Basi di CSS

Storia di CSS

CSS nasce nel 1996 per mettere un freno alla nascita di tag proprietari di ogni browser.

- La standardizzazione e lo sviluppo è gestito dal W3C
- Nel 1998 viene pubblicata la versione 2.0
- Nel 2011 viene pubblicata la versione 2.1
- Nel 2014 viene pubblicata la versione 3.0

E’ stato dimostrato che HTML5 + CSS3 è un linguaggio Turing completo

Regole CSS

Il codice CSS da utilizzare per formattare il documento HTML è composto di regole che verranno applicate dal browser DOPO aver interpretato e visualizzato il codice HTML.

Le regole potranno pertanto fare riferimento ai tag e agli attributi presenti nella pagina.

Ogni regole è formata da due parti:

- Un selettore che identifica su quale porzione di HTML applicare lo stile
- Una, o più, dichiarazioni di stile che specificano le proprietà grafiche da assegnare a ciò che è stato selezionato dal selettore

Le proprietà associate ad un selettore vanno raccolte dentro parentesi graffe.

Regole CSS

La specifica di una proprietà grafica avviene specificando la proprietà e il valore da associare, divisi da : e terminati da ;

```
Selettore
{
    proprieta1: valore1;
    proprieta2: valore2;
}
```

Regole CSS

Le regole css da utilizzare possono essere specificate in tre modi:

- Nella sezione <head> fornendo l'indirizzo di un file css esterno:

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

- Nella sezione <head> scrivendo direttamente il codice css all'interno del tag <style>

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

- Come attributo di qualsiasi tag HTML:

```
<tag style="dichiarazioni css"></style>
```

L'ultima modalità è più difficile da manutenere e pertanto è sconsigliata!

Selettori CSS – per tag (tipo)

Il selettore più semplice seleziona TUTTI i tag di uno stesso tipo presenti nella pagina. Si specifica riportando il nome del tag:

```
body
{
}


p
{
}
```

Selettori CSS – per classe

E' possibile associare ad ogni tag una classe sulla quale specificare stili css. Più tag, anche diversi tra di loro, possono condividere la stessa classe.

La classe si specifica con l'attributo class e il selettore prevede di specificare il nome della classe preceduto dal punto.

Un tag può avere più classi. Vanno specificate separandole con lo spazio.

```
<p class="paragrafo1">…</p>


.paragrafo1
{
}
```

Selettori CSS – per classe

E' possibile selezionare tutti i tag che contengono, contemporaneamente, due o più classi.

Il selettore deve riportare tutte le classi, una dopo l'altra, senza spazi.

```
<p class="paragrafo1 classe2">…</p>
```

```
.classe2.paragrafo1
{
}
```

Selettori CSS – per tag e classe

E' possibile combinare la selezione per tag con la selezione per classe, per poter applicare stili a tutti i tag di un certo tipo che hanno una determinata classe.

Il selettore deve riportare, uno dopo l'altro, il nome del tag e il nome della classe, senza spazi.

```
<p class="paragrafo1 classe2">…</p>
```

```
p.classe2
{
}
```

Selettori CSS – per identificatore

E' possibile associare ad ogni tag un identificatore univoco al quale specificare stili css. All'interno di una stessa pagina NON possono esistere più tag con lo stesso id (sebbene i browser non generino errori).

L'identificatore si specifica con l'attributo id e il selettore prevede di specificare l'identificatore preceduto dal cancelletto.

```
<p id="paragrafo1">…</p>
```

```
#paragrafo1
{
}
```

Selettori CSS – per attributo

E' possibile selezionare in base all'esistenza o al valore di un attributo. E' possibile limitare la ricerca a tag di un certo tipo o estenderla a tutti i tag. Alcuni esempi:

- `a[target]` : Tutti i tag a che presentano l'attributo target (indipendentemente dal valore)
- `a[target="_blank"]` : Tutti i tag a che presentano l'attributo target con valore _blank
- `[class="prezzo"]` : Tutti i tag che hanno la classe prezzo
- `[class*="prezzo"]` : Tutti i tag che hanno una classe il cui nome contiene prezzo
- `[class^="prezzo"]` : Tutti i tag che hanno una classe il cui nome inizia con prezzo
- `[class$="prezzo"]` : Tutti i tag che hanno una classe il cui nome finisce con prezzo
- `[class^="prezzo"][class$="netto"]` : Tutti i tag che hanno una classe il cui nome inizia con prezzo e finisce con netto

Selettori CSS – per gerarchia

Per effettuare selezioni più accurate, all'interno di strutture HTML complesse, è possibile utilizzare selettori che tengono conto della posizione relativa dei tag. Le possibilità sono molteplici, vediamone alcune:

- Selezione di tutti i tag contenuti in un altro tag, con qualsiasi livello di "parentela": Si elencano tag contenitore e tag contenuto separati da spazio

```
p a
{
}
```

- Selezione di tutte le classi contenute in un'altra classe, con qualsiasi livello di "parentela": Si elencano classe contenitore e classe contenuto separati da spazio

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

Selettori CSS – per gerarchia

- Selezione di tutti i tag contenuti immediatamente figli diretti ad un altro tag: Si elencano tag precedente e tag successivo separati da >

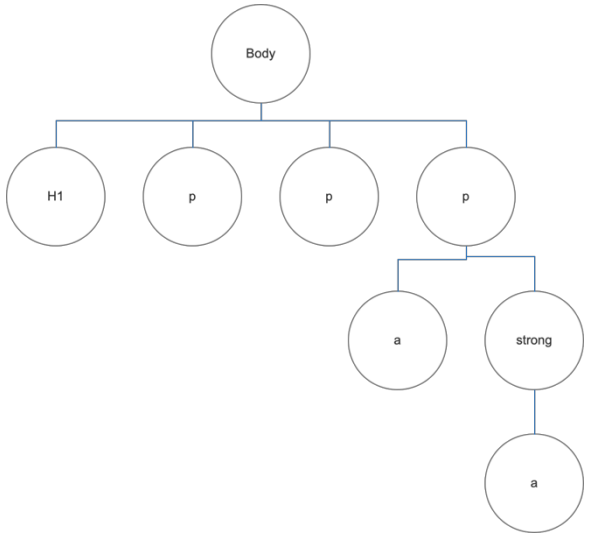
```
p > a    { }
```

- Selezione del tag immediatamente successivo ad un altro tag (solo primo fratello): Si elencano tag precedente e tag successivo separati da +

```
h1 + p   { }
```

- Selezione di tutti i tag fratelli di un altro tag, che appaiono dopo di esso: Si elencano tag precedente e tag successivo separati da ~

```
h1 ~ p   { }
```



Selettori CSS – pseudo selettori

Con i pseudo selettori possiamo identificare i figli di un tag, senza specificarne tipo, classe o id. Alcuni esempi:

- `article:first-child` : Primo elemento figlio del tag article
- `article:first-of-type` : Primo elemento di ogni tipo figlio del tag article
- `article p:first-of-type` : Primo paragrafo figlio del tag article
- `article:last-child` : Ultimo elemento figlio del tag article
- `article:last-of-type` : Ultimo elemento di ogni tipo figlio del tag article
- `article p:last-of-type` : Ultimo paragrafo figlio del tag article
- `ul li:nth-child(2)` : Secondo figlio di tipo li del tag ul
- `ul li:nth-child(odd)` : Figli dispari di tipo li del tag ul
- `ul li:nth-child(even)` : Figli pari di tipo li del tag ul

Selettori CSS – pseudo selettori

Con i pseudo selettori possiamo identificare i figli di un tag, senza specificarne tipo, classe o id. Alcuni esempi:

- `a:visited` : Tag a nello stato di visitato
- `a:link` : Tag a nello stato di non visitato
- `a:hover` : Tag a quando il mouse ci passa sopta
- `a:focus` : Tag a quando riceve il focus

Selettori CSS – pseudo selettori

Con i pseudo selettori anche identificare una parte del contenuto di un tag. Notare che in questo caso si usa il doppio carattere :

Alcuni esempi:

- `p::first-letter` : Prima lettera del paragrafo
- `p::first-line` : Prima line del paragrafo
- `p::selection` : Parte selezionata di un paragrafo
- `p::before` : Parte che precede il contenuto del paragrafo
- `p::after` : Parte che segue il contenuto del paragrafo

Selettori CSS – ottimizzare la scrittura di stili

Talvolta lo stesso stile deve essere applicato a selettori differenti e di vario tipo.

Per ottimizzare la scrittura delle regole e per agevolarne la manutenzione è possibile scrivere una unica regola associata a selettori multipli.

I vari selettori vanno divisi da virgola:

```
h1, h2, h3
{
}
```

Specificità dei selettori CSS

Talvolta lo stesso elemento è selezionato da più regole CSS. Se tali regole impostano la stessa proprietà si genera un conflitto che necessita di risoluzione.

A fronte di tag selezionati da più selettori e a fronte dell'impostazione della stessa proprietà, CSS stabilisce un ordine di priorità nel seguente modo:

1. La massima priorità è data a selettori per identificatore (per esempio #idElemento)
2. La priorità intermedia è data a selettori per classe (per esempio .nomeClasse)
3. La priorità più bassa è data a selettori per elemento (per esempio body)

Se ci sono più regole con la stessa macro-priorità la logica si complica (ne parleremo in seguito) ma l'obiettivo è sempre lo stesso: determinare la priorità.

Qualora ci fossero più selettori con la priorità massima questi vengono applicati tutti in modo sequenziale.

Specificità dei selettori CSS - Esempio

Quale stile assumerà il titolo H3 riportato nell'esempio?

```
<H3 id="MioTitolo" class="MiaClasseTitolo">Hello World</H3>


.MioTitolo { color:red; }

#MioTitolo { color:blue; font-variant: small-caps; }

#MioTitolo { color:black;}

H3 { color:yellow; }
```

Algoritmo applicazione CSS

Ricapitolando, il browser applica gli stili CSS utilizzando il seguente algoritmo (semplificato):

1. Per ogni tag il browser determina le regole che lo selezionano.
2. Se ce ne è una sola, viene applicato quello stile.
3. Se ce ne sono più di una viene calcolata la priorità di ogni regola.
4. Viene applicata la regola di priorità massima.
    1. In caso di più regole con priorità massima vengono applicati TUTTI gli stili, partendo da quello che è stato definito per primo.

Possiamo considerare la priorità come un numero binario composto da tre bit. Il primo bit, quello più significativo, vale 1 se la regola utilizza selettori per id. Il secondo bit vale 1 se la regola utilizza selettori per classe. Il terzo bit vale 1 se la regola utilizza selettori per tipo.

Ereditarietà delle regole CSS

La maggior parte degli stili CSS definiti per un elemento valgono anche per tutti i suoi figli, a meno che non vengano ridefiniti.

Gli elementi HTML pertanto ereditano lo stile CSS dal proprio genitore, ed eventualmente lo sovrascrivono con regole specifiche.

L'ereditarietà non vale per alcune proprietà specifiche come, per esempio, quelle utilizzate per definire bordi e margini.

```
<body>
        <H1>Titolo</H1>
        <H2>Sottotitolo</H2>
</body>
```

```
body { color:red }
H2   { color:black }
```

Quali colori assumeranno i due titoli?

Normalizzare il CSS

L'ereditarietà degli stili CSS comporta che il risultato finale DIPENDA dai default che il browser implementa. E' infatti il default che verrà esteso dagli stili.

Browser differenti potrebbero avere default differenti, con il risultato di avere rappresentazioni grafiche diverse.

Per ovviare al problema si può importare un css in grado di "normalizzare" i default dei browser rendendoli tutti uguali.

Un css molto utilizzato è normalize.css (https://necolas.github.io/normalize.css/)

ATTENZIONE: Il css di normalizzazione va inserito PRIMA del nostro CSS

Principali stili CSS

Gli stili impostabili via CSS sono molto numerosi. Per una trattazione completa ed approfondita si rimanda alla documentazione ufficiale (https://www.w3.org/TR/css-2021/) o ad altri siti che riportano le stesse informazioni in modo più "user friendly":

- https://developer.mozilla.org/en-US/docs/Web/CSS/Reference?retiredLocale=it
- https://cssreference.io
- https://www.w3schools.com/cssref/default.asp

In questa prima parte del modulo introdurremo solamente gli stili principali, dividendoli in:

- Stili tipografici
- Stili per sfondi
- Stili per liste e tabelle

Stili tipografici

Stile	Significato
`color`	Specifica il colore del testo
`font-family`	Specifica la famiglia di font che si vuole utilizzare
`font-weight`	Specifica il peso del carattere, ovvero lo spessore del suo tratto
`font-size`	Specifica la dimensione del carattere, ovvero a sua altezza
`font-style`	Specifica se il carattere è normale o italico
`font`	Proprietà "scorciatoia" che permette di specificare in una sola direttiva molteplici stili relativi al font
`text-align`	Consente di allineare orizzontalmente un testo
`line-height`	Specifica la dimensione in altezza della linea di testo, generando più o meno spazio tra le righe
`text-transform`	Specifica se il testo deve essere trasformato in lettere maiuscole o minscole
`text-decoration`	Specifica stili quali il sottolineato e il barrato
`text-shadow`	Specifica un'ombra nel testo

Stili per sfondi

Stile	Significato
`background-color`	Specifica il colore dello sfondo di un elemento
`background-image`	Specifica una immagine da applicare come sfondo di un elemento
`background-repeat`	Specifica se e come ripetere l'immagine di sfondo
`background-position`	Specifica come posizionare l'immagine di sfondo
`background-size`	Specifica la dimensione dell'immagine di sfondo
`background-attachment`	Specifica se l'immagine è fissa o si muove con la pagina
`background`	Proprietà "scorciatoia" che permette di specificare in una sola direttiva molteplici stili relativi allo sfondo

Stili per liste e tabelle

Stile	Significato
`list-style-type`	Specifica il tipo di marcatore da usare per la lista
`list-style-image`	Specifica una immagine da usare come marcatore
`list-style-position`	Specifica la posizione del testo rispetto al marcatore
`list-style`	Proprietà "scorciatoia" che permette di specificare in una sola direttiva molteplici stili relativi alle liste
`border-collapse`	Specifica la modalità di visualizzazione dei bordi di una cella
`border-spacing`	Specifica la distanza tra i bordi in caso di border-collapse: separate
`border`	Proprietà "scorciatoia" che permette di specificare in una sola direttiva molteplici stili relativi ai bordi (li vedremo anche più avanti)