

CSS

Massimo Martinelli

Massimo.Martinelli@isti.cnr.it



*Consiglio Nazionale delle Ricerche* - CNR  
Istituto di Scienza e Tecnologie della Informazione - ISTI



# Cosa è CSS?

## **Cascading Style Sheets** = Fogli di Stile in Cascata

- sono stati introdotti per aggiungere informazioni sulle proprietà dello stile ai documenti HTML (font, borders,...)
- Si possono utilizzare anche con XML
- Un foglio di stile è un elenco di regole  
Le regole sono composte dal nome degli elementi a cui sono applicate e dallo stile che vogliamo applicare a questi elementi
- Ciascuna regola è composta da  
un selettore, il nome dell'elemento a cui si applica,  
e una lista di proprietà da applicare a quell'elemento

```
pre { display: block; font-size: 16pt; font-weight: bold; }
```

Il contenuto dell'elemento PRE deve essere mostrato

- in un blocco (display: block) (scatola)
- con un font di 16 punti (font-size: 16pt)
- in grassetto (font-weight:bold)

# Selezione di Elementi

La parte di regola CSS che specifica a quali elementi si applica è detta *selector* - *selettore*.



termina  
ogni  
coppia  
proprietà/  
valore



# Tipi di Selettori

- Semplici I selettori possono specificare un elemento
- Raggruppamenti più elementi
- Classi elementi con un particolare attributo CLASS o ID
- Pseudo classi elementi in un particolare stato o posizione
- Pseudo elementi parte del contenuto di un elemento
- Contestuali elementi che appaiono in contesto relativo ad altri elementi

# Selettore Semplice

- Il selettore più comune è il nome di un elemento  
Esempio: h1

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

# Raggruppamento di Selettori

- *Se abbiamo regole identiche che si applicano ad elementi diversi*

```
p { display: block; margin-bottom: 10px; }
```

```
h1 { display: block; margin-bottom: 10px; }
```

- *Possiamo raggruppare gli elementi separandoli con una virgola*

```
p, h1 { display: block; margin-bottom: 10px; }
```

# Classi

- *Elementi con un particolare attributo CLASS o ID*

```
<p class="Particolare">
```

```
<p id="ID1">
```

```
p.Particolare    { color: red; }
```

```
p                { display: block; }
```

```
p#ID1            { font-weight: bold; }
```

# Pseudo Classi

- *Per selezionare elementi in un particolare stato derivato dall'interazione con lo user agent (browser)*
  - **a:link** non visitato, link non attivo
  - **a:hover** mouse sopra
  - **a:active** cliccato
  - **a:visited** visitato
- **:focus** "l'oggetto a fuoco" (l'oggetto selezionato)
- **:first-child** il primo figlio dell'elemento
- **:left** le pagine a sinistra di un documento  
*(generalmente per la stampa)*
- **:right** le pagine a destra



# Pseudo Elementi

- *La prima lettera di un paragrafo*

```
p:first-letter { font-size: 300%; }
```

- *La prima linea di un paragrafo*

```
p:first-line { font-variant: small-caps; }
```

**before, after:** si applica prima / dopo il contenuto dell'elemento.

# Selettori Contestuali

- Se la formattazione dipende dalla posizione all'interno di altri elementi (per esempio se dipende dall'elemento padre) è possibile scrivere regole che si applicano agli elementi contenuti all'interno di altri specifici elementi
- Anteporre il nome dell'elemento avo (e padre) a quello dell'elemento al quale vogliamo applicare la proprietà di stile specifica.

```
p cite { color: yellow; }
```

```
<p>paragrafo con <cite>citazione</cite> interna</p>
```

paragrafo con **citazione** interna

# Altri selettori

- **Selettore universale (\*)**:

\*: qualsiasi cosa

- **Selettori contestuali (A D P>F F + PF)**:

*Basati sugli elementi figli, fratelli, genitori*

Avod Discendente

D discendente di A

Padre>Figlio

F figlio di P

Fratello+PrimoFratello

PF primo fratello di F

```
H2+P { color: red; }
```

- **Selettori di attributi (E[attr] E[attr="valore"])**:

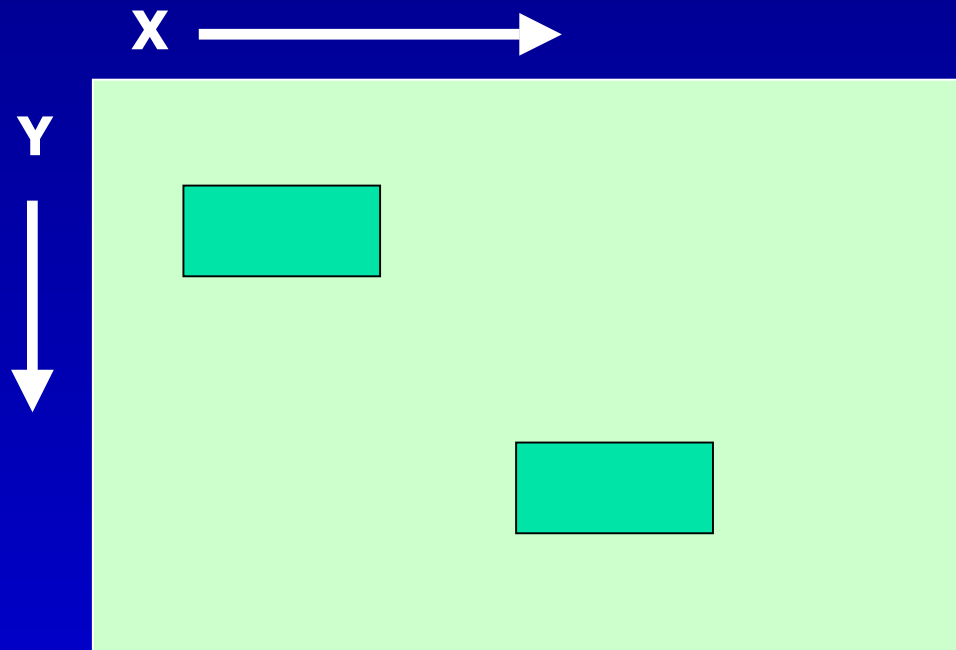
E[attr] elementi E che hanno un attributo attr

E[attr="valore"] elementi E che hanno un attributo attr il cui valore è "valore"

```
A[NAME] { color: green; }
```

# Modello di formattazione

- CSS definisce un modello di visualizzazione bidimensionale
- Gli elementi da visualizzare sono inseriti in rettangoli immaginari detti scatole (box)

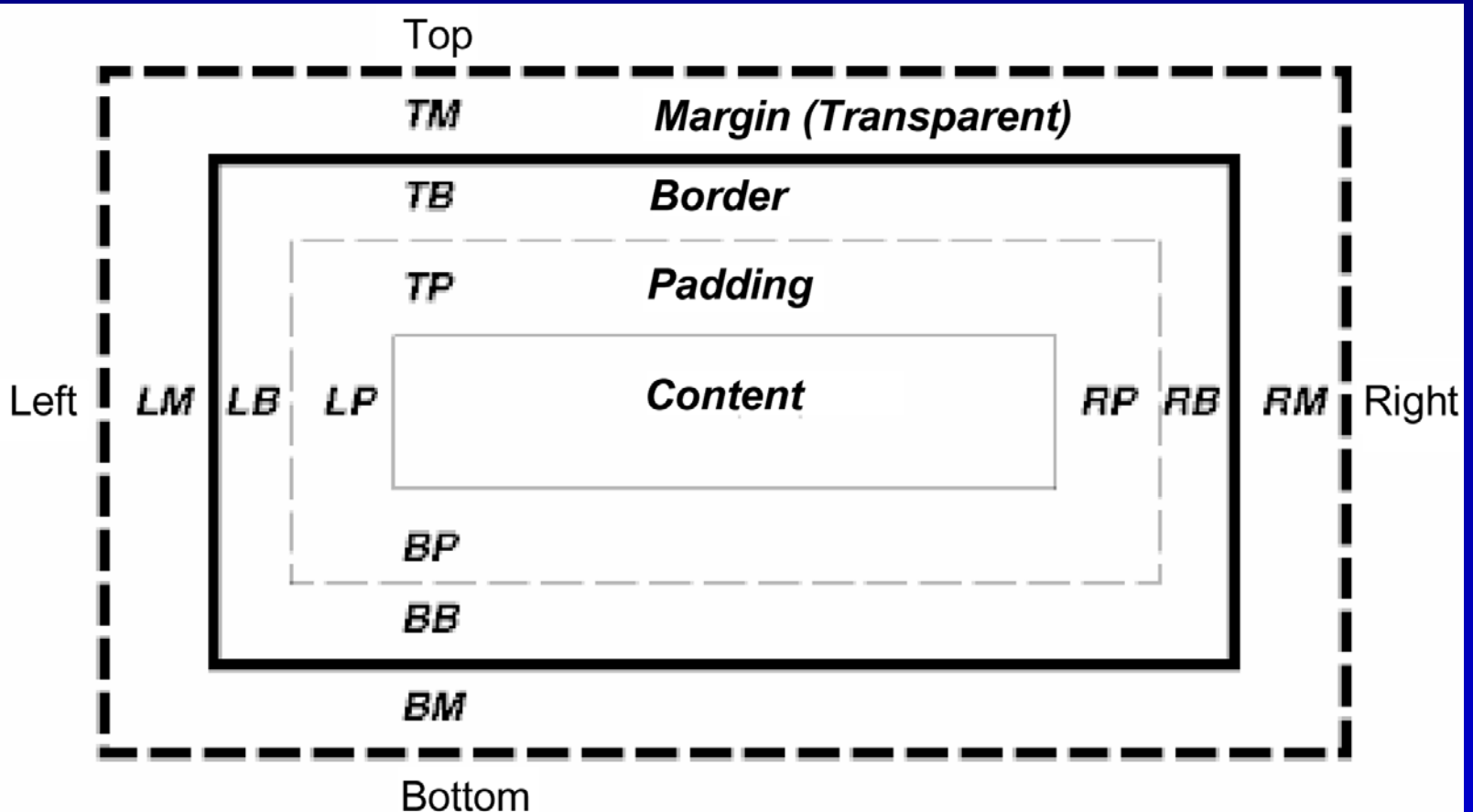


# Box

Le scatole hanno delle proprietà che permettono di definire

- altezza, larghezza, dimensione e posizione
- aree relative a margini, cuscinetti (margini interni – *padding*), bordi (trasparenti per default)

La larghezza della scatola è data dalla somma della larghezza del contenuto (ovvero dell'elemento testo o immagine) + quella delle aree dei padding, bordi e margini.



# Box

- Margini: la regione che separa una scatola dall'altra, necessariamente trasparente.
  - **margin-top, margin-bottom, margin-left, margin-right**: dimensioni del margine della scatola.
- Border: la regione ove visualizzare un bordo per la scatola (trasparente per default).
  - **border-top, border-bottom, border-left, border-right, border-width, border-color**: dimensioni ed aspetto del bordo.
  - **border-style**: può assumere come valori **none, dotted, dashed, solid, double, groove (incavo), ridge (rilievo), inset, outset**.
- Padding: la regione di respiro tra il bordo della scatola ed il contenuto. Ha il colore dello sfondo.
  - **padding-top, padding-bottom, padding-left, padding-right**: dimensioni del padding della scatola.
- Content: la regione dove sta il contenuto dell'elemento.
  - **background-color, background-image, background-repeat, background-attachment, background-position**: colore, immagine e meccanismo di ripetizione dell'immagine di sfondo della scatola.

**inset** The border makes the entire box look as though it were embedded in the canvas.

**outset** The opposite of 'inset': the border makes the entire box look as though it were coming out of the canvas.

# Tipi di elementi

- *Elementi "Block" (blocco) sono elementi che sono mostrati in un blocco per conto proprio*  
**p** (paragrafo), **div** (blocco generico), **pre** (blocco preformattato), **address** (autore della pagina), **blockquote** (citazione lunga)
  - *possono essere inclusi solo da altri elementi blocco.*
- *Elementi "Inline" sono elementi che non provocano un a capo e possono essere inclusi da qualsiasi altro elemento*  
**a** (ancora, collegamento), **em** (enfasi), **strong** (maggiore enfasi)  
**span** (generico elemento inline), **dfn** (definizione), **code** (frammento di programma), **samp** (output d'esempio), **kdb** (testo inserito dall'utente), **var** (variabile di programma), **cite** (breve citazione), **q** (citazione lunga), **abbr**, **acronym** (abbreviazioni ed acronimi), **sup**, **sub** (testo in apice e in pedice), **bdo** (bidirectional override)
- *Elementi "List-item" sono elementi con un marcatore (punto elenco, numero)*

*Ne riparlamo più tardi*

# Posizionamento

## Come si inseriscono le scatole nello spazio di visualizzazione:

- **Flusso normale di tipo blocco:** le scatole sono poste l'una sopra l'altra in successione verticale (come paragrafi).
- **Flusso normale di tipo inline:** le scatole sono poste l'una accanto all'altra in successione orizzontale (come parole della stessa riga)
- **Flusso di tipo float:** le scatole sono poste all'interno del contenitore e poi spostate all'estrema sinistra o destra della scatola, lasciando che le altre scatole vi girino intorno.
- **Posizionamento assoluto:** le scatole vengono poste nella posizione indicata indipendentemente dal flusso e da quel che la zona già visualizza (eventualmente nascondendo ciò che sta sotto).



# Modello visuale di CSS

- proprietà che controllano il tipo di posizionamento e quello di scatola:
  - **display** (`inline` | `block` | `list-item` | ... | `none`): il tipo di scatola da utilizzare per l'elemento: un blocco, un inline, una lista, una cella di tabella, ecc.
  - **position** (`static` | `relative` | `absolute` | `fixed`): il posizionamento rispetto al flusso del documento.
  - **float** (`left` | `right` | `none`): un float è una scatola scivolata all'estrema destra o sinistra del contenitore muovendo le altre per farle posto.
  - **z-index**: la posizione nello stack di scatole potenzialmente sovrapposte. Il valore più alto è più vicino al lettore, e quindi nasconde gli altri. Per default il valore di background delle scatole è trasparente.
  - **top, bottom, left, right**: coordinate della scatola
  - **width, height**: dimensioni usabili invece di right e bottom.

## Un esempio di posizionamento

```
p.abs { position: absolute; top: 40px; left: 210px;
        width: 190px; background:white;
        border-style: solid; border-width: 1px;}
p      { display: block; border-style: solid;
        border-width: 1px; }
b,i    { display:inline; border-style: solid;
        border-width: 1px; background:yellow;}
span.left { border-style: solid; border-width: 1px;
            float:left; font-size: 200%;}
```

<p>Alcune parole di un paragrafo che si estende per  
<span class="left">righe e righe</span>, cosgrave;  
da far vedere come si comporta su pigrave;  
righe.</p>

<p>Secondo paragrafo che contiene altre parole e un  
pezzo in <b>grassetto</b> ed uno in  
<i>corsivo</i>.</p>

<p class="abs">Terzo paragrafo posizionato in maniera  
assoluta dove capita </p>

# Il testo

- Del testo è possibile controllare sia gli aspetti relativi al font che quelli relativi all'organizzazione del testo nella scatola di riferimento:
  - font-family: il/i nomi del/dei font
  - font-style (`normal` | `italic` | `oblique`), font-variant (`normal` | `small-caps`), font-weight (`normal` | `bold` | `bolder` | `lighter` | `100`-<-> `900`), font-stretch (`normal` | `wider` | `narrower` | `ultra-condensed` | `extra-condensed` | `condensed` | `semi-condensed` | `semi-expanded` | `expanded` | `extra-expanded` | `ultra-expanded`): caratteristiche del font
  - text-indent, text-align (`left` | `right` | `center` | `justify`), line-height: indentazione, allineamento e interlinea delle righe della scatola.
  - text-decoration (`none` | `underline` | `overline` | `line-through` | `blink`), text-shadow: ulteriori stili applicabili al testo
  - letter-spacing e word-spacing: spaziatura tra lettere e tra parole
  - text-transform (`capitalize` | `uppercase` | `lowercase` | `none`): trasformazione della forma delle lettere.
  - white-space (`normal` | `pre` | `nowrap`): specifica la gestione dei ritorni a capo e del collassamento dei whitespace all'interno di un elemento.

# Applicazione dello stile

- E' possibile applicare lo stile CSS:
  - All'interno del il tag di riferimento
  - All'interno del tag style
  - Importandolo con il tag style
  - Indicandolo nel tag link

# Applicazione dello stile: all'interno del il tag di riferimento

...

```
<h1 style="color:red;">Intestazione</h1>
```

...

## Applicazione dello stile: all'interno del tag style

...

```
<head>
```

```
  <title>titolo della pagina</title>
```

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

```
    h1 { color: red; }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Intestazione</h1>
```

...

# Applicazione dello stile: importandolo con il tag style

xhtmlfile.html

...

```
<head>
```

```
  <title>titolo pagina</TITLE>
```

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

```
    @import url(/stili/nomefile.css);
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Intestazione</h1>
```

...

nomefile.css

```
h1 {color:red;}
```

# Applicazione dello stile: indicandolo nel tag link

xhtmlfile.html

```
...  
<head>  
  <title>titolo pagina</title>  
  <link type = "text/css"  
    rel = "stylesheet"  
    href = "/stili/nomefile.css" />  
</head>  
<body>  
  <h1>Intestazione</h1>  
...
```

nomefile.css

```
h1 {color:red;}
```

## Associazione fogli di stile ad XML:

```
<?xml-stylesheet type="text/css" href="/stili/nomefile.css"?>
```



# Commenti in CSS

- i fogli di stile CSS possono includere commenti. I commenti CSS sono come quelli del C `/* */`, non `<!-- -->` come quelli di XML e HTML.

```
/* commento */
```

```
cite { display:block; }
```

```
/* commento */
```

```
pre { display: block; font-size: 16pt; font-weight: bold; }
```

```
p { display: block; margin-bottom: 10; }
```

```
/* commento
```

```
Commento */
```

```
h1 { display: block; margin-bottom: 10; }
```

# Unità CSS

- Le proprietà CSS possono essere nomi o valori  
I nomi sono tutte parole chiave CSS
- Per quanto riguarda i valori esistono diverse possibilità.
  - Alcuni di questi sono parole chiave come "display: none" in display o come solid in "border-style: solid"
  - Lunghezze espresse come
    - Numeri interi: valori numerici assoluti
    - Valori numerici espressi in unità di misura  
0,5in (margin-top: 0.5in), 12pt (font-size: 12pt)
    - Percentuali espressi in relazione al contenitore padre
  - Altri valori possono essere URL come  
<http://www.isti.cnr.it/immagini/carta.gif>  
(background-image: url(<http://www.isti.cnr.it/immagini/carta.gif>),  
oppure colori RGB come #CC0033 (color: #CC0033)
- Esistono quattro tipi di valori di proprietà:
  1. length
  2. URL
  3. color
  4. keyword

# Esempi: nomi di proprietà e valori

<i>Nome</i>	<i>Valore</i>
display	none
font-style	italic
margin-top	0.5in
font-size	12pt
border-style	solid
color	#CC0033
background-color	white
background-image	url( <a href="http://www.isti.cnr.it/immagini/sfondoblu.gif">http://www.isti.cnr.it/immagini/sfondoblu.gif</a> )
list-style-image	url(/immagini/pallinorosso.png)
line-height	120%

# Lunghezze - Length

- Usate per definire la larghezza, altezza, dimensione di parole e lettere, spazi, indentazione di testo, altezza di linee e margini, padding, larghezze di bordi, ...
- Le lunghezze possono essere specificate in tre modi:
  1. Unità assolute      in (inches), cm, mm, pt (points), pc (pica)
  2. Unità relative      bigger, +1
  3. Percentuali          50%

# Unità di lunghezza relative

- CSS mette a disposizione tre metodi per specificare le unità relative di lunghezza.
  1. em: la larghezza della lettera *m* del font corrente
  2. ex: l'altezza della lettera *x* del font corrente
  3. px: la grandezza di un pixel
- In questo esempio lo spessore dei bordi destro e sinistro dell'elemento *p* corrisponderanno alla larghezza della lettera *m* del font corrente, i bordi superiore e inferiore alla metà dell'altezza della lettera *x* del font corrente:

```
p { border-right-width: 1em; border-left-width: 1em;  
    border-top-width: 0.5ex; border-bottom-width: 0.5ex; }
```

# Unità di lunghezza in percentuale

- E' sconsigliato usare la misura di lunghezza in pixel per vari motivi:
  - La grandezza dei pixel varia con la risoluzione
  - Con il passare del tempo i monitor incrementano la densità di punti
  - Con l'aumentare della risoluzione la specifica del pixel può rendere illeggibile il testo.
- Le lunghezze possono essere specificate in percentuale rispetto al valore della grandezza dell'elemento padre.

```
p:first-letter { font-size: 300%; }
```

```
em           { font-size: 120%; }
```

# Valori URL

- Tre proprietà CSS possono avere valori URL:  
background-image, list-styleimage, list-style.
- Anche la regola @import usa un valore URL.
- I letterali URLs sono inserite dentro url().
- Sono permesse tutte le forme relative e assolute di URL

```
body { background-image: url (http://www.isti.cnr.it/sfondo.gif) }  
table { background-image: url(/immagini/img1.gif) }  
tr { background-image: url(../immagini/img2.gif)}  
td { background-image: url(immagine.gif)}
```

- Le URL si possono includere tra apici singoli o doppi

```
body { background-image: url ("http://www.isti.cnr.it/sfondo.gif") }  
table { background-image: url("/immagini/img1.gif") }  
tr { background-image: url(' ../immagini/img2.gif')}  
td { background-image: url('immagine.gif')}
```

# escape

- Parenthesi, virgole, spazi, apici singoli (') e doppi (") negli URL devono essere precedute da un carattere backslash: '\(', '\)', '\,', '\ ' (escaped)
- oppure sostituite dall'entità corrispondente:

space	%20
,	%2C
\	%27
"	%22
(	%2B
)	%2D



# Valori di Color

- E' possibile assegnare colori agli elementi (quasi tutti) della pagina con la proprietà *color*.

Il valore di *color* può corrispondere a

- una parola chiave

CSS definisce 16 nomi di colore:

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

- una tripla RGB come valore decimale, esadecimale o in percentuale.

# Colori

<i>Colore</i>	<i>RGB Decimale</i>	<i>RGB Esadecimale</i>	<i>Percentuale RGB</i>
red	rgb(255,0,0)	#FF0000	rgb(100%, 0%, 0%)
blue	rgb(0,0,255)	#0000FF	rgb(0%, 0%, 100%)
green	rgb(0,255,0)	#00FF00	rgb(0%, 100%, 0%)
white	rgb(255,255,255)	#FFFFFF	rgb(100%, 100%, 100%)
black	rgb(0,0,0)	#000000	rgb(0%, 0%, 0%)
brown	rgb(153,102,51)	#996633	rgb(60%, 40%, 20%)
pink	rgb(255,204,204)	#FFCCCC	rgb(100%, 80%, 80%)
orange	rgb(255,204,204)	#FFCC00	rgb(100%, 80%, 0%)

```
p { color: black }  
em { color: blue}
```

- La proprietà color è ereditata dagli elementi figli.
- Nell'esempio tutti gli elementi figli di p saranno di colore nero, ad eccezione di em.

# Elementi Block, Inline e List Item

- In HTML, *em*, *strong*, *b*, *i*, e *a* sono elementi inline.  
*em*, *strong*, *b*, *i* nella riga sopra sono elementi inline, non sono separati dal resto del testo

In generale, gli elementi block-level elements sono separati dagli altri elementi da un a capo.

In HTML *p*, *blockquote*, *h1..h6* sono tutti esempi di elementi block-level.

Gli elementi list-item sono elementi block-level elements con un marcatore che li precede.

In HTML, *li* è un elemento list-item.

Elementi con proprietà *display* impostata a *none* sono invisibili e non mostrati sullo schermo. Non influiscono sulla posizione degli altri elementi visibili sullo schermo.

In HTML, *title*, *meta*, e *head* dovrebbero avere una proprietà *display = none*.

In XML, *display: none* è spesso usata per le meta informazioni.

# Tabelle

- **table** (In HTML: TABLE)
  - Un elemento che definisce una tabella (block-level).
- **inline-table** (TABLE)
  - Un elemento che definisce una tabella inline
- **table-row** (TR) un elemento che descrive una riga di celle
- **table-row-group** (TBODY) un elemento che raggruppa 1 o + righe
- **table-header-group** (THEAD)
  - Come table-row-group, ma per la visualizzazione il gruppo è mostrato prima di tutte le altre righe e gruppi di righe e dopo top caption.
- **table-footer-group** (TFOOT)
  - Come table-row-group, ma per la visualizzazione il gruppo è mostrato dopo di tutte le altre righe e gruppi di righe e prima di bottom caption.
- **table-column** (COL) un elemento che descrive una colonna di celle
- **table-column-group** (COLGROUP)
  - Specifica che un elemento raggruppa una o + colonne
- **table-cell** (TD, TH) un elemento che rappresenta una cella
- **table-caption** (CAPTION) specifica una legenda (intestazione o didascalia) per la tabella.

## Il foglio di stile di default per le tabelle

- `table { display: table }`
- `tr { display: table-row }`
- `thead { display: table-header-group }`
- `tbody { display: table-row-group }`
- `tfoot { display: table-footer-group }`
- `col { display: table-column }`
- `colgroup { display: table-column-group }`
- `td, th { display: table-cell }`
- `caption { display: table-caption }`

# font

- proprietà di base per specificare i font:

font-family

font-style

font-variant

font-weight

font-size

- Un solo attributo "font" permette di impostare tutte le proprietà (*forma abbreviata*)

# font-family

- font-family: elenco di tipi di carattere o di famiglie di tipi di carattere separati da virgola
- tipo di carattere: un particolare tipo  
Helvetica, Tahoma, "Times New Roman"  
(nomi di font che contengono spazi tra doppi apici)
- famiglie di tipi di carattere:  
serif, sans-serif, cursive, fantasy, monospace  
il browser per ogni famiglia ha un elenco di nomi di font corrispondenti  
utilizzerà il font installato nel computer locale

- Tipo di carattere (se non trova il primo usa il secondo,....)

```
body { font-family: "Times New Roman", Courier, Tahoma }
```

- Famiglia (se non trova la prima usa la seconda,....)

```
body { font-family: serif, fantasy }
```

- Tipo o famiglia (se non trova 1o usa 2o,....)

```
body { font-family: serif, Helvetica, fantasy }
```

# Famiglie di tipi di carattere

- serif (*con grazie*) “Times New Roman” (Palatino,..) T
- sans-serif (*senza grazie*) “Verdana” (helvetica,...) T
- cursive (*grafia a mano*) “*Shelley*” (ZapfChancery,...)
- fantasy (*decorativo*) “Comic Sans”,
- monospace (*spaziatura fissa*) “Courier”



# font-style

Tre possibili valori:

- normal (detto anche Roman)
- italic *italico*
- oblique simile ad italic ma generato dal computer inclinando il font attraverso un algoritmo

Questa regola imposta il font-style italic per l'elemento cite

```
cite { font-style: italic; }
```

# font-variant

Due possibili valori:

- normal                    normale
- small-caps                maiuscoletto

Se non esiste una versione di maiuscoletto riduce le lettere maiuscole alla dimensione delle lettere minuscole

- esempio la seconda parola usa lo stile font-variant:small-caps

**Prova** PROVA

# font-weight

Spessore del carattere (definisce quanto deve tendere al "bold" o al "light")

- normal
- bold
- 100
- 200
- 300
- 400
- 500
- 600
- 700
- 800
- 900

Esistono anche

- bolder
- lighter

Si riferiscono allo spessore dell'elemento genitore (più spesso, più leggero)

- Esempio

```
h1, h2 { font-weight: bold }
```

# font-size

- Determina l'altezza e la larghezza di un carattere tipico del font
- Può essere specificata come parola chiave, come valore assoluto, come valore relativo o come percentuale rispetto al carattere dell'elemento padre.

## Parole chiave (valore assoluto)

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

Si preferisce usare le parole chiave piuttosto che impostare un valore assoluto perché in questo modo sono relative al font di base della pagina. (Posso impostare un font di base della pagina e tutti gli altri saranno di dimensioni relative a questo, l'utente può scalare i font sulla base delle proprie esigenze in modo sicuro)

- **Valori relativi** (*rispetto alla grandezza del font dell'elemento padre*)  
`b { font-size: larger } (b sostituito da strong in xhtml2) (smaller, larger,)`  
b avrà un valore di font-size più grande dell'elemento padre

# font-size

- **Percentuale dell'elemento padre**

Il fonto usato per l'elemento **b** è di dimensione pari al 150% dell'elemento padre

`b { font-size: 150% }`

- **Valore assoluto di lunghezza**

Oltre ad usare pixel, centimetri e pollici, l'unità più comune è il punto. il valore di font-size per l'elemento **p** (e per i figli che ereditano) è di 14 punti.

`p { font-size: 14pt }`

- **E' consigliabile non usare unita di valori assoluti per impostare la dimensione dei font**

La dimensione varia tra le diverse piattaforme e risoluzioni utilizzabili. Inoltre se l'utente volesse aumentare o diminuire la dimensione lo può fare solo se tutte le misure sono in proporzione ad un valore di base.

# ESERCIZIO

- Scrivere un file XHTML e un foglio di stile esterno con le seguenti caratteristiche:

Intestazione (centrata, bianco, sfondo blu)

Paragrafo testo color rosso allineato a sinistra

Paragrafo testo color blu allineato a destra

# background

- Lo sfondo di un elemento (background) può essere impostato con un colore o con una immagine.
- Nel caso di una immagine questa può essere posizionata in modi differenti relativamente al contenuto dell'elemento.
- Esistono queste proprietà di base
  1. background-color
  2. background-image
  3. background-repeat
  4. background-attachment
  5. background-position
- La proprietà background permette di impostare tutti i valori in una unica regola

# background-image

- background-image
  - none (the default)
  - URL (indirizzo di una immagine)

```
p { background-image: url(immagine.gif); }
```



# background-attachment

- Generalmente l'immagine sullo sfondo è attaccata al documento.
- Quando si scorre il documento l'immagine scorre con questo.
- Con la proprietà background-attachment è possibile specificare se lo sfondo deve scorrere con il testo o deve rimanere fisso
- Con background-attachment impostato a fixed, il documento scorre ma l'immagine rimane fissa.

```
body { background-attachment: fixed;  
        background-repeat: no-repeat; }
```

# background-position

L'angolo in alto a sinistra dello sfondo è allineato per default allo stesso angolo dell'elemento a cui è associato.

E' possibile allinearlo in modo differente specificando

- la percentuale della larghezza e dell'altezza dell'elemento padre,
- la lunghezza in valore assoluto
- Usando una delle seguenti parole chiave:
  - top
  - center
  - bottom
  - left
  - center
  - Right

## Altezza e larghezza in percentuale rispetto all'elemento padre

- La coordinata  $x$  è data in percentuale da 0% (lato sinistro) a 100% (destra).
- La coordinata  $y$  è data in percentuale da 0% (alto) a 100% (basso).
- Nel seguente esempio l'angolo in alto a destra dell'immagine è allineato con l'angolo in alto a destra dell'elemento *p*.

```
p { background-image: url(foto.gif);  
background-repeat: no-repeat;  
background-position: 100% 0% }
```

# Lunghezze assolute

- La posizione assoluta 0, 0 corrisponde all'angolo in alto a sinistra (origine).
- Questa regola pone l'immagine di background immagine.gif due centimetro a destra e un centimetro in basso rispetto all'origine dell'elemento.

```
p { background-image: url(immagine.gif);  
    background-repeat: no-repeat;  
    background-position: 2cm 1cm; }
```

# background-position

```
p { background-image: url(immagine.gif);  
  background-repeat: no-repeat;  
  background-position: center center; }
```

top left  
left top  
0% 0%

top  
top center  
center top  
50% 0%

top right  
right top  
100% 0%

left  
center left  
left center  
0% 50%

center  
center center  
50% 50%

right  
center right  
right center  
100% 50%

bottom left  
left bottom  
0% 100%

bottom  
bottom center  
center bottom  
50% 100%

bottom right  
right bottom  
100% 100%