

Primi passi con JavaScript

1

In questo capitolo

- Introduzione alla programmazione (o coding)
- Introduzione a JavaScript
- Strumenti di lavoro
- Che cosa hai imparato

In questo capitolo introduciamo alcuni concetti fondamentali come informatica, computer, programmazione e programma, algoritmo e il linguaggio di programmazione JavaScript. Inoltre, presentiamo gli strumenti per lo sviluppo di codice JavaScript.

Introduzione alla programmazione (o coding)

L'*informatica* è la scienza che si occupa del trattamento e dell'elaborazione dell'informazione. Di fronte a un problema, essa distingue le figure del *risolutore* e dell'*esecutore*. Il risolutore analizza il problema e trova un modo concreto per risolverlo attraverso una serie precisa di passaggi. L'esecutore, umano o artificiale, seguendo le istruzioni che gli sono state impartite, mette in atto una serie di azioni che operano sui dati di partenza e portano ai dati finali. Al risolutore, infine, spetta anche il compito di controllare i risultati per capire se il problema sia stato effettivamente risolto oppure no.

Un *computer* è una macchina per gestire e trasformare informazioni. Come si può vedere nello schema generale della Figura 1.1, un computer acquisisce dati in ingresso (input), li elabora in base alle istruzioni contenute in un programma informatico e produce dei dati in uscita (output).



Figura 1.1 Schema base di esecuzione di un programma in un computer, o sistema di elaborazione elettronico.

Algoritmi, programmi e linguaggi

Un *programma* (informatico) è una sequenza di istruzioni scritte in un linguaggio di programmazione che dice all'esecutore che cosa fare, esattamente come una ricetta di cucina contiene le istruzioni per la preparazione di una pietanza oppure uno spartito contiene tutte le indicazioni per eseguire un brano musicale.

I linguaggi di programmazione

Il panorama dei linguaggi di programmazione è piuttosto affollato: Wikipedia (https://it.wikipedia.org/wiki/Lista_dei_linguaggi_di_programmazione) ne elenca ben 707, senza contare i “dialetti”.

Alcuni linguaggi sono più simili ai meccanismi di funzionamento interni del computer (*linguaggi di basso livello*), altri sono più vicini al linguaggio naturale

utilizzato dagli esseri umani. Tutti sono caratterizzati dalla *non ambiguità*: le parole e le istruzioni si interpretano sempre in modo univoco. Per impartire comandi a un computer non possiamo utilizzare il linguaggio naturale. Infatti la lingua italiana, come gli altri linguaggi utilizzati dagli esseri umani, è intrinsecamente ambigua, cioè il significato di una parola o frase dipende da più elementi come il contesto in cui è utilizzata. Per esempio, l'affermazione “sale in cucina” potrebbe voler raccontare di una persona che compie l'azione di salire e andare in cucina oppure voler indicare che il sale (cloruro di sodio) si trova in cucina. Nel primo caso “sale” è voce del verbo salire, nel secondo è un sostantivo e indica una particolare sostanza: il senso della frase cambia notevolmente.

Un linguaggio di programmazione è un linguaggio col quale un programmatore descrive le istruzioni che un computer deve eseguire.

NOTA

In un linguaggio di programmazione distinguiamo:

- un insieme di *simboli* coi quali scrivere parole e frasi (alcuni *caratteri speciali* assumono un significato particolare come lo spazio e la virgola che separano tra loro gli elementi delle istruzioni e il punto e virgola che chiude le istruzioni);
- un insieme di *parole chiave* che servono per descrivere le azioni da compiere;
- una *sintassi* che definisce le regole con le quali costruire le istruzioni.

I passaggi che portano dal problema al programma che lo risolve (Figura 1.2), sono i seguenti.

1. Si parte da un *problema*, cioè, genericamente, una situazione o un quesito che richiede una soluzione, partendo da elementi noti.
2. Si escogita un *algoritmo*, ovvero un metodo per risolvere il problema.
3. Si scrive il *programma* che realizza l'algoritmo.

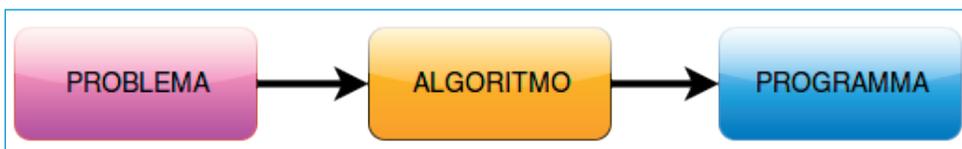


Figura 1.2 Il programma implementa un algoritmo per risolvere un problema.

La parola *algoritmo* deriva dal nome del matematico persiano medievale al-Khwarazmi e indica un procedimento definito da una sequenza finita di passi da eseguire per raggiungere un determinato obiettivo od ottenere un certo risultato.

NOTA

Nello schema problema–algoritmo–programma distinguiamo, oltre alle figure del *risolutore* e dell'*esecutore*, il *programmatore* e l'*utente finale*. In generale, queste figure possono essere diverse o sovrapposte, umane o artificiali (il risolutore, almeno per ora, è sempre un essere umano).

Caratteristiche di un algoritmo

- *Elementarità ed eseguibilità*: le istruzioni devono essere comprensibili dall'esecutore in modo che possano essere tradotte immediatamente in azioni.
- *Non ambiguità*: l'esecutore associa sempre a ogni istruzione un'azione precisa.
- *Correttezza e terminazione*: il problema proposto deve poter essere risolto in un tempo finito.
- *Generalità*: quasi sempre si vuole risolvere una classe di problemi e non solo un singolo problema. Per esempio, interessa trovare un procedimento che risolva il problema di quanto fa “ $a + b$ ” e non solo di quanto fa “ $3 + 8$ ”.

Introduzione a JavaScript

JavaScript è un *linguaggio di scripting* orientato agli oggetti e agli eventi, creato nel 1995 da Brendan Eich della Netscape Communications e di cui sono state realizzate varie versioni, dialetti e linguaggi derivati. Esiste una versione ufficiale standardizzata che si chiama ECMAScript (questo nome non ha riscosso molto successo e quindi non lo utilizzeremo).

JavaScript è un linguaggio di programmazione interpretato multiplatforma di alto livello ed è utilizzato per creare le cosiddette *pagine web dinamiche* che migliorano ed estendono le pagine web per quanto riguarda l'interazione con l'utente, la modificabilità dei contenuti e il modo in cui questi appaiono. Per esempio, quando si compila un *form* (o modulo di inserimento) per registrarsi a un servizio web, l'esecuzione di istruzioni JavaScript controlla la correttezza dei dati inseriti e segnala gli errori formali (per esempio il fatto che non sono stati specificati tutti i dati obbligatori oppure che un codice fiscale deve essere composto da 16 caratteri). Viene eseguito del codice JavaScript ogni qual volta c'è un movimento o qualche elemento della pagina (menu a comparsa, mappa interattiva ecc.) compare o viene modificato.

NOTA

JavaScript ha bisogno di un interprete che legge le istruzioni e le esegue traducendole immediatamente in istruzioni in linguaggio macchina (la lingua del microprocessore, o CPU, *Central Processing Unit*, il cervello del PC) e compiendo le azioni associate a queste ultime. Tutti i browser moderni sono dotati di un interprete JavaScript, cioè un sottoprogramma che viene richiamato ogniqualvolta dev'essere eseguito codice JavaScript.

Quanto è usato JavaScript e per che cosa

JavaScript è un linguaggio di programmazione completo e può essere utilizzato anche per realizzare applicazioni al di fuori del browser o anche per dispositivi mobili. Noi non ci occuperemo di questa tipologia di applicazioni, ma ci concentreremo sull'aspetto principale del suo successo, cioè il fatto che, essendo l'unico linguaggio di programmazione eseguito in tutti i browser moderni, sia alla base di quasi tutte le applicazioni e i siti web attuali. JavaScript è, infatti, utilizzato da tutti coloro che offrono servizi via Web come Google, Amazon, Facebook, Microsoft, Twitter, Apple ecc.

Per avere un'idea della diffusione di JavaScript possiamo consultare siti come quello di TIOBE Software (<https://www.tiobe.com/tiobe-index>), che stila una classifica mensile dei linguaggi di programmazione più “popolari” e che vede ormai da 15 anni JavaScript attorno all'ottavo posto, oppure PYPL (*Popularity of Programming Language*, <https://pypl.github.io/PYPL.html>) che nella propria classifica di gradimento colloca JavaScript al quarto posto e in crescita.

Certo la popolarità di un linguaggio non ne garantisce automaticamente la bontà, ma assicura che JavaScript verrà utilizzato ancora per molto tempo, visto che lo utilizzano milioni di programmatori e che il Web continua la sua crescita vertiginosa.

GitHub è un *repository online* molto noto, con oltre 2 milioni di progetti attivi (dati aggiornati ad agosto 2017). GitHub calcola una statistica storica dei progetti pubblicati su GitHub (<https://madnight.github.io/github/>) e vede JavaScript saldamente al primo posto, con un trend di crescita pressoché costante (Figura 1.3).

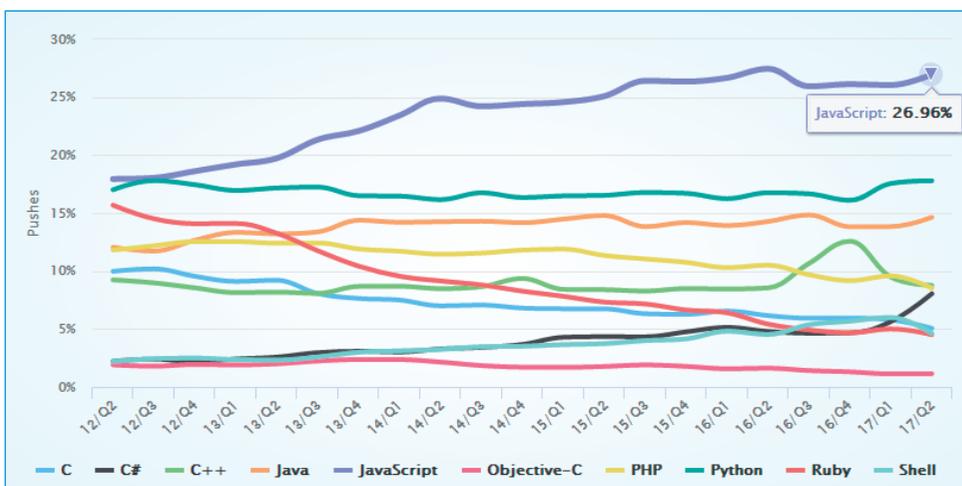


Figura 1.3 JavaScript è largamente il linguaggio più attivo sulla piattaforma GitHub.

Un altro riferimento è quello fornito dall'IEEE (*Institute of Electrical and Electronics Engineers*, la più grande organizzazione mondiale di professionisti che si occupano di nuove tecnologie), la cui classifica per il 2017 sulla “popolarità” dei linguaggi di programmazione, riportata nella Figura 1.4, evidenzia ancora una volta l'importanza di JavaScript per il mondo web.

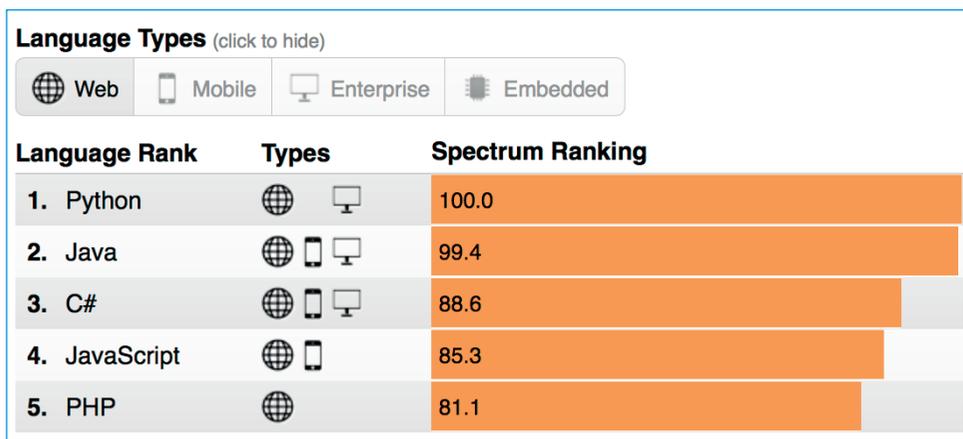


Figura 1.4 La classifica 2017 dei linguaggi di programmazione dell'IEEE (<http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017>).

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 primaria. 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 per capire meglio come funziona un mondo sempre più digitale.

JavaScript è (insieme ad altre tecnologie e linguaggi come Scratch e Python) un elemento importante di questa rivoluzione pedagogica e culturale ed è presente in molti programmi, governativi e non, per favorire il passaggio dall'alfabetizzazione digitale, orientata all'uso dei dispositivi, all'alfabetizzazione informatica vera e propria.

Un esempio di iniziative di questo tipo è *The Hour of Code* (hourofcode.com) del Ministero dell'Istruzione americano, che è stato diffuso in più di 180 nazioni. In Italia prende il nome di *Ora del Codice* (italia.code.org) ed è parte del progetto *Programma il Futuro* del nostro Ministero dell'Istruzione (programmailfuturo.it).

Strumenti di lavoro

Programmare in JavaScript richiede principalmente tre strumenti (in realtà i primi due sono conglobati nello stesso programma):

- un browser web (è consigliato Google Chrome);
- gli strumenti per sviluppatori (inclusi nel browser);
- un editor di testo.

Per quanto riguarda il primo punto, l'immane browser web, Google Chrome è indubbiamente una delle soluzioni più affermate: stabile ed efficiente, è un ottimo programma e si integra perfettamente con tutte le applicazioni web sviluppate da Google (per esempio Gmail). Anche Mozilla Firefox è una scelta valida, così come Apple Safari. Eventualmente si può valutare anche l'uso di Microsoft Edge.

In questo libro utilizzeremo solo l'ultima versione di Chrome e i suoi strumenti di sviluppo, ma per chi avesse scelto un altro browser non sarà difficile adattare quanto esposto.

Google Chrome

Per prima cosa verifica che la versione installata sia la più recente oppure procedi all'aggiornamento o all'installazione di Chrome. Qui potresti trovarti di fronte a un "dilemma ricorsivo": come fare a scaricare da Internet un browser se per accedere a Internet serve proprio un browser?

Per fortuna oggi tutti i sistemi operativi vengono forniti di un browser preinstallato col quale puoi installare Chrome seguendo le istruzioni riportate all'indirizzo <https://www.google.com/chrome/browser/desktop>.

Strumenti per sviluppatori (Web Developer Tools)

Gli strumenti per sviluppatori di Chrome sono funzionalità interne al browser rivolte agli sviluppatori di siti e applicazioni web.

Questi strumenti sono accessibili mediante il tasto F12 oppure con la combinazione di tasti Ctrl-Shift-I (Cmd-Shift-I su macOS) o, ancora, facendo clic sul pulsante *Opzioni* (che è rappresentato dai 3 puntini verticali in alto a destra nella Figura 1.5) e selezionando la voce *Altri strumenti* e poi *Strumenti per sviluppatori*.

Per avere una pagina vuota come nella figura scrivi **about:blank** nella barra degli indirizzi.

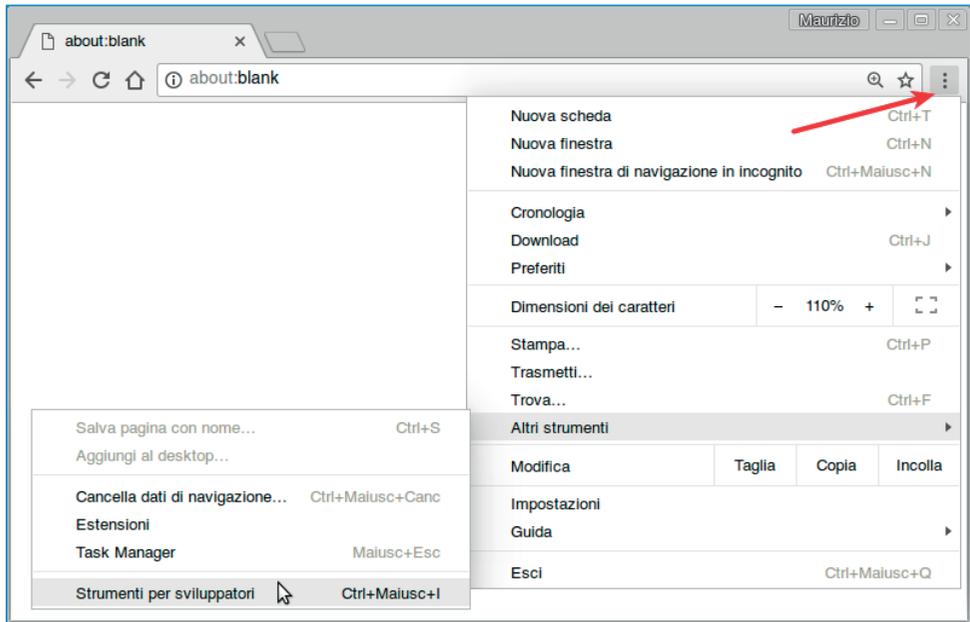


Figura 1.5 Selezione della voce Strumenti per sviluppatori in Google Chrome, raggiungibile dal pulsante Opzioni evidenziato in alto a destra.

Nella schermata che si apre (Figura 1.6) appaiono più schede (disponibili solo in inglese e che variano anche in base alle estensioni installate nel browser) che mettono a disposizione numerose funzionalità. Più avanti avremo modo di conoscerle meglio e di imparare a utilizzarle.

A questo punto seleziona la scheda *Console* (se non è già aperta). Appare un segno di maggiore e un *cursore* rappresentato da una barra verticale lampeggiante. Si tratta del *prompt dei comandi* (evidenziato dalla freccia nella Figura 1.6) che indica che l'interprete JavaScript è in attesa di ricevere ed eseguire istruzioni.

NOTA

Per operare con un computer abbiamo bisogno di un sistema che permetta di impartire comandi e fornisca il loro risultato. Si tratta di un'interfaccia tra l'essere umano e il PC che principalmente può essere di due tipi: CLI (*Command Line Interface*, o interfaccia a carattere o a linea di comando), che utilizza una *console* di solo testo sia per l'input che per l'output, oppure GUI (*Graphics User Interface*), che utilizza elementi grafici come finestre, icone, menu, pulsanti ecc. e un dispositivo hardware di puntamento quale per esempio il mouse.

Scrivi **ciao** e premi il tasto Invio. L'interprete non trova la parola "ciao" tra quelle conosciute e segnala una situazione di parola non definita e, quindi, l'impossibilità di riconoscere un'istruzione valida.

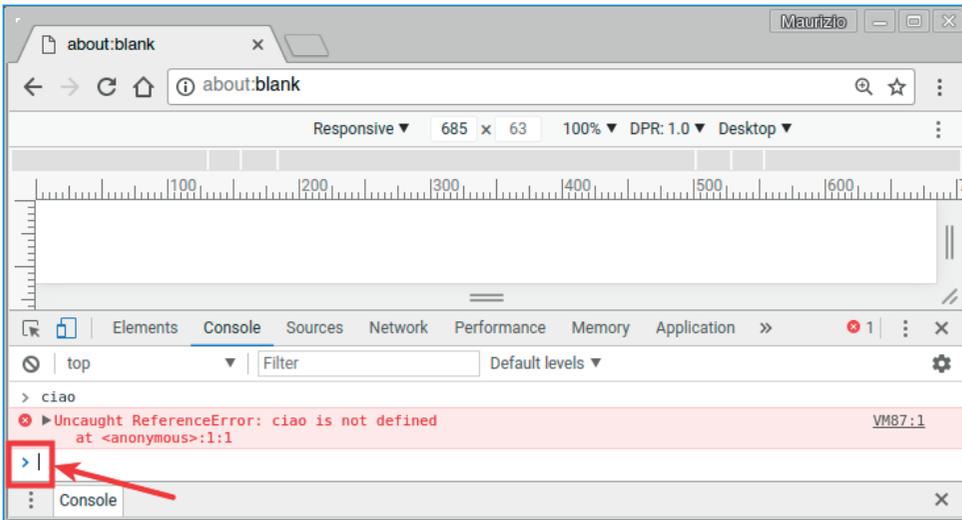


Figura 1.6 L'interprete segnala che "ciao" è un comando sconosciuto.

Editor di testi

Con gli *Strumenti per sviluppatori* di Chrome dialoghiamo con l'interprete inserendo un'istruzione per volta. Per scrivere le istruzioni di un programma vero e proprio abbiamo bisogno di un programma di scrittura di testo che permetta di salvare il *codice sorgente* JavaScript su file per poterlo modificare ed eseguire più volte. Qualsiasi editor di testo può andare bene, dal più semplice Notepad di Windows ai più raffinati. Ovviamente non bisogna confondere un *editor di testo* con un *word processor* (o programma di videoscrittura) come Word: il formato con cui viene salvato un documento di un word processor non va bene per scrivere codici sorgente costituiti da semplici caratteri di testo ma serve per comporre documenti complessi con caratteristiche grafiche e orientati alla stampa.

Esistono molti editor *open source* e gratuiti, ricchi di *plugin* (o estensioni) che facilitano la scrittura di sorgenti nei diversi linguaggi. Prova a googlare col testo: "developer editor": tra i risultati ci saranno sicuramente anche classifiche dei migliori editor del momento. Controlla la licenza con cui sono forniti e se sono gratuiti scaricane qualcuno e provalo. Noi consigliamo Atom (visibile nella Figura 1.7 e disponibile all'indirizzo <https://atom.io>); è *open source*, ha molti plugin, ha la colorazione del codice per tutti i principali linguaggi e anche l'autocompletamento, una sorta di suggerimento che completa le parole mentre le si sta scrivendo e che si rivela molto prezioso sia per programmatori in erba che per esperti. Citiamo, tra gli altri editor gratuiti molto validi, Notepad++ (<https://notepad-plus-plus.org/>) per la sua semplicità e Brackets (<http://brackets.io/>).

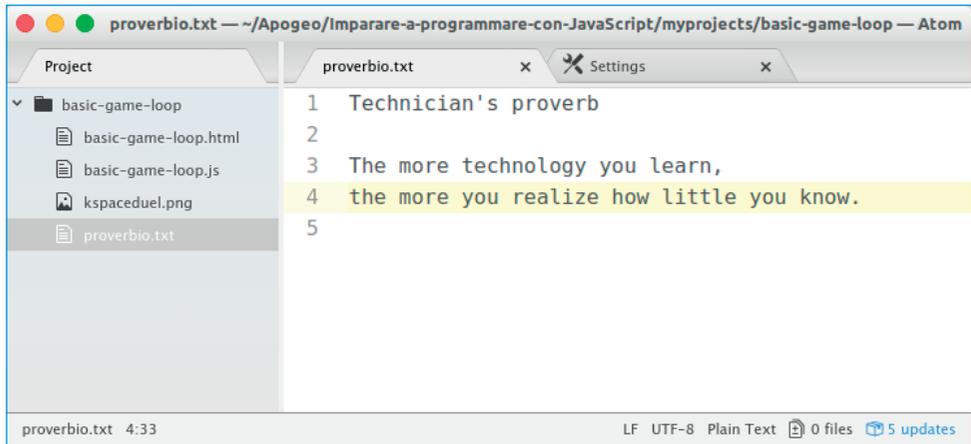


Figura 1.7 Il programma di editing per programmatori Atom (<https://atom.io>).

Più complessi, ricchi di funzionalità e sempre gratuiti sono gli IDE (*Integrated Development Environment*, ambienti integrati per lo sviluppo) Eclipse, NetBeans, Komodo e Visual Studio Code. Se sei un utente esperto di computer puoi avventurarti nell’installazione e nell’uso di questi programmi, altrimenti rimani su un semplice e più maneggevole editor di testo.

Che cosa hai imparato

- Che cosa vuol dire programmare un computer e che cosa sono gli algoritmi, i linguaggi di programmazione e i programmi.
- Che cos’è JavaScript, com’è nato, a che cosa serve e quali sono alcune sue caratteristiche.
- Alcuni ambiti di utilizzo del linguaggio e alcuni dati statistici sul suo impiego.
- Quali sono gli strumenti di lavoro disponibili col browser Google Chrome.
- Come aprire la console negli *Strumenti per sviluppatori* e impartire un comando JavaScript errato.
- Che cosa sono e a che cosa servono gli editor di testo come Atom.