

INTRODUZIONE

Che cos'è la programmazione

Per sapere cosa fare, i computer e i dispositivi elettronici hanno bisogno di software (o programmi), e scrivere queste istruzioni è compito della programmazione, detta anche coding. Benché esistano programmatori professionisti, il coding può anche essere un hobby.

I programmi informatici sono ovunque

La programmazione non riguarda più soltanto i sistemi informatici convenzionali. Il mondo sta diventando sempre più digitale e quasi tutto funziona grazie a software, incorporati in dispositivi come smartphone e tablet, elettrodomestici e perfino sistemi di trasporto.

DIVENTARE UN CODER

Una volta appresi i rudimenti della programmazione, i consigli seguenti vi aiuteranno a sviluppare ulteriormente le vostre competenze.

- **Scrivere:** Scrivete e sperimentate codici
- **Leggere:** Si può imparare molto studiando i programmi di altri
- **Apprendere vari linguaggi:** Conoscere le idee e i concetti alla base di altri linguaggi aiuta i programmatori a scegliere l'opzione migliore per ogni progetto
- **Pubblicare progetti:** Lanciare il progetto online e ottenere feedback da altri coder vi aiuterà a scrivere codici migliori

Centro dati

Un centro dati, o data centre, somiglia a un sistema informatico su larga scala. I molti server di cui dispone consentono l'accesso via Internet per immagazzinare dati o utilizzare software sul cloud, una rete globale di server remoti che fornisce servizi attraverso Internet.

Vettura

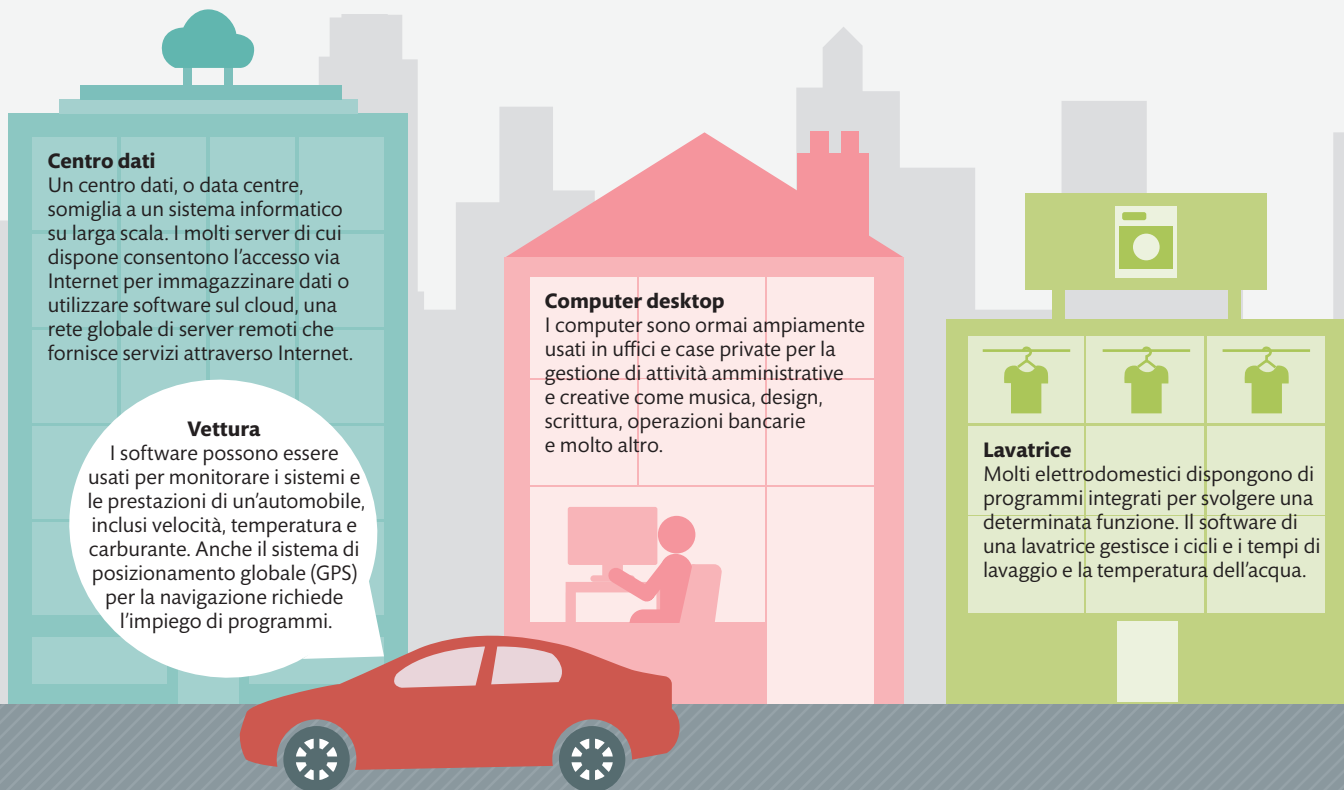
I software possono essere usati per monitorare i sistemi e le prestazioni di un'automobile, inclusi velocità, temperatura e carburante. Anche il sistema di posizionamento globale (GPS) per la navigazione richiede l'impiego di programmi.

Computer desktop

I computer sono ormai ampiamente usati in uffici e case private per la gestione di attività amministrative e creative come musica, design, scrittura, operazioni bancarie e molto altro.

Lavatrice

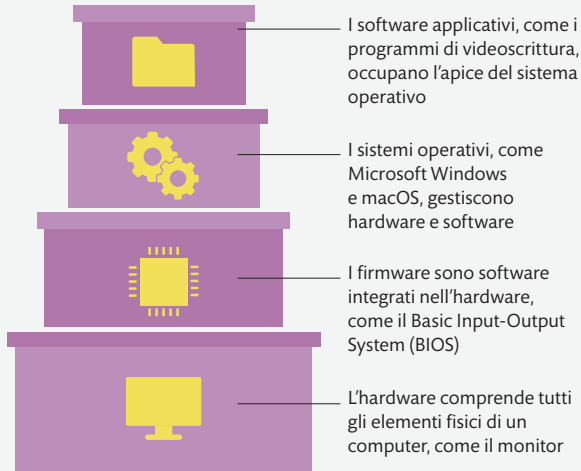
Molti elettrodomestici dispongono di programmi integrati per svolgere una determinata funzione. Il software di una lavatrice gestisce i cicli e i tempi di lavaggio e la temperatura dell'acqua.





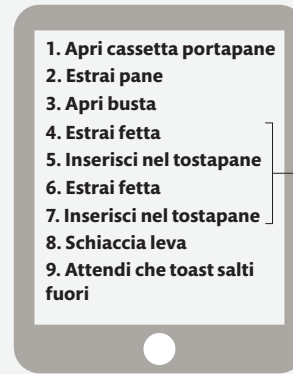
Che cos'è un programma informatico

È l'insieme delle istruzioni che un computer esegue per svolgere un compito. Può essere estremamente complesso, e a volte prevede il funzionamento simultaneo di più livelli. Microsoft Windows, per esempio, è composto da milioni di righe di istruzioni.

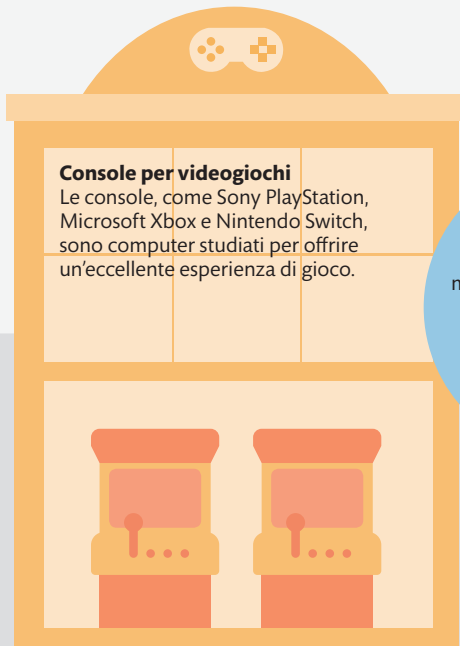


Pensare come un computer

Per scrivere un programma bisogna capire il modo in cui il computer processa le istruzioni. Per essere capiti ed eseguiti, i compiti devono essere suddivisi in piccole porzioni. Per esempio, a un robot non si può chiedere di "fare dei toast": è necessario programmare istruzioni precise e dettagliate per ogni fase.

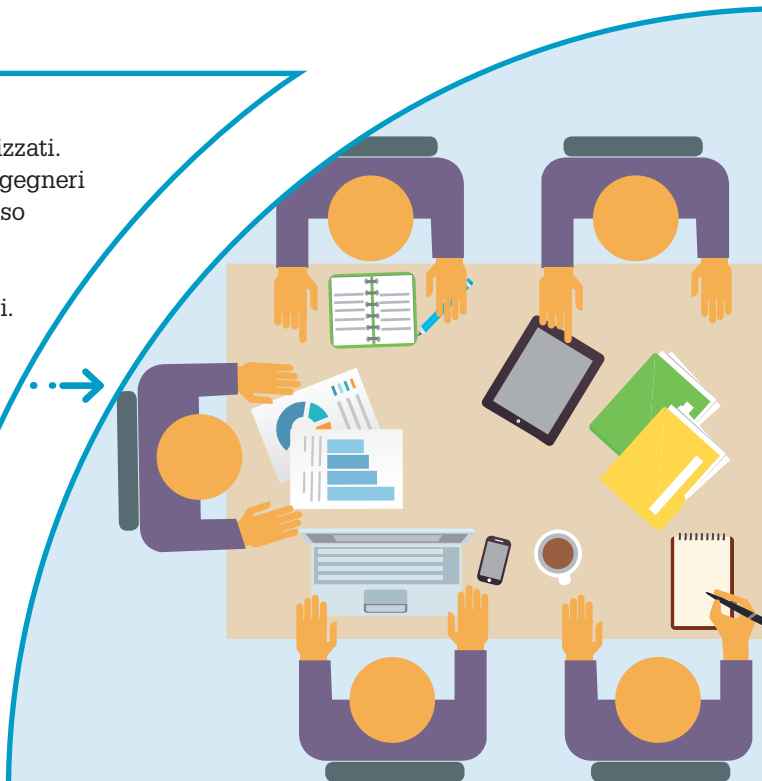


Invece di ripetere la stessa istruzione due volte, è più breve e chiaro dire «Fai due volte: estrai fetta, inserisci nel tostapane»



Manager e impiegati

Molte imprese creano e usano programmi specializzati. I sistemi software complessi sono sviluppati da ingegneri informatici, mentre i manager e gli impiegati spesso creano software semplici per risolvere problemi specifici o automatizzare mansioni che, se svolte manualmente, richiederebbero ore o perfino giorni. Avvalendosi di linguaggi di programmazione specifici, per esempio, possono creare codici per consultare database, formattare o analizzare dati, monitorare attrezzature o aggiungere funzioni a programmi di videoscrittura e a fogli elettronici (cfr. pp. 344-347).



Artisti e appassionati

Il coding può favorire molte forme di creatività. Gli artisti possono creare software per fare musica o realizzare opere d'arte con una componente interattiva; gli appassionati possono realizzare videogiochi o storie interattive, gestire semplici robot, controllare l'illuminazione o svolgere mansioni domestiche.



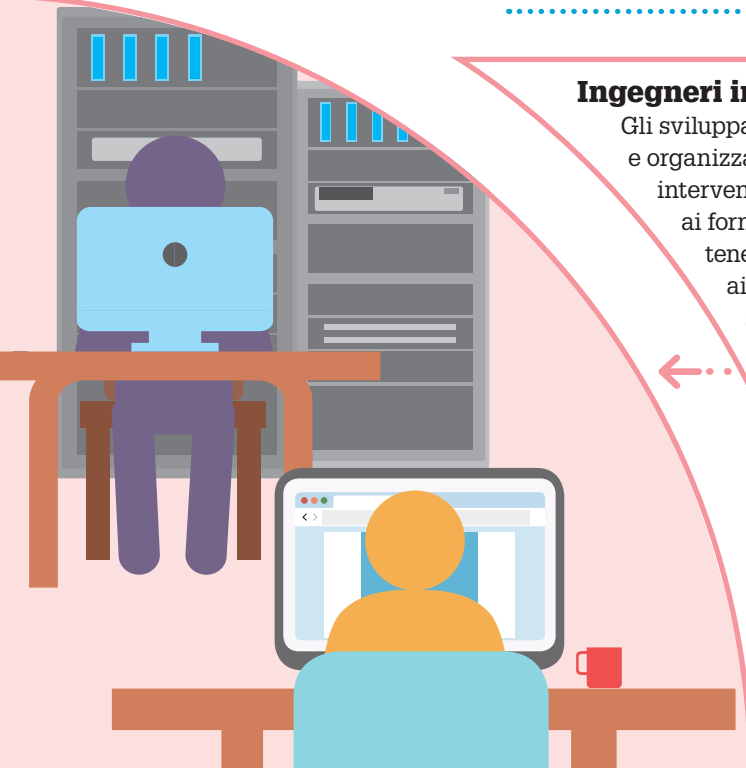
Coding nel mondo reale

Il coding fa ormai parte di quasi ogni aspetto della vita e del lavoro moderni, e possedere conoscenze basiche di programmazione consente di usare i software con maggiore efficienza, creare programmi semplici e comunicare con altri sviluppatori.



Ingegneri informatici e sviluppatori Web

Gli sviluppatori di software lavorano per imprese e organizzazioni di diverso tipo, che richiedono il loro intervento per gestire l'inventario ed effettuare ordini ai fornitori, organizzare i turni dei dipendenti e tenere nota delle ore lavorative, o inviare messaggi ai clienti. Le compagnie assicurative usano software per stabilire il premio in base al numero e al costo dei sinistri e per esaminare e approvare le polizze. I siti Web spesso affiancano ai programmi esistenti una codifica personalizzata per determinate mansioni, e gli ingegneri informatici svolgono un ruolo cruciale nello sviluppo di sistemi che soddisfino le esigenze della clientela.



Scienziati e ricercatori

Il coding è anche usato per condurre esperimenti, analizzare dati e creare referti medici. Per esempio, i neuroscienziati usano programmi per mostrare forme o parole a un paziente, registrare l'attività cerebrale e analizzare i dati per individuare le aree più attive del cervello.



25.000.000

**LAVORATORI IN TUTTO
IL MONDO SONO**

SVILUPPATORI DI SOFTWARE

Lavorare come sviluppatore di software

Creare un programma basico può sembrare un gioco da ragazzi, ma potrebbe rivelarsi sorprendentemente difficile. Sviluppare sistemi software grandi e affidabili è un compito ancora più arduo, e richiede team di esperti con competenze e ruoli specialistici.

Analisi

Gli sviluppatori stabiliscono cosa fa il software, per esempio studiando i sistemi esistenti, progettando processi nuovi o intervistando potenziali utenti per capire l'uso che ne faranno. Inoltre, in questa fase emergono vincoli e requisiti: quanti dati deve poter gestire, qual è la velocità di risposta richiesta e cosa dovrebbe fare in caso di problemi. I documenti che verranno prodotti possono comprendere da qualche foglio a centinaia di pagine.



Collaudo

Gli sviluppatori verificano il corretto funzionamento del software e risolvono i problemi in cui incorrono durante i test. Si tratta spesso della fase più lunga e meno prevedibile, causa frequente di ritardi e di costi aggiuntivi. Esistono molti tipi di test: alcuni eseguono il collaudo di ogni funzione (test di unità), altri verificano i singoli componenti (test funzionali) o il loro funzionamento armonico (test di integrazione), altri ancora l'efficienza dell'intero sistema (test di sistema).





Panoramica

Lo creazione di un software comprende quattro fasi – l'analisi, il progetto e la pianificazione, lo sviluppo e il collaudo – che possono strutturarsi in vario modo. Il modello a cascata passa in rassegna le varie fasi una sola volta ma, pur sembrando semplice, è spesso causa di problemi. Il modello incrementale esegue le fasi varie volte, costruendo una parte del sistema a ogni ciclo, mentre in quello agile ogni fase è ripetuta innumerevoli volte, aggiungendo a ogni ciclo nuove funzionalità.

Progetto e pianificazione

Gli sviluppatori decidono come funzionerà e come verrà creato il software. Per esempio, possono scegliere la lingua, realizzare schizzi delle interfaccia utente, progettare i database, suddividerli in ambiti e specificare quali sono

i file e le singole funzioni da creare. È inoltre necessario stabilire la tempistica, lo sforzo, i materiali e i costi richiesti dalla realizzazione, e determinare chi si occuperà di quale mansione e in quale fase.



Sviluppo

Gli sviluppatori creano il software, inclusi le interfaccia utente, i database, il codice e la documentazione per utenti e programmatori. Il coding, pertanto, costituisce solo una parte di una fase della realizzazione di un software, e in un certo senso la più semplice e prevedibile. Una volta ultimato ogni elemento, è necessario esaminare e valutare il codice per verificarne la correttezza e l'efficienza, poi integrarlo nel sistema complessivo.



Linguaggi di programmazione

Un linguaggio di programmazione è un insieme di parole e simboli che servono a scrivere le istruzioni per un computer. Spesso si arriva a un compromesso tra la semplicità d'uso del linguaggio e la sua efficacia.

Linguaggi ad alto e a basso livello

I linguaggi di programmazione ad alto livello sono studiati per chi non ha una comprensione approfondita dell'hardware. Spesso usano parole simili al linguaggio umano e gestiscono automaticamente alcuni aspetti del computer; in genere, questi programmi funzionano

su vari hardware. I linguaggi a basso livello, al contrario, offrono ai programmatori un controllo granulare sul computer, ma richiedono conoscenze minuziose del suo funzionamento; i programmi scritti in linguaggi a basso livello possono non essere compatibili con hardware diversi.

LINGUAGGI AD ALTO LIVELLO

- Relativa rapidità di scrittura
- Relativa facilità di comprensione
- Spesso la velocità di esecuzione è abbastanza alta
- Possono essere usati su hardware diversi
- Non richiedono la conoscenza dell'hardware

Istruzione per visualizzare un testo

```
print("Hello!")
```

Python

Linguaggio ad alto livello molto diffuso, Python è facile da leggere e da scrivere. L'istruzione visualizza il messaggio "Hello!" sullo schermo.

LINGUAGGI A BASSO LIVELLO

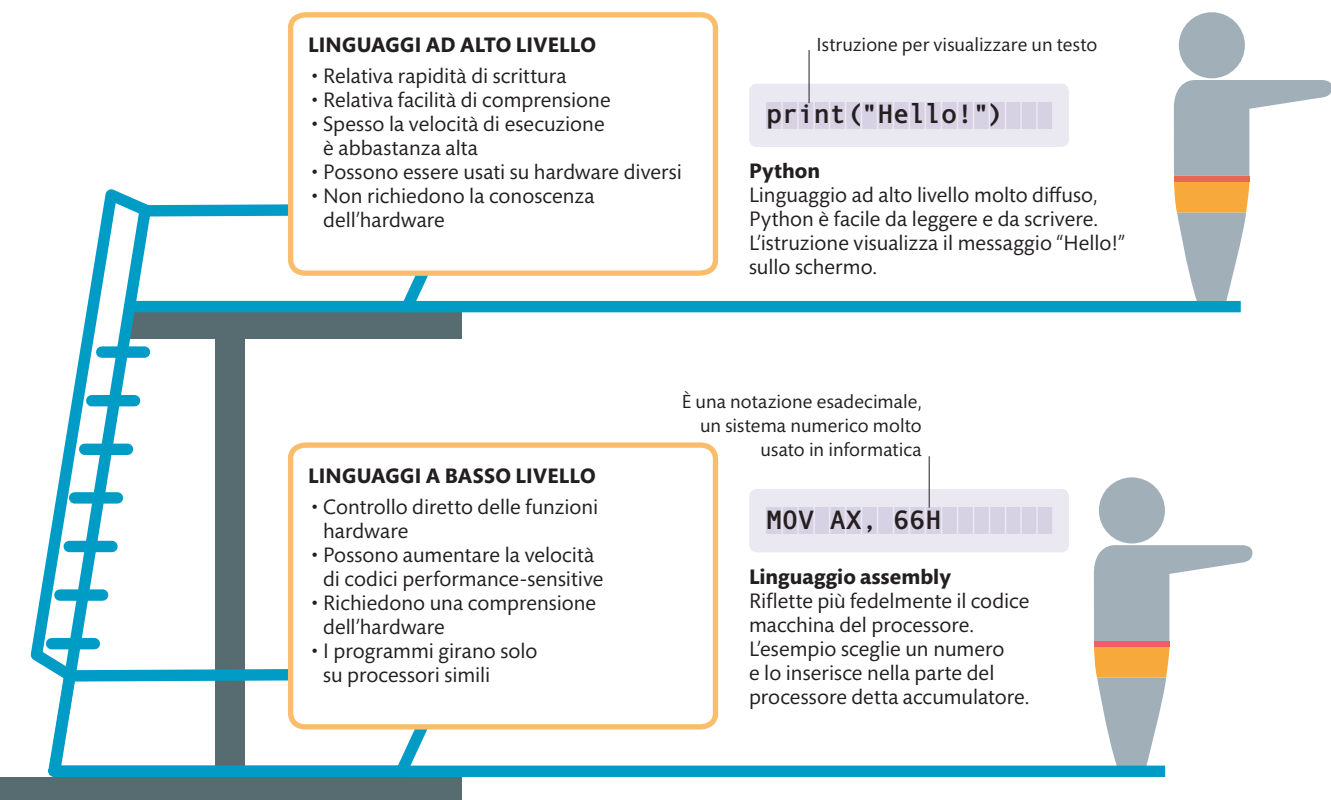
- Controllo diretto delle funzioni hardware
- Possono aumentare la velocità di codici performance-sensitive
- Richiedono una comprensione dell'hardware
- I programmi girano solo su processori simili

È una notazione esadecimale, un sistema numerico molto usato in informatica

```
MOV AX, 66H
```

Linguaggio assembly

Riflette più fedelmente il codice macchina del processore. L'esempio sceglie un numero e lo inserisce nella parte del processore detta accumulatore.



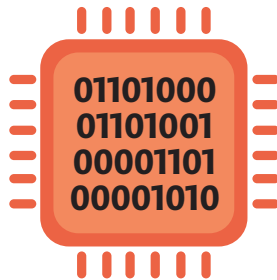


Codice macchina

Il codice a basso livello che rappresenta il modo in cui vengono impartiti i comandi all'hardware e al processore è detto codice macchina. Si tratta di un insieme di cifre binarie (composte da 1 e 0), lette e interpretate dal processore. Le istruzioni in codice macchina sono formate da un opcode (codice operativo) e da uno o più operandi: il primo dice al computer cosa fare, i secondi quali dati usare.

Microprocessore

Il microprocessore è il "cervello" del computer e controlla la maggior parte delle operazioni. Esegue i comandi e segue le istruzioni del codice.



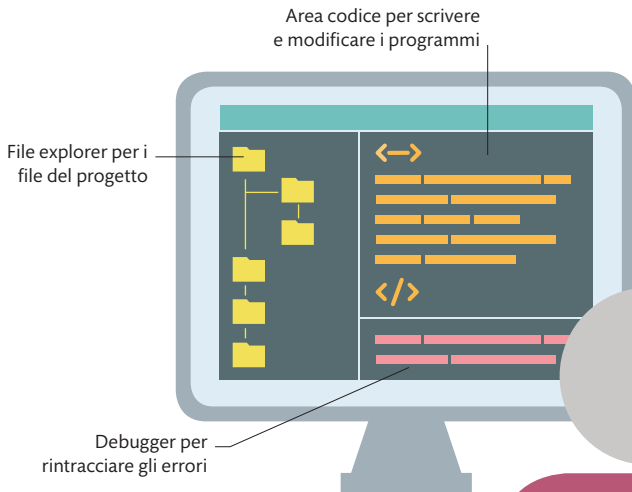
Come il computer comprende il linguaggio di programmazione

Alla fine, tutti i programmi sono ridotti al codice macchina. Molti, pur utilizzando linguaggi facili da capire, per poter essere compresi dal processore devono essere tradotti in termini di bit. L'interprete trasforma ed esegue le istruzioni durante il funzionamento del programma, mentre il compilatore traduce il programma in una sola volta, prima dell'avvio.



Usare un ambiente di sviluppo integrato (IDE)

Un IDE è un insieme di strumenti a disposizione dei programmatori. Include un editor di codice per la scrittura dei programmi ed è spesso dotato di funzioni che aumentano la produttività, come l'autocompletamento per le istruzioni e un codice cromatico per favorire la leggibilità. Alcuni prevedono un debugger per rintracciare gli errori e un compilatore o un interprete per testare ed eseguire i programmi.



APPLICAZIONI

Dopo aver imparato a programmare, le competenze elencate possono essere usate per un ampio ventaglio di progetti utili e creativi.

- **Domotica:** l'applicazione all'ambiente domestico, per esempio per gestire luci o tende
- **Videogiochi:** i videogiochi sono un modo fantastico per sperimentare con il coding, che favorisce la condivisione e il ricevimento di feedback (cfr. pp. 80-91, 178-203)
- **Robot:** usando schede Arduino o Raspberry Pi insieme a kit o componenti elettronici, è possibile programmare il proprio robot
- **Siti e app Web:** usando HTML, CSS e JavaScript (cfr. pp. 210-343) possono essere creati programmi che girano su ogni Web browser

Esempio di configurazione di un IDE

Gli IDE spesso consentono agli utenti di configurare il set-up. Nell'esempio, il programmatore sfoglia i file del progetto sulla sinistra, inserisce il codice e lo modifica sulla destra, ed esegue il debug in fondo.

Tipologie di linguaggi di programmazione

Negli anni sono stati messi a punto svariati paradigmi o filosofie per progettare i linguaggi di programmazione che, non escludendosi a vicenda, spesso incorporano vari principi di base e possono essere usati in vario modo secondo l'approccio preferito dal programmatore. Python, per esempio, può essere usato sia per la programmazione orientata agli oggetti, sia per quella procedurale; JavaScript per un paradigma orientato sia agli oggetti, sia agli eventi. La scelta dell'approccio o del linguaggio spesso dipende dalle preferenze del programmatore. Ecco alcune definizioni e classificazioni dei linguaggi di programmazione.

PROGRAMMAZIONE IMPERATIVA

I linguaggi imperativi richiedono una lista di comandi che il computer deve compiere. Il programmatore studia come va eseguito il compito, poi fornisce istruzioni passo passo al dispositivo. Si tratta di linguaggi diffusi che includono Python (cfr. pp. 94-95), C, C++ e Java.

```
user = input("What's your name?")  
print("Hello", user)
```

INPUT IN PYTHON

Programma Python per salutare un utente con il suo nome

```
What's your name? Sean  
Hello Sean
```

OUTPUT IN PYTHON

PROGRAMMAZIONE DICHIARATIVA

I programmatori comunicano al computer il risultato desiderato, senza specificare come verrà raggiunto. Nel Linguaggio Wolfram, per esempio, usando una sola riga è possibile creare una nuvola di etichette basata sulle parole della pagina di Wikipedia dedicata alla musica. Altri linguaggi dichiarativi includono SQL, usato per i database.

```
WordCloud[WikipediaData["music"]]
```

INPUT IN WOLFRAM



OUTPUT IN WOLFRAM

PROGRAMMAZIONE ORIENTATA AGLI EVENTI

In questo caso, il programma attende che accadano determinati eventi e, quando ciò avviene, avvia la sequenza corretta. Per esempio, un programma può reagire alle azioni dell'utente, all'input di un sensore o a messaggi di altri sistemi informatici. Per scrivere programmi orientati agli eventi, possono essere usati, tra gli altri, JavaScript (cfr. pp. 264-265) e Scratch (cfr. pp. 28-29).

Crea un pulsante di una pagina Web

```
<input type="button" value="Click me!"  
onClick="showMessage();" >
```

Quando si clicca il pulsante, vengono eseguite le istruzioni JavaScript `showMessage()`

Il libro usa la freccia per indicare che il codice è ripartito su due righe

SCEGLIERE UN LINGUAGGIO

La scelta del linguaggio da parte del programmatore può essere dettata dall'hardware adottato, dal team di lavoro o dal tipo di applicazione da creare, e spesso è aperta. Ecco alcuni esempi di linguaggi diffusi.

PYTHON

Linguaggio flessibile che privilegia la facilità di comprensione del codice.

JAVA

Molto usato dai servizi finanziari, dai piccoli dispositivi e dagli smartphone Android.

SCRATCH

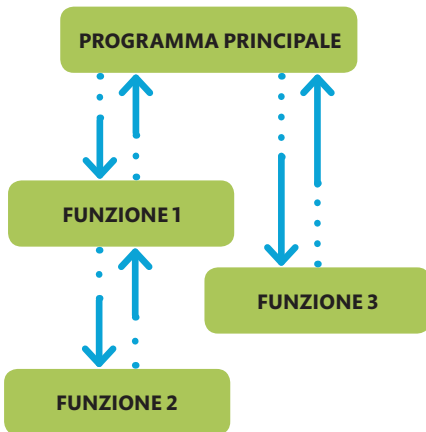
Ottimo linguaggio di programmazione per principianti, ideale per creare videogiochi semplici.

JAVASCRIPT

Linguaggio usato dalle pagine Web per l'interattività.

PROGRAMMAZIONE PROCEDURALE

È un paradigma basato su funzioni che contengono blocchi di programma riutilizzabili. Le funzioni possono avviare in ogni momento altre funzioni o riavviare se stesse, agevolando lo sviluppo, il collaudo e la gestione dei programmi. Molti linguaggi di programmazione diffusi, come Java e Python (cfr. pp. 94-95), supportano la programmazione procedurale.



PROGRAMMAZIONE ORIENTATA AGLI OGGETTI

L'idea di base è che i dati e le relative istruzioni sono conservati insieme in "oggetti", che possono interagire gli uni con gli altri per raggiungere gli obiettivi del programma. Lo scopo è realizzare un codice più modulare, facile da gestire e riutilizzabile. Molti linguaggi di programmazione diffusi, come C++, JavaScript e Python, supportano la programmazione orientata agli oggetti.



LINGUAGGI DI PROGRAMMAZIONE VISUALE

Sono linguaggi che agevolano lo sviluppo di un software usando una interfaccia a trascinamento, in modo da consentire al programmatore di velocizzare il lavoro e ridurre gli errori. Visual Basic, per esempio, prevede strumenti per progettare visivamente le interfacce utente. Scratch (cfr. pp. 28-29) è un altro linguaggio altamente visivo, spesso usato per imparare a programmare.

Programma Scratch che reagisce quando si clicca un pulsante

quando si clicca questo sprite

dire Pulsante cliccato! per 2 secondi