

# Corso HTML5

Impariamo a usare HTML5  
conoscendone le basi



## Scopo del corso

Queste slide nascono dalla mia curiosità verso HTML5 e dall'esperienza che ho accumulato nel corso degli anni.

La struttura base del corso ripercorre i passi seguiti da w3schools, li arricchisce con nuovi esempi ed una serie di slide in italiano.

# HTML

## Hyper Text Markup Language

Cos'è HTML?

- È il linguaggio standard per la creazione di pagine web tramite marcatori.
- Il suo scopo è quello di descrivere la struttura delle pagine.
- Identifica una serie di elementi strutturati.
- La sua interpretazione necessita di un browser.

# HTML – strumenti

Il portale che useremo durante il corso è

<https://codepen.io>

*CodePen is a social development environment. At its heart, it allows you to write code in the browser, and see the results of it as you build. A useful and liberating online code editor for developers of any skill, and particularly empowering for people learning to code. We focus primarily on front-end languages like HTML, CSS, JavaScript, and preprocessing syntaxes that turn into those things.*

Iscrivetevi e seguite il profilo creato apposta per il corso

<https://codepen.io/matteobaccan>

# HTML – editor

Editor

Codepen.io

Notepad

Notepad++

VisualStudio Code

Va bene qualsiasi editor, non visuale, meglio se con syntax highlighter e code completion

Le slide e i sorgenti del corso, liberamente ispirati a <https://www.w3schools.com> e costantemente aggiornati, sono disponibili a questo indirizzo

<https://github.com/matteobaccan/CorsoHTML>

# Browser

I browser sono dei programmi necessari alla corretta visualizzazione di una pagina HTML.

Esempi di browser sono:

- Chrome
- Firefox
- Edge
- Safari
- Opera

Senza i browser non sarebbe possibile visualizzare le pagine web, vista anche la loro complessità e natura dinamica.

# Com'è fatto un browser?

Un browser è composto da una serie di componenti che permettono la visualizzazione delle pagine HTML.

**Browser engine:** address bar, pulsanti di navigazione, viewport, print

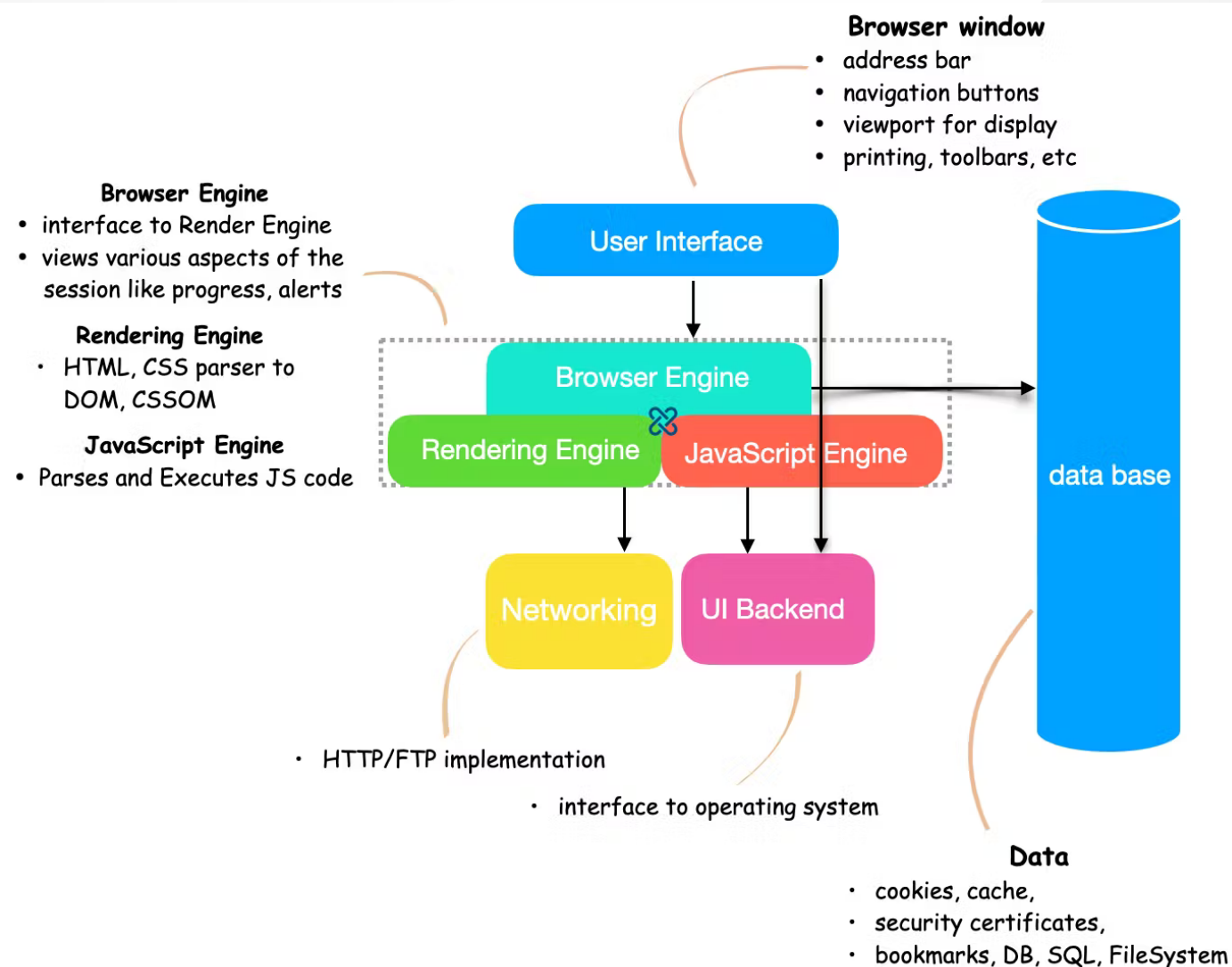
**Rendering engine:** chiama la rendering engine per visualizzare il contenuto richiesto

**Javascript interpreter:** interpreta il codice javascript

**Networking:** HTTP requests, SSL, etc

**UI backend:** disegna i widget nativi come alert, form di caricamento file etc

# Componenti del browser

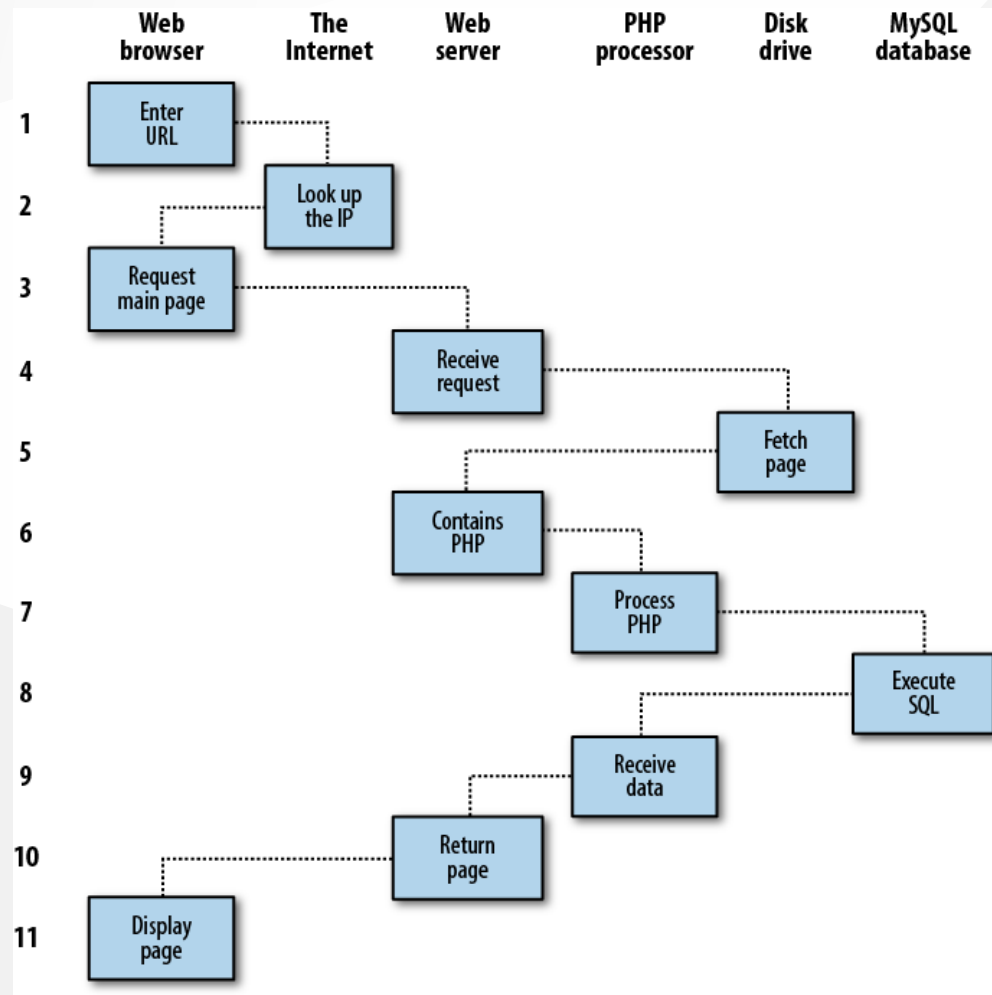




# Flusso di richiesta

- Inserimento dell'indirizzo di destinazione
- Risoluzione dell'indirizzo via DNS
- Verifica della presenza in cache della pagina/risorsa
- Eventuale download
- Gestione del codice d'errore
- Rendering della pagina

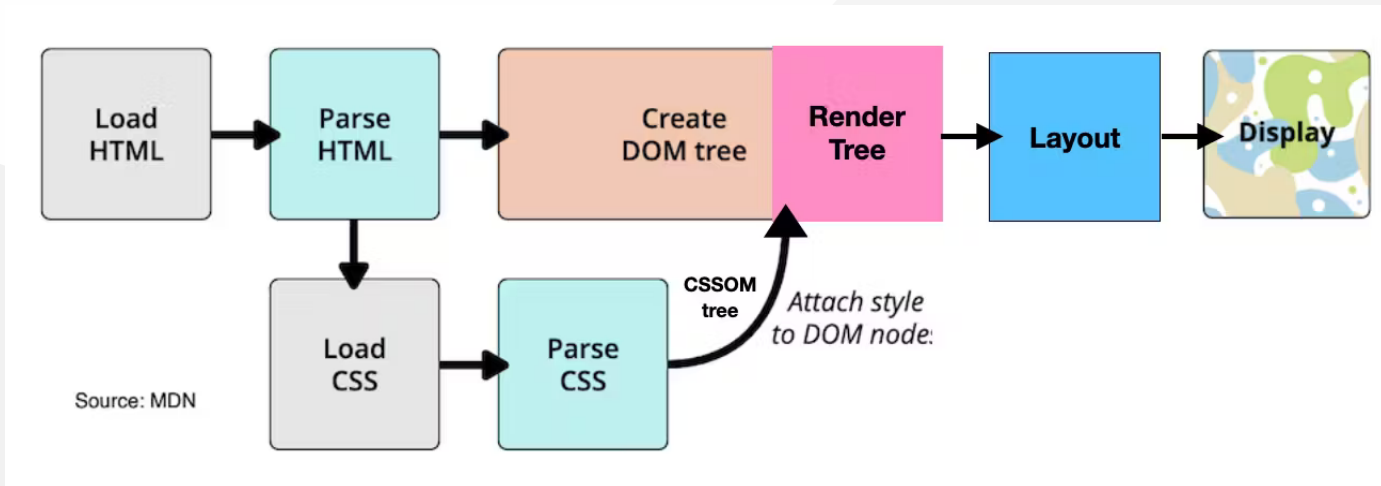
# Gestione delle richieste



<https://www.oreilly.com/library/view/learning-php-mysql/9781449337452/ch01s01.html>

# Rendering delle pagine

Il rendering è quel processo che permette di visualizzare una pagina HTML.



# HTML - esempio minimo di pagina HTML

```
<!DOCTYPE html>

<html lang="it">

<head>
<title>Titolo della pagina</title>
</head>

<body>
<h1>Intestazione</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>

</html>
```

# HTML - Elementi - TAG

Le pagine HTML sono composte da una serie di elementi, chiamati TAG, formattatine nel seguente modo:

```
<nometag> contenuto </nometag>
```

Un tag è identificato da una sequenza di caratteri, con una sintassi di apertura e chiusura

All'interno del tag viene poi inserito il suo contenuti, che a sua volta potrebbe essere del testo o altri tag

# HTML - Elementi - empty TAG

## Empty TAG

```
<br> <hr> <img>
```

Sono tag pensati per non avere un corpo.

In base al fatto che siano usati come HTML5 o XHTML5 potrebbero essere scritti in formato diverso

```
<br/> <hr/> <img/>
```

# HTML – attributi

## Attributi

```
<a href="https://www.baccan.it">Visitate Baccan.it</a>
```

I tag HTML possono avere degli attributi.

Gli attributi aggiungono delle informazioni ai tag

La loro struttura è per **chiave** = "**valore**"

# HTML – definizione

```
<!DOCTYPE html>
```

Identifica il fatto che il documento sia in formato HTML5



# HTML primo tag

```
<html lang="it">  
  
...  
  
</html>
```

É il primo elemento di una pagina html, all'interno di questo elemento è contenuta la struttura della pagina

# HEAD

```
<head>  
  
...  
  
</head>
```

È il tag che contiene le informazioni inerenti alla pagina

- Titolo
- Descrizione
- Tag per motori di ricerca
- Tag per Social Network

## HEAD - esempio

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

È il titolo della pagina: viene visualizzato all'interno del browser per dare un nome al tab che state visualizzando o dai motori di ricerca per rappresentare la vostra pagina

# BODY

```
<body>  
  
...  
  
</body>
```

Definisce il corpo della pagina html: testi, paragrafi, immagini, link, tabelle etc

# BODY - Intestazione

```
<body>  
  <h1>Intestazione</h1>  
</body>
```

È il primo degli heading disponibili in HTML5

## BODY - esempio

```
<body>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
</body>
```

Definisce un paragrafo

# Heading

```
<h1>Intestazione 1</h1>
```

```
<h2>Intestazione 2</h2>
```

```
<h3>Intestazione 3</h3>
```

Questo tipo di tag sono intestazioni

Hanno 6 livelli di profondità e servono per intestare nel modo corretto delle parti di pagina. Il loro utilizzo è utilizzato dai motori di ricerca per determinare delle parti importanti di pagina

# Heading size

Le dimensioni di default dei **font-size** degli heading sono

h1 - 2em

h2 - 1.5em

h3 - 1.17em

h4 - 1em

h5 - .83em

h6 - .67em

dove em deriva dalla parola "emphasis" e identifica una larghezza relativamente al carattere corrente.

Se il carattere corrente è di 16px il valore di 2em identifica 32px

Come peso viene invece assegnato un **font-weight: bolder**



# Style

L'attributo **style** viene utilizzato per assegnare un stile ad un determinato tag

```
<h1 style="color:red;">Intestazione</h1>  
<h2 style="color:blue;">Intestazione</h2>  
<h3 style="color:yellow;">Intestazione</h3>
```

Ad esempio per cambiare il colore del carattere di un testo

# Formattazione

```
<b> - Grassetto  
<strong> - Importante  
<i> - Italico  
<em> - Enfatizzato  
<mark> - Marked  
<small> - Smaller  
<del> - Cancellato  
<ins> - Inserted  
<sub> - Subscript  
<sup> - Superscript
```

# Citazioni

Esistono alcuni elementi utilizzati per poter fare delle citazioni

<code>&lt;abbr&gt;</code>	Defines an abbreviation or acronym
<code>&lt;address&gt;</code>	Defines contact information for the author/owner of a document
<code>&lt;bdo&gt;</code>	Defines the text direction
<code>&lt;blockquote&gt;</code>	Defines a section that is quoted from another source
<code>&lt;cite&gt;</code>	Defines the title of a work
<code>&lt;q&gt;</code>	Defines a short inline quotation

# Commenti

Esiste la possibilità di inserire del testo che verrà scartato in automatico del browser. Lo scopo del commento è puramente documentativo.

```
<!-- Questo è un commento -->
```

```
<p>Questo è un paragrafo</p>
```

# Colori

In HTML esistono 140 colori standard che possono essere chiamati per nome

[https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)

Ad ognuno è associata una sequenza RGB

# Colori - sfondo, testo e bordo

I colori possono essere usati in vari contesti, come colore di sfondo, colore di testo o di bordo

```
<h1 style="background-color:DodgerBlue;">Intestazione</h1>  
<p style="background-color:Tomato;">Paragrafo</p>
```

```
<h1 style="color:Tomato;">Intestazione </h1>  
<p style="color:DodgerBlue;">Paragrafo</p>  
<p style="color:MediumSeaGreen;">Paragrafo</p>
```

```
<h1 style="border:2px solid Tomato;">Intestazione</h1>  
<h1 style="border:2px solid DodgerBlue;">Intestazione</h1>  
<h1 style="border:2px solid Violet;">Intestazione</h1>
```

# Colori - varianti

I colori possono essere indicati in vari formati RGB, HEX, HSL, RGBA e HSLA

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>  
<h1 style="background-color:#ff6347;">...</h1>  
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>  
  
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>  
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

# Colori - RGB

Un colore RGB rappresenta le tre componenti di luce RED, GREEN e BLUE

I colori RGBA sono una estensione di RGB con l'aggiunta del canale Alpha per l'effetto di opacità

In HTML possiamo indicare un RGB con questa formula

```
rgb(red, green, blue)
```

Ogni parametro identifica l'intensità del colore in una scala da 0 a 255

I colori possibili sono 16.777.216

Il nero è rappresentato da `rgb(0, 0, 0)`

Il bianco è rappresentato da `rgb(255, 255, 255)`



# Colori RGB

RGB Calculator



rgb(255, 0, 0)

#ff0000

hsl(0, 100%, 50%)

R:

G:

B:

[https://www.w3schools.com/colors/colors\\_rgb.asp](https://www.w3schools.com/colors/colors_rgb.asp)

## Colori - HEX


Un colore HEX è la rappresentazione esadecimale di un colore RGB, usando la seguente griglia

**#RRGGBB** dove RR è il red, GG è il green e BB è il blue

Si tratta del valore esadecimale precedentemente rappresentato da 0 a 255

# Colori HEX Calculator

Hex Calculator



rgb(255, 0, 0)  
hsl(0, 100%, 50%)

R:

G:

B:

[https://www.w3schools.com/colors/colors\\_hexadecimal.asp](https://www.w3schools.com/colors/colors_hexadecimal.asp)

# Colori - HSL

HSL è acronimo di hue, saturation e lightness (tinta, saturazione e luminosità)

HSLA è la variante con Alpha channel

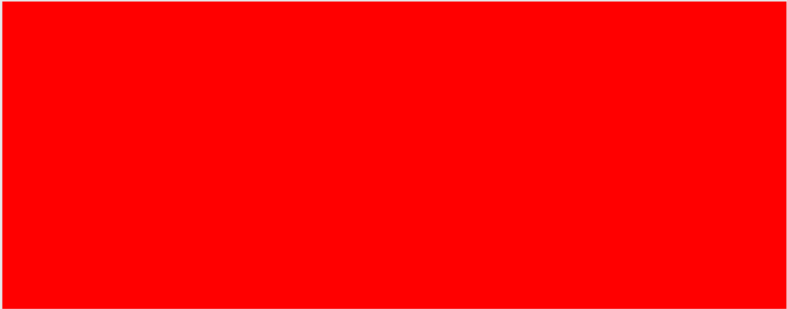
Hue è il grado di colore in una ruota da 0 a 360: 0 è il rosso, 120 il verde e 240 il blue.

Saturation è una percentuale da 0% che identifica un'ombra grigia e 100% che rappresenta il colore pieno.

Lightness è a sua volta una percentuale che va dal 0% che è il nero a 100% che è il bianco.


# Colori HSL Calculator


HSL Calculator



`hsl(0, 100%, 50%)`  
`rgb(255, 0, 0)`  
`#ff0000`

H:  

S:  

L:  

[https://www.w3schools.com/colors/colors\\_hsl.asp](https://www.w3schools.com/colors/colors_hsl.asp)

# Cos'è il CSS?

CSS è acronimo di **Cascading Style Sheets**, sono fogli che vengono utilizzati per formattare le pagine web.

Con i CSS è possibile controllare il colore, il carattere, la dimensione del testo, la spaziatura tra gli elementi, il modo in cui gli elementi sono posizionati e disposti, quali immagini di sfondo o colori di sfondo devono essere utilizzati, o le diverse visualizzazioni in base alle dimensioni dello schermo

Da notare che **cascading** identifica il fatto che uno stile applicato a un elemento padre si applicherà anche a tutti gli elementi figli all'interno dell'elemento padre

# CSS

I CSS possono essere aggiunti ai documenti HTML in 3 modi:

Inline - utilizzando l'attributo style all'interno degli elementi HTML

Interno - utilizzando un elemento `<style>` nella sezione `<head>`

Esterno: utilizzando un elemento `<link>` per collegarsi a un file CSS esterno

## CSS Inline

```
<h1 style="color:blue;">Una intestazione blue</h1>  
<p style="color:red;">Un paragrafo rosso</p>
```



# CSS Interno

...

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

...

# CSS Esterno

...

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

...

# styles.css

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

# Link

I link HTML sono collegamenti ipertestuali.

Lo scopo è quello di passare da una pagina all'altra

Essendo elementi cliccabili, quando viene spostato il mouse su un link il puntatore viene trasformato in una piccola mano.



# Link - a

## Sintassi

Il tag **<a>** ha la seguente sintassi:

**<a href="url">** testo del collegamento **</a>**

L'attributo più importante di **<a>** è l'attributo **href** , che indica la destinazione del collegamento.

Il testo del collegamento è la parte visibile all'interno della pagina HTML

Facendo clic sul testo del collegamento, il browser verrà indirizzato all'indirizzo **URL**

## Link - colori

Esiste una convenzione fra browser per visualizzare i collegamenti con un colore diverso, in base ad alcune caratteristiche

Un collegamento **non visitato** è sottolineato e **blu**

Un collegamento **visitato** è sottolineato e **viola**

Un collegamento **attivo** è sottolineato e **rosso**

Per ottenere l'attivo basta tenere cliccato il mouse sul link

# Target

L'attributo di **target**

Se non indichiamo un **target** il browser visualizzerà il link nella pagina corrente.

Per modificare questo comportamento è necessario specificare un altro **target**

L'attributo **target** può avere uno dei seguenti valori:

**\_self** - DEFAULT. Apre il documento nella stessa finestra/scheda in cui è stato cliccato

**\_blank** - Apre il documento in una nuova finestra o scheda

**\_parent** - Apre il documento nel frame genitore

**\_top** - Apre il documento in tutto il corpo della finestra

**“nomeframe”** Apre il documento in un determinato frame

## Link - esempi

```
<iframe src="http://www.acmenovara.it" name="A">  
<p>ACME Novara</p>  
</iframe>
```

```
<iframe src="https://www.google.com" name="B">  
<p>Google.com</p>  
</iframe>
```

```
<iframe src="https://www.google.it" name="C">  
<p>Google.it</p>  
</iframe>
```

```
<a href="http://www.acmenovara.it" target="B">CAMBIO B</a>
```



# Bookmark

Crea un bookmark in HTML

I bookmark possono essere utili se una pagina web è molto lunga.

Per creare un bookmark: prima occorre creare il bookmark, poi aggiungere un collegamento ad esso.

Quando si fa clic sul bookmark, la pagina scorrerà verso il basso o verso l'alto fino alla posizione del bookmark.

# IMG

Il tag HTML **<img>** viene utilizzato per incorporare un'immagine in una pagina web.

Le immagini non sono tecnicamente inserite in una pagina web; le immagini sono collegate a pagine web. Il tag **<img>** crea uno spazio per inserire l'immagine.

Il tag **<img>** è vuoto, contiene solo attributi e non ha un tag di chiusura.

Il tag **<img>** ha due attributi obbligatori:

**src** - Specifica il percorso dell'immagine

**alt** - Specifica un testo alternativo per l'immagine

25/2/1993 nasce il tag IMG

<http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html>

# Immagini di sfondo

Tramite l'attributo **style** è possibile inserire un'immagine di sfondo ad un elemento grafico

```
<div style="background-image:url('https://i.ytimg.com/vi/-cQk6kK7JVs/maxresdefault.jpg');">  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Donec ante eros, egestas vitae ullamcorper id, dignissim nec ipsum.  
  
</div>
```

Testo **Lorem ipsum** generato da <https://www.lipsum.com/>

## Immagini di sfondo - cover

Se volete che l'immagine di sfondo copra l'intero elemento, si può impostare la proprietà

**background-size: cover**

Inoltre, per assicurarsi che l'intero elemento sia sempre coperto, basta impostare la proprietà

**background-attachment: fixed**

In questo modo, l'immagine di sfondo coprirà l'intero elemento, senza cambiarne le proporzioni

# Picture

L'elemento HTML **<picture>** offre una maggiore flessibilità nello specificare le risorse dell'immagine. Contiene uno o più elementi **<source>**, ognuno dei quali si riferisce a diverse immagini tramite l'attributo **srcset**. In questo modo il browser sceglie l'immagine che meglio si adatta alla visualizzazione e/o al dispositivo corrente.

Ogni elemento **<source>** ha un attributo multimediale che definisce quando l'immagine è la più adatta.

L'elemento **<img>** indicherà il default da usare nel caso non ci siano match corretti.

```
<picture>
  <source media="(min-width: 650px)" srcset="computer.jpg">
  <source media="(min-width: 465px)" srcset="tablet.jpg">
  
</picture>
```

## Picture - perché?

Quando usare il tag **<picture>**?

- Risparmiare banda
- Usare immagini ottimizzate
- Adattare facilmente il contesto alla dimensione del device

# MAP

Il tag **<map>** viene usato per definire una mappa di immagini.

Una mappa è un'immagine con delle aree selezionabili

L'attributo **name** dell'elemento **<map>** è associato all'attributo **usemap** di **<img>** e crea una relazione tra immagine e mappa.

L'elemento **<map>** contiene un numero di elementi **<area>**, che definisce le aree cliccabili nella mappa dell'immagine

```
  
<map name="workmap">  
  <area shape="rect" title="google" coords="0,0,250,300" href="https://www.google.com">  
  <area shape="rect" title="bing" coords="250,0,500,300" href="https://www.bing.com">  
</map>
```

# Favicon

Una favicon è una piccola immagine visualizzata accanto al titolo della pagina nella scheda del browser e nel bookmark del browser

Per aggiungere una favicon ad un sito web, occorre salvarla nella directory principale del server web e chiamarla **favicon.ico**

Nel caso sia posizionata o nominata in modo diverso è necessario indicarne il percorso a livello di **head**

```
<head>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
```





# Favicon - come disegnarle

Un sito utilizzabile per disegnare delle favicon è <https://www.favicon.cc/>

Non volendo usare il formato **ico** è comunque possibile scegliere fra una serie di altri formati orizzontalmente accettati dai maggiori browser

Browser	ICO	PNG	GIF	JPEG	SVG
Edge	Yes	Yes	Yes	Yes	Yes
Chrome	Yes	Yes	Yes	Yes	Yes
Firefox	Yes	Yes	Yes	Yes	Yes
Opera	Yes	Yes	Yes	Yes	Yes
Safari	Yes	Yes	Yes	Yes	Yes

# Table

Il tag **table** serve a racchiudere delle informazioni rappresentabili in righe e colonne. Anche se molto utilizzato in passato, è stato progressivamente superato da altri costrutti responsive.

Il tag **table** contiene da più tag

table	: tabella
thead	: area di intestazione tabella
tbody	: corpo della tabella
tfoot	: piede della tabella
tr	: riga
th	: header
td	: data
caption	: titolo
colgroup	: definizione di colonne
col	: singolo elemento di definizione di colonna

## Table border

Ci sono una serie di proprietà che possono essere utilizzate per personalizzare i tag **table**, ad esempio

```
border: 1px solid black;  
border-collapse: collapse;  
border-radius: 10px;  
border-color: #96D4D4;  
background-color: #96D4D4;
```

# Table width

Con lo **style width** è possibile indicare l'occupazione di una singola colonna

```
<table style="width:100%" >
  <thead>
    <tr>
      <th style="width:10%" >Colonna 1</th> <th>Colonna 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Dato 1</td><td>Dato 2</td>
    </tr>
  </tbody>
</table>
```

# Table colspan e rowspan

Con lo **colspan** e **rowspan** è possibile indicare che una cella occupa più di una riga o di una colonna

```
<table>
  <thead> <tr> <th colspan="2" >Colonna 1</th><th>Colonna 2</th> </tr> </thead>
  <tbody>
    <tr> <td rowspan="2" >Dato 1</td> <td>Dato 2</td> <td>Dato 3</td> </tr>
    <tr> <td>Dato 2</td> <td>Dato 3</td> </tr>
  </tbody>
  <tfoot><tr><td>Piede 1</td><td>Piede 2</td><td>Piede 3</td></tr></tfoot>
</table>
```

Colonna 1		Colonna 2
Dato 1	Dato 2	Dato 3
	Dato 2	Dato 3
Piede 1	Piede 2	Piede 3

## Table caption

Con **caption** è possibile dare un titolo a una tabella

Il tag **caption** deve essere inserito subito dopo il tag **table**

...

```
<caption> Caption </caption>
```

...

## Table colgroup col

Con **colgroup** e **col** è possibile indicare una serie di caratteristiche che la singola colonna deve avere: come un colore o un bordo

Il tag **col** può anche avere un attributo che indica quante colonne subiscono il nuovo stile

```
<colgroup>
  <col span="2" style="background-color:green;">
  <col style="background-color:blue; border: 5px solid black;">
</colgroup>
```

## Liste non ordinate

L'utilizzo dei tag **ul** e **li** permette di definire delle liste non ordinate

Le liste sono rappresentate da una serie di righe contigue evidenziate da un piccolo cerchio nero iniziale

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



# Liste ordinate

L'utilizzo dei tag **ol** e **li** permette di definire delle liste ordinate

Le liste sono rappresentate da una serie di righe contigue numerate

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

# Liste descrittive

L'utilizzo dei tag **dl**, **dt** e **dd** permettono di creare delle liste descrittive

Con **dl** si definisce la lista, con **dt** si descrive il nome (term), con **dd** si descrive il termine

```
<dl>
  <dt>Cornetto</dt>
    <dd>- alla crema</dd>
    <dd>- al cioccolato</dd>
    <dd>- alla marmellata</dd>
  <dt>Latte</dt>
    <dd>- di soia</dd>
    <dd>- di capra</dd>
</dl>
```

# Blocchi e inline

Esistono due tag generici e abbastanza simili utilizzati per definire delle aree.

I tag sono **div** e **span** .

## DIV

L'elemento inizia sempre su una nuova riga

Occupava l'intera larghezza disponibile

Ha un margine inferiore e superiore

## SPAN

Non inizia su una nuova riga

Occupava solo la larghezza necessaria

Non ha un margine superiore e inferiore

# Blocchi e inline

La rappresentazione di **DIV** e **SPAN** è notevolmente diversa:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. **<div style="background-color:yellow; border: 1px solid red;">Ciao DIV</div>** Praesent laoreet hendrerit neque sed sagittis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

**Ciao DIV**

Praesent laoreet hendrerit neque sed sagittis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. **<span style="background-color:yellow; border: 1px solid red;">Ciao SPAN</span>** Praesent laoreet hendrerit neque sed sagittis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. **Ciao SPAN** Praesent laoreet hendrerit neque sed sagittis.

# Classi

L'attributo **class** serve ad indicare una classe associata ad un qualsiasi tag.

```
<div class="town">  
  <h2>Parigi</h2>  
  <p>Capitale della federazione dei pianeti uniti, situata sul Quadrante Alfa</p>  
</div>
```

La classe specificata può poi essere descritta a livello di CSS.

Da notare che il nome della classe è case sensitive.

# ID

L'attributo **id** permette di referenziare un singolo tag in modo univoco.

Il browser non effettua nessun controllo di univocità sull' **id** è quindi a carico di chi crea la pagina assicurarsi che **id** sia effettivamente univoco.

Avere un **id** ci permette di applicare delle caratteristiche ad un singolo tag e di usarlo in modo più efficiente da parte di codice javascript.

# Iframe

Il tag **iframe** permette l'inclusione di una pagina all'interno di una pagina HTML

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

L'attributo necessario per il funzionamento di un **iframe** è **url** .

Questo attributo indica l'indirizzo della pagina da includere.

# Iframe

**iframe** ha una serie di attributi che ne condizionano la visualizzazione come:

```
<iframe src="https://www.baccan.it/" title="Matteo Baccan" height="200" width="300"></iframe>  
<iframe src="https://www.baccan.it/" title="Matteo Baccan" style="height:200px;width:300px;"></iframe>
```

**Iframe** può poi essere utilizzato come target di un tag **a**

```
<iframe src="about:blank" title="Matteo Baccan" name="baccan"></iframe>  
<a href="https://www.baccan.it/" target="baccan">Baccan.it</a>
```



# Script

**script** è il tag col quale è possibile inserire del codice JavaScript per rendere dinamiche delle pagine HTML.

**script** può contenere direttamente del codice, o referenziare una pagina esterna contenente a sua volta del codice.

```
<a href="https://www.baccan.it/" id="baccan">Baccan.it</a>

<script>
    document.getElementById("baccan").innerHTML = "CIAO!!";
</script>
```

# Path

E' importante capire come sono gestiti i file in una struttura a cartelle. La corretta comprensione permette di indirizzare nel modo giusto le varie risorse.

## Rispetto alla pagina corrente

```
  
  
  
  

```

foto.jpg è nella stessa cartella  
foto.jpg è nella cartella immagini  
foto.jpg è nella cartella immagini del root folder  
foto.jpg è nella cartella padre della cartella corrente  
foto.jpg è sul sito foo.bar

# Head

**head** è il tag dove sono contenute una serie di informazioni propedeutiche alla visualizzazione corretta di una pagina e alla sua indicizzazione all'interno di motori di ricerca e social network.

Nel tag **head** sono presenti anche i riferimenti a file esterni, tipicamente **css**.

Altri elementi che entrano a far parte di questo tag sono i tag **meta**.

# Head

Il set di caratteri usato per la corretta visualizzazione del sito

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

Aggiornamento automatico

```
<meta http-equiv="refresh" content="10">
```

Impostazioni per la visualizzazione responsive

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# Head

Elementi utilizzati dai motori di ricerca

Descrizione

```
<meta name="description" content="La mia bellissima pagina">
```

Keywords

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

L'autore

```
<meta name="author" content="Matteo Baccan">
```

# Open Graph

Altri elementi che si trovano in **head** sono quelli del protocollo Open Graph.

<https://ogp.me/>

Il protocollo Open Graph consente a qualsiasi pagina Web di essere arricchita di dati utili ad un “social graph”. Ad esempio, questo protocollo viene utilizzato su Facebook per consentire a qualsiasi pagina Web di avere le stesse funzionalità di qualsiasi altro oggetto su Facebook.

# Open Graph

Le quattro proprietà minime per Open Graph sono

**og:title** : il titolo del vostro oggetto

**og:type** : il tipo, ad esempio "video.movie"

**og:image** : l'immagine che rappresenta l'oggetto

**og:url** : l'indirizzo canonico dell'oggetto

# Open Graph

## Un esempio di utilizzo delle proprietà Open Graph

```
<meta property="og:url"           content="https://www.baccan.it" />
<meta property="og:title"        content="Il sito personale di Matteo Baccan" />
<meta property="og:description"  content="Scrivo software e aiuto le aziende a scriverne di migliore" />
<meta property="og:image"        content="https://www.baccan.it/logo/29.png" />
<meta property="og:type"         content="blog" />
<meta property="og:site_name"    content="Il sito personale di Matteo Baccan" />
```



# Open Graph

Altri elementi di Open Graph sono

**og:audio** : l'audio che accompagna questo oggetto

**og:description** : la descrizione di questo oggetto

**og:determiner** : la parola che appare prima del titolo all'interno di una frase. E' possibile scegliere fra **a** , **an** , **the** , **""** , **auto** ). Se è scelto **\_\_auto\_\_** il valore sarà scelto fra **a** o **an** . Il default è vuoto **""** (blank).

**og:locale** : è la lingua dell'oggetto. Se non indicato il default è **en\_US** .

**og:locale:alternate** : l'elenco di altre lingue nelle quali la pagina è disponibile

**og:site\_name** : se l'oggetto è parte di un grande sito web, sarà il nome con il quale verrà referenziato il sito

**og:video** : l'indirizzo del video in accompagnamento della pagina

# Open Graph

Un esempio di uso avanzato di Open Graph preso dalla pagina di **Gal Gadot**

<https://www.imdb.com/name/nm2933757/> protagonista di “Wonder Woman 1984”

```
<meta property="og:url" content="http://www.imdb.com/name/nm2933757/" />
```

```
<meta property='og:image' content="https://m.media-  
amazon.com/images/M/MV5BYThjM2NIOTItYTUzMjM0ODE3LTk1MTItM2I3MDViY2U  
3MThlXkEyXkFqcGdeQXVyMTg4NDI0NDM@._V1_UY1200_CR165,0,630,1200_AL_.j  
pg" />
```

```
<meta property='og:type' content="actor" />
```

## Open Graph

```
<meta property="og:description" content="Gal Gadot, Actress: Wonder Woman  
1984. Gal Gadot is an Israeli actress, singer, martial artist, and model. She was born in  
Rosh Ha'ayin, Israel. Her parents are Irit, a teacher, and Michael, an engineer. She  
served in the IDF for two years, and won the Miss Israel title in 2004. Gal made her film  
debut in the fourth film of the Fast and Furious franchise, Fast & Furious - Solo parti..."  
/>
```

```
<meta property='og:title' content="Gal Gadot - IMDb" />
```

```
<meta property='og:site_name' content='IMDb' />
```

# Open Graph

Alcune proprietà Open Graph possono avere delle proprietà aggiuntive.

Ad esempio **og:image** può indicare alcuni dati in riferimento alla propria dimensione.

<code>og:image:width</code>	Larghezza in pixel
<code>og:image:height</code>	Altezza in pixel

# Responsive

Il responsive web design consiste nel creare pagine web che si adattino a qualsiasi dispositivo

Un responsive web design si adatterà in automatico alle diverse dimensioni dello schermo e viewport

Per poter definire che una pagina è responsive occorre aggiungere una riga nel tag **head**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Questa riga dirà al browser come controllare le dimensioni di pagina

## Responsive image

Per rendere una immagine responsive è possibile utilizzare **max-width** che ci permette di indicare la dimensione massima che avrà in larghezza ed impostare l'altezza ad **auto**

```

```

# Responsive image

Un altro elemento responsive che abbiamo visto è il tag **picture**

```
<picture>
  <source media="(min-width: 650px)" srcset="computer.jpg">
  <source media="(min-width: 465px)" srcset="tablet.jpg">
  
</picture>
```

# Responsive text

Anche gli elementi testuali possono avere delle caratteristiche responsive. Per farlo occorre impostare le dimensioni del carattere in **vw** che rappresenta il **viewport width**

Ad esempio proviamo ad impostare la grandezza di **ciao** a 10vw

```
<div style="font-size:10vw">Ciao</div>
```

Ingrandendo a diminuendo la dimensione dello schermo il testo si adeguerà di conseguenza.

Il **viewport width** è la larghezza della finestra del browser e l'unità **vw** è il suo percentile

<https://whatismyviewport.com/>



# Media Query

Un altro modo per adattare le pagine ad una particolare dimensione è l'utilizzo delle media query

In questo modo è possibile cambiare completamente layout, in base alle dimensioni dello schermo

# Media Query

All'interno di una media query è possibile indicare il momento in cui un particolare frammento di CSS deve entrare in gioco

In questo caso, il CSS viene considerato fino alla larghezza di 800px

```
@media screen and (max-width: 800px) {  
    /* CSS */  
}
```

# Code

Il tag **code** permette di evidenziare del testo in modo possa essere usato per la rappresentazione di codice, ad esempio usando un font non proporzionale

```
<code>  
  x = 1;  
  y = 2;  
  z = x + y;  
</code>
```

Il comportamento è simile all'uso di un tag **span** con font **courier**

## kbd

Il tag **kbd** permette di evidenziare una sequenza di tasti da premere: l'aspetto è simile a quello di **code**

```
<p>Visualizza il sorgente premendo <kbd>Ctrl + U</kbd></p>
```

Il comportamento è simile all'uso di un tag **span** con font **courier**

## samp

Il tag **samp** permette di simulare l'output di un programma

```
<p>Esempio di output:</p>  
<p><samp>Hello world.<br>Premi F1 per continuare</samp></p>
```

## pre

Il tag **pre** permette di mantenere la formattazione presente all'interno di un testo

```
<pre>
  x = 1;
  y = 2;
  z = x + y;
</pre>
```

# var

Il tag **var** evidenzia una variabile o una espressione

L'area del rettangolo è `<var>b</var>` per `<var>a</var>`, dove `<var>b</var>` è la base e `<var>a</var>` è l'altezza

# Layout

HTML ha diversi tag semantici che definiscono le diverse parti di una pagina.

Questi elementi permettono una standardizzazione del codice verso una serie di elementi comuni

```
<header> - Intestazione o sezione di un documento  
<nav> - Link di navigazione  
<section> - Sezioni  
<article> - Un contenuto  
<aside> - Una sidebar  
<footer> - Il piede del documento o di una sezione  
<details> - Dettagli aggiuntivi rispetto al documento  
<summary> - Un header dell'elemento <details>
```





# Layout

Esistono tecniche diverse per posizionare questi tag

- L'uso di framework CSS come Bootstrap
- L'uso delle proprietà **float** e **clear**
- L'uso di Flexbox (1)
- L'uso di Grid Layout (2)

<https://caniuse.com/flexbox>

<https://caniuse.com/css-grid>



# Validazione

Per verificare che il proprio codice HTML/CSS sia valido è possibile usare dei siti di validazione codice

Il più conosciuto è il validatore di w3.org, che è in grado di darci segnalazione di tutte le anomalie presenti nel nostro codice. Esistono però delle valide alternative

<https://validator.w3.org>

<https://html5.validator.nu>

In alternativa esistono dei plugin utilizzabili direttamente da browser, come **html-validator** per Chrome

<https://chrome.google.com/webstore/detail/html-validator/mpbelhhnfhfjnaehkcnnaknldmnocgik>

## Validazione cross device

Esistono una serie di siti web che permettono di verificare che il proprio sito venga visualizzato correttamente su una serie di device.

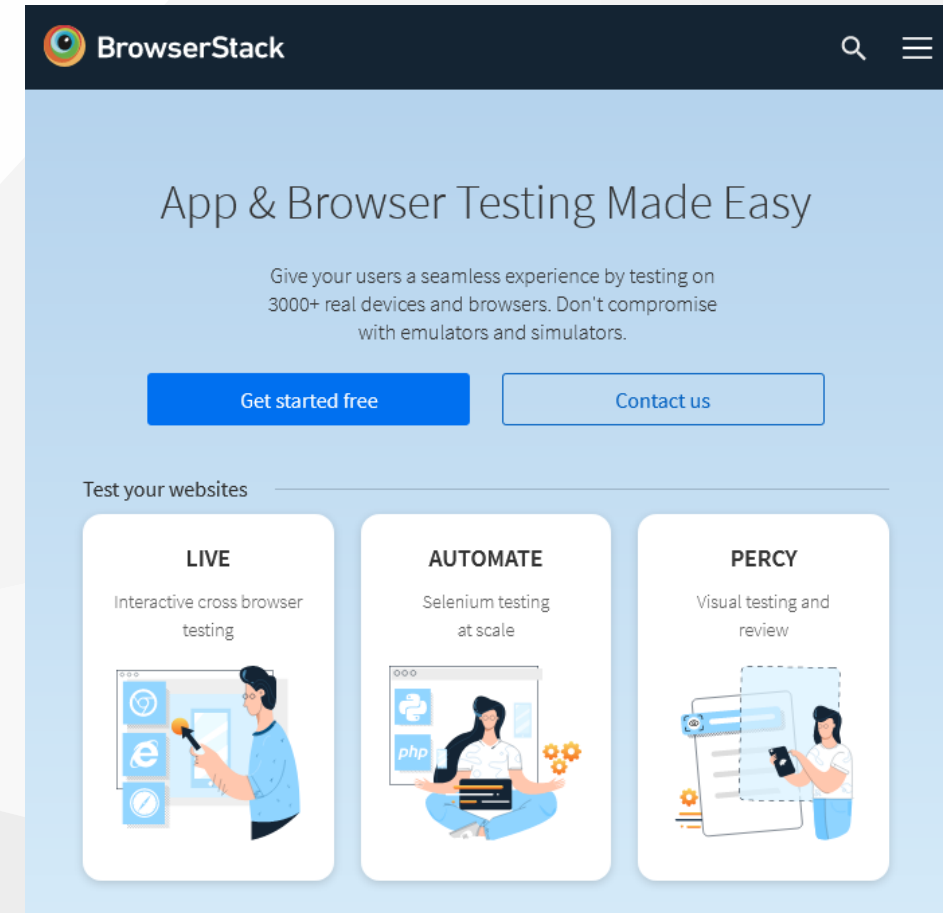
Alcuni di più famosi sono

<https://www.browserstack.com>

<https://www.browserling.com>

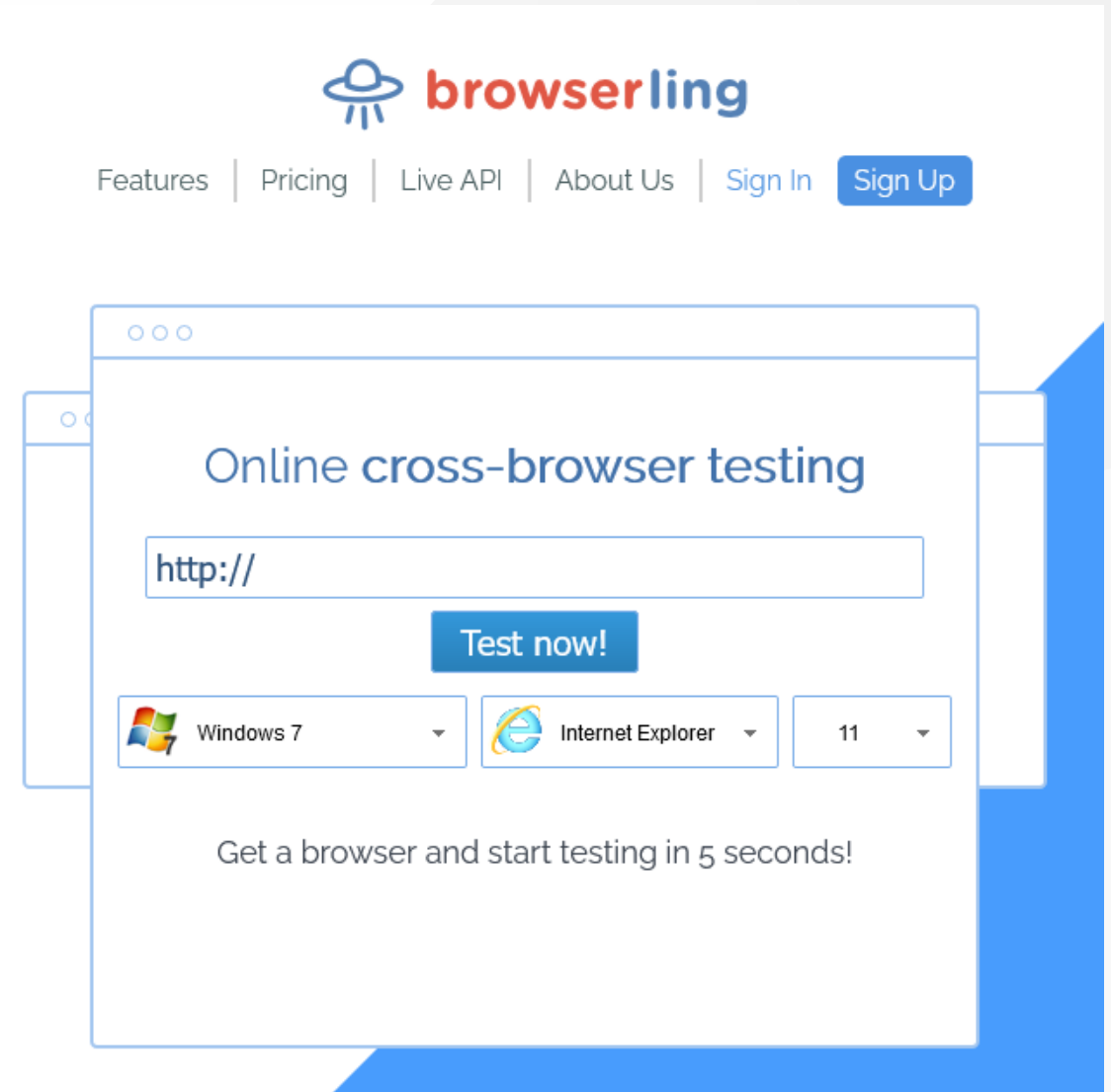
# Validazione cross device

Browserstack richiede una registrazione per poter usare la versione gratuita e mette a disposizione una simulazione su device reali per sistemi operativi Windows, Mac, IOS e Android



# Validazione cross device

Browserling permette 1 minuto di navigazione su una serie di sistemi operativi e versioni di browser



# Entity

Esistono una serie di caratteri riservati che non possono essere inseriti all'interno di una pagina HTML.

Ad esempio i caratteri `<` e `>` che rappresentano l'inizio e la fine di un **tag**.

Per superare questa limitazione è possibile ricorrere alle entity, che rappresentano una sintassi con la quale indicare la rappresentazione di un carattere riservato.

Una entity è rappresentabile in 2 modi differenti, tramite

**`&nome_entity;`**

Oppure

**`&#numero_entity;`**

# Entity

Di seguito una serie di entity largamente diffuse

	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	double quotation mark	&quot;	&#34;
'	single quotation mark (apostrophe)	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;

# Entity - simboli

Oltre ai caratteri riservati, esistono una serie di caratteri non riproducibili da tastiera, anche loro sono rappresentabili con delle entity

Un esempio sono i simboli matematici

$\forall$	<code>&amp;#8704;</code>	<code>&amp;forall;</code>	FOR ALL
$\partial$	<code>&amp;#8706;</code>	<code>&amp;part;</code>	PARTIAL DIFFERENTIAL
$\exists$	<code>&amp;#8707;</code>	<code>&amp;exist;</code>	THERE EXISTS
$\emptyset$	<code>&amp;#8709;</code>	<code>&amp;empty;</code>	EMPTY SETS
$\nabla$	<code>&amp;#8711;</code>	<code>&amp;nabla;</code>	NABLA
$\in$	<code>&amp;#8712;</code>	<code>&amp;isin;</code>	ELEMENT OF
$\notin$	<code>&amp;#8713;</code>	<code>&amp;notin;</code>	NOT AN ELEMENT OF
$\ni$	<code>&amp;#8715;</code>	<code>&amp;ni;</code>	CONTAINS AS MEMBER
$\prod$	<code>&amp;#8719;</code>	<code>&amp;prod;</code>	N-ARY PRODUCT
$\sum$	<code>&amp;#8721;</code>	<code>&amp;sum;</code>	N-ARY SUMMATION



## Entity - simboli non riproducibili

Oltre ai caratteri riservati, esistono una serie di caratteri non riproducibili da tastiera, anche loro sono rappresentabili con delle entity

Un esempio sono i simboli matematici

∅	&#8709;	&empty;	Vuoto
∈	&#8712;	&isin;	Contenuto
∉	&#8713;	&notin;	Non contenuto

Per un elenco completo è possibile utilizzare questo indirizzo

[https://www.w3schools.com/charsets/ref\\_utf\\_math.asp](https://www.w3schools.com/charsets/ref_utf_math.asp)

# Entity - emoji

Le entity possono rappresentare, non solo caratteri, ma anche emoji, che vengono visualizzate come piccole immagini, pur non essendolo.

Le emoji sono caratteri provenienti dal charset Unicode UTF-8 e sono sottoposte ad uno standard che si aggiorna in modo periodico

L'accortezza da avere per poter visualizzare i caratteri nel modo corretto è che la pagina inizi col metatag

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

<https://www.fileformat.info/info/charset/UTF-8/list.htm>

# Entity - emoji esempi

Di seguito alcuni esempi di emoji

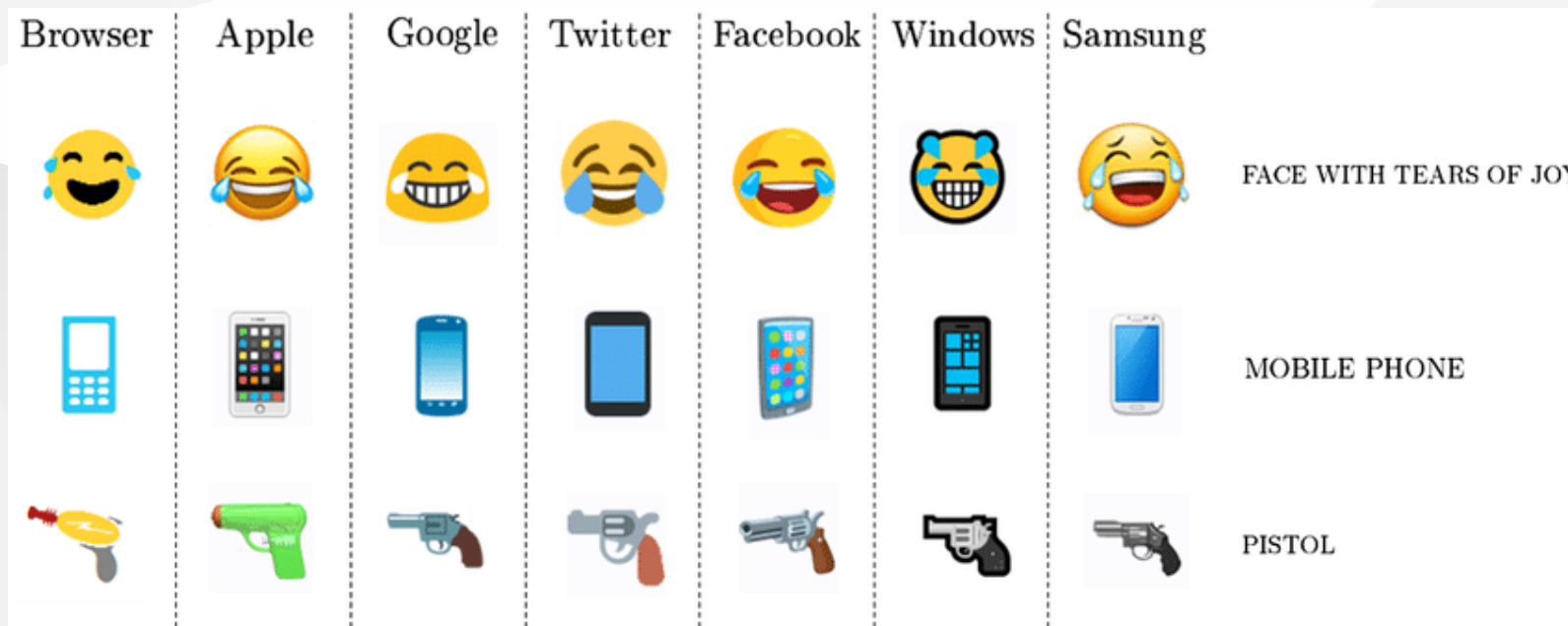
✓	&#10004;	2714
✗	&#10006;	2716
✚	&#10013;	271D
✡	&#10017;	2721
✨	&#10024;	2728
✳	&#10035;	2733
✴	&#10036;	2734
❄	&#10052;	2744

Per un elenco completo è possibile utilizzare questo indirizzo

[https://www.w3schools.com/charsets/ref\\_emoji.asp](https://www.w3schools.com/charsets/ref_emoji.asp)

# Entity - emoji - rendering

Essendo immagini, specificate tramite caratteri, le emoji vengono rappresentate diversamente in base al sistema operativo, sito web o applicazione che le rappresenta.



[https://www.researchgate.net/figure/Seven-different-implementations-of-three-different-Emoji-The-face-with-tears-of\\_fig16\\_316168050](https://www.researchgate.net/figure/Seven-different-implementations-of-three-different-Emoji-The-face-with-tears-of_fig16_316168050)

## Entity - emoji – tonalità della pelle

Alcune emoji permettono di cambiare il colore della pelle, per farlo si pone, dopo l'entity della emoji, un entity di colorazione

&#127999;	■	Carnagione scura
&#127998;	■	Carnagione abbastanza scura
&#127997;	■	Carnagione media
&#127996;	■	Carnagione abbastanza chiara
&#127995;	■	Carnagione chiara

## Entity - emoji – esempi di tonalità della pelle

Di conseguenza, se vogliamo colorare diversamente l'emoji della mano faremo

```
&#9757; Nessun tono della pelle  
&#9757;&#127999; Scuro  
&#9757;&#127998; Medio Scuro  
&#9757;&#127997; Medio  
&#9757;&#127996; Luce media  
&#9757;&#127995; Luce
```

# Form

Un form HTML serve a raccogliere l'input dell'utente.

**Form** è tag che inizia e finisce una maschera di acquisizione dati

```
<form action="/action.php">
  <label for="fname">Nome:</label>
  <input type="text" id="fname" name="fname" value="Matteo">
  <label for="lname">Cognome:</label>
  <input type="text" id="lname" name="lname" value="Baccan">
  <input type="submit" value="conferma">
</form>
```

# Form attributi

Il tag **form** ha una serie di attributi che ne modificano il comportamento

**target** si comporta analogamente al target inserito all'interno di un tag **a**

**method** è il metodo col quale i parametri devono essere passati alla chiamata http di conferma. Normalmente i metodi sono **get** o **post** . Se non indicato il default è **get**. Ci sono delle implicazioni a livello di sicurezza e di gestione di cui tener conto quando viene scelto il metodo per inviare i dati

**autocomplete** è una indicazione che viene data al browser, che indica se automatizzare il completamento dei campi con dati che sono stati precedentemente digitati

**novalidate** indica che i dati inseriti non devono essere validati al loro salvataggio



# Form elementi

Esistono più tag che rappresentano il modo col quale inserire dati in una form.

Ognuno di essi ha delle proprie caratteristiche di utilizzo

```
<input>  
<label>  
<select>  
<textarea>  
<button>  
<fieldset>  
<legend>  
<datalist>  
<output>  
<option>  
<optgroup>
```

# Form input

Il tag **input** è il primo tag utilizzato e serve ad inserire dei campi testuali monoriga. Normalmente si lega ad un tag **label** che rappresenta la sua descrizione

```
<label for="fname">Nome:</label>  
  
<input type="text" id="fname" name="fname">
```

## Form input type

Uno degli attributi più interessanti di **input** è **type**, che identifica il tipo di dati che verranno inseriti nel campo

button	checkbox	color	date	datetime	datetime-local		email	
file	hidden	image	month	number	password	radio		
range	reset	search	submit	tel	text	time	url	week

# CSS Framework

Per facilitare la scrittura di pagine HTML sono nati col tempo una serie di framework CSS utili a velocizzare il lavoro

Fra i framework più noti ci sono

- Bootstrap
- Foundation
- Bulma
- Tailwind
- Uikit
- Milligram
- Pure CSS
- Tachyons
- Materialize CSS

# Bootstrap

Cos'è Bootstrap?

Bootstrap è un framework front-end gratuito, progettato per essere mobile first

Bootstrap include modelli di progettazione basati su HTML e CSS per tipografia, moduli, pulsanti, tabelle, navigazione, modali, caroselli di immagini e molti altri, oltre a plug-in JavaScript opzionali

Bootstrap ti dà anche la possibilità di creare facilmente design reattivi

# Bootstrap - container

Bootstrap fornisce 2 classi per definire un container

## **.container**

che fornisce un contenitore reattivo a larghezza fissa

## **.container-fluid**

che fornisce un contenitore a larghezza intera, che copre l'intera larghezza della finestra

## Bootstrap - grid

Il sistema a griglia di Bootstrap è costruito con flexbox e consente fino a 12 colonne nella pagina.

Se non desideri utilizzare tutte e 12 le colonne singolarmente, puoi raggruppare le colonne insieme per creare colonne più larghe

# Bootstrap - grid

il sistema a griglia Bootstrap 5 ha sei classi:

```
.col- (dispositivi extra piccoli - larghezza dello schermo inferiore a 576px)
.col-sm- (piccoli dispositivi - larghezza dello schermo uguale o superiore a 576px)
.col-md- (dispositivi medi - larghezza dello schermo uguale o superiore a 768px)
.col-lg- (dispositivi di grandi dimensioni - larghezza dello schermo uguale o superiore a 992px)
.col-xl- (dispositivi xlarge - larghezza dello schermo uguale o superiore a 1200px)
.col-xxl- (dispositivi xxlarge - larghezza dello schermo uguale o superiore a 1400px)
```

Le classi di cui sopra possono essere combinate per creare layout più dinamici e flessibili.



# Bootstrap - table

Bootstrap dispone di classi predisposte per la visualizzazione dei tag **table**

Per utilizzarle è sufficiente indicarle a livello di attributo **class**

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

```
<table class="table table-striped">  
</table>
```

```
<table class="table table-hover">  
</table>
```

```
<table class="table table-bordered">  
</table>
```

<https://getbootstrap.com/docs/5.0/content/tables/>

## Bootstrap - testi

Bootstrap modifica lo stile standard dei tag tipografici, come **h1**, **h2**, **h3** .. **h6** , ma soprattutto aggiunge una serie di classi per la formattazione dei testi:

<code>.text-start</code>	Allinea il testo a sinistra
<code>.text-center</code>	Allinea al centro
<code>.text-decoration-none</code>	Rimuove la sottolineatura da un collegamento
<code>.text-end</code>	Allinea a destra
<code>.text-nowrap</code>	Evita il ritorno a capo
<code>.text-lowercase</code>	Minuscolo
<code>.text-uppercase</code>	Maiuscolo
<code>.text-capitalize</code>	Iniziale in maiuscolo

# Bootstrap – colori testo

Bootstrap aggiunge una serie di classi utili per raggruppare dei contesti in base ai colori. Queste classi possono essere sovrascritte da template CSS integrativi

```
.text-muted  
.text-primary  
.text-success  
.text-info  
.text-warning  
.text-danger  
.text-secondary  
.text-white  
.text-dark  
.text-body  
.text-light
```

# Bootstrap – colori sfondo

Bootstrap anche gli sfondi hanno delle colorazioni contestuali

```
.bg-primary  
.bg-success  
.bg-info  
.bg-warning  
.bg-danger  
.bg-secondary  
.bg-dark  
.bg-light
```

# Bootstrap – immagini

Per quanto riguarda le immagini, sono state introdotte una serie di classi che aiutano la loro visualizzazione e allineamento

<code>.rounded</code>	Arrotonda gli angoli dell'immagine
<code>.rounded-circle</code>	Mette l'immagine in un cerchio
<code>.img-thumbnail</code>	Mette un bordo all'immagine
<code>.float-start</code>	Allinea a sinistra
<code>.float-end</code>	Allinea a destra
<code>.mx-auto (margin:auto)</code> ed <code>.d-block (display:block)</code>	Permettono la centratura
<code>.img-fluid</code>	Crea un'immagine responsive applicando <code>max-width: 100%;</code> e <code>height: auto;</code>

# Bootstrap – Jumbotron

jumbotron è un grande box con lo scopo di richiamare l'attenzione su alcuni contenuti

```
<div class="jumbotron">  
  <h1>Titolo</h1>  
  <p>testo testo testo testo testo testo testo testo testo testo </p>  
</div>
```

# Bootstrap – Alert

Le classi di alert servono ad evidenziare un testo utilizzando un contesto, come nel caso dei testi

```
<div class="alert alert-success">  
  <strong>SUCCESSO</strong> hai fatto qualcosa di perfetto  
</div>
```

Le tipologie di classi di alert si sovrappongono a quelle già viste per i testi

```
.alert-success .alert-info .alert-warning .alert-danger  
.alert-primary .alert-secondary .alert-light .alert-dark
```

Come classi di supporto agli alert è possibile anche usare

.alert-dismissible	Alert chiudibile
.fade .show	Alert con un fade in chiusura

# Bootstrap – Button

Come per testi e alert, anche i button hanno delle classi relative al contesto

```
<button type="button" class="btn">Basic</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-info">Info</button>
```

Le tipologie di classi di btn si sovrappongono a quelle già viste per i testi

```
.btn-success .btn-info .btn-warning .btn-danger .btn-primary .btn-secondary .btn-light .btn-dark
```

Come classi di supporto è possibile usare le stesse classi con aggiunta di **outline**

```
.btn-outline-primary .btn-outline-secondary ...
```



# Bootstrap – Badge

I badge sono delle etichette di testo che subiscono il contesto come button e text

```
<h1>Esempio di intestazione <span class="badge bg-secondary">badge</h1>
```

In questo caso il suffisso da usare è **bg**

La classe di supporto dei badge è

**rounded-pill** Arrotonda i bordi del badge

## Bibliografia

<https://mahipal.dev/web-browser-engine-overview-for-beginners> : spiegazione delle strutture del browser

<https://www.oreilly.com/library/view/learning-php-mysql/9781449337452/ch01s01.html> : browser flow

<https://www.w3schools.com> : vari esempi e spunti

## Disclaimer

L'autore ha generato questo testo in parte con GPT-3, il modello di generazione del linguaggio su larga scala di OpenAI. Dopo aver generato la bozza della lingua, l'autore ha rivisto, modificato e rivisto la lingua a proprio piacimento e si assume la responsabilità ultima del contenuto di questa pubblicazione.