

Lezione 2: Formattazione del testo

Ripasso

- (X)HTML è una versione di HTML basata sul metalinguaggio XML e ha una sintassi rigorosa. Serve per definire la struttura logica dei documenti.
- I fogli di stile CSS (Cascading Style Sheet) permettono di definire gli aspetti di presentazione dei documenti.
- Quindi:
 - (X)HTML: struttura
 - CSS: presentazione

Attributi comuni - 1

- Classi (“class”) e identificativi (“id”) consentono di assegnare un’etichetta (un “nome”) ad un qualunque elemento.
- Gli id identificano in modo univoco un singolo elemento all’interno di un documento HTML;

```
<p id=“mittende”>...</p>
```

- Le **classi** possono essere associate ad un numero indefinito di elementi.

```
<p class=“classeUno”>...</p>
```

Attributi comuni - 2

- Esempio “id” e “class”:

```
<body>
```

```
  <p id="mittente">Fabiana</p>
```

```
  <p id="destinatario">Agnese</p>
```

```
  <p class="domanda">p>Quali sono gli  
    argomenti della seconda lezione?</p>
```

```
  <p class="domanda">Qual è la differenza  
    tra classi e identificativi?</p>
```

```
</body>
```

Attributi comuni - 3

- Più classi possono essere associate allo stesso elemento, con la seguente sintassi:

```
<p class="classeUno  
classeDue">...</p>
```

CSS: sintassi

Selettori HTML

- Selettori HTML: utilizzano i nomi dei tag HTML (h1, p, a...). Le dichiarazioni saranno applicate a tutti gli elementi caratterizzati dal tag usato.
- Esempio:

```
h1 { font-size: 2em; }
```

La dichiarazione viene applicata a tutti gli elementi di tipo h1

Nomenclatura

- I CSS permettono di definire gli aspetti di presentazione di un documento (X)HTML.

- Esempio:



```
h1 { font-size: 2em; }
```

The diagram illustrates the structure of a CSS rule. A large red bracket above the entire line labels it as "regola". A smaller red bracket below "font-size:" labels it as "proprietà". Another red bracket below "2em;" labels it as "valore". A red bracket below "h1 {" labels it as "selettore". A red bracket below the entire content inside the curly braces labels it as "dichiarazione (termina con \";\");".

selettore
(definisce gli elementi a cui si applica la regola)

dichiarazione (termina con “;”)

Selettori id

- Selettori id: si basano sul valore dell'attributo "id". Le dichiarazioni saranno applicate a tutti gli elementi il cui "id" corrisponde a quello indicato (uno solo per documento!).

- Esempio:

`#mittente { font-size: 2em; }` Nelle regole CSS, i selettori id sono introdotti dal simbolo "#"



La dichiarazione viene applicata all'elemento il cui id è "mittente"

Selettori di classe - 1

- Selettori di classe: si basano sul valore dell'attributo "class". Le dichiarazioni saranno applicate a tutti gli elementi la cui classe corrisponde a quella indicata.

- Esempio:

 .argomento { font-size: 2em; }

Nelle regole CSS,
i selettori di classe sono
introdotti dal simbolo "."

La dichiarazione viene applicata agli elementi della classe "argomento".

- Se ad un elemento sono collegate più classi, gli verranno applicate tutte le regole corrispondenti.

Selettori di classe - 2

- Si può restringere il campo di applicazione di un selettore di classe ad un solo tipo di tag accodandolo al selettore HTML corrispondente.

- Esempio:

```
h1.argomento { font-size: 2em; }
```

La dichiarazione viene applicata a tutti gli elementi di tipo h1 il cui attributo “class” ha valore “argomento.”

Selettori raggruppati

- Si possono associare le stesse dichiarazioni a più selettori separando i nomi degli stessi con una virgola.
- Non è necessario definire regole separate.
- Esempio:

```
#mittente, .argomento, h2 {font-size: 2em;}
```

Selettori annidati

- Si possono applicare delle regole CSS a degli elementi annidati indicando, prima del nome dell'elemento prescelto, una sequenza di elementi genitore, separati da spazi. Il selettore diventa più specifico!
- Esempio:

```
#mittente p a { font-size: 2em; }
```

La dichiarazione viene applicata agli elementi di tipo “a” contenuti in un elemento “p” contenuto a sua volta in un elemento con identificativo “mittente”.
- Non è necessario indicare tutti gli elementi genitore!

Valori: colori - 1

- **Valori RGB:** sono costituiti da tre valori (numeri da 0 a 255 o percentuali) che indicano i livelli di rosso, verde e blu. (red, green, blu)

```
p { color: rgb(0, 0, 255); }
```

- Si utilizza il sistema di **codifica esadecimale** anziché quello decimale. Si basa su un sistema di numerazione in base 16 che comprende le cifre da **0 a 9** e da **A a F**. I valori esadecimali sono costituiti dal simbolo “#” seguito da 3 o 6 caratteri esadecimali (le cifre vanno da 0 a 9 e poi da a ad f).

```
p { color: #0000ff; }
```

Valori: colori - 2

- Nomi di colori: esistono 17 valori standard, oltre a transparent: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow.

	Black		Yellow		Red		Maroon
	Gray		Lime		Green		Olive
	Silver		Aqua		Blue		Navy
	White		Fuchsia		Purple		Teal

Colori “sicuri”

- **Web Safe:** sono 216 colori che si vedono nello stesso modo, indipendentemente dal browser in cui appaiono. Sono composti (in notazione esadecimale) soltanto da combinazioni delle seguenti coppie: 00, 33, 66, 99, CC e FF.
 - Esempio: “#990066” è web safe, “#99006F” no
- **Web Smart:** sono 4096 colori che si vedono nello stesso modo sui monitor con 56000 o 16 milioni di colori. Sono composti da combinazioni di tutte le possibili coppie di cifre (da 0 a F).
 - Esempio: “#5522EE” è web smart

Link utili

- Alcune risorse online per trovare i codici RGB e esadecimali dei colori:
 - <http://cloford.com/resources/colours/websmart1.htm>
 - <http://colorblender.com/>
 - <http://www.ficml.org/jemimap/style/color/wheel.html>
 - <http://www.colorschemer.com/online.html>

Commenti

- Si può inserire un commento in qualunque punto del codice CSS con la seguente sintassi:

```
/*testo del commento*/
```

Applicare i CSS ad (X)HTML

CSS e XHTML

- Ci sono tre tipi di collegamento tra CSS e XHTML
 - **CSS esterni**
 - **CSS embedded**
 - **CSS inline**

1.CSS esterni

- Le regole CSS si trovano in un file esterno, con estensione .css.
- Struttura e presentazione sono completamente separate!
- Il file CSS è richiamato all'interno del file (X)HTML attraverso un apposito elemento "link" all'interno dell'elemento "**head**"

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

2.CSS embedded

- Le regole CSS sono contenute in un elemento “**style**” all’interno dell’elemento “**head**” del documento (X)HTML.
- Struttura e presentazione non sono completamente separate: si usa solo se ci sono regole specifiche per una sola pagina.

```
<style type="text/css">  
  a { font-size: 2em; }  
</style>
```

3.CSS inline

- Le regole CSS sono contenute nell'attributo “**style**” dell'elemento che si vuole modificare (nel body).
- Struttura e presentazione non sono separate: si usa solo per test e prove rapide!

```
<p style="color: olive;">testo del  
paragrafo</p>
```

Al lavoro! - 1

- Nella vostra cartella, create una sottocartella (es. “css”) che conterrà i vostri file css.
- Dentro questa cartella create il vostro primo file CSS:
 - Aprite Notepad++ (o un altro editor)
 - Create un nuovo documento vuoto
 - Salvate questo documento nella sottocartella, indicando:
 - Nome file: nome.css
 - Salva come: All types (*.*)

Al lavoro! - 2

- Create la vostra prima regola css, che modifica il colore del font:

```
p { color: #0000ff; }
```

- Aprite il vostro file xhtml e inserite l'elemento "link" nell'"head" per linkare il file CSS:

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

- Aprite il file xhtml con il browser: cosa succede?

Ordine di applicazione

- Nel caso in cui ci siano definizioni di stile ripetute (con lo stesso selettore), è sempre l'ultima a prevalere. Nel caso di dichiarazioni in fogli di stile diversi, l'ordine di applicazione è il seguente (prevale l'ultimo foglio applicato):
 - CSS di default del browser
 - CSS esterno
 - CSS embedded
 - CSS inline
- Nel caso più dichiarazioni (con selettori diversi) si applichino allo stesso elemento, prevarrà quella con il selettore più **specifico** (ad es. l'id prevale sulla classe).

Dare stile al testo con i CSS


Dare stile al testo coi CSS

significa impostare con i CSS:

- Il tipo di font
- Il colore del font
- La dimensione del font
- Grassetto e corsivo del font
- Allineamento
- Sottolineatura
- Spaziature, bordi, margini (box)

Tipo di font - 1

- Il tipo di font si può scegliere attraverso la proprietà CSS “**font-family**”. Il tipo di font viene generalmente definito per l’elemento body e sarà così utilizzato per tutto il documento.
- Esempio:

```
body { font-family: “Times New Roman”; }
```
- I nomi di font composti da più parole vanno inseriti tra virgolette.
-  Possono essere visualizzati solo i font presenti sul computer dell’utente. I font più comunemente utilizzati sono: “Times New Roman”, arial, helvetica (per Mac), Century Gothic, Avant Garde (per Mac), verdana.

Tipo di font - 2

- Se si indicano come valore di “font-family” più nomi di font, separati da virgola, il browser visualizzerà il primo disponibile. In questo modo, si può controllare la presentazione su diversi sistemi operativi o quando la prima scelta non è disponibile.
- E’ bene indicare, come ultima alternativa, un tipo di font generico, come “serif”, “sans-serif” o “monospace”.
- Esempio:

```
h2 { font-family: arial, helvetica, sans-serif; }
```

Colore

- La proprietà “**color**” permette di controllare il colore del testo.
- La proprietà “**background-color**” (applicabile anche ad elementi non testuali) permette di definire il colore di sfondo.
- Esempio:

```
body { color: #666;  
        background-color: #FFF; }
```
- E' pratica comune definire dei colori “di base” per l'elemento “body”, che saranno applicati a tutto il documento. Si possono poi definire colori diversi per elementi specifici.

Valori: unità di misura

- Le unità di misura possono si distinguono in assolute e relative.
- Assolute: punti (pt), pixel (px)
- Relative: percentuali(%), em

- Esempio:

```
h2 { font-size: 2em; }
```



Il suffisso che indica l'unità di misura si scrive subito dopo il numero, senza spazi.

Em è un'unità di misura relativa che si basa sulla dimensione del carattere M maiuscolo del font dell'elemento contenitore

Dimensione

- La dimensione dei font viene controllata attraverso la proprietà “**font-size**”.
- Le unità di misura più utilizzate sono i pixel (px), gli em e le percentuali (%).
- In genere, è meglio usare unità di misura relative (em , %) per consentire all’utente di ridimensionare il testo. Si consiglia di definire il valore di “font-size” in % per il “**body**” e in em per gli altri elementi.

```
body { font-size: 80%; }  
h1 { font-size: 3em; }
```

Grassetto & corsivo

- La proprietà CSS “**font-weight**” definisce il “peso” (lo spessore...) di un font. I valori più comuni sono “bold” (grassetto) e “normal”.

- Esempio:

```
h2 { font-weight: bold; }
```


- La proprietà “**font-style**” definisce lo “stile” (l’inclinazione...) del testo. I valori più comuni sono “**italic**” (corsivo) e “normal”.

- Esempio:

```
h2 { font-style: italic; }
```

Allineamento orizzontale

- Con la proprietà “**text-align**” si gestisce l’allineamento del testo all’interno di un elemento di blocco (ad esempio un paragrafo).
 - “left” (allineato a sinistra),
 - “right” (allineato a destra),
 - “center” (centrato),
 - justify (giustificato).
- Esempio:

```
body { text-align: center; }
```
-  Sul Web, il testo allineato a sinistra risulta più leggibile del testo giustificato.


Maiuscole & minuscole

- La proprietà CSS “**font-variant**” valorizzata con (=con valore) “*small caps*” converte il testo in maiuscoletto.
- La proprietà CSS “*text-transform*” può avere tre valori:
 - “*capitalize*” (rende tutte le iniziali maiuscole),
 - *uppercase*” (tutto il testo maiuscolo),
 - *lowercase*” (tutto il testo minuscolo).
- Esempio:

```
h1 { text-transform: uppercase; }
```

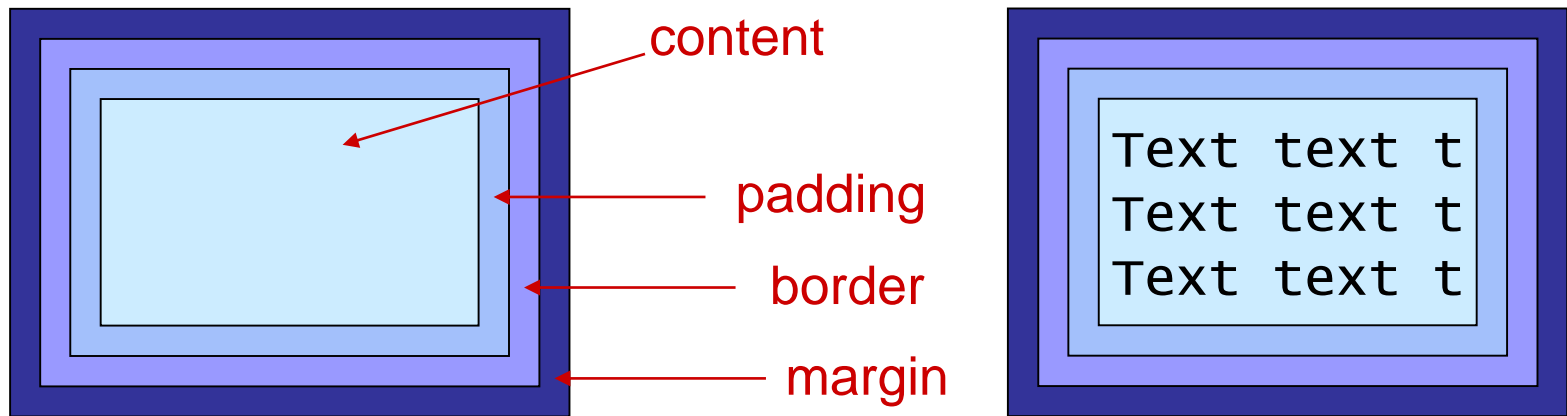
Sottolineature

- La proprietà “**text-decoration**” può avere 4 valori: “*underline*” (sottolineato), “*overline*” (riga sopra al testo), “*line-through*” (testo barrato), “*none*” (nessuna sottolineatura).
- Esempio:

```
del { text-decoration: none; }
```
-  Nei doc ipertestuali gli elementi sottolineati sono normalmente riconosciuti come collegamenti (*link*): meglio usare le sottolineature con cautela per non confondere l'utente!

Il box model: introduzione

- Ciascun elemento di un documento ipertestuale (ad es, un paragrafo o un'intestazione...) è inserito in un contenitore (box) a strati, le cui caratteristiche possono essere manipolate tramite apposite proprietà CSS.



Box model per un paragrafo

Padding

- Le dimensioni di ogni singolo lato dell'area di padding possono essere definite attraverso le proprietà **“padding-top”**, **“padding-right”**, **“padding-bottom”**, **“padding-left”**. Attraverso la proprietà **“padding”** si può definire il padding per tutti i lati contemporaneamente.
- I valori per il padding sono unità di misura, ad esempio in em o in pixel.
- Esempio:

```
h1 { padding: 2em; }
```
- Quando si definisce uno sfondo per un elemento, con il semplice colore o con un'immagine (vedremo come...) questo occupa l'area definita da contenuto e padding.

Border - 1

- La proprietà “**border-width**” (e border-top-width, border-right-width, border-bottom-width, border-left-width) permette di definire le **dimensioni** del bordo, come per il padding.
- La proprietà “**border-color**” (e border-top-color, border-right-color, border-bottom-color, border-left-color) permette di definire il **colore** del bordo.

Border - 2

- La proprietà “**border-style**” (e `border-top-style`, `border-right-style`, `border-bottom-style`, `border-left-style`) permette di definire lo “**stile**” del bordo. I valori più comuni sono: **solid** (continuo), **dashed** (tratteggiato) e **dotted** (puntinato).
-  E' obbligatorio definire lo stile del bordo perché le dichiarazioni relative a colore e dimensione abbiano un effetto visibile.
- Esempio:

```
p { border-style: solid;  
    border-color: blue;  
    border-style: dotted; }
```

Border - 3

- Le caratteristiche dei bordi possono essere gestite in modo più sintetico attraverso la proprietà “border” (e border-top, border-right, border-bottom, border-left), in cui il primo valore corrisponde alla dimensione, il secondo allo stile e il terzo al colore.
- Esempio:

```
p { border: 1px dashed silver;  
      border-left: 2px solid maroon; }
```

Margin

- Il “**margin**” rappresenta l’involucro esterno del box. Come per il padding, se ne possono definire le dimensioni tramite le proprietà “margin-top”, “margin-right”, “margin-bottom”, “margin-left” e “margin”.
- Esempio:

```
p { margin-bottom: 15px; }
```

Al lavoro! - 3

- Create un file xhtml con almeno due paragrafi e due intestazioni (una di primo e una di secondo livello).
- Create un css esterno e collegatelo al vostro file xhtml.
- Create delle regole css per:
 - Dare un colore diverso alle due intestazioni;
 - Usare un font diverso per intestazioni e paragrafi;
 - Dare all'intestazione di primo livello dimensioni doppie rispetto a quelle dei paragrafi.

Al lavoro!-4

- Nel vostro documento XHTML assegnate a due elementi (per esempio due paragrafi) l'attributo "class" con lo stesso valore.
- Assegnate ad altri due elementi l'attributo id (come devono essere i valori?).
- Create delle regole css per:
 - Dare un colore diverso ai due elementi selezionabili tramite id;
 - Usare un font particolare per gli elementi selezionabili tramite class;

Al lavoro! - 5

- Definite una regola css per impostare le caratteristiche del box model di uno dei paragrafi come segue:
 - Dimensione di padding 3em
 - Bordo con dimensione 4em, di colore verde e tratteggiato;
 - Un margine di 10px