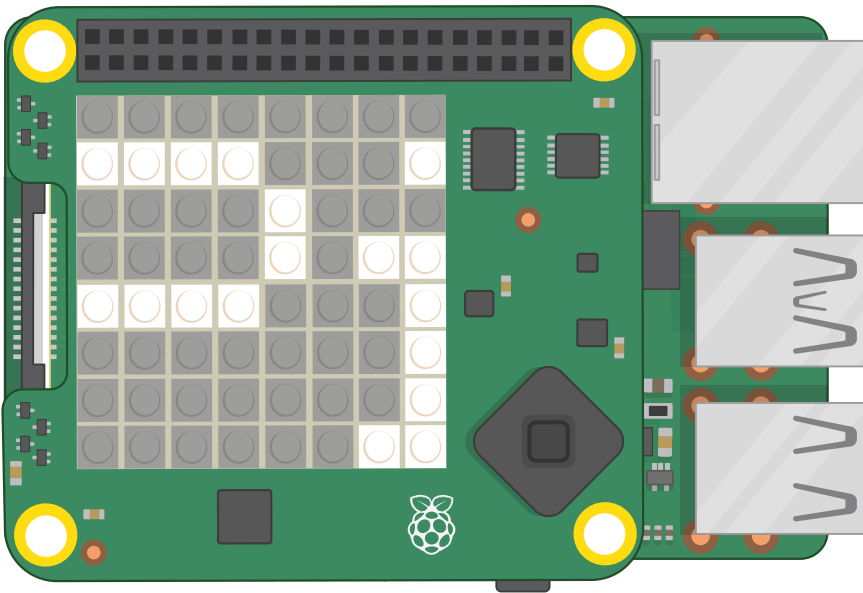
```
sense.stick.direction_up = orientation
sense.stick.direction_right = temperature
sense.stick.direction_down = compass
sense.stick.direction_left = humidity
sense.stick.direction_middle = pressure
```

Queste righe assegnano un sensore a ciascuna delle cinque possibili direzioni sul joystick: spostando verso l'alto legge dal sensore di orientamento, verso il basso dal magnetometro, verso sinistra dal sensore di umidità, verso destra dal sensore di temperatura e premendo al centro dal sensore di pressione.

Infine, è necessaria una ripetizione principale in modo che il programma continui ad attendere le pressioni del joystick e non si fermi immediatamente. Al termine del programma digita quanto segue:

```
while True:
    pass
```

Fai clic su Run e prova a spostare il joystick per effettuare una lettura da uno dei sensori (**Figura 7-29**). Quando ha finito di scorrere il risultato, spostalo in un'altra direzione. Complimenti: hai costruito un tricorder portatile che renderebbe orgogliosa la Federazione dei Pianeti!



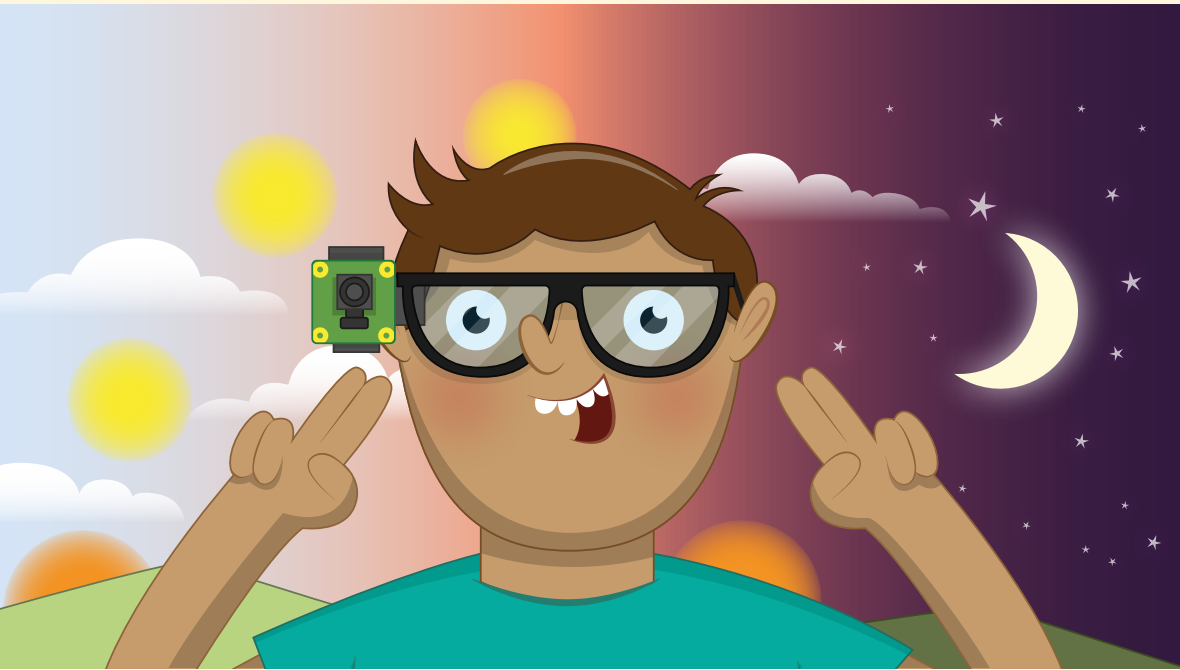
▲ **Figura 7-29:** Ogni lettura scorre sul display

Per altri progetti Sense HAT, dai un'occhiata ai link in **Appendice D Ulteriori letture**.

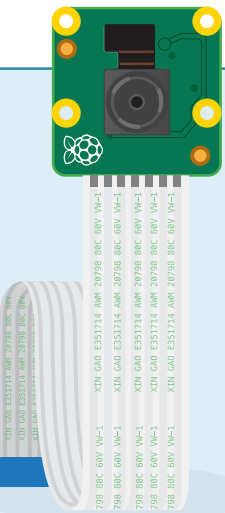
Capitolo 8

Fotocamera di Raspberry Pi

Con il Camera Module o la High Quality Camera potrai scattare foto e video in alta risoluzione e creare fantastici progetti di visione artificiale direttamente sul tuo Raspberry Pi



Se desideri realizzare contenuti visivi, conosciuti nel campo della robotica come *visione artificiale*, allora il Camera Module (modulo fotocamera) opzionale o la nuova High Quality Camera (fotocamera ad alta qualità) Raspberry Pi sono un'ottima base di partenza. Il Camera Module / HQ Camera è un piccolo circuito stampato quadrato con un sottile cavo a nastro; collegalo alla porta dell'interfaccia seriale della fotocamera (CSI) sul tuo Raspberry Pi (non disponibile per Raspberry Pi 400) per avere immagini fisse ad alta risoluzione e segnali video in movimento che possono essere utilizzati così come sono o integrati nei tuoi programmi.



TIPI DI FOTOCAMERA

Sono disponibili tre tipi di fotocamera Raspberry Pi: il Camera Module standard, la versione 'NoIR', e la High Quality (alta qualità) Camera. Se desideri scattare foto e video standard in ambienti ben illuminati, ti consigliamo il Camera Module standard, mentre se vuoi utilizzare obiettivi speciali e vuoi la migliore qualità dell'immagine, ti consigliamo di optare per la HQ Camera. Il Camera Module NoIR (chiamato così perché non ha un filtro a infrarossi o IR), è progettato per essere usato con sorgenti di luce infrarossa per scattare foto e video al buio. Se vuoi realizzare una cassetta per uccelli, una telecamera di sicurezza o altri progetti che prevedono la visione notturna, opta per la versione NoIR, ma ricordati di acquistare una sorgente di luce infrarossa.

I Camera Modules standard e NoIR Raspberry Pi si basano su un sensore d'immagine Sony IMX219 *da 8 megapixel*, il che significa che può scattare foto con un massimo di 8 milioni di pixel. Lo fa catturando immagini fino a 3280 pixel di larghezza per 2464 di altezza. Oltre alle immagini fisse, il Camera Module è in grado di acquisire filmati con risoluzione Full HD a una velocità di 30 fotogrammi al secondo (30 fps). Per un movimento più fluido o anche per creare un effetto a rallentatore, la fotocamera può essere impostata per acquisire con una frequenza di fotogrammi più elevata abbassando la risoluzione: 60fps per video a 720p e fino a 90 fps per 480p (VGA).

La High Quality Camera utilizza un sensore Sony IMX477 da 12,3 megapixel, più grande di quello dei moduli standard e NoIR quindi in grado di fare entrare più luce, per immagini di qualità superiore. A differenza dei moduli fotocamera, la fotocamera HQ non include un obiettivo e senza di esso non potrai scattare foto o video. Puoi utilizzare qualsiasi obiettivo con un attacco C o CS oppure altri obiettivi dotati di un adattatore appropriato per tale attacco. Per dettagli su come innestare una lente, vedi l'**Appendice F: Impostazione della High Quality Camera**.

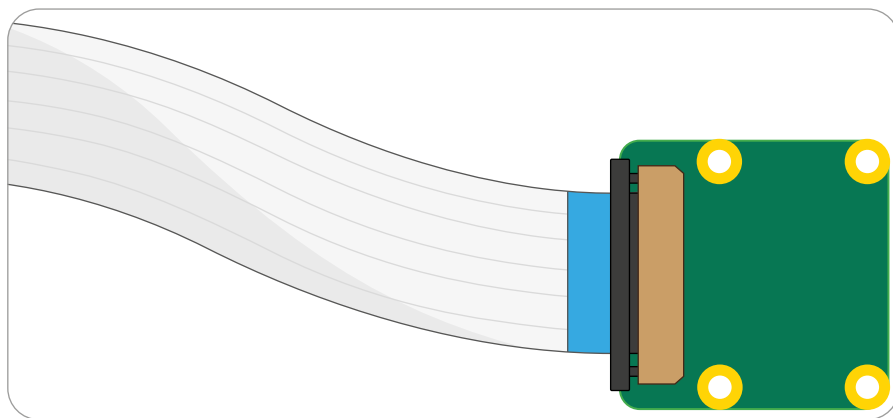
RASPBERRY PI 400

Purtroppo i Camera Modules Raspberry Pi non sono compatibili con Raspberry Pi 400. Puoi utilizzare una webcam USB, ma non potrai servirti degli strumenti software specifici per il Camera Module Raspberry Pi inclusi nel sistema operativo.

Come installare la fotocamera

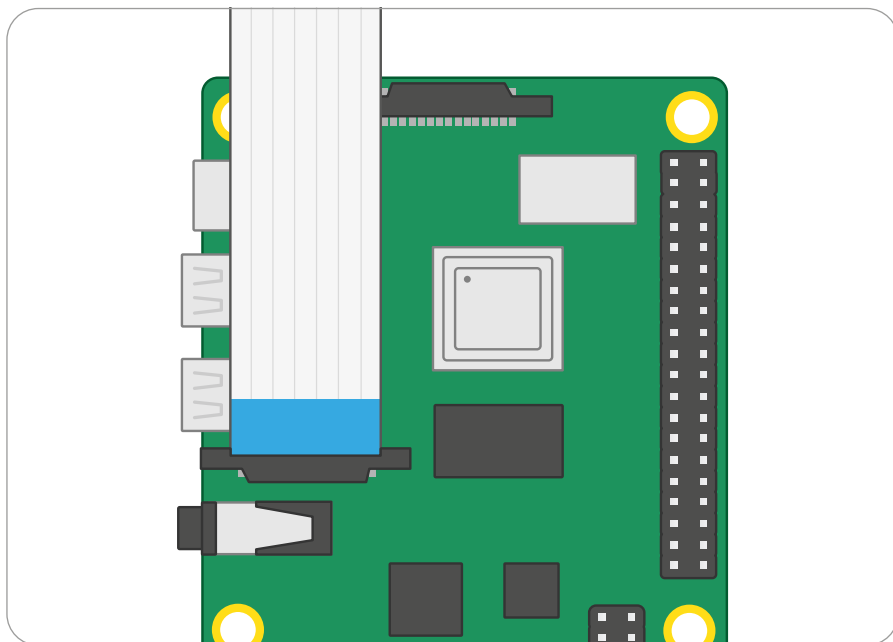
Come qualsiasi componente hardware, il Camera Module o la HQ Camera deve essere collegato o scollegato da Raspberry Pi solo quando l'alimentazione è spenta e il cavo di alimentazione è scollegato. Se il tuo Raspberry Pi è acceso, scegli Arresta dal menu Raspberry, attendi lo spegnimento, quindi scollega l'alimentazione.

Nella maggior parte dei casi, il cavo a nastro in dotazione sarà già collegato al Camera Module o alla HQ Camera; in caso contrario, capovolgi la scheda della fotocamera in modo che il sensore sia sulla parte inferiore e individua il connettore piatto in plastica. Afferra con cautela i bordi sporgenti con le unghie e tira verso l'esterno fino a estrarre parzialmente il connettore. Fai scorrere il cavo a nastro con i bordi argentati verso il basso e la plastica blu rivolta verso l'alto sotto alla linguetta appena estratta, quindi spingi delicatamente la linguetta in posizione fino a sentire un clic (**Figura 8-1**). Non importa quale estremità del cavo utilizzi. Se il cavo è installato correttamente, sarà dritto e non si scollegherà se viene tirato leggermente; in caso contrario, estrai la linguetta e riprova.



▲ **Figura 8-1:** Collegamento del cavo a nastro al Camera Module

Installa l'altra estremità del cavo nello stesso modo. Individua la porta della fotocamera (o CSI) su Raspberry Pi e tira delicatamente la linguetta verso l'alto. Se Raspberry Pi è installato in un case, rimuovilo. Posiziona Raspberry Pi con la porta HDMI rivolta verso di te, fai scivolare il cavo a nastro in modo che i bordi argentati si trovino alla tua sinistra e la plastica blu a destra (**Figura 8-2**), quindi premi delicatamente la linguetta per farla tornare in posizione. Se il cavo è installato correttamente, sarà dritto e non si scollegherà se viene tirato leggermente; in caso contrario, estrai la linguetta e riprova.



▲ **Figura 8-2:** collegamento del cavo a nastro alla porta fotocamera/CSI su Raspberry Pi

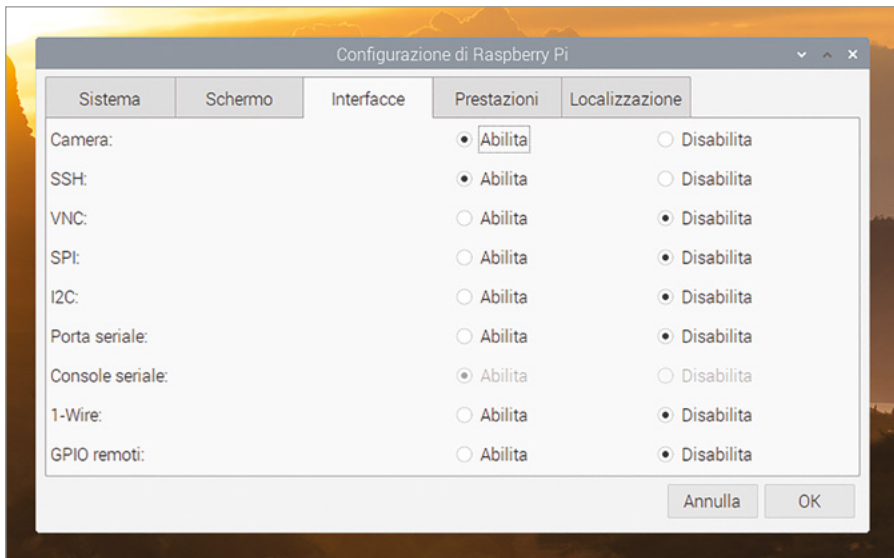
Il Camera Module presenta un piccolo pezzo di plastica blu che copre l'obiettivo per proteggerlo dai graffi durante la produzione, la spedizione e l'installazione. Individua la piccola linguetta di plastica e rimuovila delicatamente dall'obiettivo per iniziare a utilizzare la fotocamera.



REGOLAZIONE DELLA MESSA A FUOCO

Il Camera Module viene solitamente fornito con una rotellina di plastica per la regolazione della messa a fuoco dell'obiettivo. La messa a fuoco predefinita è generalmente perfetta, tuttavia se utilizzi la fotocamera per lavori molto ravvicinati puoi far scorrere la rotella sull'obiettivo e ruotarla delicatamente per regolare manualmente la messa a fuoco. Per la messa a fuoco della fotocamera HQ, dai un'occhiata all'[Appendice F](#).

Collega l'alimentatore al Raspberry Pi e fai caricare Raspberry Pi OS. Prima di poter utilizzare la fotocamera, è necessario indicarla come collegata a Raspberry Pi: apri il menu facendo clic sull'icona a forma di lampone, seleziona la categoria Preferenze e fai clic su Configurazione di Raspberry Pi. Quando lo strumento è stato caricato, fai clic sulla scheda Interfacce, cerca la voce Camera nell'elenco e fai clic sul pulsante rotondo a sinistra di "Abilita" per attivarlo (**Figura 8-3**, sul retro). Fai clic su OK e lo strumento ti chiederà di riavviare Raspberry Pi. Una volta riavviato, la fotocamera sarà pronta per l'uso!

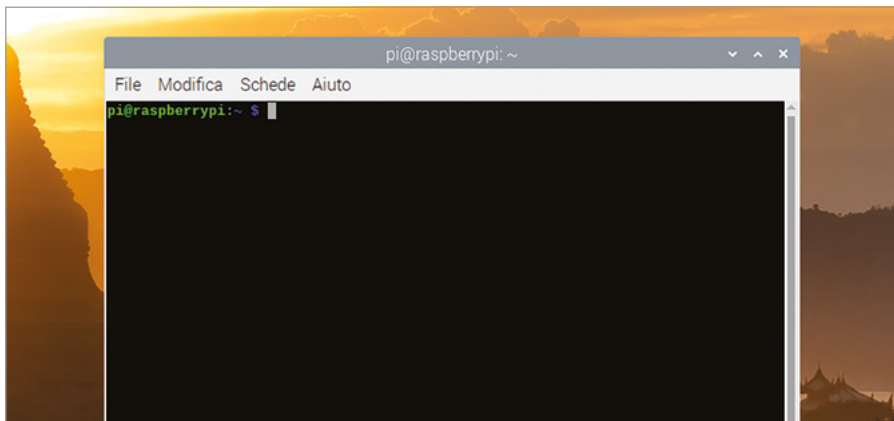


▲ **Figura 8-3:** abilitare la fotocamera nella configurazione Raspberry Pi

Come verificare il funzionamento della fotocamera

Per verificare che il Camera Module o la HQ Camera sia installato correttamente e che sia stata abilitata l'interfaccia nello strumento Configurazione di Raspberry Pi, è possibile utilizzare lo strumento **raspistill**. Questo, insieme a **raspivid** per i video, è progettato per acquisire immagini dalla fotocamera mediante l'*interfaccia a riga di comando (CLI)* Raspberry Pi.

A differenza dei programmi usati finora, non troverai raspistill nel menu. Fai clic sull'icona a forma di lampone per caricare il menu, seleziona la categoria Accessori, quindi fai clic su LXTerminal. Visualizzerai una finestra nera con una scritta verde e blu (**Figura 8-4**): questo è il *terminale*, che consente di accedere all'interfaccia a riga di comando.

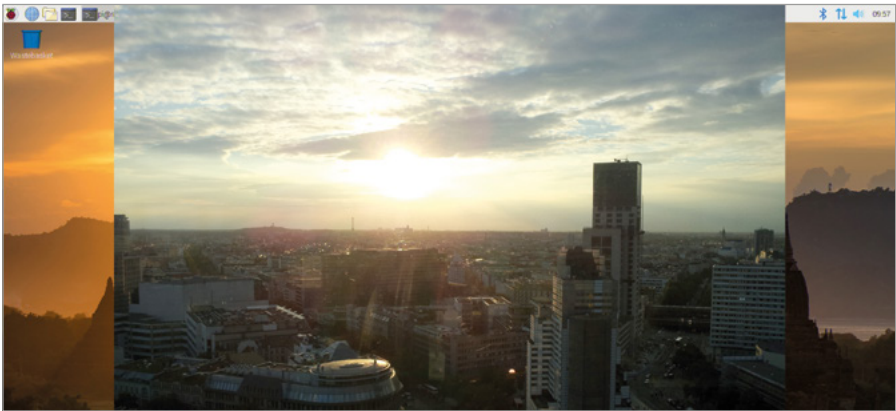


▲ **Figura 8-4:** Apri una finestra LXTerminal per inserire i comandi

Per testare la fotocamera, digita quanto segue in LXTerminal:

```
raspistill -o test.jpg
```

Non appena premi il tasto **INVIO**, vedrai apparire sullo schermo un'immagine di ciò che la fotocamera vede (**Figura 8-5**). Questa è una *live view* e, a meno che tu non fornisca indicazioni diverse a raspistill, durerà 5 secondi. Trascorsi i 5 secondi, la fotocamera scatterà una foto e la salverà nella cartella Home con il nome **test.jpg**. Se vuoi acquisire un'altra immagine, digita nuovamente lo stesso comando, ma assicurati di cambiare il nome del file di output dopo il comando **-o** o sovrascriverai la prima foto.



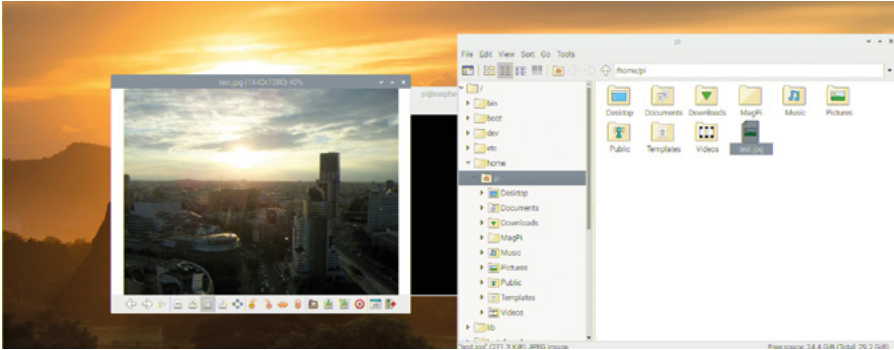
▲ **Figura 8-5: live view della fotocamera**

Se la live view era girata, dovrai indicare a raspistill che la fotocamera è ruotata. Il Camera Module è progettato perché il cavo a nastro esca dal bordo inferiore; se esce dai lati o dalla parte superiore, come con alcuni accessori di terze parti per l'attacco, è possibile ruotare l'immagine di 90, 180 o 270 gradi utilizzando l'opzione **-rot**. Per una fotocamera montata con il cavo che esce dalla parte superiore è sufficiente utilizzare il seguente comando:

```
raspistill -rot 180 -o test.jpg
```

Utilizza un valore di rotazione di 90 gradi se il cavo a nastro esce dal bordo destro e di 270 gradi se esce dal sinistro. Se l'immagine originale presentava un'angolazione sbagliata, prova a correggerla usando l'opzione **-rot**.

Per visualizzare la tua foto, apri il File Manager dalla categoria Accessori nel menu a forma di lampone: l'immagine che hai acquisito, chiamata **test.jpg**, è salvata nella cartella **home/pi**. Cerca nell'elenco dei file, quindi fai doppio clic per caricarla in un visualizzatore di immagini (**Figura 8-6**). Puoi anche allegare l'immagine alle e-mail, caricarla su siti Web tramite il browser o trascinarla in un dispositivo di archiviazione esterno.



▲ **Figura 8-6:** apertura dell'immagine acquisita

Cos'è picamera

Con Python e la pratica libreria picamera puoi controllare in modo flessibile il Camera Module o la HQ Camera. In questo modo hai il pieno controllo sulle funzioni di anteprima e acquisizione di immagini e video della fotocamera e puoi integrarle nei tuoi programmi, anche combinandole con programmi che utilizzano il modulo GPIO con la libreria GPIO Zero.



PROGRAMMAZIONE PYTHON

I progetti di questo capitolo presuppongono che tu conosca il linguaggio di programmazione Python, Thonny IDE, nonché i pin GPIO di Raspberry Pi. Se non l'hai già fatto, leggi il **Capitolo 5, Come programmare con Python** oppure il **Capitolo 6, Physical computing con Scratch e Python**.

Chiudi LXTerminal, se è ancora aperto, facendo clic sulla X nella parte superiore destra della finestra, quindi carica Thonny dalla categoria Programmazione del menu a forma di lampone. Salva il tuo nuovo progetto come **Fotocamera**, quindi inizia a importare le librerie di cui il programma ha bisogno digitando quanto segue nell'area dello script:

```

from picamera import PiCamera
from time import sleep
camera = PiCamera()

```

L'ultima riga consente di controllare il modulo della fotocamera o la fotocamera HQ utilizzando la funzione **camera**. Per iniziare, digita quanto segue:

```

camera.start_preview()
sleep(10)
camera.stop_preview()

```

Fai clic su Run e il desktop verrà nascosto, al suo posto vedrai un'anteprima a schermo intero di tutto ciò che viene visualizzato dalla fotocamera (**Figura 8-7**). Prova a spostarla o ad agitare la mano davanti all'obiettivo e vedrai l'immagine sullo schermo cambiare. Dopo 10 secondi l'anteprima si chiuderà e il programma terminerà, ma, a differenza dell'anteprima di raspistill, non verrà salvata nessuna immagine.



▲ **Figura 8-7:** una live view a tutto schermo della vista della fotocamera

Se l'orientamento dell'anteprima era sbagliato, è possibile ruotare l'immagine. Sotto alla riga **camera = PiCamera()**, digita:

```

camera.rotation = 180

```

Se l'anteprima era ruotata, la riga rimetterà le cose a posto. Come con raspistill, **camera.rotation** consente di ruotare l'immagine di 90, 180 o 270 gradi, a seconda che il cavo esca dal lato destro, superiore o sinistro del Camera Module. Ricordati di usare **camera.rotation** all'inizio di qualsiasi programma che scrivi per evitare di acquisire immagini o video ruotati erroneamente.

Acquisizione di immagini

Per acquisire un'immagine anziché mostrare solo l'anteprima, è necessario modificare il programma. Inizia riducendo il ritardo per l'anteprima: individua la riga **sleep(10)** e modificala in:

```
sleep(5)
```

Quindi aggiungi quanto segue:

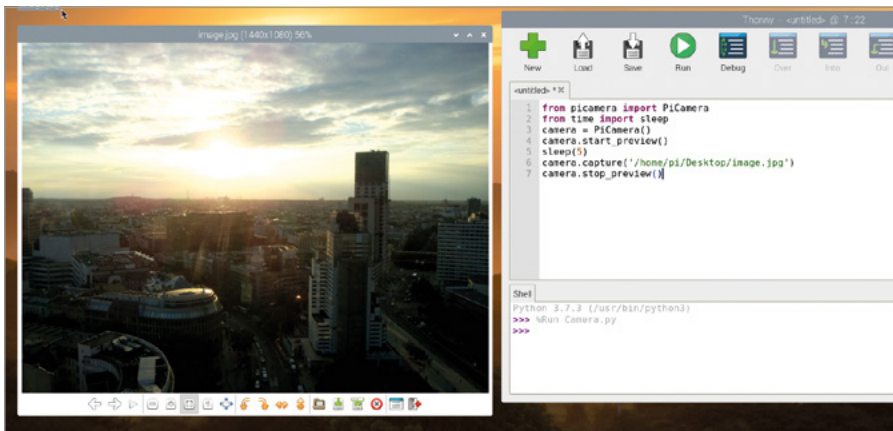
```
camera.capture('/home/pi/Desktop/image.jpg')
```



È ORA DI REGOLARE

Quando la fotocamera è in modalità anteprima, analizza il video per vedere se deve regolare le impostazioni per ottenere la migliore qualità. Lo vedrai se ti trovi in un ambiente molto buio o luminoso: in un primo momento sarà difficile vedere qualcosa nell'anteprima, poi diventerà più nitida. Per dare alla telecamera il tempo di regolarsi, aggiungi sempre un'anteprima di almeno 2 secondi al programma prima di acquisire un'immagine.

La funzione **camera.capture** dice a Python di salvare un fermo immagine e ha bisogno di sapere non solo come deve essere chiamata l'immagine, ma anche in quale cartella deve essere salvata. In questo esempio la salverai sul desktop; si trova appena sotto il Cestino. Se la finestra di Thonny è d'incalcio, ti sarà sufficiente fare clic e trascinarla sulla barra del titolo per spostarla. Fai doppio clic sul file per visualizzare l'immagine acquisita (**Figura 8-8**). Complimenti: hai programmato una fotocamera!



▲ **Figura 8-8:** apertura dell'immagine acquisita

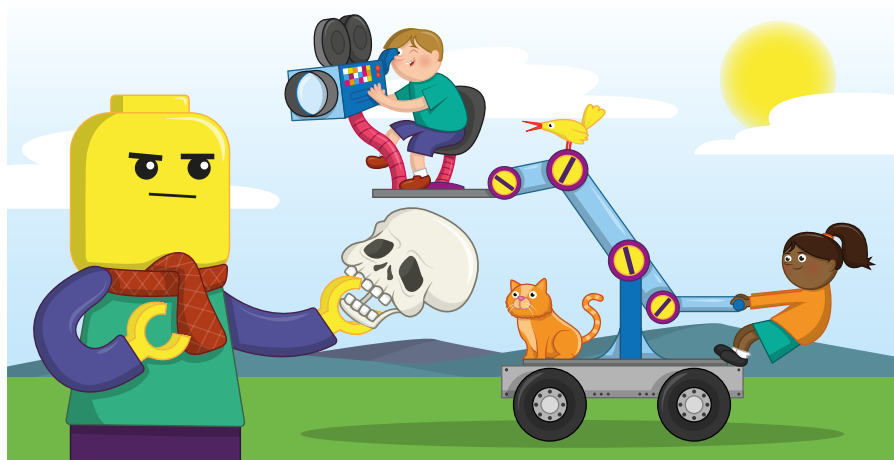
Acquisizione di video

Oltre a scattare fotografie, è possibile acquisire video. Elimina tutto ciò che si trova tra le righe `camera.start_preview()` e `camera.stop_preview()`, quindi digita quanto segue sotto a `camera.start_preview()`:

```
camera.start_recording('/home/pi/Desktop/video.h264')
sleep(10)
camera.stop_recording()
```

Verrà visualizzata l'anteprima della fotocamera, come prima, ma questa volta sarà anche memorizzata su un file sul desktop. Attendi i 10 secondi che hai indicato a Python, magari fai un balletto davanti alla fotocamera per rendere il video interessante; quando l'anteprima si chiuderà, troverai il file video sul desktop.

Per riprodurre il video, fai doppio clic sul file **video.h264** sul desktop. Il video inizierà ad essere riprodotto e se hai fatto un balletto lo vedrai! Al termine del video, il software del lettore si chiude con un simpatico messaggio in LXTerminal. Complimenti: ora sai come acquisire video utilizzando il Camera Module di Raspberry Pi o la fotocamera HQ!



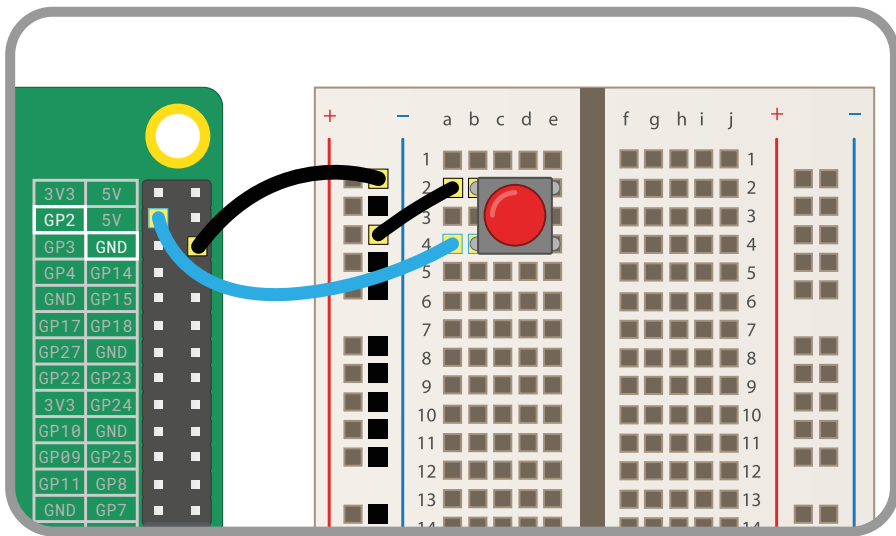
Animazione in stop-motion con pulsante

Utilizzando ciò che hai appreso in questo capitolo e, per quanto riguarda il collegamento dell'hardware al terminale GPIO Raspberry Pi, nel **Capitolo 6** sul **Physical computing**, è il momento di costruire qualcosa di speciale: il tuo personale studio di animazione in stop-motion.

L'animazione in stop-motion si ottiene scattando numerose foto di oggetti fermi, come modellini di auto o action figure, spostandoli leggermente tra una foto e l'altra. Anche se gli oggetti non si muovono mai in nessuna delle immagini, se li mostri uno dopo l'altro abbastanza velocemente sembrerà che si muovano alla velocità impostata.

Per questo progetto, avrai bisogno di un interruttore a pulsante, una breadboard, un cavo jumper maschio-maschio (M2M) e un paio di cavi jumper maschio-femmina (M2F). Se non si dispone di una breadboard puoi collegare l'interruttore utilizzando cavi femmina-femmina (F2F), ma sarà più difficile da premere. Se hai bisogno di rinfrescare la memoria su questi componenti, dai un'occhiata al **Capitolo 6, Physical computing con Scratch e Python**. Avrai anche bisogno di oggetti da animare: puoi usare quello che preferisci, da una figura in argilla a una macchinina o action figure.

Inizia creando il circuito: aggiungi il pulsante alla breadboard, quindi collega il conduttore di terra della breadboard a un polo di messa a terra su Raspberry Pi (contrassegnato con GND nella **Figura 8-9**) con un cavo jumper maschio-femmina. Utilizza un cavo jumper maschio-maschio per collegare un connettore dell'interruttore al conduttore di terra sulla breadboard, quindi un cavo jumper maschio-femmina per collegare l'altro conduttore dell'interruttore al pin GPIO 2 (contrassegnato con GP2 nella **Figura 8-9**).



▲ **Figura 8-9:** schema di collegamento per il collegamento di un pulsante ai pin GPIO

Crea un nuovo progetto in Thonny e salvalo come **Stop Motion**. Inizia importando e impostando le librerie necessarie per l'utilizzo della fotocamera e della porta GPIO:

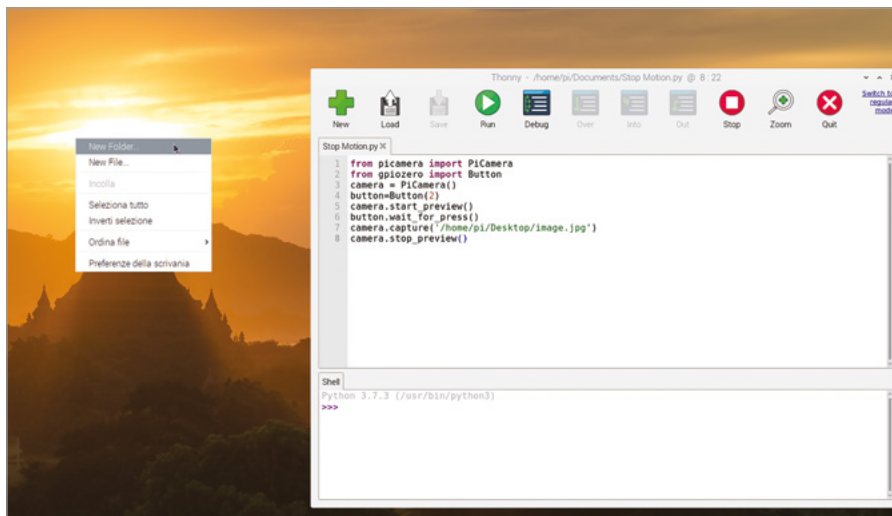
```
from picamera import PiCamera
from gpiozero import Button
camera = PiCamera()
button = Button(2)
```

Successivamente digita quanto segue:

```
camera.start_preview()
button.wait_for_press()
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Fai clic su Run e vedrai un'anteprima di ciò che la tua fotocamera sta puntando. L'anteprima rimarrà sullo schermo fino a quando non premi l'interruttore a pulsante: premilo ora e l'anteprima si chiuderà dopo che il programma avrà salvato l'immagine sul desktop. Trova l'immagine, chiamata **image.jpg**, e fai doppio clic per aprirla e verificare che il programma funzioni.

L'animazione in stop-motion prevede la creazione di molte immagini fisse per dare l'impressione di movimento quando vengono utilizzate in sequenza. Salva le immagini in una cartella in modo da non occupare tutto il desktop. Fai clic con il tasto destro del mouse su un punto qualsiasi del desktop dove non ci siano già file o icone, quindi scegli Crea nuovo e Cartella (**Figura 8-10**). Chiama la cartella **animation**, il tutto in lettere minuscole, quindi fare clic sul pulsante OK.



◀ **Figura 8-10:** creazione di una nuova cartella per le immagini acquisite

Per non dover riavviare il programma ogni volta che acquisisci un'immagine per l'animazione, dovrai fare in modo che venga eseguito in ripetizione. A differenza delle ripetizioni precedenti, questa deve chiudersi al momento giusto, altrimenti se fermi il programma durante la visualizzazione dell'anteprima della fotocamera, non sarai più in grado di vedere il desktop. Per fare questo, dovrai utilizzare due istruzioni speciali: **try** ed **except**.

Inizia cancellando le righe dopo `camera.start_preview()`, e scrivi:

```
frame = 1
```

Creerai così una nuova variabile, **frame**, che verrà utilizzata dal programma per memorizzare il numero di fotogramma corrente. La userai a breve per assicurarti di salvare un nuovo file ogni volta, altrimenti sovrascriverai la prima immagine ogni volta che premi il pulsante.

Successivamente imposta la ripetizione digitando:

```
while True:
    try:
```

La nuova istruzione **try** indica a Python di eseguire qualsiasi codice si trovi all'interno (che sarà il codice per acquisire le immagini). Digita:

```
        button.wait_for_press()
        camera.capture('/home/pi/Desktop/animation/frame%03d.jpg' % frame)
        frame += 1
```

Ci sono un paio di trucchi in queste tre righe di codice. Il primo è nel nome del file acquisito: utilizzando `%03d` Python aggiungerà davanti alla cifra alcuni zeri, in modo da formare un numero a tre cifre. Così "1" diventerà "001", "2" "002" e "10" "010". In questo modo i file verranno mantenuti nell'ordine corretto e non verranno sovrascritti.

`% frame` al termine della riga indica a Python di usare il numero della variabile del fotogramma nel nome del file. Per assicurarti che ogni file sia univoco, l'ultima riga (`frame += 1`) incrementa la variabile `frame` di 1. La prima volta che premi il pulsante, **frame** passerà da 1 a 2, in seguito da 2 a 3 e così via.

Al momento, però, il codice non uscirà in modo pulito quando hai finito di scattare le foto. Per far sì che ciò avvenga, è necessario un **except** per **try**. Digita quanto segue, ricordandoti di rimuovere un livello di rientranza sulla prima riga in modo che Python sappia che non fa parte della sezione **try**:

```
except KeyboardInterrupt:
    camera.stop_preview()
    break
```

Il programma finito avrà questa struttura:


```

from picamera import PiCamera
from time import sleep
from gpiozero import Button
camera = PiCamera()
button = Button(2)
camera.start_preview()
frame = 1
while True:
    try:
        button.wait_for_press()
        camera.capture('/home/pi/Desktop/animation/frame%03d.jpg'
% frame)
        frame += 1
    except KeyboardInterrupt:
        camera.stop_preview()
        break

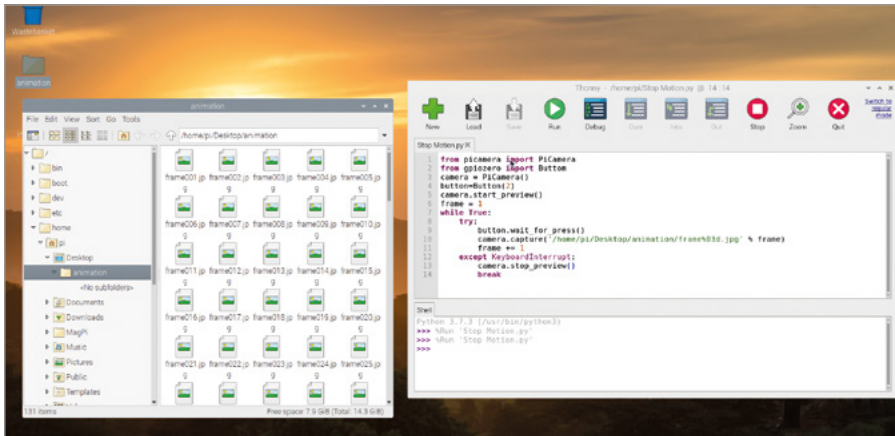
```

Prova a fare clic su Run, ma anziché premere il pulsante, premi i tasti **CTRL** e **C** della tastiera. Non è necessario premere entrambi i tasti contemporaneamente: basta tenere premuto **CTRL**, premere e rilasciare **C**, quindi rilasciare **CTRL**. Questi due tasti agiscono come un'interruzione e dicono a Python di smettere di fare quello che sta facendo. Senza **except KeyboardInterrupt**: Python terminerebbe immediatamente e lascerebbe l'anteprima della fotocamera sullo schermo; con quella riga invece Python esegue qualsiasi codice sia all'interno: in questo caso, il codice che gli indica di terminare l'anteprima della fotocamera e uscire.

Ora sei pronto per iniziare ad acquisire immagini per l'animazione in stop-motion! Posiziona il Camera Module o la fotocamera HQ in modo da inquadrare gli oggetti da animare e assicurati che non si muova, poiché se si muove rovina l'effetto. Posiziona gli oggetti nelle posizioni di partenza, quindi fai clic su Esegui per avviare il programma. Controlla l'anteprima e premi il pulsante per acquisire il primo fotogramma.

Sposta leggermente gli oggetti (meno li sposti tra un fotogramma e l'altro, più fluida sarà l'animazione finita) e premi di nuovo il pulsante per acquisire un altro fotogramma. Continua fino alla fine dell'animazione: più fotogrammi acquisisci, più lunga sarà l'animazione.

Quando hai finito, premi **CTRL+C** per chiudere il programma, quindi fai doppio clic sulla cartella **animation** sul desktop per vedere le immagini acquisite (**Figura 8-11**, sul retro). Fai doppio clic su qualsiasi immagine per aprirla e vederla più dettagliatamente!



▲ **Figura 8-11: immagini acquisite nella cartella**

Al momento però tutto quello che hai è una cartella piena di immagini fisse. Per creare un'animazione, è necessario trasformarle in un video. Fai clic sull'icona a forma di lampone per caricare il menu, seleziona la categoria Accessori e fai clic su LXTerminal. Si apre un'interfaccia a riga di comando, trattata in dettaglio nell'**Appendice C**, che consente di digitare comandi per Raspberry Pi. Quando LXTerminal si carica, per prima cosa cambia cartella accedendo a quella che hai creato. Per farlo digita:

```
cd Desktop/animation
```

È importante che la "D" di "Desktop" sia in maiuscolo, infatti Raspberry Pi OS *distingue tra maiuscole e minuscole*, il che significa che se non si digita un comando o un nome di cartella esattamente come è stato scritto in origine non funzionerà. Una volta cambiata cartella, digita:

```
ffmpeg -i frame%03d.jpg -r 10 animation.h264
```

Viene usato un programma chiamato **ffmpeg**, che prende le immagini fisse nella cartella e le converte in un video chiamato **animation.h264**. Nota: se ffmpeg non è disponibile, è possibile installarlo con **sudo apt-get install ffmpeg**. Questo processo può richiedere alcuni minuti, a seconda del numero di fotogrammi acquisiti; saprai che è finito quando vedrai riapparire il prompt di LXTerminal.

Per riprodurre il video, cerca il file **animation.h264** nella cartella **animation** e fai doppio clic su di esso per aprirlo. In alternativa, è possibile riprodurlo da LXTerminal digitando quanto segue:

```
omxplayer animation.h264
```

Una volta che il video è stato caricato vedrai l'animazione in stop-motion prendere vita. Congratulazioni: hai trasformato Raspberry Pi in un potente studio di animazione!

Se l'animazione si muove troppo velocemente o lentamente, cambia la parte **-r 10** del comando **ffmpeg** inserendo un numero inferiore o superiore: questa è la frequenza di fotogrammi, ovvero il numero di immagini fisse in un secondo di video. Un numero più basso renderà l'animazione più lenta ma meno fluida, mentre con un numero più alto sarà più fluida, ma molto più rapida.

Se vuoi salvare il video, assicurati di trascinarlo dal desktop nella cartella Video, altrimenti alla prossima esecuzione del programma sovrascriverai il file!

Impostazioni avanzate della fotocamera

Se necessiti di maggiore controllo sul Camera Module o HQ Camera Raspberry Pi, è possibile utilizzare la libreria Python `picamera` per accedere a varie impostazioni. Di seguito sono indicate queste impostazioni, insieme ai valori predefiniti, così potrai includerle nei tuoi programmi.

camera.awb_mode = 'auto'

Imposta la modalità automatica di bilanciamento del bianco della fotocamera su una delle seguenti modalità: **off**, **auto**, **sunlight**, **cloudy**, **shade**, **tungsten**, **fluorescent**, **incandescent**, **flash** o **horizon**. Se le tue foto o video sono un po' troppo blu o gialli prova con una modalità diversa.

camera.brightness = 50

Imposta la luminosità dell'immagine della fotocamera, da più scura (a 0) a più luminosa (a 100).

camera.color_effects = None

Cambia l'effetto di colore attualmente in uso nella fotocamera. Normalmente questa impostazione non dovrebbe essere modificata, ma se si fornisce una coppia di numeri è possibile modificare il modo in cui la fotocamera registra il colore: prova **(128, 128)** per creare un'immagine in bianco e nero.

camera.contrast = 0

Imposta il contrasto dell'immagine. Un numero più alto farà sembrare l'immagine più definita, mentre un numero più basso la farà sembrare più sbiadita. Puoi utilizzare qualsiasi numero compreso tra -100 per il contrasto minimo e 100 per il contrasto massimo.

camera.crop = (0.0, 0.0, 1.0, 1.0)

Questa riga consente di ritagliare l'immagine rimuovendo le parti laterali e superiori e lasciando solo la parte dell'immagine che ti interessa. I numeri rappresentano la coordinata X, la coordinata Y, la larghezza e l'altezza e per impostazione predefinita comprendono l'immagine completa. Prova a ridurre gli ultimi due numeri (0.5 e 0.5 è un buon punto di partenza) per vedere l'effetto di questa impostazione.

camera.exposure_compensation = 0

Imposta la *compensazione* dell'esposizione della fotocamera, consentendo di controllare manualmente la quantità di luce catturata per ogni immagine. Controlla la fotocamera stessa anziché solo le impostazioni di luminosità. I valori vanno da -25 per un'immagine molto scura a 25 per un'immagine molto luminosa.

camera.exposure_mode = 'auto'

Imposta la *modalità di esposizione* o la logica usata dal Camera Module/fotocamera HQ per decidere l'esposizione di un'immagine. Le modalità possibili sono: **off**, **auto**, **night**, **backlight**, **spotlight**, **sports**, **snow**, **beach**, **verylong**, **fixedfps**, **antishake** e **fireworks**.

camera.framerate = 30

Consente di impostare il numero di immagini acquisite per creare un video al secondo o la *frequenza di fotogrammi*. Una frequenza di fotogrammi più elevata consente di creare un video più fluido, ma occupa più memoria. Una frequenza di fotogrammi più elevata richiede una risoluzione inferiore, che può essere impostata tramite **camera.resolution**.

camera.hflip = False

Consente di capovolgere l'immagine della fotocamera lungo l'asse orizzontale, o X, quando impostato su **True**.

camera.image_effect = 'none'

Applica uno dei vari effetti al video e sarà visibile nell'anteprima, così come nelle immagini e video salvati. Gli effetti disponibili sono: **blur**, **cartoon**, **colorbalance**, **colorpoint**, **colorswap**, **deinterlace1**, **deinterlace2**, **denoise**, **emboss**, **film**, **gpen**, **hatch**, **negative**, **none**, **oilpaint**, **pastel**, **posterise**, **saturation**, **sketch**, **solarize**, **washedout** e **watercolor**.

camera.ISO = 0

Modifica l'impostazione ISO della fotocamera, variandone la sensibilità alla luce. Per impostazione predefinita, la fotocamera regola questa impostazione automaticamente in base alla luce disponibile. Puoi impostare autonomamente l'ISO utilizzando uno dei seguenti valori: 100, 200, 320, 400, 500, 640, 800. Più alto è l'ISO, migliore è il rendimento della fotocamera in ambienti poco illuminati, tuttavia l'immagine o il video acquisito saranno più sgranati.

camera.meter_mode = 'average'

Controlla il modo in cui la fotocamera decide la quantità di luce disponibile quando si imposta l'esposizione. Il valore predefinito è una media della quantità di luce disponibile in tutta l'immagine; altre possibili modalità sono **backlit**, **matrix** e **spot**.

camera.resolution = (1920, 1080)

Imposta la risoluzione dell'immagine o del video acquisito, rappresentata da due numeri per la larghezza e l'altezza. Le risoluzioni più basse occuperanno meno spazio di archiviazione e ti consentiranno di utilizzare una frequenza di fotogrammi più elevata, mentre quelle più alte hanno una qualità migliore, ma occupano più spazio di archiviazione.

camera.rotation = 0

Controlla la rotazione dell'immagine, da 0 a 90, 180 e 270 gradi. Usa questa impostazione se non riesci a posizionare la telecamera in modo che il cavo a nastro esca dalla parte inferiore.

camera.saturation = 0

Controlla la saturazione dell'immagine o quanto sono vividi i colori. I valori possibili vanno da -100 a 100.

camera.sharpness = 0

Controlla la nitidezza dell'immagine o quanto sono vividi i colori. I valori possibili vanno da -100 a 100.

camera.shutter_speed = 0

Controlla la velocità di apertura e chiusura dell'otturatore durante l'acquisizione di immagini e video. È possibile impostare manualmente la velocità dell'otturatore in microsecondi, con tempi di posa più lunghi per condizioni di scarsa illuminazione e più brevi in condizioni di luce più intensa. Generalmente questa impostazione può essere lasciata ai valori predefiniti (modalità automatica).

camera.vflip = False

Consente di capovolgere l'immagine della fotocamera lungo l'asse verticale, o Y, quando impostato su **True**.

camera.video_stabilization = False

Quando impostato su **True**, la stabilizzazione video è attiva. È necessario attivarla solo se la fotocamera è in movimento durante la registrazione, ad esempio se è collegata a un robot o viene trasportata, al fine di ridurre il tremolio del video acquisito.

Per ulteriori informazioni su queste impostazioni e sulle impostazioni aggiuntive non riportate nel presente capitolo, dai un'occhiata a [**picamera.readthedocs.io**](https://www.raspberrypi.org/documentation/software/picamera/).

Appendice A

Come installare un sistema operativo su una scheda microSD



Per installare facilmente il sistema operativo di Raspberry Pi (già noto come Raspbian) nel tuo Raspberry Pi, non dovrai fare altro che acquistare le schede microSD con NOOBS (New Out Of the Box Software) preinstallato presso qualsiasi rivenditore certificato Raspberry Pi. In alternativa, puoi eseguire manualmente l'installazione sulla scheda microSD vuota o riutilizzata, seguendo le istruzioni per l'utilizzo di Raspberry Pi Imager.

ATTENZIONE

Se hai acquistato una scheda microSD con NOOBS già preinstallato, non dovrai fare altro che collegarla al Raspberry Pi. Queste istruzioni si riferiscono a schede microSD vuote o già usate sulle quali si desidera installare un nuovo sistema operativo. Se la scheda microSD dovesse contenere già altri file, questi verranno sovrascritti ed eliminati, quindi assicurati di aver fatto il backup prima.

Come scaricare Raspberry Pi Imager

Basato su Debian, Raspberry Pi OS è il sistema operativo ufficiale di Raspberry Pi. Il modo più semplice per installare il sistema operativo Raspberry Pi su una scheda microSD è quello di utilizzare lo strumento Raspberry Pi Imager, scaricabile da [rpf.io/downloads](https://www.raspberrypi.org/downloads/). Questo metodo sostituisce l'installazione del sistema operativo tramite NOOBS, sebbene quest'ultimo sia ancora disponibile nella stessa pagina di download.

L'applicazione Raspberry Pi Imager è disponibile per computer Windows, macOS e Ubuntu Linux, quindi scegli la versione più adatta al tuo sistema.

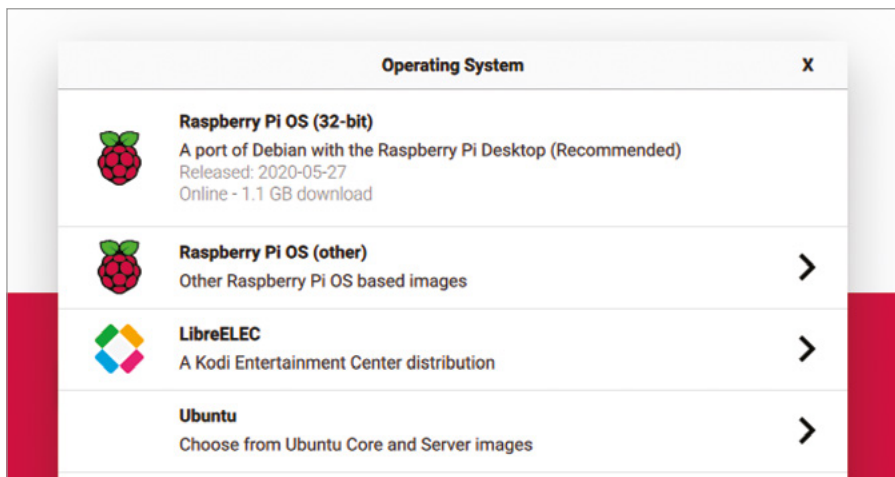
Su macOS, fai doppio clic sul file DMG che hai scaricato. Per consentire l'esecuzione delle applicazioni scaricate da "App Store e sviluppatori identificati", potresti dover modificare le impostazioni di sicurezza e privacy. Trascina quindi l'icona Raspberry Pi Imager nella cartella Applicazioni.

Su un PC Windows, fai doppio clic sul file EXE che hai scaricato. Quando richiesto, seleziona il pulsante "Sì" per autorizzarne l'esecuzione. Quindi, fai clic su "Installa" per avviare l'installazione.

Come scrivere il sistema operativo sulla scheda microSD

Collega la scheda microSD al PC o al computer Mac. A meno che il tuo computer non abbia già un lettore di schede integrato, dovrai disporre di un adattatore USB per schede microSD. La scheda non necessita di essere preformattata.

Esegui l'applicazione Raspberry Pi Imager, quindi fai clic su "Choose OS" per selezionare il sistema operativo che desideri installare. Si consiglia di selezionare il sistema operativo standard Raspberry Pi OS ma, se preferisci la versione più leggera (Lite) o quella completa (con tutti i software consigliati preinstallati), seleziona "Raspberry Pi OS (other)". Ci sono anche opzioni per installare LibreELEC (scegli la versione per il modello Raspberry Pi in uso) e Ubuntu Core o Server.



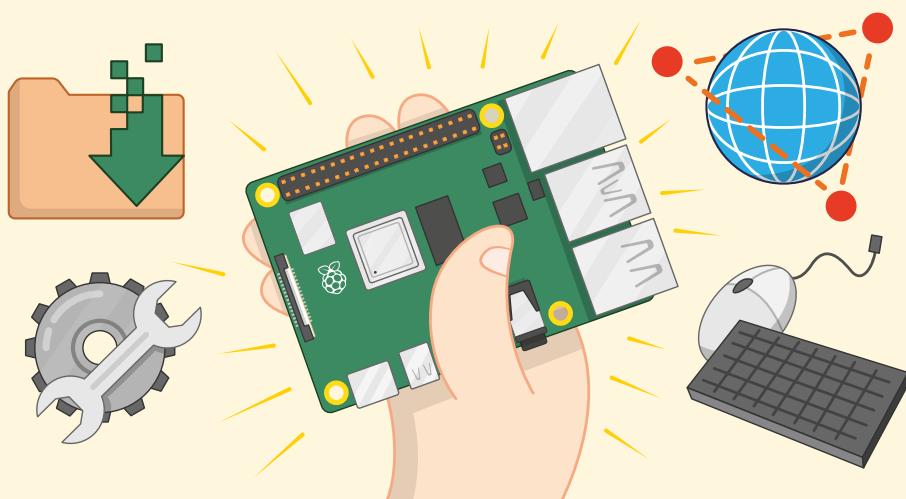
Nota: se desideri installare un altro sistema operativo, come Lakka, è sufficiente scaricare il file immagine dal relativo sito Web e selezionare l'opzione "Use Custom" in Raspberry Pi Imager.

Dopo avere selezionato un sistema operativo, fai clic sul pulsante "Choose SD card" e seleziona la tua scheda microSD (di solito c'è solo un'opzione).

Infine, fai clic sul pulsante "Write" e attendi mentre l'applicazione scrive il sistema operativo selezionato sulla scheda e lo verifica. Una volta completata l'operazione, potrai rimuovere la scheda microSD, inserirla nel Raspberry Pi e avviarla nel sistema operativo appena installato.

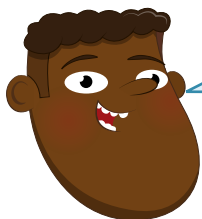
Appendice B

Come installare e disinstallare il software



Raspberry Pi OS viene fornito con una selezione di rinomati pacchetti software accuratamente scelti dalla Raspberry Pi Foundation, ma questi non sono gli unici pacchetti funzionanti su un Raspberry Pi. Attenendoti alle seguenti istruzioni, potrai cercare altri tipi di software e installarli e disinstallarli a piacere, ampliando le capacità del tuo Raspberry Pi.

Le istruzioni descritte in questa appendice si aggiungono a quelle del **Capitolo 3, Come utilizzare Raspberry Pi** dove si spiega come utilizzare lo strumento Recommended Software. Se non sai di cosa si tratta, leggi il capitolo prima di utilizzare le procedure descritte in questa appendice.



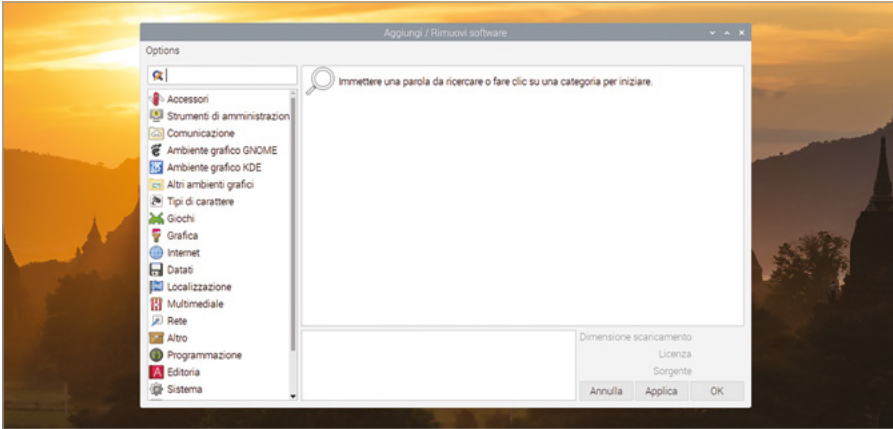
CAPACITÀ DELLA SCHEDA

Aggiungendo altri software al Raspberry Pi occuperai spazio sulla scheda microSD. Una scheda da 16 GB o più grande ti permetterà di installare più software. Per verificare se la scheda che intendi utilizzare è compatibile con Raspberry Pi, visita il sito rpf.io/sdcardlist.

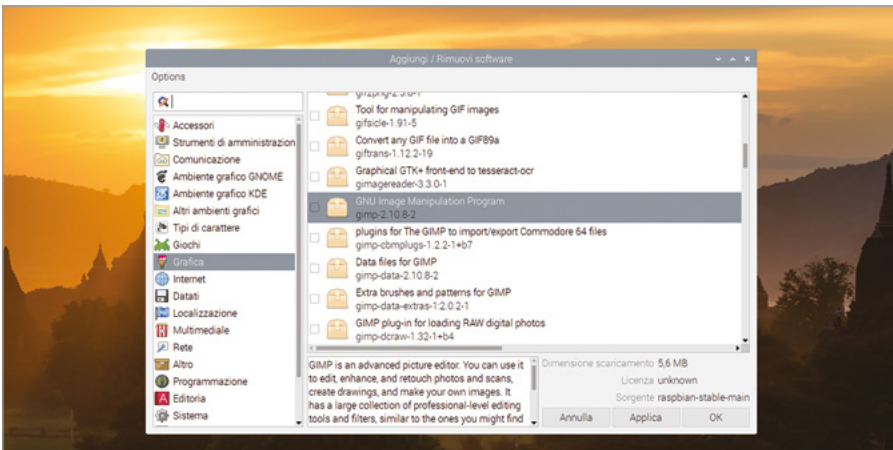


Come cercare i software disponibili

Per visualizzare e cercare i pacchetti software disponibili per Raspberry Pi OS utilizzando i cosiddetti *repository di software*, fai clic sull'icona a lampone per caricare il menu, seleziona la categoria Preferenze, quindi fai clic sullo strumento di aggiunta o rimozione del software. Dopo pochi secondi, verrà visualizzata la finestra dello strumento.



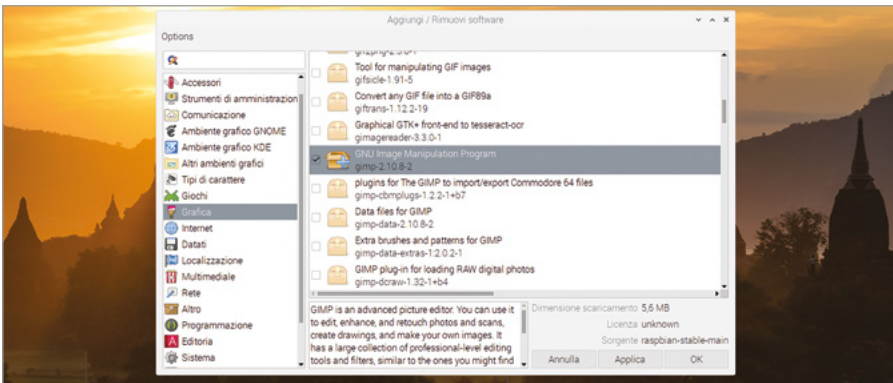
La parte sinistra della finestra per l'aggiunta o la rimozione del software contiene un elenco di categorie che sono le stesse del menu principale che viene visualizzato quando si fa clic sull'icona a lampone. Facendo clic su una di esse, potrai visualizzare l'elenco dei software disponibili in quella categoria. Nella casella in alto a sinistra della finestra, puoi anche inserire un termine di ricerca, come "editor di testo" o "gioco", e visualizzare l'elenco dei pacchetti software corrispondenti per qualsiasi categoria. Facendo clic su uno qualsiasi dei pacchetti, verranno visualizzate ulteriori informazioni a riguardo nello spazio in basso nella finestra.



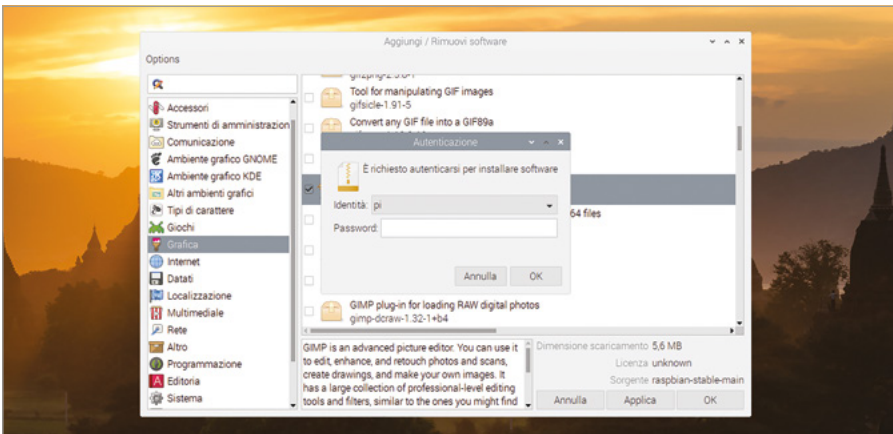
Se per la categoria scelta sono disponibili molti pacchetti software, potrebbe essere necessario un po' di tempo prima che lo strumento di aggiunta o rimozione del software finisca di analizzare l'elenco.

Come installare il software

Per selezionare un pacchetto per l'installazione, spunta la casella accanto a esso. Puoi installare più di un pacchetto alla volta semplicemente selezionando quelli che ti interessano. L'icona accanto al pacchetto diventerà una scatola aperta con il simbolo "+", per confermare che verrà installato.

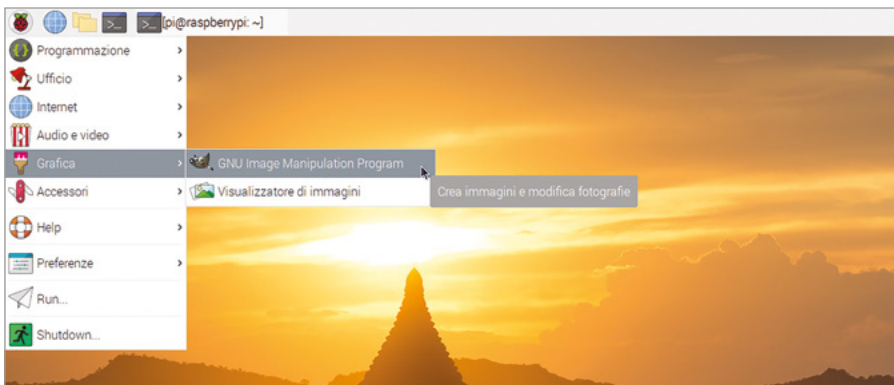


Quando hai scelto, fai clic sul pulsante OK o su Applica. L'unica differenza è che se selezioni OK lo strumento di aggiunta o rimozione del software verrà chiuso una volta completata l'installazione, mentre se selezioni Applica rimarrà aperto. Per evitare che il software venga aggiunto o rimosso da altri utenti, ti verrà chiesto di inserire la password personale per confermare la tua identità.



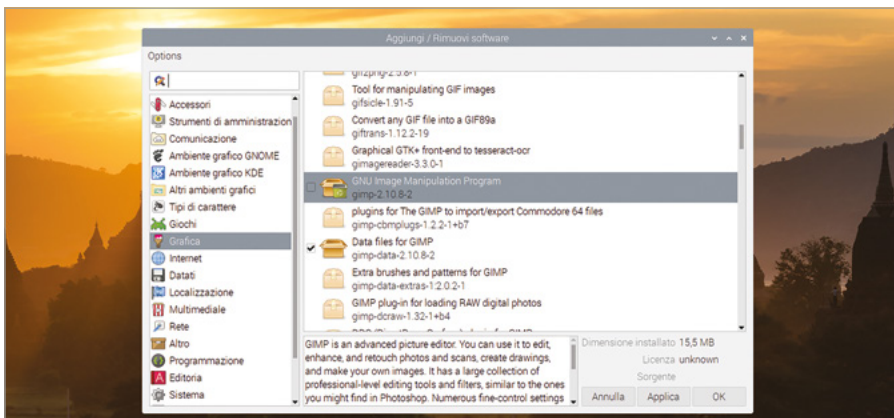
Noterai che quando installi un singolo pacchetto, ne verranno installati altri. Sono le cosiddette *dipendenze*, pacchetti che servono al software per funzionare come, ad esempio, bundle di effetti sonori per un gioco o database da collegare a un server Web.

Una volta installato il software, dovresti trovarlo facendo clic sull'icona del lampone per caricare il menu e trovare la categoria del pacchetto software. Tieni presente che la categoria del menu non corrisponde sempre a quella dello strumento di aggiunta o rimozione del software, mentre alcuni software non hanno proprio una voce nel menu. Questo software è noto come *software a riga di comando* e deve essere eseguito sull'LXTerminal. Per ulteriori informazioni sulla riga di comando e sull'LXTerminal, consulta l'**Appendice C, Interfaccia a riga di comando**.



Come disinstallare il software

Per selezionare un pacchetto da rimuovere o *disinstallare*, cercalo nell'elenco dei pacchetti (in questo caso la funzione di ricerca torna utile) e toglie la spunta dalla casella corrispondente al pacchetto che vuoi rimuovere. Puoi disinstallare più di un pacchetto alla volta semplicemente deselegando quelli che vuoi rimuovere. L'icona accanto al pacchetto diventerà una scatola aperta vicino a un cestino, per confermare che verrà disinstallato.



Come prima, puoi scegliere se fare clic su OK o su Applica per iniziare a disinstallare i pacchetti software selezionati. A meno che tu non l'abbia già fatto, ti verrà chiesto di confermare la password e che desideri rimuovere anche i pacchetti dipendenti del software. Quando la disinstallazione sarà terminata, il software scomparirà dal menu con l'icona a lampone, ma i file creati con il software, come le immagini per un pacchetto grafico o i salvataggi di un gioco, non verranno rimossi.

ATTENZIONE



Tutti i software installati nel sistema operativo Raspberry Pi OS vengono visualizzati nello strumento di aggiunta o rimozione del software, incluso quello necessario per l'esecuzione del Raspberry Pi. I pacchetti che il desktop non carica più possono essere rimossi, ma ti consigliamo di non disinstallare nulla se non sei sicuro di quelli che potrebbero ancora servirti. Se li hai già rimossi, reinstalla Raspberry Pi OS seguendo le istruzioni descritte nel **Capitolo 2, Introduzione a Raspberry Pi** oppure reinstalla il sistema operativo consultando l'**Appendice A**.

Appendice C

Interfaccia a riga di comando



La maggior parte dei software su Raspberry Pi è gestibile dal desktop, ma alcuni sono accessibili solo utilizzando una modalità testuale nota come *interfaccia a riga di comando (CLI)* in un'applicazione denominata LXTerminal. La maggior parte degli utenti non avrà mai bisogno di utilizzare la CLI, ma per coloro che vogliono saperne di più, questa appendice offre una piccola introduzione.

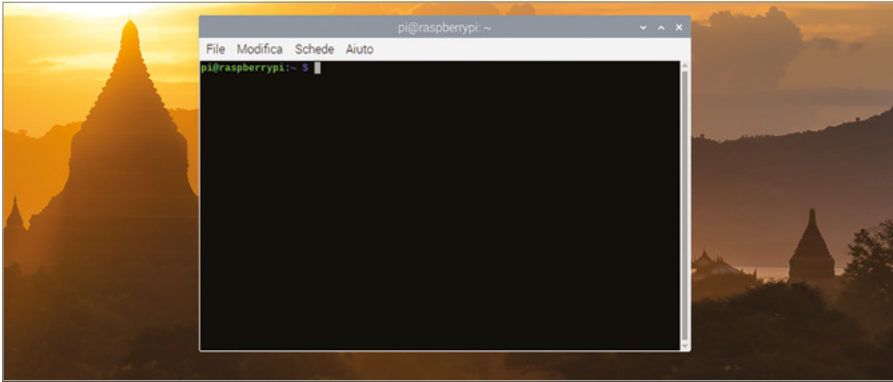


MAGGIORI INFORMAZIONI

Questa appendice non funge da guida esaustiva all'interfaccia a riga di comando di Linux. Per un'analisi più dettagliata dell'utilizzo della CLI, visita il sito rpf.io/terminal in un browser Web.

Come caricare l'LXTerminal

La CLI è accessibile tramite LXTerminal, un pacchetto software che carica un componente tecnicamente noto come *telescrivente virtuale (VTY)*, nome che risale ai primi tempi dei computer, quando gli utenti lanciavano i comandi tramite una grande macchina da scrivere elettromeccanica invece che da tastiera e monitor. Per caricare il pacchetto LXTerminal, fai clic sull'icona a lampone per caricare il menu, scegli la categoria Accessori, quindi fai clic su LXTerminal.



Puoi trascinare la finestra di LXTerminal sul desktop, ridimensionarla, ingrandirla o ridurla a icona come qualsiasi altra finestra. Puoi ingrandire la dimensione dei caratteri per leggere meglio, oppure ridurla per visualizzare più testo nella finestra: fai clic sul menu Edit e scegli rispettivamente Zoom In o Zoom Out, oppure tieni premuto il tasto **CTRL** sulla tastiera insieme al **+** o al **-**.

Il prompt

La prima cosa che viene visualizzata in LXTerminal è il *prompt* in attesa che tu inserisca le istruzioni. Il prompt su un Raspberry Pi che esegue Raspberry Pi OS si presenta così:

```
pi@raspberrypi:~ $
```

La prima parte del prompt, **pi**, è il tuo nome utente, la seconda parte, dopo **@**, è il nome host del computer che stai utilizzando che per impostazione predefinita è **raspberrypi**. Dopo **:** c'è una tilde, **~**, un'abbreviazione che si riferisce alla home directory e rappresenta la tua *directory di lavoro corrente (CWD)*. Infine, il simbolo **\$** indica che l'utente è un *utente senza privilegi*, il che significa che, per eseguire operazioni come aggiungere o rimuovere software, è richiesta una password.

Come muoversi

Prova a digitare quanto segue, quindi premi il tasto **INVIO**:

```
cd Desktop
```

Vedrai il prompt diventare:

```
pi@raspberrypi:~/Desktop $
```

Come vedi, la directory di lavoro corrente è cambiata: prima eri nella home directory, indicata con la `~`, mentre adesso sei nella sottodirectory **Desktop** della directory principale. Per farlo, hai usato il comando `cd` - *cambia directory*.



DISTINZIONE TRA MAIUSCOLE E MINUSCOLE

L'interfaccia a riga di comando del sistema operativo Raspberry Pi OS necessita che per i comandi o i nomi venga rispettata la distinzione tra caratteri maiuscoli e minuscoli. Se quando provi a cambiare directory visualizzi il messaggio "no such file or directory", verifica che la D di Desktop sia maiuscola.

Ci sono quattro modi per tornare alla home directory: provali tutti, tornando ogni volta nella sottodirectory **Desktop**. Il primo modo è:

```
cd ..
```

I due puntini `..` sono un'altra scorciatoia che porta alla "directory sopra questa", nota anche come *directory padre*. Poiché la directory sopra **Desktop** è la tua home directory, questa scorciatoia ti riporta lì. Torna alla sottodirectory **Desktop** e prova il secondo metodo:

```
cd ~
```

In questo caso, viene utilizzato il simbolo `~` che significa letteralmente "fai diventare la mia home directory". A differenza di `cd ..`, che ti porta alla directory padre di qualsiasi directory in cui ti trovi, questo comando funziona da qualsiasi posizione. C'è però un modo ancora più semplice:

```
cd
```

Senza il nome di una directory, per impostazione predefinita, `cd` punta direttamente alla home directory. L'ultimo modo per tornare alla home directory è quello di digitare:

```
cd /home/pi
```

Questo comando utilizza quello che viene chiamato *percorso assoluto*, che funzionerà indipendentemente dalla directory di lavoro corrente. Quindi, come `cd` o `cd ~`, ti riporterà alla home directory da qualsiasi posizione. A differenza degli altri metodi, però, dovrai conoscere il tuo nome utente.

Come gestire i file

Per esercitarti a lavorare con i file, vai alla directory **Desktop** e digita quanto segue:

touch Test

Sul desktop verrà visualizzato un file chiamato **Test**. Il comando **touch** viene normalmente utilizzato per aggiornare le informazioni di data e ora su un file, ma se, come in questo caso, il file non esiste, lo crea.

Prova a digitare:

cp Test Test2

Sul desktop verrà visualizzato un altro file chiamato **Test2** che è una *copia* identica del file originale. Eliminalo digitando:

rm Test2

Questo comando *rimuove* il file.

ATTENZIONE

A differenza della cancellazione dei file tramite il File Manager grafico, che li memorizza nel cestino da cui possono essere eventualmente recuperati, i file cancellati utilizzando il comando **rm** verranno eliminati definitivamente. Assicurati di scrivere i comandi correttamente!

Ora prova:

mv Test Test2

Questo comando *sposta* il file e consente di sostituire il file **Test** originale, che scomparirà, con il file **Test2**. Il comando di spostamento, **mv**, può essere utilizzato per rinominare i file.

Quando non ti trovi sul desktop, però, dovrai comunque essere in grado di visualizzare i file contenuti in una directory. Digita:

ls

Questo comando *elenca* il contenuto della directory corrente, o di qualsiasi altra directory indicata. Per ulteriori dettagli, tra cui l'elenco di tutti i file nascosti e le dimensioni dei file, prova ad aggiungere altri commutatori:

ls -larth

Questi commutatori controllano il comando **ls**: **l** dà il comando di creare un lungo elenco verticale; **a** gli dice di mostrare tutti i file e le directory, compresi quelli che normalmente sarebbero nascosti; **r** inverte il normale ordine di successione; **t** ordina per data e ora di modifica e, insieme a **r**, ti permette di visualizzare i file più vecchi in cima all'elenco e quelli più recenti in fondo; **h**, infine, consente di utilizzare dimensioni leggibili, rendendo l'elenco più facile da comprendere.

Come eseguire i programmi

Alcuni programmi possono essere eseguiti solo a riga di comando, mentre altri hanno interfacce sia grafiche che a riga di comando come, ad esempio, Raspberry Pi Software Configuration Tool, che normalmente si carica dal menu con l'icona a lampone.

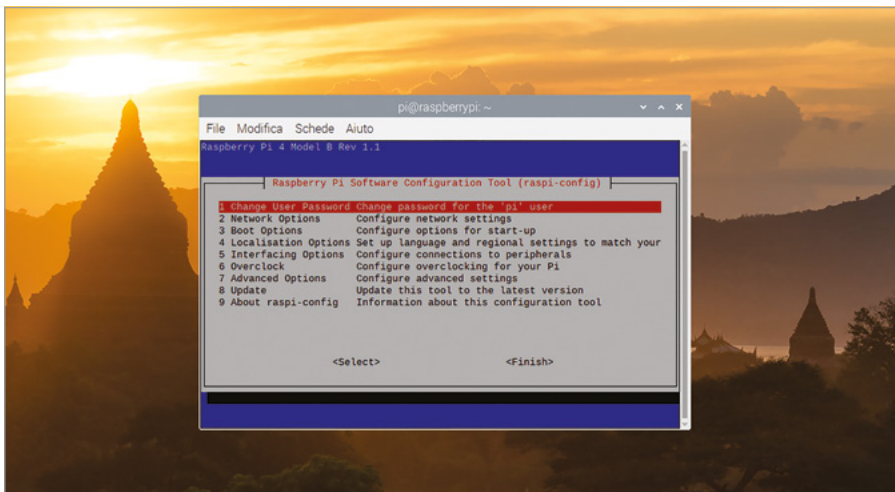
Digita:

```
raspi-config
```

Verrà visualizzato un messaggio di errore che ti informa che il software può essere eseguito solo come *root*, l'account utente avanzato sul tuo Raspberry Pi. Per utilizzare tale account, come indicato anche nel messaggio, devi digitare:

```
sudo raspi-config
```

La parte **sudo** del comando significa *cambia utente* e dice a Raspberry Pi OS di eseguire il comando come utente *root*.



Dovrai usare **sudo** solo quando un programma ha bisogno di *privilegi* avanzati, ad esempio quando si tratta di installare o disinstallare un software o di settare le impostazioni di sistema. Un gioco, per esempio, non dovrebbe mai essere eseguito utilizzando **sudo**.

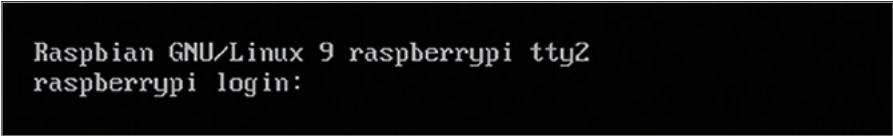
Premi due volte il tasto **TAB** per selezionare Finish e **INVIO** per uscire da Raspberry Pi Software Configuration Tool e tornare all'interfaccia a riga di comando. Infine, digita:

```
exit
```

Questo comando termina la sessione dell'interfaccia a riga di comando e chiude l'applicazione LXTerminal.

Come utilizzare i terminali TTY

L'applicazione LXTerminal non è l'unico modo per utilizzare l'interfaccia a riga di comando: è possibile anche passare a uno dei numerosi terminali già in funzione, noti come *telescriventi* o TTY. Tieni premuti i tasti **CTRL** e **ALT** sulla tastiera mentre premi **F2** per passare a "tty2".



```
Raspbian GNU/Linux 9 raspberrypi tty2
raspberrypi login:
```

Dovrai effettuare nuovamente l'accesso con il tuo nome utente e password, dopodiché potrai utilizzare l'interfaccia a riga di comando proprio come in LXTerminal. L'utilizzo di questi TTY è utile quando, per qualsiasi motivo, l'interfaccia principale del desktop non funziona.

Per uscire dai TTY, tieni premuto **CTRL+ALT**, quindi premi **F7** e il desktop verrà nuovamente visualizzato. Premi nuovamente **CTRL+ALT+F2** per tornare a "tty2" e ritrovare tutto ciò che era in esecuzione.

Prima di cambiare di nuovo, digita:

```
exit
```

Quindi premi **CTRL+ALT+F7** per tornare al desktop. Prima di uscire dai TTY è importante eseguire il comando di uscita poiché chiunque abbia accesso alla tastiera potrebbe passare a un TTY senza dovere inserire la password.

Complimenti, hai mosso i tuoi primi passi nell'interfaccia a riga di comando del sistema operativo Raspberry Pi OS.

Appendice D

Ulteriori letture



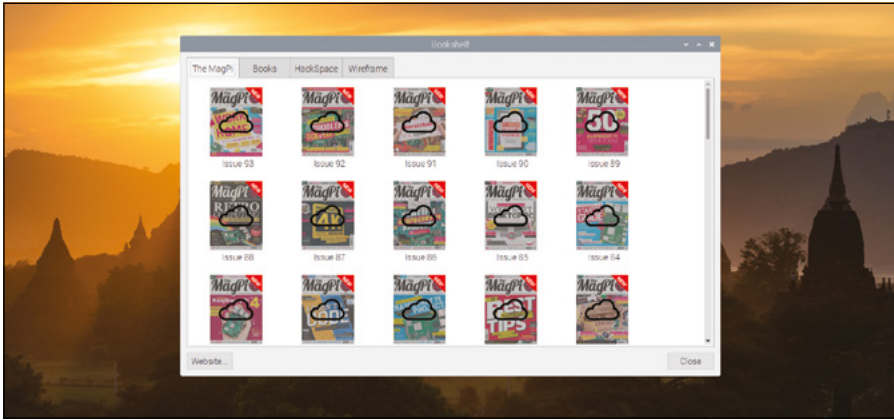
La *Guida introduttiva ufficiale Raspberry Pi* ti aiuta a muovere i primi passi con il tuo Raspberry Pi ma non approfondisce tutti gli aspetti e le funzionalità a tua disposizione. Raspberry Pi vanta una community vasta, composta da persone che lo utilizzano per tutte le attività possibili, dai giochi alle applicazioni di rilevamento, dalla robotica all'intelligenza artificiale e tanto altro ancora.

In questa appendice troverai alcuni spunti per elaborare idee progettuali, piani di lezione e altro materiale per ampliare le conoscenze che hai acquisito con la *Guida introduttiva*.

Bookshelf

► Menu Raspberry > Guida > Bookshelf

Bookshelf è un'applicazione inclusa in Raspberry Pi OS che ti consente di scorrere, scaricare e leggere le versioni digitali delle pubblicazioni di Raspberry Pi Press, compresa la Guida introduttiva (versione inglese). Fai clic sull'icona del menu a forma di lampone e seleziona Guida e poi Bookshelf per scaricare l'applicazione, quindi scegli tra tantissimi libri e riviste e scarica gratuitamente i tuoi preferiti per leggerli in qualsiasi momento.



Il blog di Raspberry Pi

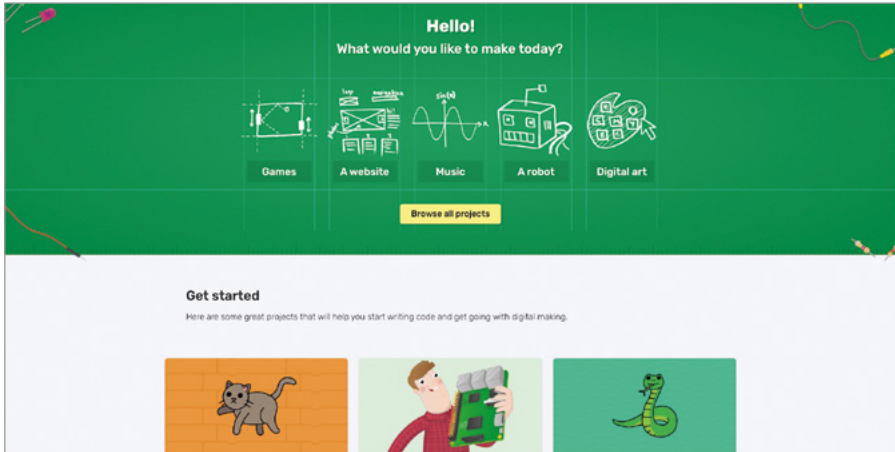
► rpf.io/blog

La prima fonte cui fare riferimento per scoprire le ultime novità sul mondo Raspberry Pi è il blog ufficiale, dove troverai tutti gli aggiornamenti più recenti relativi al lancio di nuovi hardware e ai materiali didattici e una raccolta dei migliori progetti, campagne e iniziative della community. Se vuoi restare sempre aggiornato sulle novità Raspberry Pi, questo è il posto giusto.

Raspberry Pi Projects

► rpf.io/projects

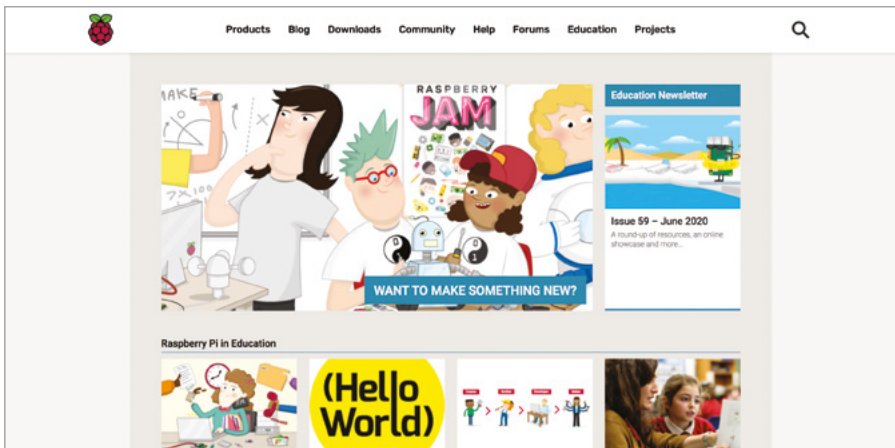
Il sito ufficiale Raspberry Pi Projects offre tutorial dettagliati per la realizzazione di diverse tipologie di progetti, dalla creazione di giochi e musica allo sviluppo del proprio sito Web o alla creazione di robot con scheda Raspberry Pi. La maggior parte dei progetti è disponibile in diverse lingue e livelli di difficoltà adatti a tutti, dai principianti assoluti agli utenti più esperti.



Raspberry Pi Education

► rpf.io/education

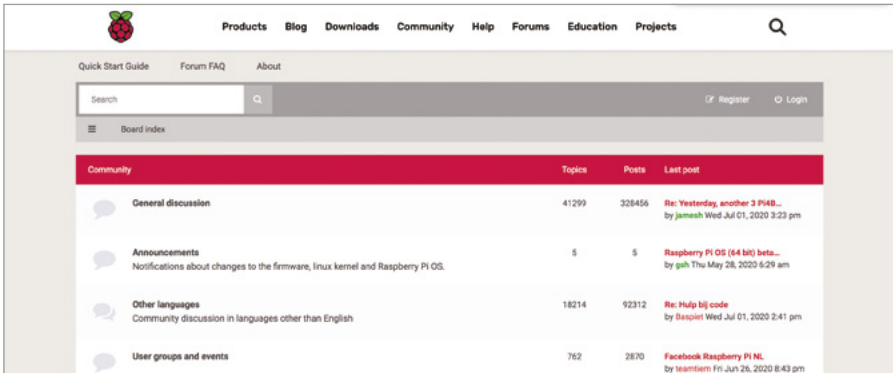
Il sito ufficiale di Raspberry Pi Education offre newsletter, didattica online e progetti a tutti coloro che si occupano di formazione. Il sito è collegato anche ad altre risorse, tra cui il programma di formazione Picademy, i programmi di codifica Code Club e CoderDojo gestiti da volontari e gli eventi internazionali Raspberry Jam.



I forum di Raspberry Pi

► rpf.io/forums

Nei forum di Raspberry Pi, gli appassionati di questo prodotto possono incontrarsi e parlare di tutto, dai problemi dei principianti alle questioni più tecniche, e c'è anche un'area "off-topic" per discutere di argomenti più generici.



Il magazine The MagPi

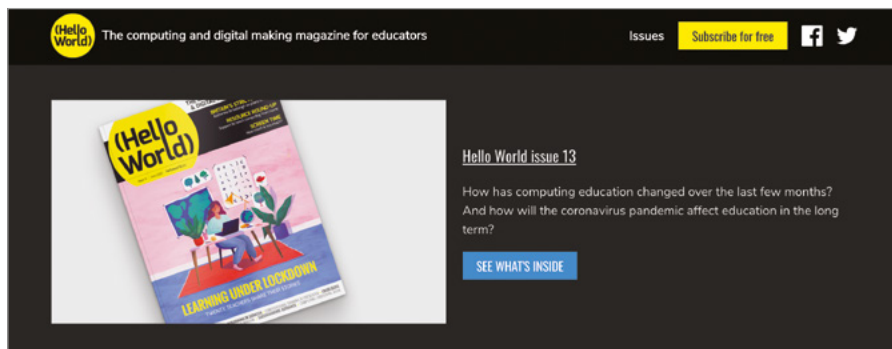
► magpi.cc

The MagPi, la rivista ufficiale di Raspberry Pi supportata in gran parte dalla community internazionale di Raspberry Pi, è un mensile patinato che contiene tutto ciò di cui hai bisogno, dai tutorial e dalle guide alle recensioni e alle ultime novità. Lo trovi in tutte le edicole e nei supermercati oppure puoi scaricarlo gratuitamente in formato digitale con la licenza Creative Commons. The MagPi pubblica anche libri e riviste su diversi argomenti, anch'essi disponibili per l'acquisto in formato cartaceo o per il download gratuito.

Il magazine Hello World

► helloworld.cc

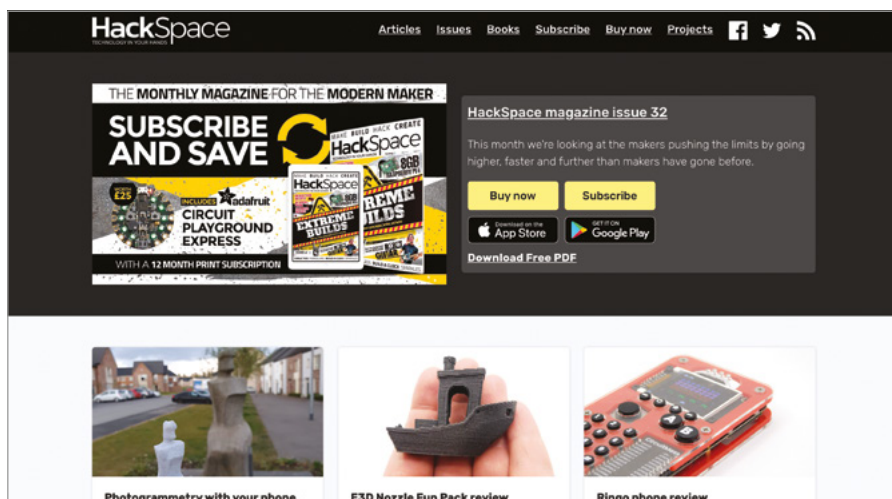
Con uscite quadrimestrali, Hello World è disponibile gratuitamente per insegnanti, volontari e bibliotecari del Regno Unito. Tutti gli altri possono scaricare le copie digitali gratuite sotto licenza Creative Commons o acquistare gli abbonamenti alla versione cartacea disponibili in commercio.



Il magazine HackSpace

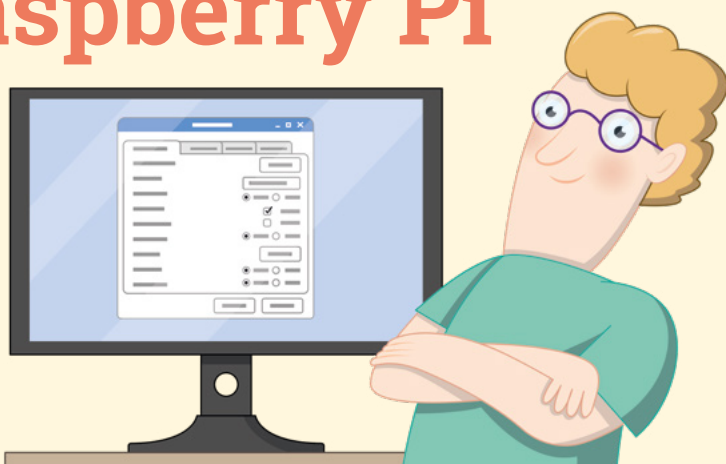
► hsmag.cc

Rivolto a un pubblico più ampio rispetto a The MagPi, il magazine HackSpace è incentrato sulla community degli sviluppatori, con recensioni di hardware e software, tutorial e interviste. Se sei interessato a espandere i tuoi orizzonti oltre Raspberry Pi, HackSpace è un ottimo punto di partenza. Lo puoi trovare in edicola e nei supermercati in versione cartacea o scaricarlo gratuitamente in formato digitale.



Appendice E

Strumento Configurazione di Raspberry Pi



Lo strumento Configurazione di Raspberry Pi è un potente pacchetto che ti consente di regolare tutta una serie di impostazioni sul tuo Raspberry Pi, dalle interfacce ai programmi per controllarlo in rete. Poiché all'inizio potresti demoralizzarti un po', questa appendice ti illustrerà nel dettaglio ciascuna impostazione, spiegandone l'utilità.

Puoi avviare lo strumento Configurazione di Raspberry Pi dal menu facendo clic sull'icona a forma di lampone, sotto la categoria Preferenze, oppure puoi eseguirlo dall'interfaccia a riga di comando o dall'LXTerminal usando il comando **raspi-config**. I layout della versione a riga di comando e della versione grafica differiscono tra loro, con opzioni che vengono visualizzate in categorie diverse a seconda della versione utilizzata. Questa appendice si riferisce alla versione grafica.

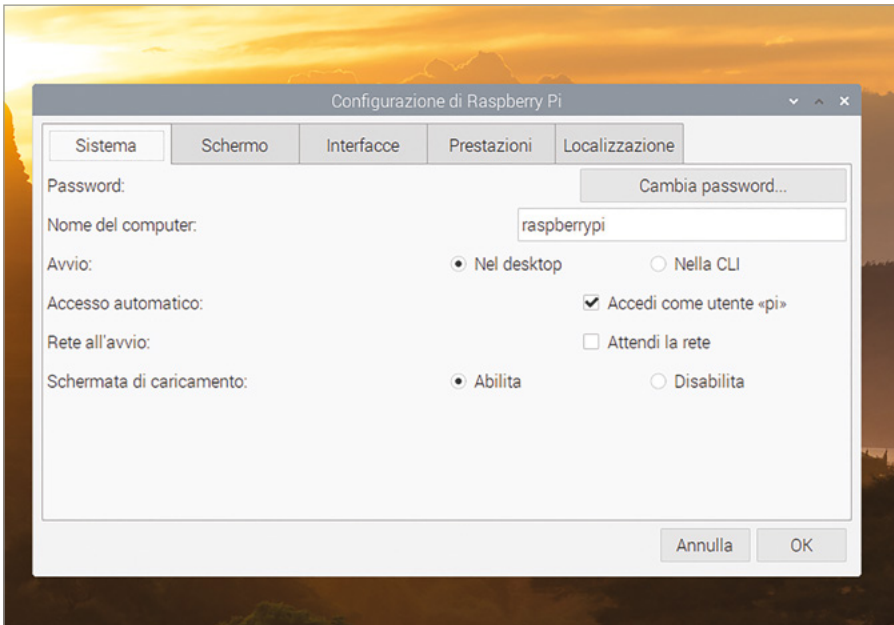
ATTENZIONE

A meno che tu non abbia bisogno di modificare una specifica impostazione, è meglio non utilizzare lo strumento Configurazione di Raspberry Pi. Se stai aggiungendo un nuovo componente hardware al tuo Raspberry Pi, ad esempio un HAT audio o un Camera Module (modulo fotocamera), segui le istruzioni per sapere quale impostazione modificare. In caso contrario, non modificare le impostazioni predefinite.



Scheda Sistema

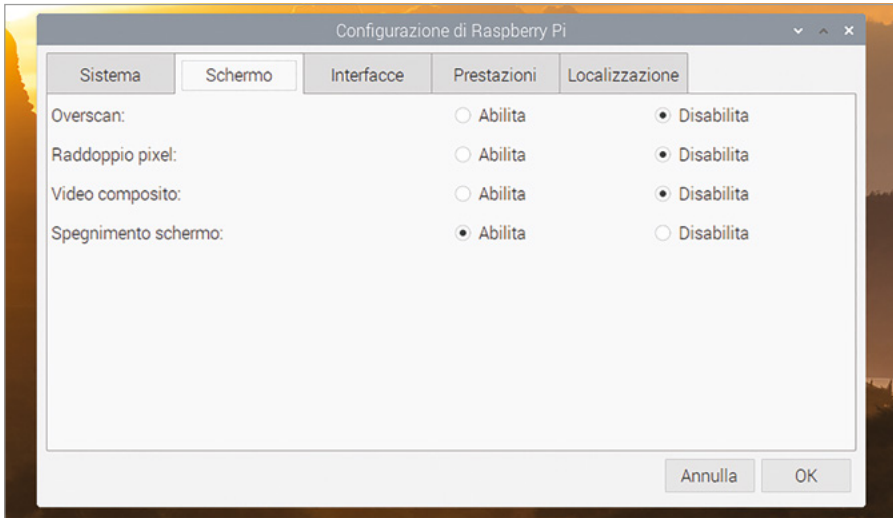
La scheda Sistema contiene le opzioni che controllano le varie impostazioni del sistema operativo Raspberry Pi OS.



- **Password:** fai clic sul pulsante "Cambia password..." per impostare una nuova password per l'account utente in uso che, per impostazione predefinita, è "pi".
- **Nome del computer:** Il nome con cui un Raspberry Pi si identifica sulle reti. Se hai più di un Raspberry Pi sulla stessa rete, ogni dispositivo deve avere un nome univoco.
- **Avvio:** impostando l'avvio su "Nel desktop" (impostazione predefinita) verrà caricato il desktop di Raspberry Pi OS che già conosci. Se invece lo imposti su "Nella CLI" si caricherà l'interfaccia a riga di comando come descritto nell'**Appendice C, Interfaccia a riga di comando**.
- **Accesso automatico:** quando selezioni "Accedi come utente "pi"" (impostazione predefinita), Raspberry Pi OS carica il desktop senza bisogno di inserire nome utente e password.
- **Rete all'avvio:** quando spunti la casella "Attendi rete", Raspberry Pi OS non si caricherà finché non sarà disponibile una connessione di rete funzionante.
- **Schermata di caricamento:** quando è impostata su "Abilita" (impostazione predefinita), i messaggi di avvio di Raspberry Pi OS saranno nascosti dietro una schermata grafica di avvio.

Scheda Schermo

La scheda Schermo contiene le impostazioni che controllano la visualizzazione dello schermo.



■ **Overscan:** questa impostazione controlla se l'uscita video su Raspberry Pi include o meno delle strisce nere intorno ai bordi per adattare il formato a quello di molti televisori. Se le strisce nere sono presenti, imposta questa opzione su "Disabilita", al contrario, lasciala impostata su "Abilita".

■ **Raddoppio pixel:** se stai utilizzando uno schermo ad alta risoluzione ma di piccole dimensioni, puoi attivare Raddoppio pixel per ingrandire le immagini e visualizzare meglio tutti i contenuti.

■ **Video composito:** questa opzione controlla l'uscita video composito disponibile sul connettore jack audio-video (AV), se utilizzato con un adattatore tip-ring-ring-sleeve (TRRS). Se desideri utilizzare l'uscita video composito invece di quella HDMI, imposta l'opzione su "Abilita", altrimenti lasciala disabilitata.

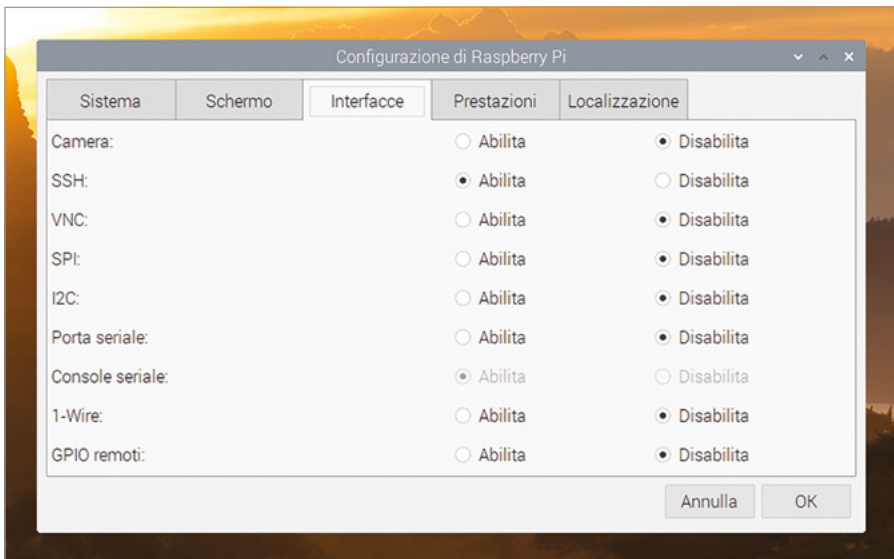
■ **Spegnimento schermo:** questa opzione consente di attivare e disattivare lo spegnimento dello schermo (il timeout che spegne lo schermo dopo alcuni minuti).

Scheda Interfacce

La scheda Interfacce contiene le impostazioni che controllano le interfacce hardware disponibili su Raspberry Pi.

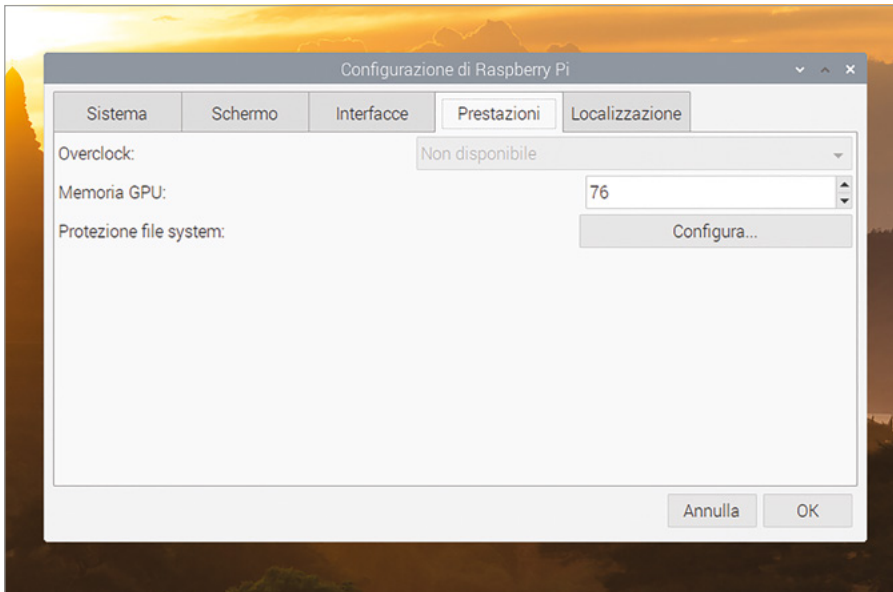
■ **Camera:** abilita o disabilita l'interfaccia seriale della fotocamera (CSI), da utilizzare con un Raspberry Pi Camera Module (modulo fotocamera).

- **SSH:** abilita o disabilita l'interfaccia Secure Shell (SSH); consente di aprire un'interfaccia a riga di comando su Raspberry Pi da un altro computer della rete utilizzando un client SSH.
- **VNC:** abilita o disabilita l'interfaccia Virtual Network Computing (VNC); consente di visualizzare il desktop di Raspberry Pi da un altro computer della rete utilizzando un client VNC.
- **SPI:** abilita o disabilita l'interfaccia seriale periferica (SPI), utilizzata per controllare alcuni componenti hardware aggiuntivi che si collegano ai pin GPIO.
- **I2C:** abilita o disabilita l'interfaccia Inter-Integrated Circuit (I²C), utilizzata per controllare alcuni componenti hardware aggiuntivi che si collegano ai pin GPIO.
- **Porta seriale:** abilita o disabilita la porta seriale di Raspberry Pi, disponibile sui pin GPIO.
- **Console seriale:** abilita o disabilita la console seriale, un'interfaccia a riga di comando disponibile sulla porta seriale. Questa opzione è disponibile solo se l'impostazione della porta seriale sopra citata è impostata su "Abilita".
- **1-Wire:** abilita o disabilita l'interfaccia 1-Wire, utilizzata per controllare alcuni componenti hardware aggiuntivi che si collegano ai pin GPIO.
- **GPIO remoti:** abilita o disabilita un servizio di rete che consente di controllare i pin GPIO di Raspberry Pi da un altro computer in rete utilizzando la libreria GPIO Zero. Maggiori informazioni sui GPIO remoti sono consultabili su gpiozero.readthedocs.io.



Scheda Prestazioni

La scheda Prestazioni contiene le impostazioni che controllano la quantità di memoria disponibile e la velocità di esecuzione del processore Raspberry Pi.



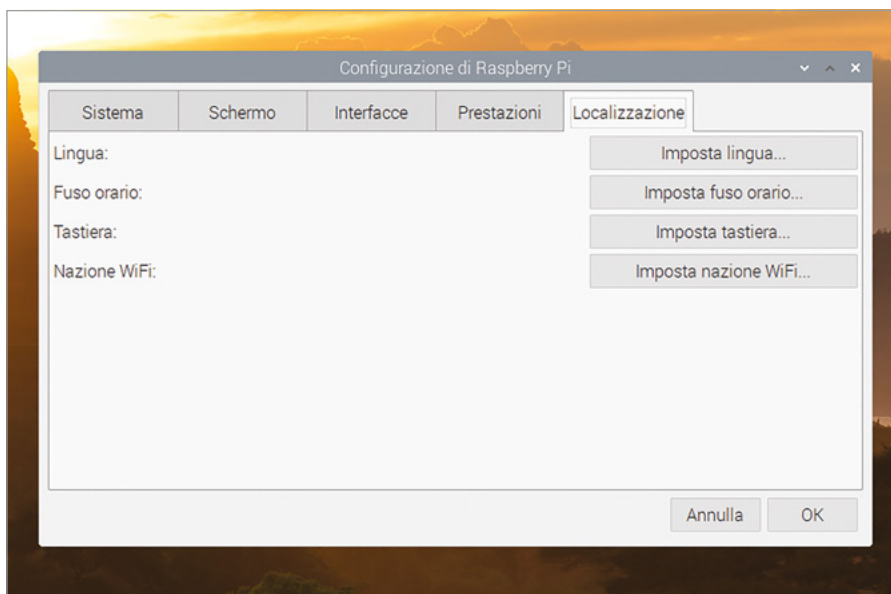
■ **Overclock:** permette di scegliere tra una serie di impostazioni che migliorano le prestazioni del tuo Raspberry Pi aumentando però il consumo di energia e la produzione di calore e, in alcuni casi, diminuendo la durata complessiva del dispositivo. Non disponibile per tutti i modelli di Raspberry Pi.

■ **Memoria GPU:** permette di impostare la quantità di memoria riservata all'uso del processore grafico di Raspberry Pi. Valori superiori a quello predefinito possono migliorare le prestazioni per complicate operazioni di rendering 3D e per attività di GPU general-purpose (GPGPU) a discapito della quantità di memoria disponibile per Raspberry Pi OS. Valori inferiori, invece, possono migliorare le prestazioni per attività che richiedono un impiego elevato di memoria ma rallentano altre funzionalità, come il rendering 3D, la fotocamera e le funzioni di riproduzione video selezionate, o le rendono non disponibili.

■ **Protezione file system:** permette di bloccare il file system di Raspberry Pi affinché le modifiche vengano effettuate solo su un disco RAM virtuale piuttosto che essere scritte sulla scheda microSD, in modo da riportare il sistema a uno stato pulito a ogni riavvio.

Scheda Localizzazione

La scheda Localizzazione contiene le impostazioni che controllano per quale area geografica è stato progettato il tuo Raspberry Pi, comprese le impostazioni di layout della tastiera.



- **Lingua:** un'impostazione di sistema che consente di scegliere la lingua, il paese e il set di caratteri. Cambiando la lingua da qui, cambierà solo la lingua visualizzata nelle applicazioni per le quali è disponibile una traduzione.
- **Fuso orario:** consente di impostare il fuso orario selezionando l'area geografica desiderata seguita dalla città più vicina. Se il tuo Raspberry Pi è collegato alla rete ma l'orologio mostra l'ora sbagliata, è probabile che il fuso orario non sia stato impostato correttamente.
- **Tastiera:** permette di scegliere il tipo di tastiera, la lingua e il layout. Se la tua tastiera digita le lettere o i simboli sbagliati, puoi impostarla correttamente da qui.
- **Nazione WiFi:** consente di impostare il proprio paese ai fini del regolamento sulle radiocomunicazioni. Assicurati di selezionare il paese in cui il tuo Raspberry Pi viene utilizzato: selezionando un altro paese, potresti non riuscire a connetterti ai punti di accesso WiFi nelle vicinanze e commettere una violazione della legge sulle trasmissioni radiotelevisive. Prima di poter utilizzare il WiFi è necessario impostare un paese.

Appendice F

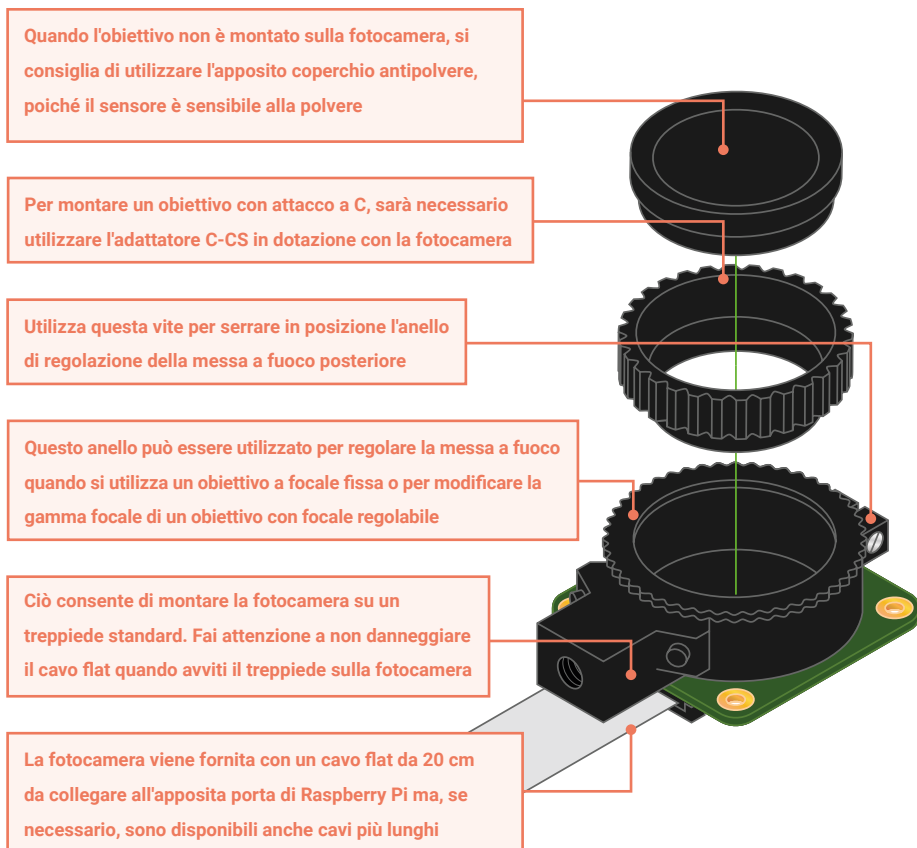
Impostazione della High Quality Camera

La High Quality Camera (fotocamera di alta qualità) è in grado di scattare immagini con risoluzione superiore rispetto al Camera Module (modulo fotocamera) standard. A differenza di quest'ultima, non dispone di un obiettivo in dotazione, ma può essere utilizzata con qualsiasi obiettivo standard con attacco C o CS. Per iniziare a muovere i primi passi, sono disponibili per l'acquisto insieme alla fotocamera gli obiettivi da 6 e 16 mm.

Obiettivo da 6 mm con attacco CS

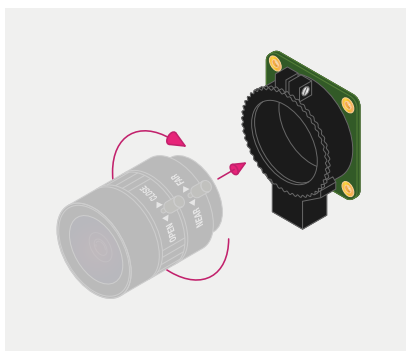
Senza spendere troppo, è possibile acquistare un obiettivo da 6 mm che permette anche ai principianti di scattare foto di qualità. Può essere utilizzato anche per le macro perché consente di mettere a fuoco gli oggetti anche a una distanza molto ravvicinata.





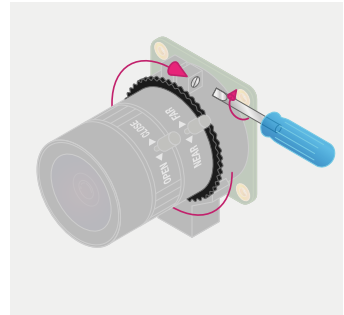
01 Come montare l'obiettivo

L'obiettivo da 6 mm è un dispositivo di montaggio CS, quindi non necessita dell'anello adattatore C-CS (vedi il grafico sopra). Se l'adattatore è montato, l'obiettivo non metterà a fuoco correttamente, quindi, se necessario, rimuovilo. Ruota l'obiettivo fino in fondo in senso orario nell'anello di regolazione della messa a fuoco posteriore.



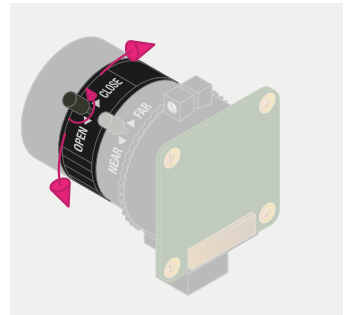
02 Anello di regolazione della messa a fuoco posteriore e vite di serraggio

L'anello di regolazione della messa a fuoco posteriore deve essere avvitato completamente fino a raggiungere la lunghezza di messa a fuoco posteriore più corta. Utilizza la vite di serraggio della messa a fuoco posteriore per assicurarti che non si sposti da questa posizione quando regoli il diaframma o la messa a fuoco.



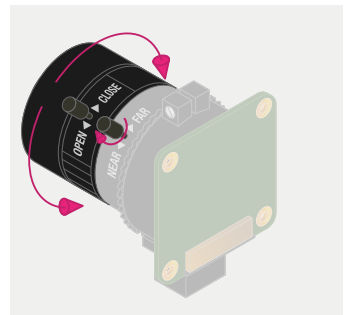
03 Diaframma

Per regolare il diaframma, tieni la fotocamera con l'obiettivo rivolto verso l'esterno. Ruota l'anello centrale tenendo fermo l'anello esterno, il più lontano possibile dalla fotocamera. Ruota in senso orario per chiudere il diaframma e ridurre la luminosità dell'immagine. Ruota in senso antiorario per aprire il diaframma. Quando sei soddisfatto della quantità di luce, serra la vite sul lato dell'obiettivo per bloccare il diaframma.



04 Messa a fuoco

Per prima cosa, blocca in posizione l'anello di messa a fuoco interno, contrassegnato con "NEAR ◀▶FAR", stringendo l'apposita vite. Impugna la fotocamera con l'obiettivo rivolto verso l'esterno. Ruota entrambi gli anelli esterni dell'obiettivo in senso orario fino a quando l'immagine non è a fuoco. Ci vorranno quattro o cinque rotazioni complete. Per regolare la messa a fuoco, ruota i due anelli esterni in senso orario per mettere a fuoco un oggetto vicino. Ruotali in senso antiorario per mettere a fuoco un oggetto lontano. Una volta regolata la messa a fuoco, potrebbe essere necessario regolare nuovamente il diaframma.



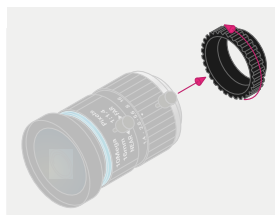
Obiettivo da 16 mm con attacco CS

L'obiettivo da 16 mm consente di scattare immagini di qualità superiore rispetto all'obiettivo da 6 mm. Ha un angolo di visuale stretto che è più adatto a inquadrare oggetti lontani.



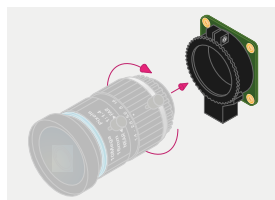
01 Come montare l'adattatore C-CS

Assicurati che l'adattatore C-CS fornito con la HQ Camera sia montato sull'obiettivo da 16 mm. Questo obiettivo presenta un attacco a C, quindi ha una messa a fuoco posteriore più lunga rispetto all'obiettivo da 6 mm e necessita dell'adattatore.



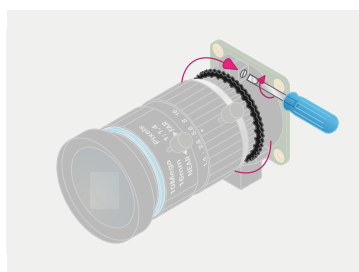
02 Come montare l'obiettivo sulla fotocamera

Ruota l'obiettivo da 16 mm e l'adattatore C-CS fino in fondo in senso orario nell'anello di regolazione della messa a fuoco posteriore.



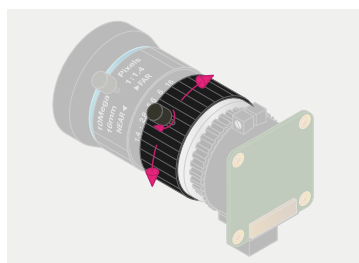
03 Anello di regolazione della messa a fuoco posteriore e vite di serraggio

L'anello di regolazione della messa a fuoco posteriore deve essere avvitato completamente. Utilizza la vite di serraggio della messa a fuoco posteriore per assicurarti che non si sposti da questa posizione quando regoli il diaframma o la messa a fuoco.



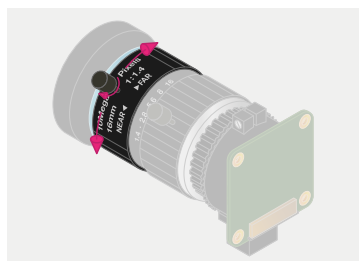
04 Diaframma

Per regolare il diaframma, tieni la fotocamera con l'obiettivo rivolto verso l'esterno. Tenendo ferma la fotocamera, ruota l'anello interno, quello più vicino alla fotocamera. Ruota in senso orario per chiudere il diaframma e ridurre la luminosità dell'immagine. Ruota in senso antiorario per aprire il diaframma. Quando sei soddisfatto della quantità di luce, serra la vite sul lato dell'obiettivo per bloccare il diaframma nella posizione desiderata.



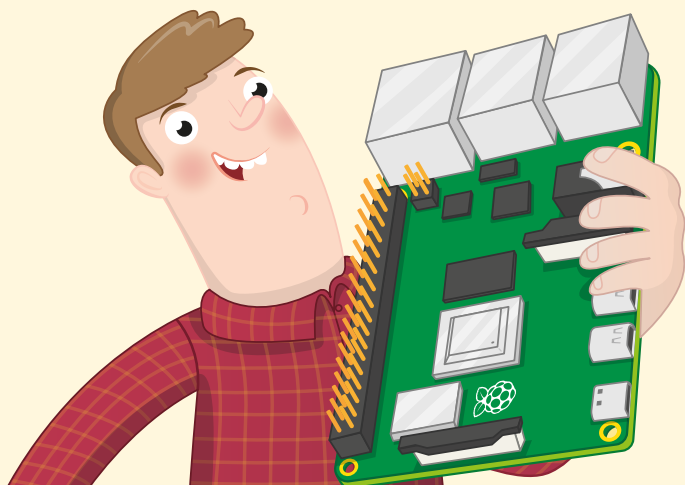
05 Messa a fuoco

Per regolare la messa a fuoco, impugna la fotocamera con l'obiettivo rivolto verso l'esterno. Ruota l'anello di regolazione, contrassegnato con "NEAR ◀▶ FAR", in senso antiorario per mettere a fuoco un oggetto vicino. Ruotalo in senso orario per mettere a fuoco un oggetto lontano. Una volta regolata la messa a fuoco, potrebbe essere necessario regolare nuovamente il diaframma.



Appendice G

Specifiche tecniche di Raspberry Pi



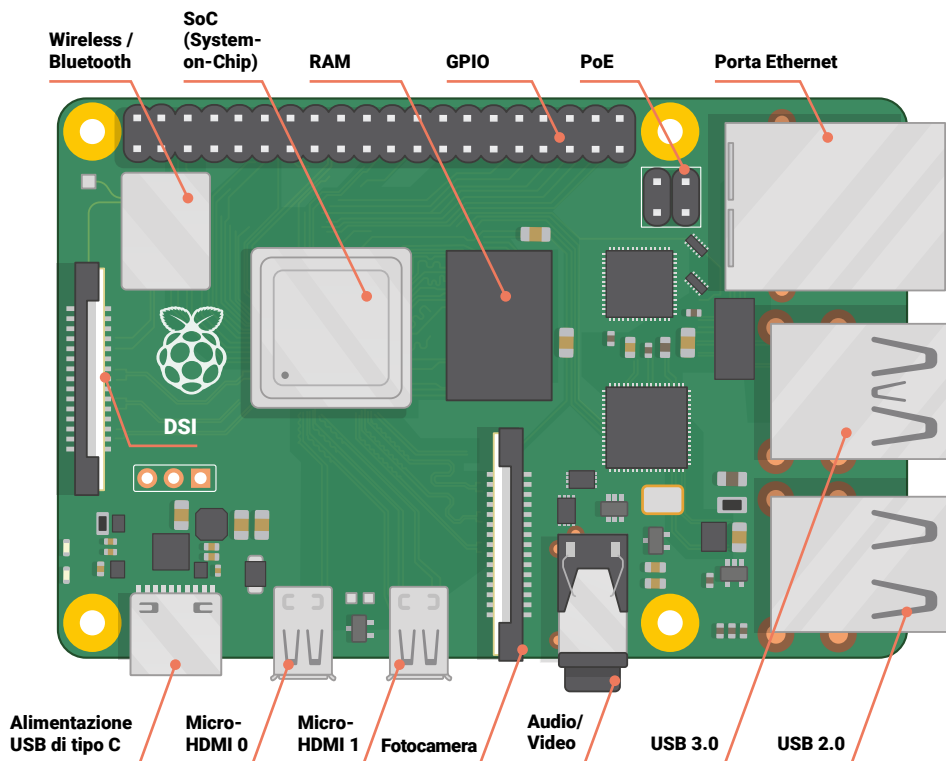
I vari componenti e le caratteristiche di un computer sono noti come "Specifiche tecniche" e forniscono le informazioni necessarie per mettere a confronto computer di tipologie diverse.

Tali specifiche possono sembrare confuse all'inizio, sono molto tecniche e non è necessario conoscerle per usare un Raspberry Pi, ma sono qui descritte per i lettori più curiosi.

Il sistema su chip del Raspberry Pi 4 Modello B e del Raspberry Pi 400 è un Broadcom BCM2711B0, una sigla che troverai scritta sul coperchio metallico (solo sul Raspberry Pi 4). Questo sistema è dotato di quattro core di unità di elaborazione centrale (CPU) ARM Cortex-A72 a 64 bit, ognuno dei quali funziona a 1,5 GHz o 1,8 GHz (1,5 o 1,8 miliardi di cicli al secondo), e di un'unità di elaborazione grafica (GPU) Broadcom VideoCore VI (Six) che funziona a 500 MHz (500 milioni di cicli al secondo) per video e rendering 3D come i giochi.

Il sistema su chip è dotato di RAM (memoria ad accesso casuale) LPDDR4 (Low-Power Double-Data-Rate 4) da 2 GB, 4 GB o 8 GB (due, quattro o ottomila milioni di byte) (4 GB su Raspberry Pi 400) che funziona a 3200 MHz (tremiladuecento milioni di cicli al secondo). Questa memoria è condivisa tra processore centrale e processore grafico. Lo slot per schede microSD supporta fino a 512 GB (512 miliardi di byte) di memoria.

La porta Ethernet supporta connessioni fino a 1 gigabit (1000 Mbps, 1000-Base-T), mentre la radio supporta le reti Wi-Fi 802.11ac che operano sulle bande di frequenza 2,4 GHz e 5 GHz, Bluetooth 5.0 e connessioni Bluetooth Low Energy (BLE).



Di seguito è fornito un elenco delle specifiche del Raspberry Pi 4:

- **CPU:** ARM Cortex-A72 quad-core a 64 bit a 1,5GHz
- **GPU:** VideoCore VI a 500 MHz
- **RAM:** 1 GB, 2 GB o 4 GB di LPDDR4
- **Rete:** Gigabit Ethernet, dual-band 802.11ac, Bluetooth 5.0, Bluetooth Low Energy
- **Uscite audio/video:** connettore jack AV analogico da 3,5 mm, 2 micro-HDMI 2.0
- **Connettività periferiche:** 2 porte USB 2.0, 2 porte USB 3.0, interfaccia seriale della fotocamera, interfaccia seriale del display (DSI)
- **Archiviazione dati:** microSD, fino a 512 GB
- **Alimentazione:** 5 volt a 3 ampere con USB Tipo-C
- **Extra:** terminale GPIO a 40 pin, compatibilità Power over Ethernet (con hardware aggiuntivo)



Le specifiche tecniche del Raspberry Pi 400 sono:

- **CPU:** ARM Cortex-A72 quad-core a 64 bit a 1,8 GHz
- **GPU:** VideoCore VI a 500 MHz
- **RAM:** 4 GB di LPDDR4
- **Reti:** Gigabit Ethernet, dual-band 802.11ac, Bluetooth 5.0, Bluetooth Low Energy
- **Uscite audio/video:** 2 micro-HDMI 2.0
- **Connettività periferiche:** 1 porta USB 2.0, 2 porte USB 3.0
- **Archiviazione dati:** microSD, fino a 512 GB (16 GB in dotazione)
- **Alimentazione:** 5 volt a 3 ampere con USB Tipo-C
- **Extra:** Terminale GPIO a 40 pin

Appendice H

Guida all'uso e avvertenze di sicurezza di Raspberry Pi



Raspberry Pi


Prodotto e distribuito da
Raspberry Pi Trading Ltd
Maurice Wilkes Building
Cowley Road
Cambridge
CB4 0DS
UK
www.raspberrypi.org

Conformità alle normative e informazioni sulla sicurezza di **Raspberry Pi**

Raspberry Pi 4 Modello B
FCC ID: 2ABCB-RPI4B
IC ID: 20953-RPI4B

Raspberry Pi 400
FCC ID: 2ABCB-RPI400
IC ID: 20953-RPI400

IMPORTANTE: prima di collegare l'apparecchio all'alimentazione, leggere attentamente le istruzioni di installazione su www.raspberrypi.org/safety

 **ATTENZIONE:** può provocare cancro o danni agli apparati riproduttivi - www.P65Warnings.ca.gov.

Tutte le informazioni relative alle normative e i certificati sono disponibili sul sito www.raspberrypi.org/compliance



IFETEL: 2019LAB-ANCE4957
Numero di certificato per Raspberry Pi 4 Modello B.



Numero di registrazione TRA
ER73381/19
Numero di certificato per Raspberry Pi 4 Modello B.



CCA019LP1120T2

Numero di certificato per Raspberry Pi 4 Modello B.



NTC
Omologato:
N°: ESD-GEC-1920098C
Numero di certificato per Raspberry Pi 4 Modello B.



TA-2019/750 OMOLOGATO
Numero di certificato per Raspberry Pi 4 Modello B.



I marchi adottati HDMI, HDMI High-Definition Multimedia Interface e il logo HDMI sono marchi o marchi registrati di HDMI Licensing Administrator, Inc. negli Stati Uniti e in altri paesi.



LA GUIDA INTRODUTTIVA ufficiale Raspberry Pi

Dal genio degli sviluppatori inglesi di Raspberry Pi, nasce un computer compatto, smart e ricco di potenzialità. Realizzato con la stessa tecnologia contenuta in uno smartphone, Raspberry Pi è stato progettato per aiutarti a imparare il linguaggio di codifica, a scoprire come funzionano i computer e a costruire le tue personalissime meraviglie tecnologiche. Questa guida è stata scritta per mostrarti quanto sia facile iniziare a usarlo.

Scopri come:

- > impostare il tuo Raspberry Pi, installare il suo sistema operativo e iniziare a utilizzare questo computer completamente funzionale;
- > cominciare a codificare progetti seguendo procedure dettagliate che utilizzano i linguaggi di programmazione Scratch 3 e Python;
- > collegare componenti elettronici e divertirti a creare incredibili progetti.

raspberrypi.org

