

CASCADING STYLE SHEET

Elementi base di CSS e i Frames

CSS

CASCADING STYLE SHEET

- ✘ I fogli di stile a cascata (detti anche semplicemente fogli di stile) vengono usati per definire la rappresentazione di documenti HTML, XHTML e XML.
- ✘ L'introduzione dei fogli di stile si è resa necessaria per separare i contenuti dalla formattazione.



Wikipedia

CASCADING STYLE SHEET



```
foundation.css
1085 button, .button {
1086   background-color: #008cba;
1087   border-color: #007095;
1088   color: white;
1089   border-style: solid;
1090   border-width: 0px;
1091   cursor: pointer;
1092   font-family: "Helvetica Neue",
1093   font-weight: normal;
1094   line-height: normal;
1095   margin: 0 0 1.25rem;
1096   position: relative;
1097   text-decoration: none;
1098   text-align: center;
1099   display: inline-block;
1100   padding-top: 1rem;
1101   padding-right: 2rem;
1102   padding-bottom: 1.0625rem;
1103   padding-left: 2rem;
}

app.css
1 button, .button {
2   background-color: slategray;
3   border-color: slategray;
4   color: whitesmoke;
5 }
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

SEPARAZIONE CONTENUTO/LAYOUT

- ✘ I tag html erano stati in origine studiati per definire il contenuto di un documento html
- ✘ Il layout del documento doveva essere preso in carica dal browser, senza usare tag di formattazione

SINTASSI CSS

- ✘ La più semplice regola css è composta di tre parti: ***un selettore, una proprietà ed un valore***:
`selettore{proprietà: valore}`
- ✘ Il selettore può essere l'elemento html che si vuole definire
- ✘ La proprietà è l'attributo che si vuole cambiare
- ✘ Alla proprietà è associato un valore
- ✘ La proprietà e il valore sono separati da due-punti e racchiusi tra parentesi graffe `body {color: black}`

COME INSERIRE UN FOGLIO DI STILE

- ✘ Fogli di stile esterni
- ✘ Fogli si stile interni
- ✘ Stile inline

FOGLI DI STILE ESTERNI

- ✘ Foglio esterno: ideale se l'applicazione ha molte pagine
- ✘ Si può cambiare l'aspetto di un intero sito modificando un solo file
- ✘ Ogni pagina deve essere collegata al foglio di stile
- ✘ Tag <link> all'interno della sezione head

```
<head>
```

```
  <link rel="stylesheet"  
  type="text/css"  
  href="miostile.css" >  
</head>
```

Il browser leggerà le definizioni di stile dal file miostile.css, e formatterà il documento di conseguenza

I LINK IN HTML

In html i link si realizzano nel modo seguente

Ecco un esempio di

``link a un documento nella stessa directory `` del documento corrente , un esempio di:

``link a un documento in una sotto - directory `` e un esempio di

``link a un documento in una directory superiore ``.

SCRIVERE UN CSS ESTERNO

- ✘ Un foglio di stile esterno può essere scritto in qualsiasi editor di testo
- ✘ Il file non deve contenere alcun tag html
- ✘ Il foglio di stile deve essere salvato con estensione .css
- ✘ Un esempio di foglio di stile:

```
body {  
background-image:  
    url("immagini/miosfondo.gif");  
}  
p {margin-left: 20px}
```

FOGLI DI STILE INTERNI

- ✘ Un foglio di stile interno può essere usato quando un documento html ha uno stile unico
- ✘ Si definisce nella sezione head con il tag <style>

```
<head>
```

```
<style type="text/css">
```

```
hr {color: red}
```

```
p {margin-left: 20px}
```

```
body {background-image:  
url("immagini/foto.jpg") }
```

```
</style>
```

```
</head>
```

Il browser leggerà le definizioni di stile, e formatterà il documento in accordo ad esse

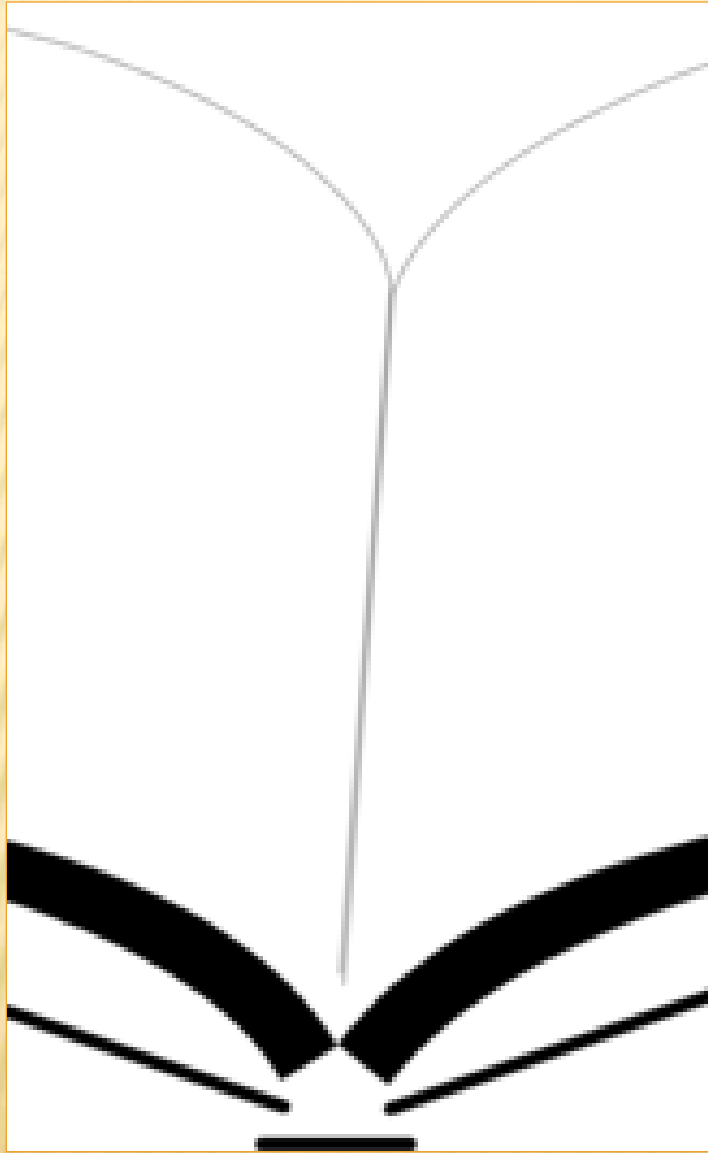
STILE INLINE

- ✘ Uno stile inline fa perdere molti dei vantaggi dei fogli di stile, perché mischia il contenuto con la presentazione
- ✘ Metodo da usare quando uno stile deve essere applicato ad una sola occorrenza di un elemento in un documento
- ✘ Per definire uno stile inline si usa l'attributo style dell'elemento da formattare
- ✘ L'attributo style può contenere qualsiasi proprietà CSS
- ✘ L'esempio mostra come cambiare il colore e il margine sinistro di un singolo paragrafo

```
<p style="color: green; margin-left: 20px">  
    Testo del paragrafo </p>
```

ORDINE DI PRIORITÀ (CASCATA)

- ✘ Tutti gli stili vengono raccolti a cascata in un nuovo foglio di stile virtuale secondo le seguenti regole
- ✘ Default del browser (priorità più bassa)
- ✘ Foglio di stile esterno
- ✘ Foglio di stile interno (nella sezione <head>)
- ✘ Stile inline (nell'elemento html – priorità più alta)



CSS

Sintassi

REGOLE

- ✘ Se il valore è composto da più parole, bisogna racchiuderlo tra virgolette doppie
- ✘ Se si vuole specificare più di una proprietà, bisogna separarle con un punto-e-virgola
- ✘ L'esempio mostra come definire un paragrafo con allineamento centrato e colore del testo rosso
- ✘

```
p {  
  text-align: center;  
  color: red;  
  font-family: "sans serif";  
}
```

RAGGRUPPARE I SELETTORI

- ✘ Si possono raggruppare i selettori: bisogna separare i selettori con una virgola
- ✘ Nell'esempio sotto, gli elementi di titolo sono raggruppati; ogni elemento di titolo sarà verde

```
h1, h2, h3, h4, h5, h6  
{  
    color: green;  
}
```

SELETTORE DI CLASSE

- ✘ Con il selettore di classe si possono definire stili diversi per uno stesso tipo di elemento html
- ✘ Es. due tipi di paragrafo: uno allineato a destra e uno allineato al centro
- ✘ `p.firma {text-align: right}`
- ✘ `p.importante {text-align: center}`
- ✘ Bisogna impostare l'attributo class nel documento html
- ✘ `<p class="firma">`
Paragrafo che sarà allineato a destra
`</p>`
- ✘ `<p class="importante">`
Paragrafo che risulterà centrato.
`</p>`

CLASSE

- ✘ Si può anche omettere il nome di tag nel selettore per definire uno stile da applicare a tutti gli elementi html di una certa classe
- ✘ Esempio: allineare al centro tutti gli elementi di classe *importante*
- ✘ `.importante {text-align: center}`
- ✘ Sia l'elemento `h1` che quello `p` sono di classe `importante`
- ✘ Entrambi gli elementi seguiranno le regole del selettore `.importante`
- ✘

```
<h1 class="importante">
  Titolo centrato
</h1>
<p class="importante">
  Anche questo paragrafo centrato
</p>
```

SELETTORE ID

- ✗ Mentre un selettore class può applicarsi a molti elementi su una pagina, il selettore id si applica ad un solo elemento
- ✗ Un attributo id deve essere unico all'interno del documento
- ✗ Es.: regola di stile che sarà applicata ad un elemento p che ha il valore di id posto a para1
- ✗

```
p#para1 {  
    text-align: center;  
    color: red;  
}
```
- ✗ Es.: regola di stile che sarà applicata all' elemento avente come id il valore xyz
- ✗

```
#xyz {color: green}
```
- ✗

```
<h1 id="xyz">Esempio di testo</h1>
```

COMMENTI

- ✘ Si possono inserire nel css commenti che possono essere d'aiuto quando il codice sorgente deve essere in seguito modificato
- ✘ I commenti vengono ignorati dal browser
- ✘ Un commento comincia con `/*` e termina con `*/`
- ✘

```
/* commento*/  
p {  
  text-align: center;  
  /* altro commento */  
  color: black;  
  font-family: arial;  
}
```

CSS

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!--Google Webmaster-->
<meta name="google-site-verification"
content="2AUbZzsdsIgyaQMm16c6GVpJ98NIXx3GbMorPJBDxM" />
<title>Albuquerque Web Design + Print Design - Roady Studios</title>
<link href="css/rs5.css" rel="stylesheet" type="text/css">
```

```
<!--Starting Google Analytics-->
<script type="text/javascript">
```

```
var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-11755188-1']);
_gaq.push(['_trackPageview']);

(function() {
var ga = document.createElement('script'); ga.type =
'text/javascript'; ga.async = true;
ga.src = ('https:' == document.location.protocol ? 'https://ssl' :
'http://www') + '.google-analytics.com/ga.js';
var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
})();
</script>
</head>
```

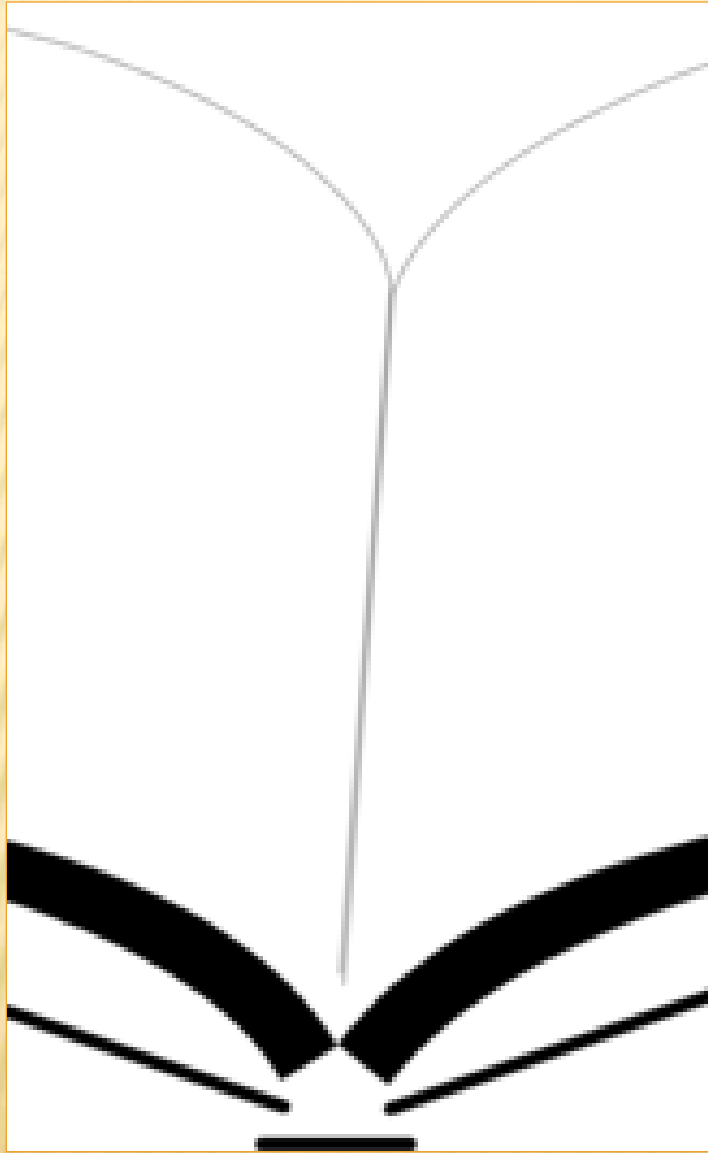
HTML



```
body {
font-family: "Helvetica Neue", Helvetica, Arial, sans-
font-size: 100%;
line-height: 1.5;
color: #000;
background-image: url('../images/bkgd.jpg');
}
#logo-header {
width: 100%;
text-align: center;
}
#logo-header img {
max-width: 80%;
height: auto;
}
#descrip-header {
width: 100%;
text-align: center;
}
.worktile img {
margin: 10px;
box-shadow: 0px 0px 7px #aaa;
-moz-box-shadow: 0px 0px 7px #aaa;
-webkit-box-shadow: 0px 0px 7px #aaa;
}
```

CSS





CSS

Proprietà

SFONDO

- ✘ background-color – Imposta il colore di sfondo di un elemento
- ✘ background-image – Imposta una immagine come sfondo – url(...)
- ✘ background-repeat – Imposta la ripetizione di una immagine di sfondo – repeat, repeat-x, repeat-y, no-repeat
- ✘ background-attachment – Immagine di sfondo fissa o che scorre con il testo – scroll, fixed
- ✘ background-position – Posizione di partenza di una immagine di sfondo – top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right, x-% y-%, x-pos y-pos
- ✘ background – Imposta tutte assieme le proprietà dello sfondo

SFONDO

```
✘ body { background-color: #FFFF00;  
background-image: url('2.jpg');  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: center center; }
```

TESTO

- ✘ color – Colore del testo
- ✘ direction – Direzione del testo – ltr, rtl
- ✘ text-align – Allineamento – left, right, center, justify
- ✘ text-decoration – Decorazione – none, underline, overline, line-through, blink
- ✘ text-indent – Indentazione della prima riga – length, %
- ✘ text-shadow – Ombra
- ✘ text-transform – Controlla le lettere – none, capitalize, uppercase, lowercase

TESTO

- ✘ `div {font-family: Georgia, "Times New Roman", serif;}`
- ✘ `p { font-family: verdana, arial, sans-serif; font-size: 12px; font-style: italic; font-weight: bold; }`

FONT

- ✘ font-style – Stile, corsivo – normal, italic
- ✘ font-variant – Maiuscoletto o normale – normal, small-caps
- ✘ font-weight – Peso, grassetto – normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900
- ✘ font-size – Dimensione – xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %
- ✘ font-family – Lista con priorità di nomi di famiglie di font e/o nomi di famiglie generiche – family-name, generic-family (serif, sans-serif, cursive, fantasy, monospace)
- ✘ font – Imposta tutte le proprietà del font

DIMENSIONI FONT

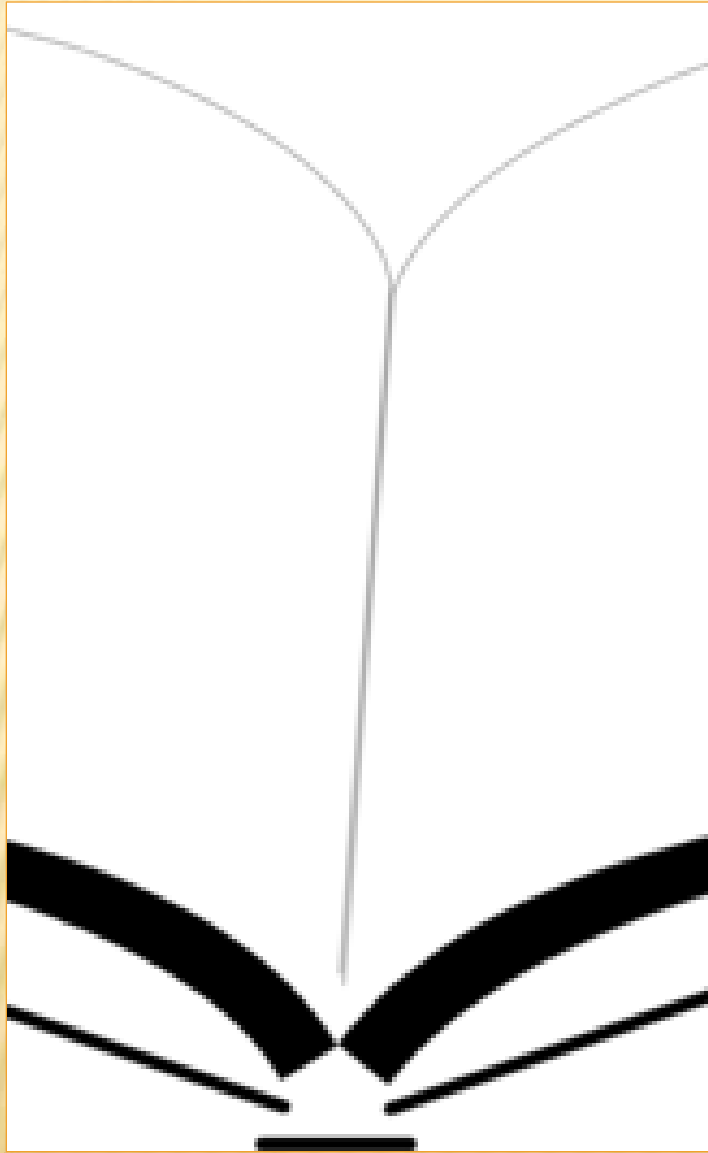
- ✘ px: misura in pixel (fissa rispetto a preferenze utente e risoluzione monitor)
- ✘ pt: punti tipografici, $1/72$ pollice (fissa rispetto a preferenze utente)
- ✘ em: rispetto alla dimensione di font attuale per l'elemento %: rispetto alla dimensione di font dell'elemento genitore

DIMENSIONI FONT

- ✘ `p {font-size: 12px;}`
- ✘ `div {font-size: 50%;}`
- ✘ `h2 {font-size: 1.2em;}`

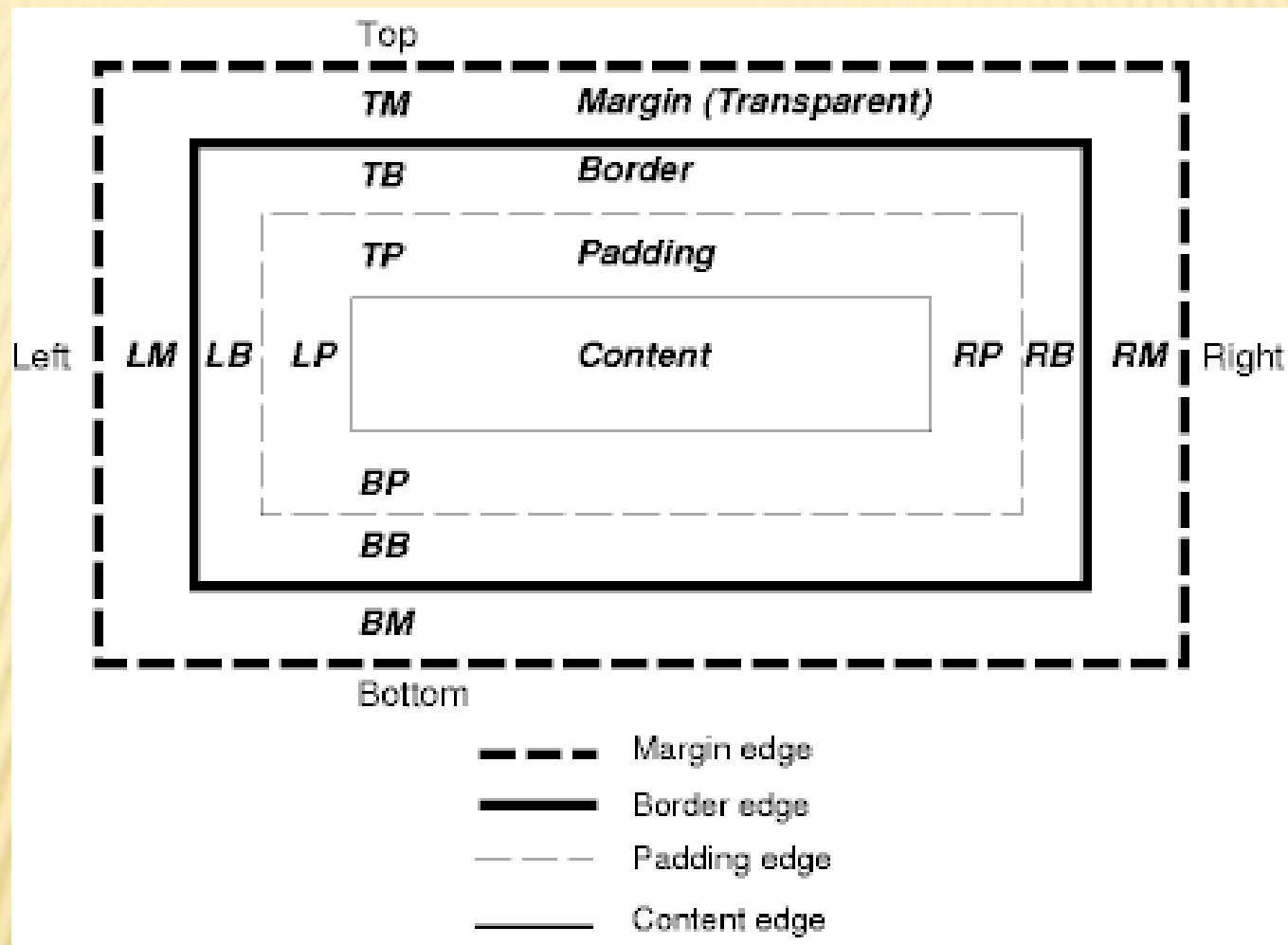
CSS





CSS

Box model



BORDI

- ✘ border-width – Ampiezza dei quattro bordi – 1-4 valori ()
- ✘ border-style – Quattro bordi – 1-4 valori – none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
- ✘ border-color – Colore – 1-4 valori – color
- ✘ border – Imposta tutte le proprietà (uguali) per i quattro bordi
important { border: 2px dashed red; }
- ✘ border-top, border-right, ... – Imposta tutte le proprietà per il bordo superiore, destro, ...
- ✘ blockquote { border-left: 2px solid black; }
- ✘ I bordi top, right, bottom, left possono essere cambiati indipendentemente usando proprietà separate: es. border-top-width

BORDER E OUTLINE: GESTIRE I BORDI CON I CSS

Definire lo stile di un singolo bordo

Iniziamo a vedere come impostare le proprietà per un singolo bordo. Questa la sintassi di base con le proprietà singole:

```
selettore {  
border-<lato>-color: <valore>;  
border-<lato>-style: <valore>;  
border-<lato>-width: <valore>;  
}
```

BORDER E OUTLINE: GESTIRE I BORDI CON I CSS

In entrambi gli esempi di sintassi sostituite a <lato> uno degli indicatori dei quattro lati: top, right, bottom o left.

Per quanto concerne i valori, come si vede dall'elenco delle proprietà, di ciascun lato si possono definire per il bordo tre aspetti:

il colore (color);

lo stile (style);

lo spessore (width).

I valori possibili per il colore sono:

- ✘ un qualsiasi colore;
- ✘ la parola chiave inherit.

BORDER E OUTLINE: GESTIRE I BORDI CON I CSS

Lo stile di un bordo può invece essere espresso con una delle seguenti parole chiave:

| <i>Stile bordo</i> | <i>Descrizione</i> |
|--------------------|--|
| none | l'elemento non presenta alcun bordo e lo spessore equivale a 0 |
| hidden | equivalente a none |
| dotted | bordo a puntini |
| dashed | bordo a lineette |
| solid | bordo solido e continuo |
| double | bordo solido, continuo e doppio |
| groove | tipo di bordo in rilievo |
| ridge | altro tipo di bordo in rilievo |
| inset | effetto 'incastonato' |
| outset | effetto 'sbalzato' |

MARGINI ESTERNI E INTERNI

- ✘ margin – Spazio attorno agli elementi – 1-4 valori
- ✘ È possibile usare valori negativi per sovrapporre il contenuto
- ✘ I margini top, right, bottom, left possono essere cambiati indipendentemente usando proprietà separate, es. margin-top
- ✘ È possibile usare la scorciatoia margin per impostare tutti i margini
- ✘ padding – Spazio tra il bordo e il contenuto degli elementi – 1-4 valori
- ✘ Non sono ammessi valori negativi
- ✘ I padding top, right, bottom, left possono essere cambiati indipendentemente usando proprietà separate, es. padding-top
- ✘ È possibile usare la scorciatoia padding per impostare tutti i padding

DIMENSIONI

- ✘ width – Ampiezza – auto, length, %
- ✘ height – Altezza – auto, length, %

POSIZIONE

- ✘ bottom – Distanza del limite inferiore di un elemento rispetto all'elemento che lo contiene – auto, %, length
- ✘ left – Sinistra ... – auto, %, length
- ✘ right – Destra ... – auto, %, length
- ✘ top – Superiore – auto, %, length
- ✘ vertical-align – Allineamento verticale – top, middle, bottom, baseline, sub, super, text-top, text-bottom, length, %
- ✘ position – Modalità di posizionamento – static, relative, absolute, fixed, inherit
- ✘ float – Elementi posizionati a lato, circondati da testo – left, right, none, inherit

VISIBILITÀ

- ✘ **z-index** – Ordine nello stack (rilievo, valori più alti sono in primo piano) – auto, number
- ✘ **overflow** – Impostazione per contenuto più ampio dell'area disponibile – visible, hidden, scroll, auto
- ✘ **visibility** – Visibilità – visible, hidden (occupa spazio)
- ✘ **display** – Visualizzazione – none (non occupa spazio), block, inline, ...

HTML (dalla versione 4) consente di visualizzare più documenti nella stessa finestra del browser attraverso l'uso di **frames**

L'uso dei frame in HTML prevede la presenza di un documento principale (detto **frameset document**) che definisce la strutturazione in frames (sotto-aree) della finestra del browser e contiene i riferimenti delle pagine da richiamare nei vari frames.

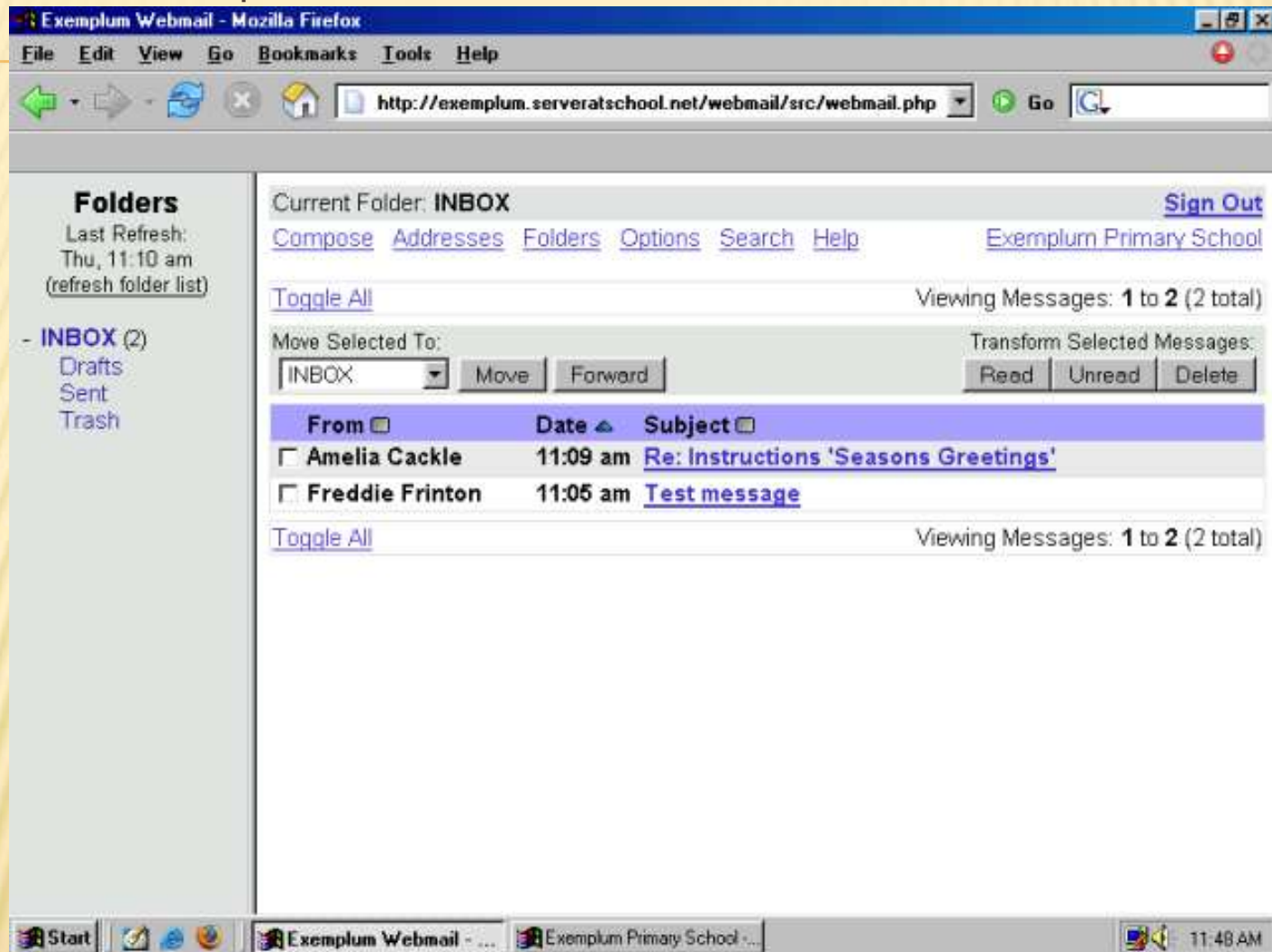
Le pagine web richiamate nei vari frames sono normali documenti HTML.

E' prevista una minima forma di interazione tra i frames, tipicamente un link presente in un frame può essere aperto in un altro frame

(tipico esempio: frame a lato della finestra con indice delle pagine del sito)

FRAMES (1)

Un classico esempio:



FRAMES (2)

Il frameset document è un documento HTML che al posto del `<body>` ha un tag `<frameset>` che descrive la strutturazione in frames della pagina

Il tag `<frameset>` può avere due attributi **rows** e **cols** che specificano le dimensioni di ogni riga e colonna. In questo modo si può strutturare la pagina come una tabella

Le dimensioni possono essere espresse in pixel o in percentuale sulla dimensione della finestra del browser:

- `<frameset cols="200,800">` indica una pagina con due colonne di larghezza 200 e 800 pixel
- `<frameset cols="200,800,*">` indica una pagina con tre colonne in cui ***** rappresenta \tutto quello che avanza in larghezza"
- `<frameset cols="30%,70%">` indica una pagina con due colonne
- `<frameset cols="50%,50%" rows="50%,50%">` indica una pagina divisa in quattro parti uguali

FRAMES (3)

Il contenuto del tag `<frameset>` e una sequenza di tag `<frame>`, uno per ogni frame della pagina

Il tag `<frame>` ha due attributi principali: **name** che associa un nome al frame, e **src** che contiene l'url del documento HTML da caricare nel frame

I vari frame vengono associati alle posizioni nella griglia definita dal tag `<frameset>` in maniera ordinata da sinistra a destra e poi dall'alto verso il basso

All'interno del tag `<frameset>` ci può essere un tag `<noframes>` che contiene un body HTML da visualizzare se il browser non supporta i frames

FRAMES (4)

Un semplice esempio di documento HTML con frames:

```
<html>
  <head><title>Documento con frames</title></head>
  <frameset cols="300,*">
    <frame name="elenco" src="elenco.html"/>
    <frame name="notizie" src="http://www.repubblica.it"/>
  </frameset>
</html>
```

Dove elenco.html potrebbe essere il seguente:

```
<html>
  <head><title>Elenco</title></head>
  <body>
    <h3>Elenco siti di notizie</h3>
    <ul>
      <li>Repubblica</li>
      <li>Corriere della Sera</li>
      <li>La Stampa</li>
    </ul>
  </body>
</html>
```

FRAMES (5)

Il risultato:



FRAMES (6)

E' possibile fare in modo che i link presenti in un frame vengano aperti in un altro frame o nell'intera pagina

Si usa l'attributo target di <a> specificando il nome del frame (attributo name di <frame>) in cui si vuole aprire il link

Ad esempio, modifichiamo elenco.html come segue:

```
<html>
  <head><title>Elenco</title></head>
  <body>
    <h3>Elenco siti di notizie</h3>
    <ul>
      <li><a href="http://www.repubblica.it"
        target="notizie">Repubblica</a></li>
      <li><a href="http://www.corriere.it"
        target="notizie">Corriere della Sera</a></li>
      <li><a href="http://www.lastampa.it"
        target="notizie">La Stampa</a></li>
    </ul>
  </body>
</html>
```

FRAMES (7)

Il risultato:



FRAMES (8)

INSERIRE UN FOGLIO DI STILE INTERNO CSS

In alcuni casi è utile definire un foglio di stile interno per dichiarare un elemento particolare, un **box**, questo elemento va dichiarato (e ne vanno definite le caratteristiche) nell'head del foglio html in questo modo:

```
<html><head>
<title>.....</title>
<style type="text/css">
#box {
.....
.....}
</style>
.....
</head>
```


INSERIRE UN FOGLIO DI STILE INTERNO CSS 2

Il box per poter essere utilizzato va poi richiamato all'interno del body in questo modo:

```
<body>
```

```
<div id="box">.....</div>
```

```
.....
```

```
.....
```

```
</body>
```

```
</html>
```

All'interno del tag `<div id=....>` possiamo inserire qualunque elemento di testo.

GESTIONE DEL COLORE

Si tratta di sedici keyword che definiscono i colori della palette VGA standard di Windows:

Ecco la lista completa:

black | navy | blue | maroon | purple | green | red | teal |
fuchsia | olive | gray | lime | aqua | silver | yellow | white

L'ordine è quello dei colori visualizzati nell'immagine.



GESTIONE DEL COLORE 2

Visti i sistemi per rappresentare i colori, dobbiamo ora chiarire un aspetto importante. Per ogni elemento si possono definire almeno tre colori:

- ✘ il colore di primo piano;
- ✘ il colore di sfondo;
- ✘ il colore del bordo;

La proprietà `color` definisce esclusivamente:

- ✘ il colore di primo piano, ovvero quello del testo;
- ✘ il colore del bordo di un elemento quando non si imposti esplicitamente quest'ultimo con le proprietà `border` o `border-color`.

OPACITY

Con opacity è possibile definire il livello di trasparenza di qualunque elemento. Di fatto, come recita la specifica, con opacity regoliamo il modo in cui un oggetto presente sulla pagina si fonde nella resa a schermo con lo sfondo.

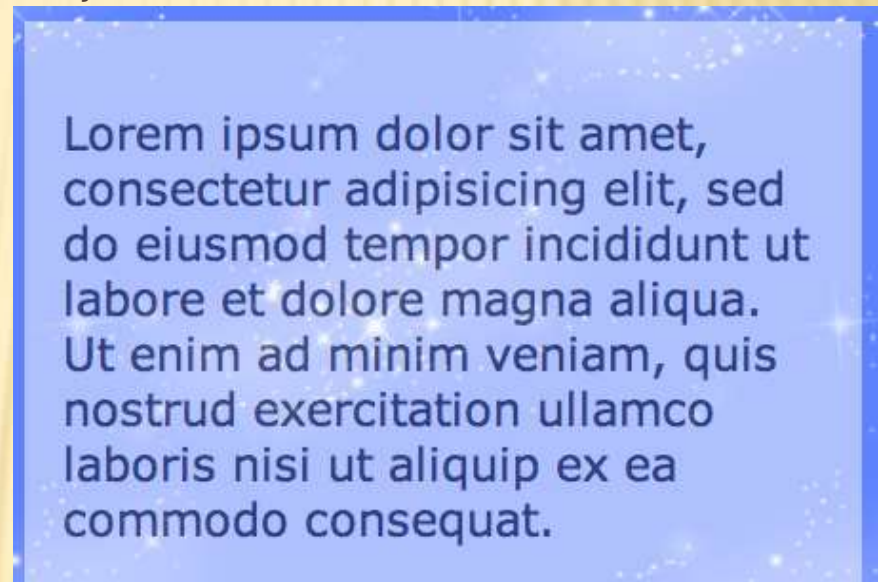
```
div {background:white; color:black; opacity: 0.5}
```

esempio

```
#box1 {  
width:400px;  
height:400px;  
padding:20px;  
color:#000;  
background-color: white;  
opacity:0.5;  
}
```

Le cose essenziali da notare sono due:

- ✘ il box è ora semi-trasparente (si intravede la texture dello sfondo);
- ✘ la trasparenza applicata con opacity viene ereditata da tutti gli elementi discendenti del div #box1; è quindi semi-trasparente anche il testo, e sarebbe stata semi-trasparente anche un'immagine inserita all'interno del div.



COLORE RGB

Da sempre, nei CSS, è possibile definire un colore con una notazione RGB, impostando cioè un valore per ciascun componente della tripletta di colori che è alla base di questo modello: Rosso (Red), Verde (Green), Blue (Blue).

I valori possono essere espressi in percentuale:

```
body {background-color: rgb(100%, 100%, 100%)}
```

oppure usando una scala che va da 0 a 255:

```
body {background-color: rgb(255, 255, 255)}
```

SFONDI MULTIPLI

I CSS3, invece, abbattano questo limite, consentendo agli sviluppatori di definire più immagini di sfondo per uno specifico elemento.

Ecco un esempio di codice che mostra come definire le immagini multiple:

```
background-image: url(immagine1.png), url(immagine2.png);
```

Come si vede, i valori relativi all'URL di ciascuna immagine vanno separati da una virgola. a specifica chiarisce poi un fattore decisivo: le immagini di sfondo vengono visualizzate su livelli diversi. L'immagine definita per prima nella lista apparirà sul primo livello, la seconda su un livello posto dietro al primo, e così via. La cosa è rilevante, ad esempio, nel caso di immagini che vadano a sovrapporsi o nel caso si utilizzino texture (vanno messe per ultime).

SFONDI MULTIPLI: ESEMPI

```
#box {  
width: 497px;  
height: 700px;  
margin: 20px auto;  
border: 1px solid #333;  
background-image:  
url(topRibbon.png),  
url(bottomRibbon.png),  
url(paper.jpg);  
background-position: top left,  
bottom left;  
background-repeat: no-repeat, no-  
repeat, no-repeat;  
background-color: white;  
}
```



BORDER-RADIUS

È una proprietà a sintassi abbreviata (shorthand) con cui è possibile definire con una sola regola i valori di arrotondamento per tutti e quattro gli angoli di un elemento. La proprietà accetta fino a quattro valori per ciascun raggio dell'angolo (quattro per il raggio orizzontale e quattro per quello verticale). I valori relativi al raggio verticale, se espressi, vanno separati da quelli relativi al raggio orizzontale con una slash (/).

Partiamo da degli esempi di base in cui al div **#box** è stata applicata questa semplice regola:

```
#box {border-radius: 20px}
```

```
#box {border-radius: 20px 40px 60px 80px}
```


BORDER-RADIUS

Vediamo ora un esempio completo:

```
<html><head>
<title>Bordi - Border-radius - Esempio</title>
<style type="text/css">
#box {
background: #4c74be;
width: 300px;
padding: 30px;
border-radius: 20px 40px 20px 40px/130px 150px 70px 90px;
</style>
</head>
<body>
<div id="box">Questa è una prova di uso di un box con gli spigoli arrotondati
</div>
</body>
</html>
```

BOX-SHADOW

`box-shadow: 5px 5px 10px 2px #333333;`

La sintassi di base prevede la definizione di quattro valori numerici:

Il primo valore (5px) imposta lo spostamento dell'ombra in senso orizzontale. Essendo il valore positivo, l'ombra viene collocata sul lato destro del box. Se si usano valori negativi l'ombra viene collocata sul lato sinistro.

Il secondo valore (5px) imposta lo spostamento in senso verticale. Dal momento che anche in questo caso si è usato un valore positivo, l'ombra viene collocata sul lato inferiore del box. Usando valori negativi l'ombra viene collocata sul lato superiore del box.

Il terzo valore (10px) definisce il livello di sfocatura (blur radius) dell'ombra: più alto è questo valore, più sfocata apparirà l'ombra. Si usi 0 se si desidera un'ombra netta e senza sfocatura. Non sono permessi valori negativi.

Il quarto valore (2px) imposta il livello di diffusione (spread radius) dell'ombra. Più i valori sono alti, più l'ombra tenderà ad espandersi, in tutte le direzioni. Se invece si usano valori negativi, l'ombra tende a contrarsi, fino a scomparire del tutto.

La nostra regola si chiude con l'indicazione del colore dell'ombra (#333333).

BOX-SHADOW 2

Esempio

```
#box1 {  
background: rgb(238, 238,  
238);  
width:200px;  
padding: 10px;  
box-shadow: 5px 5px 10px 2px  
#333333;  
}
```

✘ Si ottiene così:

Box 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

BOX-SHADOW 3 LA PAROLA CHIAVE INSET

Usando questo valore l'ombra non si colloca all'esterno del box, ma al suo interno.

Esempio

```
#box2 {  
background: rgb(238, 238,  
238);  
width:200px;  
padding: 10px;  
box-shadow: 5px 5px 10px 2px  
#333333 inset;  
}
```

✘ Si ottiene così:

Box 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

BACKGROUND-ORIGIN

La proprietà background-origin serve a specificare relativamente a quale area di un elemento avviene il posizionamento di un'immagine di sfondo. Di fatto, è usata sempre in combinazione con le proprietà background-image e background-position.

background-origin non può essere applicata quando e se la proprietà background-attachment è impostata sul valore fixed.

La proprietà prevede tre diversi valori:

- ✘ padding-box: la posizione dello sfondo è relativa all'area del padding (padding box);
- ✘ border-box: la posizione dello sfondo è relativa al bordo definito per l'elemento;
- ✘ content-box: la posizione dello sfondo è relativa all'area del contenuto.

BACKGROUND-ORIGIN: ESEMPIO E SINTASSI

Sul primo box (Box 1), abbiamo posizionato lo sfondo relativamente all'area del padding:

```
div#box1 {  
width: 450px;  
padding: 200px;  
border: 10px dashed red;  
background: url(' ');  
background-repeat: no-repeat;  
background-position: 0 0;  
background-origin: padding-box;  
}
```

Si ottiene così



LAYOUT MULTICOLONNA

Sono le proprietà essenziali, quelle con cui si imposta il layout multi-colonna nei suoi fondamenti:

- ✘ column-width
- ✘ column-count
- ✘ column-gap
- ✘ column-rule

Al momento queste quattro proprietà sono le uniche del modulo a godere di un supporto esteso sui vari browser.

Proprietà per la gestione delle colonne

Le ultime due proprietà definite nel modulo consentono di gestire altrettanti importanti aspetti di un layout multi-colonna:

- ✘ column-span
- ✘ column-fill

Con column-span possiamo decidere per quante colonne del layout un certo elemento si può espandere.

columnf-fill serve a bilanciare nel modo adeguato il contenuto tra le varie colonne.

LAYOUT MULTICOLONNA: COLUMN-WIDTH E COLUMN-GAP

Un primo modo per suddividere il testo su più colonne consiste nell'uso della proprietà column-width. Ecco il codice:

```
#container {
width: 750px;
margin: 20px auto;
-column-width: 350px;
column-gap: 50px;
}
```

Si ottiene:



LAYOUT MULTICOLONNA: COLUMN-WIDTH E COLUMN-GAP, COMPATIBILITÀ

Per rendere il contenuto dei nostri elementi compatibile con la maggior parte dei browser è buona norma inserire del codice che migliora l'eseguibilità da parte dei browser stessi:

```
-webkit-column-width:.....;  
-webkit-column-gap:.....;  
-moz-column-width:.....;  
-moz-column-gap:.....;  
}
```

LAYOUT MULTICOLONNA: COLUMN-WIDTH E COLUMN-GAP, LEGGIBILITÀ DEL CODICE

Complessivamente si ottiene quindi per esempio:

```
#prova_colonne {  
width: 750px;  
margin: 20px auto;  
-webkit-column-width:350px;  
-webkit-column-gap:25px;  
-moz-column-width:350px;  
-moz-column-gap:25px;  
column-width: 350px;  
column-gap: 25px;  
}
```

BIBLIOGRAFIA-SITOGRAFIA

- ✘ Cloud, Gallo Sirsi, Minerva Scuola
- ✘ PHP, Camagni Nicolassi, Hoeply
- ✘ http://www.disi.unige.it/person/MoggiE/LI04/lez3/appunti_html.pdf
- ✘ <http://www.di.unito.it/~rossano/DIDATTICA/INF-0607/dams-laboratorio-appunti-2.pdf>
- ✘ <http://www.di.unipi.it/~milazzo/teaching/AA1011-WebProg/slides/3-HTML-Partel.pdf>
- ✘ <http://www.html.it/guide/guida-CSS/>