

A teal diagonal stripe runs from the top-left towards the center of the dark purple rectangle.

1

ABC DELLA PROGRAMMAZIONE

Prima di imparare a leggere e scrivere codice JavaScript, è necessario prendere contatto con alcuni concetti chiave della programmazione, che saranno trattati nelle seguenti sezioni.

A

Che cos'è uno script e come scriverne uno?

B

Come si collocano i computer nel mondo che li circonda?

C

Come si scrive uno script per una pagina web?

Dopo aver introdotto le basi, i prossimi capitoli mostreranno come usare il linguaggio JavaScript per controllare il funzionamento dei browser.



1/a

CHE COS'È
UNO SCRIPT?
E COME SI SCRIVE?

UNO SCRIPT È UNA SEQUENZA DI ISTRUZIONI

Uno script è costituito da una serie di istruzioni che il computer può seguire per conseguire un determinato obiettivo.

In pratica gli script sono paragonabili ai seguenti oggetti.

RICETTE

Seguendo le istruzioni di una ricetta, una dopo l'altra e nell'ordine corretto, i cuochi cucinano un piatto che non hanno mai preparato prima. Alcuni script sono brevi e riguardano un'unica situazione; sono come ricette per preparare piatti semplici. Altri possono svolgere più operazioni, come una ricetta può essere per un piatto complesso, di tre portate. Un'altra analogia: per chi è alle prime armi nell'arte della cucina o della programmazione, è necessario del tempo da dedicare all'acquisizione della terminologia.

GUIDE

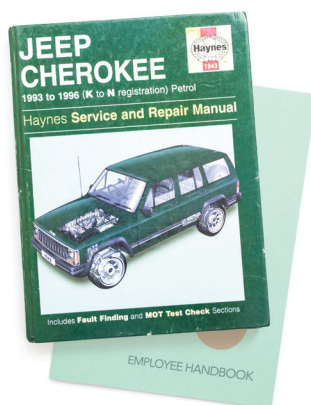
Le grandi aziende spesso forniscono ai nuovi assunti una guida, contenente le procedure da seguire nelle varie situazioni. Per esempio, la guida per un hotel può contenere i passi da seguire nelle varie situazioni, per esempio al momento del check in, nella corretta pulizia di una stanza, quando scatta un allarme e così via.

In ognuna di queste situazioni, i dipendenti devono seguire solo i passi specifici di questo tipo di evento (non devono seguire ogni singolo passo dell'intera guida quando devono solo eseguire un check in). Analogamente, in uno script complesso, il browser deve utilizzare solo una parte del codice disponibile.

MANUALI

I meccanici di auto spesso fanno riferimento ai manuali per la manutenzione. Questi manuali contengono una serie di test per verificare il funzionamento degli organi principali dell'auto, più i dettagli sul modo in cui intervenire per risolvere eventuali problemi.

Per esempio, vi possono essere alcune indicazioni sul controllo dei freni. Se i test danno esito positivo, il meccanico può procedere con i test successivi senza riparare i freni. Se invece i test non hanno successo, il meccanico dovrà seguire le istruzioni fornite per ripararli. Il meccanico può quindi tornare indietro e verificare nuovamente il funzionamento dei freni per controllare che il problema sia risolto. Se ora i freni passano il test, il meccanico sa di aver terminato e di poter procedere con il test successivo. Analogamente, gli script consentono al browser di controllare la situazione attuale e di svolgere solo quei passi necessari per svolgere una determinata azione.

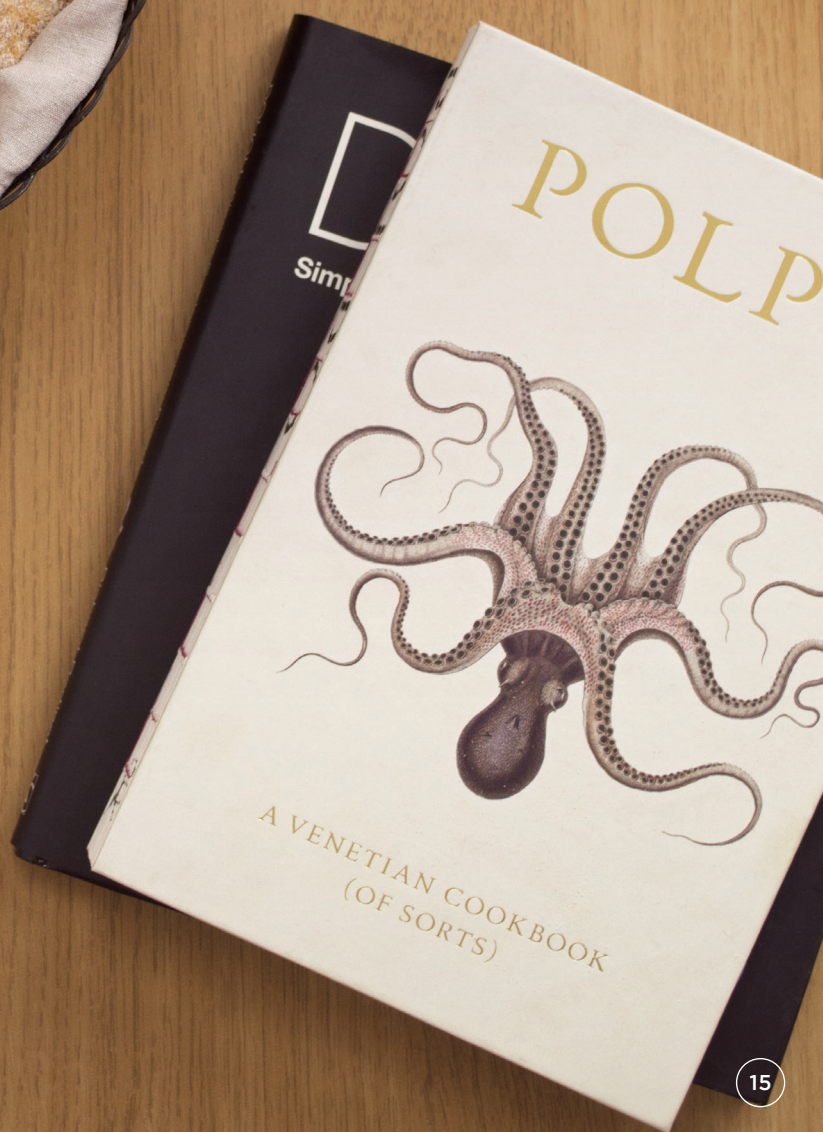




Gli script sono costituiti da istruzioni che il computer deve eseguire, una dopo l'altra.

Un browser può utilizzare parti differenti dello script a seconda del modo in cui l'utente interagisce con la pagina web.

Gli script possono eseguire sezioni differenti del codice in risposta alla situazione in cui si trovano a operare.



REALIZZARE UNO SCRIPT

Per realizzare uno script, occorre stabilire i propri obiettivi e poi elencare i compiti da completare, in modo da trovare una soluzione.

Gli esseri umani possono risolvere problemi complessi senza pensarci troppo su, per esempio siamo in grado di guidare, preparare la colazione o inviare un messaggio di posta elettronica anche senza istruzioni dettagliate. Ma la prima volta che si svolgono queste operazioni, ogni compito può sembrare difficile. Pertanto, quando acquisiamo una nuova capacità, spesso tendiamo a suddividere le cose in compiti più semplici, facili da apprendere uno alla volta. Con l'esperienza, questi singoli compiti divengono sempre più familiari e ci sembrano molto più semplici.

Alcuni degli script che leggerete o scriverete entro il termine di questo libro, saranno piuttosto complessi; almeno inizialmente potrebbero intimorire un po'. Tuttavia, uno script non è altro che una sequenza di brevi istruzioni, ognuna delle quali viene svolta con lo scopo di risolvere il problema generale. Questo è il motivo per cui creare uno script è come scrivere una ricetta o un manuale che consenta a un computer di risolvere un problema, un passo alla volta.

Vale però la pena di notare che un computer non impara, con l'esperienza, a svolgere i propri compiti, come è normale per noi; deve seguire le stesse specifiche istruzioni ogni volta che svolge il proprio compito. Pertanto il programma deve fornirgli dettagli sufficienti per eseguire il compito come se fosse, ogni volta, la sua prima volta.





Partite dall'idea del risultato generale che volete ottenere e suddividetelo in passi più semplici.

1: DEFINIRE L'OBIETTIVO

Innanzitutto dovete definire il compito che volete svolgere. È come se il computer dovesse risolvere un problema per voi.

2: PROGETTARE LO SCRIPT

Per progettare uno script dovete suddividere l'obiettivo in una serie di compiti da completare per poter risolvere il problema.

A questo punto potete scrivere i singoli passi che il computer deve svolgere per completare ogni singolo compito (e anche ogni informazione di cui può aver bisogno per svolgere tale compito), un po' come se scriveste una ricetta che il computer deve seguire.

3: PROGRAMMARE CIASCUN PASSO

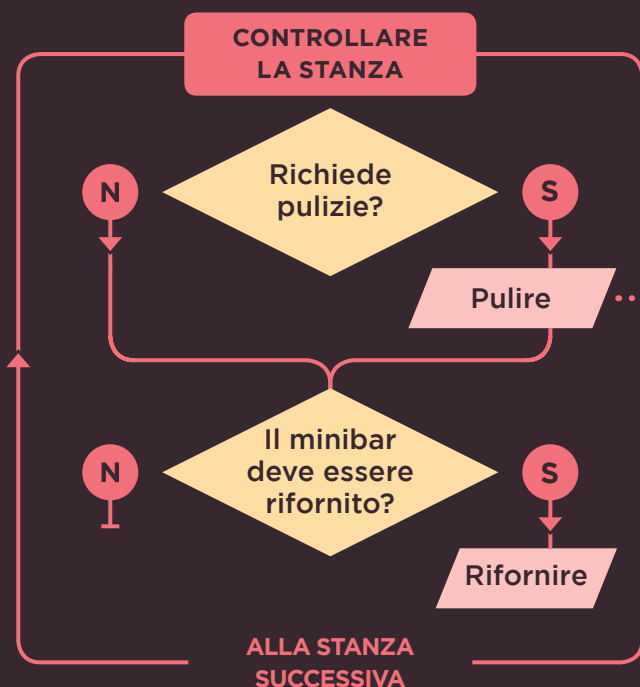
Ognuno dei passi deve essere scritto in un linguaggio di programmazione comprensibile al computer. Nel nostro caso si tratta di JavaScript.

Per quanto abbiate la tentazione di iniziare a programmare di getto, vale la pena la pena di dedicare del tempo a progettare lo script, prima di iniziare a programmare.

PROGETTARE UNO SCRIPT: I COMPITI

Una volta definito l'obiettivo dello script, potete considerare i singoli compiti necessari per conseguirlo. Questa visione “ad alto livello” dei compiti può essere rappresentata tramite un diagramma di flusso.

DIAGRAMMA DI FLUSSO: COMPITI PER LA PULIZIA DI UNA CAMERA D'ALBERGO



PROGETTARE UNO SCRIPT: I PASSI

Ogni singolo compito può essere suddiviso in una sequenza di passi. Quando sarete pronti per programmare lo script, questi passi possono essere tradotti in singole righe di codice.

LISTA: PASSI DA COMPIERE PER PULIRE UNA STANZA

- PASSO 1** Togliere le lenzuola
- PASSO 2** Pulire tutte le superfici
- PASSO 3** Passare l'aspirapolvere
- PASSO 4** Mettere le nuove lenzuola
- PASSO 5** Togliere asciugamani e saponi usati
- PASSO 6** Pulire toilette, vasca, lavandino
- PASSO 7** Mettere nuovi asciugamani e saponi
- PASSO 8** Pulire il pavimento del bagno

Come vedrete nella prossima pagina, i passi che devono essere seguiti da un computer per svolgere un determinato compito, spesso sono molto differenti da quelli che potrebbe svolgere un essere umano.

DAI PASSI AL CODICE

Ogni passo di ogni compito rappresentato in un diagramma di flusso deve essere scritto in un linguaggio comprensibile dal computer.

In questo libro ci occupiamo del linguaggio di programmazione JavaScript e del suo impiego nei browser web.

Come accade quando si impara una nuova lingua, occorre occuparsi di due elementi.

- Il **vocabolario**: le parole che il computer è in grado di comprendere.
- La **sintassi**: come unire queste parole per creare istruzioni che il computer sia in grado di seguire.

Oltre a imparare a conoscere il linguaggio, chi è alle prime armi nel campo della programmazione dovrà anche imparare a fare in modo che il computer svolga determinati tipi di obiettivi adottando un approccio **programmatico** alla soluzione dei problemi.

I computer sono sistemi logici e obbedienti. Devono conoscere ogni singolo dettaglio di ciò che ci si aspetta da loro e poi procederanno senza mai mettere in discussione nulla. Per il fatto che richiedono tipi di istruzione differenti rispetto alle persone, tutti coloro che imparano a programmare commettono molti errori, specialmente all'inizio. Ma non scoraggiatevi; nel Capitolo 10 incontreremo vari modi per scoprire che cosa è andato storto. I programmatori chiamano questa operazione **debugging**.





Occorre imparare a "pensare" come un computer, poiché le macchine risolvono le operazioni in modo differente rispetto agli esseri umani.

I computer risolvono i problemi in modo **programmatico**; ovvero seguendo una serie di istruzioni, una dopo l'altra. I tipi di istruzioni di cui hanno bisogno, spesso sono molto differenti da quelle che potreste impartire a un essere umano. Pertanto, nel corso del libro non apprenderemo solo il vocabolario e la sintassi utilizzata da JavaScript, ma vedremo anche come scrivere istruzioni che il computer sia in grado di seguire. Per esempio, osservando l'immagine a sinistra come valutare quale figurina è più alta? Un computer richiede istruzioni esplicite, da eseguire passo dopo passo.

1. Trovare l'altezza della prima figurina.
2. Stabilire che questa è la "figurina più alta".
3. Trovare l'altezza delle figurine rimanenti, una dopo l'altra, e confrontarla con l'altezza della "figurina più alta" che è stata trovata fino a quel momento.
4. In ciascun passo, se si trova una figurina la cui altezza è maggiore della "figurina più alta" attuale, la nuova figurina diverrà la nuova "figurina più alta".
5. Dopo aver controllato tutte le figurine, fornire il risultato: qual è la figurina più alta?

Pertanto, il computer deve controllare ogni singola figurina, a turno, e per ognuna di esse svolgere un test "È più alta rispetto alla figurina più alta attuale?". Dopo aver svolto questa operazione per ogni singola figurina, sarà in grado di fornire la propria risposta.

DEFINIRE UN OBIETTIVO E PROGETTARE LO SCRIPT

Considerate il modo in cui potete affrontare un altro tipo di script.
Questo esempio calcola il costo di una targhetta.
Ai clienti verrà addebitato un costo per lettera.

La prima cosa che dovete fare è definire gli obiettivi dello script, ovvero che cosa volete ottenere:
"I clienti possono chiedere di creare una targhetta con il proprio nome. Ogni lettera ha un costo di 5\$. Quando l'utente inserisce un nome, mostrare il costo totale".

Poi si deve suddividere il tutto in una serie di compiti che devono essere svolti per conseguire l'obiettivo.

- 1. Lo script si attiva quando si fa clic sul pulsante.
- 2. Accetta il nome specificato nel campo di testo.
- 3. Controlla che l'utente abbia inserito un valore.
- 4. Se l'utente non ha inserito nulla, presenta un messaggio che chiede di specificare il nome.
- 5. Se l'utente ha inserito un nome, calcola il costo della targhetta moltiplicando il numero di lettere per il costo per lettera.
- 6. Al termine presenta il costo della placchetta.

I numeri di passi corrispondono al diagramma di flusso presentato nella pagina a fronte.

CUSTOM SIGNAGE

Enter name:

SHOW COST

CUSTOM SIGNAGE

Enter name: Please enter a name below...

THOMAS

SHOW COST

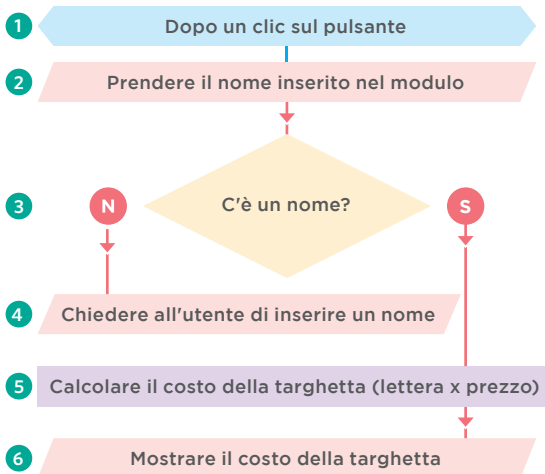
CUSTOM SIGNAGE

\$30

T H O M A S

TRACCIARE I COMPITI IN UN DIAGRAMMA

Spesso gli script devono svolgere compiti differenti in situazioni differenti. Potete utilizzare dei diagrammi di flusso per definire tali compiti. I diagrammi di flusso mostrano i percorsi fra ogni singolo passo.

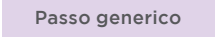


Le frecce indicano il modo in cui lo script procede da un compito al successivo. Forme differenti corrispondono a tipi di compiti differenti. In alcuni casi si tratta di decisioni, le quali fanno sì che il codice segua percorsi differenti.

Nel Capitolo 2 vedremo come trasformare questo esempio in codice. Vedremo molti esempi di diagrammi di flusso nel corso di questo libro e presenteremo il codice che aiuta a gestire questi tipi di situazioni.

I programmatori esperti possono usare diagrammi più complessi, progettati in modo specifico per rappresentare il codice. Tuttavia tali tipi di diagrammi presentano una curva di apprendimento più ripida. Questi diagrammi informali vi aiuteranno invece a rappresentare il funzionamento degli script nella fase di apprendimento del linguaggio.

LEGGENDA DEI DIAGRAMMI DI FLUSSO

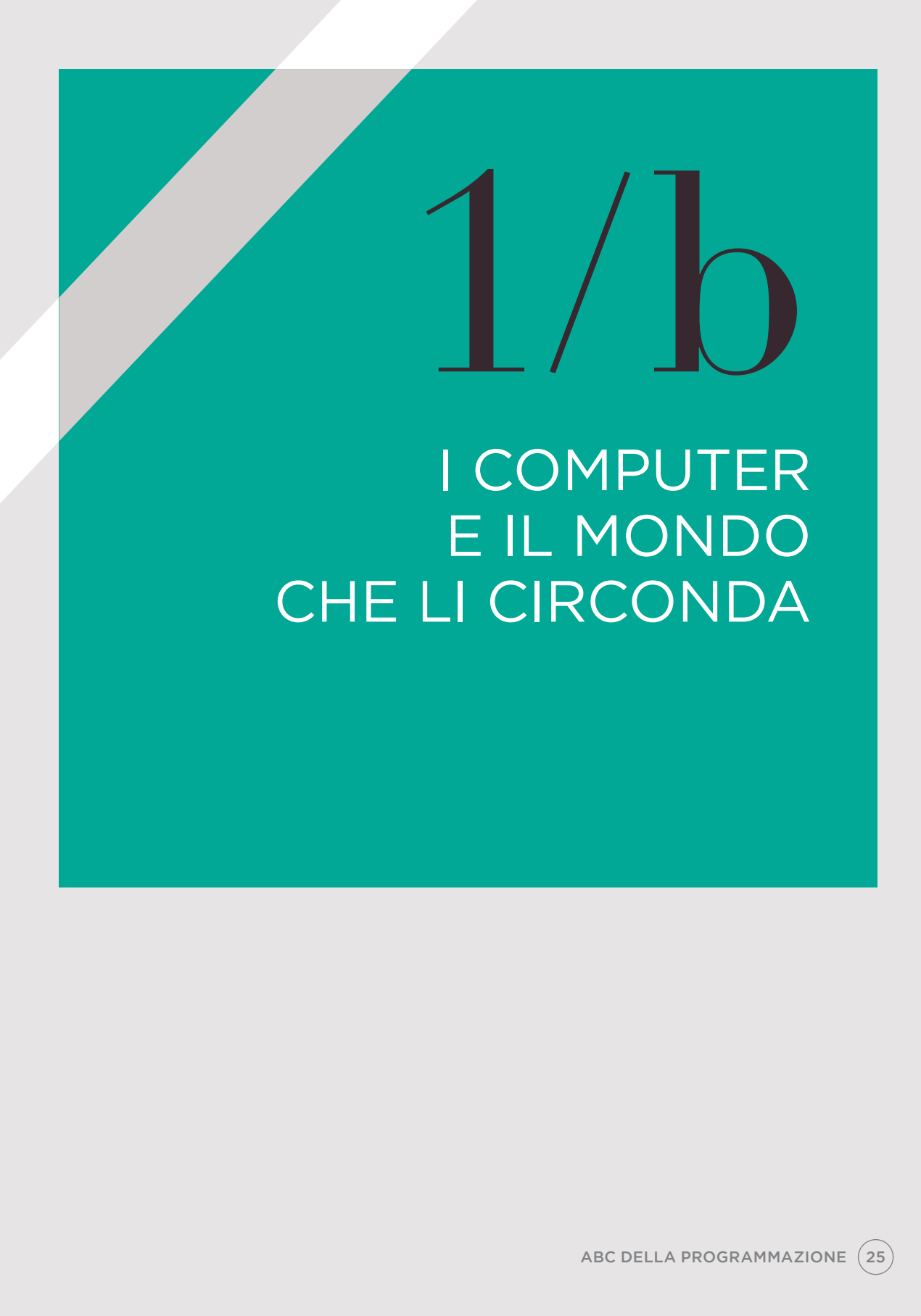
 Passo generico	 Evento
 Input o output	 Decisione

RIEPILOGO

ABC DELLA PROGRAMMAZIONE

A Che cos'è uno script? E come si scrive?

- ▶ Uno script è costituito da una serie di istruzioni che il computer può eseguire per raggiungere un obiettivo.
- ▶ Ogni volta che si esegue lo script, questo potrebbe utilizzare solo un sottoinsieme di tutte le istruzioni.
- ▶ I computer affrontano i compiti in modo diverso dagli esseri umani e pertanto le istruzioni devono consentire al computer di risolvere il compito in modo programmatico.
- ▶ Per imparare a creare uno script, suddividete l'obiettivo in una serie di compiti e poi svolgete tutti i passi necessari per completare tale compito, con l'aiuto di un diagramma di flusso.

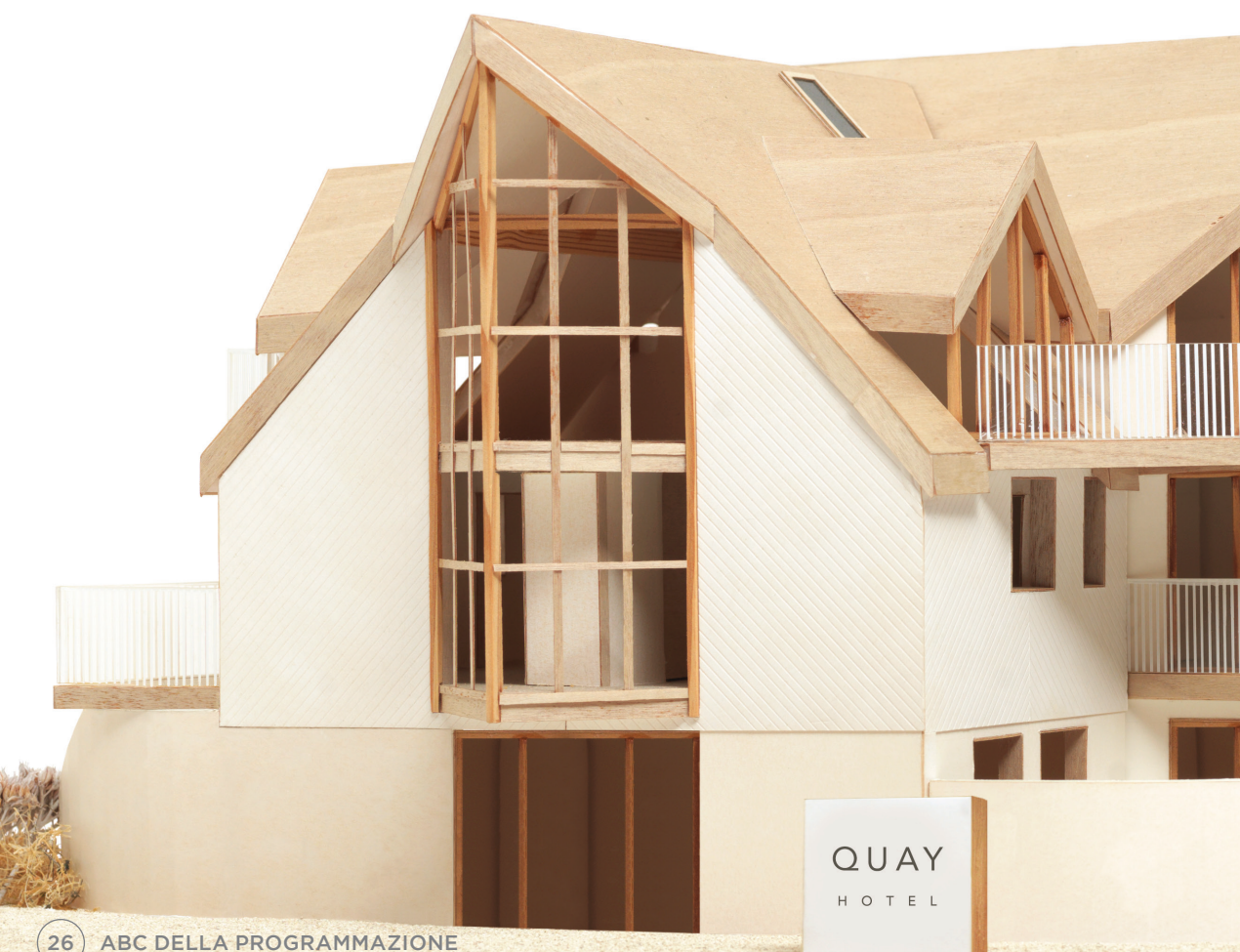


1/b

I COMPUTER
E IL MONDO
CHE LI CIRCONDA

I COMPUTER CREANO MODELLI DEL MONDO UTILIZZANDO I DATI

Ecco il modello di un hotel, con tanto di alberi, persone e auto, anch'essi modelli. Per un essere umano, è chiaro quale tipo di oggetto reale è rappresentato da ciascuno di questi modelli.



Un computer non ha un concetto predeterminato di hotel o di auto. Non sa a che cosa servono. Il vostro portatile o telefono non ha alcuna marca preferita di auto, ne sa che cosa significhi valutare un hotel.

Pertanto, come possiamo utilizzare i computer per creare la prenotazione di una stanza d'albergo o dei videogame in cui i giocatori possono guidare un'auto? La risposta è che i programmatori creano un tipo di modello molto differente, rivolto in modo specifico ai computer.

I programmatori creano questi modelli utilizzando dati. La cosa non è strana o complicata come può sembrare, perché i dati sono tutto ciò di cui il computer ha bisogno per seguire le istruzioni che gli fornirete per svolgere le sue attività.



OGGETTI E PROPRIETÀ

Se non poteste vedere l'immagine dell'hotel e delle auto, i dati raccolti nei riquadri informativi potranno raccontarvi molte cose sulla scena rappresentata.

OGGETTI (ELEMENTI FISICI)

In programmazione, ogni elemento fisico presente nel mondo può essere rappresentato con un **oggetto**. Qui vi sono due diversi **tipi** di oggetti: l'hotel e l'auto.

I programmatori direbbero che vi è **una istanza** dell'oggetto hotel e **due istanze** dell'oggetto auto.

Ogni oggetto può essere dotato di:

- proprietà;
- eventi;
- metodi.

Insieme, queste tre parti formano un modello funzionante di tale oggetto.

PROPRIETÀ (CARATTERISTICHE)

Tutte le auto hanno delle caratteristiche in comune. Entrambe le auto possono avere una marca, un colore e una cilindrata. È anche possibile determinarne la velocità corrente. I programmatori chiamano queste caratteristiche le **proprietà** di un oggetto.

Ogni proprietà ha un **nome** e un **valore** e ognuna di queste coppie nome/valore dice qualcosa su quella specifica istanza dell'oggetto.

La proprietà più ovvia dell'hotel è il nome (name). Il valore di tale proprietà è Quay. Potete conoscere il numero delle stanze dell'hotel osservando il valore che si trova a lato della sua proprietà rooms.

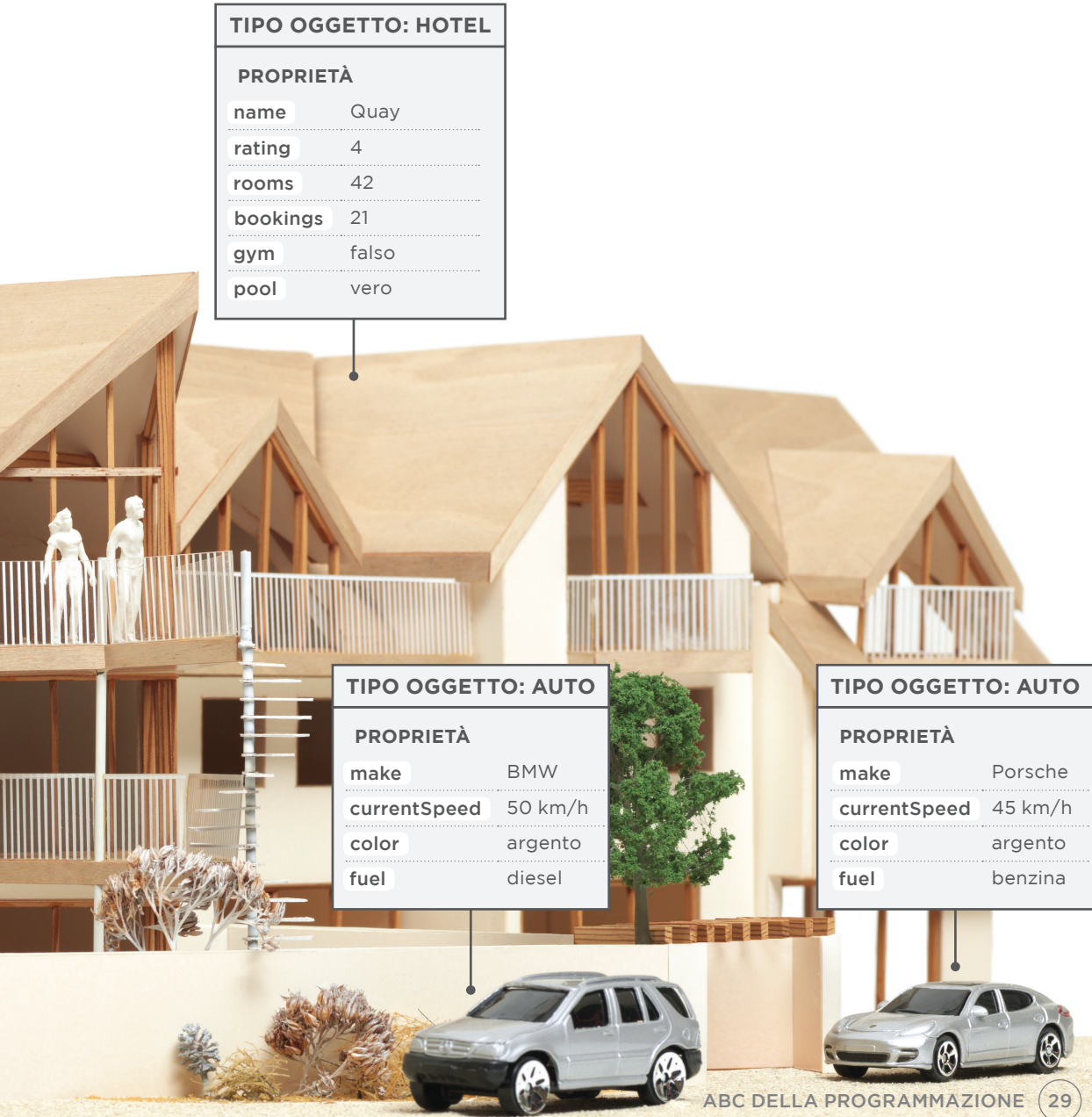
L'idea delle coppie nome/valore viene utilizzata anche in HTML e CSS. In HTML, un attributo è come una proprietà; attributi differenti hanno nomi differenti e ciascun attributo può avere un valore. Analogamente, in CSS potete cambiare il colore di un titolo creando una regola che fornisce alla proprietà color un valore specifico, oppure potete cambiare il tipo di carattere con cui è scritto, assegnando alla proprietà font-family un determinato valore. Le coppie nome/valore vengono utilizzate molto nella programmazione.

OGGETTO HOTEL

L'oggetto hotel1 utilizza nomi di proprietà e valori che informano su questo specifico hotel, come il nome, la sua valutazione, il numero di stanze di cui è dotato e quante stanze sono occupate. Potete anche scoprire se questo hotel offre o meno determinati servizi.

OGGETTI AUTO

Gli oggetti auto condividono le stesse proprietà, ma ognuna di esse ha valori differenti per queste proprietà. Le proprietà dicono la marca dell'auto, la velocità, il colore e il tipo di carburante richiesto.



TIPO OGGETTO: HOTEL	
PROPRIETÀ	
name	Quay
rating	4
rooms	42
bookings	21
gym	falso
pool	vero

TIPO OGGETTO: AUTO	
PROPRIETÀ	
make	BMW
currentSpeed	50 km/h
color	argento
fuel	diesel

TIPO OGGETTO: AUTO	
PROPRIETÀ	
make	Porsche
currentSpeed	45 km/h
color	argento
fuel	benzina

EVENTI

Nel mondo reale, le persone interagiscono con gli oggetti. Queste interazioni possono cambiare il valore delle proprietà di questi oggetti.

CHE COS'È UN EVENTO?

Vi sono vari modi con cui le persone possono interagire con un tipo di oggetto. Per esempio, in un'auto il conducente ha a disposizione almeno due pedali. L'auto è progettata per rispondere in modo differente quando il conducente agisce sui pedali:

- l'acceleratore aumenta la velocità dell'auto;
- il freno riduce la velocità dell'auto.

Analogamente, i programmi sono progettati per svolgere operazioni differenti quando l'utente interagisce con il computer in modi differenti. Per esempio, facendo clic sul link di un contatto in una pagina web si può richiamare il modulo del contatto e inserendo del testo nella casella di ricerca si può attivare automaticamente la funzionalità di ricerca.

Un **evento** è il modo usato dal computer per dire: "Ehi, è appena accaduto qualcosa!"

CHE COSA FA UN EVENTO?

I programmatori hanno la facoltà di scegliere a quali eventi rispondere. Quando si verifica un determinato evento, questo può essere utilizzato per attivare una determinata sezione del codice.

Gli script, spesso, usano eventi differenti per attivare funzionalità differenti.

Pertanto uno script stabilirà a quali eventi vuole rispondere il programmatore e quale parte dello script deve essere eseguito quando si verifica uno di questi eventi.

L'OGGETTO HOTEL

Un hotel riceverà regolarmente prenotazioni per le sue stanze. Ogni volta che viene riservata una stanza, si può utilizzare un evento book per attivare quella parte del codice che incrementa il valore della proprietà bookings. Analogamente, un evento cancellazione attiva quella parte del codice che riduce il valore della proprietà bookings.

GLI OGGETTI AUTO

Il conducente può accelerare e frenare, mentre guida l'auto. L'evento accelerate lancia il codice che incrementa il valore della proprietà currentSpeed; l'evento brake lancia il codice che riduce il valore della proprietà currentSpeed. Nella prossima pagina parleremo del codice che risponde agli eventi e cambia il valore delle proprietà.



METODI

I metodi rappresentano le attività che le persone devono svolgere utilizzando gli oggetti. I metodi possono leggere o modificare i valori delle proprietà di un oggetto.

CHE COS'È UN METODO?

I metodi, in genere, rappresentano il modo in cui le persone (o altri oggetti) interagiscono con un oggetto.

Sono un po' come domande o istruzioni che:

- dicono qualcosa sull'oggetto (leggendo le informazioni conservate nelle sue proprietà);
- modificano il valore di una o più proprietà dell'oggetto.

CHE COSA FA UN METODO?

Il codice di un metodo può contenere molte istruzioni che, tutte insieme, descrivono il compito da svolgere.

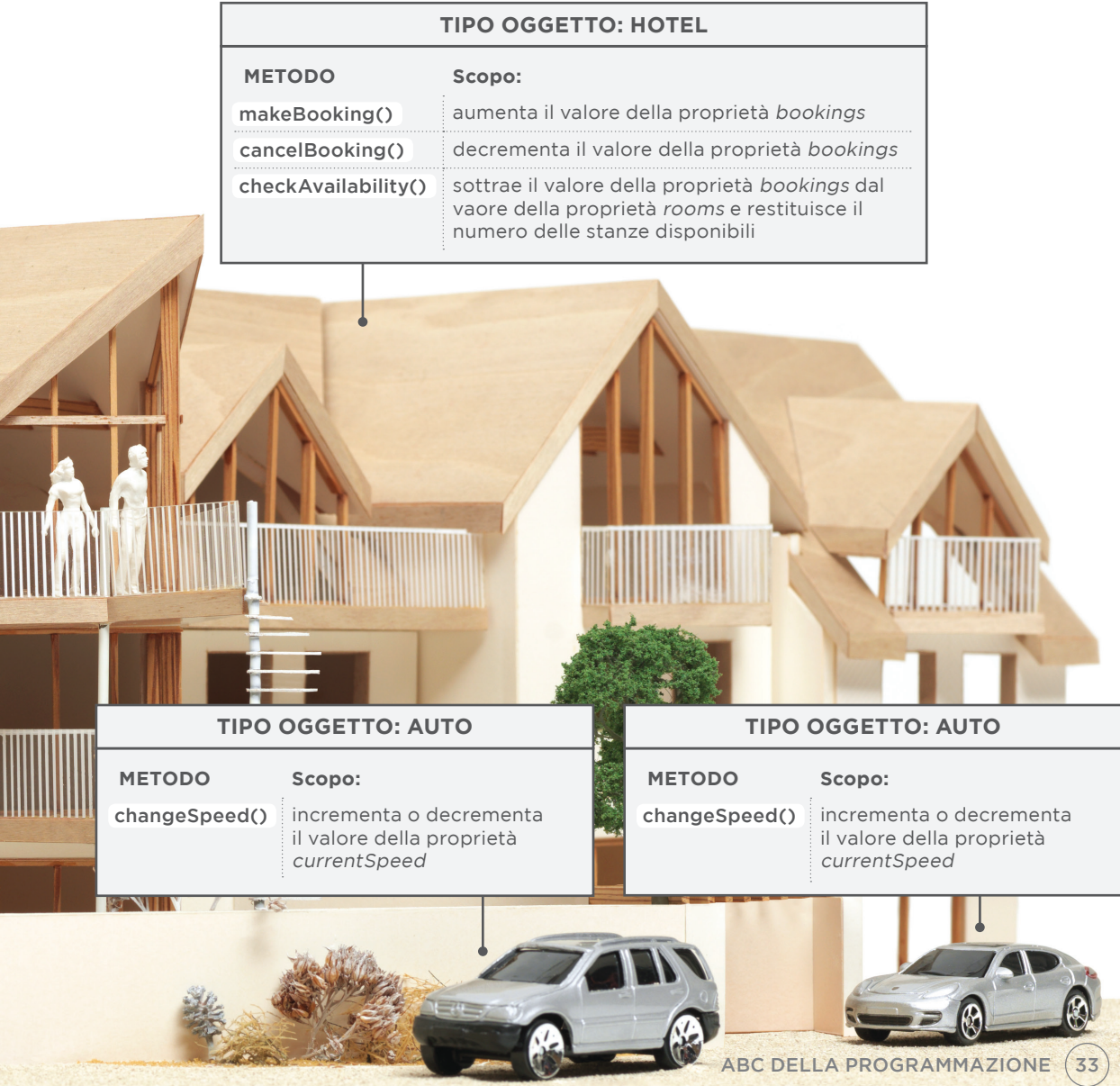
Quando utilizzate un metodo, non sempre avete bisogno di sapere *come* svolge un determinato compito; vi basta solo imparare a porre la domanda corretta e a interpretare le risposte fornite dal metodo.

L'OGGETTO HOTEL

Gli hotel vengono normalmente interpellati per sapere se hanno stanze libere. Per rispondere a questa domanda, si può scrivere un metodo che sottrae il numero di prenotazioni al numero totale delle stanze. Possono essere utilizzati anche dei metodi che aumentano e riducono il valore della proprietà prenotazioni nel momento in cui le stanze vengono prenotate o cancellate.

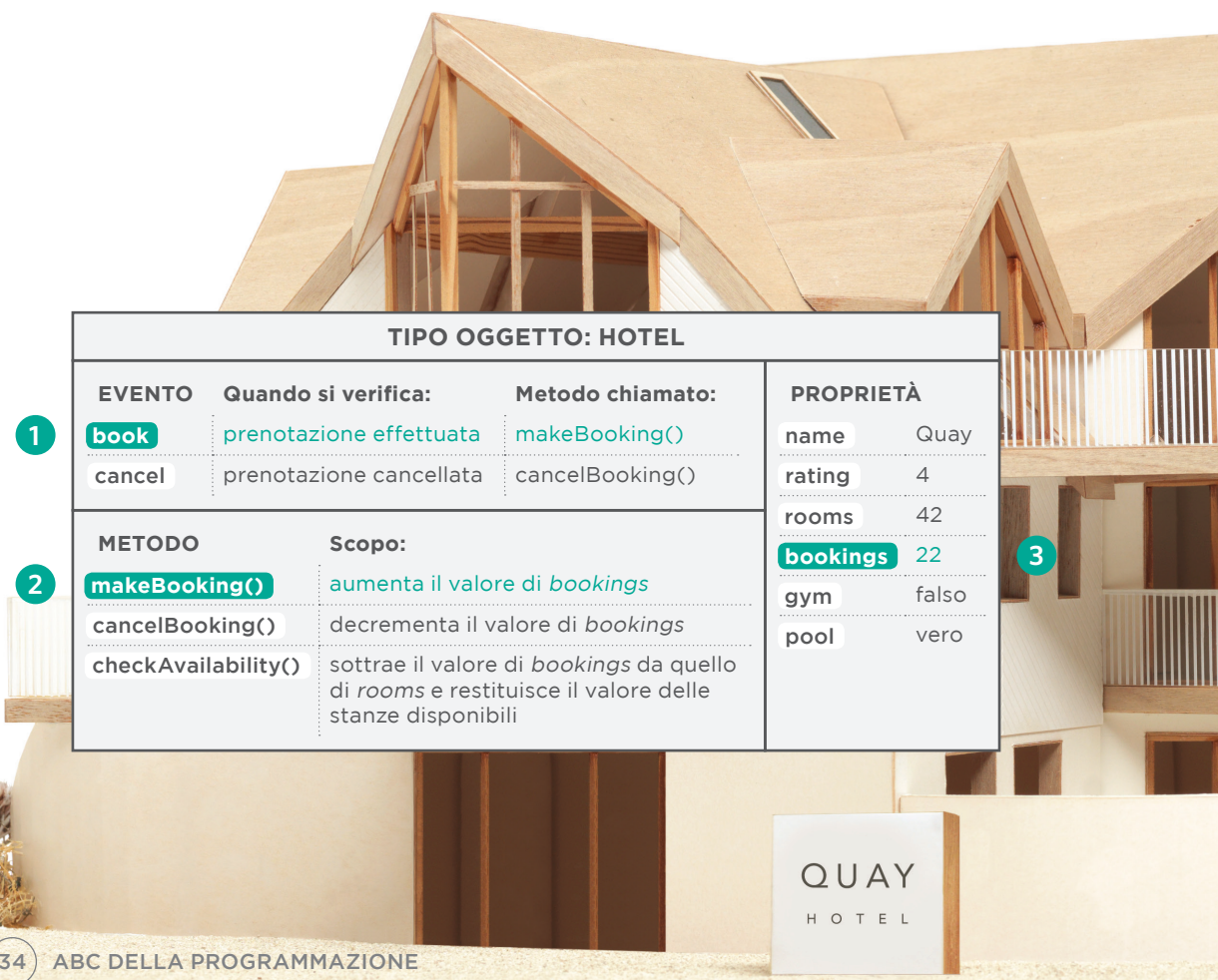
GLI OGGETTI AUTO

Il valore della proprietà `currentSpeed` deve variare nel momento in cui il conducente accelera o frena. Il codice per incrementare o decrementare il valore della proprietà `currentSpeed` dovrà essere scritto in un metodo, il quale potrebbe chiamarsi `changeSpeed()`.



RICAPITOLANDO

I computer utilizzano dei dati per creare modelli di oggetti che sono presenti nel mondo reale. Gli eventi, i metodi e le proprietà di un oggetto sono tutti elementi correlati: gli eventi possono scatenare dei metodi e i metodi possono leggere o modificare le proprietà di un oggetto.



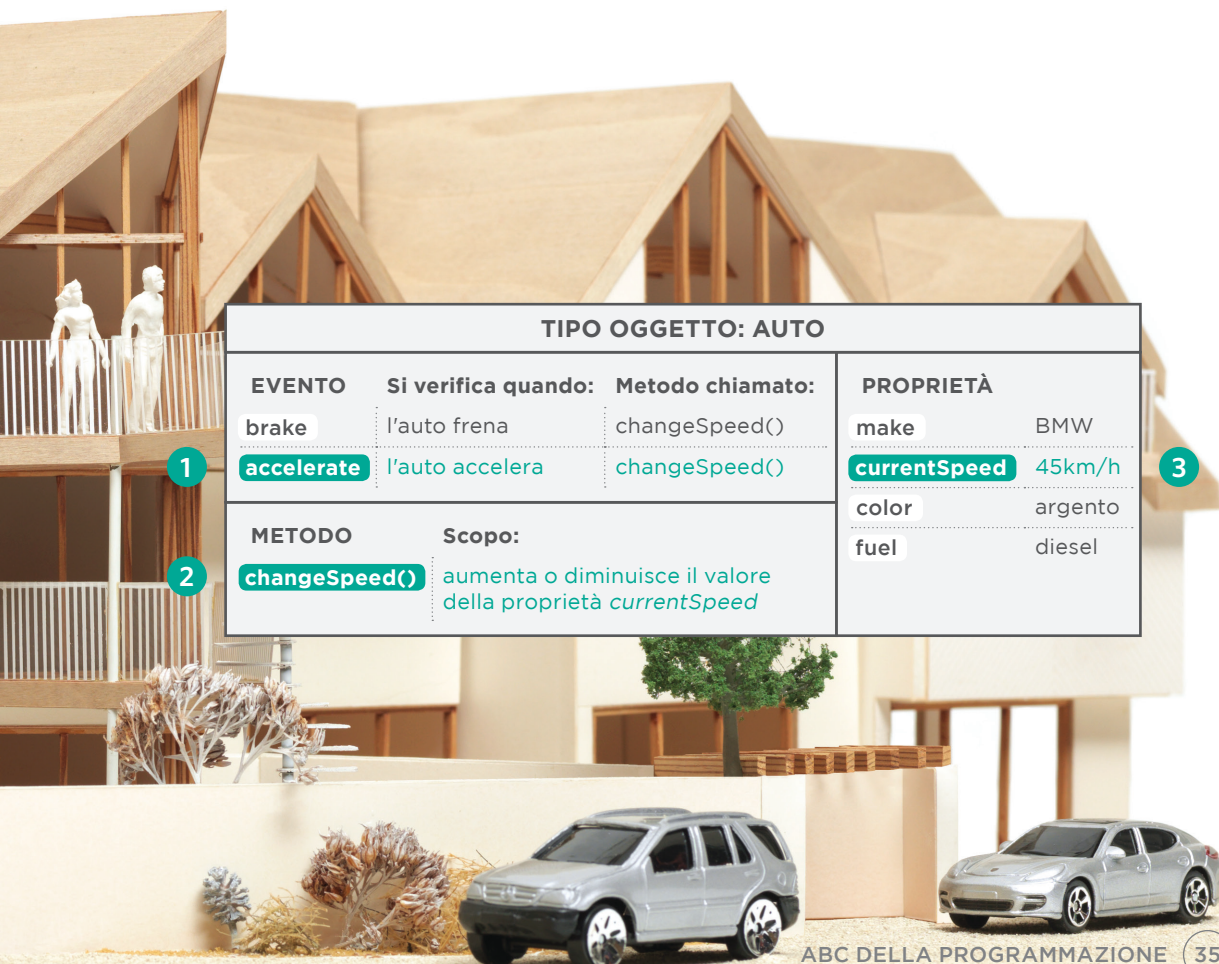
TIPO OGGETTO: HOTEL			
EVENTO	Quando si verifica:	Metodo chiamato:	PROPRIETÀ
1 book	prenotazione effettuata	makeBooking()	name Quay
cancel	prenotazione cancellata	cancelBooking()	rating 4
METODO	Scopo:		rooms 42
2 makeBooking()	aumenta il valore di <i>bookings</i>		3 bookings 22
cancelBooking()	decrementa il valore di <i>bookings</i>		gym falso
checkAvailability()	sottrae il valore di <i>bookings</i> da quello di <i>rooms</i> e restituisce il valore delle stanze disponibili		pool vero

OGGETTO HOTEL

- 1. Quando viene effettuata una prenotazione, viene lanciato l'evento book.
- 2. L'evento book lancia il metodo makeBooking(), che incrementa il valore della proprietà bookings.
- 3. Il valore della proprietà bookings viene modificato per riflettere il numero di stanze ancora disponibili nell'hotel.

OGGETTI AUTO

- 1. Quando il conducente preme sull'acceleratore, viene lanciato l'evento accelerate.
- 2. L'evento accelerate richiama il metodo changeSpeed(), che a sua volta incrementa il valore della proprietà currentSpeed.
- 3. Il valore della proprietà currentSpeed riflette la velocità dell'auto.



TIPO OGGETTO: AUTO			
EVENTO	Si verifica quando:	Metodo chiamato:	PROPRIETÀ
brake	l'auto frena	changeSpeed()	make BMW
accelerate	l'auto accelera	changeSpeed()	currentSpeed 45km/h
METODO	Scopo:		color argento
changeSpeed()	aumenta o diminuisce il valore della proprietà currentSpeed		fuel diesel

I BROWSER WEB SONO PROGRAMMI CHE SI BASANO SU OGGETTI

Abbiamo visto come i dati possano essere utilizzati per creare modelli di un hotel o di un'auto. I browser web creano modelli simili della pagina web che stanno mostrando e della finestra in cui viene visualizzata.

L'OGGETTO WINDOW

Nella pagina a destra, potete vedere un modello di un computer con un browser aperto sullo schermo.

Il browser rappresenta ciascuna finestra o scheda utilizzando un oggetto `window`. La proprietà `location` dell'oggetto `window` indicherà l'indirizzo URL della pagina corrente.

L'OGGETTO DOCUMENT

La pagina web corrente caricata in ciascuna finestra è modellata tramite l'oggetto `document`.

La proprietà `title` dell'oggetto `document` ci dice ciò che si trova fra il tag di apertura `<title>` e il tag di chiusura `</title>` della pagina web, mentre la proprietà `lastModified` dell'oggetto `document` ci dice la data in cui questa pagina è stata aggiornata per l'ultima volta.

TIPO OGGETTO: WINDOW

PROPRIETÀ

location <http://www.javascriptbook.com/>

TIPO OGGETTO: DOCUMENT

PROPRIETÀ

URL <http://www.javascriptbook.com/>

lastModified 09/04/2014 15:33:37

title Learn JavaScript & jQuery -
A book that teaches you
in a nicer way

L'OGGETTO DOCUMENT RAPPRESENTA LA PAGINA HTML

Utilizzando l'oggetto `document`, potete accedere e anche modificare i contenuti che gli utenti vedono nella pagina e rispondere al modo in cui gli utenti interagiscono con essa.

Come altri oggetti che rappresentano elementi del mondo reale, l'oggetto `document` ha le seguenti caratteristiche.

PROPRIETÀ

Le proprietà descrivono le caratteristiche della pagina web corrente (per esempio il titolo della pagina).

METODI

I metodi svolgono dei compiti associati al documento attualmente caricato nel browser (per esempio richiedere informazioni a un determinato elemento o aggiungere nuovi contenuti).

EVENTI

È possibile rispondere agli eventi che si verificano, come quando un utente fa clic o tap su un elemento.

Poiché tutti i principali browser web implementano l'oggetto `document` nello stesso modo, gli sviluppatori di browser:

- hanno già implementato delle proprietà alle quali fare accesso per scoprire qual è la pagina corrente caricata nel browser;
- hanno già scritto dei metodi che permettono di svolgere i compiti più comuni che riguardano una pagina HTML.

In questo libro vedremo come lavorare con questo oggetto. In pratica, l'oggetto `document` è uno dei tanti oggetti supportati dalla maggior parte dei browser. Quando il browser crea un modello di una pagina web, non crea solo un oggetto `document`, ma crea un nuovo oggetto per ogni elemento della pagina. Insieme, questi oggetti, sono descritti nel modello **Document Object Model**, che incontreremo nel Capitolo 5.

TIPO OGGETTO: DOCUMENT

PROPRIETÀ

URL	<code>http://www.javascriptbook.com/</code>
lastModified	<code>09/04/2014 15:33:37</code>
title	<code>Learn JavaScript & jQuery - A book that teaches you in a nicer way</code>

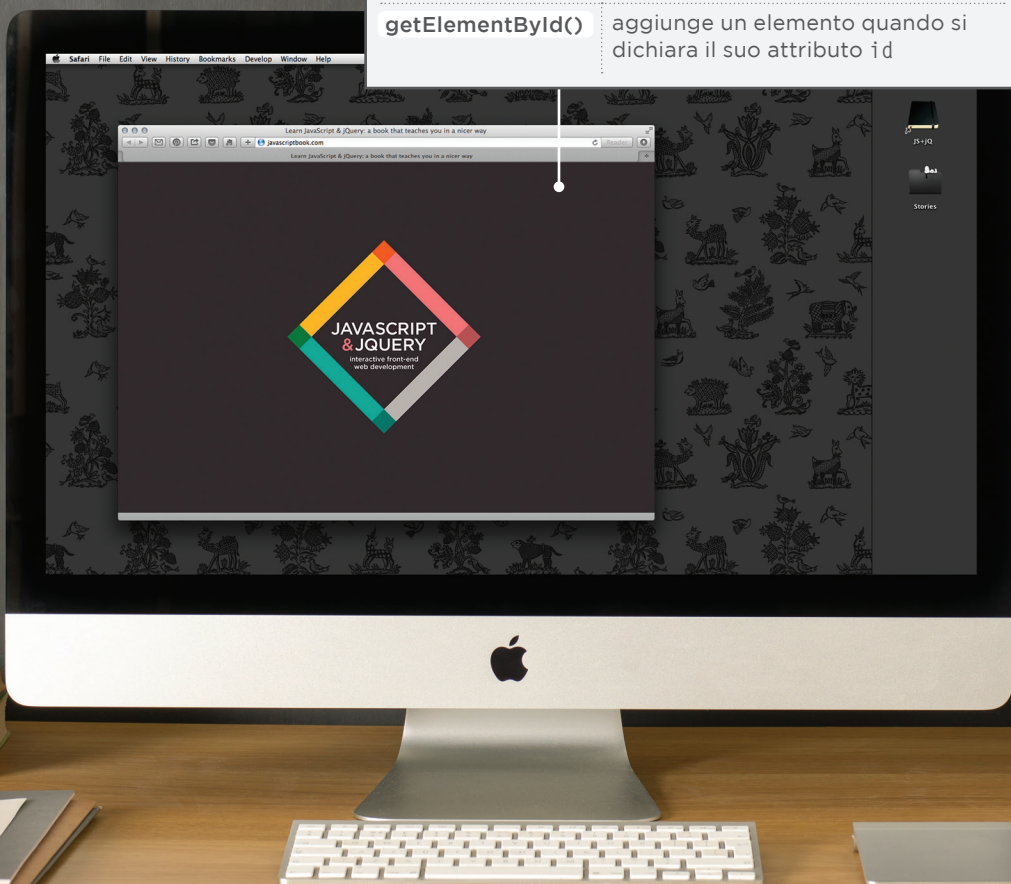
EVENTO Quando si verifica:

load	fine del caricamento di pagina e attività
click	l'utente fa clic con il mouse sulla pagina
keypress	l'utente preme un tasto

METODO

Scopo:

write()	aggiunge contenuti al documento
getElementById()	aggiunge un elemento quando si dichiara il suo attributo id



LA PAGINA WEB DAL PUNTO DI VISTA DEL BROWSER

Per imparare a modificare il contenuto di una pagina HTML con JavaScript, occorre sapere in quale modo il browser interpreta il codice HTML e gli applica uno stile.

1: RICEVE UNA PAGINA COME CODICE HTML

Ogni pagina di un sito web può essere vista come un documento distinto. In pratica, il Web è costituito da molti siti, ognuno costituito da uno o più documenti.

2: CREA UN MODELLO DELLA PAGINA E LO SALVA IN MEMORIA

Il modello visibile nella pagina a destra è una rappresentazione di una semplicissima pagina. La sua struttura ricorda un albero genealogico. In cima al modello vi è l'oggetto **document**, che rappresenta l'intero documento.

Sotto l'oggetto **document**, ogni casella rappresenta un nodo. Ognuno di questi nodi è un altro oggetto. Questo esempio mostra tre tipi di nodi, che rappresentano gli elementi, il testo all'interno degli elementi e l'attributo.

3: USA UN MOTORE DI RENDERING PER MOSTRARE LA PAGINA

Se non vi è codice CSS, il motore di rendering applicherà gli stili predefiniti. Tuttavia il codice HTML di questo esempio si collega a un foglio stile CSS, pertanto il browser richiede tale file e, in base a questo, adatta la visualizzazione della pagina. Quando il browser legge le regole CSS, il motore di rendering le elabora e le applica agli elementi corrispondenti. È così che il browser colloca gli elementi nella posizione corretta, applicando colori, tipi di caratteri e altre caratteristiche.

Tutti i principali browser usano l'interprete JavaScript per tradurre le istruzioni JavaScript in istruzioni interpretabili dal computer.

Quando si usa JavaScript nel browser, una parte del browser funge da interprete (il motore per script).

L'interprete prende le istruzioni JavaScript e le traduce in istruzioni interpretabili dal browser, il quale esegue i compiti richiesti.

In un linguaggio di programmazione interpretato, come JavaScript, ogni riga di codice viene tradotta indipendentemente, mentre lo script viene eseguito.

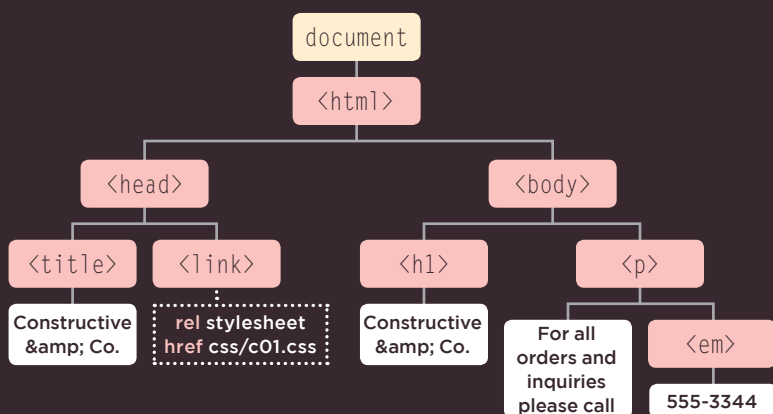
```

<!DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive & Co.</h1>
    <p>For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>

```

1

Il browser riceve una pagina HTML.



2

Crea un modello della pagina e lo salva in memoria.

- OGGETTO
- ELEMENTO
- TESTO
- ⋮ ATTRIBUTI



3

Presenta la pagina sullo schermo utilizzando un motore di rendering.

RIEPILOGO

ABC DELLA PROGRAMMAZIONE

B: I computer e il mondo che li circonda

- ▶ I computer creano modelli del mondo utilizzando i dati.
- ▶ I modelli utilizzano degli oggetti per rappresentare gli elementi fisici. Gli oggetti possono avere *proprietà*, che ci parlano dell'oggetto, *metodi*, che svolgono operazioni utilizzando le proprietà dell'oggetto, *eventi*, che vengono lanciati quando un utente interagisce con il computer.
- ▶ I programmatori possono scrivere codice per dire, in pratica: "Quando si verifica questo evento, esegui questo codice".
- ▶ I browser web utilizzano il codice HTML per creare un modello della pagina web. Ciascun elemento crea il proprio nodo (che è un altro tipo di oggetto).
- ▶ Per rendere interattive le pagine web, potete scrivere del codice che sfrutti il modello che il browser ha predisposto per la pagina web.



1/c

COME SI SCRIVE
UNO SCRIPT
PER UNA PAGINA WEB

COME HTML, CSS E JAVASCRIPT LAVORANO INSIEME

Prima di affrontare il linguaggio JavaScript, parliamo del modo in cui si colloca, insieme al codice HTML e CSS, nelle pagine web.

Gli sviluppatori web normalmente impiegano tre linguaggi per creare le pagine web: HTML, CSS e JavaScript.

Quando possibile, cercate di mantenere i tre linguaggi in file distinti: una pagina HTML che si collega ai file CSS e JavaScript.

Ogni linguaggio forma un livello distinto, con uno scopo distinto. Ogni livello, da sinistra a destra, si basa sul precedente

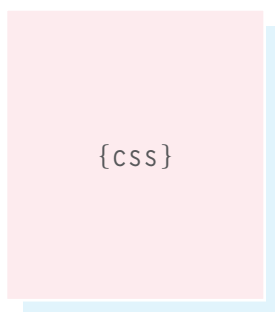


`<html>`

LIVELLO DEI CONTENUTI

file .html

Questo è il livello in cui si trovano i contenuti della pagina. Il codice HTML dà struttura alla pagina e aggiunge i suoi elementi semantici.

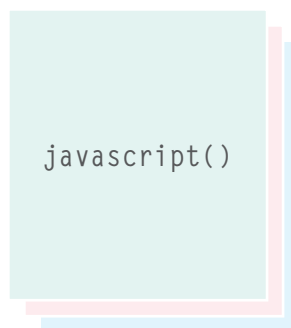


`{css}`

LIVELLO DELLA PRESENTAZIONE

file .css

Le regole CSS migliorano l'aspetto della pagina HTML, chiedendo di presentare i contenuti in modo particolare: gli sfondi, i bordi, le dimensioni dei riquadri, i colori, i tipi di caratteri e così via.



`javascript()`

LIVELLO DEI COMPORTAMENTI

file .js

In questo livello si controlla il comportamento della pagina, aggiungendo l'interattività. L'idea è quella di tenere il più possibile il codice JavaScript all'interno di file distinti.

I programmatori chiamano questa suddivisione **separazione degli ambiti** (*separation of concerns*).

ARRICCHIMENTO PROGRESSIVO

Questi tre livelli costituiscono la base di un noto approccio alla costruzione delle pagine web, chiamato "arricchimento progressivo".

Mano a mano che sul mercato si affacciavano sempre più apparecchi collegabili a Internet, questo concetto è stato adottato in modo sempre più esteso.

Non si tratta solo di variare le dimensioni dello schermo, ma anche di cambiare le velocità e le funzionalità offerte da ciascun dispositivo.

Inoltre, alcuni utenti disattivano JavaScript, ed è necessario assicurarsi che la pagina web funzioni anche per loro

Constructive & Co.

For all orders and inquiries please call 555-3344



HTML PURO

Partendo dal livello HTML è possibile concentrarsi sugli elementi più importanti del sito: i contenuti.

Trattandosi di puro HTML, questo livello dovrà operare con tutti i tipi di dispositivi, essere accessibile a tutti gli utenti e caricarsi rapidamente anche con connessioni lente.

HTML+CSS

Aggiungendo delle regole CSS, in un file distinto, separiamo le regole, che controllano l'aspetto della pagina, dai contenuti.

Potete utilizzare lo stesso foglio stile in tutto il vostro sito, accelerandone il caricamento e facilitandone la manutenzione. Oppure potete applicare fogli stile differenti agli stessi contenuti, per creare modalità di visualizzazione differenti degli stessi dati.

HTML+CSS+JAVASCRIPT

Il codice JavaScript va aggiunto per ultimo, per migliorare l'utilizzabilità della pagina o l'esperienza interattiva col sito.

Il fatto che JavaScript sia a parte significa anche che la pagina funzionerà anche se l'utente non può o non vuole caricare codice JavaScript. È anche possibile riutilizzare il codice su più pagine, accelerando il caricamento del sito e facilitandone la manutenzione.

CREARE UN SEMPLICE SCRIPT JAVASCRIPT

Il codice JavaScript è puro testo, proprio come quello HTML e CSS. Pertanto non sono necessari nuovi strumenti per creare uno script. Qui vediamo una pagina HTML con un saluto che cambia in base all'ora.

1 Create una cartella in cui inserire l'esempio, chiamata c01, poi lanciate l'editor di codice e inserite il testo qui a destra.

Un file JavaScript non è che un puro file di testo (come i file HTML CSS), ma con l'estensione .js, pertanto salvate questo file con il nome add-content.js

Non preoccupatevi, per il momento, del significato del codice; quello che ci interessa è provare a creare uno script e a collegarlo a una pagina HTML.

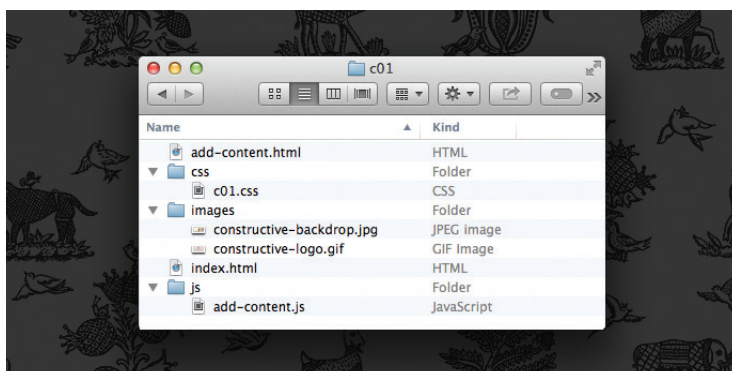
2 Caricate il CSS e le immagini di questo esempio dal sito web che accompagna questo libro: www.javascriptbook.com

Per mantenere organizzati i file, così come i file CSS spesso si trovano in una cartella chiamata styles o css, i file JavaScript potrebbero trovarsi in una cartella chiamata scripts, javascript o js. In questo caso, salvate il file in una cartella chiamata js.

```
var today = new Date();
var hourNow = today.getHours();
var greeting;

if (hourNow > 18) {
    greeting = 'Good evening!';
} else if (hourNow > 12) {
    greeting = 'Good afternoon!';
} else if (hourNow > 0) {
    greeting = 'Good morning!';
} else {
    greeting = 'Welcome!';
}

document.write('<h3>' + greeting + '</h3>');
```



Ecco la struttura dei file che verrà dall'esempio. Ricordate che le lettere maiuscole e minuscole sono considerate differenti.

COLLEGARE UN FILE JAVASCRIPT A UNA PAGINA HTML

Quando volete utilizzare del codice JavaScript in una pagina web, potete utilizzare l'elemento HTML `<script>` per chiedere al browser di caricare lo script. L'attributo `src` dirà dove si trova il file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive & Co.</h1>
    <script src="js/add-content.js"></script>
    <p>For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>
```

3 Nell'editor di codice, inserite il codice HTML qui a sinistra. Salvate il file con il nome `add-content.html`.

L'elemento HTML `<script>` carica nella pagina il file JavaScript. Il suo attributo `src` ne specifica il percorso.

Ciò chiede al browser di trovare e caricare il file `script` (esattamente come l'attributo `src` di un tag `<img>`).



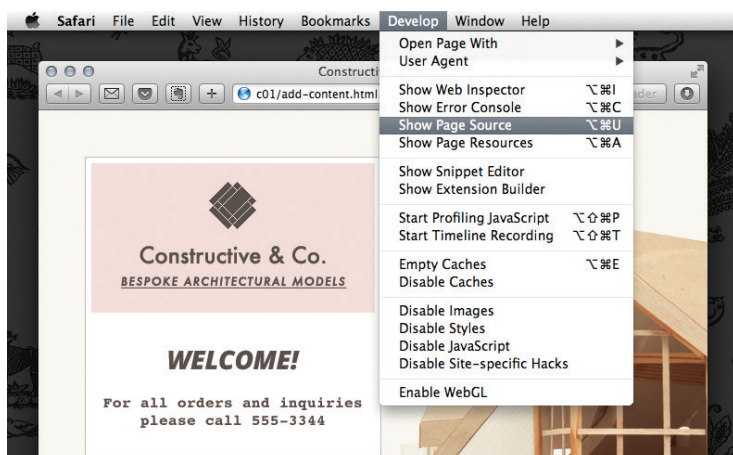
4 Aprite nel browser il file `add-content.html`. Il codice JavaScript aggiungerà alla pagina un saluto (in questo caso *Good Afternoon!*). Questo saluto è stato generato dal file JavaScript, non è presente nel file HTML.

Nota: Internet Explorer talvolta impedisce di lanciare codice JavaScript quando si carica una pagina dal disco rigido. In questo caso usate i browser Chrome, Firefox, Opera o Safari.

IL CODICE SORGENTE NON VIENE TOCCATO

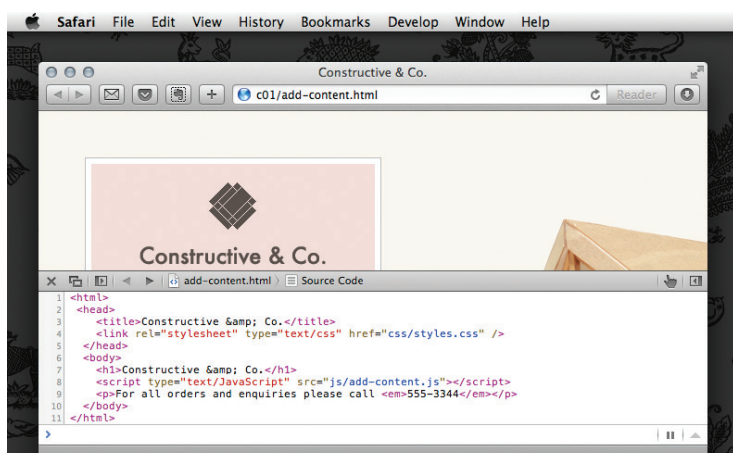
Se provate a osservare il codice sorgente dell'esempio che avete appena creato, noterete che il codice HTML non è stato toccato.

5 Dopo aver provato l'esempio nel browser, osservate il codice sorgente della pagina. Questa opzione si trova normalmente nel menu *Visualizza, Strumenti o Sviluppo* del browser.)



6 Il codice sorgente della pagina web non presenta il nuovo elemento che è stato aggiunto alla pagina, ma solo il riferimento al file JavaScript.

Nel corso del libro, scoprirete che la maggior parte degli script viene aggiunta appena prima del tag di chiusura `</body>`. In genere questo è il luogo migliore per caricare gli script.



DOVE COLLOCARE LO SCRIPT NELLA PAGINA

Potete anche trovare del codice JavaScript nel codice HTML, fra i tag `<script>` e `</script>`, ma è sempre meglio collocare gli script in un file a parte.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive & Co.</h1>
    <script>document.write('<h3>Welcome!</h3>');
    </script>
    <p>For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>
```

7 Infine, provate ad aprire il file HTML, a eliminare l'attributo `src` dal tag `<script>` e ad aggiungere il nuovo codice rappresentato a sinistra fra il tag di apertura `<script>` e il tag di chiusura `</script>`. L'attributo `src` non è più necessario, poiché il codice JavaScript si troverà direttamente nella pagina HTML.

Come già detto a pagina 44, è meglio non "mescolare" il codice JavaScript e HTML in questo modo. Tuttavia è il caso di menzionare questa possibilità.

8 Aprite il file HTML nel browser web e di nuovo verrà presentato il saluto.

Come potete immaginare, `.write()` scrive il contenuto nel documento (la pagina web). Si tratta di un modo semplice per aggiungere contenuti alla pagina, ma non sempre è il migliore. Il Capitolo 5 analizza vari modi per intervenire sul contenuto di una pagina.



USARE GLI OGGETTI E I METODI

Questa riga di codice JavaScript mostra come utilizzare gli oggetti e i metodi. In gergo si dice **richiamare** un metodo di un oggetto.

L'oggetto **document** rappresenta l'intera pagina web. Tale oggetto è implementato da tutti i browser web e potete utilizzarlo semplicemente specificandone il nome.

Il metodo **write()** dell'oggetto **document** consente di scrivere nuovi contenuti nella pagina in cui è situato l'elemento **<script>**.

OGGETTO

METODO

```
document.write('Good afternoon!');
```

OPERATORE MEMBRO

L'oggetto **document** offre vari metodi e proprietà, i quali sono i **membri** di tale oggetto.

Potete accedere ai membri di un oggetto specificando un punto fra il nome dell'oggetto e quello del membro. Il punto è detto **operatore membro**.

PARAMETRI

Quando un metodo richiede delle informazioni aggiuntive per poter operare, tali informazioni devono essere specificate fra parentesi.

Ogni informazione specificata rappresenta un **parametro** del metodo. In questo caso, il metodo **write()** deve sapere che cosa scrivere nella pagina.

Dietro le quinte, il browser utilizza molto più codice per visualizzare le parole, ma non c'è alcun motivo per chiedersi in quale modo il browser svolga questa operazione.

Basta sapere come richiamare l'oggetto e il metodo e come specificare le informazioni di cui ha bisogno per svolgere il compito. Il resto avverrà da solo.

Sono disponibili molti oggetti, come l'oggetto **document**, e molti metodi, come il metodo **write()**, utili per realizzare ogni genere di script.

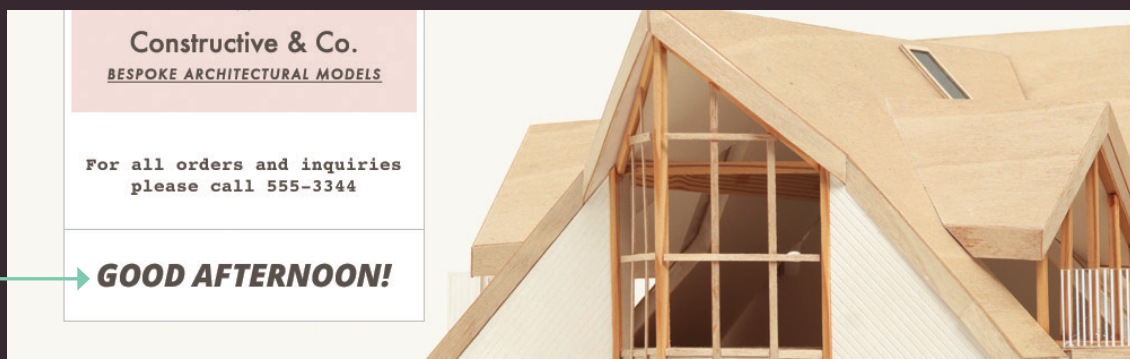
IL CODICE JAVASCRIPT VIENE ESEGUITO NEL PUNTO IN CUI SI TROVA NELLA PAGINA HTML

Quando il browser incontra un elemento `<script>`, carica lo script e controlla se deve svolgere qualche operazione.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive & Co.</h1>
    <p>For all orders and inquiries please call <em>555-3344</em></p>
    <script src="js/add-content.js"></script>
  </body>
</html>
```

Notate come l'elemento `<script>` possa essere spostato sotto il primo paragrafo; questo cambierà la posizione del saluto nella pagina.

Questo ha ovviamente implicazioni sulla collocazione degli elementi `<script>`, il che può influenzare il tempo di caricamento della pagina (vedi pagina 356).



RIEPILOGO

ABC DELLA PROGRAMMAZIONE

C: Come si scrive uno script per una pagina web

- ▶ È meglio mantenere il codice JavaScript in un file a parte. I file JavaScript sono comuni file di testo (come le pagine HTML e i fogli stile CSS), ma adottano l'estensione `.js`.
- ▶ L'elemento HTML `<script>`, presente nelle pagine HTML, dice al browser di caricare il file JavaScript, un po' come si usa l'elemento `<link>` per caricare un file CSS.
- ▶ Osservando il codice sorgente della pagina nel browser, il codice introdotto da JavaScript non risulta presente nel codice HTML, poiché lo script opera sul modello della pagina web creato dal browser.