

Appunti modulo 1: introduzione HTML / CSS - 30/07/2020

1) INTRODUZIONE HTML

L'HTML è il linguaggio che serve per programmare siti web (N.B: La figura del programmatore web è molto presente nell'industria, tuttavia i siti web moderni si compongono di decine di migliaia di linee di codice complesso. In queste pagine ci occuperemo della parte *front-end* dei siti ovvero della parte grafica e dell'impostazione del layout del sito; non ci occuperemo invece della parte *back-end* ovvero tutto quello che avviene "sotto il cofano").

Per iniziare si crea un file (tasto destro -> nuovo -> documento di testo), si assegna un nome e si cambia l'estensione da *.txt* a *.html* (l'estensione è la particella del nome preceduta dal punto che indica il tipo di file; ".html" indica che si tratta di un sito).

Aperto il file con un editor di testo (es. trascinando il file appena creato su *sublime text*) visualizziamo il codice (che andremo in seguito a scrivere) del sito. Aperto invece lo stesso file con un browser (facendo doppio click semplicemente, o tasto destro -> apri con -> Microsoft Edge) visualizziamo il sito (e cioè il risultato finale) rappresentato dal codice scritto.

Con queste due finestre aperte possiamo quindi, scrivere il codice, salvare, aggiornare la pagina web e vedere i risultati in tempo reale.

Per programmare in HTML si usano i **tag** (ad esempio `<.....>`).

- La maggioranza dei comandi hanno una coppia di tag, uno di apertura e uno di chiusura. Ciò che scriviamo fra il tag di apertura e quello di chiusura si chiama *argomento* ed è la parte di codice su cui avrà effetto il comando indicato dai tag in questione.
- Alcuni comandi invece (ad. Esempio quello per le immagini) si compongono di un solo tag il cui argomento o valore è assegnato all'interno dello stesso (es ``).

Un sito HTML si compone di due parti essenziali: un'intestazione **head** e un corpo **body**.

- Nell'intestazione *head* possiamo definire il titolo della pagina con `<title>...</title>` e possiamo inoltre scrivere i *meta-tag* che sono dei tag speciali che posti nell'intestazione permettono di definire proprietà del documento. I meta tag non producono alcun tipo di effetto grafico nella visualizzazione della pagina.
 - o Ad esempio possiamo scrivere nell' *head*: `<meta charset = "utf-8">` che indica che stiamo lavorando su una versione recente del linguaggio HTML. (Tale tag è opzionale).
- Nel corpo invece, definiamo tutta la parte grafica del sito (es. paragrafi, testi, immagini).

Vediamo alcuni tag, il loro comando e posizionamento.

Tag di apertura	Tag di chiusura (se presente)	Comando	Dove va posizionato?
Struttura			
<code><html></code>	<code></html></code>	Indica che stiamo facendo un sito.	All'esterno di tutto
<code><head></code>	<code></head></code>	Indica l'intestazione	All'interno di <code><html></code>
<code><title></code>	<code></title></code>	Indica il titolo di pagina	All'interno di <code><head></code>
<code><body></code>	<code></body></code>	Indica il corpo	All'interno di <code><html></code>
Collegamenti e immagini			
<code></code>	<code></code>	Tra le virgolette scriviamo un indirizzo, Tra i due tag ci sarà l'elemento (ad esempio un testo) che verrà collegato all'indirizzo.	Nel body
<code></code>		Visualizza l'immagine all'indirizzo impostato.	Nel body
Testo			
<code><p></code>	<code></p></code>	Il testo all'interno sarà un paragrafo.	Nel body
<code><h1></code> (da 1 a 6)	<code></h1></code>	Titolo: h1 è il più grande.	Nel body... prima di un paragrafo

<code><center></code>	<code></center></code>	Il testo all'interno sarà centrato.	Su un elemento testuale.
<code></code>	<code></code>	Neretto	Su un elemento testuale.
<code><i></code>	<code></i></code>	Corsivo	Su un elemento testuale.
<code><u></code>	<code></u></code>	Sottolineato	Su un elemento testuale.
<code>
</code>		Per andare a capo	Nel body.
<code><hr></code>		Crea una linea orizzontale	Nel body.
Liste			
<code></code>	<code></code>	Crea una lista non numerata.	Nel body.
<code></code>	<code></code>	Crea una lista numerata.	Nel body.
<code></code>	<code></code>	Aggiunge un elemento nella lista.	In una lista <code></code> o <code></code>

2) INTRODUZIONE CSS

Per modificare lo stile del sito (colori, sfondo, ecc.), usiamo i comandi CSS.

All'interno dell'intestazione, introduciamo il tag `<style> </style>` al cui interno scriveremo i comandi CSS per modificare i vari elementi grafici.

Il generico comando CSS è formattato come segue:

```
selezionatore {
    comando;
}
```

Il selezionatore è una parte o elemento del sito (ad esempio un paragrafo, un titolo o il corpo intero) su cui avrà effetto il comando.

N.B. avere cura di ricordarsi il ";" a fine di ogni comando.

Vediamo due comandi di base:

- **color: valore**
- **background-color: valore**

Se il colore è tra quelli basilari (come green, red, yellow ecc..) possiamo indicarlo scrivendolo (es. `color: red`), altrimenti dobbiamo specificarlo scrivendo: `color: rgb(x, y, z)` in cui al posto di x, y, z scriveremo degli opportuni valori da 0 a 255 che individuano il colore nella palette rgb.

Esempio:

```
body{
    background-color: rgb(0,0,200);
}
```

Indica che il body del sito, (e quindi il sito intero) avrà uno sfondo identificato dal colore rgb(0,0,200).

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

Indica che tutti (!) i titoli h1 avranno colore rosso.

ID

Come facciamo però se vogliamo colorare di rosso solo un particolare titolo h1 e non tutti quanti??

- Abbiamo bisogno di identificare i vari tag con un id.
Per esempio se nel codice troviamo:
`<h1> Titolo bellissimo </h1>`
Lo possiamo identificare dagli altri h1 scrivendo:
`<h1 id="titolo-principale"> Titolo bellissimo </h1>`

Tra le doppie virgolette abbiamo inserito un nome identificativo del tag.

Il nome identificativo è arbitrario, con la regola che non possiamo inserire spazi bianchi e che ci può essere solo un elemento con lo stesso nome.

Adesso per colorare di rosso solo il tag "titolo-principale" scriveremo:

```
#titolo-principale{  
    color: red;  
}
```

Il "#" nel selezionatore indica che stiamo indicando un'identità.

CLASSI

Se volessimo colorare solo 6 (ad esempio) titoli h1 su un totale di 10 titoli presenti nel sito, potremmo assegnare a ciascuno degli h1 che vogliamo colorare un id, ma questo risulterebbe scomodo, in quanto per ogni id, dovremmo scrivere un nuovo comando CSS del tipo sopra.

Introduciamo quindi le classi tramite il quale possiamo identificare una serie di tag, ad esempio:

```
<p class="paragrafi-gialli"> paragrafo 1 </p>  
<p class="paragrafi-gialli"> paragrafo 2 </p>  
<p class="paragrafi-verdi"> bellissimo paragrafo </p>  
<p class="paragrafi-verdi"> bla bla bla bla bla bla bla </p>
```

Abbiamo identificato i primi due paragrafi con una classe e i secondi 2 con un'altra.

Adesso per colorarli possiamo scrivere:

```
.paragrafi-gialli{  
    color: yellow;  
}  
.paragrafi-verdi{  
    color: green;  
}
```

Il "." nel selezionatore indica che stiamo indicando una classe.

3) ALTRI TAG HTML

LE TABELLE

Vediamo come inserire delle tabelle.

Il tag che indica la tabella è **<table>** **</table>**

all'interno di cui dovremo specificare il contenuto di ciascuna riga e in ciascuna riga dobbiamo indicare il contenuto delle celle.

- **<thead>** **</thead>** indica che le righe che seguiranno faranno parte dell'intestazione della tabella (saranno in grassetto). (Va posizionato all'interno di <table>).
- **<tbody>** **</tbody>** indica che le righe che seguiranno faranno parte del corpo della tabella. (Va posizionato all'interno di <table>).
- **<tr>** **</tr>** crea una nuova riga. (va posizionato all'interno di <thead> o <tbody>).
- **<th>** **</th>** crea una nuova cella nell'intestazione della tabella. (va posizionato all'interno di <tr> e <thead>).
- **<td>** **</td>** crea una nuova cella nel corpo della tabella. (va posizionato all'interno di <tr> e <tbody>).

Anche con le tabelle possiamo analogamente usare i comandi CSS per modificarne lo stile.

I COMMENTI

Come in altri linguaggi di programmazione, anche in HTML possiamo commentare.

I commenti sono delle porzioni di testo inserite nel codice che non hanno valore per l'esecuzione del programma ma, hanno lo scopo di aiutare il programmatore a giostrarsi tra le righe di codice.

Per commentare scriviamo `<!-- Commento a caso -->`. Il testo del commento è ovviamente arbitrario ad eccezione della struttura del tag `<!-- -->`. Possiamo inserire commenti in qualsiasi parte di codice HTML.

4) PROPRIETA' TESTUALI CSS

FONT FAMILY

Per cambiare la famiglia di font dei testi introduciamo un nuovo comando CSS:

font-family: valore;

Nel campo del valore possiamo scrivere una delle seguenti font-family:

- serif
- sans-serif
- cursive
- fantasy
- monospace

Il browser alla lettura del comando, mostrerà all'utente il font predefinito (sul pc dell'utente) per quella famiglia di font.

Possiamo anche specificare un determinato font stando attenti a scriverlo tra doppie virgolette, ad esempio:

- "Helvetica"
- "Arial"
- ...

Non è detto che il pc (su cui verrà visualizzato il sito) abbia il font specificato, per cui è utile specificare una font-family che verrà utilizzata in questo caso. Utilizziamo quindi il comando

font-family: "nome font", nomeFamiglia;

FONT SIZE

Per cambiare la dimensione del font dei testi introduciamo un nuovo comando CSS:

font-size: valore;

Selezionando il body e modificando la dimensione del testo, notiamo che i titoli rimangono più grandi rispetto al testo normale.

Ci sono 2 unità di misura principali per il valore:

- *em* È l'unità di misura dei titoli. Indica quante volte più grande rispetto a un testo normale sarà un determinato titolo.
- *px* Si usa per tutti gli altri elementi.

Esempio:

```
body {  
    font-size: 12px;  
}  
h2 {  
    font-size: 1.5em;  
}
```

In questo modo indichiamo che ogni testo del corpo (eccezione per i titoli) avrà dimensione 12px mentre i titoli h2 avranno dimensione = $1,5 * 12px = 18px$.

FONT STYLE

Per cambiare lo stile dei font dei testi in CSS, introduciamo ulteriori nuovi comandi.

font-weight: bold; per rendere il testo in grassetto.

font-style: italic; per rendere il testo in corsivo.

Se dobbiamo cambiare rispettivamente sia lo stile, che la dimensione, che il carattere, possiamo farlo in unico comando:

```
font: nomeStile dimensione_px nomeFont;
```

ad esempio:

```
font: italic 30px "Cooper Black";
```

ALTRE PROPRIETA'

Continuano le proprietà CSS per i testi, vediamo ancora qualcuna:

- *text-decoration: underline;* per sottolineare il testo
- *line-height: valore_px;* per cambiare la spaziatura tra le linee
- *text-align: [left, center, right];* per cambiare l'allineamento del testo

5) IMPAGINAZIONE CSS

GRUPPI CSS

CSS è utilizzabile anche per spostare e impaginare i vari elementi.

Ma come facciamo a spostare vari elementi insieme?

Introduciamo 2 nuovi tag HTML (utili solo se usati in combinazione con comandi CSS):

- ` `

Viene usato per raggruppare elementi testuali.

Esempio:

Se nel body troviamo

```
<p> questo è il paragrafo migliore di questo file </p>
```

Se vogliamo colorare di rosso solo le parole "è il paragrafo migliore" possiamo fare come segue:

```
<p> questo <span id = "par"> è il paragrafo migliore </span> di questo file </p>
```

E nel tag `<style>` possiamo scrivere

```
#par{
```

```
    Color: red;
```

```
}
```

- Se invece vogliamo raggruppare elementi diversi usiamo

```
<div> </div>
```

Per esempio all'interno di div possono esserci titoli h1, paragrafi, liste ecc.

Si usa analogamente a quanto visto con *span*.

Lo *span* crea un elemento "in-line" ovvero il testo raggruppato rimarrà sulla stessa linea, mentre il *div* crea un elemento "block" e aggiunge una nuova linea prima e dopo.

LARGHEZZA, ALTEZZA E OVERFLOW CSS

Possiamo modificare la larghezza e altezza del *div* (e del suo *block*) specificando la misura in px o in % (100% indica che occupa tutto lo spazio disponibile) con il comando:

```
height: valore;
```

```
width: valore;
```

Se impostiamo un valore del *div* troppo piccolo vediamo che il testo fuoriesce dai bordi (se lo sfondo del div è diverso da bianco risulta bene).

Il valore in eccesso è gestito dal comando *overflow*:

```
overflow: visible; indica che la parte in eccesso sarà visibile.
```

```
overflow: hidden; nasconde la parte in eccesso.
```

```
overflow: auto; aggiunge delle barre di scorrimento per visualizzare il testo, se necessario.
```

Analogamente possiamo controllare singolarmente *overflow-x* e *overflow-y* per controllare il contenuto che fuoriesce rispettivamente in orizzontale e verticale.

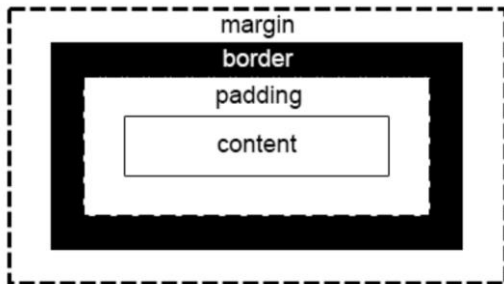
Anche per i le immagini possiamo definire *height* e *weight* analogamente per quanto visto con i div.

RIQUADRI CSS

Ogni elemento che scriviamo, è visto dal browser come un riquadro.

Ogni riquadro ha i seguenti elementi indicati in *the box model*:

The Box Model



Per i margini usiamo i seguenti comandi:

margin-left: valore_px; per il margine di sinistra
margin-right: valore_px; per il margine di destra
margin-top: valore_px; per il margine in alto
margin-bottom: valore_px; per il margine sotto

oppure usando un solo comando:

margin: sopra destra sotto sinistra; (es. *margin: 30px 20px 30px 20px;*)

Per definire i bordi invece usiamo il comando:

border: spessore stile colore; (es. *border: 2px dotted red;*)

Gli stili che possiamo usare sono ad esempio:

dotted, solid, groove, double, ridge, dashed

Analogamente a *margin*, anche per *border* possiamo usare singolarmente *border-top*, *border-left* ecc... specificando sempre le 3 proprietà richieste.

Per aggiungere un riempimento tra il contenuto e il bordo possiamo specificare la proprietà *padding* con il comando:

padding: valore_px;

POSIZIONI

Inizialmente e in assenza di comandi CSS (come quelli che andremo a definire), tutti gli elementi hanno una posizione di default: gli elementi *in-line* sono posizionati da sinistra a destra mentre gli elementi *block* dall'alto al basso.

Vediamo ora come spostare gli elementi in una pagina web:

Usiamo prima uno (e uno solo) tra i 2 comandi:

position: absolute;
position: relative;

e successivamente uno o più comandi tra

top: valore_px;
bottom: valore_px;
left: valore_px;
right: valore_px;

Se abbiamo specificato *position: relative* i valori indicati successivamente sono relativi al posizionamento di default che quindi risulta sfasato di un certo numero di px; se invece abbiamo indicato *position: absolute* i valori successivi indicano una coordinata (assoluta) sullo schermo e quindi il posizionamento di default viene totalmente ignorato.

Se posizioniamo 2 (o più) elementi nella stessa posizione, per esempio con il comando *position: absolute* e specificando in seguito gli stessi valori di posizione, notiamo che gli elementi saranno sovrapposti.

Possiamo decidere l'ordine di sovrapposizione con il comando:

z-index: valore;

in cui specifichiamo nel campo del valore un numero intero.

In questo modo gli elementi con un *z-index* alto staranno sopra a quelli con uno basso.

Un ultimo comando (sempre da abbinare a uno dei 4 precedenti *top, bottom, ...*) è il seguente:

position: fixed;

è analogo a *position: absolute* ma la differenza è che anche scorrendo la pagina con *position: fixed* l'elemento non scorrerà con essa, ma rimarrà sullo schermo esattamente nella posizione che abbiamo definito, mentre tutti gli altri elementi senza questa proprietà si muoveranno normalmente.

ELEMENTI FLOAT

Se abbiamo un'immagine e un testo, ad esempio, è interessante capire come possiamo far avvolgere l'immagine dal testo.

La proprietà *float* ci permette di far fluttuare elementi dentro e intorno ad altri elementi.

Scrivendo:

float: [right, top, left, bottom]; (Esempio: *float: left;*)

Indichiamo che l'elemento fluttuerà a destra, sinistra, ... come specificato.

Generalmente se facciamo fluttuare un *div* dobbiamo assegnare anche una larghezza (inferiore al 100% della pagina) altrimenti il *div* occuperà tutto lo spazio disponibile e non vedremo l'effetto.

6) ALTRI SELETTORI CSS

CLASSI MULTIPLE

Come possiamo fare se per esempio abbiamo la necessità che tutti i paragrafi siano di un determinato colore e solo alcuni abbiano un *font-size: 30px* ? Potremmo creare tanti *id* diversi, ma ciò risulterebbe molto scomodo.

Risolviamo questo problema con le classi multiple:

Possiamo associare più classi ad uno stesso tag, ad esempio:

```
<p class= "titolone petaloso"> bla bla bla bla bla bla bla </p>
```

```
<p class= "titolone"> ciao </p>
```

Scrivendo 2 parole spaziate abbiamo associato al primo tag le 2 classi *titolone* e *petaloso*, mentre nel secondo tag abbiamo associato solo la classe *titolone*.

Per ognuna delle 2 classi possiamo ovviamente specificare regole diverse:

```
.petaloso{
    color: red;
}
.titolone{
    font-size: 30px;
}
```

COMBINARE I SELETTORI DI CLASSI E DI ELEMENTO

Se molti elementi appartengono alla stessa classe, possiamo definire delle regole che si applicano a tutta la classe (come abbiamo già visto), ma possiamo inoltre creare delle regole che si applicano solo ad alcuni tipi di elementi della classe.

Ad esempio:

```
<h2 class= "in-evidenza"> il sistema nervoso </h2>
```

```
<h3 class= "in-evidenza">il sistema nervoso parasimpatico </h3>
```

```
<h3 class= "in-evidenza">il sistema nervoso ortosimpatico </h3>
```

```
.in-evidenza{
    font-family: sans serif;
}
h2.in-evidenza{
    background-color: red;
}
```

In questo caso tutti gli elementi con la classe *in-evidenza* avranno un carattere *sans serif* mentre gli *h2* che appartengono sempre alla classe *in-evidenza* avranno uno sfondo rosso.

Possiamo cioè scrivere nel selettore, prima del “.” (che precede il nome della classe) un nome di tag. Se il tag specificato appartiene alla classe specificata, allora avrà gli effetti di tale regola.

SELETTORI DISCENDENTI

Per specificare gli elementi su cui la regola CSS avrà effetti possiamo specificare i “tag genitori” rispetto agli elementi che vogliamo selezionare.

Vediamo subito un esempio:

```
.purple { ha effetto su tutti gli elementi della classe purple.
    background-color: purple;
}
p.purple { ha effetto su tutti gli elementi contenuti in p con classe purple.
    color: purple;
}
p.purple { ha effetto su tutti gli elementi p con classe purple.
    color: purple;
}
ul.spacey li strong { ha effetto su tutti gli elementi strong contenuti in li contenuti in ul con classe spacey.
    line-height: 2em;
}
```

```
<ul class="spacey">
    <li> <strong class="purple">Purple</strong> </li>
</ul>
<p>
    Ciao <strong class="purple"> come stai? </strong> stai
</p>
```

Per specificare un tag come “discendenza” aggiungiamo quindi uno spazio dopo aver scritto il nome del tag stesso; questo indicherà che gli elementi selezionati sono all’interno del tag “genitore”. (Corrisponde ad indicare un “percorso”).

RAGGRUPPARE I SELETTORI CSS

Possiamo raggruppare i selettori, separandoli da una virgola; questo è utile se dobbiamo introdurre le stesse regole per 2 o più selettori.

Ad esempio, al posto di

```
h1 {
    font-family: fantasy;
}
h2 {
    font-family: fantasy;
}
```

Possiamo scrivere semplicemente

```
h1, h2 {
    font-family: fantasy;
}
```


PSEUDO-CLASSI

Le pseudo-classi sono indicate dai ":" e si usano per selezionare alcuni elementi in base ad alcune proprietà o ad alcuni stati dinamici (che cambiano nel il tempo).

Vediamo un esempio:

```
a:link {
  color: red;
}
a:visited {
  color: orange;
}
a:hover {
  background-color: rgb(255, 214, 255);
}
a:active {
  background-color: rgb(255, 214, 255);
}
a:focus {
  background-color: rgb(255, 214, 255);
}
```

La "a" iniziale indica il tag selezionato (in questo caso ricordiamo che "a" seleziona i link") seguono i ":" e segue il nome della pseudo-classe.

Vediamo le pseudo-classi per i link:

- `:link` Seleziona tutti i link che l'utente non ha ancora visitato. (valore di default blue)
- `:visited` Seleziona tutti i link che l'utente ha già visitato. (valore di default viola)
- `:hover` Seleziona i link solo per il tempo in cui l'utente ci passa sopra.
- `:active` Seleziona i link per il tempo che intercorre tra il click e il rilascio del mouse sul link stesso.
- `:focus` Seleziona i link quando un elemento riceve il focus ad esempio se stiamo selezionando un link tramite tab da tastiera.

Gli ultimi 3 `hover active e focus` sono simili tra loro e normalmente assegnamo a tutti e 3 lo stesso valore, possiamo quindi raggrupparli:

```
a:active, a:focus, a:hover {
  comando;
}
```

SPECIFICITA' CSS

Il browser ad ogni selezionatore (e quindi alle regole contenuto in esso) associa un punteggio di specificità. Se per esempio ad una stessa parte della pagina abbiamo associato 3 regole con 3 selezionatori diversi in 3 comandi diversi, avrà effetto solo il comando il cui selezionatore risulta essere più specifico. Se 2 comandi hanno la stessa specificità, avrà effetto il comando situato più in basso.

In generale, un `id` è più specifico di una `class` che a sua volta è più specifica di un nome di un tag (elemento).

Vediamo un esempio:

```
p {
  font-family: sans-serif;
  color: orange;
}
.info-paragraph {
  color: blue;
}
#main-paragraph {
  color: green;
}
```

```
<p id="main-paragraph" class="info-paragraph important"> L'oceano pacifico è l'oceano più grande del mondo </p>
```

Il paragrafo avrà colore verde perché `#main-paragraph` è più specifico di `.info` che è più specifico di `p`.