

10

INTRODUZIONE A CSS

- ▶ A che cosa serve CSS
- ▶ Come funziona CSS
- ▶ Regole, proprietà e valori

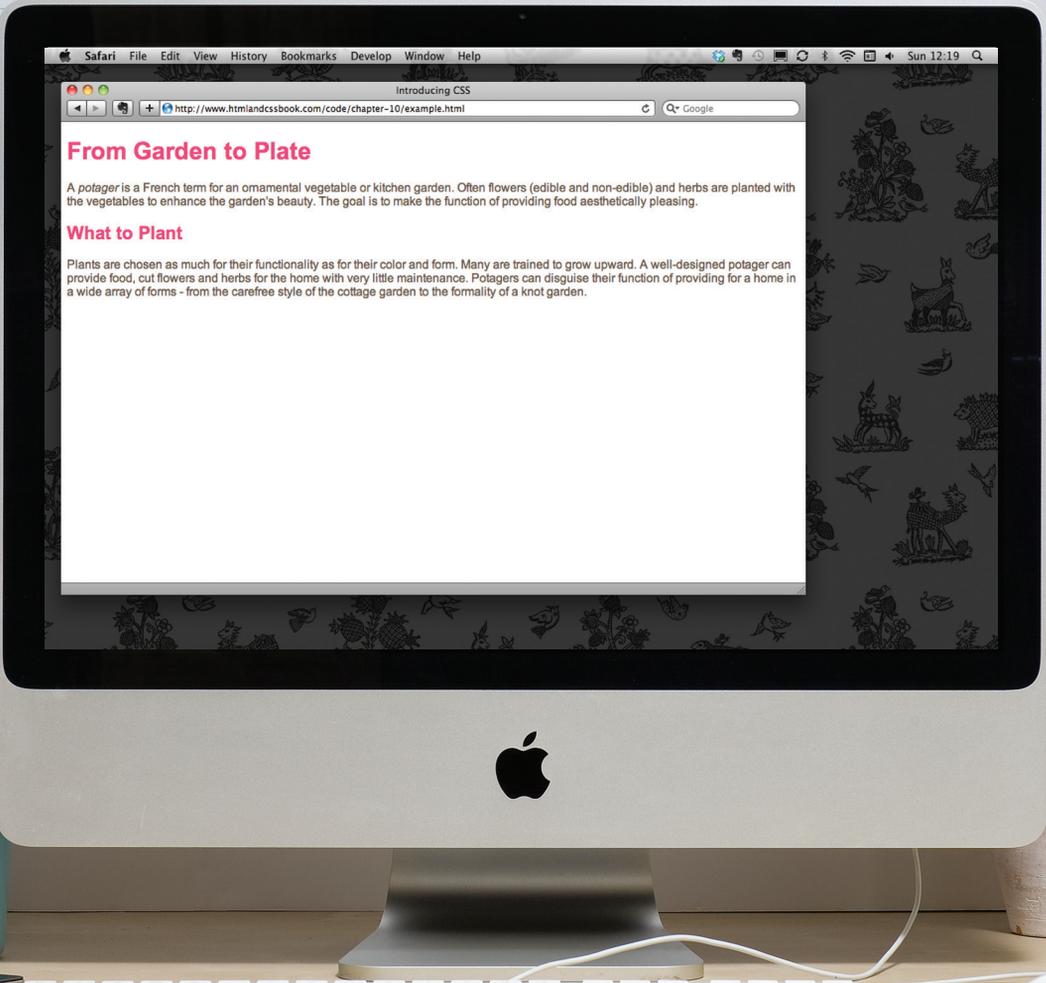
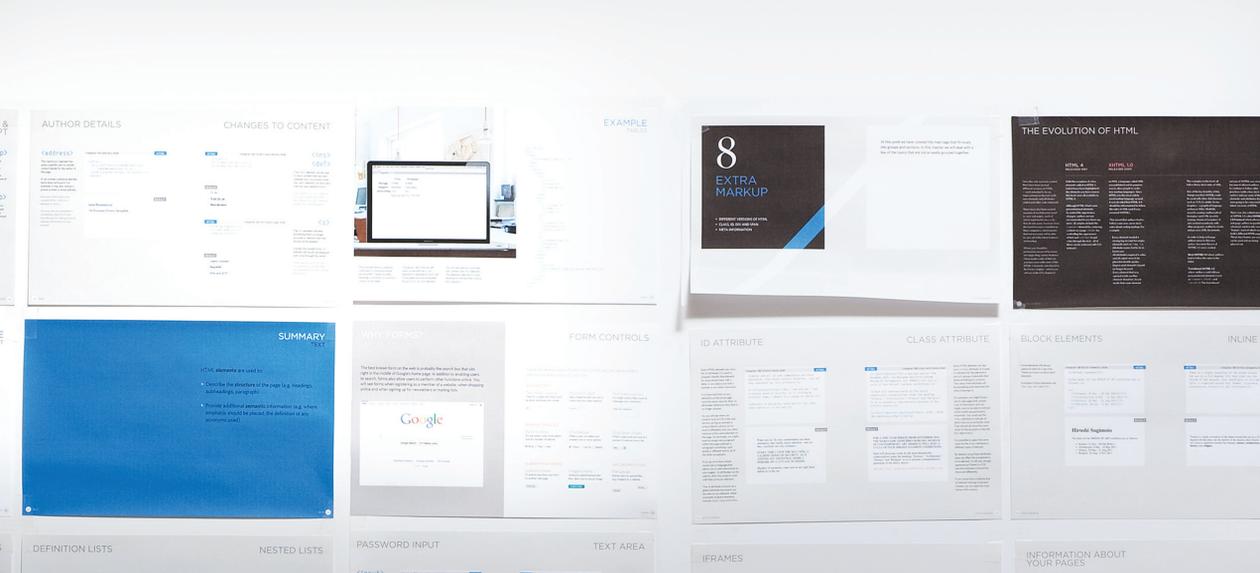
In questo capitolo vedremo come rendere più attraenti le pagine web, controllandone l'aspetto tramite elementi CSS.

Il linguaggio CSS consente di creare delle regole che specificano l'aspetto di un elemento. Per esempio, potete richiedere che lo sfondo della pagina sia di colore crema, che tutti i paragrafi vengano rappresentati in grigio, con il tipo di carattere Arial o che tutti i titoli di primo livello debbano essere in blu, corsivo e con tipo di carattere Times.

Dopo aver imparato a scrivere una regola CSS, l'apprendimento del linguaggio riguarda principalmente lo studio delle varie proprietà utilizzabili. In questo capitolo affronteremo i seguenti argomenti.

- Introduzione al funzionamento di CSS.
- Come scrivere regole CSS.
- Come applicare le regole CSS alle pagine HTML.

I capitoli successivi di questa parte del volume si occuperanno invece di tutte le varie proprietà CSS utilizzabili.



Safari File Edit View History Bookmarks Develop Window Help

Introducing CSS

http://www.htmlandcssbook.com/code/chapter-10/example.html

From Garden to Plate

A *potager* is a French term for an ornamental vegetable or kitchen garden. Often flowers (edible and non-edible) and herbs are planted with the vegetables to enhance the garden's beauty. The goal is to make the function of providing food aesthetically pleasing.

What to Plant

Plants are chosen as much for their functionality as for their color and form. Many are trained to grow upward. A well-designed potager can provide food, cut flowers and herbs for the home with very little maintenance. Potagers can disguise their function of providing for a home in a wide array of forms - from the carefree style of the cottage garden to the formality of a knot garden.



CAPIRE CSS: PENSARE DENTRO AL BOX

Il modo migliore per capire come funziona CSS consiste nell'immaginare che un box invisibile riquadri ogni elemento HTML.

A destra, potete vedere una semplice pagina HTML.

Nella pagina a fronte vedete la stessa pagina HTML, con dei rettangoli che indicano i vari elementi, con lo scopo di immaginare il modo in cui CSS tratta ciascun elemento, ovvero inserito nel proprio box.

The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

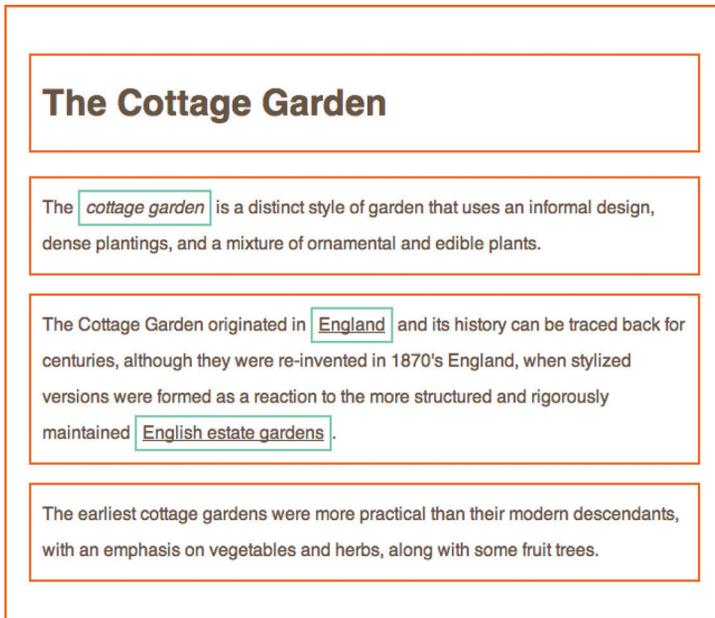
ELEMENTI DI BLOCCO E IN LINEA

Nelle pagine 185 e 186 abbiamo mostrato la differenza esistente fra elementi di blocco e in linea e il modo in cui vengono visualizzati dai browser.

Gli elementi di blocco iniziano sempre su una nuova riga. Fra gli esempi vi sono gli elementi `<h1>`-`<h6>`, `<p>` e `<div>`.

Gli elementi in linea sono inseriti direttamente nel testo e non partono da una nuova riga. Fra gli esempi vi sono ``, `<i>`, ``, `` e ``.

CSS consente di creare regole che controllano il modo in cui ogni singolo box (e il suo contenuto) viene presentato.



In questo esempio, gli elementi di blocco vengono rappresentati con un rettangolo rosso e gli elementi in linea con un rettangolo verde.

L'elemento `<body>` crea il primo box e poi gli elementi `<h1>`, `<h2>`, `<p>`, `<i>` e `<a>` creano i propri box al suo interno.

Utilizzando CSS potete aggiungere una cornice attorno a questi box, specificare la sua larghezza e altezza o aggiungere un colore di sfondo. Potete anche controllare il testo contenuto nel box, per esempio il suo colore, le sue dimensioni e il tipo di carattere che utilizza.

ESEMPI DI STILI

BOX

Larghezza e altezza.
Bordo (colore, spessore e stile).
Colore di sfondo e immagini.
Posizione nella finestra del browser.

TESTO

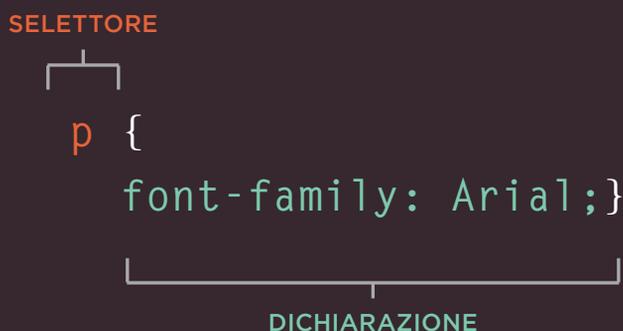
Tipo di carattere.
Dimensioni.
Colore.
Corsivo, grassetto, maiuscolo, minuscolo, maiuscoletto.

SPECIFICI

Vi sono anche dei modi specifici in cui si può specificare lo stile di determinati elementi, come liste, tabelle e moduli.

CSS ASSOCIA DELLE REGOLE DI STILE AGLI ELEMENTI HTML

CSS funziona associando delle regole agli elementi HTML. Queste regole governano il modo in cui deve essere visualizzato il contenuto degli elementi specificati. Una regola CSS è costituita da due parti: un **selettore** e una **dichiarazione**.



Questa regola indica che tutti gli elementi `<p>` devono essere presentati con il tipo di carattere Arial.

Il **selettore** indica l'elemento al quale si applica la regola. La stessa regola può essere applicata a più elementi, specificandoli uno dopo l'altro separati da una virgola.

La **dichiarazione** indica l'aspetto dello stile degli elementi indicati dal selettore. Le dichiarazioni sono suddivise in due parti (una proprietà e un valore) separate da un segno di due punti.

LE PROPRIETÀ CSS DETERMINANO L'ASPETTO DEGLI ELEMENTI

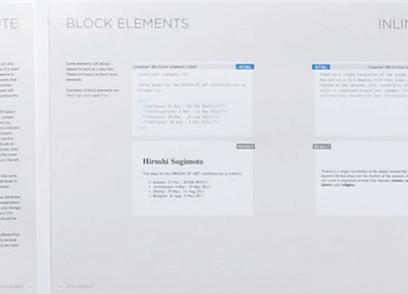
Le dichiarazioni CSS sono situate all'interno di parentesi graffe e ciascuna è costituita da due parti: una **proprietà** e un **valore**, separate da un segno di due punti. Si possono specificare varie proprietà in una dichiarazione, separandole tramite un punto e virgola.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;}  
└──┬──┘ └──┬──┘  
  PROPRIETÀ  VALORE
```

Questa regola indica che tutti gli elementi `<h1>`, `<h2>` e `<h3>` debbano essere presentati nel tipo di carattere Arial, in colore giallo.

Le **proprietà** indicano l'aspetto dell'elemento che si vuole modificare. Per esempio, il colore, il tipo di carattere, l'altezza, la larghezza e il bordo.

I **valori** specificano le impostazioni da utilizzare per le proprietà scelte. Per esempio, se volete specificare la proprietà del colore, allora il valore dovrà essere il colore da assegnare al testo di questi elementi.



ESEMPIO

INTRODUZIONE A CSS

Qui potete vedere una semplice pagina web modificata tramite codice CSS.

Questo esempio utilizza due documenti: il file HTML (`example.html`) e un file CSS distinto (`example.css`). La quinta linea riga del codice HTML utilizza l'elemento `<link>` per indicare dove è situato il file CSS.

Nella prossima pagina vedrete che le regole CSS possono essere collocate anche all'interno delle pagine HTML e vedremo quando è il caso di comportarsi in questo modo.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Introducing CSS</title>
    <link href="css/example.css" type="text/css"
          rel="stylesheet" />
  </head>
  <body>
    <h1>From Garden to Plate</h1>
    <p>A <i>potager</i> is a French term for an
      ornamental vegetable or kitchen garden ... </p>
    <h2>What to Plant</h2>
    <p>Plants are chosen as much for their functionality
      as for their color and form ... </p>
  </body>
</html>

body {
  font-family: Arial, Verdana, sans-serif;}
h1, h2 {
  color: #ee3e80;}
p {
  color: #665544;}
```

USO DI UN FILE CSS ESTERNO

<link>

L'elemento <link> può essere utilizzato in un documento HTML per dire al browser dove trovare il file CSS utilizzato per impostare lo stile della pagina. È un elemento che non prevede un tag di chiusura ed è situato all'interno dell'elemento <head>. Deve avere tre attributi.

href

Specifica il percorso del file CSS (normalmente è collocato in una cartella chiamata `css` o `styles`).

type

Questo attributo specifica il tipo del documento collegato. Il valore dovrebbe essere `text/css`.

rel

Specifica la relazione esistente fra la pagina HTML e il file cui è collegata. Il valore dovrà essere `stylesheet` quando si fa riferimento a un file CSS.

Una pagina HTML può usare più fogli di stile CSS. Dovrà però usare un elemento <link> per ogni file CSS utilizzato. Per esempio, alcuni autori utilizzano un file CSS per controllare la presentazione (come il tipo di carattere e il colore) e un altro per controllare la disposizione degli elementi.

chapter-10/using-external-css.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css"
          rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
      varieties. They are usually described as
      early, second early and maincrop.</p>
  </body>
</html>
```

chapter-10/styles.css

CSS

```
body {
  font-family: arial;
  background-color: rgb(185,179,175);}
h1 {
  color: rgb(255,255,255);}
```

OUTPUT

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

USO DI UN CSS INTERNO

HTML + CSS chapter-10/using-internal-css.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Internal CSS</title>
    <style type="text/css">
      body {
        font-family: arial;
        background-color: rgb(185,179,175);}
      h1 {
        color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
      varieties. They are usually described as
      early, second early and maincrop.</p>
  </body>
</html>
```

OUTPUT

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

In HTML 4 e XHTML Transitional, è anche possibile applicare l'attributo `style` alla maggior parte degli elementi che appaiono nel corpo della pagina. Le regole CSS che

compaiono all'interno del valore dell'attributo si applicano solo a tale elemento. Si dovrebbe evitare di utilizzare questo attributo in ogni nuovo sito, ma è il caso di menzionare questa

<style>

Potete anche includere delle regole CSS all'interno della pagina HTML, collocandole in un elemento `<style>` che normalmente è situato nell'elemento `<head>` della pagina.

L'elemento `<style>` dovrebbe utilizzare l'attributo `type` per indicare che gli stili sono specificati in CSS. Il valore dovrà quindi essere `text/css`.

Quando si realizza un sito con più di una pagina, è sempre meglio utilizzare fogli di stile CSS esterni, per i seguenti motivi.

- Consente a tutte le pagine di utilizzare le stesse regole di stile, senza doverle ripetere per ciascuna pagina.
- Mantiene il contenuto separato rispetto all'aspetto della pagina.
- Offre la possibilità di cambiare gli stili utilizzati dalle pagine, intervenendo su un solo file e non su ogni singola pagina.

tecnica poiché la troverete impiegata nel vecchio codice. Ecco un esempio che interviene sul colore del testo di un unico paragrafo, applicando il rosso: `<p style="color:red;">`

SELETTORI CSS

Vi sono vari tipi di selettori CSS che consentono di indirizzare le regole verso specifici elementi di un documento HTML.

La tabella che si trova nella pagina a fronte introduce i selettori CSS più comunemente utilizzati.

In questa pagina, trovate un file HTML che mostra a quali elementi si applicano questi selettori.

I selettori CSS distinguono fra lettere maiuscole e minuscole e pertanto i nomi di elementi e i valori degli attributi devono essere esatti.

Vi sono alcuni selettori più avanzati che consentono di selezionare gli elementi sulla base degli attributi e dei loro valori, argomento di cui parleremo a pagina 292.

Internet Explorer 7 è stata la prima versione del browser a supportare gli ultimi due selettori della tabella (i selettori sibling), e pertanto il loro uso è meno comune rispetto agli altri selettori elencati.

chapter-10/css-selectors.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Selectors</title>
  </head>
  <body>
    <h1 id="top">Kitchen Garden Calendar</h1>
    <p id="introduction">Here you can read our
      handy guide about what to do when.</p>
    <h2>Spring</h2>
    <ul>
      <li><a href="mulch.html">
        Spring mulch vegetable beds</a></li>
      <li><a href="potato.html">
        Plant out early potatoes</a></li>
      <li><a href="tomato.html">
        Sow tomato seeds</a></li>
      <li><a href="beet.html">
        Sow beet seeds</a></li>
      <li><a href="zucchini.html">
        Sow zucchini seeds</a></li>
      <li><a href="rhubarb.html">
        Deadhead rhubarb flowers</a></li>
    </ul>
    <p class="note">
      This page was written by
      <a href="mailto:ivy@example.org">
        ivy@example.org</a> for
      <a href="http://www.example.org">Example</a>.
    </p>
    <p>
      <a href="#top">Top of page</a>
    </p>
  </body>
</html>
```

SELETTORE	SIGNIFICATO	ESEMPIO
SELETTORE UNIVERSALE	Si applica a tutti gli elementi del documento	<code>* {}</code> Tutti gli elementi della pagina
SELETTORE DI TIPO	Corrisponde a specifici elementi	<code>h1, h2, h3 {}</code> Individuano gli elementi <code><h1></code> , <code><h2></code> e <code><h3></code>
SELETTORE DI CLASSE	Corrisponde a un elemento il cui attributo <code>class</code> ha il valore corrispondente a quanto viene specificato dopo il punto	<code>.note {}</code> Individua ogni elemento il cui attributo <code>class</code> ha valore <code>note</code> <code>p.note {}</code> Individua solo gli elementi <code><p></code> il cui attributo <code>class</code> ha valore <code>note</code>
SELETTORE DI ID	Corrisponde a un elemento il cui attributo <code>id</code> ha valore corrispondente a quello specificato dopo il simbolo <code>#</code>	<code>#introduction {}</code> Individua l'elemento il cui attributo <code>id</code> ha valore <code>introduction</code>
SELETTORE DEL FIGLIO	Corrisponde a un elemento che è figlio diretto di un altro	<code>li>a {}</code> Individua ogni elemento <code><a></code> figlio di un elemento <code></code> (ma non gli altri elementi <code><a></code> della pagina)
SELETTORE DEL DISCENDENTE	Corrisponde a un elemento che è un discendente di un altro elemento specificato (non solo un figlio diretto di tale elemento)	<code>p a {}</code> Individua ogni elemento <code><a></code> ospitato da un elemento <code><p></code> , anche se fra di essi vi sono altri elementi
SELETTORE DEL FRATELLO ADIACENTE	Corrisponde a un elemento che è l'immediato fratello di un altro	<code>h1+p {}</code> Individua il primo elemento <code><p></code> dopo un elemento <code><h1></code> (ma non gli altri elementi <code><p></code>)
SELETTORE DEL FRATELLO GENERICO	Corrisponde a un elemento che è fratello di un altro, anche se non deve essere l'elemento immediatamente precedente	<code>h1~p {}</code> Se vi sono due elementi <code><p></code> che sono fratelli di un elemento <code><h1></code> , questa regola si applicherà a entrambi

COME SI APPLICANO A CASCATA LE REGOLE CSS

Se a uno stesso elemento si applicano due o più regole, è importante capire quale ha la precedenza.

ULTIMA REGOLA

Se i due selettori sono identici, avrà la precedenza l'ultimo. Qui potete vedere che il secondo selettore `i` ha la precedenza rispetto al primo.

SPECIFICITÀ

Se un selettore è più specifico degli altri, avrà la precedenza la regola più specifica rispetto a quella più generale. In questo esempio:

`h1` è più specifico di `*`
`p b` è più specifico di `p`
`p#intro` è più specifico di `p`

IMPORTANTE

Potete aggiungere `!important` dopo ogni valore di una proprietà per indicare che deve essere considerata più importante rispetto alle altre regole che si applicano allo stesso elemento.

Capire come funzionano a cascata le regole CSS significa scrivere fogli di stile più semplici, poiché diventa possibile creare regole generiche che si applicano alla maggior parte degli elementi e poi modificare le proprietà per i singoli elementi che devono avere un aspetto differente.

chapter-10/cascade.html

HTML

```
<h1>Potatoes</h1>
<p id="intro">There are <i>dozens</i> of different
  <b>potato</b> varieties.</p>
<p>They are usually described as early, second early
  and maincrop potatoes.</p>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```

OUTPUT

Potatoes

There are *dozens* of different **potato** varieties.

They are usually described as early, second early and maincrop potatoes.

EREDITARIETÀ

HTML

chapter-10/inheritance.html

```
<div class="page">
  <h1>Potatoes</h1>
  <p>There are dozens of different potato
    varieties.</p>
  <p>They are usually described as early, second
    early and maincrop potatoes.</p>
</div>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  padding: 10px;}
.page {
  border: 1px solid #665544;
  background-color: #efefef;
  padding: inherit;}
```

OUTPUT

Potatoes

There are dozens of different potato varieties.

They are usually described as early, second early and maincrop potatoes.

Se specificate le proprietà `font-family` o `color` dell'elemento `<body>`, queste verranno applicate alla maggior parte degli elementi che da esso dipendono. Questo è dovuto al fatto che il valore della proprietà `font-family` viene **ereditato** dagli elementi dipendenti. Questo evita di dover applicare le stesse proprietà a tutti gli elementi e il risultato è che i fogli di stile sono più semplici.

Potete confrontare questo con le proprietà `background-color` o `border`. Queste **non vengono ereditate** dagli elementi dipendenti. Se queste venissero ereditate da tutti gli elementi dipendenti, la pagina risulterebbe davvero incomprensibile.

Potete forzare l'ereditarietà delle proprietà dagli elementi genitori utilizzando `inherit` per il valore delle proprietà. In quest'esempio, l'elemento `<div>` con una classe chiamata `class` eredita le dimensioni di `page` dalla regola CSS che si applica all'elemento `<body>`.

PERCHÉ UTILIZZARE FOGLI DI STILE ESTERNI?

Nella realizzazione di un sito web vi sono vari vantaggi nel collocare le regole CSS in un foglio di stile esterno.

Tutte le pagine web possono condividere lo stesso foglio di stile. Questo può essere ottenuto facilmente utilizzando l'elemento `<link>` in ciascuna pagina HTML del sito e rimandando allo stesso documento CSS. Questo significa che il codice non dovrà essere ripetuto all'interno di ogni pagina, il che produce meno codice e pagine HTML più compatte.

Una volta che l'utente ha scaricato il foglio di stile CSS, la parte rimanente del sito verrà caricata più rapidamente. Per apportare una modifica all'aspetto del sito, dovrete intervenire solo su un file CSS e verranno automaticamente aggiornate tutte le pagine. Per esempio, potete modificare lo stile di ogni elemento `<h1>` intervenendo sul foglio di stile

CSS, invece di dover cambiare le regole CSS di ogni singola pagina. Il codice HTML risulterà anche più facile da leggere e da modificare, poiché non conterrà una grande quantità di regole CSS. Generalmente è considerata una buona abitudine far sì che il contenuto del sito sia separato dalle regole che ne determinano l'aspetto.

Talvolta potreste considerare l'idea di collocare le regole CSS nella stessa pagina del codice HTML.

Se dovete creare un'unica pagina, potreste decidere di inserire le regole nello stesso file, per tenere tutto in un unico luogo. Tuttavia, molti autori ritengono che sia molto meglio mantenere il codice di formattazione CSS su un file distinto.

Se avete una pagina che richiede alcune regole aggiuntive (che non vengono utilizzate dal resto del sito), potreste considerare la possibilità di inserire il codice CSS all'interno della pagina. Tuttavia, ancora una volta molti autori ritengono che sia molto meglio mantenere tutte le regole di formattazione CSS su un file distinto.

La maggior parte degli esempi di questo libro colloca le regole CSS nella sezione `<head>` del documento (utilizzando l'elemento `<style>`), invece di utilizzare un documento distinto. Questo semplicemente per evitare di costringere ad aprire due file per vedere il funzionamento degli esempi CSS.

VERSIONI DIFFERENTI DI CSS E PECULIARITÀ DEI BROWSER

CSS1 è stato rilasciato nel 1996 e CSS2 è seguito due anni dopo. Il lavoro su CSS3 è durato anni ed è ancora in corso, ma la maggior parte dei browser ha già iniziato a implementarlo.

Così come esistono più versioni di HTML, vi sono anche state più versioni di CSS. I browser non hanno implementato tutte

le funzionalità di CSS insieme e quindi alcuni vecchi browser non supportano tutte le proprietà CSS.

Ne parliamo per chiarire che vi sono situazioni in cui le proprietà CSS potrebbero non comportarsi nel modo previsto.

Ogni esperto utilizzatore di CSS vi dirà che alcuni browser visualizzano determinate proprietà CSS in modi imprevisti. Ma individuare e risolvere questi bug è piuttosto facile, sapendo come comportarsi.

Prima di lanciare un nuovo sito, è importante collaudarlo su più browser, poiché vi potrebbero essere leggere differenze nel modo in cui i browser visualizzano le pagine.

Utilizzando questi strumenti, potete pensare di controllare il sito su più sistemi operativi (PC, Mac e Linux) e con le vecchie versioni dei principali browser oltre che con le versioni più recenti.

Alcuni di questi bug verranno trattati all'interno di queste pagine, ma vi sono altri più subdoli che si verificano solo in alcune rare situazioni o su vecchi browser che ormai non vengono quasi più utilizzati.

Non avete bisogno di molti computer per collaudare il sito, in quanto vi sono strumenti online che mostrano l'aspetto di una pagina all'interno di più browser:

BrowserCam.com
BrowserLab.Adobe.com
BrowserShots.org
CrossBrowserTesting.com

Osservando l'aspetto del sito su più browser, potreste scoprire che alcuni elementi della pagina non hanno l'aspetto previsto.

Quando una proprietà CSS non viene visualizzata nel modo previsto, in genere si parla di **peculiarità del browser** o di **bug CSS**.

Se incontrate un bug CSS, utilizzate il vostro motore di ricerca preferito per cercare di trovare una soluzione, oppure provate a controllare questi siti:

PositionIsEverything.net
QuirksMode.org

RIEPILOGO

INTRODUZIONE A CSS

- ▶ CSS tratta ciascun elemento HTML come se apparisse all'interno di un box e utilizza delle regole per indicare l'aspetto di tale elemento.
- ▶ Le regole sono costituite da selettori (che specificano gli elementi cui si applica la regola) e dichiarazioni (che indicano l'aspetto di tali elementi).
- ▶ Tipi di selettori differenti consentono di rivolgere le regole su elementi differenti.
- ▶ Le dichiarazioni sono costituite da due parti: le proprietà dell'elemento da modificare e i valori di queste proprietà. Per esempio, la proprietà `font-family` imposta il tipo di carattere e il valore `arial` specifica Arial come tipo di carattere preferito.
- ▶ Le regole CSS normalmente sono contenute in documenti distinti, ma possono essere presenti anche all'interno di una pagina HTML.