

COS'E' L'HTML?

Il linguaggio utilizzato dalle pagine del World Wild Web, l'**HTML** (HyperText Markup Language), non è un complesso linguaggio di programmazione bensì un "linguaggio di formattazione", ovvero un linguaggio di istruzioni composto di elementi (tag o etichette) che definiscono le caratteristiche di un documento e ne guidano la composizione sullo schermo. Questi elementi e i loro attributi funzionano come indicazioni stilistiche che vengono interpretate dal browser (ad esempio Internet Explorer) e disposte sulla finestra. Tutti i comandi HTML sono facilmente riconoscibili: sono sempre racchiusi tra i segni di minore e maggiore (< e >), e presentano sempre un comando di apertura (come <html>) e quasi sempre uno di chiusura (</html>) contrassegnato dalla presenza del carattere / (slash). Per editare le pagine HTML è sufficiente disporre di un Word Processor qualunque e salvare il file con l'estensione ".htm" oppure ".html". Per chi comincia da zero è consigliabile utilizzare il notepad (Blocco Note) di Windows. Esistono tuttavia appositi editor visuali che permettono di operare con maggiore semplicità (Frontpage, Dreamweaver).

TAG E ATTRIBUTI

Un **tag** HTML è un comando che permette di delineare un elemento, e di influenzarlo attraverso l'istruzione che contiene. Ad esempio se vogliamo applicare il grassetto ad una parola, useremo la coppia di tag (tag di apertura e tag di chiusura) e al loro interno digiteremo la parola in questione. Un **attributo**, associato ad un tag, è un'ulteriore proprietà di formattazione. ES: testo - l'attributo 'face' formatta il nostro testo, associando ad esso il carattere Arial.

IMPOSTARE UNA PAGINA HTML

Tutte le pagine che desideriamo realizzare dovranno avere necessariamente un'impostazione di base per essere riconosciute e processate dal browser:

```
<html>
<head>
<title> Titolo della pagina </title>
</head>
<body>
.....
</body>
</html>
```

I tag <html></html> aprono e chiudono la pagina. Nei tag <head></head> andranno (oltre al titolo della pagina) ulteriori informazioni sul documento web note come meta-tag , script, fogli di stile, ecc. I tag <title></title> conterranno il titolo del nostro documento. Infine i tag <body></body> conterranno il corpo della pagina, ossia il testo e tutti gli altri elementi (immagini, tabelle, ...) che intendiamo rendere visibili.

IL TAG BODY

Come abbiamo visto, all'interno dei tag <body></body> si inseriscono tutti gli elementi che verranno visualizzati a video. Questo tag supporta molti attributi, tra cui, i più importanti sono:

bgcolor - colore di sfondo. Nome (in inglese) o codice colore (in esadecimale). Ad esempio "red" oppure "#FF0000".

background - immagine di sfondo;

topmargin - margine superiore;

leftmargin - margine sinistro;

bottommargin - margine inferiore;

rightmargin - margine destro;

ES: <body topmargin = 100 bgcolor = "#FF0000"></body> - in questo caso il margine superiore della nostra pagina avrà ampiezza 100 e il colore di sfondo sarà rosso.

FORMATTAZIONE DEL TESTO

Quelli che seguono sono i tag più usati per formattare il testo (stile del testo, colore del testo, grandezza del testo).

I tag di titolo (o **header**) creano la necessaria distinzione tra titoli, sottotitoli, corpo del testo ed eventuali citazioni; ad esempio:

<h1> Testo </h1> → **Testo**
 <h2> Testo </h2> → **Testo**
 <h3> Testo </h3> → **Testo**
 <h4>Testo </h4> → **Testo**
 <h5>Testo </h5> → **Testo**
 <h6>Testo </h6> → **Testo**

Il tag **FONT** è utilizzato per definire tipo, grandezza e colore del testo racchiuso in esso. Ad esempio, attraverso la riga di codice:

```
<font face = "verdana" size=2 color ="#0000FF"> testo </font>
```

stabilisco che la parola 'testo', contenuta nei tag , sia di tipo verdana (attributo face), di grandezza 2 (attributo size) e di colore blu (attributo color).

Altri Tag:

 Testo : applicano lo stile grassetto al testo che è racchiuso in essi (ad esempio **Testo**)

<i> *Testo* </i>: applicano lo stile corsivo al testo che è racchiuso in essi (ad esempio *Testo*)

<u> Testo </u>: applicano lo stile sottolineato al testo che è racchiuso in essi (Testo).

PARAGRAFI, ALLINEAMENTO E BLOCCHI

Definire un paragrafo e allineare il testo

Per definire un paragrafo occorre utilizzare gli appositi tag

<p> **testo del paragrafo** </p>

Tutto il testo che racchiudono, farà parte dello stesso paragrafo. Per allineare il testo occorre utilizzare l'attributo align.

ES1: <p align = center> testo allineato al centro</p> - il testo sarà allineato al centro della pagina;

ES2: <p align = left> testo allineato a sinistra</p> - il testo sarà allineato sul lato sinistro della pagina;

ES3: <p align = right> testo allineato a destra</p> - il testo sarà allineato sul lato destro della pagina;

ES4: <p align = justify> testo giustificato</p> - il testo sarà giustificato e si adatterà alla pagina e ai margini.

Interruzioni di riga

Le interruzioni di riga servono per interrompere appunto una riga di testo in un punto, facendola riprendere a capo. Il tag utilizzato è

(break) e non prevede un ulteriore tag di chiusura.

Definire dei blocchi

Spesso in html è utile definire dei blocchi, siano essi di solo testo, testo e immagini, o altri elementi. La coppia di tag utilizzata è <div> </div> che indicano l'inizio e la fine di un blocco. Il tag <div> supporta l'attributo align (come abbiamo visto per i paragrafi).

Inserire una riga orizzontale

Possiamo inserire una linea di separazione per evidenziare maggiormente il distacco tra due blocchi ricorrendo al comando:

<hr>

(horizontal rule). Anche in questo caso non è previsto un tag di chiusura. E' possibile variare l'aspetto della linea con gli attributi **size** (spessore della linea), **width** (larghezza in percentuale della linea rispetto all'intera schermata), **align** (per centrarla o allinearla a sinistra o a destra) e **color** (colore della linea). Ad esempio:

```
<hr size=4 width=60% color="blue">
```

INSERIRE UN'IMMAGINE

Per inserire un'immagine in HTML, viene usato il tag insieme all'attributo **src**:

```
<img src ="immagine">
```

(se l'immagine non si trova nella stessa cartella, occorre definire il path). Le immagini inseribili in una pagina web devono avere estensione **.jpg** oppure **.gif**. Come per il tag
, anche il tag è di sola apertura (non prevede un tag di chiusura).

Principali Attributi: Tra gli attributi che il tag supporta, quelli che occorre conoscere sono:

alt - testo che appare quando ci fermiamo col puntatore sull'immagine o quando non visualizziamo l'immagine stessa

border - valore numerico che esprime lo spessore dell'eventuale bordo che si vuol dare all'immagine

width - valore numerico che esprime la larghezza in pixel (o percentuale)

height - valore numerico che esprime la larghezza in pixel (o percentuale)

Ad esempio:

```
<img src ="immagine.gif" width=40 height=70 alt ="la mia immagine" border ="0">
```

Stiamo inserendo l'immagine 'immagine.gif', la larghezza è di 40 pixel, l'altezza di 70, il testo alternativo o esplicativo è 'la mia immagine', il bordo è uguale a 'zero' (nessun bordo).

LE LISTE

Le liste non ordinate sono dei semplici elenchi le cui singole voci sono precedute da un pallino: sono generate dal tag `` per segnalare l'inizio della lista e dal comando `` fatto precedere a ogni voce, più i relativi comandi di chiusura `` e ``. Ad esempio il codice:

```
<ul>
<li> Primo elemento della lista </li>
<li> Secondo elemento della lista </li>
<li> Terzo elemento della lista </li>
</ul>
```

produrrà sullo schermo il seguente risultato:

- **Primo elemento della lista**
- **Secondo elemento della lista**
- **Terzo elemento della lista**

Usando l'attributo **type** (con i valori `disc`, `circle` o `square`) si può modificare la forma del contrassegno della lista (cerchio vuoto, cerchio pieno, quadrato).

Le liste ordinate sono elenchi le cui singole voci sono precedute da un numero: sono generate dal tag `` per segnalare l'inizio della lista e dal comando `` fatto precedere a ogni voce, più i relativi comandi di chiusura `` e ``. Ad esempio il codice:

```
<ol>
<li> Primo elemento della lista </li>
<li> Secondo elemento della lista </li>
<li> Terzo elemento della lista </li>
</ol>
```

produrrà sullo schermo il seguente risultato:

1. **Primo elemento della lista**
2. **Secondo elemento della lista**
3. **Terzo elemento della lista**

Usando l'attributo **type** (con i valori `A`, `a`, `I`, oppure `i`) si può modificare la forma del contrassegno della lista (lettere maiuscole, lettere minuscole, numeri romani maiuscoli, numeri romani minuscoli).

COLLEGAMENTI IPERTESTUALI E ANCORAGGI CON NOME

Probabilmente rappresentano la parte più importante nell'HTML poichè ci permettono di spostarci all'interno di un sito, visitandone tutte le pagine, o da un sito all'altro. La coppia di tag utilizzata per inserire un collegamento ipertestuale è `<a> `. (a sta per *anchor*: ancoraggio). Al tag `<a>` bisogna associare l'attributo **href** che indica l'URL (Uniform Resource Locator) della pagina cui vogliamo indirizzarci. All'interno dei tag `<a>` andrà il testo o l'immagine, a cui vogliamo assegnare il collegamento.

Se, ad esempio, vogliamo fare in modo che il visitatore del sito, cliccando sull'immagine 'immagine.gif' venga portato al sito 'http://www.itcprogrammatori.net', useremo la seguente sintassi:

```
<a href ="http://www.itcprogrammatori.net"> <img src ="immagine.gif" > </a>
```

L'uso dei TARGET

Il tag `<a>` prevede, tra gli altri, l'attributo **target** ="destinazione del collegamento", dove per destinazione del collegamento si intende in quale finestra sarà aperta la pagina a cui è collegato il link su cui andremo a cliccare. Se non

impostiamo alcun target verrà riutilizzata la medesima finestra del browser. Se invece all'attributo target daremo il valore **_blank**, vale a dire target="_blank", cliccando sul collegamento ipertestuale, la pagina ad esso collegata sarà aperta in una nuova finestra. L'impiego dei target è molto utile se si utilizzano i frame (vedi più avanti).

Ancoraggi con nome

Creare un ancoraggio con nome significa definire un 'segnalibro' all'interno di una pagina web e far puntare ad esso un collegamento ipertestuale. Si tratta in pratica di costituire un link che consente la navigazione all'interno dello stesso documento web. Prima di tutto occorre definire il segnalibro:

```
<a name = "segnalibro">Testo</a>
```

In secondo luogo bisognerà creare il collegamento che punta al segnalibro così creato:

```
<a href = "#segnalibro">Vai al segnalibro creato</a>
```

Ora cliccando sul link 'Vai al segnalibro creato', saremo portati direttamente alla parola 'Testo'. (è possibile assegnare segnalibri non solo al testo ma anche ad altri elementi, come ad esempio alle immagini).

Invio di posta elettronica

Un collegamento ipertestuale che si incontra spesso nelle pagine web è quello relativo all'indirizzo di posta elettronica dell'autore della pagina. Ciò permette, ad esempio, agli utenti di inviare dei commenti o dei feedback all'autore stesso. Ad esempio:

```
<a href = "mailto: nome.cognome@alice.it" > Scrivi all'autore </a>
```

LE TABELLE

Le tabelle in HTML, oltre ad un uso classico (catalogazione di dati), vengono usate per creare il layout delle pagine web. Per inserire una tabella si utilizzano i tag **<table></table>**. Al loro interno vanno ancora i tag **<tr></tr>** che marcano una riga; nidificati tra questi, a loro volta, vanno i tag **<td></td>** che delineano invece le colonne (celle). All'interno di questi ultimi andranno gli elementi racchiusi nella tabella. Il codice sorgente avrà così la seguente struttura:

```
<table>
<tr>
<td>testo,immagini,altri elementi</td>
</tr>
</table>
```

Attributi delle tabelle (e delle celle in esse contenute) sono:

border – spessore del bordo della tabella (border = 0 rende il bordo invisibile);

caption – didascalia (con align = top oppure bottom, perché appaia sopra oppure sotto la tabella);

width - larghezza tabella o cella espressa in pixel o percentuale;

height - altezza tabella o cella espressa in pixel o percentuale;

align - allineamento tabella o allineamento elementi cella (center, left, right, justify);

bgcolor - colore di sfondo della tabella o della cella, espresso in codici esadecimali o nomi dei colori;

background - immagine di sfondo della tabella o della cella;

valign - allineamento verticale degli elementi contenuti nella cella (top, middle, bottom...);

cellspacing - lo spazio (in pixel) tra le celle che compongono la tabella;

cellpadding - lo spessore (in pixel) del margine delle celle che compongono la tabella;

Esistono inoltre due ulteriori attributi, **colspan** = numero e **rowspan** = numero, che applicati ad una cella (td) specificano rispettivamente il numero di colonne e righe espandibili su una cella:

1	2
3	

ecco il codice che genera questa piccola tabella:

```
<table width="80%" border="1" cellspacing="2" cellpadding="2" align="center">
<tr> .....(inizio prima riga)
<td>1</td> .....(prima cella/colonna)
<td>2</td> .....(seconda cella/colonna)
</tr> .....(fine prima riga)
<tr> .....(inizio seconda riga)
<td colspan="2">3</td> .....(terza cella/colonna)
</tr> .....(fine seconda riga)
</table>
```

I MODULI

Generalmente l'utilizzo dei **moduli (form)** è associato all'impiego di script installati sul server (cgi, pagine asp, php, ...) oppure per inviare dei dati ad un indirizzo di posta elettronica. I tag che vengono usati per delineare un modulo in HTML sono `<form></form>`. Al loro interno vengono inseriti ulteriori tag che richiamano gli altri elementi del modulo:

```
<form action="mailto:webmaster@guidainlinea.com" method="get/post">
```

(in questo esempio il modulo invia i dati che andremo ad inserire all'indirizzo e-mail contenuto nell'action; per il metodo d'invio, **method**, si consiglia di utilizzare 'post'. In questo modo i dati saranno inviati al server separatamente.)

Elementi di un Form

1) `<input type="text" size="30" maxlength="50" name="nome2" value="valore_iniziale">`

(visualizza una casella di testo all'interno della quale è possibile digitare dei dati)

2) `<input type="checkbox" name="nome3" checked>`

(visualizza una casella spuntata che può essere selezionata o deselezionata dall'utente. In questo esempio è selezionata poichè è presente l'attributo 'checked'. Naturalmente è possibile inserire del testo prima o dopo il tag che affiancherà la casella di testo)



3) `<input type="password" name="nome4" size="15" maxlength="50">`

(visualizza una casella di testo particolare, dedicata all'inserimento delle password. Infatti, se proviamo ad inserire del testo al suo interno, questo sarà criptato)

4) `<input type="radio" name="scelta" value="scelta1">`
`<input type="radio" name="scelta" value="scelta2" checked >`
`<input type="radio" name="scelta" value="scelta3">`

(utile allorchè si debba selezionare una sola tra più opzioni di scelta. Infatti selezionandone una si deselezionano automaticamente le altre. Come si può notare, la casella centrale è già selezionata perché è stato inserito l'attributo 'checked' all'interno del tag)



5) `<input type="reset" value="reimposta">`

(visualizza un pulsante che permette di resettare/azzerare il modulo, cancellando eventualmente tutti i dati precedentemente inseriti)

6) `<input type="submit" value="invia">`

(visualizza un pulsante che permette l'invio del modulo)

Introduzione ai Fogli di stile

Fogli di Stile CSS

Il **CSS** (*Cascading Style Sheets* o *Fogli di stile*) è un linguaggio usato per definire la formattazione di documenti [HTML](#), [XHTML](#) e [XML](#).

Le regole per comporre il CSS sono contenute in un insieme di direttive (*Recommendations*) emanate a partire dal 1996 dal [W3C](#).

L'introduzione del CSS si è resa necessaria per *separare i contenuti dalla formattazione* e permettere una programmazione più chiara e facile da utilizzare

Fogli di Stile CSS

Un normale documento Html unisce insieme contenuti, struttura e stile.

Infatti volendo visualizzare una normale riga di testo con font verdana, grandezza 3 e colore blu avremmo:

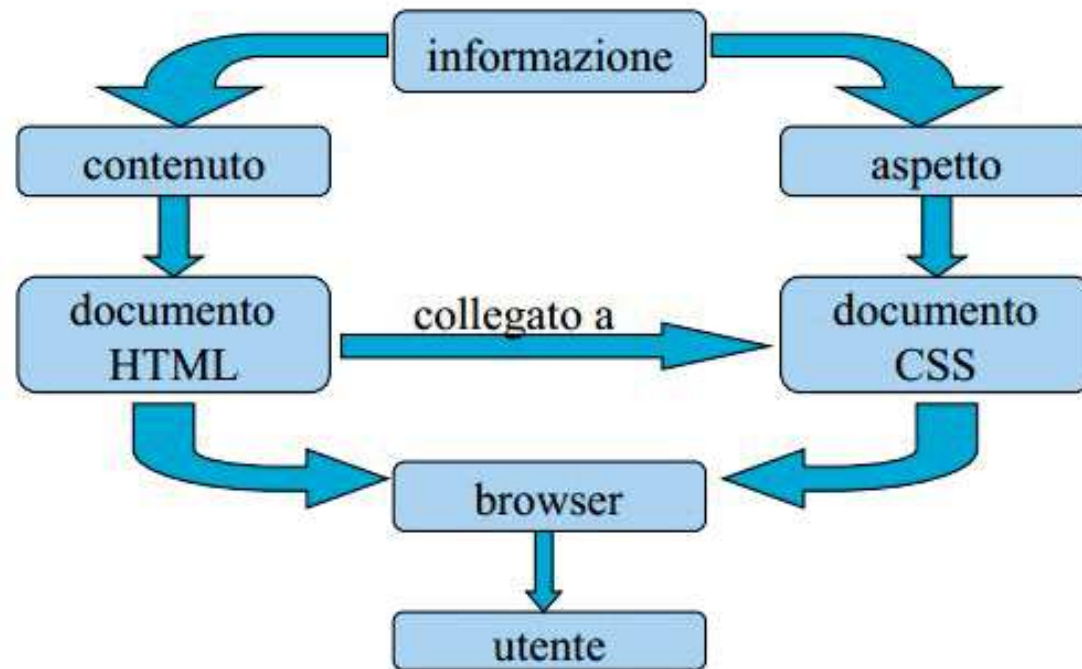
```
<FONT FACE=VERDANA SIZE=3 COLOR=BLUE>Testo comune</FONT>
```

come si può vedere, il contenuto (Testo comune) e' contenuto nello stile (font, size e color).

Se, si volesse modificare il font di un sito formato da 200 documenti, l'operazione, oltre che lunga, Potrebbe risultare snervante.

Fogli di Stile CSS

- I **CSS** risolvono questo problema perchè *separano il contenuto dalla presentazione*. Il contenuto è nella pagina html, mentre la presentazione è riservata ai fogli di stile.



Un foglio di stile è un insieme di regole, accompagnate, volendo, da qualche nota di commento.

CSS esterni e interni

- È **esterno** un foglio di stile definito in un file separato dal documento. Si tratta di semplici documenti di testo editabili anche con il Blocco Note o TextEdit ai quali si assegna l'estensione **.css**.
- Un foglio di stile si dice invece **interno** quando il suo codice è compreso in quello del documento. A seconda che si lavori con un CSS esterno o interno variano sintassi e modalità di inserimento. Rispetto a queste diverse modalità si parla di fogli di stile
 - ❑ **Collegati**
 - ❑ **Incorporati**
 - ❑ **In linea.**

Fogli collegati

I fogli collegati sono **esterni**. Si tratta di fogli di stile definiti in files separati dal documento.

```
<html>
```

```
<head>
```

```
  <title>Inserire i fogli di stile in un documento</title>
```

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

```
</head>
```

```
<body>
```

```
  .....
```

Fogli incorporati

I fogli incorporati sono quelli inseriti direttamente nel documento HTML tramite l'elemento **<style>**. Anche in questo caso la dichiarazione va posta all'interno della sezione **<head>**:

```
<html>
<head> <title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
  body { background: #FFFFCC; }
</style> </head>
<body>
.....
```

Gestione del colore



Figura 1 - La palette VGA di Windows

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

#RRGGBB

Le semplici 16 parole chiave non esauriscono ovviamente la gamma dei colori visualizzabili su un monitor moderno.

Già in HTML era possibile impostare il colore di un elemento servendosi di codici esadecimali.

In essi, le prime due lettere (o numeri) corrispondono ai valori per il colore rosso (RED), la seconda coppia fa riferimento al verde (GREEN), l'ultima al blue (BLUE).

La proprietà color

Visti i sistemi per rappresentare i colori, dobbiamo ora chiarire un aspetto importante. Per ogni elemento si possono definire almeno tre colori:

- il colore di primo piano (foreground color)
- il colore di sfondo (background color)
- il colore del bordo (border color)

La proprietà color definisce esclusivamente:

- il colore di primo piano, ovvero quello del testo
- il colore del bordo di un elemento quando non si imposti esplicitamente quest'ultimo con le proprietà border o border-color

[Esempio](#)

Fogli in linea

L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'**attributo 'style'**. Esso fa parte della collezione di attributi (X)HTML definita **Common**: si tratta di quegli attributi applicabili a tutti gli elementi. La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea. La sintassi generica è la seguente:

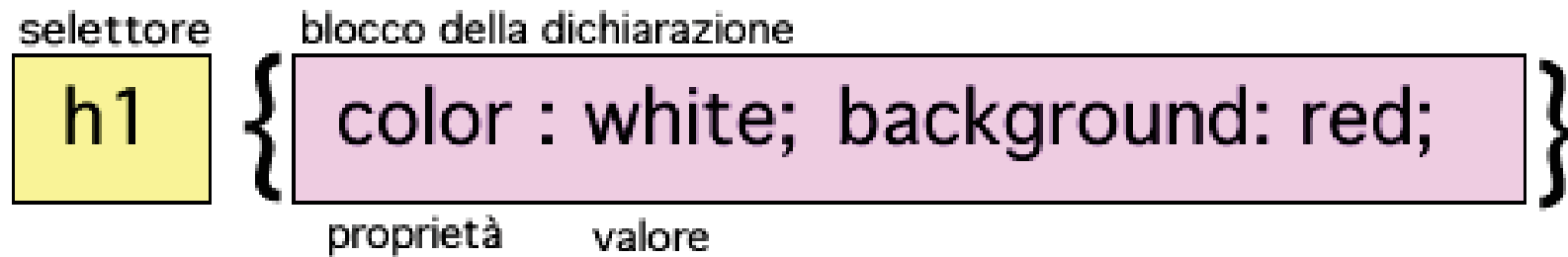
```
<elemento style="regole_di_stile">
```

Se, ad esempio, si vuole formattare un titolo **H1** in modo che abbia il testo di colore rosso e lo sfondo nero, scriveremo:

```
<h1 style="color: red; background: black;">...</h1>
```

Le cose da osservare nel codice sono due. Come valore di **style** si possono dichiarare più regole di stile. Esse vanno separate dal punto e virgola. I due punti si usano invece per introdurre il valore della proprietà da impostare.

Com'è fatta una regola



Questa è una tipica struttura di una regola CSS. Essa è composta da due blocchi principali:

- **il selettore**
- **il blocco delle dichiarazioni**

Il selettore serve a definire la parte del documento cui verrà applicata la regola. In questo caso, ad esempio, verranno formattati tutti gli elementi **<H1>**: lo sfondo sarà rosso, il colore del testo bianco.

Il blocco delle dichiarazioni è delimitato rispetto al selettore e alle altre regole da **due parentesi graffe**. Al suo interno possono trovare posto più dichiarazioni. Esse sono sempre composte da una coppia:

- **Proprietà**
- **valore**

La proprietà definisce un aspetto dell'elemento da modificare (margini, colore di sfondo, etc) secondo il valore espresso. Proprietà e valore devono essere separati dai due punti.

Proprietà singole e a sintassi abbreviata

Le proprietà singole sono la maggior parte: impostano per un dato elemento o selettore un singolo aspetto. Con le proprietà a sintassi abbreviata è possibile invece definire con una sola dichiarazione un insieme di proprietà.

Esempio :

```
p{  
margin-top: 10px;  
margin-right: 5px;  
margin-bottom: 10px;  
margin-left: 5px;  
}
```

Proprietà Singole

Proprietà a sintassi abbreviate

```
p {margin: 10px 5px 10px 5px;}
```

I selettori

Fondamentalmente una regola CSS viene applicata ad un **selettore**. La parola parla da sé: si tratta di una semplice dichiarazione che serve a **selezionare** la parte o le parti di un documento soggette ad una specifica regola

Selettore di elementi

È il più semplice dei selettori. È costituito da uno qualunque degli elementi di HTML.

Sintassi:

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

```
p {background: white; font: 12px Verdana, arial, sans-serif;}
```

```
table {width: 200px;}
```

[Esempio](#)

È possibile nei CSS raggruppare diversi elementi al fine di semplificare il codice. Gli elementi raggruppati vanno separati da una **virgola**.

```
h1, h2, h3 {background: white;}
```

[Esempio](#)

I selettori

Selettore universale

Anche nei **CSS** abbiamo un **jolly**. Il selettore universale serve a selezionare tutti gli elementi di un documento. Si esprime con il carattere * (asterisco).

Sintassi

Esempio

```
*{ color: black; }
```

I selettori

Selettori delle classi

Per definire una classe si usa far precedere il nome da un semplice punto: `.nome_della_classe`

Esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      .verde { color:green;}
      .blu { color:blue;}
      .rosso { color:red;}
    </style>
  </head>
  <body>
    <h1 class="verde">Il titolo è in verde</h1>
    <h1 class="blu">Il titolo è in blu</h1>
    <h1 class="rosso">Il titolo è in rosso</h1>
    <em class="blu">La scritta in corsivo è in blu</em>
    <p class="verde">Il paragrafo normale è in verde</p>
  </body>
</html>
```

Le classi sono importanti perché possono essere applicate più volte ad elementi del documento e ad elementi diversi

I selettori

Selettori delle classi

Esiste un secondo tipo di sintassi: **<elemento>.nome_della_classe**

Esso è più restrittivo rispetto alla sintassi generica perchè applica la regola solo all'elemento specificato esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      body { background:#FFFF99; }
      h1.verde { color:green; background:#FFFF99 }
    </style>
  </head>
  <body>
    <h1>Questo titolo non è verde</h1>
    <h1 class="verde">Questo titolo è verde</h1>
  </body>
</html>
```

[Esempio](#)

Selettore delle identità ID

Un altro selettore speciale, oltre alle classi, è il selettore delle identità (ID selectors).

Le caratteristiche sono sempre le stesse e ciò che vale per le classi vale anche per i selettori.

L'unica differenza sostanziale è nella sintassi del comando che, anziché avere un punto prima del nome della classe, ha una cassetta # e nel fatto che un selettore di identità può essere inserito una volta sola in ogni documento

(ovviamente si possono inserire selettori aventi nomi diversi).

La sintassi di un selettore ID è semplicissima. Basta far precedere il nome dal simbolo di cancelletto #: `#nome_id`

Selettore delle identità ID

Esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      #verde { color:green;}
      #blu { color:blue;}
      #rosso { color:red;}
    </style>
  </head>
  <body>
    <h1 id="verde">Il titolo è in verde</h1>
    <h1 id="blu">Il titolo è in blu</h1>
    <h1 id="rosso">Il titolo è in rosso</h1>
  </body>
</html>
```



Esempio

Classi e ID

La fondamentale differenza fra classi ed ID riguarda l'uso che viene fatto dell'attributo **id** in un documento HTML.

In un documento HTML l'attributo **id** è usato per identificare in **modo univoco** un elemento.

In pratica, se si assegna ad un paragrafo l'id "formattatesto", non si può più usare questo valore nel resto della pagina.

Di conseguenza, l'ID **#formattatesto** dichiarato nel CSS trasformerà solo quel paragrafo specifico.

Una singola classe, al contrario, può essere assegnata a più elementi, anche dello stesso tipo.

Classi e ID

Esempio:

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

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

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

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

```
<style type="text/css">
  .rosso{color:red;}
</style>
```

La classe **.testorosso** presente nel CSS formatterà allo stesso modo il testo del paragrafo, del div e della tabella.

Questo invece è sbagliato. **(usato due volte)**

```
<p id="rosso">....</p>
```

```
<div id="rosso">...</div>
```

```
<style type="text/css">
  #rosso{color:red;}
</style>
```

La strategia dovrà dunque essere questa : se uno stile va applicato ad un solo specifico elemento conviene usare un **ID**, se invece si prevede di usarlo più volte ovvero su più elementi conviene definire nel CSS una **classe**.

Il contenitore div

La sua definizione è *generic block-level element*
Cioè contenitore generico block level.

Il fatto che sia un elemento block-level ci garantisce il fatto che possa contenere qualsiasi tipo di elemento html. Inoltre, la sua presentazione naturale (quindi senza fogli di stile) è totalmente neutra:

infatti questo elemento si presenta di default senza margini, bordi o padding.

È quindi il contenitore per eccellenza per realizzare layout senza l'uso di tabelle:

ad ogni div portante verrà associata una sezione della pagina.

Esso è utilizzato soprattutto per creare layout senza tabelle sfruttando le caratteristiche del cosiddetto **BOX MODEL**