



CSS: CASCADING STYLE SHEETS

MODULO 8

CSS

- È il linguaggio standard per la stilizzazione di documenti HTML
- Assolve al compito di informare il browser circa l'aspetto da conferire ai vari tag presenti nel documento.
- Attraverso i CSS è possibile definire colori, dimensioni, sfondi e quant'altro riguarda l'aspetto visuale degli elementi presenti nella pagina.
- I fogli di stili possono essere di 3 tipologie, in base al contesto in cui essi vengono utilizzati:
 - **INLINE:** o in linea. Si utilizza l'attributo style all'interno di un tag HTML

```
<p style="color: #FF0000;">Testo in colore rosso</p>
```
 - **INTERNI:** vengono anche detti incorporati. Tutte le proprietà vengono trasferite nel tag `<style>` interno al tag `<head>`
 - **ESTERNI:** viene creato un nuovo file con estensione css che deve essere collegato alla pagina HTML

CSS - ESEMPIO

- INLINE: o in linea. Si utilizza l'attributo style all'interno di un tag HTML

```
<p style="color: #FF0000;">Testo in colore rosso</p>
```

- INTERNI: vengono anche detti incorporati. Tutte le proprietà vengono trasferite nel tag `<style>` interno al tag `<head>`

- ```
<html>
 <head> ...
 <style type="text/css">
 p {color: #FF0000;}
 </style>
 </head>
 <body> ...
 <p>Testo in colore rosso</p>
 </body>
</html>
```

# CSS - ESEMPIO

- INLINE: o in linea. Si utilizza l'attributo style all'interno di un tag HTML
- ESTERNI: viene creato un nuovo file con estensione css che deve essere collegato alla pagina HTML

```
<p style="color: #FF0000;">Testo in colore rosso</p>
```

```
<html>
 <head> ...
 <link rel="stylesheet" type="text/css" href="css/style.css">
 </head>
 <body> ...
 <p>Testo in colore rosso</p>
 </body>
</html>
```

Il file style.css deve contenere il codice:

```
p { color: #FF0000; }
```

# CSS - SINTASSI

- La sintassi la conosciamo già, avendola vista con gli style inline:



- Si utilizza il nome del tag, seguito da una parentesi graffa aperta ed una chiusa
- All'interno si mettono tutte le proprietà con i relativi valori associati separati da ;

# CSS – EREDITARIETA'

- Se un elemento è contenuto all'interno di un altro elemento questo ne eredita le sue proprietà.
- Vediamo un esempio direttamente all'interno dell'editor: `css-ereditarieta.html`
- Un paragrafo essendo di fatto contenuto all'interno di `body` ne eredita le sue proprietà, a meno che queste non vengano cambiate ridefinendo il paragrafo stesso.
- Modifichiamo il file sopra inserendo una nuova proprietà per il paragrafo, ad esempio cambiando colore

# CSS – COMMENTI

- Nei file CSS i commenti si inseriscono utilizzando la seguente sintassi:

```
/* commento qui */
```

# CSS – SELETTORE UNIVERSALE

- Il selettore universale serve a selezionare tutti gli elementi di un documento. Si esprime con il carattere `*`.

```
* {color: red;}
```

# CSS – SELETTORE DI TIPO

- I selettori di tipo sono quelli che abbiamo fin qui visto, ovvero rappresentano i tag HTML.
- Possiamo perciò modificare lo stile di qualsiasi tag HTML scrivendo:

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

```
p {color: green}
```

```
li {color: blue}
```

# CSS – SELETTORE DI CLASSE

- Ogni elemento HTML può essere associato a una o più classe
- L'attributo che si utilizza è:

`class="classe1 classe2 classe3 ... classeN"`

- Vediamo un esempio:

HTML            `<p class="testorosso">....</p>`

CSS             `.testorosso {color: red;}`

- Una classe può essere assegnata anche a elementi di diversa natura:

`<p class="testorosso">....</p>`

`<div class="testorosso">....</div>`

`<ol class="testorosso">....</ol>`

# CSS – SELETTORE DI ID

- Ogni elemento HTML può essere associato ad un id univoco
- L'attributo che si utilizza è:

`id="identificatore"`

- Vediamo un esempio:

HTML            `<p id="testorosso">....</p>`

CSS             `#testorosso {color: red;}`

# CSS – RAGGRUPPARE I SELETTORI

- Nei CSS è possibile raggruppare selettori differenti a cui si vuole assegnare le stesse proprietà.
- Ogni selettore va separato da una virgola!!!!
- Vediamo un esempio:

```
h1, h2, h3 {color: red;}
```

```
h4, h5, h6 {color: blue;}
```

# CSS – SELETTORI

- Attraverso i CSS possiamo mischiare diversi selettori per applicare le proprietà ad elementi in maniera più restrittiva.

```
nome-tag.nome-classe {proprietà: valore;}
```

- Vediamo un esempio:

HTML

```
<h1 class="testorosso">Titolo</h1>
```

```
<p class="testorosso">Paragrafo</p>
```

CSS

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

# CSS – ESERCIZIO

- Creare una pagina web con nome `css-prova.html`
- Creare un foglio di stile con nome `css-style.css` e lo salviamo nella cartella `css`
- Collegare il foglio di stile appena creato nella pagina web di prima
- Nel foglio di stile aggiungere le proprietà per definire il colore di ogni paragrafo, di ogni titolo e lo sfondo del `body`:
  - `h1, h3, h5` di colore rosso;
  - `h2, h4, h6` di colore fuxia;
  - `p` di colore blue;
  - Sfondo del `body` di colore grigio chiaro
- Aggiungere un titolo `h1` (Esercitazione con gli stili), uno `h2` (Proviamo i colori) e il paragrafo (In questo paragrafo giochiamo con la parola "corso")
- Fare in modo che nella parola "corso", le lettere pari abbiano dimensione doppia delle dispari. (Suggerimento: usiamo gli `span`)

# CSS – ESERCIZIO

- Inserire una linea orizzontale
- Aggiungere un nuovo paragrafo con classe "grassetto"
- Inserire un paragrafo con id "testo\_personalizzato"
- Inserire un paragrafo con classe "grassetto" e id "testo\_personalizzato"
- Aggiungere la classe "grassetto" e l'id "testo\_personalizzato" al file css
  - grassetto deve mostrare il testo in grassetto, utilizziamo font-weight: bold;
  - testo\_personalizzato deve avere un colore di sfondo, un colore del testo e una dimensione differente da quello standard
- Inserire un testo con un link avente classe "raddoppia"
- Raddoppia deve raddoppiare le dimensioni del testo solo quando si passa con il mouse sopra al link

# CSS – ESERCIZIO

- Creare un nuovo file esempio-css.html
- Aggiungere un DIV e creiamo la regola corrispondente
  - Aggiungere il bordo => border
  - Mettere un colore di sfondo
  - Impostare una larghezza del 100%
  - Impostare un margine inferiore di 50px
- Dentro il DIV inserire un elenco puntato con le voci di menu
  - Homepage, Pagina 1, Pagina 2, Pagina 3
- Inserire 2 DIV affiancati
  - Per entrambi i DIV inserire il bordo e un colore (diversi)
  - Nel primo mettere un testo e impostare la larghezza al 75%
  - Nel secondo inserire un'immagine e impostare la larghezza del DIV al 20%

# CSS – MENU ORIZZONTALE STANDARD

- Per creare un menu orizzontale occorre definire una lista puntata ed applicare le seguenti regole di CSS.

```
ul {
 list-style-type: none;
 overflow: hidden;
}
```

```
li {
 margin-right: 20px;
 float: left;
}
```

# CSS – SELETTORI CON CLASSI MULTIPLE

- Attraverso i CSS possiamo combinare diversi selettori per applicare le proprietà ad elementi in maniera più restrittiva.

```
nometag.classe1.classe2 {proprietà: valore;}
```

- Vediamo un esempio:

HTML            <h1 class="testorosso sottolineato">Titolo</h1>

CSS             .testorosso {color:red;}

                  .testorosso.sottolineato {color: red; text-decoration:underline;}

# CSS – SELETTORI

- Attraverso i CSS possiamo combinare diversi selettori per applicare le proprietà ad elementi in maniera più restrittiva.

```
nometag#nomeid {proprietà: valore;}
```

- Vediamo un esempio:

HTML            <h1 id="testorosso">Titolo</h1>

CSS             h1#testorosso {color:red;}

# CSS – ESERCIZIO

- Riprendere l'ultimo esempio ed applicare gli stili necessari affinché:
- Il menu sia fissato in alto: scorrendo la pagina il menu deve essere sempre visibile;
- Inserire un altro DIV in fondo alla pagina, inserendo le informazioni del footer:
  - larghezza del 100%;
  - Margin top di 50px;
  - Colore del testo e colore di sfondo differente dal resto;
  - Il footer dovrà rispecchiare la seguente struttura:

DIV con classe "footer"

DIV 1 "footer1" (33%)

DIV 2 "footer2" (33%)

DIV 3 "footer3" (33%)

/DIV

# CSS – ESERCIZIO

- Nel DIV 1 inserire il logo e l'intestazione del sito
- Nel DIV 2 inserire il tag span con dentro scritto "Torna in Alto"
- Nel DIV 3 inserire i crediti relativi al creatore del sito Web
- Creare delle regole CSS che:
  - Nel primo DIV:
    - Allinei a destra il logo ed imposti un'altezza pari a 40px;
    - Scriva in colore #FF6905 il testo relativo all'intestazione del sito
  - Nel secondo DIV:
    - Scriva nel colore #05F0FF "Torna in Alto"
  - Nel terzo DIV:
    - Scriva in corsivo e sottolineato

# CSS – SELETTORI DI RELAZIONE

- Una categoria fondamentale di selettori CSS è rappresentata dai cosiddetti combinatori (detti anche selettori di relazione). Hanno la funzione di mettere in relazione elementi presenti all'interno dell'albero del documento.

• Selettore	Simbolo
• Selettore di discendenti	(spazio)
• Selettore di figli	>
• Selettore di fratelli adiacenti	+
• Selettore generale di fratelli	~

# CSS – SELETTORI DI RELAZIONE

- Selettore di discendenti (spazio)
- Il selettore di discendenti è quello più utilizzato dei quattro.
- Il combinatore seleziona un elemento che è discendente di un altro elemento.

```
div#container p {color: red}
```

- Il codice appena visto serve ad assegnare lo stile solo ai paragrafi contenuti nel div#container, ovvero ai paragrafi discendenti del div con id container.

# CSS – SELETTORI DI RELAZIONE

- Selettore di figli >
- Il selettore di figli consente di selezionare un elemento che è figlio diretto dell'elemento padre.
- Questo selettore è solo in apparenza simile al selettore di discendenti.
- La differenza sta nella relazione di discendenza tra gli elementi, che in questo caso deve essere di primo livello.

```
body > p {color: red}
```

# CSS – SELETTORI DI RELAZIONE

- Selettore di fratelli adiacenti +
- Il selettore di fratelli adiacenti serve ad assegnare le regole CSS agli elementi che si trovano allo stesso livello di un altro elemento.
- In pratica, questo tipo di selettore consente di assegnare uno stile all'elemento fratello immediatamente adiacente.

```
h1 + h2 {color: red; text-decoration: underline}
```

# CSS – SELETTORI DI RELAZIONE

- Selettore generale di fratelli ~
- L'ultimo combinatore (~) è una generalizzazione di quello visto in precedenza. Esso assegna uno stile a tutti gli elementi che sono fratelli.

```
h1 ~ h2 {color: red; text-decoration: underline}
```

# CSS – SELETTORI DI PSEUDOCCLASSI

- Una pseudo-classe non definisce la presentazione di un elemento ma di un particolare stato di quest'ultimo.
- In buona sostanza, grazie alle pseudo-classi possiamo impostare uno stile per un elemento al verificarsi di certe condizioni.
  
- `:link`                    aspetto di partenza
- `:visited`                aspetto una volta visitato
- `:hover`                    aspetto quando passiamo col mouse
- `:active`                    aspetto quando viene attivato (click del mouse)
- `:focus`                    aspetto quando si riceve il focus
- `:first-child`              aspetto del primo figlio di un elemento

# CSS – SELETTORI DI PSEUDOCCLASSI

- Per esempio, per cambiare colore ai link occorre:

```
<style>
 a:link {color:green; background-color:transparent;
 text-decoration:none}
 a:visited {color:pink; background-color:transparent;
 text-decoration:none}
 a:hover {color:red; background-color:transparent;
 text-decoration:underline}
 a:active {color:yellow; background-color:transparent;
 text-decoration:underline}
</style>
```

# CSS – SELETTORI DI PSEUDOELEMENTI

- Vi sono parti in un documento HTML che non sono rappresentate da nessun elemento in particolare, ma che è possibile comunque modificare e formattare secondo i propri desideri con i CSS grazie ai cosiddetti pseudo-elementi.
- `::first-letter`      aspetto della singola lettera iniziale
- `::first-line`      aspetto della prima riga
- `::before`      contenuto prima dell'elemento
- `::after`      contenuto dopo l'elemento
- `::selection`      contenuto selezionato

# CSS – SELETTORI DI ATTRIBUTI

- I selettori di attributo consentono di selezionare gli elementi all'interno di una pagina in base ai loro attributi e assegnare così lo stile desiderato. È possibile quindi identificare gli elementi di una pagina senza essere costretti ad assegnare loro una classe o un id.
- E[attribute]                      presenza attributo
- E[attribute=value]              attributo con valore esatto
- E[attribute~=value]            attributo con valore in una lista con spazi
- E[attribute|=value]            attributo con valore in una lista con trattini
- E[attribute^=value]            attributo con valore di inizio
- E[attribute\$=value]            attributo con valore di fine
- E[attribute\*=value]            attributo con valore ovunque

# CSS – CONFLITTI

- Per uno stesso elemento possiamo definire delle proprietà attraverso i selettori di tipo, di classe, di id, ereditato, ecc...
- Occorre, perciò, definire un ordine, una priorità:
  1. Selettore di ID
  2. Selettore di CLASSE
  3. Selettore di TIPO
  4. Selettore EREDITATO
- Esiste un modo per evitare i conflitti? Si utilizza la parola chiave **!important**.