

Corso di Web Programming

5. Cascading Style Sheets (CSS)

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
<http://www.di.unipi.it/~milazzo>
milazzo@di.unipi.it

Corso di Laurea in Informatica Applicata
A.A. 2010/2011

Sommario

1 Introduzione ai CSS

2 CSS e HTML

- Usare CSS in HTML
- I tag `<div>` e ``

3 La sintassi di CSS

- Definizione
- Alcune proprietà di base

4 Il modello visuale di CSS

- Il box model
- Il posizionamento degli elementi

Introduzione ai CSS (1)

- HTML è nato con l'idea di descrivere i contenuti in maniera separata rispetto agli aspetti di presentazione
- Le prime versioni dei browser permettevano agli **utenti** di definire l'aspetto dei vari elementi dei documenti HTML visualizzati (dimensione dei font, colori, ecc...)
- Successivamente la diffusione e l'evoluzione dei siti web ha reso sempre più importante la **gestione centralizzata** degli aspetti di presentazione
- HTML si è così arricchito di tag e attributi tipografici che sono poi finiti anche nella definizione degli standard del linguaggio
- La possibilità di visualizzare documenti HTML in dispositivi di natura diversa e l'interazione con le applicazioni hanno poi stimolato una separazione tra gli aspetti di contenuto e quelli di presentazione, favorendo la nascita di **linguaggi di stile** specializzati

Introduzione ai CSS (2)

- Cascading Style Sheets (CSS) è il linguaggio di stile che, tra le numerose proposte, si è imposto come “compagno” standard di HTML
- La particolarità di questo linguaggio è che prevede (e incoraggia) la presenza di fogli di stile multipli, che agiscono uno dopo l'altro **in cascata**
- Questo permette di avere controllo sia da parte dell'autore che del lettore di un documento HTML (ogniuno di quali può applicare uno o più fogli di stile al documento)
- Un'altra caratteristica vincente di CSS è l'indipendenza dall'insieme di elementi e attributi HTML
 - ▶ si può usare con qualunque versione di HTML
 - ▶ si può usare anche con linguaggi di markup simil-HTML (tipicamente XML)

Introduzione ai CSS (3)

Esistono diversi **livelli** di specifica di CSS:

- CSS level 1 (1991): è un linguaggio di formattazione visiva. Permette di specificare caratteristiche tipografiche e di presentazione di ogni elemento di un documento HTML
- CSS level 2 (1998): attualmente in versione 2.1, introduce il supporto ai media multipli (es. aural) e un linguaggio di layout sofisticato e complesso
- CSS level 3 è ancora in fase di sviluppo e migliora/estende la gestione di layout sofisticati, colori, sfondi, bordi, animazioni, trasformazioni, etc...

Il supporto dei browser a CSS è complesso e difficile da realizzare. Attualmente, comunque, le versioni più recenti dei principali browser supportano abbastanza fedelmente la versione 2.1

Usare CSS in HTML (1)

- CSS fornisce informazioni sull'aspetto di elementi di un documento HTML descritti tramite i tag
- Un foglio di stile CSS è costituito da regole simili alla seguente:

```
p { font-family: Arial; font-size: 12pt; }
```

che stabilisce che tutti per visualizzare il contenuto di tutti i tag <p> nel documento deve essere usato font Arial di dimensione 12 punti

- HTML 4 prevede che tutti i tag di un documento possano avere attributi `id` e `class`
 - ▶ `id` deve essere associato ad un identificativo univoco di un'occorrenza di un tag nell'intero documento (non può essere ripetuto)
 - ▶ `class` deve essere associato ad un identificativo di classe che può essere usato più volte nel documento
- CSS può usare i valori di `id` e `class` per specificare che una regola si applica ad uno specifico elemento del documento o a tutti gli elementi della stessa classe (vedremo...)

Usare CSS in HTML (2)

- Il concetto di cascata in CSS significa che regole successive che si possono applicare ad un elemento del documento HTML devono essere applicate in sequenza

- Ad esempio, applicando in sequenza:

- ▶ `p { font-family: Arial; font-size: 12pt; }`
- ▶ `p { color: red; font-size: 11pt; }`
- ▶ `p { margin-left: 15pt; color: green; }`

otterremo lo stesso risultato di

- ▶ `p { font-family: Arial; font-size: 11pt; margin-left: 15pt; color: green; }`

- Questa funzionalità è utile in particolare in due casi:
 - ▶ Quando si prevede di applicare più fogli di stile (insiemi di regole) allo stesso documento HTML
 - ▶ Quando si vogliono applicare regole generali (ad esempio a tutti i tag `<p>`) + regole specifiche per singoli elementi o classi (ad esempio a tutti i tag `<p class="menu">`)

Usare CSS in HTML (3)

- HTML prevede l'uso di fogli di stile CSS in quattro modi diversi:
 - ▶ Posizionato direttamente nel tag di riferimento tramite l'attributo `style`
 - ▶ All'interno di un tag `<style>`
 - ▶ Importato dal tag `<style>`
 - ▶ Indicato dal tag `<link>`
- Vediamo degli esempi....

Usare CSS in HTML (4)

CSS posizionato direttamente nel tag di riferimento tramite l'attributo `style`:

```
<html>
  <head>
    <title>Pagina di prova</title>
  </head>
  <body>
    <h1 style="color:blue;">Questo titolo deve essere blu</h1>
    <p>Contenuto della pagina di prova</p>
  </body>
</html>
```

Usare CSS in HTML (5)

CSS all'interno di un tag <style>:

```
<html>
  <head>
    <title>Pagina di prova</title>
    <style type="text/css">
      h1 { color: blue; }
    </style>
  </head>
  <body>
    <h1>Questo titolo deve essere blu</h1>
    <p>Contenuto della pagina di prova</p>
  </body>
</html>
```

Usare CSS in HTML (6)

CSS importato dal tag <style>:

```
<html>
  <head>
    <title>Pagina di prova</title>
    <style type="text/css">
      @import url(extfile.css)
    </style>
  </head>
  <body>
    <h1>Questo titolo deve essere blu</h1>
    <p>Contenuto della pagina di prova</p>
  </body>
</html>
```

dove extfile.css è il seguente

```
h1 { color: blue; }
```

Usare CSS in HTML (7)

CSS indicato dal tag <link>:

```
<html>
  <head>
    <title>Pagina di prova</title>
    <link type="text/css" rel="stylesheet" href="extfile.css">
  </head>
  <body>
    <h1>Questo titolo deve essere blu</h1>
    <p>Contenuto della pagina di prova</p>
  </body>
</html>
```

dove extfile.css è il seguente

```
h1 { color: blue; }
```

I tag <div> e (1)

- Con CSS si possono specificare regole di presentazione da associare a qualunque tag
- Talvolta, però, si vorrebbero definire regole da applicare
 - ▶ ad un'intera porzione del documento HTML (che potrebbero includere molti tag diversi),
 - ▶ o a un piccolo frammento di testo all'interno di un paragrafo (che potrebbe non essere delimitato da un tag)
- Per rendere possibile la definizione di queste regole in HTML sono stati definiti i tag <div> e

I tag <div> e (2)

- Il tag <div> viene usato per delimitare un'area (anche rilevante) del documento HTML a cui applicare delle regole stilistiche comuni;
- Esempi di utilizzo:
 - ▶ per suddividere il documento in sezioni, ogniuna delimitata da un bordo colorato
 - ▶ per isolare l'area che contiene l'indice del sito in modo da posizionarla alla sinistra della pagina
- Il tag viene usato per identificare un piccolo frammento (tipicamente di testo) all'interno di un paragrafo, una lista, una tabella, ecc...
- Esempi di utilizzo:
 - ▶ Per creare un'effetto evidenziatore su singole parole o porzioni di frase

I tag `<div>` e `` (3)

- Tecnicamente, la differenza tra `<div>` e `` consiste nel fatto che:
 - ▶ `<div>` appartiene alla categoria degli elementi di tipo `block` (come i paragrafi, le tabelle, ecc...) , ovvero elementi che, se usati in sequenza, vengono visualizzati uno sotto l'altro con un po' di spazio tra uno e l'altro
 - ▶ `` appartiene alla categoria degli elementi di tipo `inline` (come ``, ``, ecc...), ovvero elementi che, se usati in sequenza, vengono visualizzati uno accanto all'altro
- Gli attributi `id` e `class` potranno poi essere usati per differenziare le varie occorrenze di `<div>` e `` nel documento e applicarvi regole di stile diverse

1 tag <div> e (4)

Facciamo un esempio di uso di <div> e :

```
<html>
  <head><title>Prova div e span</title></head>

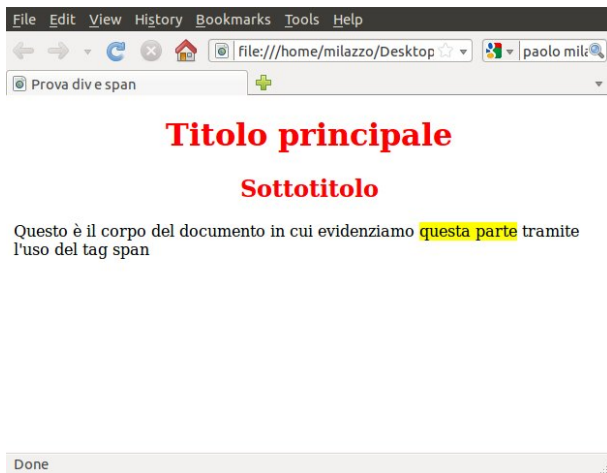
  <style type="text/css">
    div.titolo {
      text-align: center;
      color: red;
    }
    span#qp {
      background-color: yellow;
    }
  </style>

  <body>
    <div class="titolo">
      <h1>Titolo principale</h1>
      <h2>Sottotitolo</h2>
    </div>

    <div class="corpo">
      <p>Questo &grave; il corpo del documento in cui evidenziamo
        <span id="qp">questa parte</span> tramite l'uso del tag
        span</p>
    </div>
  </body>
</html>
```


I tag <div> e (5)

Il risultato dell'esempio:



La sintassi di CSS (1)

- Un foglio di stile CSS è costituito da una sequenza di **regole**
- Ogni regola consiste di un **selettore** e di una o più **dichiarazioni** racchiuse tra parentesi graffe (e ogniuna terminata da “;”)
- Ogni dichiarazione consiste di una **proprietà** ed un **valore** (separati da “:”)

```
selettore { proprietà1:valore1; proprietà2:valore2; ... }
```

- Il selettore è solitamente il tag di cui la regola specifica l'aspetto
- Le proprietà sono caratteristiche di stile che possono essere associate agli elementi. Sono definite dalla specifica di CSS e sono in tutto 53 per CSS level 1 e ben 121 per CSS level 2
- I valori dipendono dalla proprietà a cui sono assegnati...

```
p { font-family:Arial; font-size:12pt; }
```

La sintassi di CSS (2)

```
selettore { proprietà1:valore1; proprietà2:valore2; ... }
```

Il selettore può avere una delle seguenti forme:

- **Selettore di tipo.** Un tag (senza parentesi <,>): body, p, table, div, span, ...
- **Selettore di prossimità.** Hanno tre forme: E F, E>F e E+F, dove E ed F sono tag. Significano che la regola si applica agli elementi F che siano discendenti (ossia contenuti), figli diretti o immediatamente seguenti ad elementi E
- **Selettore di attributi.** E[attr] e E[attr=val]. Significano che la regola si applica agli elementi E in cui l'attributo attr è usato con qualunque valore o con valore specificato

...segue

La sintassi di CSS (3)

```
selettore { proprietà1:valore1; proprietà2:valore2; ... }
```

Il selettore può avere una delle seguenti forme:

- **Selettore di classe.** Può avere due forme `E.val` e `E#val`, che corrispondono a `E[class="val"]` e `E[id="val"]`. Significano che la regola si applica a tutti gli elementi `E` in cui gli attributi `class` e `id`, rispettivamente, sono assegnati a `val`,
Si può **omettere** `E`, e in questo caso la regola si applica a tutti gli elementi in cui `class` vale `val`, o all'elemento (unico) in cui `id` vale `val`, qualunque esso sia
- **Selettore di pseudo-classe.** Può avere una delle seguenti forme:
 - ▶ `E:first-child` si applica all'elemento che è primo figlio di `E` (per ogni occorrenza di `E` nel documento)
 - ▶ `E:link` e `E:visited` si applicano ai link non ancora o già visitato
 - ▶ `E:hover`, `E:active` e `E:focus` si applicano agli elementi **solo quando** vi passi sopra il mouse, vi si clicchi sopra o l'elemento è selezionato

La sintassi di CSS (4)

selettore { proprietà₁:valore₁; proprietà₂:valore₂; ... }

Il selettore può avere una delle seguenti forme:

- **Selettore di pseudo-elementi.** Può avere una delle seguenti forme:
 - ▶ `E:first-line` si applica alla prima riga di testo contenuta all'interno di E
 - ▶ `E:first-letter` si applica alla prima lettera contenuta all'interno di E
 - ▶ `E:before` e `E:after` si possono usare per aggiungere qualcosa (un'immagine, un testo,...) immediatamente prima o dopo l'elemento E
- **Selettore universale.** Ha la forma `*` e si applica a tutti gli elementi del documento HTML
- **Raggruppamento di selettori.** Indicando più selettori separati da virgole si ottiene che la regola viene applicata a tutti i selettori specificati

Un po' di esempi (1)

Per fare un po' di esempi consideriamo il seguente (body di) documento HTML

```
<body>
<h1>Ricetta: la <span class="piatto">pasta al pomodoro</span></h1>
<p>Vediamo ora come preparare un'ottima <span class="piatto">pasta
al pomodoro.</span> Innanzitutto, mettere a bollire in una
pentola capiente abbondante acqua.</p>
<p>Tagliare a fettine sottili la <span class="ingr">cipolla</span>
e soffriggerla con l'<span class="ingr">olio</span> in una
padella fino a quando non sar&agrave; ben dorata. In seguito
aggiungere i <span class="ingr">pomodori</span> e il
<span class="ingr">basilico</span>.</p>
<p>Quando bolle l'acqua, salarla e versare gli <span class="ingr">
spaghetti</span>. Una volta che gli spaghetti sono cotti,
scolarli e aggiungervi il sugo di pomodoro.</p>
</body>
```

Un po' di esempi (2)

Vediamo come il browser visualizza il documento HTML:

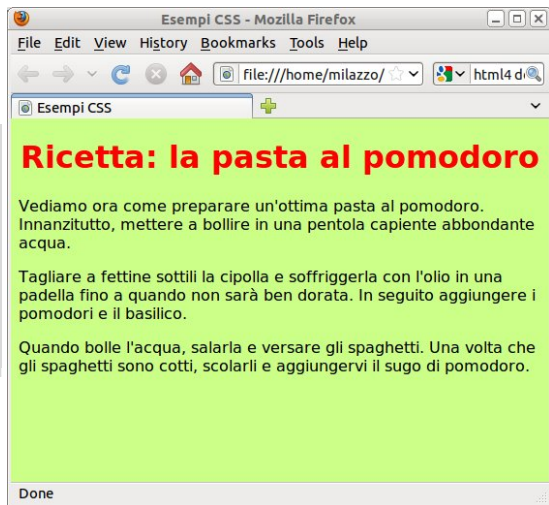


Un po' di esempi (3)

Primo foglio di stile:

```
body
{
  background-color: #CCFF88;
  font-family: sans-serif;
}

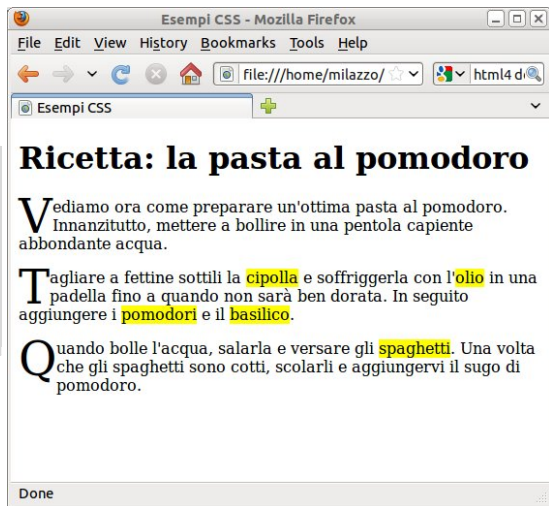
h1
{
  color: red;
  text-align: center;
}
```



Un po' di esempi (4)

Secondo foglio di stile:

```
span.ingr
{
  background-color: yellow;
}
p:first-letter
{
  font-size: 300%;
  float: left;
}
```



The screenshot shows a Mozilla Firefox browser window titled "Esempi CSS - Mozilla Firefox". The address bar shows the file path "file:///home/milazzo/html4 d...". The page content is a recipe titled "Ricetta: la pasta al pomodoro". The text is styled with a yellow background for the ingredients list and a large, bold, serif font for the first letter of each paragraph. The browser's status bar at the bottom shows "Done".

Ricetta: la pasta al pomodoro

Vediamo ora come preparare un'ottima pasta al pomodoro. Innanzitutto, mettere a bollire in una pentola capiente abbondante acqua.

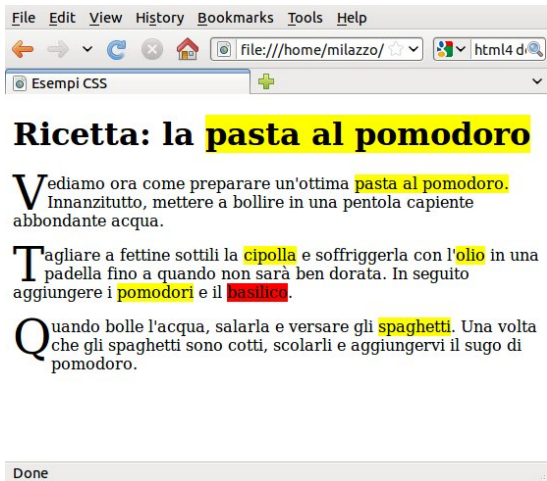
Tagliare a fettine sottili la **cipolla** e soffriggerla con l'**olio** in una padella fino a quando non sarà ben dorata. In seguito aggiungere i **pomodori** e il **basilico**.

Quando bolle l'acqua, salarla e versare gli **spaghetti**. Una volta che gli spaghetti sono cotti, scolarli e aggiungervi il sugo di pomodoro.

Un po' di esempi (5)

Terzo foglio di stile:

```
span.ingr, span.piatto
{
  background-color: yellow;
}
span:hover
{
  background-color: red;
}
p:first-letter
{
  font-size: 300%;
  float: left;
}
```



The screenshot shows a web browser window with the address bar displaying "file:///home/milazzo/" and a search engine icon. The browser title is "Esempi CSS". The main content of the page is a recipe titled "Ricetta: la pasta al pomodoro". The title is highlighted in yellow. The first paragraph starts with a large initial letter 'V' and contains the text "Vediamo ora come preparare un'ottima pasta al pomodoro." The second paragraph starts with a large initial letter 'T' and contains the text "Tagliare a fettine sottili la cipolla e soffriggerla con l'olio in una padella fino a quando non sarà ben dorata. In seguito aggiungere i pomodori e il basilico." The third paragraph starts with a large initial letter 'Q' and contains the text "Quando bolle l'acqua, salarla e versare gli spaghetti. Una volta che gli spaghetti sono cotti, scolarli e aggiungervi il sugo di pomodoro." The browser's status bar at the bottom shows "Done".

Alcune proprietà di base

Vediamo ora alcune (buona parte) delle proprietà che si possono usare nelle regole CSS.

L'elenco completo delle proprietà si trova nella specifica di CSS definita dal consorzio W3C o in numerosi siti e tutorial sul web.

Proprietà di background (1)

Le seguenti proprietà riguardano l'aspetto dello sfondo del documento:

- `background-color`: specifica il colore dello sfondo
- `background-image`: specifica l'immagine da usare come sfondo (di default ripetuta in orizzontale e verticale fino a riempire la finestra)
- `background-repeat`: specifica se l'immagine di sfondo deve essere ripetuta solo in orizzontale, solo in verticale o non deve essere ripetuta;
- `background-attachment`: specifica se l'immagine di sfondo deve avere posizione fissa o essere "scrollata" assieme alla pagina
- `background-position`: specifica la posizione in cui l'immagine di sfondo deve essere visualizzata

Proprietà di background (2)

Possibili valori delle proprietà di background:

<code>background-color</code>	<code>nome di colore (red,blue,...)</code> <code>valore rgb (es. rgb(255,0,0))</code> <code>valore rgb esadecimale (es. #FF0000)</code> <code>transparent</code>
<code>background-image</code>	<code>url('...url immagine...')</code> <code>none</code>
<code>background-repeat</code>	<code>repeat</code> <code>repeat-x</code> <code>repeat-y</code> <code>no-repeat</code>
<code>background-attachment</code>	<code>scroll</code> <code>fixed</code>
<code>background-position</code>	<code>left top, left center, left bottom</code> <code>right top, right center, right bottom</code> <code>center top, center center, center bottom</code> <code>x% y%</code> <code>xpos ypos</code>

Proprietà di background (3)

Le proprietà di background possono essere espresse in una forma compatta come nel seguente esempio:

```
background: red url('picture.gif') no-repeat right top;
```

che corrisponde a

```
background-color: red;  
background-image: url('picture.gif');  
background-repeat: no-repeat;  
background-position: right top;
```

Si noti che nella forma compatta:

- L'ordine dei valori è fissato;
- Non è necessario indicare tutti i valori: i valori mancanti saranno sostituiti con quelli di default.

Proprietà di testo (1)

Le seguenti proprietà riguardano l'aspetto del testo contenuto in un tag del documento:

- `color`: specifica il colore del testo
- `text-align`: specifica l'allineamento del testo (sinistra, destra, centrato e giustificato)
- `text-decoration`: specifica testo decorato (sottolineato, barrato, lampeggiante, ecc...)
- `text-transform`: specifica se il testo deve essere visualizzato in maiuscolo o minuscolo
- `text-indent`: specifica l'indentazione della prima linea di testo
- Altre: `line-height`, `vertical-align`, `word-spacing`, `direction`

Proprietà di testo (2)

Possibili valori delle proprietà di testo:

<code>color</code>	nome di colore (red,blue,...) valore rgb (es. <code>rgb(255,0,0)</code>) valore rgb esadecimale (es. <code>#FF0000</code>)
<code>text-align</code>	left, right, center, justify
<code>text-decoration</code>	none (usato spesso per togliere sottolineatura ai link) underline, overline line-through, blink
<code>text-transform</code>	none, capitalize uppercase, lowercase
<code>text-indent</code>	lunghezza (es. 50px) x%

Proprietà relative ai font (1)

- I font da usare per il testo contenuto in un dato elemento del documento HTML possono essere descritti in modo generico usando un descrittore di classe generico (`serif`, `sans-serif` o `monospace`)
- Oppure possono essere descritti indicando esattamente il nome del font da usare (es. `Times New Roman`)
- Il font da usare si specifica tramite la proprietà `font-family`
- La proprietà `font-family` si aspetta una lista di descrittori di font (separati da virgole).
 - ▶ Il browser scandirà la lista fino a che non trova un font conosciuto;
 - ▶ Per questo spesso la lista consiste di tre elementi: un nome di font specifico, un nome di font alternativo e simile al primo, una classe di font generica
- **NOTA:** Se il nome del font contiene spazi, bisogna racchiuderlo tra virgolette!

Proprietà relative ai font (2)

Esempi di specifiche di font:

- `font-family: "Times New Roman", Times, serif;`
- `font-family: Arial, Helvetica, sans-serif;`
- `font-family: Impact, Charcoal, sans-serif;`
- `font-family: Verdana, Geneva, sans-serif;`
- `font-family: "Courier New", Courier, monospace;`
- `font-family: "Lucida console", Monaco, monospace;`

Proprietà relative ai font (3)

La dimensione del font può essere specificata in modo assoluto o relativo:

- Una dimensione assoluta può essere espressa in pixel: es. 12px
- Una dimensione relativa può essere espressa in vari modi:
 - ▶ In percentuale, rispetto alla dimensione del del contesto (elemento contenitore): es. 80%
 - ▶ In maniera relativa rispetto alla dimensione del font impostata dal browser, usando l'unità di misura em (di default 1em = 16px)
 - ▶ In maniera simbolica attraverso i valori `small`, `medium`, `large`, `smaller`, `larger`,.....
- I font con dimensione assoluta a volte (specie con vecchie versioni di IE) non vengono ridimensionati correttamente:
 - ▶ Una soluzione valida è impostare il la dimensione per il tag `body` usando un valore percentuale, e le altre dimensioni usando em

Proprietà relative ai font (4)

Le seguenti proprietà riguardano i font usati nel documento:

- `font-family`: specifica (un elenco) di font da usare per un elemento
- `font-style`: specifica se il testo è normale o corsivo
- `font-size`: specifica la dimensione del font (assoluta o relativa)
- `font-weight`: specifica “il peso” (lo spessore) del font con un valore numerico (da 100 a 700) o con parole chiave (e.g. `bold`)
- `font-variant`: specifica se il testo deve essere visualizzato in minuscolo

Proprietà relative ai font (5)

Possibili valori delle proprietà di background:

font-family	lista di descrittori di font
font-style	normal italic oblique (poco usato)
font-size	xx-small, x-small, small medium, large, x-large, xx-large smaller, larger dimensione x%
font-weight	normal, bold lighter, bolder 100, 200, ..., 900
font-variant	normal small-caps

Proprietà relative ai font (6)

Le proprietà relative ai font possono essere espresse in una forma compatta come nel seguente esempio:

```
font: italic normal bold 15px Arial, sans-serif;
```

che corrisponde a

```
font-style: italic;  
font-variant: normal;  
font-weight: bold;  
font-size: 15px;  
font-family: Arial, sans-serif;
```

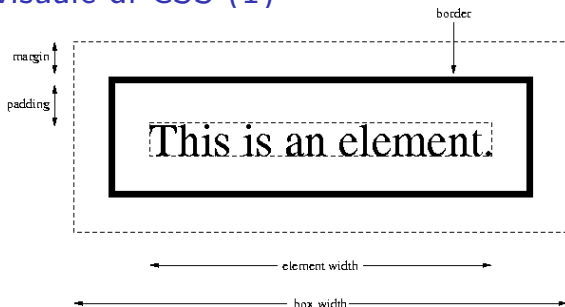
Si noti che nella forma compatta:

- L'ordine dei valori è fissato;
- I valori corrispondenti a `font-size` e `font-family` sono obbligatori
- Gli altri valori non sono obbligatori: i valori mancanti verranno sostituiti con quelli i default

Proprietà relative alle liste

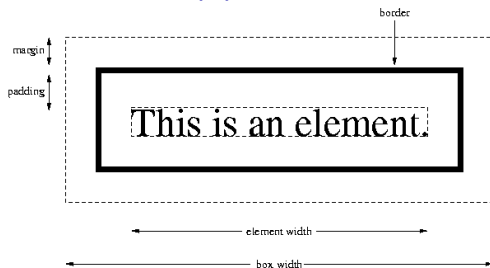
- Esistono inoltre proprietà specifiche per le liste (numerata e non numerata) che consentono di impostare il simbolo (o l'immagine) da visualizzare a sinistra di ogni elemento di una lista non numerata, o di impostare il formato della numerazione degli elementi di una lista numerata.
- Queste proprietà (che non vediamo in dettaglio) sono: `list-style`, `list-style-image`, `list-style-position` e `list-style-type`
- Ad esempio: `list-style:none`; toglie il punto nero dagli elementi di una lista non numerata

Il modello visuale di CSS (1)



- Il modello visuale di CSS (ovvero l'insieme di principi riguardo la disposizione degli elementi) è detto **Box Model**
- Ogni elemento (tag) di un documento HTML è visto come una scatola che consiste di
 - ▶ Uno spazio di margine (`margin`)
 - ▶ Un bordo (`border`) che può essere trasparente, una linea continua o tratteggiata, ecc...
 - ▶ Uno spazio di separazione del contenuto dal bordo (`padding`)
 - ▶ Il contenuto vero e proprio del tag (`content`)

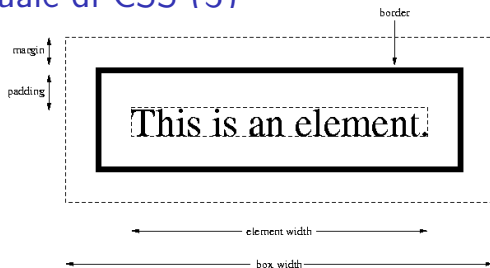
Il modello visuale di CSS (2)



- Le dimensioni (larghezza e altezza) **del contenuto** possono essere specificate tramite le proprietà `width` e `height`
- Le dimensioni possono essere specificate usando valori precisi o in percentuale rispetto al blocco contenitore

<code>width</code>	dimensione (es. 100px) x%
<code>height</code>	dimensione (es. 100px) x%

Il modello visuale di CSS (3)



L'aspetto del bordo può essere specificato tramite le seguenti proprietà:

- `border-style`: specifica lo stile del bordo (linea continua, tratteggiata, doppia, 3D, ...)
- `border-width`: specifica lo spessore del bordo in pixel o usando valori predefiniti (`thin`, `medium` e `thick`)
- `border-color`: specifica il colore del bordo
- Inoltre, esistono versioni specifiche delle tre proprietà precedenti che riguardano un solo lato del box (es. `border-left-style`, `border-right-width`, `border-top-color`, `border-bottom-style`, ...)

Il modello visuale di CSS (4)

Possibili valori delle proprietà del bordo:

<code>border-style</code>	<code>none, dotted, dashed</code> <code>solid, double, groove</code> <code>ridge, inset, outset</code>
<code>border-width</code>	<code>thin, medium, thick</code> dimensione (es. <code>5px</code>)
<code>border-color</code>	nome di colore (<code>red, blue, ...</code>) valore rgb (es. <code>rgb(255,0,0)</code>) valore rgb esadecimale (es. <code>#FF0000</code>)

Le proprietà specifiche per i singoli lati prevedono valori analoghi alle corrispondenti proprietà per tutti e quattro i lati (in tabella)

Il modello visuale di CSS (5)

Le proprietà relative ai bordi possono essere espresse in una forma compatta come nel seguente esempio:

```
border: 5px solid red;
```

che corrisponde a

```
border-width: 5px;  
border-style: solid;  
border-color: red;
```

Si noti che nella forma compatta:

- L'ordine dei valori è fissato;
- Il valore corrispondente a `border-style` è obbligatorio
- Gli altri valori non sono obbligatori: i valori mancanti verranno sostituiti con quelli i default

Il modello visuale di CSS (6)

Un'altra forma compatta che si può usare quando lati diversi devono avere proprietà diverse è la seguente:

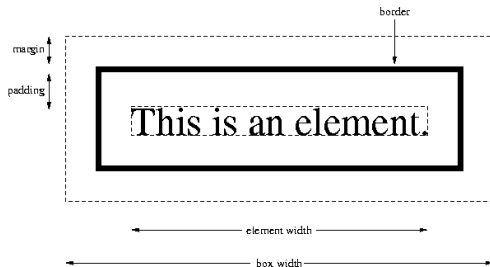
```
border-style: solid none dashed dotted
```

che corrisponde a:

```
border-top-style: solid  
border-right-style: none  
border-bottom-style: dashed  
border-left-style: dotted
```

- L'ordine dei lati è fissato (senso orario)
- Se si indicano meno di 4 valori, ai lati mancanti vengono applicati gli stili dei corrispondenti lati simmetrici
- Forme compatte analoghe sono definite per `border-width` e `border-color`

Il modello visuale di CSS (7)



Le dimensioni dei margin e padding si possono impostare tramite le proprietà:

- `margin-top`, `margin-right`, `margin-bottom` e `margin-left` per quanto riguarda il margin
- `padding-top`, `padding-right`, `padding-bottom` e `padding-left` per quanto riguarda il padding

Il modello visuale di CSS (8)

Sia per il margin che per il padding esistono forme compatte:

```
margin: 10px 20px 0px 15px
```

```
padding: 10px 20px 0px 15px
```

che corrispondono a:

```
margin-top: 10px  
margin-right: 20px  
margin-bottom: 0px  
margin-left: 15px
```

```
padding-top: 10px  
padding-right: 20px  
padding-bottom: 0px  
padding-left: 15px
```

- L'ordine dei lati è fissato (senso orario)
- Se si indicano meno di 4 valori, ai lati mancanti vengono applicati i valori dei corrispondenti lati simmetrici

Il modello visuale di CSS (9)

- Alle proprietà relative al margin può essere assegnato il valore **auto**
- In questo caso il browser è libero di decidere
- Normalmente, se sia `margin-left` che `margin-right` sono impostati ad `auto` i browser associano lo stesso valore ad entrambe le proprietà
 - ▶ Questo è un modo comunemente usato per **centrare** un elemento (quando la proprietà `width` dell'elemento è impostata a un valore preciso)
 - ▶ In Internet Explorer questo metodo funziona solo se il documento HTML specifica il `doctype`

Proprietà relative alle tabelle

- Le proprietà relative al box model (`border`, `margin`, `padding`, ecc...) si usano anche per formattare le tabelle
- Esistono anche delle proprietà specifiche per impostare alcuni dettagli di visualizzazione che non vedremo in dettaglio
- Ad esempio: `border-collapse: collapse`; consente di sovrapporre i bordi delle celle adiacenti in una tabella

Un esempio di uso del box model (1)

Riprendiamo l'esempio:

```
<body>
  <h1>Ricetta: la <span class="piatto">pasta al pomodoro</span></h1>
  <p>Vediamo ora come preparare un'ottima <span class="piatto">pasta
  al pomodoro.</span> Innanzitutto, mettere a bollire in una
  pentola capiente abbondante acqua.</p>
  <p>Tagliare a fettine sottili la <span class="ingr">cipolla</span>
  e soffriggerla con l'<span class="ingr">olio</span> in una
  padella fino a quando non sar&agrave; ben dorata. In seguito
  aggiungere i <span class="ingr">pomodori</span> e il
  <span class="ingr">basilico</span>.</p>
  <p>Quando bolle l'acqua, salarla e versare gli <span class="ingr">
  spaghetti</span>. Una volta che gli spaghetti sono cotti,
  scolarli e aggiungervi il sugo di pomodoro.</p>
</body>
```

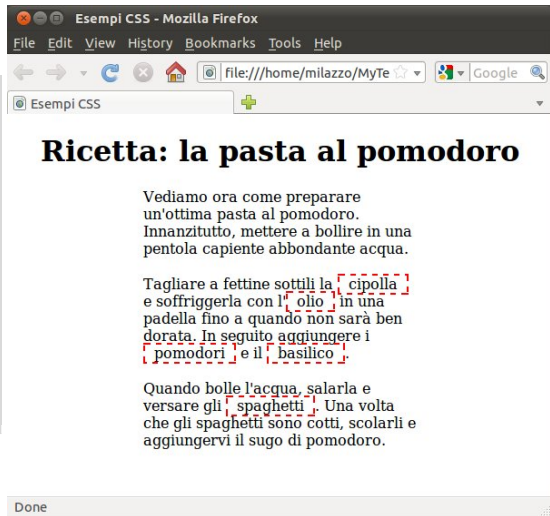
Un esempio di uso del box model (2)

Il foglio di stile:

```
h1
{
  text-align: center;
}

p
{
  width: 300px;
  margin: 20px auto;
}

span.ingr
{
  border: 2px dashed red;
  padding: 0px 10px;
}
```



The screenshot shows a Mozilla Firefox browser window titled "Esempi CSS - Mozilla Firefox". The address bar shows the file path "file:///home/milazzo/MyTe". The page content is a recipe titled "Ricetta: la pasta al pomodoro". The text is styled according to the CSS code shown in the previous block. The title is centered. The paragraphs are centered and have a width of 300px. The word "ingrediente" is highlighted with a red dashed border and padding.

Ricetta: la pasta al pomodoro

Vediamo ora come preparare un'ottima pasta al pomodoro. Innanzitutto, mettere a bollire in una pentola capiente abbondante acqua.

Tagliare a fettine sottili la **cipolla** e soffriggerla con l'**olio** in una padella fino a quando non sarà ben dorata. In seguito aggiungere i **pomodori** e il **basilico**.

Quando bolle l'**acqua**, salarla e versare gli **spaghetti**. Una volta che gli spaghetti sono cotti, scolarli e aggiungervi il sugo di pomodoro.

Done

Il posizionamento degli elementi (1)

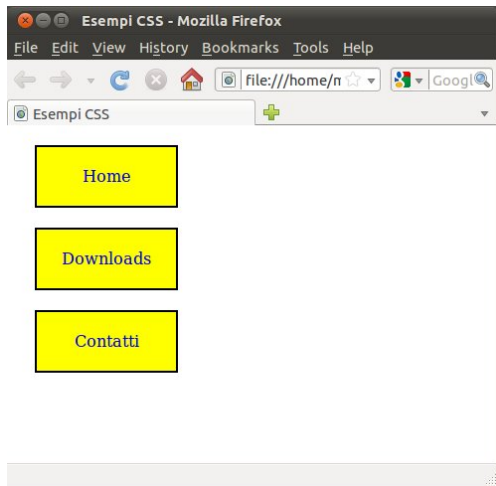
- Il box model consente di descrivere la dimensione, posizione e spaziatura di ogni elemento di un documento HTML
- Il browser legge in sequenza i vari tag del documento HTML e visualizza i contenuti disponendo le corrispondenti “scatole” (box) in uno dei seguenti modi (a seconda del tag usato):
 - ▶ Gli elementi di tipo **block** (es. `div`, `p`, `li`, ecc...) vengono disposti l'uno sotto l'altro in successione verticale
 - ▶ Gli elementi di tipo **inline** (es. `span`, `strong`, ecc...) vengono disposti l'uno accanto all'altro in successione orizzontale
- Inoltre, usando opportune proprietà CSS si può
 - ▶ Visualizzare un elemento di tipo `block` come se fosse `inline` e viceversa
 - ▶ Impostare una visualizzazione **float**, in cui la scatola viene spostata all'estrema sinistra o destra, lasciando che le altre scatole vi girino intorno
 - ▶ Impostare una **posizione assoluta** per una scatola nella finestra del browser (eventualmente nascondendo ciò che già occupava la posizione specificata)

Il posizionamento degli elementi (2)

- La proprietà `display` consente di impostare la modalità di visualizzazione di un elemento
 - ▶ `block`, `inline`, `none` (elemento non visualizzato), e altri più specifici
- Esempio: menù disposto in orizzontale

```
a
{
  display: block;
  border: 2px solid black;
  background-color: yellow;
  width: 100px;
  margin: 20px;
  padding: 20px;
  text-align: center;
  text-decoration: none;
}
```

```
<a href="...">Home</a>
<a href="...">Downloads</a>
<a href="...">Contatti</a>
```

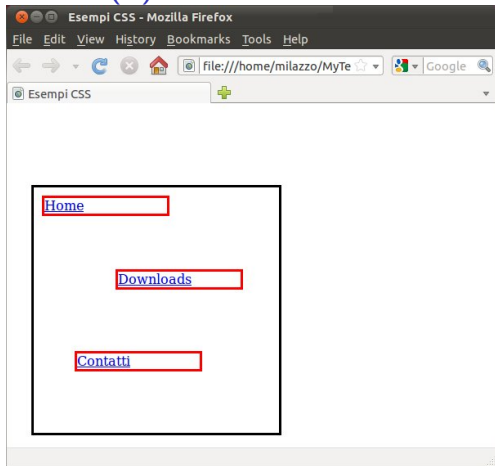


Il posizionamento degli elementi (3)

- La proprietà `position` consente di specificare il tipo di posizionamento dell'elemento all'interno della finestra del browser.
- Possibili valori:
 - ▶ `static` (valore di default) segue il flusso normale di visualizzazione degli elementi
 - ▶ `relative` consente di specificare la posizione in maniera relativa rispetto al punto in cui l'elemento sarebbe visualizzato di default
 - ▶ `fixed` consente di specificare una posizione fissata relativa alla finestra del browser (l'elemento non viene "scrollato via" (rimane fermo) quando si agisce sulla barra di scorrimento)
 - ▶ `absolute` consente di specificare una posizione con coordinate assolute all'interno dello spazio messo a disposizione dall'elemento che lo contiene
- Le posizioni (assolute e relative) sono espresse tramite le proprietà `left`, `top`, `right` e `bottom`

Il posizionamento degli elementi (4)

```
div#menu
{
  width: 300px;
  height: 300px;
  border: solid;
  position: fixed;
  top: 100px;
  left: 30px;
}
div#menu a
{
  border: solid red;
  width: 150px;
  position: absolute;
}
a#a1 {
  top: 10px;
  left: 10px;
}
a#a2 {
  top: 100px;
  left: 100px;
}
a#a3 {
  top: 200px;
  left: 50px;
}
```



```
<div id="menu">
  <a id="a1" href="...">Home</a>
  <a id="a2" href="...">Downloads</a>
  <a id="a3" href="...">Contatti</a>
</div>
```

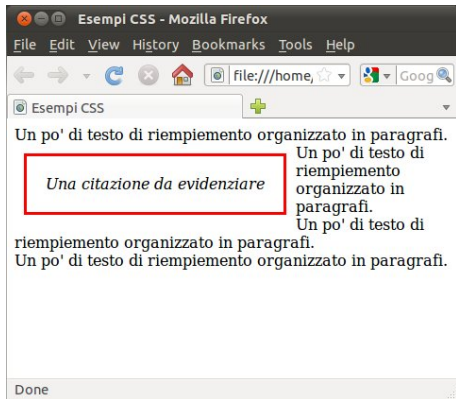
Il posizionamento degli elementi (5)

- Con CSS è possibile anche “accumulare” elementi sulla sinistra (o sulla destra) usando la proprietà `float`
- Impostando `float:left`; l’elemento viene visualizzato all’estrema sinistra, con gli altri elementi disposti attorno ad esso
- Se si setta la proprietà `float` per più elementi questi vengono “impilati”, sulla sinistra (o sulla destra), eventualmente disposti su più righe
- Gli elementi si ridispongono, se necessario, quando si ridimensiona la finestra del browser
- Per evitare che gli altri elementi del documento HTML si dispongano attorno all’elemento floating si può usare la proprietà `clear`

Il posizionamento degli elementi (5)

Un esempio di floating:

```
p
{
  margin: 0px;
}
p.citazione
{
  float: left;
  border: solid red;
  font-style: italic;
  margin: 10px;
  padding: 20px;
}
```



applicato a:

```
<p>Un po' di testo di riempimento organizzato in paragrafi</p>
<p class="citazione">Una citazione da evidenziare </p>
<p>Un po' di testo di riempimento organizzato in paragrafi</p>
<p>Un po' di testo di riempimento organizzato in paragrafi</p>
<p>Un po' di testo di riempimento organizzato in paragrafi</p>
```

Il posizionamento degli elementi (6)

Altri esempi di uso di floating:

- Il sito web di questo corso include un menù disposto alla sinistra del documento usando la proprietà `float`
- Il corso CSS su <http://www.w3schools.com> nella sezione “CSS floating” mostra alcuni esempi interessanti sull'uso di `float`

Altre cose da dire su CSS

Varie cose che sono rimaste da dire su CSS:

- E' possibile inserire commenti in CSS usando lo stile Java multilinea.
Ovvero: `/* ... */`
- **IMPORTANTE:** alcuni browser (e.g. IE) usano i CSS in modo corretto solo se il documento HTML specifica il doctype!!!
- Si possono associare più fogli di stile ad un documento HTML lasciando al lettore la possibilità di scegliere (con certi browser):
vedere
<http://www.w3.org/Style/Examples/007/alternatives.html>
- Un valore che può essere associato a qualunque proprietà è `inherit`, con l'effetto di far ereditare il valore di quella proprietà dall'elemento padre
- La versione 2 di CSS consente di definire regole diverse per diversi tipi di media (schermo, stampa, audio, ...) attraverso la keyword `media` (che non vediamo)
- Il corso CSS su <http://www.w3schools.com> offre alcuni esempi interessanti (navigation bar, image gallery,)