

CSS

Cascading Style Sheets

CSS aka fogli di stile

- ✓ Prima dei Cascade Style Sheets (CSS) o "Fogli di stile a cascata", in HTML le opportunità di definire l'aspetto delle pagine web erano limitate a specifici TAG (ad esempio il tag FONT) e si basavano sull'uso di tabelle per la definizione del layout della pagina.
- ✓ I CSS sono lo strumento definito dal W3C per definire l'aspetto delle pagine web. Sono essenzialmente un'insieme di regole per istruire il browser su come debba essere presentata la pagina web.
- ✓ Con i CSS per separare il contenuto (es la scritta "titolo 1"), dalla formattazione (es il colore rosso e il grassetto) il contenuto della pagina diventa:

```
<p class="formattaTitoli"> titolo 1 </p>
```

- ✓ e la colorazione del testo è affidata alla classe "formattaTitoli", descritta in un'altra parte del documento, o anche in un file separato. Per cambiare l'aspetto delle pagine è sufficiente editare la classe "formattaTitoli".

CSS - TAG Style

- ✓ Le regole dei CSS possono essere specificate direttamente nel codice (X)HTML tramite il tag STYLE posto nell'HEAD del documento (X)HTML. Se ne dovrebbe limitare l'uso a casi particolari.

```
<head>
  [...]
  <style type="text/css">
    <!-- body { font:80% Verdana,Helvetica, sans-serif; }
    p { font: 1.00em/1.20em verdana, helvetica, sans-serif }
    [...] -->
  </style>
  [...]
</head>
```

CSS sintassi

- ✓ La sintassi delle regole di applicazione dello stile segue lo schema:

```
Selettore{  
    proprietà: valore;  
    proprietà: valore;  
}
```

- ✓ Esempi:

```
titolo {  
    font-family: Arial;  
    font-size: 18pt;  
    color: #0000FF;  
    text-align: center;  
}
```

formatta l'elemento titolo

```
argomento titolo {  
    font-family: Arial  
}
```

formatta l'elemento
<titolo> che sta dentro
l'elemento <argomento>

CSS sintassi

- ✓ E' possibile scrivere tutto inline:

```
selettore { proprietà:valore; proprietà: valore }
```

- ✓ Esempi:

```
h1 { font-size: 25px; color: #660000; }  
h2 { font-size: 15px; color: #990000; }
```

questo codice definisce il colore e le dimensioni del testo per
gli elementi h1 e h2

il selettore font-size serve a specificare la dimensione dei
font, il selettore color serve a definire il colore del testo.

CSS proprietà e valori (1 di 2)

proprietà	valori	descrizione
background-color	red,green,#663366	indica il colore dello sfondo di un elemento
background-image	url	specifica l'immagine da visualizzare come sfondo di un elemento
border-style	none, dotted, double...	specifica lo stile del bordo da applicare ad un elemento
color	red,green,#CCCCCC	specifica il colore di un elemento
display	block,inline,none	specifica le modalità
font-family	times, helvetica, arial	indica il tipo di carattere da utilizzare
font-size	small, large, 10pt, 12pt	indica le dimensioni dei caratteri
font-weight	normal, bold, 100, 200	indica lo spessore dei caratteri
height, width	30px, 100px, 75%	indicano l'altezza e la larghezza di un elemento

CSS proprietà e valori (2 di 2)

proprietà	valori	descrizione
position	absolute, relative, static	indica il tipo di posizionamento da applicare
text-decoration	none, underline, blink	indica alcune decorazioni da applicare al testo
z-index	1,2,3	specifica l'ordine di posizionamento degli elementi nello spazio (sovrapposizione)

Proprietà e Valori

- ✓ Per la specifica dei valori delle proprietà verrà utilizzata la seguente sintassi (la stessa utilizzata dal W3C):
 - `<Abc>`: valore della proprietà
 - `A B C`: keyword richieste nell'ordine specificato
 - `A | B`: deve apparire A o B
 - `A || B`: deve apparire A o B oppure ambedue
 - `[Abc]`: le parentesi quadre sono utilizzate per raggruppare gli oggetti
 - `Abc*`: Abc appare zero o più volte
 - `Abc+`: Abc appare una o più volte
 - `Abc?`: Abc è opzionale
 - `Abc{A,B}`: Abc deve apparire almeno A volte e al massimo B volte

Proprietà: font-family

✓ Sintassi:

font-family: <nome font>* ||

<nome font>

- qualsiasi nome di font (per esempio: verdana, helvetica, georgia)

- serif
- sans-serif
- monospace
- cursive [sconsigliato poiché di difficile lettura]
- fantasy [sconsigliato poiché di difficile lettura]

✓ Esempio:

```
p { font-family: verdana, helvetica, "Trebuchet MS", sans-serif }
```

Il font adottato sarà il primo disponibile partendo da sinistra. E' bene specificare sempre un font generico, applicato qualora nessuno dei font elencati fosse disponibile. Font con nomi composti da due o più parole vanno fra virgolette: ad esempio "Trebuchet MS"

Proprietà: font-style

- ✓ Sintassi:

`font-style: normal | italic | oblique;`

- ✓ Esempio:

```
quote { font-style: italic }
```

Questa regola rende il testo italico (corsivo). Il valore `oblique` inclina in senso opposto. Il valore `normal` non introduce variazioni.

Proprietà: font-variant

- ✓ Sintassi:

font-variant: normal | small-caps

- ✓ Esempio:

```
.maiuscoletto { font-variant: small-caps }
```

Si noti che .maiuscoletto equivale alla definizione della classe "maiuscoletto" nel CSS. In HTML faremo riferimento ad essa con class = "maiuscoletto". Il valore small-caps crea l'effetto "Maiuscoletto", il valore normal non introduce variazioni.

Proprietà: font

✓ Sintassi:

font: <valore>

<valore>

- [<font-style> || <font-variant> || <font-weight>]? <font-size> [/ <line-height>]? <font-family>

✓ Esempio:

```
p.paragrafo-speciale {  
    font: bold 1.00em/1.50em verdana, helvetica, "Trebuchet MS", sans-serif;  
}
```

La proprietà font può essere utilizzata come un metodo veloce per definire le diverse proprietà font-family, font-style, ecc. Si noti la possibilità di specificare anche la proprietà <line-height> (una proprietà per specificare l'interlinea). Quando si utilizza la proprietà font è necessario specificare almeno i valori per le proprietà font-size e font-family.

Proprietà: color

- ✓ Sintassi:

color: <colore>

- ✓ Esempio:

```
a { color: #009 }  
em {color: red }
```

La proprietà color permette di specificare il colore del testo di un selettore

Proprietà: background

✓ Sintassi:

`<valore><colore> || <immagine> || <ripetizione> || <scrolling> || <posizione>`

`<colore>` valori proprietà color

`<immagine>` url dell'immagine

`<ripetizione>`: repeat | repeat-x | repeat-y | no-repeat

`<scrolling>`: scroll | fixed

`<posizione>`: [`<percentuale>` | `<lunghezza>`]{1,2} | [top | center | bottom] || [left | center | right]

✓ Esempio:

```
body { background:#FFFFFF url(/images/sfondo.gif) no-repeat fixed top right; }
```

Proprietà: word-spacing

✓ Sintassi:

word-spacing: <valore>

- <valore> : normal | <lunghezza>

✓ Esempio:

```
.parole-distanziate { word-spacing: 1.00em }  
.parole-ravvicinate { word-spacing: -0.20em }
```

Proprietà: letter-spacing

✓ Sintassi:

letter-spacing: <valore>

- <valore> : normal | <lunghezza>

✓ Esempio:

```
.lettere-distanziate { letter-spacing: 0.50em }  
.lettere-ravvicinate { letter-spacing: -0.15em }
```


Proprietà: text-indentation

✓ Sintassi:

```
text-indentation: <valore>  
    <valore> : <lunghezza> | <percentuale>
```

✓ Esempio:

```
p { text-indentation: 2em; }
```

em è la dimensione tipografica di una M maiuscola

Proprietà: text-decoration

✓ Sintassi:

text-decoration: <valore>

<valore> : none | [underline || overline || line-through || blink]

✓ Esempio:

```
#barra-di-navigazione A { text-decoration: none }  
/* link non sottolineato */
```

CSS - Selettori

- ✓ I selettori servono a specificare a quale elemento del documento la regola debba essere applicata. I principali selettori sono:
 - ✓ Tag HTML
 - ✓ Classi
 - ✓ Identificatori
 - ✓ Pseudo-classi

Tag HTML

- ✓ Ogni tag HTML è un selettore. Ad esempio:

```
p { text-indent: 2em; }  
a { font-weight: bold; }
```

- ✓ Il codice sopra riportato specifica al browser che i paragrafi devono avere il capoverso con un rientro di due caratteri e che i link devono essere in grassetto, migliorandone l'identificazione all'interno del testo.

Classi (1 di 3)

- ✓ Spesso si ha la necessità di associare ad uno stesso tag HTML diversi stili, per fare ciò è possibile ricorrere alle classi.

```
/* carattere piccolo */  
p.nota { font-size: small }  
/* area con sfondo giallo */  
p.evidenzia { background-color: yellow; }
```

Le due classi sono *nota* ed *evidenzia* e vengono specificate separando il nome del tag e il nome della classe con un punto.

- ✓ La classe viene specificata nel codice HTML attraverso l'attributo CLASS:

```
<p class="nota">Questo paragrafo è una nota</p>  
<p class="evidenzia">Questo paragrafo è evidenziato in giallo</p>
```

La sintassi utilizzata specifica che la classe è associata al TAG P.

Classi (2 di 3)

- ✓ In questo modo è possibile associare lo stesso nome a classi associate a TAG differenti:

- ✓ CSS

```
p.nota { font-size: small } /* carattere piccolo */  
span.nota { color: #999999; } /* testo in grigio */
```

- ✓ (X)HTML

```
<p class="nota">Questo paragrafo è una nota</p>  
<p>Testo normale (<span class="nota">nota nel  
testo</span>) testo normale ...</p>
```

Classi (3 di 3)

- ✓ Le classi possono anche essere associate a nessun TAG HTML, in questo modo la stessa classe può essere associata a più selettori:

- ✓ CSS

```
.nota { font-size: small } /* carattere piccolo */
```

- ✓ (X)HTML

```
<p class="nota">Questo paragrafo è una nota</p>
```

```
<div class="nota">Questa sezione è una nota</div>
```

- ✓ A un selettore è possibile associare una e una sola classe. Ad esempio non è valido il seguente codice:

```
p.classe1.classe2 { ... }
```

Pseudo-classi (1 di 2)

- ✓ Pseudo-classi e pseudo-elementi sono classi ed elementi speciali, automaticamente riconosciuti dai browser che supportano i CSS. Non è dunque necessario specificare nulla nel codice HTML.
- ✓ Le pseudo-classi creano distinzioni fra elementi, ad esempio i link visitati e i link attivi sono distinti attraverso due pseudo-classi.
- ✓ La sintassi è la seguente:
`selettore:pseudoClasse { proprietà: valore }`

Pseudo-classi (2 di 2)

- ✓ Definizione comune dei link (LOVE / HATE)

```
a:link { color: blue; }  
a:visited { color: purple; }  
a:hover { color: red; }  
a:active { color: darkred; }
```

- ✓ Il codice CSS indica al browser che i link devono essere blu, i link visitati devono essere viola, i link sui quali il puntatore del mouse si trova devono essere rossi (effetto roll-over), i link attivi devono essere rosso scuro.
- ✓ Nel codice XHTML non è necessario specificare nulla:

```
<a href="URL">questo link cambierà colore in base al suo  
stato</a>
```

Identificatori (1 di 2)

- ✓ Gli identificatori sono simili alle classi, ma servono a definire un'entità unica all'interno del documento. Se le classi, come la parola stessa indica, definiscono un insieme di elementi concettualmente omogenei (ad esempio le note di un testo possono essere accorpate nella classe note), l'identificatore deve definire un elemento che non si ripete all'interno del documento, ad esempio la nota per il copyright.
- ✓ Il nome del identificatore deve essere preceduto dal carattere '#'.
`#nota-copyright { font-size: xx-small }`
- ✓ L'identificatore viene specificato nel codice HTML attraverso l'attributo ID:
`#nota-copyright { font-size: xx-small }`

Identificatori (2 di 2)

- ✓ Nel codice non devono esistere due tag HTML associati allo stesso identificatore, non possono dunque esistere due tag HTML con lo stesso valore per l'attributo ID.
- ✓ Il nome del identificatore deve essere preceduto dal carattere '#'.

```
<p id="nota-copyright">Questo paragrafo è la nota per il  
copyright</p>
```

```
<p id="nota-copyright">E' errato inserire questo paragrafo</p>
```

Selettori in cascata (1 di 2)

- ✓ Ci sono numerosi altri modi più o meno elaborati per specificare dei selettori, alcuni dei quali non completamente supportati da browser largamente diffusi (ad esempio Internet Explorer 6 e inferiori).
- ✓ Un metodo ben supportato e molto utile è quello di utilizzare selettori in cascata:

```
selettore1 selettore2 { proprietà: valore }
```

Selettori in cascata (2 di 2)

- ✓ In questo caso la regola si applica al selettore2 se questo è contenuto nel selettore1. Si consideri il seguente codice CSS:

```
code i { color: red; }
```

Questo codice specifica che il tag I, se inserito nel TAG CODE, deve contenere del testo di colore rosso.

- ✓ Consideriamo il seguente codice HTML:

```
<p><code>testo normale <i>testo corsivo</i> testo  
normale</code></p>
```

```
<p>testo normale <i>testo corsivo</i> testo normale</p>
```

```
testo normale testo corsivo testo normale
```

```
testo normale testo corsivo testo normale
```

Raggruppamento

- ✓ Qualora più selettori dovessero rispettare le stesse regole, è possibile raggrupparli

```
h1, h2, h3, h4, h5, h6 { font-family: sans-serif }
```

- ✓ Il codice è equivalente al seguente:

```
h1 { font-family: sans-serif }  
h2 { font-family: sans-serif }  
h3 { font-family: sans-serif }  
h4 { font-family: sans-serif }  
h5 { font-family: sans-serif }  
h6 { font-family: sans-serif }
```

CSS2 Box Model (1 di 4)

- ✓ Il CSS2 Box Model descrive i contenitori rettangolari che vengono generati per gli elementi nella struttura del documento e disposti secondo la formattazione visuale definita.

CSS2 Box Model (2 di 4)

- ✓ Il perimetro di ciascuna delle quattro aree (content, padding, border e margin) si chiama "edge" :
 - ✓ Il content edge o inner edge circonda il contenuto degli elementi visualizzati
 - ✓ Il padding edge circonda il padding box. Se il padding box ha larghezza zero, il padding edge coincide con il content edge . Il padding edge di un box definisce i bordi del blocco-contenuto.
 - ✓ Il border edge circonda il bordo del box. Se il border box ha larghezza zero, il border edge coincide con il padding edge.
 - ✓ Il margin edge o outer edge circonda il margin box. Se il margin box ha larghezza zero, il margin edge coincide con il border edge.

CSS2 Box Model (3 di 4)

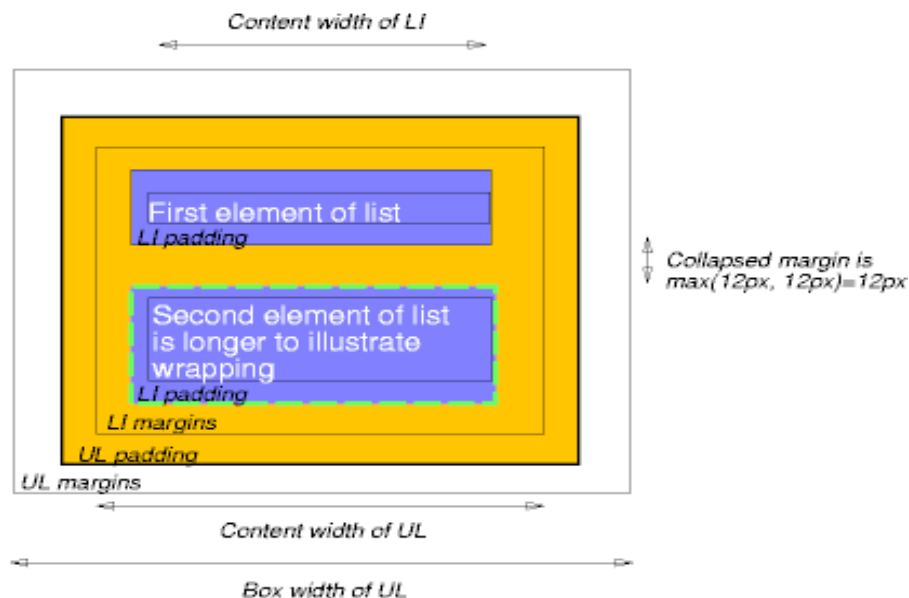
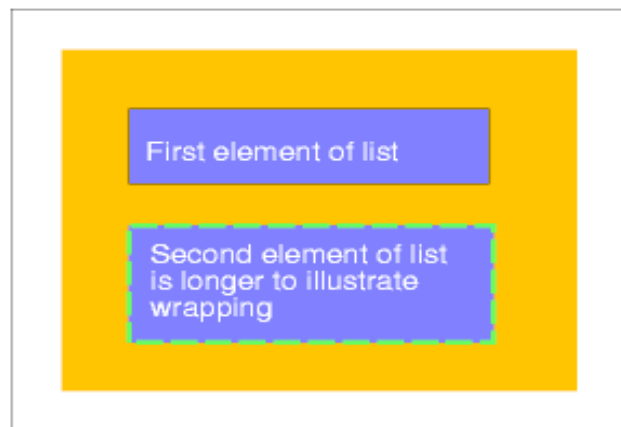
CSS

```
ul {  
  background: green;  
  margin: 12px 12px 12px 12px;  
  padding: 3px 3px 3px 3px;  
}  
li {  
  color: black;  
  background: gray;  
  margin: 12px 12px 12px 12px;  
  padding: 12px 0px 12px 12px;  
  list-style: none;  
}  
li.withborder {  
  border-style: dashed;  
  border-width: medium;  
  border-color: black;  
}
```

XHTML

```
<UL>  
  <LI>FIRST ELEMENT OF LIST</LI>  
  <LI CLASS="WITHBORDER">SECOND  
ELEMENT OF LIST IS LONGER TO  
ILLUSTRATE WRAPPING.</LI>  
</UL>
```

CSS2 Box Model (4 di 4)



CSS2 Visual Formatting Model (1 di 2)

- ✓ Nel Visual Formatting Model ogni elemento della struttura del documento genera zero o più contenitori a seconda del box model.
- ✓ La disposizione di queste caselle è disciplinata da:
 - ✓ dimensioni del box e tipo
 - ✓ schema di posizionamento (flusso normale, float o assoluto)
 - ✓ le relazioni tra gli elementi della struttura del documento
 - ✓ informazioni esterne (ad esempio, dimensione della finestra, le dimensioni intrinseche delle immagini,ecc.)

CSS2 Visual Formatting Model (2 di 2)

- ✓ Ci sono vari tipi di box. Il tipo di box dipende dal suo comportamento nel visual formatting model. La proprietà 'display' specifica il tipo di un box.
 - ✓ Elementi block-level: i block-level sono gli elementi del documento di origine che sono formattati visivamente come blocchi (ad esempio, paragrafi). Diversi valori della proprietà 'display' rendono un elemento block-level: 'block', 'list-item', 'compact', 'run-in' e 'table'.
 - ✓ Elementi inline-level: gli inline-level sono gli elementi del documento di origine, che non formano nuovi blocchi di contenuto, il contenuto è distribuito in linee (ad esempio, pezzi di testo sottolineati all'interno di un paragrafo, immagini inline, ecc). Diversi valori della proprietà 'display' rendono un elemento inline: 'inline', 'inline-table', 'compact' e 'run-in'. Elementi inline-level generano riquadri in riga.

CSS2 Schema di posizionamento (1 di 4)

- ✓ Nei CSS2 un box può essere redatto in base a tre schemi di posizionamento:
 - ✓ Flusso normale. Nei CSS2 il flusso normale consiste nella formattazione a blocchi dei block-box e la formattazione in linea degli inline-box, il posizionamento relativo di block e inline-block e il posizionamento di compact e run-in box.
 - ✓ Absolute. Nel modello di posizionamento assoluto, un box è rimosso del tutto dal normale flusso (non ha alcun impatto sui successori) e gli è assegnata una posizione fissata rispetto ad un blocco contenitore.
 - ✓ Float. Nel modello galleggiante un box prima si posiziona in accordo al flusso normale, poi è tolto dal flusso e spostato a sinistra o a destra, per quanto possibile. Il contenuto può fluire lungo il lato di un galleggiante.

CSS2 Schema di posizionamento (2 di 4)

- ✓ Flusso normale: i box nel flusso normale appartengono ad un contesto di formattazione, che può essere isolato o in linea, ma non entrambi contemporaneamente. I block-box partecipano al contesto di formattazione a blocchi. Gli inline-box partecipano al contesto di formattazione inline:
 - ✓ In un contesto di formattazione a blocchi, le caselle sono disposte una dopo l'altra, in verticale, partendo dalla cima di un blocco contenitore. La distanza verticale tra due caselle successive è determinata dalle proprietà 'margin'. In un contesto di formattazione a blocchi la sinistra di ogni scatola tocca il bordo esterno del lato sinistro del blocco contenitore (per la formattazione da destra a sinistra invece vale il viceversa). Questo è vero anche in presenza di float, sebbene la zona contenuta di un box può ridursi a causa di essi.
 - ✓ In un contesto di formattazione in riga le caselle sono disposte orizzontalmente una dopo l'altra, cominciando dalla cima di un blocco contenitore. I margini orizzontali, i bordi e il padding sono rispettati tra queste caselle. Quando diversi riquadrati inline non possono essere collocati orizzontalmente all'interno di una singola casella linea, essi sono distribuiti fra due o più caselle impilati verticalmente.

CSS2 Schema di posizionamento (3 di 4)

- ✓ Gli attributi `'position'` e `'float'` determinano quale degli algoritmi di posizionamento dei CSS2 è utilizzato per calcolare la posizione di una casella.
- ✓ `'position'` può essere `static`, `relative`, `absolute` o `fixed`:
 - ✓ `Static`, il box è un normale box che si posiziona secondo il normale flusso. Le proprietà `'left'` e `'top'` non si applicano.
 - ✓ `Relative`, la posizione del box è calcolata in base al flusso normale (questo è chiamato la posizione nel flusso normale). Poi il box è spostato rispetto alla sua posizione normale. Quando un box B è posizionato in modo relativo, la posizione dei box successivi è calcolata come se B non fosse stato spostato.
 - ✓ `Absolute`, la posizione del box (ed eventualmente le dimensioni) è specificato con le proprietà `'left'`, `'right'`, `'top'` e `'bottom'`. Queste proprietà specificano offset rispetto al blocco contenitore del box. Un box posizionato in modo assoluto viene portato fuori dal normale flusso. Questo significa che non hanno alcun impatto sul layout degli elementi successivi.
 - ✓ `Fixed`: la posizione del box è calcolata secondo il modello `'assoluto'`, ma inoltre il box è fissato rispetto ad alcuni riferimenti.

CSS2 Schema di posizionamento (4 di 4)

- ✓ Float: un float è una casella che si trova spostata a sinistra o a destra sulla linea di corrente. La caratteristica più interessante di un float è che il contenuto può scorrere lungo il suo fianco (caratteristica che può essere vietata dalla proprietà 'clear'). Il contenuto scorre lungo il lato destro di un box float-left e lungo il lato sinistro di un box float-right.
- ✓ Qualsiasi box float diventa un block-box che si sposta verso sinistra o verso destra fino che il suo bordo esterno tocca non tocca il bordo del blocco contenitore o il bordo esterno di un altro galleggiante. La parte superiore del box float è allineata con la parte superiore del box inline corrente (o inferiore del box block precedente se non esiste nessun box inline). Se non c'è abbastanza spazio orizzontale sulla linea corrente per il float, questo è spostato verso il basso, riga per riga, fino a una linea con sufficiente spazio.
- ✓ Dal momento che un float non è nel flusso, i box block creati dopo di esso scorrono sotto in verticale, come se il float non esistesse. Tuttavia, i box inline creati accanto al float sono accorciati per fare spazio ad esso. Qualsiasi contenuto nella riga corrente prima di un box float è riposizionato nella prima riga disponibile sul lato del float.

CSS2 esempio normal flow: display block

XHTML

```
<DIV CLASS="FIRST">  
  FIRST {DISPLAY: BLOCK}  
  <DIV CLASS="SECOND">SECOND {DISPLAY: BLOCK}</DIV>  
  <DIV CLASS="THIRD">THIRD {DISPLAY: BLOCK}</DIV>  
</DIV>
```

first {display: block}

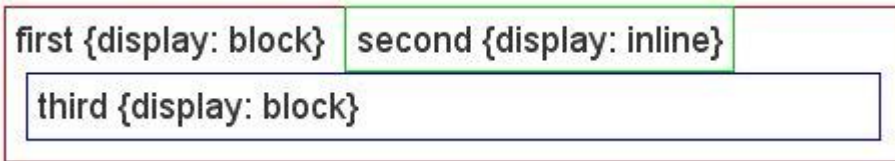
second {display: block}

third {display: block}

CSS2 esempio normal flow: display inline

XHTML

```
<DIV CLASS="FIRST">  
  FIRST {DISPLAY: BLOCK}  
  <DIV CLASS="SECOND" STYLE="DISPLAY: INLINE">  
    SECOND {DISPLAY: INLINE}  
  </DIV>  
  <DIV CLASS="THIRD">THIRD {DISPLAY: BLOCK}</DIV>  
</DIV>
```



XHTML

```
<DIV CLASS="FIRST">  
  FIRST {DISPLAY: BLOCK}  
  <DIV CLASS="SECOND" STYLE="DISPLAY: NONE">  
    SECOND {DISPLAY: NONE}  
  </DIV>  
  <DIV CLASS="THIRD">THIRD {DISPLAY: BLOCK}</DIV>  
</DIV>
```

first {display: block}

third {display: block}

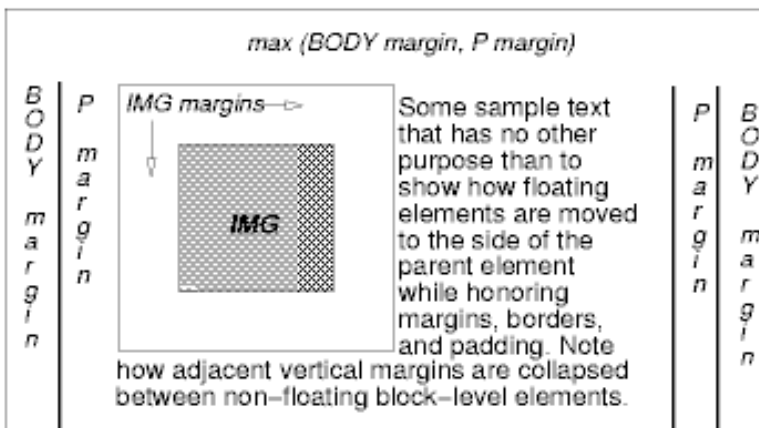
CSS2 esempio float

CSS

```
img { float: left }  
body, p, img { margin: 2em }
```

XHTML

```
<P>  
  <IMG SRC="IMG.GIF" ALT="THIS IMAGE WILL ILLUSTRATE FLOATS" />  
  SOME SAMPLE TEXT THAT HAS NO OTHER...  
</P>
```



Uno sguardo al futuro

✓ CSS3

<http://www.css3.com/about-css3/>

e alcuni link utili

✓ CSS Vault <http://cssvault.com/>

✓ CSS Beauty <http://www.cssbeauty.com/>

✓ A list apart <http://alistapart.com/>

✓ Boxes and arrows <http://www.boxesandarrows.com/>

✓ layout-o-matic

<http://www.inknoise.com/experimental/layoutomatic.php>

✓ Colourlovers <http://colourlovers.com/>