

Internet e HTML

David De Nora
www.webtocom.com





Internet e il WWW

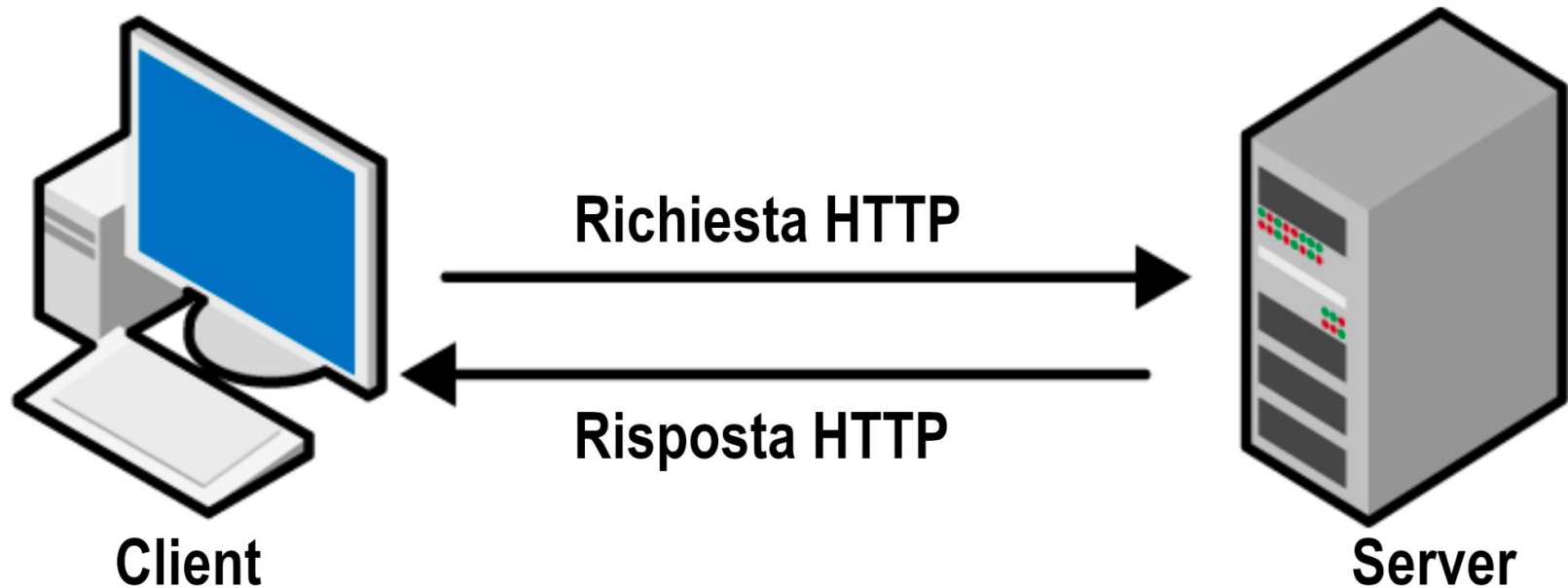
- Spesso i due termini vengono confusi ma sono due cose diverse.
- **Internet** è una rete internazionale di dispositivi (computer e router) e connessioni fisiche che consentono la comunicazione tra dispositivi anche a grandi distanze.
- Il **World Wide Web** (o WWW o web) è il principale servizio di Internet.
- Il web che consente l'accesso ad un insieme molto vasto di contenuti collegati tra loro attraverso collegamenti ipertestuali (**link**).
- In altre parole attraverso le infrastrutture di internet è possibile fruire delle informazioni presenti sul web.



Web Server, Client e HTTP

- Un **Web Server** è un computer connesso alla rete internet, la cui funzione è inviare file ai **Browser** di dispositivi connessi anch'essi ad internet.
- Il browser (Firefox, Chrome, ...) è il **Client** che invia richieste al web server.
- In alcuni casi si indica come client la macchina che invia la richiesta invece che il browser.
- Client e Server comunicano attraverso una lingua comune che è il protocollo **HTTP** (HyperText Transfer Protocol).
- Il server risponde alle richieste del browser.
- Il browser è in grado di gestire le risposte del server.
- Tutte le comunicazioni tra client e server si basano su uno schema domanda-risposta.

Schema di comunicazione Server-Client





Gli indirizzi web: URL

- Tutte le risorse disponibili sul web sono reperibili attraverso la loro **URL**.
- URL sta per **Uniform Resource Locator** e può essere tradotto con: **localizzatore uniforme di risorse**.
- Un URL ha la seguente struttura:

protocollo://dominio/risorsa

ad esempio:

<http://www.webtocom.com/index.php>



Gli indirizzi web: URL

Abbiamo visto che un url ha la seguente struttura:

protocollo://dominio/risorsa => <http://www.webtocom.com/index.php>

Se la risorsa è essere omessa ne verrà caricata una di default ad esempio index.html:

<http://www.google.com/>

Una risorsa può essere anche un PATH, come in:

<http://www.miosito.it/chi-siamo/la-societa.html>



HTML

- Il browser oltre a comunicare con il server si occupa dell'impaginazione delle pagine web.
- Una **pagina web** è un file di testo il cui contenuto è scritto in linguaggio HTML.
- **HTML** è l'acronimo di **HyperText Markup Language** ed è il linguaggio che viene interpretato dal browser per strutturare la visualizzazione ed i contenuti della pagina web.
- Un **sito web** non è altro che insieme di pagine web sullo stesso dominio (e omogenee tra loro).
- L'HTML **non produce ERRORI**.



I Tag

- L'HTML è un linguaggio composto da **Tag**.
- I tag sono gli elementi di base dell'HTML e possono essere combinati tra loro per produrre le pagine.
- I tag sono di due tipi:
 - **Accoppiati** (la maggior parte)
 - **Autochiudenti** (pochi ma molto usati)
- I tag servono ad assegnare proprietà o a svolgere funzioni.
- I tag possono essere messi uno dentro l'altro o come si dice più spesso **annidati**.
- I tag hanno un valore semantico quindi il loro nome indica la loro funzione.
- I tag si chiudono come le parentesi: l'ultimo che si apre è il primo che si chiude.



Esempi di tag

```
<title>Titolo della pagina</title>
```

```
<p>Io sono un paragrafo</p>
```

```

```

```
<br>
```

```
<h2>Titolo</h2>
```



Attributi dei tag

Nei tag di apertura dei tag accoppiati o nel tag autochiudente è possibile inserire gli attributi.

Gli attributi servono a specificare delle proprietà aggiuntive al tag.

Ad esempio l'attributo `charset` del tag definisce il tipo di carattere della pagina:

```
<meta charset="UTF-8">
```

Generalmente gli attributi si definiscono nel seguente modo:

```
<tag attr1="val" attr2="val" ... >contenuto del tag </tag>
```



Struttura di una pagina HTML

Le pagine HTML iniziano con una prima riga di definizione:

```
<!DOCTYPE html>
```

Questo serve a dire che il contenuto della pagina è HTML.

Tutto il codice della nostra pagina è contenuto all'interno del tag html

```
<html lang="en">
```

```
...
```

```
</html>
```



Struttura di una pagina HTML

All'interno dell'head ci sono due tag “fratelli”, head e body:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```



Il tag head

Il tag head è il primo tag contenuto nel tag html e contiene tutte le informazioni riguardanti la pagina, ossia i metadati. Quello che inseriamo nell'head non apparirà nella pagina.

All'interno dell'head si trovano i tag meta e title:

```
<head>  
  <meta charset="UTF-8"> -> set di caratteri usato nella pagina  
  <title>Titolo della pagina</title>  
</head>
```





Il tag body

Il tag body è il corpo della pagina, al suo interno si inserisce tutto quello che si vuole visualizzare.

Body si apre subito dopo head e precede la chiusura di html

```
<body>  
  <!-- contenuto della pagina -->  
</body>
```

**Tutti i tag presenti in una pagina HTML
devono essere inseriti o in head o in body**



Esempio di pagina HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Titolo della pagina</title>
</head>
<body>
  <!-- contenuto della pagina -->
</body>
</html>
```



I commenti

I commenti non saranno visualizzati nel browser, ma sono utili per documentare il tuo codice.

Un commento HTML può essere aggiunto usando la seguente sintassi:

```
<!-- questo è un commento HTML-->
```

Un commento HTML si apre usando `<!--` e si chiude con `-->` .



I titoli

L'html mette a disposizione 6 tipi di titoli che vanno dal titolo principale della pagina a vari sottotitoli di importanza decrescente.

Per il titolo principale della pagina si usa il tag **<h1>**, mentre il sottotitolo meno rilevante è **<h6>**. I titoli vengono rappresentati in maniera “decrescente”, **<h1>** è il più grande **<h6>** il più piccolo

```
<h1>titolo principale</h1>
```

```
<h2>sottotitolo principale</h2>
```

```
...
```

```
<h6>sottotitolo meno importante</h6>
```



Formattazione del testo

Esistono numerosi tag per la formattazione del testo:

`testo in grassetto`

`testo corsivo`

`<mark>testo evidenziato</mark>`

`^{apice}`

`_{pedice}`

`<small>testo più piccolo</small>`



I CSS

CSS sta per **Cascading Style Sheets**, cioè fogli di stile a cascata.

Tramite i CSS si può definire tutto quello che riguarda l'aspetto grafico di una pagina web: i colori e la dimensione dei font, lo spazio fra gli elementi, le loro dimensioni e posizionamento.

Inoltre si possono definire i colori e le immagini di sfondo.

I fogli di stile sono composti da regole.

I fogli di stile possono essere inseriti nelle pagine in 3 modi:

- inline (ok per imparare e fare prove)
- nell'head (ok per imparare e fare prove)
- inclusi (migliore e consigliato)



CSS Inline

Il modo più semplice per assegnare proprietà CSS ad un elemento è quello di utilizzare l'attributo `style="..."` all'interno di un tag:

```
<h1 style="background-color: white;"></h1>
```

All'interno del tag stile saranno inserite una serie di dichiarazioni cioè coppie ***proprietà: valore;*** separate tra di loro dal punto e virgola.

Il CSS inline assegna proprietà al solo elemento in cui appare.



CSS nel tag head

I CSS possono essere inseriti all'interno del tag head utilizzando il tag style:

```
<head>
  <style>
    body {background-color: powderblue;}
    h1   {color: blue;}
    p    {color: red;}
  </style>
</head>
```



CSS esterni

I CSS esterni sono il metodo più utilizzato quando si ha a che fare con un sito web composto da più pagine web ed è necessario avere una formattazione omogenea delle pagine.

In questa modalità i CSS sono scritti in un file esterno con con estensione .css (ad esempio style.css) e vengono caricati nella pagina inserendo nell'head il tag link:

```
<link rel="stylesheet" href="style.css">
```

All'interno del file style.css saranno contenute tutte le regole css.

CSS esterni

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Foglio di stile esterno</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Foglio di stile esterno</h1>
  <p>Esempio di foglio di stile esterno</p>
</body>
</html>
```

style.css

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```



Dove inserire i CCS?

Sebbene l'HTML ci consenta di inserire i CSS in 3 modi diversi la scelta migliore e consigliata è quella di inserire i fogli di stile esterni:

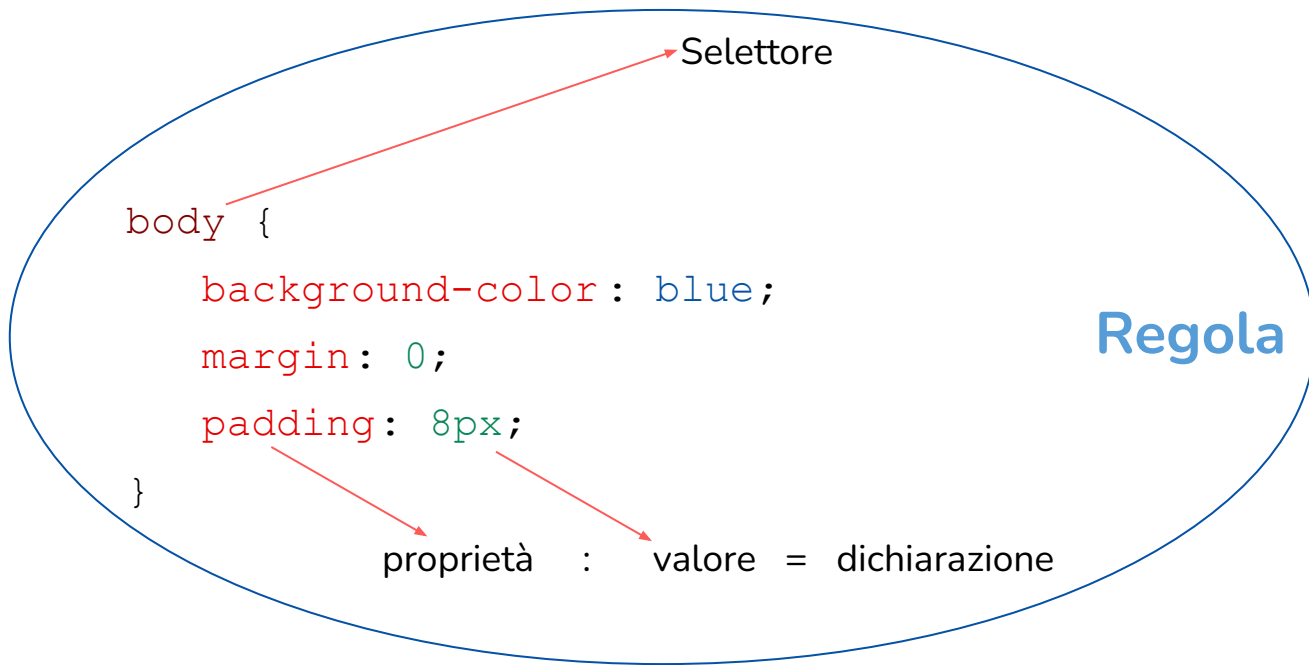
- sono applicabili a più pagine HTML
- facilitano la scrittura delle pagine
- facilitano la manutenzione dei CSS

I fogli di stile sono nati con l'intento di separare il contenuto (HTML) dalle regole di visualizzazione, cioè il layout della pagina, che si definisce con i fogli di stile.

I CSS inline o nel tag head violano questo principio generale.



Sintassi dei CSS: le regole





I selettori dei CSS

Esistono tre selettori di base dei CSS che sono strettamente legati alla struttura dell'HTML, i selettori di base possono essere combinati in modo da ottenere selettori più specifici.

Quelli di base sono:

- tag (definiti dall'HTML)
- classi
- id



I selettori dei CSS: tag

I tag sono le parole chiave definite dall'HTML. Per creare una regola basta scrivere il nome del tag e inserire tra parentesi graffe le definizioni.

La regola verrà applicata a tutti i tag per cui è stata definita

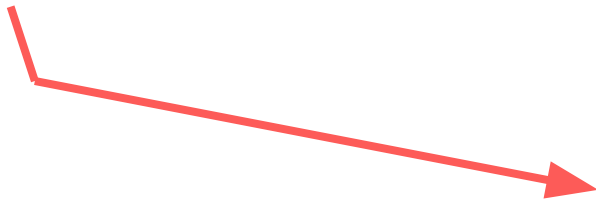
```
p {  
  background-color: blue;  
  margin: 0;  
  padding: 8px;  
}
```



I selettori dei CSS: le classi

Le classi CSS sono dei raggruppamenti di definizioni il cui intento è quello di essere applicati più volte ed a tag diversi, i loro nomi sono liberi ed iniziano con il punto. Vengono assegnati ai tag mediante l'attributo **class**:

```
.sfondoBlu{ background-color: blue;}
```



```
<p class="sfondoBlu">p sfondo blu</p>
```

```
<h1 class="sfondoBlu">h1 sfondo blu</h1>
```



I selettori dei CSS: gli ID

Gli ID CSS sono dei raggruppamenti di definizioni il cui intento è quello di essere applicati **un'unica volta** all'interno della pagina, i loro nomi sono liberi e devono iniziare con # (cancelletto). Vengono applicati al tag desiderato con l'attributo **id**:

```
#main{  
  width: 80%;  
  margin: auto;  
}
```



```
<div id="main">id main</div>
```



Combinazioni di selettori

```
a,div{color: #000;}  
p a{background-color: aqua;}  
p>a{background-color: aqua;}  
p.red{color: #f00;}  
p .red{color: #f00;}
```

```
/*applicato ad a e div*/  
/*applicato ai tag a contenuti p*/  
/*applicato ai tag a figli di p*/  
/*applicato ai tag p con classe red*/  
/*applicato ai tag con classe red contenuti in p*/
```



Conflitti tra definizioni CSS

Supponiamo di avere i seguenti codici CSS e HTML:

```
p { color: red; }                                <p class="blu">come sarò?</p>
.blu { color: #00F; }
```

La definizione di p e della classe .blu sono in conflitto.
In questo caso sarà blu perché vige la seguente gerarchia:

id > classe > tag



Specificità del CSS

Esiste un algoritmo che calcola la specificità di una regola CSS, in buona sostanza ci dice che meno elementi sono coinvolti da una regola più questa è specifica.

Per calcolare la specificità viene assegnato un punteggio ai selettori:

- style inline 1000
- ID 100
- classe 10
- tag 1

La specificità degli elementi composti viene calcolata sommando i vari punteggi.



Specificità del CSS

Selettore	Punteggio	Totale
p	1	1
p.class	1 + 10	11
#main	100	100
<p style="color: pink;">	1000	1000
#navbar p#demo	100 + 1 + 100	201
*	0	0 (il selettore universale è ignorato)



Specificità del CSS

Abbiamo appena visto che le dichiarazioni CSS inline sono le più potenti, vale anche la seguente gerarchia:

tag style > CSS esterno

quindi se una stessa regola è presente sia nel css esterno che all'interno del tag style la regola all'interno del tag style sovrascrive il CSS esterno, ovviamente solo dove ci sono dei conflitti.



I collegamenti ipertestuali (*link*)

Per aggiungere un collegamento ad una risorsa esterna si usa il seguente codice:

```
<a href="url">Testo cliccabile</a>
```

Si utilizza il tag `<a>` con l'attributo `href`, il valore di questo attributo è la pagina dove andremo se clicchiamo sull'area cliccabile.

I link possono anche essere applicati alle immagini.



I percorsi dei collegamenti

All'interno dell'attributo href="..." possiamo inserire due tipi di percorsi:

- assoluti
- relativi

I percorsi **assoluti** iniziano con **http(s)://** e sono **esterni** al sito:

```
<a href="https://blog.webtocom.com/">Il mio blog</a>
```



Percorsi relativi

I percorsi relativi sono quelli che si utilizzano per navigare all'interno di uno stesso sito.

Vediamo alcuni esempi:

```
<a href="file.html">Stessa directory</a>  
<a href="sottocartella/file.html">File nella sottocartella</a>  
<a href="../file.html">Cartella superiore</a>  
<a href="/">Root del sito</a>  
<a href="/file.html">A partire dalla root del sito</a>
```



I colori dei collegamenti

Di default i link hanno i seguenti colori:

- Un link non visitato è di colore blu
- Un link visitato è viola
- Un link attivo è rosso

Un link è attivo nel momento in cui ci clicchiamo sopra.

Inoltre sempre per le impostazioni iniziali un link è sottolineato.



Collegamenti e CSS

Togliamo la sottolineatura ai link:

```
a { text-decoration: none; }
```

Pseudo-classi

```
a:visited { color: orange; }
```

```
a:active { color: hotpink; }
```

:hover descrive il comportamento del link quando ci andiamo sopra con il mouse

```
a:hover { background-color: lightgreen; }
```



I colori dell'HTML e CSS

In origine i colori potevano essere inseriti all'interno delle pagine utilizzando la codifica **RGB**, dove R=Red, G=Green e B=Blue. In pratica tutti i colori vengono codificati mescolando rosso, blu e verde.

Le quantità di ciascun colore sono espresse in numeri esadecimali che vanno da 00 a FF, ossia da 0 a 255.

Le combinazioni possibili sono quasi 17 milioni, dato che si possono combinare 256 tonalità di ciascun colore: $256 \times 256 \times 256 = 16.777.216$



Codifica dei colori

I colori RGB hanno la seguente codifica:

#RRGGBB

La prima coppia di numeri rappresenta (in esadecimale) la quantità di rosso, la seconda quella di verde e la terza quella di blu.

Ad esempio:

#000000 -> Nero

#FF0000 -> Rosso pieno

#00FF00 -> Verde pieno

#0000FF -> Blu pieno

#FFFFFF -> Bianco

#800080 -> Viola



Codifica compatta dei colori

Quando abbiamo un colore in cui le due cifre di ciascun colore RGB sono doppie si può utilizzare una codifica compatta ad esempio:

- `#FF0077` -> `#F07`
- `#CCEE33` -> `#CE3`
- `#000000` -> `#000`
- `#FFFFFF` -> `#FFF`

Con la codifica compatta possono essere rappresentati $16 \times 16 \times 16 = 4096$ colori.



Css Named Colors

Oltre alla codifica RGB si possono utilizzare nelle dichiarazioni css dei colori tramite il loro nome.

E' possibile usare i nomi in inglese dei principali colori, ad esempio:

- Black
- Orange
- White
- Blue
- Yellow
- Lime
- Green
- Teal
- Gray
- Silver



CSS rgb e rgba

Oltre ai 3 modi visti in precedenza, in una regola CSS i colori possono essere assegnati alle proprietà usando rgb e rgba.

```
.classe1 {  
  background-color: rgb(200, 100, 80); -> quantità dei colori decimale da 0 a 255  
}  
  
.classe2 {  
  background-color: rgba(200, 100, 80, 0.2); -> come sopra ma con opacità (da 0 a 1)  
}
```



Le immagini nell'HTML

Nell'HTML le immagini vengono inserite attraverso il tag `img`:

```

```

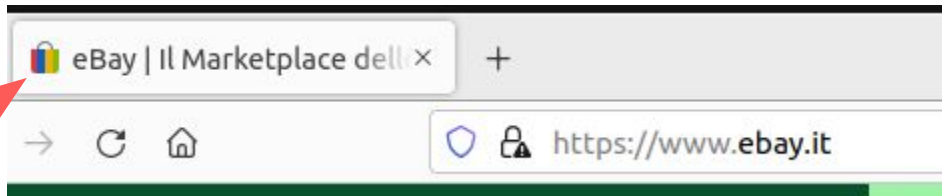
Il tag `img` è autochiudente, con l'attributo **`src=""`** assegnamo il percorso dell'immagine, con il tag **`alt=""`** definiamo un testo alternativo, che viene mostrato se l'immagine non è disponibile o mentre si carica.

Se non assegniamo regole CSS l'immagine occuperà uno spazio pari alle sue dimensioni.



Favicon

La favicon è una piccola immagine che appare vicino al titolo sulla linguetta del browser.





Favicon

Per aggiungere la favicon al proprio sito si usa il tag link che va inserito dopo il titolo.

```
<head>
  ...
  <title>Mio sito</title>
  <link rel="icon" type="image/x-icon" href="/img/favicon.ico">
</head>
```

Oltre ai file con estensione .ico si possono usare immagini 16x16.

Per creare una favicon può utilizzare degli appositi siti come <https://www.favicon.cc/>



Le tabelle

L'HTML consente la disposizione dei dati in righe e colonne.

```
<table>
  <tr>
    <th>Colonna 1</th>
    <th>Colonna 2</th>
  </tr>
  <tr>
    <td>Cella 1</td>
    <td>Cella 2</td>
  </tr>
  <tr>
    <td>Cella 3</td>
    <td>Cella 4</td>
  </tr>
</table>
```

Tag della tabella

Intestazione della tabella

Tag della riga

Tag della cella

Le tabelle vengono disegnate senza bordi, se li vogliamo aggiungere dobbiamo utilizzare i CCS.



Le Tabele

In realtà le tabelle non sono composte da righe e colonne ma da:

- celle inserite in righe
- righe una sotto l'altra

**NON C'È NESSUN VINCOLO CHE OBBLIGA A METTERE
LO STESSO NUMERO DI CELLE SU OGNI RIGA!!**

La dimensione delle tabelle, se non specificata, è data dal contenuto.

Le celle di una colonna hanno la stessa larghezza, quelle di una riga la stessa altezza.



Le Tabelle

```
table{  
    border:1px solid mediumseagreen;  
}  
th,td{  
    border:1px solid mediumseagreen;  
}
```

Colonna 1	Colonna 2
Cella 1	Cella 2
Cella 3	Cella 4



Le Tabelle

```
table,th,td{  
  border:1px solid mediumseagreen;  
  border-collapse: collapse;  
}
```

Colonna 1	Colonna 2
Cella 1	Cella 2
Cella 3	Cella 4



Le Tabelle: colspan

Le tabelle HTML possono avere celle, td o th, che occupano più righe.

```
<table>
  <tr>
    <td colspan="2">Titolo</td>
  </tr>
  <tr>
    <td>Cella 1</td>
    <td>Cella 2</td>
  </tr>
</table>
```

Titolo	
Cella 1	Cella 2



Le Tabelle: rowspan

Il rowspan è analogo al colspan ma in verticale.

```
<table>
  <tr>
    <th rowspan="2">Titolo</th>
    <td>Cella 1</td>
  </tr>
  <tr>
    <td>Cella 2</td>
  </tr>
</table>
```

Titolo	Cella 1
	Cella 2



CSS allineare il testo

L'allineamento predefinito del testo è a sinistra, utilizzando la proprietà ***text-align*** è possibile modificarlo:

```
text-align: center;    -> allinea al centro  
text-align: right;     -> allinea a destra  
text-align: justify;   -> testo giustificato  
text-align: left;      -> allinea a sinistra (predefinito)
```

Il testo allineato a sinistra o a destra è detto “*a bandiera*”.



Liste puntate e numerate

L'HTML prevede due tipi di tag per le liste.

Liste puntate:

```
<ul> -> unordered list  
  <li>elemento 1</li> -> list item  
  <li>elemento 2</li>  
</ul>
```

- elemento 1
- elemento 2

Liste numerate:

```
<ol> -> ordered list  
  <li>elemento 1</li> -> list item  
  <li>elemento 2</li>  
</ol>
```

1. elemento 1
2. elemento 2



Gli attributi di ol

L'attributo **start="numero"** consente di far iniziare la lista da un numero diverso da 1:

```
<ol start="4">  
  <li>elemento 4</li>  
  <li>elemento 5</li>  
</ol>
```

La lista partirà da 4

L'attributo **type="1/A/a/I/i"** consente di modificare la modalità di conteggio.

- 1 -> numeri in sequenza
- A -> lettere maiuscole in sequenza
- a -> lettere minuscole in sequenza
- I -> numeri romani maiuscoli
- i -> numeri romani minuscoli



Display degli elementi

Display a blocco:

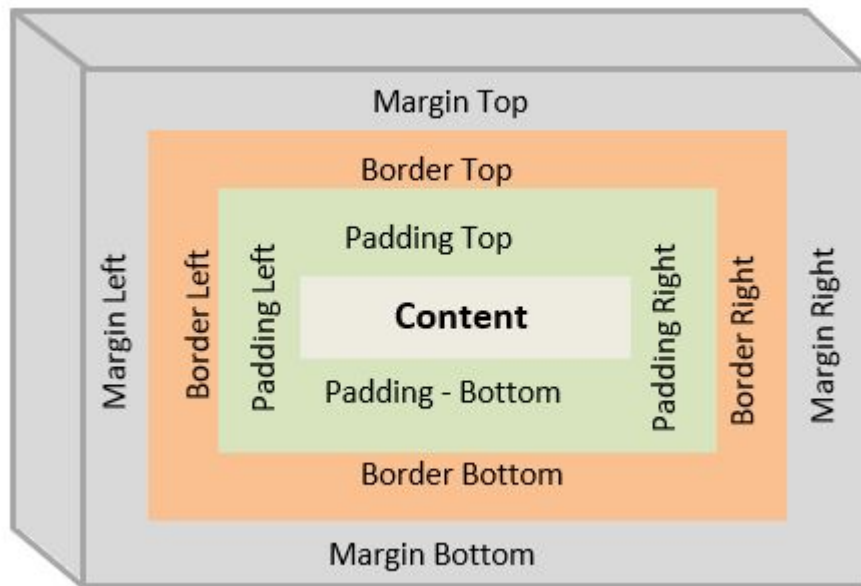
- gli elementi iniziano su una nuova linea
- quello che li segue inizia su una nuova linea
- occupano tutto lo spazio orizzontale disponibile
- es: h1, p, ul, li, div

Display inline:

- gli elementi non iniziano su una nuova linea
- occupano solo lo spazio a loro disposizione
- es: a, strong, em, sup, sub, span



Box Model CSS



Tutti i tag con **display a blocco** sono disegnati seguendo questo **modello**.

Il margine è **esterno**.
Padding e Content sono **interni**.

Il border è il **confine** tra interno ed esterno.



Proprietà CSS del Box Model

Possiamo definire le due dimensioni del content attraverso le proprietà **height** e **width**:

```
div{  
    height: 150px;  
    width: 180px;  
}
```

Altezza e larghezza possono essere in maniera percentuale.

L'altezza in percentuale viene calcolata solo se assegnata ad un box contenuto in un box con altezza in pixel.



Padding

Il padding è lo spazio (in px o %) che separa il content dal bordo. Il padding è interno quindi assumerà le caratteristiche dello sfondo.

Ci sono vari modi per assegnare il padding:

- padding-left, padding-right, padding-top, padding-bottom
- padding : 5px -> (padding in tutte le direzioni)
- padding : 5px 10px (sopra/sotto e destra/sinistra)
- padding : 0px 80px 120px; (sopra, dx/sx sotto)
- padding : 5px 3px 8px 9px (in senso orario a partire da sopra)



Border

I bordi possono essere assegnati, agli oggetti con display a blocco, singolarmente (border-top, border-left, ...) o insieme (border). A prescindere da come assegnamo il bordo solitamente si assegnano 3 valori:

- dimensione del bordo in px 0 %
- tipo di bordo: solid, dashed, dotted, ridge
- colore del bordo

Se il colore non è presente verrà usato quello del testo.

La dimensione è opzionale, il tipo va messo sempre.



Border

Box dashed

Box solid gray

Box dotted teal



Margini

Le proprietà CSS dei margini sono usate per creare spazio attorno agli elementi, al di fuori dei bordi.

Con il CSS si ha il pieno controllo dei margini, si possono impostare i margini in ogni dimensione di un elemento (top, right, bottom, e left). Vale quanto detto per il padding:

- margin-left, margin-right, margin-top, margin-bottom
- margin : 5px -> (margin in tutte le direzioni)
- margin : 5px 10px (sopra/sotto e destra/sinistra)
- margin : 0px 80px 120px; (sopra, dx/sx sotto)
- margin : 5px 3px 8px 9px (in senso orario a partire da sopra)



margin: auto;

Utilizzando la proprietà margin si possono centrare i box all'interno di altri box.

Assegnando il valore auto ai margini orizzontali, un oggetto con display a blocco può essere centrato all'interno di un altro oggetto con display a blocco.

Si può fare in tre modi:

- margin: auto;
- margin: 10px auto;
- margin-left: auto; margin-right: auto;

Per risultare centrato l'oggetto contenuto deve essere più stretto di quello che lo contiene.



Gli iframe

Gli iframe si usano per visualizzare una pagina web all'interno di un'altra pagina web (*embedding*), la sintassi del tag è:

```
<iframe src="url" title="descrizione"></iframe>
```

Gli iframe sono usati ad esempio per includere i video di youtube o le mappe di google nella propria pagina.

Hanno display a blocco e quindi possono essere usati altezza, larghezza, margini e bordi



Immagini di sfondo

Usando regole CSS è possibile assegnare immagini di sfondo attraverso la proprietà ***background-image: url('file,img')***. Ad esempio:

```
body{ background-image: url('sfondo.jpg');}
```

Le immagini di sfondo di default vengono affiancate orizzontalmente e verticalmente in modo da coprire tutto lo spazio a disposizione, come se fossero delle mattonelle.

Se non la vogliamo ripetere basta usare la dichiarazione:

```
background-repeat: no-repeat;
```



Immagini di sfondo

Ad esempio:

```
body{  
  background-image: url('sfondo.jpg');  
  background-repeat: no-repeat;  
  background-position: center;  
  background-color: orange;  
}
```

Centra l'immagine di sfondo e copre il resto dello spazio con il colore arancione.



CSS: opacity

Abbiamo già visto che l'opacità di un colore può essere assegnata tramite `rgba()`.

E' possibile applicare una opacità ad un elemento attraverso la proprietà `opacity`:

```
opacity: 0.7;
```

Il valore varia tra 0 e 1.

Il valore di default è 1, se poniamo valore 0 l'oggetto a cui è applicata non sarà visibile.



Text decoration

Il testo può essere “decorato” in vari modi:

```
p {  
  text-decoration-line: line-through;  
  text-decoration-color: red;  
  text-decoration-style: double;  
  text-decoration-thickness: 5px;  
}
```

overline, underline, line-through,
none

double, dashed, dotted, wavy,
solid, none



Text transform

La proprietà text-transform consente di modificare la visualizzazione dei caratteri di un elemento:

```
p.maiuscolo {  
    text-transform: uppercase;  
}  
p.minuscolo {  
    text-transform: lowercase;  
}  
p.iniziali-maiuscole {  
    text-transform: capitalize;  
}
```



Spazio tra testo

Esistono diverse proprietà per definire gli spazi del testo, queste proprietà accettano come valore delle dimensioni.

```
p.textSpacing{  
  text-indent: 15px;      -> indentazione prima riga  
  letter-spacing: 5px;    -> spazio fra le lettere  
  line-height: 25px;     -> altezza di una riga  
  word-spacing: 5px;     -> spazio tra parole  
}
```



Applicare ombra al testo

Utilizzando i CSS possiamo assegnare l'ombra al testo.

```
h1 {  
  color: #000;  
  text-shadow: 2px 2px 5px red;  
}
```

ombra orizzontale

ombra verticale

blur

Titolo h1



Assegnare i font

È possibile assegnare il tipo di font agli elementi attraverso la proprietà **font-family**:

```
font-family: Arial, Helvetica, sans-serif;
```

Il browser cercherà nel dispositivo il font Arial, poi cercherà Helvetica, se non trova nessuno dei due utilizzerà il generico font di sistema sans-serif.

Per applicare dimensioni ai font si usa la proprietà **font-size**:

```
font-size: 16px;
```



Cambiare il display di un elemento

Come abbiamo visto tutti gli elementi dell'html hanno un display preimpostato. Si può cambiare il display attraverso l'omonima proprietà CSS:

```
display: inline;  
display: block;  
display: none;
```

Assegnando display **none** ad un oggetto l'oggetto non verrà visualizzato e non occuperà spazio.

Il display none si utilizza nelle animazioni, nei menù a discesa ecc. ecc.



Display inline-block

Il display:inline-block è una via di mezzo tra i due display, infatti un oggetto con questa proprietà:

- ha un'altezza ed una larghezza
- ha anche margini orizzontali e verticali
- non ha i break-line prima e dopo



Larghezze in percentuale

Quando si utilizzano le larghezze in percentuale dobbiamo tenere presente che ci stiamo riferendo alla grandezza del contenitore:

```
table.orari{  
  width: 600px;  
  margin: auto;  
}  
table.orari td{  
  width: 25%;  
}
```

→ td sarà larga 150px



Larghezze in percentuale

```
table.orari{  
  width: 80%;  
  margin: auto;  
}  
table.orari td{  
  width: 25%;  
}
```

td larga 20% del contenitore di
table

```
<div style="width: 50%;">  
  <div style="width: 50%;"></div>  
</div>
```

div interno largo 25% del
contenitore del div esterno



Il flusso dell'HTML

Il flusso dell'HTML è la modalità con la quale vengono disposti gli oggetti in una pagina.

Gli oggetti con display a blocco vengono messi uno sotto all'altro, gli oggetti con display inline sono principalmente caratteristiche del testo.

Attraverso i CSS possibile modificare il flusso dell'HTML in due modi:

- proprietà float
- posizionamento dell'oggetto



Float

Il float è una proprietà CSS che consente di alterare il flusso dell'HTML.

Serve a disporre gli oggetti, con display a blocco, uno affianco all'altro invece che uno sotto l'altro.

Per disporre gli uno o più oggetti affiancati è necessario che la somma delle loro larghezze non sia minore o uguale al 100%.

Gli oggetti che flottano escono dal flusso dell'HTML, per questo è come se avessero altezza nulla.

Gli oggetti che li seguono si pongono al disotto



Float left

```
#container{  
  width: 80%;  
  margin: 15px auto;  
}  
.fl{  
  float: left;  
  width: 25%;  
  background-color: aliceblue;  
}  
.lg{  
  background-color: lightgreen;  
}
```

```
<div id="container">  
  <div class="fl">1</div>  
  <div class="fl lg">2</div>  
  <div class="fl">3</div>  
  <div class="fl lg">4</div>  
</div>
```

[float-left.html](#)



Float right

```
#container{  
    width: 80%;  
    margin: 15px auto;  
}  
.fr{  
    float: right;  
    width: 25%;  
    background-color: aliceblue;  
}  
.lg{  
    background-color: lightgreen;  
}
```

```
<div id="container">  
    <div class="fr">1</div>  
    <div class="fr lg">2</div>  
    <div class="fr">3</div>  
    <div class="fr lg">4</div>  
</div>
```

[float-right.html](#)



Altezza degli oggetti flottanti

Gli oggetti che flottano non hanno altezza quindi gli oggetti che li seguono si posizionano dove dovrebbero iniziare gli oggetti che flottano.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sodales vel ante eu auctor. Integer viverra ante at odio porta, commodo sodales tortor dictum. Aliquam id ultrices ex. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Suspendisse potenti. Nunc quis neque vel eros consequat ornare in quis arcu. Cras maximus consectetur sem nec tristique. Fusce auctor feugiat felis, bibendum dapibus velit dictum id. Sed ac turpis sit amet massa consectetur egestas.



Clear

Dopo degli oggetti che flottano è sempre necessario inserire un oggetto che abbia la proprietà CSS clear impostata. Attraverso il clear si ripristina il flusso dell'HTML.

`clear: left;` -> agisce su float:left

`clear: right;` -> agisce su float:right

`clear: both;` -> agisce su entrambi i float

Bisogna ricordare di inserire un clear dopo un float



classe .clear

Generalmente quando si ha a che fare con il float si crea un classe clear da poter utilizzare dopo i float:

```
.clear{  
  clear: both;  
}
```

Inseriremo questa classe nei nostri fogli di stile ogni volta che utilizzeremo gli oggetti flottanti.



La proprietà **position**

La proprietà ***position*** serve a specificare il tipo di posizionamento di un elemento. Ci sono cinque possibili valori per questa proprietà:

- static
- relative
- fixed
- absolute
- sticky



La proprietà `position: static`

Questo tipo di posizionamento è quello di default, gli elementi a cui viene assegnato seguono il normale flusso dell'html.



La proprietà `position: relative`

Un oggetto con posizionamento relativo è posizionato come uno static.

A differenza dello static un un oggetto con ***position: relative*** hanno effetto le quattro proprietà di posizionamento:

- `top`
- `right`
- `left`
- `bottom`



La proprietà position: fixed

Un oggetto con posizionamento fixed è posizionato fisso rispetto all'area visibile del browser (viewport). Il che vuol dire che mantiene sempre la stessa posizione anche mentre scorriamo la pagina.

Gli oggetti fixed sono al di fuori del flusso HTML.

Per definire la sua posizione si usano le proprietà top, right, bottom, left.

Se non sono assegnate l'oggetto sarà posizionato in alto a destra (top: 0; left: 0)



La proprietà position: absolute

Un oggetto con posizionamento assoluto è posizionato in maniera assoluta rispetto al primo oggetto che lo contiene a cui è stato assegnato un posizionamento assoluto o relativo.

Se non c'è un oggetto che lo contiene con le caratteristiche dette, sarà posizionato rispetto al body. A differenza di un oggetto fixed, un oggetto con posizionamento assoluto si muoverà durante lo scrolling.

Gli oggetti absolute sono al di fuori del flusso HTML.

Per definire la sua posizione si usano le proprietà top, right, bottom, left. Se non sono assegnate l'oggetto sarà posizionato in alto a destra (top: 0; left: 0).



La proprietà position: sticky

Un oggetto con **position:sticky** è posizionato in base allo scrolling della pagina ed è una via di mezzo tra relative e fixed.

Fino a che non si scolla verso il basso e relative, poi mantiene la sua posizione come un fixed.



Position absolute e fixed

Oltre alle caratteristiche già viste gli oggetti con ***position:fixed*** e ***position:absolute*** hanno una larghezza che non è più pari al 100%, ma hanno una larghezza pari alle dimensioni del loro contenuto.

Questa è una caratteristica comune a tutti gli elementi che escono dal flusso dell'HTML.