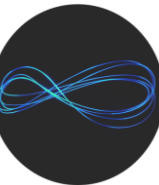


# CORSO HTML5 CSS3



DOCENTE: NICOLA CIACO



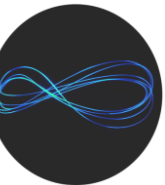


# HTML

## HyperText Markup Language

In informatica è il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web.

È un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C).



# HTML

## Un po' di storia



L'HTML è stato sviluppato verso la fine degli anni '80 da **Tim Berners-Lee** al CERN di Ginevra.

Nel giugno del 1993 Berners-Lee rende pubblica una prima definizione formale dell'HTML e del protocollo HTTP, i due *strumenti* che hanno portato Internet in ogni angolo del pianeta.



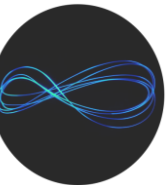
# HTML

## Un po' di storia



Negli anni 90 il [World Wide Web](#) si diffonde rapidamente grazie alla nascita di [Browser](#) commerciali.

Il primo Browser ad entrare nelle case fu [Nescape](#), seguito da Internet Explorer della Microsoft, che per una decina ha avuto l'egemonia del Web.





# HTML

Facciamo un po' di ordine

Abbiamo fin qui usato alcuni termini come:

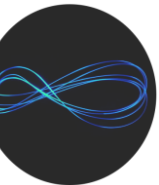
World Wide Web - Internet - HTML - linguaggio di markup - protocollo  
HTTP - Browser

...e da qui in avanti utilizzeremo altri termini come:

Client - Server - Indirizzo IP - Url

Qualcuno di voi starà pensando:

*« Ma che c'entra tutta 'sta roba coi siti web? »*



# HTML

Facciamo un po' di ordine

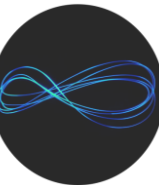
## **World Wide Web: la « ragnatela mondiale »**

Il World Wide Web (WWW) è uno dei principali servizi di Internet che permette di navigare e usufruire di un insieme vastissimo di contenuti collegati tra loro attraverso legami (*link*)

## **Internet**

Da [Wikipedia](#): Internet è una rete mondiale di reti di computer ad accesso pubblico.

Si tratta di un'interconnessione globale tra reti informatiche di natura ed estensione diversa, resa possibile da una suite di protocolli di rete comune chiamata "[TCP/IP](#)", dal nome dei due protocolli principali, il TCP e l'IP, che costituiscono la "lingua comune" con cui i computer connessi ad Internet (gli host) comunicano tra loro, indipendentemente dalla loro sottostante architettura hardware e software.



# HTML

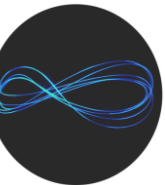
Facciamo un po' di ordine



## L'architettura Client-Server

Indica un'[architettura di rete](#) nella quale genericamente un computer client (o terminale) si connette a un server per l'utilizzo di un certo servizio.

Più semplicemente, limitando il concetto al web, possiamo dire che i server sono computer che contengono i *file fisici* che costituiscono i siti web, e che, tramite il Web, ne permettono l'accesso ai client.



# HTML

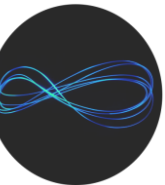
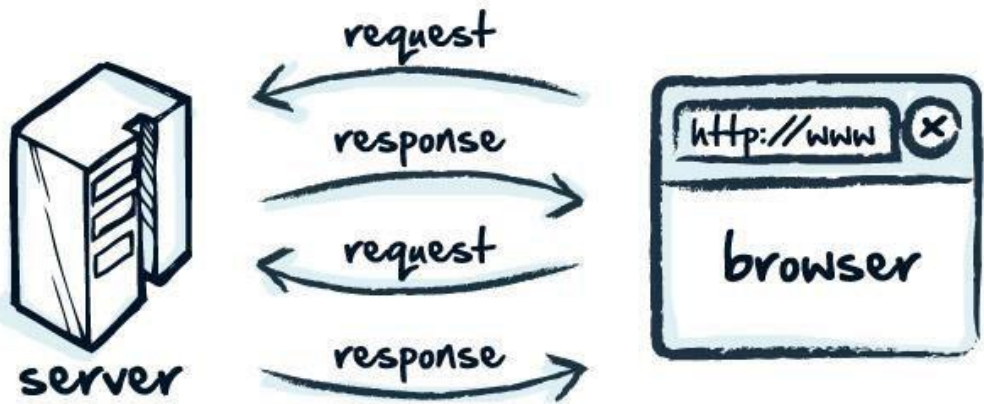
Facciamo un po' di ordine

## Dal Client al Server: andata e ritorno

Da un Client (computer, tablet, smartphone, ecc...) connesso a Internet è possibile **richiedere** un contenuto a un Server, che **restituisce** al Client il contenuto richiesto.

Questo processo è possibile grazie all'insieme di questi 4 elementi:

- Il protocollo [HTTP](#).
- Un' [URL](#) (che in realtà *maschera* anche [l'IP Address](#) del Server).
- Un Browser.
- Il linguaggio di markup HTML.





# HTML

## Facciamo un po' di ordine

### Il protocollo HTTP

L'HyperText Transfer Protocol (HTTP) (protocollo di trasferimento di un ipertesto) è usato come principale sistema per la trasmissione d'informazioni sul web tra Client e Server.

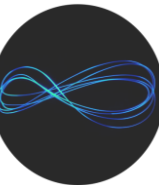
### Indirizzo IP

È un protocollo di interconnessione di reti (Inter-Networking Protocol), nato per interconnettere reti eterogenee per tecnologia, prestazioni e gestione.

L'indirizzo IP potremmo definirlo più semplicemente come l'**indirizzo univoco** di ogni *macchina* connessa a Internet. Ogni indirizzo IP viene descritto (ad oggi) con 4 numeri in base decimale (da 0 a 255) separati dal simbolo "punto". Un esempio di indirizzo IP è 192.0.34.166.

### Url

La Uniform Resource Locator è una **sequenza di caratteri** che identifica **univocamente** l'indirizzo di una risorsa in Internet (un documento, un'immagine, un video) presente su un Server, rendendola accessibile a un client che ne faccia richiesta attraverso un Browser.



# HTML

## Facciamo un po' di ordine

### Il Browser

Il web browser è un software che permette il recupero, la presentazione e la navigazione di risorse web.

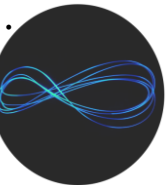
Tali risorse (pagine web, immagini, video, file di testo) possono essere messe a disposizione sul World Wide Web, su una rete locale, o ancora sullo stesso computer dove il browser è in esecuzione.

In parole povere il Browser è il software (installato sul Client) che permette l'accesso al World Wide Web.

Il Browser implementa le regole definite nel protocollo http che gli permettono di dialogare con il server

Inoltre il browser è in grado di interpretare e renderizzare il codice html e di visualizzare i contenuti ipertestuali delle pagine html sullo schermo.

I browser più diffusi sono [Google Chrome](#), [Microsoft Edge](#), [Mozilla Firefox](#), e [Safari](#)



# HTML

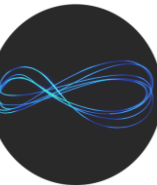
Facciamo un po' di ordine

## ...ed ecco l'HTML!!

Abbiamo iniziato dicendo che l'HTML è « *il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web...* »

A questo punto possiamo descrivere l'HTML dicendo che è « *il linguaggio di markup che, grazie al Browser, è in grado di formattare i contenuti richiesti da un Client a un Server, rendendoli più facilmente leggibili e accessibili dalle persone* ».

Senza l'HTML i contenuti Web sarebbero consultabili solo dagli *addetti del settore*.



# DALL'HTML ALL'HTML5

Un'ultima slide prima di buttarci sul codice...

L'organizzazione che si occupa di standardizzare la sintassi del linguaggio HTML è il [W3C](#) (World Wide Web Consortium).

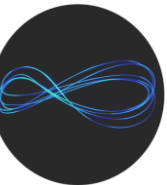
Il W3C ha rilasciato diverse versioni di questo linguaggio che negli anni si è ovviamente evoluto insieme a tutta la tecnologia Web.

Le prime versioni ufficiali del linguaggio rilasciate dal Consorzio W3C sono state l'HTML 2.0, l'HTML 3.2 e l'HTML 4.0.

Nel 1999 l'HTML 4.0 si è evoluto in XHTML.

Con la nascita dell'XHTML è stato introdotto, come standard, l'utilizzo dei CSS (Cascading Style Sheets) a supporto delle pagine HTML. In questi anni lo standard dei CSS si è anch'esso evoluto, raggiungendo ad oggi la terza versione ufficiale (CSS3).

Nel 2006, infine, l'HTML ha visto la nascita della sua quinta versione ufficiale, diventando così HTML5.



# HTML

Cosa occorre prima di iniziare a scrivere codice

**1. Uno o più browser per visualizzare le pagine**

**2. Un editor testuale per scrivere il codice HTML e CSS**

E' possibile usare perfino il blocco note di Windows, o altri semplici editor testuali, per scrivere codice HTML, ma è consigliabile l'utilizzo di un software più evoluto che faciliti la scrittura del codice:

- [Brackets](#) (gratuito)
- [Sublime Text](#) (gratuito)
- [Adobe Dreamweaver](#) (a pagamento, ma molto diffuso nelle Web Agency)
- [Atom](#) (gratuito)
- [Visual Studio Code](#) (gratuito)
- [Una lista di altri editor free](#)



# Le estensioni dei FILE

I **file HTML** sono riconoscibili grazie alle estensioni **.html** oppure **.htm**

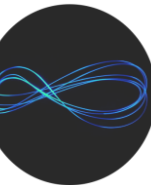
I **file CSS** sono riconoscibili grazie all'estensione **.css**

Nelle pagine web possono essere presenti file di diverso tipo che vengono interpretati dai browser e che completano la struttura dei siti web.

- .js (Javascript e jQuery)
- .jpeg / .jpg / .gif / .png / .svg (immagini)
- .php / .asp / .aspx / .jsp / ... (pagine con codice lato server)
- .doc / .pdf / .zip / ... (file visualizzabili o scaricabili direttamente dal browser)
- .mp3 / .mp4 / .ogg / .webm / ... (file audio e video)

## I nomi dei file

Possono essere alfanumerici, ma è vietato l'uso di caratteri speciali (!,\*,.,#) e si sconsiglia l'uso del carattere *blank* (spazio vuoto). E' convenzione utilizzare esclusivamente lettere minuscole per nominare i file.



# I TAG HTML

All'interno di ogni pagina HTML è presente una serie di marcatori (TAG), che servono a comunicare al Browser la tipologia di formattazione che si intende dare al contenuto della pagina.

I TAG hanno differenti nomi a seconda della loro funzione.

I TAG vanno inseriti tra parentesi angolari (**<TAG>**), la chiusura del TAG viene indicata con uno "/" ("slash"): **</TAG>**. Qualsiasi parola racchiusa tra parentesi angolari verrà interpretata dal Browser come un TAG.

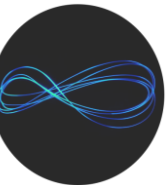
Il contenuto testuale va inserito tra l'apertura e la chiusura del TAG medesimo.

Ecco un esempio: `<tag>contenuto</tag>`

Nel caso volessimo rendere la parola « HTML » in grassetto, il codice da utilizzare è il seguente:

`<strong>HTML</strong>`

Se la parola in grassetto « HTML » facesse parte di una frase, la stessa regola andrebbe applicata in questo modo: `lo adoro l'<strong>HTML</strong>`



# I TAG HTML

## Annidamento

Una caratteristica importante del codice HTML è che i TAG possono essere annidati l'uno dentro l'altro. Nelle strutture web è quasi sempre necessario farlo:

```
<tag1>  
  contenuto 1  
  <tag2>contenuto 2</tag2>  
</tag1>
```

E' bene notare che i TAG si aprono e si chiudono in modo *speculare*. Nell'esempio sopra il **<tag1>** si apre prima del **<tag2>**, ma si chiude dopo. In pratica il **<tag1>** contiene al suo interno il **<tag2>**.

*« Una metafora potrebbe essere quella delle parentesi matematiche {[( ... )]} »*

Va precisato che esistono anche TAG utilizzati senza testo da formattare al proprio interno e che quindi non prevedono una struttura di "apertura" e "chiusura".

Un esempio è il TAG utilizzato per mandare a capo un testo: **<br>**





# I TAG HTML

E' possibile sommare più TAG per una stessa istruzione.  
Il risultato a video sarà che gli effetti dei due diversi TAG si sommano:

```
<strong><em>grassetto - corsivo</em></strong>
```

(i due TAG annidati nell'esempio sopra rendono il testo sia ***grassetto che corsivo***)

L'ordine con cui sono annidati due diversi TAG è indifferente:

```
<em><strong>corsivo - grassetto</strong></em>
```

Non ha senso annidare due TAG identici: i loro effetti non si possono sommare



# I TAG HTML

## Gli attributi dei TAG

All'interno dei TAG possono essere aggiunte ulteriori specifiche, dette attributi (o proprietà).

Prendiamo come esempio il TAG che serve a creare i link nelle pagine: `<a>`

Il TAG `<a>` ha la peculiarità di funzionare esclusivamente se al suo interno è presente l'attributo « `href` », attributo che serve a completare il TAG `<a>` con l'indirizzo di destinazione.

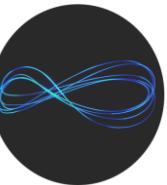
L'attributo all'interno del TAG sarà così strutturato:

- nome dell'attributo (es: `href`)
- segno di uguale (=) *Serve a dividere il nome dell'attributo dal suo valore*
- virgolette (" ") al cui interno è specificato il "valore" (es: `http://www.google.it`)

Un esempio: `<a href="http://www.google.it">Vai al sito google.it</a>`

In questo esempio il TAG `<a>` indica al Browser di essere in presenza di un link.

L'attributo `href="http://www.google.it"` indica al Browser l'indirizzo di destinazione del link.



# I TAG HTML

## Gli attributi dei TAG

E' possibile indicare all'interno dello stesso TAG più attributi differenti.

All'esempio riportato nella slide precedente `<a href="http://www.google.it">Vai al sito google.it</a>` potremmo aggiungere (ad esempio) un attributo che comunichi al Browser di aprire il link (<http://www.google.it>) in una nuova scheda (dello stesso Browser).

Questo attributo ha il nome di « **target** » e il valore necessario a indicare al Browser di aprire il link in una nuova scheda è **"\_blank"**

Il codice HTML sarà quindi scritto in questo modo:

`<a href="http://www.google.it" target= "_blank" >Apri il sito google.it in una nuova scheda del Browser</a>`

E' indifferente se la posizione degli attributi all'interno del TAG viene invertita (il risultato sarebbe identico):

`<a target= "_blank" href="http://www.google.it"> Apri il sito google.it in una nuova scheda del Browser</a>`



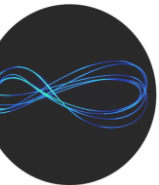
# I TAG HTML

## I problemi con i TAG

Un codice HTML scritto male non può procurare problemi al Browser (ad es. una chiusura inaspettata della pagina), ma può procurare problemi di formattazione delle pagine, per cui il risultato *estetico/funzionale* potrebbe essere compromesso.

### Gli errori più comuni

- Se un TAG che prevede "apertura" e "chiusura" non viene chiuso il suo effetto può essere molto diverso a seconda del tipo di browser utilizzato o del comando specifico
- Un TAG di chiusura senza il corrispondente TAG di apertura non sortisce alcun effetto.
- Se un TAG contiene una sintassi scorretta, viene quasi sempre ignorato dal browser.
- Se un TAG è *interrotto* nella sua *forma convenzionale*, ad esempio se il segno minore "<" viene erroneamente scritto due volte "<<" il browser lo interpreta come testo che mostrerà all'interno della pagina web.



# I TAG HTML

## Maiuscolo o minuscolo?

L'HTML è "case *unsensitive*": è del tutto indifferente se i TAG vengono scritti in maiuscolo o in minuscolo.

<A HREF="[HTTP://WWW.GOOGLE.IT](http://www.google.it)"> o <a href="<http://www.google.it>">

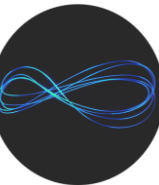
vengono letti allo stesso modo dal Browser.

I Cascading Style Sheets (CSS) sono anch'essi "case *unsensitive*".

Tuttavia, le famiglie di font, le URL a immagini e ad altri riferimenti diretti potrebbero esserlo (*case sensitive*).

**La convenzione è quella di scrivere il codice HTML, e quello dei CSS, prevalentemente in minuscolo.**

Nell'HTML scrivere in maiuscolo o minuscolo, ad ogni modo, non costituisce alcun errore.

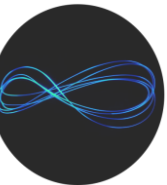


# La struttura base di un documento HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Untitled Document</title>
  </head>
  <body>
    ...contenuto visibile della pagina web ...
  </body>
</html>
```

Un documento HTML è diviso in due parti principali (identificate da due TAG specifici):

**<head> ... </head>** racchiude le informazioni tecniche relative al documento (non visibili agli utenti)  
**<body> ... </body>** racchiude il contenuto vero e proprio delle pagine (visibile dagli utenti)



# La struttura base di un documento HTML

Analizzando l'esempio di codice della slide precedente, la prima istruzione è

```
<!doctype html>
```

E' questa una riga di codice che indica al Browser l'utilizzo delle specifiche del World Wide Web Consortium relative al linguaggio HTML5.

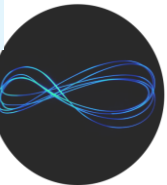
In passato venivano ovviamente utilizzate specifiche diverse e che è possibile trovare ancora in tantissimi siti web. Sia le specifiche dell'XHTML che quelle dell'HTML 4.0 sono ancora del tutto supportate dai Browser.

## **Specifica XHTML**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">
```

## **Specifica HTML 4.0**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```



# La struttura base di un documento HTML

Prima di passare alla formattazione dei contenuti, analizziamo i due TAG contenuti nell'<head>.

L'argomento <head> nel dettaglio verrà ripreso più avanti nel nostro corso.

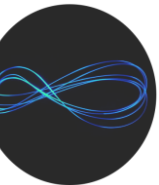
```
<title>Titolo della pagina web</title>
```

Il <title> è il titolo della pagina ed è l'unico\* TAG dell'<head> visibile agli utenti. Compare infatti in alto sulla barra del Browser. È bene compilarlo da subito, per evitare che il sito abbia pagine senza titolo.

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

Serve al Browser per codificare i caratteri del nostro sito web: fare una lista di tutti i caratteri utilizzati al mondo (pensiamo solo alle tante lettere dell'alfabeto che esistono!).

\* In realtà c'è anche un altro TAG presente nell'<head> e visibile all'esterno e serve a mostrare un'immagine accanto al titolo (*favicon*). Vedremo questo TAG più avanti, quando tratteremo le immagini.







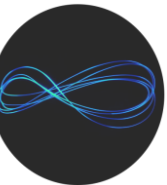
# Formattazione di un documento HTML

## **Formattazione di struttura (o logica)**

Indica un raggruppamento per categoria dei vari elementi del testo, in base alla loro funzione logica: titoli di vario livello (capitolo, paragrafo e sotto paragrafo), didascalia, link, immagini, ecc. La formattazione logica è gestita dal codice HTML.

## **Formattazione di presentazione (fisica)**

Riguarda esclusivamente l'aspetto che vogliamo far assumere alle varie parti del nostro testo: modalità corsivo o grassetto; grandezza, tipo e colore del carattere; allineamento del paragrafo, ecc. La formattazione fisica è gestita dai fogli di stile CSS.



# Formattare il testo

Heading: il titolo di un articolo

```
<h1>Il titolo più importante della pagina</h1>
```

```
<h2>Titolo meno importante</h2>
```

```
<h3>Titolo meno importante</h3>
```

```
<h4>Titolo meno importante</h4>
```

```
<h5>Titolo meno importante</h5>
```

```
<h6>Titolo meno importante</h6>
```

Ogni pagina HTML ben formattata dovrebbe contenere un titolo racchiuso in un TAG **<h1>**



# Formattare il testo

## I paragrafi

```
<h1>Lucy in the sky with diamonds</h1>
```

```
<h2>Beatles</h2>
```

```
<p>Picture yourself in a boat on a river<br>With tangerine trees and marmalade skies<br>Somebody calls you, you answer quite slowly<br>A girl with kaleidoscope eyes</p>
```

```
<p>Cellophane flowers of yellow and green<br>Towering over your head<br>Look for the girl with the sun in her eyes<br>And she's gone </p>
```

```
<p>Lucy in the sky with diamonds<br>Lucy in the sky with diamonds<br>Lucy in the sky with diamonds. Ah... ah...</p>
```

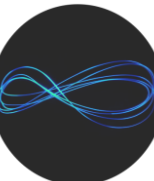
## I TAG utilizzati nell'esempio

**<h1>** Il titolo della pagina

**<h2>** Sottotitolo

**<p>** Paragrafo

**<br>** Il TAG che serve a mandare «accapo» un testo



# Formattare il testo

## Modificare l'aspetto del testo

**<strong>Testo in grassetto</strong>**

*<em>Testo in corsivo</em>*

<u>Testo sottolineato</u>

~~<strike>Testo barrato</strike>~~

$E = mc^2$

$H_2O$

~~<del>Testo cancellato</del>~~

*<cite>Questa è una citazione</cite>*

**<blockquote>**

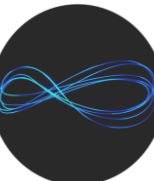
Questa è una citazione lunga almeno un paragrafo e viene mostrata indentata rispetto al resto del testo

**</blockquote>**

`<code>var somma = 1 + 1;</code>`

**<acronym title="Organizzazione delle Nazioni Unite">ONU</acronym>**

**<abbr title="Hypertext Markup Language">HTML</abbr>**



# Formattare il testo

## Gli elenchi ordinati

Gli elenchi ordinati sono contraddistinti dall'enumerazione degli elementi che compongono la lista.

Avremo quindi una serie progressiva ordinata e individuata da numeri.

Il TAG da utilizzare per aprire un elenco ordinato è `<ol>` («ordered list») e gli elementi sono individuati dal tag `<li>` («list item»)

### Codice HTML

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

### Risultato a video

1. Primo elemento
2. Secondo elemento
3. Terzo elemento

Il TAG `<ol>` lascia una riga di spazio prima e dopo il testo che lo circonda (fatta eccezione però l'inclusione di un nuovo elenco all'interno di un elenco preesistente).

Gli elementi dell'elenco sono sempre rientrati di uno spazio verso destra.



# Formattare il testo

## Gli elenchi non ordinati

Gli elenchi non ordinati sono individuati dal TAG `<ul>` e compongono un elenco di elementi non ordinati (comunemente chiamato «elenco puntato»).

Il TAG da utilizzare per aprire un elenco ordinato è `<ul>` («unordered list») e gli elementi sono individuati dal tag `<li>` («list item»).

### Codice HTML

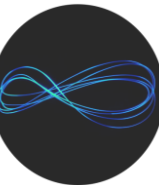
```
<ul>
  <li>Primo elemento</li>
  <li>Secondo elemento</li>
  <li>Terzo elemento</li>
</ul>
```

### Risultato a video

- Primo elemento
- Secondo elemento
- Terzo elemento

Il TAG `<ul>`, come il TAG `<ol>`, lascia una riga di spazio prima e dopo il testo che eventualmente lo circonda (fatta eccezione però l'inclusione di un nuovo elenco all'interno di un elenco preesistente).

Gli elementi dell'elenco sono sempre rientrati di uno spazio verso destra.



# Formattare il testo

## Gli elenchi annidati

E' possibile inserire, all'interno di un elenco, ulteriori elenchi annidati.

### Codice HTML

```
<ul>
  <li>Elemento 1 della prima lista</li>
  <li>Elemento 2 della prima lista
    <ul>
      <li>Elemento 1 della seconda lista
        <ul>
          <li>Elemento 1 della terza lista</li>
          <li>Elemento 2 della terza lista</li>
        </ul>
      </li>
      <li>Elemento 2 della seconda lista</li>
      <li>Elemento 3 della seconda lista</li>
    </ul>
  </li>
  <li>Elemento 3 della prima lista</li>
</ul>
```

### Risultato a video

- Elemento 1 della prima lista
- Elemento 2 della prima lista
  - Elemento 1 della seconda lista
    - Elemento 1 della terza lista
    - Elemento 2 della terza lista
  - Elemento 2 della seconda lista
  - Elemento 3 della seconda lista
- Elemento 3 della prima lista



# I link e l'ipertestualità

Potremmo definire quello dedicato ai link il TAG per antonomasia. Di fatto la più grande innovazione del web è stata quella di permettere agli utenti di passare da un documento all'altro con un semplice click. Il TAG che serve a creare i link è `<a>`

Perché un link funzioni è necessario aggiungere all'interno del TAG `<a>` l'attributo `href` che serve a indicare al Browser l'indirizzo di destinazione.

```
<a href="https://www.udemy.com/course/svg-per-sviluppatori-web">Corso SVG</a>
```

In questo esempio:

- il TAG `<a>` indica al Browser di essere in presenza di un link;
- l'attributo `href="https://www.udemy.com/course/svg-per-sviluppatori-web"` indica al Browser qual è l'indirizzo di destinazione del link;
- La frase compresa tra `<a>` e `</a>` (Corso SVG) è il testo visibile al visitatore e che sarà *linkabile*.





# I link e l'ipertestualità

Il TAG `<a>` non si limita a *linkare* due pagine dello stesso sito web.  
Le possibilità di link sono molto più ampie:

- Link tra due pagine di due siti web diversi
- Link a due punti diversi della stessa pagina (ancora)
- Link a file .doc o .pdf (sono visualizzati nel Browser)
- Link a file scaricabili come .zip o .rar
- Link a immagini .jpg, .gif, .png (sono visualizzati nel Browser)
- Link a file audio .mp3, .ogg (sono visualizzati nel Browser)
- Link a file video .mp4, .ogg, .webm (sono visualizzati nel Browser)

I link, utilizzando un'opportuna sintassi, riescono persino a interagire con il client, invitandolo a mettere in funzione un particolare software installato. Ecco alcuni esempi:

- Link a indirizzo email (invita il Client ad aprire il software di gestione della Posta Elettronica, ad es. Outlook)
- Link a numero di telefono (consente allo smartphone di far partire automaticamente la telefonata)
- Link a Skype (consente al client di far partire la chiamata tramite Skype)



# I percorsi assoluti e relativi

I siti web sono organizzati in cartelle e sottocartelle, così come organizziamo di solito i documenti presenti nei nostri computer.

Per questo è necessario *sapersi muovere* tra le cartelle utilizzando il codice HTML.

Per fare una prima distinzione tra un percorso assoluto e uno relativo potremmo dire che:

- nel primo caso l'indirizzo indicato è indipendente da qualsiasi pagina web venga richiamato;
- nel secondo caso l'indirizzo indicato è in relazione con la pagina web da cui viene richiamato.

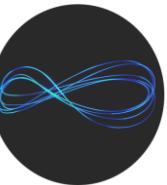
## Un semplice esempio.

Se mi chiedessero «Dove si trova il *MINISTERO DEGLI ESTERI*?»

1. Potrei rispondere in maniera **assoluta**: «in Piazzale della Farnesina Roma, N. 1 ».
2. Potrei rispondere in maniera **relativa** (mettiamo il caso che io e il mio interlocutore ci trovassimo in via Maqueda): «vai fino al semaforo laggiù e girà a sinistra. Da lì sono 200 metri »

Nel **primo caso** l'indicazione è valida in qualsiasi parte del mondo.

Nel **secondo** è valida solo se gli interlocutori condividono la stessa posizione geografica (sono entrambi in via Maqueda).



# I percorsi assoluti e relativi

Un esempio di **percorso assoluto** sul web:

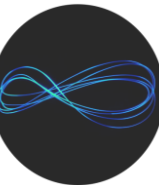
Il link all'immagine del marchio della scuola LABFORTTRAINING (immagine collocata in una particolare cartella del sito [www.labforttraining.it](http://www.labforttraining.it))

```
<a href="http://www.labforttraining.it/immagini/logo-labforttraining.png">Il marchio LABFORTTRAINING</a>
```

<b>http://</b>	Indica al browser di utilizzare il protocollo per navigare nel web (l'http)
<b><a href="http://www.labforttraining.it/">www.labforttraining.it/</a></b>	Indica di fare riferimento al sito <a href="http://www.labforttraining.it/">www.labforttraining.it</a> (di fatto la cartella principale del sito web)
<b>immagini/</b>	Indica che la risorsa indicata si trova all'interno di una cartella chiamata "immagini"
<b>logo-labforttraining.png</b>	Indica che l'immagine da collegare si chiama "logo-labforttraining" e che è di tipo ".png"

Il carattere / (slash) indica di fatto al Browser di entrare in una cartella.

In questo caso l'indirizzo web è raggiungibile da qualsiasi pagina web del mondo (percorso assoluto).



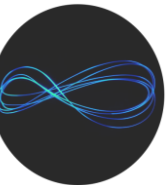
# I percorsi assoluti e relativi

## I percorsi relativi.

Sono molto più usati dei percorsi assoluti.

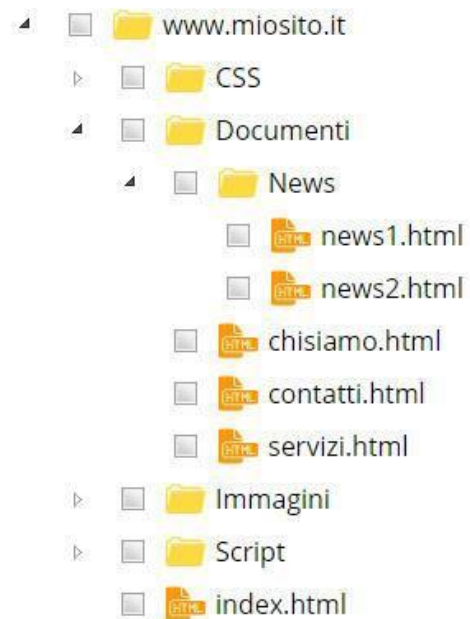
Queste le motivazioni principali che ci spingono a lavorare con i percorsi relativi:

1. spesso vi troverete a fare riferimento a documenti situati nel vostro stesso sito, quindi ogni file condivide lo stesso punto di partenza (*www.miosito.it*);
2. quando state sviluppando il sito sul vostro computer (cioè "in locale") e ancora non avete pubblicato il codice su un server web, non è possibile lavorare con i percorsi assoluti (a meno non desideriate dover modificare tutti i percorsi in fase di pubblicazione del sito ☺) ;
3. Un codice impostato su percorsi relativi può essere replicato o spostato facilmente, senza dover fare grosse modifiche a tutti i link.



# I percorsi assoluti e relativi

## I percorsi relativi



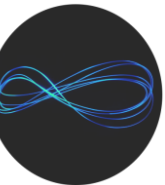
I percorsi relativi fanno riferimento alla posizione degli altri file rispetto al documento in cui ci si trova in quel momento.

Nell'esempio illustrato qui accanto poniamo il caso che dalla pagina **index.html** vogliamo far riferimento al file **news2.html**, che si trova all'interno della directory «**News**», a sua volta contenuta dalla directory «**Documenti**».

La sintassi del link sarà questa:

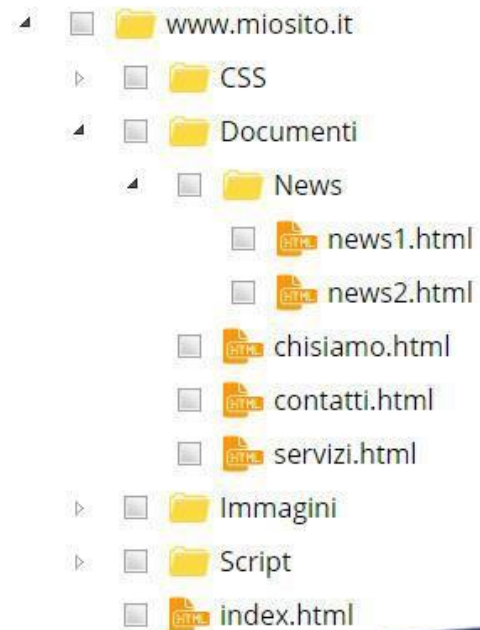
```
<a href="Documenti/News/news2.html">
```

La sintassi **nomefolder/** indica al browser di entrare in una cartella grazie all'utilizzo dello **/** (slash)



# I percorsi assoluti e relativi

## I percorsi relativi

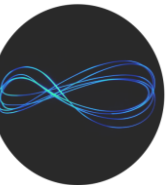


Nell'esempio illustrato qui accanto poniamo il caso contrario a quello precedente, ovvero dalla pagina **news2.html**, che si trova all'interno della directory «**News**» a sua volta contenuta dalla directory «**Documenti**», vogliamo raggiungere la pagina **index.html**

La sintassi del link sarà questa:

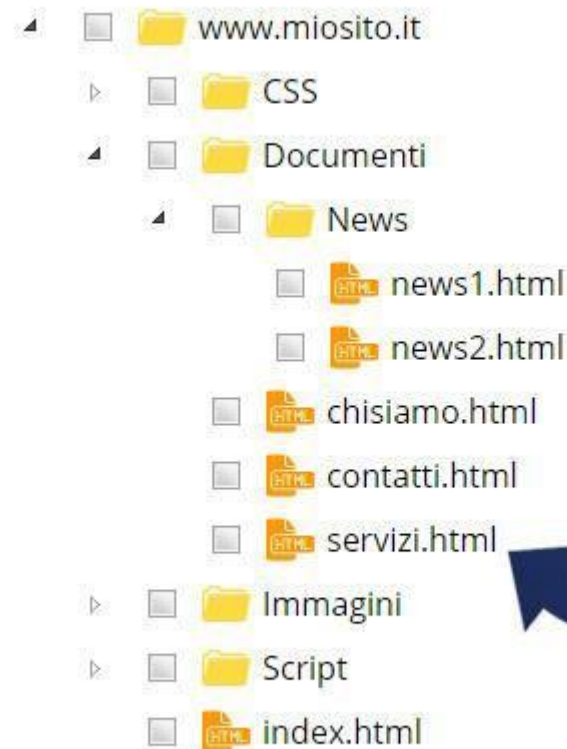
```
<a href="../../index.html">
```

La sintassi **../** indica al browser di uscire dalla cartella in cui si trova.



# I percorsi assoluti e relativi

## I percorsi relativi



Nell'esempio qui accanto poniamo il caso in cui dalla pagina **news2.html**, che si trova all'interno della directory «**News**» a sua volta contenuta dalla directory «**Documenti**», vogliamo raggiungere la pagina **servizi.html** contenuta dalla directory «**Documenti**».

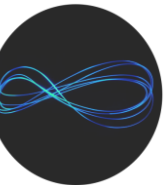
La sintassi del link sarà questa:

```
<a href="../servizi.html">
```

In questo caso potremmo utilizzare anche la sintassi:

```
<a href="/Documenti/servizi.html">
```

Nel caso (come questo) il link inizi con uno **/** (slash) stiamo indicando al Browser di prendere come indirizzo di partenza la cartella principale del nostro sito (detta **root**), in questo esempio [www.miosito.it](http://www.miosito.it)



# I link e l'ipertestualità

Il TAG `<a>` non serve solo a *linkare* due risorse diverse, ma potrebbe collegare due punti della stessa pagina. In questo caso i link prendono il nome convenzionale di **ancore**

La sintassi prevede l'utilizzo di due TAG `<a>` nella stessa pagina: il primo come link, il secondo come punto di ancoraggio.

Il primo link sarà scritto in questo modo:

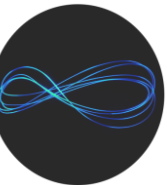
```
<a href="#ancora1">Leggi gli approfondimenti</a>
```

Il simbolo **#** (cancelletto) serve ad indicare al Browser di cercare un'ancora nella stessa pagina, invece di *linkare* a risorse esterne.

La sintassi della zona di ancoraggio sarà questa:

```
<a name="ancora1">Qui c'è la mia ancora</a>
```

- Il TAG `<a>` che determina l'ancoraggio utilizza l'attributo **name** invece dell'**href**.
- Il nome dell'ancora e del link devono essere uguali, l'unica differenza sta nel **#** (cancelletto) che precede il nome nel TAG `<a>` che determina il link.
- L'ancora `<a name=""></a>` può non contenere testo (può quindi essere lasciata vuota).
- Il nome delle ancore dovrà sempre contenere almeno una lettera (alfanumerico). Un nome composto solo da numeri potrebbe non funzionare bene.





# I link e l'ipertestualità

## Gli attributi dei link

Se volessimo indicare al Browser di aprire il link in una nuova finestra, è possibile farlo utilizzando l'attributo **target**

Il valore da indicare, per far in modo che il browser apra la pagina in una nuova finestra, è “\_blank”

```
<a href="https://stackoverflow.com" target= "_blank">Apri il sito STACKOVERFLOW in una nuova pagina</a>
```

Un altro attributo molto utilizzato nei link è **title**.

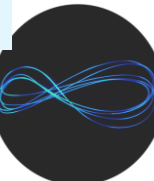
Serve a far visualizzare un testo esplicativo per l'elemento a cui l'attributo si riferisce. Questa spiegazione aggiuntiva favorisce l'accessibilità del sito anche ai disabili (per esempio alle persone ipovedenti).

Grazie a questo attributo, lasciando il cursore del mouse per qualche secondo sul link, vedrete comparire un'etichetta con il testo specificato nell'attributo.

```
<a href="https://stackoverflow.com" title= " Vai al sito di STACKOVERFLOW, a code forum">apri stackoverflow</a>
```

Un esempio con l'unione di entrambi gli attributi:

```
<a href=" https://stackoverflow.com " target="_blank" title="Questo link apre una nuova pagina del Browser in cui sarà visualizzato il sito di STACKOVERFLOW, a code forum ">stackoverflow.com</a>
```



# I link e l'ipertestualità

## Link ad altre risorse

Se volessimo linkare un file .zip presente nel nostro sito web, basta indicare al Browser il percorso di destinazione

```
<a href="https://www.stackoverflow.com/archivio/file.zip">Scarica la cartella compressa</a>
```

Nota che il percorso sopra non esiste veramente (quindi se provi ad aprirlo ottieni un errore 404), è solo esemplificativo

L'estensione del file (.html / .jpg / .zip / .mp3) viene interpretata dal browser autonomamente, per cui saprà sempre come comportarsi (se quindi visualizzare un file, un'immagine, aprire il lettore multimediale, ecc..).

E' possibile indicare al browser di interagire con alcuni software presenti sul client.

## Linkare a una mail

```
<a href="mailto:servizioclienti@euronics.it" title="Scrivi una mail a EURONICS">servizioclienti@euronics.it</a>
```

## Linkare un numero di telefono (molto utile per gli smartphone)

```
<a href="tel:+390289663366" title="Telefona a EURONICS">0289663366</a>
```

## Avviare una chiamata Skype

```
<a href="skype:euronics.it" title="Usa skype per contattare EURONICS">Skype</a>
```



# Le immagini

Il TAG che serve a inserire un'immagine in una pagina HTML è `<img>`

Perché il TAG funzioni è necessario aggiungere l'attributo `src=""` che serve a indicare al Browser il percorso per raggiungere l'immagine.

E' raccomandabile inserire anche un ulteriore attributo, necessario a scrivere una descrizione dell'immagine: `alt=""`

```

```

Keyword	Descrizione
<code>&lt;img&gt;</code>	Abbreviazione di "image", è <b>il nome del TAG</b>
<code>src</code>	Abbreviazione di "source", è il percorso (URL) in cui il Browser troverà il file da mostrare
<code>alt</code>	È il "testo alternativo" che appare solo se, per qualche motivo, il browser non riesce a mostrare l'immagine. Questo attributo non è obbligatorio, ma risulta fondamentale per l'accessibilità e per i motori di ricerca

Poiché il TAG `<img>` non può contenere alcun testo, o un altro TAG, non possiede un corrispettivo TAG di chiusura.

I formati di immagine utilizzati sul web sono [.jpg](#) (o jpeg), [.gif](#) e [.png](#).

La qualità delle immagini visualizzata non supera i 72 dpi, per cui è inutile utilizzare immagini a 300 dpi: avrebbero la stessa resa estetica a fronte di un peso molto superiore.



# Le immagini

Il peso delle immagini incide moltissimo sul caricamento delle pagine web, per cui è necessario utilizzare forme di compressione che, a fronte di una resa estetica ottimale, le alleggeriscano il più possibile. I software di grafica hanno la funzione per esportare immagini in *formato per il web*. Esistono anche alcuni siti web che offrono la possibilità di comprimere le immagini.

Tra tutti segnalo [tinypng.com](http://tinypng.com) (per le immagini png) e [tinyjpg.com](http://tinyjpg.com) (per le immagini jpg) e [compressor.io](http://compressor.io)

Le immagini, all'interno di un documento HTML, vengono impaginate come fossero dei testi:

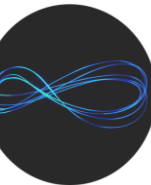
## Esempio di codice

Il marchio `` è composto da un cerchio celeste che racchiude un lampetto bianco.

## Risultato a video



È possibile modificare il posizionamento delle immagini, rispetto al testo che le circonda, grazie ai fogli di stile CSS



# Le immagini

## Il favicon

Il favicon è quella immagine che spesso troviamo accanto al *title* della pagina nella *tab* del Browser.

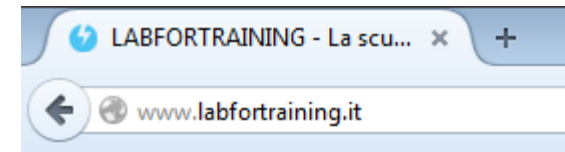
Per visualizzare l'immagine in quella particolare posizione del Browser non utilizzeremo il TAG `<img>` ma il TAG `<link>` corredato da due attributi: `href` (invece che l'attributo `src`) e `rel` (che indica il *tipo di legame* che si sta istaurando)\*.

Il TAG `<link>` è inserito all'interno del TAG `<head>`

## Esempio di codice

```
<head>
  <title>LABFORTRAINING - La scuola di web a Roma</title>
  <link rel="icon" href="http://www.labfortraining.it/favicon.ico" />
</head>
```

## Risultato a video



In questo esempio abbiamo utilizzato un tipo di file `.ico`, utilizzato spesso per il favicon. Possono, in alternativa, essere utilizzati i formati di immagine `.png` e `.gif`

\* L'attributo `rel` determina diversi tipi di legame istaurati attraverso il TAG `<link>`.

Scopriremo più avanti che, per esempio, anche i fogli di stile vengono richiamati attraverso il TAG `<link>`, ma in quel caso il tipo di legame è `rel="stylesheet"`



# Le tabelle

Le tabelle sono elementi utilissimi per formattare a righe e colonne informazioni che ripetono lo stesso schema.

## Esempio di codice

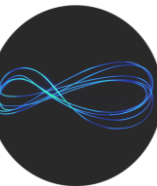
```
<table border="1">
  <tr>
    <td>I elemento della prima riga</td>
    <td>II elemento della prima riga</td>
    <td>III elemento della prima riga</td>
  </tr>
  <tr>
    <td>I elemento della seconda riga</td>
    <td>II elemento della seconda riga</td>
    <td>III elemento della seconda riga</td>
  </tr>
  <tr>
    <td>I elemento della terza riga</td>
    <td>II elemento della terza riga</td>
    <td>III elemento della terza riga</td>
  </tr>
</table>
```

## Risultato a video

I elemento della prima riga	II elemento della prima riga	III elemento della prima riga
I elemento della seconda riga	II elemento della seconda riga	III elemento della seconda riga
I elemento della terza riga	II elemento della terza riga	III elemento della terza riga

Tag	Descrizione
<code>&lt;table&gt;</code>	apre la tabella
<code>&lt;tr&gt;</code>	" <i>table row</i> ": indica l'apertura di una riga
<code>&lt;td&gt;</code>	" <i>table data</i> ": indica l'apertura di una cella all'interno di una riga

L'attributo `border="1"` utilizzato in questo esempio rende visibile il bordo della tabella, va comunque precisato che è obsoleto e che tutte le caratteristiche estetiche (bordo compreso) vengono oggi gestite dai fogli di stile



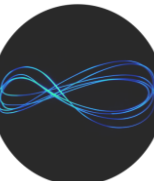
# Le tabelle

Esistono dei TAG (opzionali) che possono essere utilizzati per organizzare a gruppi le righe delle tabelle:

<code>&lt;caption&gt;</code>	Indica l'intestazione (il titolo) della tabella
<code>&lt;thead&gt;</code>	Indica l'intestazione delle singole celle: è la parte iniziale della tabella
<code>&lt;tfoot&gt;</code>	Indica il piede, la conclusione della tabella
<code>&lt;tbody&gt;</code>	Indica il corpo della tabella, quella che contiene il contenuto vero e proprio

			} <code>&lt;caption&gt;</code>
			} <code>&lt;thead&gt;</code>
			} <code>&lt;tbody&gt;</code>
			} <code>&lt;thead&gt;</code>

```
<table>
  <caption></caption>
  <thead>
    <tr> <td></td><td></td><td></td> </tr>
  </thead>
  <tbody>
    <tr> <td></td><td></td><td></td> </tr>
    <tr> <td></td><td></td><td></td> </tr>
    <tr> <td></td><td></td><td></td> </tr>
  </tbody>
  <tfoot>
    <tr> <td></td><td></td><td></td> </tr>
  </tfoot>
</table>
```



# Le tabelle

## Raggruppare celle con rowspan e colspan

È possibile raggruppare le celle all'interno delle colonne. Per ottenere questo risultato è necessario specificare che una cella deve occupare il posto di 2 (o più) colonne. In questo caso nel TAG `<td>` si aggiunge l'attributo **colspan**, specificando come valore il numero di celle che devono essere occupate.

### Esempio di codice

```
<table>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td colspan="2"></td> </tr>
</table>
```

Con l'attributo **rowspan** (da aggiungere sempre all'interno di `<td>`) è invece possibile creare delle celle che occupino più di una riga.

### Esempio di codice

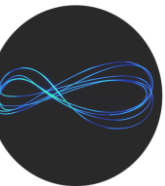
```
<table>
  <tr> <td></td> <td rowspan="2"></td> <td></td> </tr>
  <tr> <td></td> <td></td> </tr>
</table>
```

### Risultato a video

	<td colspan="2">	

### Risultato a video

	<td rowspan="2">	



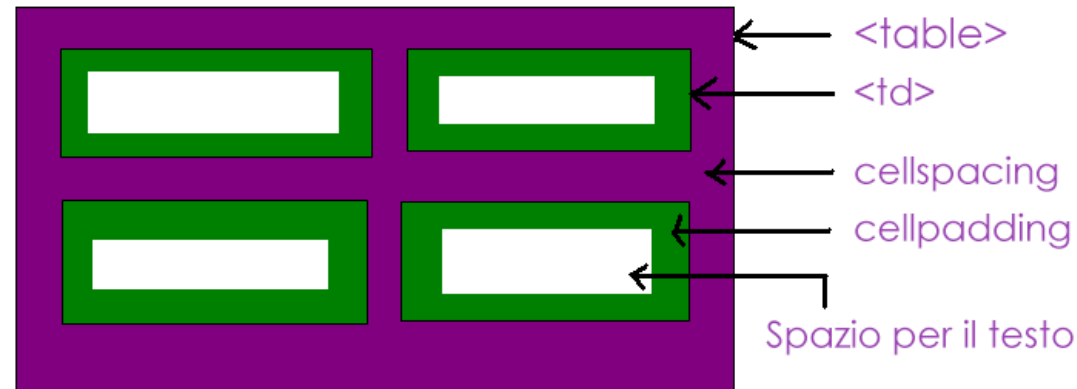


# Le tabelle

## Gli attributi cellpadding e cellspacing

L'attributo **cellspacing** determina lo spazio che divide tra esse le celle.

Il **cellpadding**, invece, determina lo spazio che c'è tra il perimetro di ogni singola cella e il proprio contenuto.



Anche se sono proprietà ormai obsolete, visto che la gestione estetica dei contenuti HTML è demandata ai CSS, va comunque preso in considerazione che questi due attributi, *di default*, non hanno un valore pari a zero, per cui incidono (anche se in minima parte) sulla resa estetica delle tabelle. E', per cui, si può aprire un TAG `<table>` in questo modo anche se la gestione con css è più moderna

```
<table cellpadding="0" cellspacing="0"></table>
```



# Creare i Layout dei siti web

## Elementi blocco (block) ed elementi in linea (inline)

Una pagina HTML è composta, di fatto, da un insieme di rettangoli (box). Non importa che si tratti di paragrafi, titoli, tabelle o immagini: si tratta sempre di box rettangolari.

Non tutti i box sono però uguali.

Alcuni contengono altri box al loro interno; altri sono invece contenuti all'interno dei primi.

Alcuni, se si trovano in mezzo al testo, lasciano che esso scorra loro intorno senza interrompere il suo flusso e senza andare a capo. Questa semplice considerazione ci offre la rappresentazione della fondamentale distinzione tra elementi blocco - **block** (quelli predisposti a contenere altri box) ed elementi in linea - **inline** (non predisposti a contenere altri box).

Il comportamento degli elementi **block** è riconoscibile perché, quando usati, questi TAG (di default) vanno a capo e occupano l'intera riga del contenitore genitore.

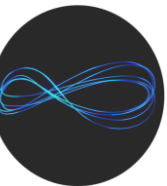
Gli elementi **inline**, invece, non vanno a capo e occupano lo spazio necessario a contenere il proprio contenuto.

Per fare un esempio:

Il TAG **<h1>** è di tipo **block** (ogni titolo va a capo e, il testo che segue, va a capo a sua volta)

Il TAG **<strong>** è di tipo **inline** e rende in grassetto il testo in se contenuto (senza mandare a capo alcuna parola).

Queste caratteristiche native dei TAG possono essere modificate arbitrariamente con l'uso dei CSS.

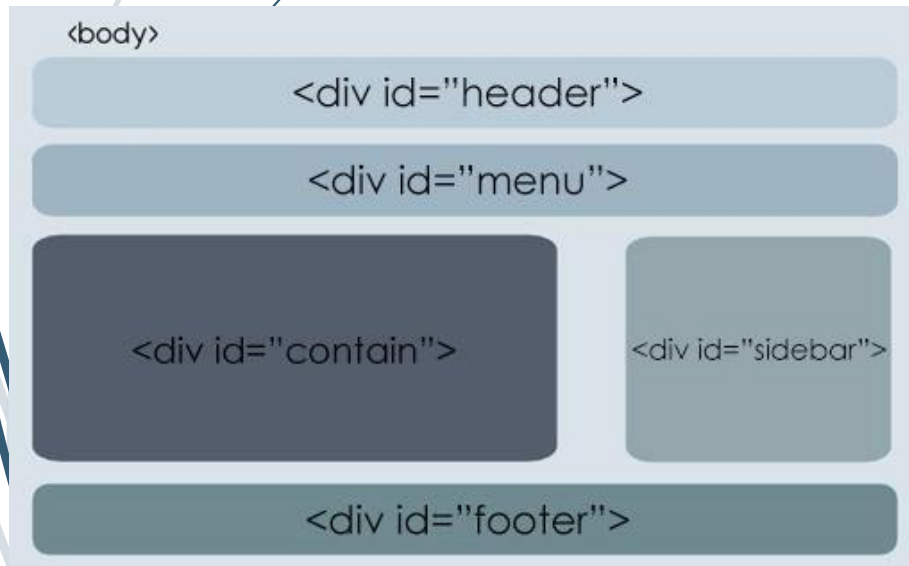


# Creare i Layout dei siti web

## Il TAG <div>

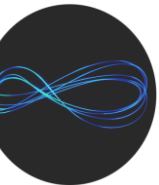
I siti web sono strutturati in parallelepipedi, ovvero hanno una *struttura* ossea formata da tanti rettangoli (o quadrati) nei quali vengono disposti la maggior parte dei contenuti (menu di navigazione, testi, header e footer, immagini e video,...)

Fino all'avvento dell'HTML5 il TAG che ha assolto sempre a questa funzione è il **<div>**



Qui accanto è rappresentato un tipico layout HTML4. Va precisato, però, che il TAG **<div>** serve esclusivamente a creare la **formattazione logica** e non quella fisica (estetica), che è invece demandata ai fogli di stile CSS.

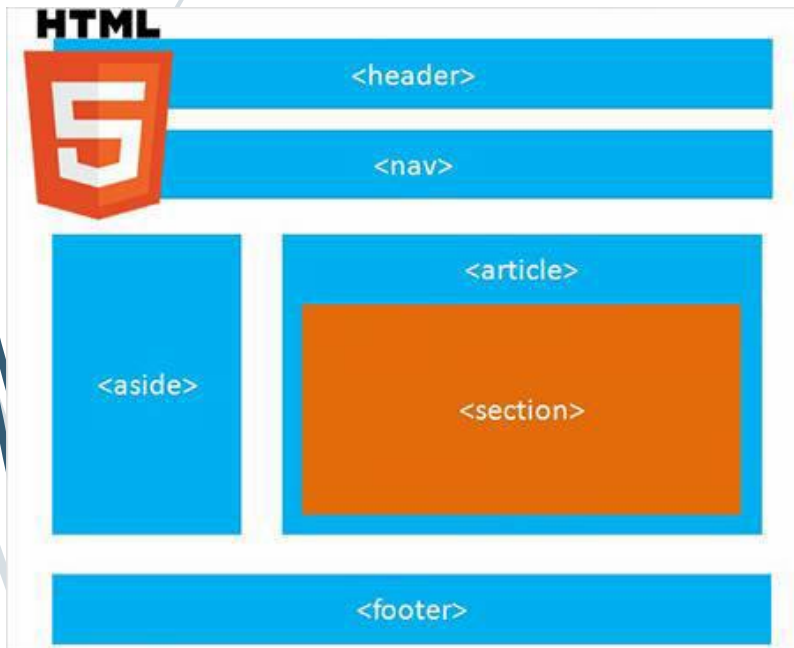
L'attributo **id** utilizzato nell'esempio specifica un **ID univoco** per un elemento HTML (ovvero il valore inserito deve essere univoco all'interno del documento HTML). L'attributo **id** non aggiunge alcuna funzionalità visibile al TAG HTML ma è molto utilizzato per *puntare* a uno stile in un file CSS o per essere *manipolato* tramite JavaScript e jQuery.



# Creare i Layout dei siti web

## I TAG semantici

Con l'HTML5 sono nati dei nuovi TAG (*semantici*) utili a creare i Layout, e che hanno in parte sostituito l'utilizzo del TAG `<div>`. Sull'esperienza fatta dai Web Designer, sono stati rilasciati dei nuovi TAG che hanno la caratteristica di avere dei nomi di più facile comprensione.

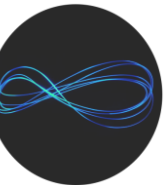


Qui accanto è rappresentato un tipico layout HTML5.

Come si può notare non compare più il TAG `<div>` (al quale è necessario associare un ID che lo renda univoco), ma sono utilizzati dei nuovi TAG.

Va comunque specificato che anche questi nuovi TAG non includono alcun tipo di formattazione estetica, ma rientrano sempre nell'ambito di quella logica.

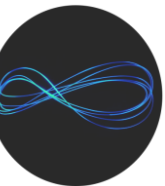
Potremmo affermare (forzando un po' la mano) che questi nuovi TAG sono dei `<div>` con dei nomi semanticamente più comprensibili.



# I nuovi TAG (strutturali)

Ecco un elenco di alcuni dei principali TAG HTML5 utilizzati per creare i layout web.

TAG	Descrizione	
<b>&lt;article&gt;</b>	Definisce un articolo in un document HTML	<a href="#">Esempio</a>
<b>&lt;aside&gt;</b>	Definisce un'area in cui raccogliere informazioni aggiuntive al contenuto (ad es. notizie correlate)	<a href="#">Esempio</a>
<b>&lt;details&gt;</b>	Definisce un'area in cui aggiungere info aggiuntive che un utente può decidere di vedere oppure no	<a href="#">Esempio</a>
<b>&lt;dialog&gt;</b>	Definisce una dialog box (un popup) all'interno di una pagina	<a href="#">Esempio</a>
<b>&lt;figcaption&gt;</b>	Definisce la didascalia per il TAG <figure>	<a href="#">Esempio</a>
<b>&lt;figure&gt;</b>	Definisce un contenuto di tipo immagine, illustrazione, digramma, ecc...	<a href="#">Esempio</a>
<b>&lt;footer&gt;</b>	Definisce il footer (piè di pagina ) di un documento HTML o di una <section>	<a href="#">Esempio</a>
<b>&lt;header&gt;</b>	Definisce l'header (intestazione) di un documento HTML o di una <section>	<a href="#">Esempio</a>
<b>&lt;main&gt;</b>	Definisce la parte principale del documento	<a href="#">Esempio</a>
<b>&lt;mark&gt;</b>	Definisce un testo evidenziato	<a href="#">Esempio</a>
<b>&lt;nav&gt;</b>	Definisce l'area in cui è presente il menu di navigazione del sito web	<a href="#">Esempio</a>
<b>&lt;section&gt;</b>	Definisce una sezione all'interno di un documento HTML	<a href="#">Esempio</a>
<b>&lt;summary&gt;</b>	Definisce il titolo (intestazione) per il TAG <details>	<a href="#">Esempio</a>
<b>&lt;time&gt;</b>	Definisce un contenuto di tipo data/ora	<a href="#">Esempio</a>



# I TAG *multimediali*

Una delle novità più importanti dell'HTML5 è stata l'introduzione di alcuni TAG in grado di gestire, direttamente dal Browser (e senza l'utilizzo di plugin esterni), i file video e audio.

TAG	Descrizione
<audio>	Definisce un contenuto di tipo audio
<source>	Definisce il file audio o video da inserire nei TAG <audio> e <video>
<video>	Definisce un contenuto di tipo video

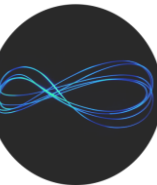
## Esempio di codice del TAG <video>

```
<video>  
  <source src="file-video.mp4">  
</video>
```

## Esempio di codice di TAG <audio>

```
<audio>  
  <source src="file-audio.mp4">  
</audio>
```

Questi tag sono supportati dai browser da una certa versione in su (Firefox 3.5+, Safari 3.1+, Opera 10.5+, Chrome 5.0+, IE9+).



# I TAG *multimediali*

## Formati dei file multimediali

Nei browser moderni mp4 ed mp3 sono supportati. Tuttavia anche gli altri formati sono abbastanza supportati con il vantaggio di essere più piccoli. Quindi puoi pubblicare almeno 2 file (di estensione diversa) Dello stesso file, ma *fortunatamente* sarà il Browser a scegliere autonomamente quale leggere. Come Sviluppatore devi avere l'accortezza di mettere nell'ordine prima i formati più leggeri come webm ed ogg.

Se hai un video in un formato, per generarti gli altri puoi usare degli strumenti gratuiti come [VEED](#)

## Esempio di codice del TAG **<video>**

```
<video>
  <source src="file-video.webm">
  <source src="file-video.ogg">
  <source src="file-video.mp4">
</video>
```

## Esempio di codice di TAG **<audio>**

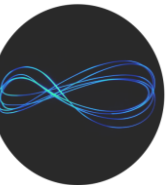
```
<audio>
  <source src="file-audio.webm">
  <source src="file-audio.ogg">
  <source src="file-audio.mp3">
</audio>
```



# I TAG *multimediali*

I TAG `<video>` e `<audio>` hanno a corredo anche una serie di attributi dedicati.

TAG	Descrizione
<b>autoplay</b>	L'attributo autoplay consente di riprodurre un file multimediale non appena la pagina è caricata senza dover necessariamente cliccare sul play.
<b>autobuffer</b>	L'attributo autobuffer, utilizzato solo quando autoplay non è usato (altrimenti è ignorato) istruisce il browser sulla gestione del buffer. Lo stesso browser scarica in background il file multimediale al caricamento della pagina in modo tale che quando l'utente lo avvia, quest'ultimo sarà riprodotto quasi interamente (dipende dalla sua lunghezza) senza interruzioni.
<b>controls</b>	L'attributo controls (se dichiarato), consente di visualizzare i controlli: Play, Stop, barra di avanzamento e (per i video) il pulsante per visualizzare il video a tutto schermo.
<b>loop</b>	Attributo booleano che, se valorizzato, consente di ricaricare e riprodurre ciclicamente un file multimediale.
<b>poster</b>	Presente solo per il TAG <code>&lt;video&gt;</code> . Con questo attributo è possibile definire la URL di un'immagine da mostrare prima e/o dopo la riproduzione del video o, comunque, quando il filmato non è disponibile. E' molto utile poiché la possibilità che un video non sia disponibile, sia stato rimosso o altro non è poi così remota.
<b>preload</b>	Se questo attributo è presente, il file multimediale verrà precaricato al caricamento della pagina, in modo tale da essere pronto per una riproduzione senza attese. Se l'attributo autoplay è stato dichiarato, preload viene ignorato.





# Il TAG <iframe>

Il TAG `<iframe>` è una *finestra* che permette di avere, nel proprio sito, una *vista* sul contenuto di un altro sito web.

E' un TAG molto utilizzato nel caso di condivisione di contenuti.

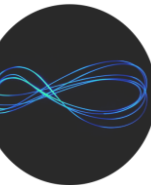
Qualche esempio?

In quanti siti avete visto un video di YouTube?

Grazie a questo particolare TAG è possibile *incorporare* all'interno della pagina web di un proprio sito un video che è invece fisicamente presente sul sito di YouTube.

Ecco un esempio di TAG `<iframe>` che permette di visualizzare un video in qualsiasi altra pagina web:

```
<iframe width="420" height="315" src="https://www.youtube.com/embed/hTWKbfoikeg?rel=0" frameborder="0" allowfullscreen></iframe>
```



# I moduli

## Il TAG <form>

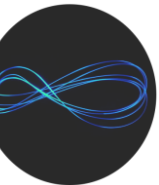
La *grandezza* del Web è senz'altro la possibilità, per chiunque, di interagire con i siti web. La possibilità, per esempio, di iscriversi a servizi di vario tipo (basti pensare alla nostra area riservata LABFORYOU), e soprattutto di essere parte di veri e propri *mondi virtuali*, come Facebook.

Per poter utilizzare questi servizi è necessario, prima di tutto, avere un sistema che permetta di raccogliere i dati dell'utente (nome, cognome, email, ...). Per fare questo si utilizzano i moduli (form).

La raccolta dati è solitamente organizzata in due parti:

- una pagina principale che contiene i vari campi dei form, che consentono all'utente di effettuare delle scelte, scrivere del testo, inserire un'immagine;
- una pagina secondaria che viene richiamata dalla principale e che si occupa di processare e raccogliere i dati. Si tratta di una pagina di programmazione che si trova sul server (php, aspx, jsp).

La pagina principale è invece composta da codice HTML.



# I moduli

## Il TAG <form>

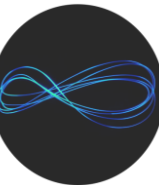
<form> è un TAG di blocco che racchiude in se tutti gli elementi con i quali l'utente potrà interagire.

L'utilità del TAG <form> consiste proprio nel raccogliere tutte le informazioni inserite e inviarle a una specifica pagina.

Questo un esempio di apertura del TAG <form>:

```
<form name="miaform" action="pagina-di-arrivo.php" method="post"></ifram>
```

Attributo	Descrizione
<b>name</b>	È l'attributo che ci consente di dare un nome arbitrario alla form. E' un attributo opzionale che però può diventare molto importante nel caso si utilizzi javascript
<b>action</b>	E' l'attributo che serve a indicare alla form dove (a quale pagina) inviare i dati raccolti
<b>method</b>	Consente di decidere la modalità di invio dei dati. Il valore <b>post</b> (il più utilizzato) consente di inviare i dati senza che essi vengano visualizzati nella barra degli indirizzi del browser. Il valore alternativo a <b>post</b> è <b>get</b> . In questo caso oi dati viaggeranno in chiaro sulla barra degli indirizzi del browser



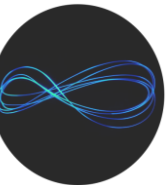
# I moduli

## Il TAG <input>

Il TAG **<input>** è di gran lunga il più utilizzato nelle form. Grazie all'attributo **type** può avere diverse forme e utilità. Ecco una tabella riassuntiva per i diversi valori attribuibili a **type**:

```
<input type="text" >
```

Valore	Descrizione
<b>button</b>	È un bottone generico che non ha nessun effetto preimpostato, ma può essere molto utile con javascript
<b>checkbox</b>	Consente all'utente di fare delle scelte multiple
<b>file</b>	Permette all'utente di aprire il filesystem del proprio client e selezionare un file (da <i>uploadare</i> tramite modulo)
<b>hidden</b>	E' un campo nascosto che può contenere un valore utile al programmatore (non è modificabile dall'utente)
<b>image</b>	È un bottone generico che ha la possibilità di essere visualizzato come immagine
<b>password</b>	E' un campo che <i>maschera</i> (con dei pallini) il valore inserito dall'utente
<b>radio</b>	Consente all'utente di fare delle scelte esclusive
<b>reset</b>	E' un pulsante che permette di resettare l'interno modulo. In pratica cancella tutti i dati eventualmente inseriti
<b>submit</b>	E' il pulsante di invio dei dati
<b>text</b>	L'input per antonomasia, quello che viene utilizzato per richiedere il nome, il cognome, ecc...



# I moduli

## Il TAG <input>

Al TAG **<input>** con l'avvento dell'HTML5 sono state assegnati nuovi valori. Alcuni di essi ancora non sono supportati da tutti i Browser.

**email** `<input type="email">`

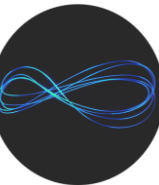
E' utilizzato per indicare gli indirizzi email. Permette una validazione lato client nativa.

E' molto utile soprattutto per gli smartphone e i tablet, in quanto suggerisce automaticamente di visualizzare la tastiera impostata per gli indirizzi email.

Oltre ai nuovi type sono stati rilasciati dei nuovi attributi molto interessanti, funzionanti con tutti i TAG input

Attributo	Descrizione
<b>maxlength</b>	(esiste anche nell'HTML4) serve a limitare il numero di caratteri da inserire nell'<input>
<b>required</b>	Rende il campo obbligatorio
<b>placeholder</b>	Inserisce una <i>label</i> nel campo, che sparisce quando si inserisce un valore reale
<b>disabled</b>	Disabilita l'inserimento di qualsiasi dato. Il campo è visibile ma non può essere modificato.
<b>autofocus</b>	Seleziona automaticamente il selettore del mouse nel campo

```
<input type="email" placeholder="Inserisci la tua email" maxlength="50" required autofocus>
```



# I moduli

## Il TAG <input>

**Number** `<input type="number" min="18" max="90" placeholder="Indica la tua età">`

E' utilizzato per indicare un valore numerico.

Gli smartphone e i tablet visualizzeranno la tastierina numerica.

Nell'esempio sono presenti anche gli attributi **min** e **max** che servono a limitare in range la cifra da inserire

**Range** `<input type="range" min="1" max="0">`

Realizza uno slider tramite il quale è possibile scegliere un valore.

Il codice non si differenzia molto dal tipo **number**

**Url** `<input type="url" required>`

Da utilizzare per indicare indirizzi web

Gli smartphone e i tablet visualizzeranno la tastierina che contiene il punto e le principali estensioni (.com, .it, ecc...).



# I moduli

## Il TAG <input>

### Il tipo *data*

Ha comunque la caratteristica di mostrare un *datepicker* che permette di selezionare data e ora  
*datetime* è deprecato

```
<input type="date">
```

Type	Descrizione
<b>date</b>	La forma più comune, consente di scegliere una data
<b>month</b>	Limita ai soli mese e anno
<b>week</b>	Consente di selezionare il numero corrispondente alla settimana nell'arco dell'anno.
<b>time</b>	Consente di selezionare un orario
<b>datetime</b>	Consente di selezionare data e ora (DEPRECATO!)
<b>datetime-local</b>	Consente di selezionare data e ora. L'uso, rispetto al precedente <i>datetime</i> , è indicato per applicazioni che fanno uso di orari internazionali



# I moduli

## Il TAG <input>

**Pattern** `<input type="text" pattern="([0-9]{5})" required placeholder="Inserisci il CAP">`

Serve a costruire dei controlli di validazione personalizzati.

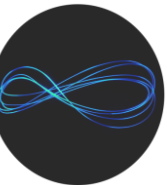
Nell'esempio sopra per evitare errori di compilazione del CAP, il pattern così impostato consente di inserire SOLO NUMERI per un numero ne superiore ne inferiore di 5.

Un sito di riferimento per i pattern: <http://html5pattern.com>

### Altri attributi molto importanti per i campi input

Attributo	Descrizione
<b>name</b>	E' fondamentale per dare un nome arbitrario ai diversi campi input. Questo attributo è necessario per dar catturare ai linguaggi lato server i valori inseriti dagli utenti
<b>value</b>	Serve per impostare un valore di default a un campo input

```
<input type="text" name="nome_utente" value="John">
```





# I moduli

## Gli altri TAG dei moduli

**textarea** `<textarea name="commento"></textarea>`

A differenza del tag `<input type="text">` questo TAG permette di inserire testo su più righe (ad es. un post su facebook)

**select**

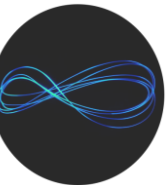
Realizza una tendina che permette una selezione esclusiva

```
<select name="component-select">
  <option selected value="Component_1">Component_1</option>
  <option value="Component_2">Component_2</option>
  <option value="Component_3">Component_3</option>
  <option value="Component_4">Component_4</option>
</select>
```

**button** `<button type="button">Questo è un pulsante</button>`

E' stato introdotto con l'HTML e sostituisce l' `<input type="button">`

E' possibile passare, come valore del type, `submit` e `reset`



# Il TAG <head>

Nella testa dei documenti HTML è possibile inserire informazioni supplementari che non vengono visualizzate dall'utente (fatta eccezione per i TAG <title> e <link rel="icon"> che abbiamo già visto). Queste informazioni completano però il codice HTML

## I METATAG

Il TAG <meta> apporta informazioni aggiuntive ai Browser e/o ai motori di ricerca. Viene completato da altri attributi

### Description

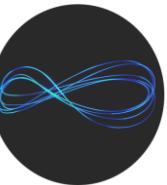
```
<meta name="description" content="BOOLEAN è la scuola di web a Milano che organizza corsi di Web Design, Web Development, Web Content Management e Graphic Design">
```

Serve a raccogliere una descrizione breve del contenuto della pagina web (ideale tra i 140 e i 160 caratteri). Non viene visualizzata dagli utenti ma viene mostrata dai Motori di Ricerca all'interno della SERP.

### Keywords

```
<meta name="keywords" content="web design, web development, corsi web, corsi copy, corso grafica, grafica web, corso html, corso html5, corso responsive, responsive web design, social media marketing, seo, corsi seo, corso di seo, corso php, corsi php a roma, corso jquery, javascript e jquery" />
```

Come la description serve esclusivamente per i motori di ricerca. Va detto che oggi è un meta-tag sempre meno utilizzato perché di fatto non porta benefici all'indicizzazione



# Il TAG <head>

## Author

```
<meta name="author" content="John Doe">
```

Serve per dare informazioni sull'autore del documento

## Copyright

```
<meta name="copyright" content="BOOLEAN srl">
```

Serve per dare informazioni sul copyright

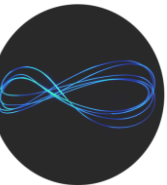
## Refresh

```
<meta http-equiv="refresh" content="5">
```

Serve a ricaricare la pagina automaticamente. Il valore inserito nell'attributo content indica il numero di secondi che intercorre tra l'uno e l'altro refresh

```
<meta http-equiv="refresh" content="5;url=http://gazzettadellosport.it">
```

Con questa seconda soluzione è possibile, dopo 5 secondi (in questo esempio), reindirizzare la pagina verso il link suggerito



# Il TAG <head>

Nell'<head> vanno inseriti, solitamente, anche i collegamenti a risorse esterne, come fogli di stile e file javascript/jquery

## Collegare i fogli di stile

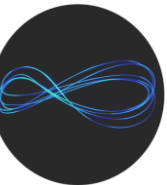
```
<link rel="stylesheet" href="css/file.css">
```

Il TAG <link> con l'attributo rel="stylesheet" indicano il collegamento a un foglio di stile esterno. Nell'attributo href viene indicato il percorso in cui il Browser potrà trovare il file css.

## Collegare uno script javascript

```
<script type="text/javascript" src="script/file.js"></script>
```

Il TAG <script> con l'attributo type="text/javascript" indica un collegamento di tipo javascript. Nell'attributo src viene indicato il percorso in cui il Browser troverà il file js



# Open Graph Protocol

Il Web Semantico è un argomento trattato da moltissimo tempo, ma concretamente si è fatto ancora troppo poco per far sì che questa idea prenda realmente piede. Facebook, si è mosso da qualche anno in questa situazione implementando il così detto Open Graph Protocol per sopperire ad alcune mancanze ancora troppo evidenti nel mondo del Web.

L'Open Graph Protocol, sfruttando i meta-tag inclusi nelle pagine web, permette a ciascuna pagina presente in Internet di essere rappresentata sotto forma di un oggetto, caratterizzato da alcune proprietà chiave, e di poter essere inclusa in quello che viene chiamato "grafo sociale", ovvero una sorta di mappatura di tutte le pagine Internet e di quello che esse rappresentano e contengono.

L'Open Graph Protocol è fortemente apprezzato perché, nonostante realizzato da una piattaforma di tiratura mondiale, non preclude nessun utilizzo "privato" anche da altri portali o applicazioni in genere che possono ottenere le informazioni da ciascuna pagina web senza prescindere da Facebook. Esistono infatti altre piattaforme che usano l'Open Graph Protocol per realizzare un proprio "grafo sociale" interno (ad esempio LinkedIn e Twitter).

Per configurare una pagina web come un oggetto in grado di entrare a far parte di un "grafo sociale" più grande è necessario impostare una serie di meta-tag all'interno del TAG `<head>` della pagina con le informazioni richieste dallo specifico TAG.



# Open Graph Protocol

Alcuni tra i meta-dati più utilizzati:

Meta-dato	Descrizione
<b>og:title</b>	rappresenta il titolo del nostro oggetto nel "grafo sociale" (attenzione non è in nessun modo collegato al <title> della pagina)
<b>og:type</b>	rappresenta il tipo di oggetto
<b>og:image</b>	url di un immagine che rappresenta l'oggetto
<b>og:url</b>	url che identifica l'oggetto e che rappresenta l'id univoco che l'oggetto avrà
<b>og:description</b>	rappresenta una breve descrizione dell'oggetto
<b>og:locale</b>	Rappresenta la <i>localizzazione</i> della pagina web
<b>og:site_name</b>	Il nome del sito web che incapsula l'oggetto Open Graph

```
<head>
  <meta property='og:title' content='HTML5 e CSS3' />
  <meta property="og:type" content="article" />
  <meta property='og:image' content='http://www.epicode.com/corsi/icone/corso-web-developer.png' />
  <meta property='og:url' content='https://epicode.com/it/corso-web-developer/' />
  <meta property='og:description' content='Corso base di Html5 e CSS3 a Roma' />
  <meta property="og:locale" content="it_IT" />
  <meta property="og:site_name" content="EPICODE - Scuola di Web a Roma" />
</head>
```

