

Introduzione

Prima di iniziare è il caso di spendere qualche parola per spiegarti come è fatto questo libro, a cosa serve ogni capitolo e via dicendo.

A chi si rivolge questo libro

Questo libro si rivolge a tutti quelli che vogliono imparare a programmare. Nessuno escluso. Dai 9 ai 99 anni. Con l'aiuto dei genitori possono leggere questo libro anche i bambini sotto i 9 anni.

Requisiti per la lettura

Non ci sono requisiti per la lettura di questo libro, oltre quello di avere un computer ed essere collegati a Internet. Non è necessario avere una connessione permanente, ma servirà per scaricare e installare Scratch 3.0, condividere i propri progetti e vedere quelli degli altri.

Struttura del libro

Il libro contiene molte figure, che ti aiuteranno a capire come sono fatti i programmi in Scratch e cosa otterrai eseguendoli.

L'impostazione sarà per lo più basata sul *learning by doing* (imparare facendo) e *learning by design* (imparare progettando). D'altronde, i *programmi* di Scratch si chiamano anche *progetti*.

L'idea è semplice e consiste in tre, a volte quattro, passaggi.

- Introduzione di un progetto da costruire (obiettivo).
- Presentazione di una possibile realizzazione del progetto (implementazione).
- Presentazione di elementi di Scratch, nozioni teoriche della programmazione e altri strumenti che servono a raggiungere l'obiettivo (teoria, linguaggio e strumenti).
- Proposta di alcuni spunti per migliorare o estendere quanto costruito.

I concetti e gli strumenti sono affrontati in funzione della risoluzione di problemi specifici (*problem solving*). In qualche capitolo troverai alcune ripetizioni, utili comunque a riprendere e approfondire gli argomenti meno semplici.

Se trovi qualche errore, ci scusiamo in anticipo e ti chiediamo cortesemente di segnalarlo scrivendo a libri@apogonline.com.

Box

Troverai alcuni box, riconoscibili dallo sfondo azzurro, con approfondimenti sul tema affrontato in un determinato punto del testo. Puoi leggerli o decidere di ignorarli se non ti interessa saperne di più su quell'argomento.

NOTA

Ci saranno qua e là anche delle **note**, più brevi dei box e accompagnate da una caratteristica icona, con suggerimenti e consigli per capire meglio i contenuti trattati ed evitare gli errori più comuni. Per questo motivo è meglio non ignorarle.

In fondo a ogni capitolo c'è un piccolo riassunto, utile a monitorare i tuoi progressi.

Presentazione dei capitoli

Capitolo 1 – Cos'è Scratch: presenta Scratch, la sua storia, la sua filosofia e la comunità di appassionati e programmatori che ruota attorno a questo progetto. Spiega, inoltre, come installarlo e usarlo online e offline e descrive gli elementi principali di Scratch – *stage*, *sprite* e *script* – e il suo sistema di coordinate cartesiane.

Capitolo 2 – Disegni e animazioni: introduce i blocchi di Scratch e presenta programmi per il disegno e l'animazione con istruzioni delle categorie *Movimento*, *Situazioni*, *Controllo* e *Penna*. Il capitolo affronta anche alcune tematiche generali della programmazione, come le strutture di controllo sequenza e ciclo, gli algoritmi e gli errori di sintassi e logici.

Capitolo 3 – Sprite e fumetti animati: introduce i concetti di programmazione concorrente e guidata dagli eventi e di comunicazione e sincronizzazione

attraverso lo scambio di messaggi, presenta progetti per l'animazione degli *sprite* con più tecniche (effetti grafici, cambi di costume ecc.) e aggiunge i suoni, guidandoti nella realizzazione di storie e fumetti animati a partire dalla progettazione di uno *storyboard*, passando attraverso la creazione dei personaggi, per arrivare alla scrittura degli script per riprodurre l'azione prevista nella sceneggiatura.

Capitolo 4 – Quiz e giochi: attraverso la creazione di quiz e semplici videogiochi spiega la struttura di controllo selezione, o biforcazione (se... altrimenti), i numeri pseudocasuali, le immagini digitali raster e vettoriali, la programmazione procedurale, la progettazione *top-down* e *bottom-up*, le procedure, le variabili e l'input da tastiera.

Capitolo 5 – Giochi, cloni e stringhe: utilizza alcuni giochi, più complessi di quelli del capitolo precedente, animazioni e quiz con immagini per affrontare l'argomento delle stringhe e la clonazione degli *sprite*.

Capitolo 6 – Matematica e musica: ti fa capire come la matematica sia necessaria per la programmazione, soprattutto se si vuole fare qualcosa di impegnativo come un'applicazione di tipo statistico. Il capitolo approfondisce i concetti di variabile, tipo di dato, valore e operazione (matematica, relazionale e booleana), introduce le liste e approfondisce le funzionalità dei blocchi della categoria **Suono** con la creazione di programmi legati alla musica.

Capitolo 7 – Arcade: i videogiochi 2D: ti guida nella programmazione di alcuni videogiochi arcade 2D che mettono assieme quanto appreso nel resto del testo, approfondendo, in particolare, le collisioni tra *sprite*, l'input tramite tastiera e mouse e la gestione di più scenari e del punteggio del gioco.

Capitolo 8 – L'ecosistema Scratch, estensioni e funzionalità avanzate: conclude il libro con due semplici programmi che fanno uso della webcam, spiega come distribuire i progetti creati con Scratch e illustra brevemente l'ecosistema che è nato attorno a questo progetto, fatto di community, programmi didattici ed estensioni per aggiungere nuove funzionalità e collegarlo ad alcuni sistemi hardware.

L'arte della programmazione

L'informatico americano Donald Knuth è famoso, tra l'altro, per aver scritto una serie di libri intitolati *The Art of Computer Programming*, dedicati all'analisi degli algoritmi.

La programmazione è, infatti, un connubio di arte e tecnica, di teoria e pratica. Spesso ci sono più modi per arrivare alla soluzione di un problema: alcuni sono più semplici, altri più veloci, alcuni utilizzano meno risorse del computer, altri semplicemente sono più "eleganti"!

Coding e computational thinking a scuola

Negli ultimi anni, in molti stati, tra cui l'Italia, i sistemi scolastici hanno riconosciuto l'importanza del pensiero algoritmico, o computazionale (*computational thinking*), e della programmazione (*computer programming* o *coding*) per la formazione e l'educazione fin dalla scuola dell'infanzia. L'idea è che le conoscenze e le competenze informatiche non si possano limitare all'uso di dispositivi e programmi, ma debbano andare oltre, arrivando, appunto, a insegnare i principi dell'informatica e della programmazione.

Scratch è un punto fermo di questa rivoluzione pedagogica e culturale ed è presente in quasi tutti i programmi, governativi e non, per passare dall'alfabetizzazione digitale all'alfabetizzazione informatica vera e propria.

Un esempio di iniziative di questo tipo, The Hour of Code (<https://hourofcode.com>) del Ministero dell'Istruzione americano, è stato portato in più di 180 nazioni. In Italia l'Ora del Codice (<https://programmairfuturo.it/come/ora-del-codice>) è parte del progetto Programma il Futuro del Ministero dell'Istruzione dell'Università e della Ricerca.

Progetti presentati nel libro

Tutti i progetti presentati nel testo o proposti come esercizi sono reperibili sul sito di Scratch, accedendo ai progetti dell'utente Scratch3Libro all'indirizzo <https://scratch.mit.edu/users/Scratch3Libro>. Inoltre, una copia dei progetti e i file per le immagini e i suoni utilizzati nel testo sono reperibili sul sito di Apogeo, all'indirizzo <https://bit.ly/apo-scratch-3-0>. Indicheremo queste risorse come *File degli esempi*. Scarica e decomprimi il relativo file ZIP sul Desktop o in un'altra cartella del tuo PC.

Credits

Software

Scratch (<https://scratch.mit.edu>), sviluppato dal Lifelong Kindergarten Group al MIT Media Lab del Massachusetts Institute of Technology.

Inkscape (<https://inkscape.org>), editor per la grafica vettoriale gratuito, open source e multiplatforma.

GIMP (<https://gimp.org>), programma per l'elaborazione di immagini raster gratuito, open source e multiplatforma.

Progetti

I seguenti programmi Scratch sono tratti da lavori di Alberto Montresor, docente di algoritmi all'Università di Trento e anima del CoderDojo Verona, di cui è cofondatore con altre persone, tra le quali c'è anche l'autore di questo testo.

- **Un messaggio stellare**
- **Un testo telegrafico**
- **Pong**
- **Can che abbia non dorme**
- **Il rompibolle**

Asteroids, illustrato nel Capitolo 7, è un riadattamento di un progetto di Francesco Fornasa, sempre del CoderDojo Verona.

Il programma **Un compleanno sorprendente**, illustrato nel Capitolo 3, prende spunto da una bella e originale storia di Camilla Tognoli, quando nel 2010 frequentava la classe IV B della scuola primaria Manzoni di Verona con l'attiva e innovativa maestra Laura Ciogli.

Immagini e suoni

Laddove nel testo non è espressamente indicata la fonte (in base ai termini di licenze come la Creative Commons Attribution-Share Alike, CC BY-SA, <http://creativecommons.org/licenses/by-sa/3.0/deed.it>), i file per le immagini e i suoni utilizzati nei progetti del libro sono di pubblico dominio o richiedono l'attribuzione della paternità dell'autore e sono scaricabili da siti quali <https://pixabay.com>, <https://opengameart.com>, <https://www.freepik.com>, <https://www.kenney.nl> e <https://www.vecteezy.com>.

Per i giochi **Color Pong** e **Arkanoid** sono state utilizzate le immagini di Ari Feldman rese gentilmente disponibili all'indirizzo <https://widgetworx.com/sprite-lib>.

Ringraziamenti

Questo libro ha visto la luce grazie al contributo di molte persone, tra cui: Lorenzo Boscaini, piccolo ninja che ha “eseguito” il libro e i suoi programmi, Luca Boscaini, fratello dell'autore, e Flavia Lughezzani, che hanno minuziosamente rivisto le bozze (qui abbiamo terminato con parenti e amici), il FabLab di Verona (www.veronafablab.it) per l'apertura e la disponibilità anche nell'accogliere le attività del CoderDojo e, infine, Marco Beri, che ha condiviso la nascita di questa avventura.