

Corso di Programmazione Web

<HTML / CSS>

Giuseppe Solla

1

Visual Studio Code

Come creare una pagina web

Partiamo dall'html, è utilizzato per definire il contenuto di una pagina web quindi: testo, immagini, video, sono definiti all'interno di una pagina web tramite html. Tutto quello che riguarda l'aspetto grafico quindi: colore, caratteri, stili, spaziature è definito dai CSS.

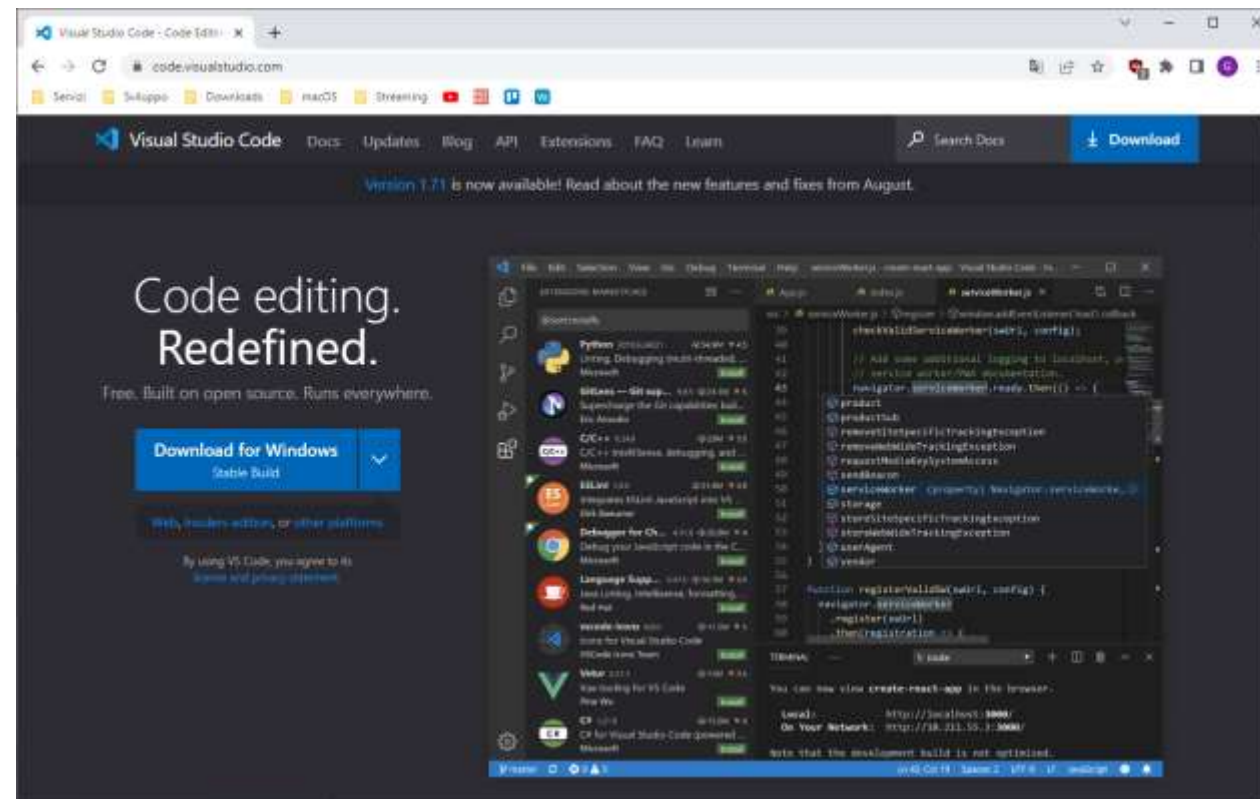
Per modificare e scrivere html, css, javascript, ecc. abbiamo bisogno di un editor. Se non ne avete ancora uno scaricate VSCode (consigliato per questo corso) oppure Atom o Sublime Text. Tutti sono editor professionali, gratuiti e aiutano molto nella scrittura del codice avendo numerosi plugin a supporto.

VSCode

L'editor VSCode, della Microsoft, è disponibile per il download sul sito:

<https://code.visualstudio.com/>

Come da figura:



VSCode

Cliccando sul bottone blu <Download for Windows> si scarica l'applicativo che va installato semplicemente eseguendo il programma scaricato.

Una volta installato occorre configurarlo per lo sviluppo in Html, Css e PHP aggiungendo i plugin necessari.

Italian Language Pack for Visual Studio Code - Necessario ad italianizzare l'editor.

Live Sass Compiler - Necessario per la compilazione automatica dei file .scss in .css oppure .min.css

Inline Bookmarks - Serve per evidenziare delle righe del codice con dei segnalibri per poi raggiungerle velocemente cliccando sulla vista aggiunta al pannello a destra.

VSCode

00

PHP Intelephense - fornisce un supporto avanzato di completamento automatico e refactoring per il linguaggio PHP.

Format HTML in PHP - Esegue la formattazione in automatico dell'HTML contenuto nel codice PHP.

Auto Rename Tag - In automatico rinomina il tag di chiusura rinominando quello di apertura.

VSCode – Se Emmet non funziona

1. avvia il tuo codice VS. Fare clic su Impostazioni o premere **Ctrl+,** (Control + Virgola) per aprire le Impostazioni codice VS.
2. fare clic sulla scheda Estensioni sul lato sinistro delle impostazioni. Fare clic su HTML.
3. fare clic sul collegamento ipertestuale "Modifica nelle impostazioni: json" per modificare le impostazioni in formato JSON.
4. all'interno delle parentesi graffe, inserisci il seguente codice sotto il codice JSON già scritto:

```
"emmet.triggerExpansionOnTab": true,  
"files.associations": {"*html": "html", "*.css": "css"},  
"emmet.useInlineCompletions": true,
```

Salva il file.

Ora se provi ad applicare Emmet nel tuo codice, funzionerà senza problemi !!



HTML

HyperText Markup Language

HTML (da Wikipedia)

In informatica l'**HyperText Markup Language** (traduzione letterale: linguaggio a marcatori per ipertesti), comunemente noto con l'acronimo **HTML**, è un [linguaggio di markup](#).

Nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web (definita appunto dal markup) e la sua rappresentazione, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal [World Wide Web Consortium](#) (W3C).

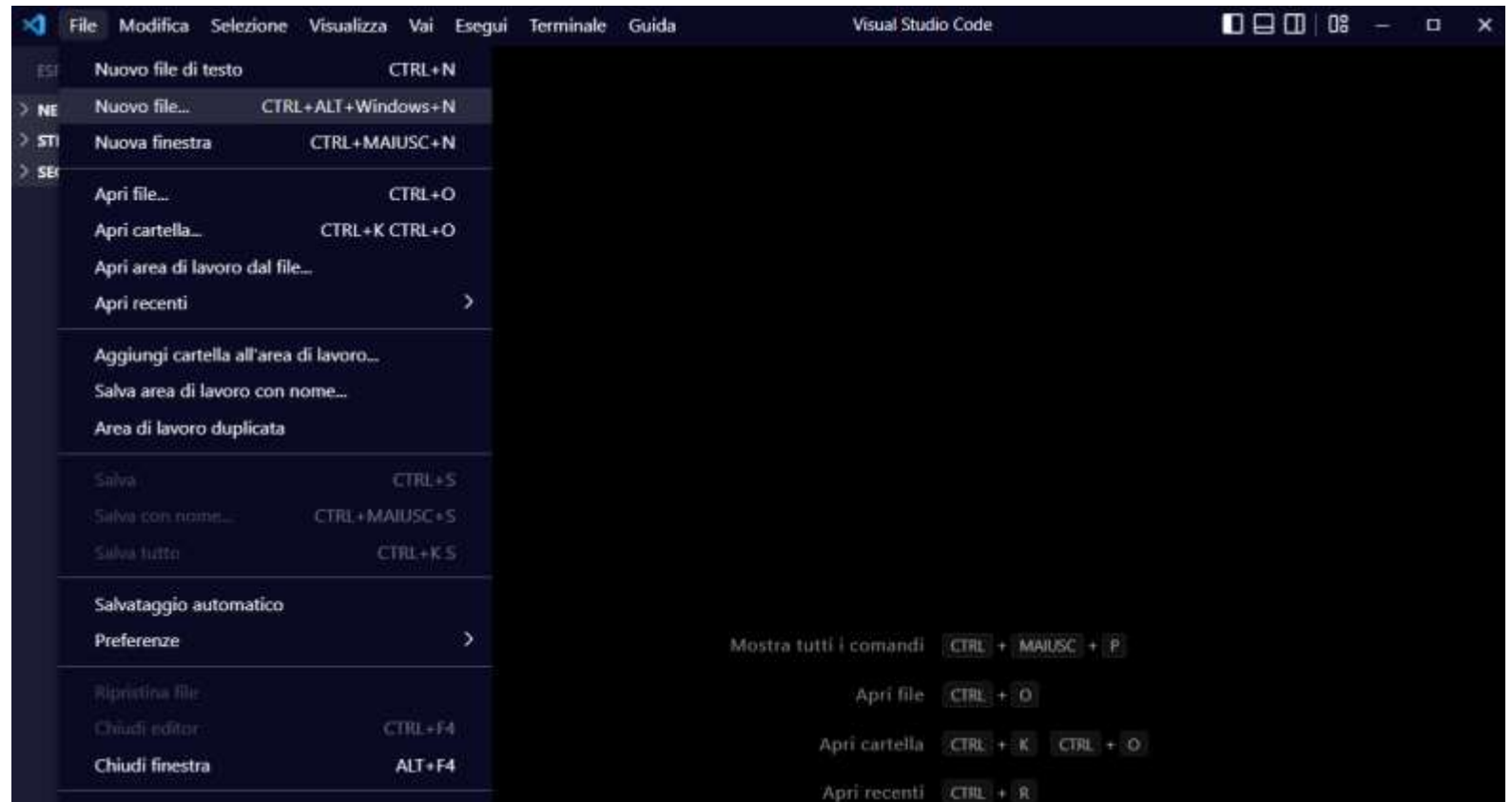
HTML (da Wikipedia)

È derivato dall'SGML, un metalinguaggio finalizzato alla definizione di linguaggi utilizzabili per la stesura di documenti destinati alla trasmissione in formato elettronico. La versione attuale, la quinta, è stata rilasciata dal W3C nell'ottobre 2014.

Il motivo principale che ha spinto il W3C e i suoi membri a sviluppare [HTML5](#) è stata la necessità di fornire direttamente le funzionalità che in precedenza erano fruibili tramite estensioni proprietarie all'esterno dei browser, come Adobe Flash e simili. Un secondo obiettivo che gli sviluppatori si erano prefissati era quello di garantire una maggiore compatibilità tra i diversi browser, indipendentemente dalla piattaforma software utilizzata, e principalmente mirata all'espansione dei dispositivi mobili.

La nostra prima pagina

Creiamo la nostra prima pagina web in html. Apriamo il nostro VSCode e creiamo un nuovo file: **File > Nuovo file** e lo chiamiamo index.html, selezionata la cartella Desktop (per comodità) lo salviamo.



La nostra prima pagina

Un file con estensione .html è un file che aperto nel browser vedremo una pagina web. Apriamo il file .html che abbiamo appena creato (doppio click sul file) vedremo una pagina bianca nel nostro browser di default. Questo logicamente perché non abbiamo ancora scritto nessun codice html al suo interno.

Abbiamo chiamato la nostra pagina index.html perché la maggior parte dei server utilizza questa pagina come pagina di default (la Home del sito). Se carichiamo questa pagina su uno spazio web "standard" questa verrà visualizzata quando digiteremo www.nomesito.com.

I tag HTML

I mattoncini con il quale è costruita la nostra pagina web sono i tag html. I tag sono parole chiave racchiuse tra i segni maggiore e minore. Quelli che servono per costruire una pagina web base non sono molti. Il primo che vediamo è `<html>`. Il tag html è il tag che racchiude tutta la nostra pagina web.

`<html>` è il tag di apertura di una pagina html.

`</html>` è il tag di chiusura di una pagina html.

Tutto quello che è all'interno di questi due tag è considerato parte della nostra pagina html. **Tutti i tag (a parte pochissimi casi come `` e `<input>`) necessitano di un tag di chiusura.** La sintassi prevede un tag di apertura (per esempio `<html>`) e un tag di chiusura (per esempio `</html>`) all'interno dei due tag va inserito il contenuto. Il tag di chiusura ha sempre il backslash "/" prima della keyword del tag.

I tag HTML

Gli altri due tag fondamentali per costruire una pagina web sono il tag **head** ed il tag **body**.

Il tag head è un tag “funzionale”.

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

Serve a contenere tutte una serie di informazioni che non vengono direttamente mostrate nella pagina ma servono alla pagina stessa per funzionare correttamente. Per esempio informazioni per i motori di ricerca e per il browser. Il tag body contiene quello che viene visualizzato nella pagina web.

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

I tag HTML

Mettendo insieme i nostri 3 tag fondamentali creiamo la struttura base di una semplice pagina web.

```
<!doctype html>
<html>
  <head>
    <!-- Contenuto dell'head -->
  </head>
  <body>
    <!-- Contenuto del Body -->
  </body>
</html>
```

Il tag **<!doctype html>** all'inizio della pagina indica al browser che stiamo scrivendo una pagina in html 5 (va inserito sempre). La sintassi **<!-- -->** è quella utilizzata per fare i commenti tutto quello dentro **<!-- -->** non viene considerato dal browser.

I tag HTML

Il tag p (paragrafo) è il tag usato per inserire testo libero all'interno della pagina web:

```
<p></p>
```

Proviamo a visualizzare un messaggio nella nostra pagina web:

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <!-- Contenuto dell'head -->
```

```
  </head>
```

```
  <body>
```

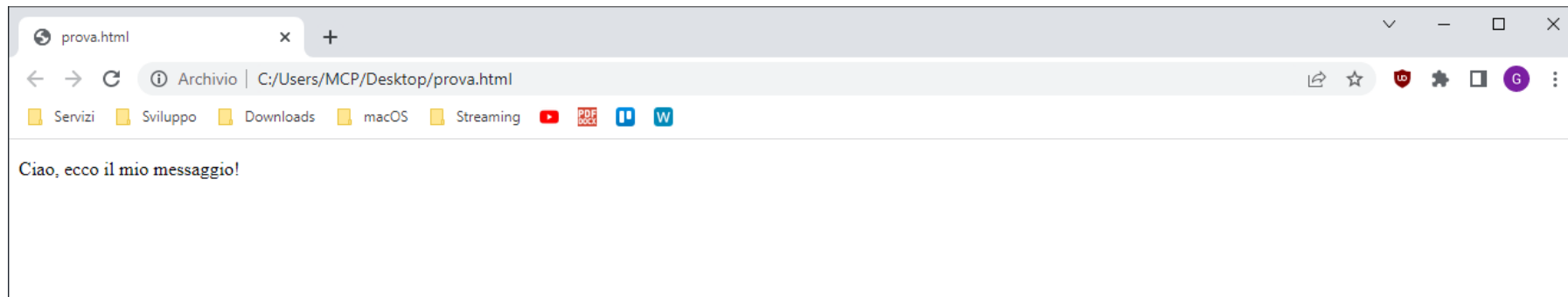
```
    <p>Ciao, ecco il mio messaggio!</p>
```

```
  </body>
```

```
</html>
```


I tag HTML

Nel codice html abbiamo utilizzato il nuovo tag `<p>` il testo racchiuso al suo interno è un paragrafo. Salviamo index.html e aggiorniamo la pagina web, dovremmo vedere nel browser il nostro messaggio.



I tag HTML

Fatto questo incominciamo ad utilizzare il nostro head, introducendo un altro tag molto importante:

```
<title></title>
```

Questo tag viene utilizzato dai motori di ricerca per sapere il titolo del contenuto che è inserito nella pagina.

```
<!doctype html>  
<html>  
  <head>  
    <title>La mia Pagina Web</title>  
  </head>  
  <body>  
    <p>Ciao, ecco il mio messaggio!</p>  
  </body>  
</html>
```

Il title è visualizzato nelle tab del browser e nell'elenco dei preferiti quando si salva la pagina.

I tag HTML

Altro tag essenziale è il tag `<a>`.

`<a>`

Questo è il tag per definire i link (i collegamenti tra pagine web) ci permette di creare un collegamento dalla nostra pagina ad un'altra pagina.

La sintassi di un link è la seguente:

`Questo è un link al sito di Google`

Il tag `<a>` ha sempre bisogno dell'attributo `href="..."` (Hypertext REference).

Gli attributi sono informazioni che inseriamo dentro i tag con la sintassi `nome-attributo="valore-attributo"`.

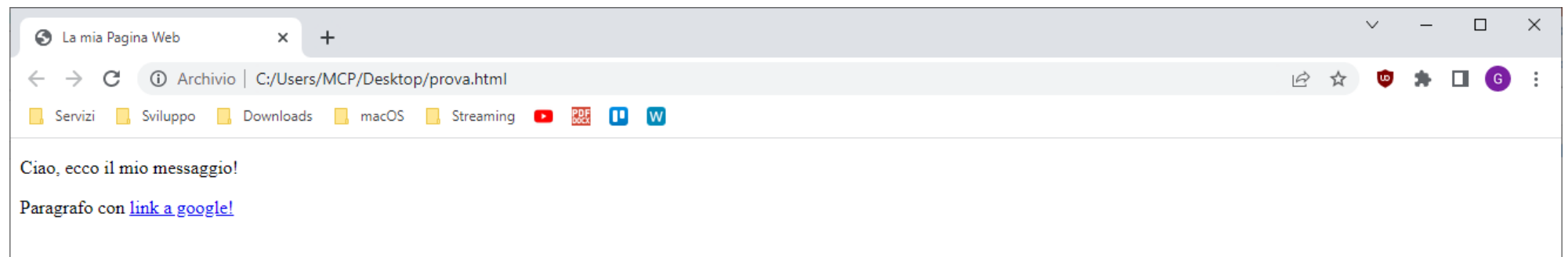
Nel caso dei link l'attributo `href` contiene l'indirizzo della pagina web (l'Url) alla quale vogliamo arrivare tramite il link.

I tag HTML

Se aggiorniamo la nostra pagina con l'esempio appena fatto avremo:

```
<!doctype html>
<html>
  <head>
    <title>La mia Pagina Web</title>
  </head>
  <body>
    <p>Ciao, ecco il mio messaggio!</p>
    <p>Paragrafo con
      <a href="http://www.google.com">Questo è un link a sito di Google</a>
    </p>
  </body>
</html>
```

Ora se abbiamo inserito il link correttamente dovremmo vedere il testo all'interno del tag `` blu sottolineato.



Esercitazione

Proviamo a scrivere il codice per la nostra prima pagina web utilizzando le istruzioni che abbiamo visto inserendo il titolo, qualche paragrafo e qualche link.

I tag HTML

Un altro tag che ha sempre bisogno di un attributo è il tag ``

``

Questo tag necessita sempre dell'attributo `src="..."` (source) nel quale definiamo il percorso dell'immagine da mostrare. A differenza di tutti i tag visti fin ora `` non ha bisogno del tag di chiusura. La sintassi di una immagine base è la seguente:

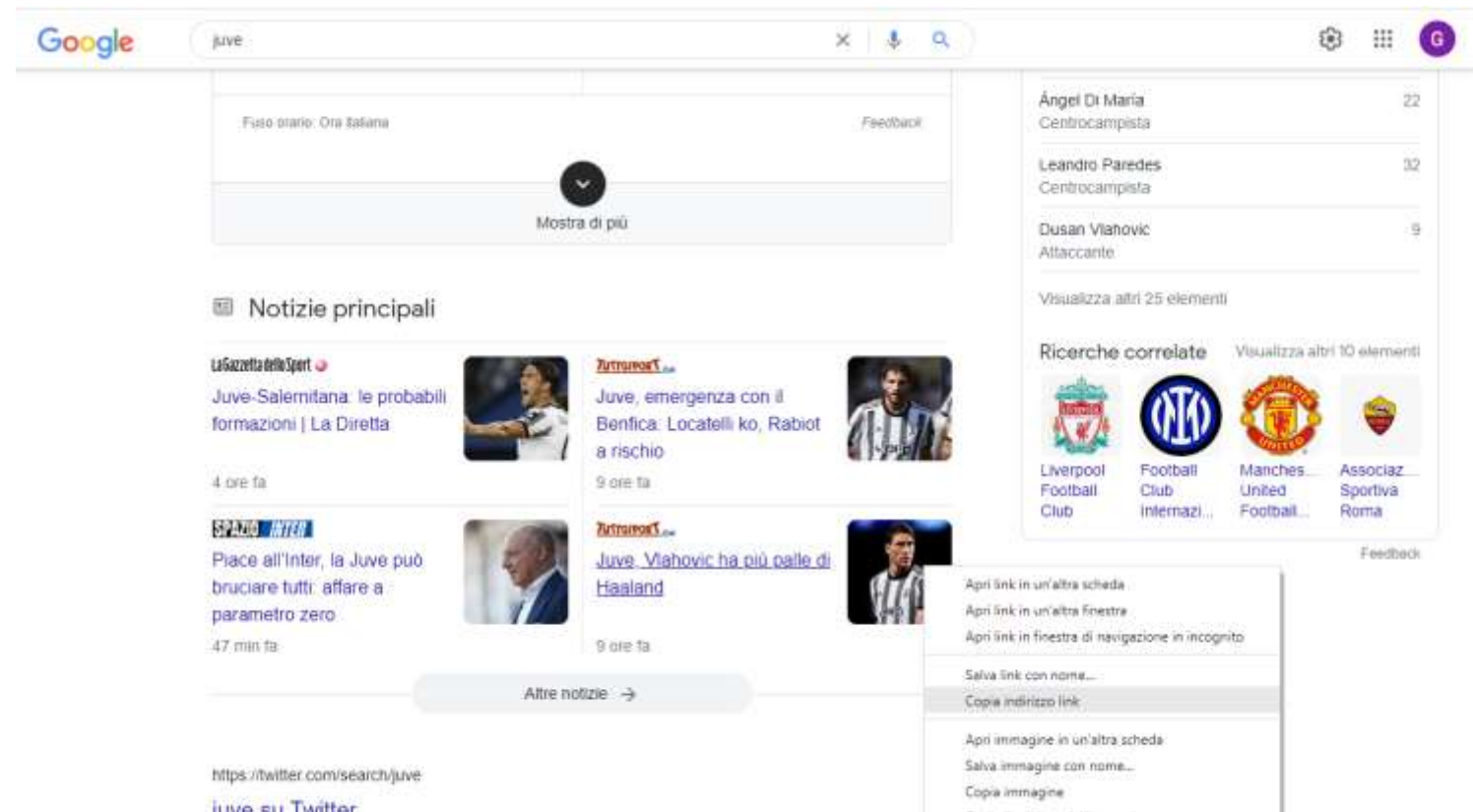
```

```

Notiamo la presenza dell'attributo `alt`, questo attributo viene utilizzato se l'immagine per qualche motivo non viene caricata, è molto importante anche per i motori di ricerca che lo utilizzano per capire cosa c'è nell'immagine.

I tag HTML

Aggiorniamo la nostra pagina con un inserita da google, facendo tasto destro sull'immagine posso copiare l'indirizzo internet



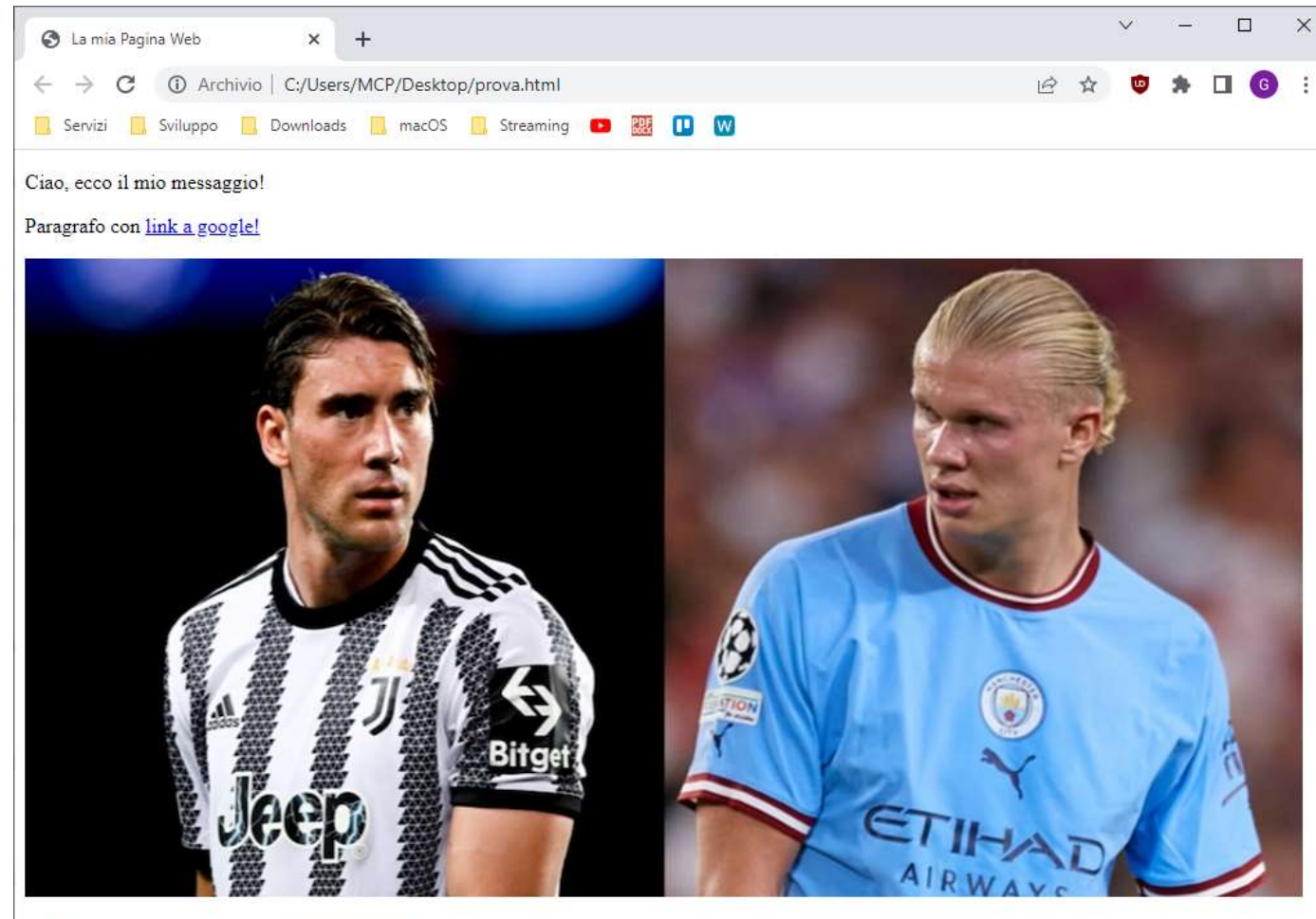
I tag HTML

Inseriamo l'indirizzo copiato nel html.

```
<!doctype html>
<html>
<head>
  <title>La mia Pagina Web</title>
</head>
<body>
  <p>Ciao, ecco il mio messaggio!</p>
  <p>Paragrafo con
    <a href="http://www.google.com">Questo è un link a sito di Google</a>
  </p>
  
</body>
</html>
```


I tag HTML

E aggiorniamo la pagina del browser:



I tag HTML

I tag lista vengono utilizzati spesso all'interno del testo e in modo funzionale nei menu.

` `

Le liste `` (unordered list), vengono utilizzate per rappresentare liste di elementi all'interno di una pagina, la sintassi è la seguente:

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

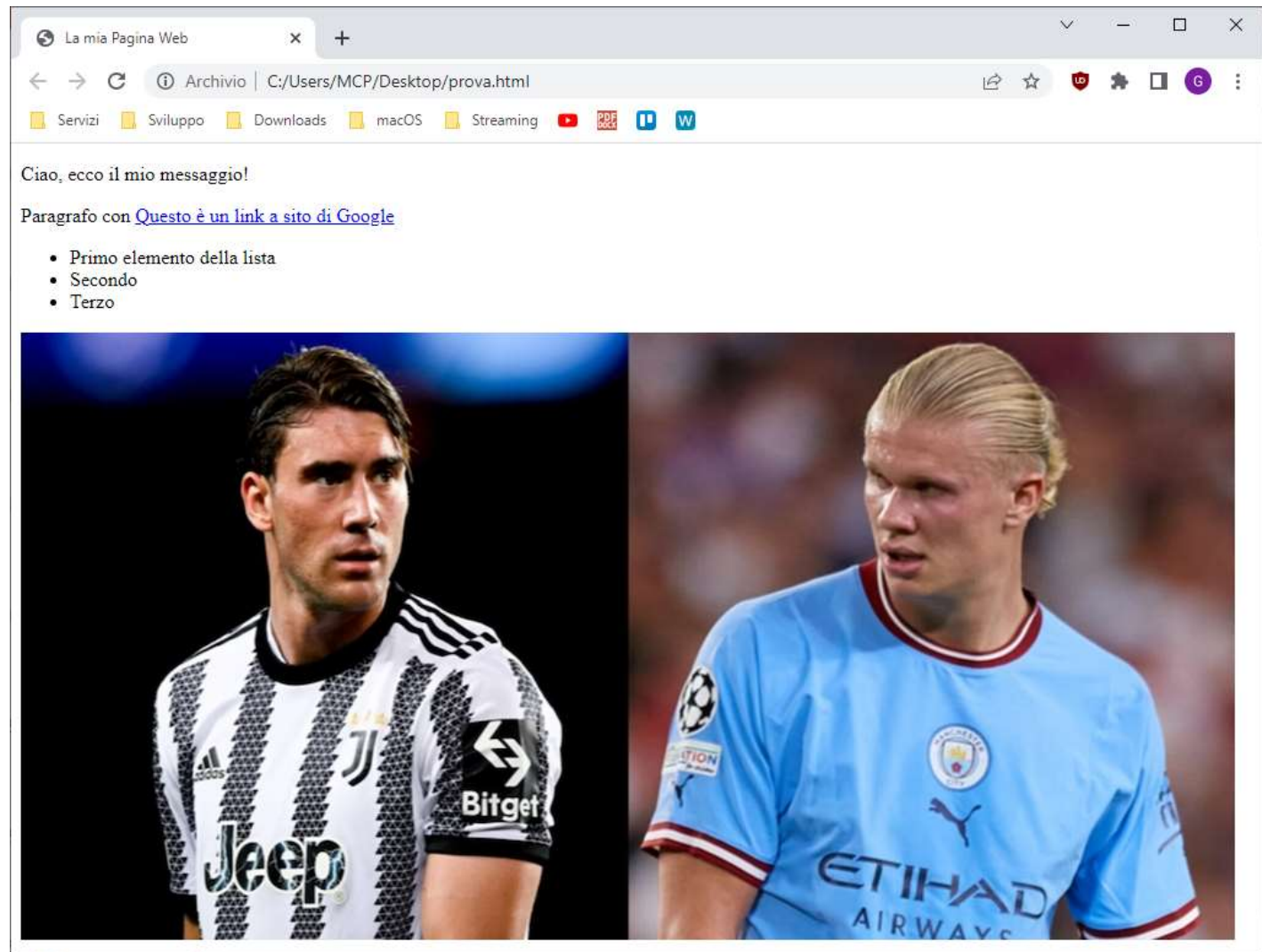
I tag HTML

All'interno dell'ul abbiamo gli (List item), aggiorniamo il nostro esempio:

```
<!doctype html>
<html>
  <head>
    <title>La mia Pagina Web</title>
  </head>
  <body>
    <p>Ciao, ecco il mio messaggio!</p>
    <p>Paragrafo con
      <a href="http://www.google.com">Questo è un link a sito di Google</a>
    </p>
    <ul>
      <li>Primo elemento della lista</li>
      <li>Secondo</li>
      <li>Terzo</li>
    </ul>
    
  </body>
</html>
```

I tag HTML

Ecco il risultato nel browser:



Esercitazione

Proviamo a scrivere il codice per la nostra prima pagina web utilizzando le istruzioni che abbiamo visto inserendo il titolo, qualche paragrafo, qualche link, un piccolo menu e due immagini prese da internet.

I tag HTML

I tag di titolazione vengono utilizzati per definire i titoli con 6 livelli di importanza

`<h1> <h2> <h3> <h4> <h5> <h6>`

Il tag `<h1>` è quello con maggior peso dal punto di vista gerarchico. Questo tag viene utilizzato per definire il titolo principale del nostro contenuto.

Per essere ottimizzata per i motori di ricerca la nostra pagina deve avere un solo h1 con stesso testo del **title** della pagina.

Modifichiamo la nostra pagina html di conseguenza.

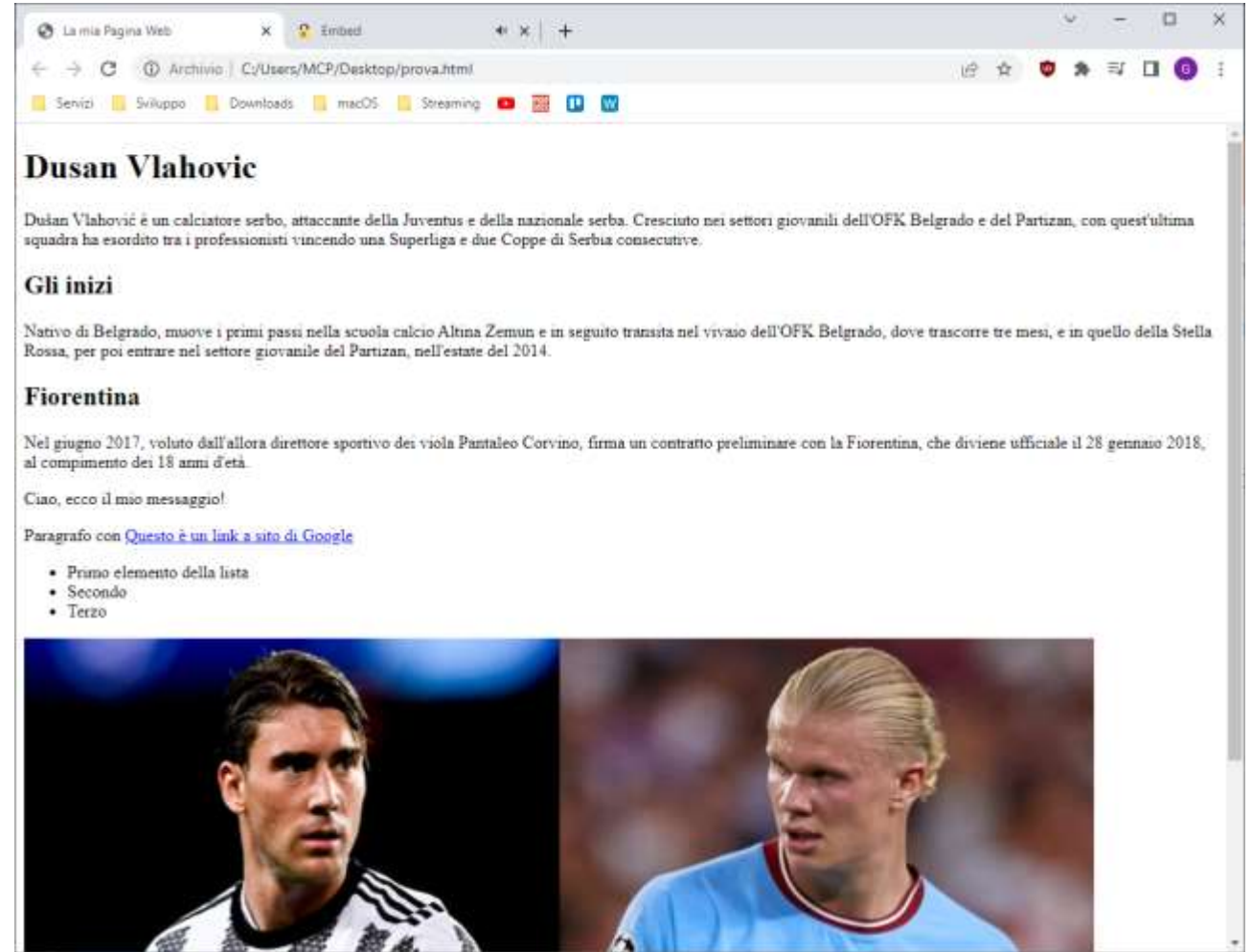
I tag HTML

```
<!doctype html>
<html>
  <head>
    <title>La mia Pagina Web</title>
  </head>
  <body>
    <h1>Dusan Vlahovic</h1>
    <p>Dušan Vlahović è un calciatore serbo, attaccante della Juventus e della nazionale serba. Cresciuto nei settori giovanili dell'OFK Belgrado e del Partizan, con quest'ultima squadra ha esordito tra i professionisti vincendo una Superliga e due Coppe di Serbia consecutive.
    </p>
    <h2>Gli inizi</h2>
    <p>Nativo di Belgrado, muove i primi passi nella scuola calcio Altina Zemun e in seguito transita nel vivaio dell'OFK Belgrado, dove trascorre tre mesi, e in quello della Stella Rossa, per poi entrare nel settore giovanile del Partizan, nell'estate del 2014.
    </p>
    <h2>Fiorentina</h2>
    <p>Nel giugno 2017, voluto dall'allora direttore sportivo dei viola Pantaleo Corvino, firma un contratto preliminare con la Fiorentina, che diviene ufficiale il 28 gennaio 2018, al compimento dei 18 anni d'età.</p>
    <p>Ciao, ecco il mio messaggio!</p>
    <p>Paragrafo con <a href="http://www.google.com">Questo è un link a sito di Google</a></p>
    <ul>
      <li>Primo elemento della lista</li>
      <li>Secondo</li>
      <li>Terzo</li>
    </ul>
    
  </body>
</html>
```

I tag HTML

Nell'esempio con `<h1>` definiamo il titolo del contenuto della pagina, vediamo poi che ho definito degli `<h2>` questi sono i titoli dei sotto-paragrafi.

Si possono utilizzare sotto paragrafi fino a 6 livelli (personalmente non sono mai andato sotto il 4).



I tag HTML: table

Una tabella è una rappresentazione di dati disposti in righe e colonne. In realtà, è più simile a un foglio di calcolo.

In HTML, con l'aiuto delle tabelle, puoi organizzare dati come immagini, testo, collegamenti e così via in righe e colonne di celle.

L'uso delle tabelle nel Web è diventato più popolare di recente grazie agli straordinari tag che riguardano le tabelle HTML e che ne semplificano la creazione e la progettazione.

Quindi per creare una tabella in HTML dovrai usare i tag ed il più importante è il tag `<table>` che è il contenitore principale della tabella, mostra dove inizierà la tabella e dove finirà.

I tag HTML: table

Esempio del codice di una tabella:

```
<table>
  <tr>
    <td>Cella 1</td>
    <td>Cella 2</td>
    <td>Cella 3</td>
  </tr>
  <tr>
    <td>Cella 4</td>
    <td>Cella 5</td>
    <td>Cella 6</td>
  </tr>
</table>
```

Cella 1	Cella 2	Cella 3
Cella 4	Cella 5	Cella 6

I tag HTML: table

Come aggiungere un'intestazione ad una tabella

`<th>` Viene utilizzato per aggiungere intestazioni alle tabelle.

Nei progetti di base l'intestazione della tabella prenderà sempre la riga superiore, il che significa che avremo il `<th>` dichiarato nella prima riga della tabella seguito dai dati effettivi nella tabella.

Per impostazione predefinita, il testo passato nell'Intestazione è centrato e in grassetto.

I tag HTML: table

```
<table>
  <tr>
    <th>NOME DI BATTESIMO</th>
    <th>COGNOME</th>
    <th>INDIRIZZO E-MAIL</th>
  </tr>
  <tr>
    <td>Hillary</td>
    <td>Nyakundi</td>
    <td>tavoli@mail.com</td>
  </tr>
  <tr>
    <td>Lary</td>
    <td>Mak</td>
    <td>sviluppatore@mail.com</td>
  </tr>
</table>
```

NOME DI BATTESIMO	COGNOME	INDIRIZZO E-MAIL
Hillary	Nyakundi	tavoli@mail.com
Lary	ma	sviluppatore@mail.com

I tag HTML: table

Come aggiungere una didascalia ad una tabella

L'uso principale dell'aggiunta di una didascalia alla tabella è fornire una descrizione dei dati rappresentati nella tabella.

La didascalia può essere posizionata nella parte superiore o inferiore della tabella e per impostazione predefinita sarà sempre centrata.

Per inserire una didascalia in una tabella, utilizzare il tag `<caption>`.

I tag HTML: table

```
<table>
  <caption>Risorse di codifica gratuite</caption>
  <tr>
    <th>SITI</th>
    <th>CANALI YOUTUBE</th>
    <th>APP MOBILI</th>
  </tr>
  <tr>
    <td>Campo di codice libero</td>
    <td>Campo di codice libero</td>
    <td>Enki</td>
  </tr>
  <tr>
    <td>Scuole W3</td>
    <td>Accademico</td>
    <td>Eroe della programmazione</td>
  </tr>
  <tr>
    <td>Khan Academy</td>
    <td>Il treno dei codici</td>
    <td>Impara da solo</td>
  </tr>
</table>
```

Risorse di codifica gratuite		
SITI	CANALI YOUTUBE	APP MOBILI
Campo di codice libero	Campo di codice libero	Enki
Scuole W3	Accademico	Eroe della programmazione
Khan Academy	Il treno dei codici	Impara da solo

I tag HTML: table

Come aggiungere un'intestazione, un corpo e un piè di pagina di una tabella

Proprio come un sito Web o qualsiasi altro documento ha tre sezioni principali: intestazione, corpo e piè di pagina, così fa una tabella.

In una tabella sono divisi utilizzando attributi vale a dire:

`<thead>`- fornisce un'intestazione separata per la tabella

`<tbody>`- contiene il contenuto principale della tabella

`<tfoot>`- crea un piè di pagina separato per la tabella

I tag HTML: table

```

<table>
  <thead>
    <tr>
      <th colspan="2">October</th>
      <th colspan="2">November</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Sales</td>
      <td>Profit</td>
      <td>Sales</td>
      <td>Profit</td>
    </tr>
    <tr>
      <td>$200,00</td>
      <td>$50,00</td>
      <td>$300,000</td>
      <td>$70,000</td>
    </tr>
  </tbody>
  <tfoot>
    <tr><th colspan= "4">November was more produstive</th></tr>
  </tfoot>
</table>

```

OCTOBER		NOVEMBER	
Sales	Profit	Sales	Profit
\$200,00	\$50,00	\$300,000	\$70,000
NOVEMBER WAS MORE PRODUSTIVE			

I tag HTML: table

Quando usare una tabella

Ci sono molte situazioni in cui le tabelle potrebbero tornare utili durante lo sviluppo dei tuoi progetti, puoi utilizzare le tabelle quando vuoi confrontare dati con caratteristiche condivise come le differenze tra A e B o i punteggi della squadra X con quelli di Y.

Puoi anche usarne uno se vuoi dare una panoramica dei dati numerici.

Un buon esempio di questo è quando stai cercando di rappresentare i voti degli studenti o anche i punteggi delle squadre.

E una tabella può aiutare a trovare rapidamente informazioni specifiche disposte in modo chiaro. Ad esempio, se stai esaminando un lungo elenco di nomi, una tabella può essere utilizzata per suddividere l'elenco in modo da renderlo facile per i lettori.

Esercitazione

Utilizzando le nozioni che abbiamo appreso realizziamo una tabella simile a quella di seguito mostrata (anche con altri colori):

Squadra	Punteggio	Goal realizzati	Goal subiti
Juventus	100	50	1
Roma	80	40	30
Napoli	50	10	20
....			

Tempo disponibile per la prova: **30 minuti.**

Inviare il file prodotto rinominato come ***cognome.html*** ed inviarlo alla mail:

info@synthesi.net

I tag HTML: div

Il tag **DIV** è uno dei tag più importanti del linguaggio **HTML** in quanto può essere definito come un elemento "neutro" del markup grazie al quale è possibile assolvere diversi compiti tra cui, soprattutto, la definizione di diverse porzioni all'intero di una pagina web.

Il termine DIV nasce come abbreviazione di "divide" (separatore) e, in un certo senso, la funzione di questo tag consiste proprio in questo: creare delle sezioni all'interno di una pagina web al fine di **separare diverse aree della pagina** come, ad esempio, l'*header*, il *menu* ed il *footer*.

Nel web-design tradizionale (cioè precedente ad HTML5), infatti, le aree della pagina vengono definite mediante dei DIV opportunamente stilizzati mediante l'uso dei CSS. In questo modo, pertanto, il DIV estrinseca la sua funzionalità tipica.

I tag HTML: div

Come detto il tag DIV è "neutro" nel senso che, se non opportunamente stilizzato, non offre alcun risultato visibile all'interno della pagina web se non un semplice ritorno a capo (questo perché il DIV è un elemento di tipo *block* e produce, cioè, un ritorno a capo). Quindi:

```
<div>Questa è una linea...</div>
```

```
<div>Questa è un'altra linea...</div>
```

Ne consegue che senza un'opportuna stilizzazione CSS il risultato sarà del tutto insoddisfacente: a differenza di altri tag che posseggono un aspetto caratterizzante il tag DIV è "invisibile" perché appartiene unicamente alla struttura della pagina. Scopo dei DIV, quindi, è quello di fungere da semplici contenitori il cui aspetto, dimensione, posizione e funzione devono essere definiti, di volta in volta, dal designer della pagina web.

I tag HTML: video

L' elemento HTML incorpora un lettore multimediale che supporta la riproduzione di video nel documento. È possibile utilizzarlo anche per il contenuto audio, ma l'elemento potrebbe fornire un'esperienza utente più appropriata.

```
<video controls autoplay>
```

```
  <source src="ilnostrovideo.webm" type="video/webm">  
  <source src="ilnostrovideo.mp4" type="video/mp4">
```

Scarica il

```
<a href="/media/cc0-videos/flower.webm">WEBM</a>
```

oppure in

```
<a href="/media/cc0-videos/flower.mp4">MP4</a>
```

video.

```
</video>
```

I tag HTML: video

autoplay

Un attributo booleano; se specificato, il video inizia automaticamente a essere riprodotto non appena può farlo senza interrompere il caricamento dei dati.

controls

Se questo attributo è presente, il browser offrirà i controlli per consentire all'utente di controllare la riproduzione del video, inclusi volume, ricerca e pausa/ripresa della riproduzione.

height

L'altezza dell'area di visualizzazione del video, in pixel CSS (solo valori assoluti; nessuna percentuale).

I tag HTML: video

loop

Un attributo booleano; se specificato, il browser cercherà automaticamente di tornare all'inizio al raggiungimento della fine del video.

muted

Un attributo booleano che indica l'impostazione predefinita dell'audio contenuto nel video. Se impostato, l'audio verrà inizialmente silenziato. Il suo valore predefinito è false, il che significa che l'audio verrà riprodotto durante la riproduzione del video.

poster

Un URL per un'immagine da mostrare durante il download del video. Se questo attributo non è specificato, non viene visualizzato nulla finché non è disponibile il primo frame, quindi il primo frame viene visualizzato come frame poster.

I tag HTML: video

preload

Questo attributo enumerato ha lo scopo di fornire un suggerimento al browser su ciò che secondo l'autore porterà alla migliore esperienza utente per quanto riguarda il contenuto caricato prima della riproduzione del video. Può avere uno dei seguenti valori:

- None → indica che il video non deve essere precaricato.
- Metadata → indica che vengono recuperati solo i metadati video (ad es. lunghezza).
- Auto → indica che è possibile scaricare l'intero file video, anche se non è previsto che l'utente lo utilizzi.
- stringa vuota → sinonimo del autovalore.

Il valore predefinito è diverso per ciascun browser. Le specifiche consigliano di impostarlo su metadata.

I tag HTML: video

src

L'URL del video da incorporare. Questo è facoltativo; puoi invece utilizzare l' <source> elemento all'interno del blocco video per specificare il video da incorporare.

width

La larghezza dell'area di visualizzazione del video, in pixel CSS (solo valori assoluti; nessuna percentuale).

I tag HTML: nav

L'intestazione di una sezione può contenere elementi di supporto alla navigazione. Se presenti, questi elementi sono racchiusi in un tag `<nav>`.

Non ha importanza che questi link puntino ad aree diverse della stessa pagina, pagine diverse dello stesso sito o esterne a esso. È invece essenziale che questi collegamenti rappresentino un ausilio alla navigazione.

```
<nav>
  <ul>
    <li><a href="articolo_primo.html">Primo articolo</a></li>
    <li><a href="articolo_due.html">Archivio</a></li>
    <li><a href="ultimo_articolo.html">Ultimo Articolo</a></li>
    <li><a href="faq.html">Domande e Risposte (FAQ)</a></li>
  </ul>
</nav>
```

I tag HTML: section

Il tag `<section>` demarca un insieme di contenuti tra loro logicamente correlati.

La specifica fa riferimento in modo esplicito ad alcuni casi d'uso: il capitolo di un libro, le sezioni da cui è composta una tesi, le sezioni di una homepage (contatti, news ecc.).

Esso non è da intendersi come un contenitore generico privo di significato, utilizzato solo per applicare una serie di stili o come obiettivo di codice di scripting. In tal caso rimane sempre valido l'uso dell'elemento `<div>`. Inoltre, non si presta per un contenuto reso disponibile via RSS.

Ricapitolando, è bene non ricorrere a questo tag per soli motivi di stile o di scripting da attribuire a una certa area del documento: in questi casi il tag `<div>` assolve meglio questo compito; quando l'area che si intende contrassegnare è messa a disposizione via RSS; in tal caso il tag `<article>`, introdotto anch'esso con questa versione del linguaggio, si presta meglio allo scopo.

I tag HTML: article

Si usa il tag `<article>` per marcare un contenuto che sia, almeno in linea di principio, utilizzabile o distribuibile in modo indipendente dal resto del documento o anche solo della sezione in cui è racchiuso.

Esempi tipici di questi contenuti possono essere l'articolo di un quotidiano online, la pubblicazione di un messaggio in un forum o presso un blog o, ancora, il commento pubblicato da un utente.

I tag `<article>` possono essere annidati l'uno nell'altro: si pensi per esempio all'elemento più esterno come quello usato per contrassegnare l'articolo di un blog all'interno del quale sono inseriti i commenti degli utenti, ciascuno demarcato a sua volta da un tag `<article>`.

Esempio di codice per <article>

```
<article>
  <a href="open-data-intro.html">
    <header>
      <h1>Open Data libera le informazioni</h1>
      <p>Scritto da Gabriele Gigliotti</p>
      <p>Pubblicato il <time pubdate datetime="2010-10-22T15:30+01:00">22-10-2010</time>.</p>
    </header>
  </a>
  <p>Libero accesso ai dati (grezzi). In questo slogan si racchiude il concetto alla base
    dell'Open Data". Questa libertà presenta due declinazioni:</p>
  <ul>
    <li>rimozione dei vincoli posti da diritti d'autore e, in generale, da licenze ...</li>
    <li>fruizione dei dati attraverso formati non proprietari che agevolino il ...</li>
  </ul>
  <aside>
    <p>"Quali sono i dati di cui si chiede a gran voce la circolazione senza restrizioni?"</p>
  </aside>
  <p>Nel 2009, sir Tim Berners-Lee ha tenuto una sessione al <a href="http://www.ted.com/">
    <abbr title="Technology Entertainment and Design">TED</abbr></a>
    (una serie di conferenze promosse, ogni anno, da una associazione non-profit) ....
  </p>
  <p>Ma quali sono i dati di cui si chiede a gran voce la circolazione senza restrizioni?
    <a href="open-data-intro.html" title="vai alla versione integrale dell'articolo">
      [vai all'articolo]</a>
  </p>
</article>
```

I tag HTML: aside

Questo elemento marca un contenuto che è in relazione con l'elemento in cui `<aside>` è annidato, identificando tuttavia una relazione debole.

Per esempio, se il tag viene usato nell'ambito di un elemento `<article>` si può dire che le informazioni contrassegnate con `<aside>`, se rimosse, non incidono negativamente sulla completezza dell'articolo, poiché il tag `<aside>` si limita ad arricchirlo con contenuti che sono solo "tangenzialmente" correlati.

Questo tag può essere utilizzato per riportare il virgolettato di un testo, per aggiungere alcune note a margine o, infine, per introdurre uno spazio pubblicitario.

I tag HTML: select

Il tag **select** serve, all'interno di un modulo generato in un tag "form", a creare una casella di riepilogo, in sola lettura, da cui è possibile selezionare ed utilizzare i dati ivi compresi.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML select Tag</title>
  </head>
  <body>
    <form action="https://www.dominio.ext/nomefile.php" method="post">
      <select name="dropdown">
        <option value="1" selected>Data Structures</option>
        <option value="2">Data Mining</option>
      </select>
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

I tag HTML: select




Al tag select è necessario assegnare un nome con l'attributo "name", sia per il recupero dei dati lato server che per la verifica delle selezioni lato client. L'attributo "id", invece, è obbligatorio ai fini del rispetto delle specifiche dell'XHTML.

Gli item dell'elenco che si viene a creare vengono definiti nel tag **option**, e relativo tag di chiusura, il quale memorizza il valore dell'item stesso nell'attributo "value", il quale può tranquillamente essere differente dal testo mostrato a video all'interno della casella stessa, come nell'esempio precedente. Ci sono due importanti attributi che possono essere utilizzati per modificare la casella, sia a livello stilistico che funzionale:

- **size** - numerico, stabilisce la lunghezza della "select": considerate che ogni unità, grossomodo, equivale ad una riga in cui viene riportato un singolo item;
- **multiple** - non contiene valori, va specificato così com'è; utile solo in caso in cui sia presente "size", permette la selezione multipla degli item attraverso la pressione del tasto [Ctrl] della tastiera.

I tag HTML: select

Il tag HTML <select> supporta anche i seguenti attributi aggiuntivi:

Attributo	Valore	Descrizione
autofocus 	autofocus	Specifica che al caricamento della pagina il focus dev' essere sulla select.
disabled	disabled	Disabilita l'input. Non saranno ammessi cambiamenti da parte dell' utente e non potrà essere impostato il focus, il tasto tab salterà direttamente la select.
form 	form_id	Specifica uno o più form.
multiple	multiple	Se impostato, potranno essere selezionati più elementi per volta.
name	name	Assegna un nome alla select
required 	richiesto	Prima di inviare i dati sarà obbligatorio scegliere un valore.
size	numerico	Definisce il numero di elementi visibili nella select

Struttura tipica pagina html5

```
<!DOCTYPE html> <!--The new DOCTYPE declaration, much easier than before-->
<head> <!--Like before, this is where the style declaration goes-->
</head>

<body>

  <header> <!--This marks the header of the page and includes the navigation -->
    <nav>
    </nav>
  </header>

  <section> <!--This starts the content section -->

    <header>
    <!--This time the header tag marks the beginning of the content section -->
    </header>

    <article>
    </article>

    <footer>
    <!--As with the header tag, this time the footer marks the end of the content section -->
    </footer>

  </section> <!--This ends the content section -->

  <aside>
  <!--Here comes the sidebar content-->
  </aside>

  <footer>
  <!--The footer, which marks the end of the page, is here-->
  </footer>

</body>
</html>
```

Altri tag HTML

Molti elementi in HTML5 hanno valore semantico che aiuta i motori di ricerca nella indicizzazione.

<header> e <footer>

Come si può immaginare dal nome, questi sono la testata e il piè di pagina (dell'intera pagina o di una singola sezione e possono essere ripetuti nella pagina);

** e <i>**

La "b" sta per bold, ossia grassetto mentre la "i" sta per italic ossia corsivo.

<mark>

Annota una parte di testo per enfatizzarne il significato. I browser che riconoscono e interpretano correttamente questo elemento ne propongono il testo come se fosse evidenziato

Altri tag HTML

<hr>

Le iniziali da cui è composto l'elemento ("hr" sta per Horizontal Rule, ossia riga orizzontale) ne tradiscono quel significato visuale che ora è venuto meno.

In HTML5 <hr> è definito come tag da impiegare per impostare un'interruzione tematica. In altri termini, l'elemento segna il punto di passaggio tra un argomento che si conclude e un altro che inizia: per esempio, un cambio di scena in un copione.

Ricerca i termini

- Definizione di **linguaggio di programmazione**
- Cos'è un **Web Server**
- Cos'è **Linux**



Web Forms 2.0

Un modulo (**form**) è un componente di una pagina Web che dispone di controlli del modulo, come testo, pulsanti, caselle di controllo, intervallo o controlli di selezione del colore.

Un utente può interagire con tale modulo, fornendo dati che possono poi essere inviati al server per ulteriori elaborazioni (es. restituzione dei risultati di una ricerca o di un calcolo). In molti casi non sono necessari script lato client, sebbene sia disponibile un'API in modo che gli script possano aumentare l'esperienza dell'utente o utilizzare moduli per scopi diversi dall'invio di dati a un server. La scrittura di un modulo consiste in diversi passaggi, che possono essere eseguiti in qualsiasi ordine: scrittura dell'interfaccia utente, implementazione dell'elaborazione lato server e configurazione dell'interfaccia utente per comunicare con il server.

Web Forms 2.0

Qualsiasi **form** inizia con un elemento `<form>` all'interno del quale sono inseriti i controlli. La maggior parte dei controlli è rappresentata dall'elemento `<input>`, che per impostazione predefinita fornisce un controllo di testo. Per etichettare un controllo, viene utilizzato l'elemento `label`; il testo dell'etichetta e il controllo stesso vanno all'interno dell'elemento `label`.

Ogni parte di un modulo è considerata un paragrafo ed è in genere separata dalle altre parti mediante elementi. Di seguito un esempio che ci porterà alla fine a creare un modulo per ordinare una pizza:

```
<form>
  <p>
    <label>Customer name: <input></label>
  </p>
</form>
```

Web Forms 2.0

Per consentire all'utente di selezionare la dimensione della pizza, possiamo utilizzare una serie di pulsanti di opzione. Anche i pulsanti di opzione utilizzano l'elemento `<input>`, questa volta con un attributo `type` con il valore `radio`. Per fare in modo che i pulsanti di opzione funzionino come un gruppo, viene assegnato loro un nome comune utilizzando l'attributo `name`. Per raggruppare insieme dei controlli, come i pulsanti di opzione, si può utilizzare l'elemento `fieldset`. Il titolo di un tale gruppo di controlli è dato dal primo elemento in `fieldset`, che deve essere un elemento `legend`.

```
<form>
  <p><label>Customer name: <input></label></p>
  <fieldset>
    <legend> Grandezza della pizza </legend>
    <p><label> <input type=radio name=size value="Piccola"> Piccola </label></p>
    <p><label> <input type=radio name=size value="Media"> Media </label></p>
    <p><label> <input type=radio name=size value="Grande"> Grande </label></p>
  </fieldset>
</form>
```


Web Forms 2.0

Per scegliere i condimenti, possiamo utilizzare le caselle di controllo. Questi usano l'elemento `<input>` con un attributo `type` con il valore `checkbox`:

```
<form>
```

```
.....
```

```
<fieldset>
```

```
<legend> Condimenti </legend>
```

```
<p><label><input type=checkbox name="condimento1" value="Bacon">
```

```
  Bacon </label></p>
```

```
<p><label><input type=checkbox name="condimento2" value="Formaggio">
```

```
  Extra formaggio</label></p>
```

```
<p><label><input type=checkbox name="condimento3" value="Cipolla">
```

```
  Cipolla </label></p>
```

```
<p><label> <input type=checkbox name="condimento4" value="Funghi">
```

```
  Funghi </label></p>
```

```
</fieldset>
```

```
</form>
```

Web Forms 2.0

La pizzeria ha bisogno di un modo per contattare il cliente. A tale scopo, possiamo utilizzare i controlli dei moduli specifici per i numeri di telefono (`<input>` con il `type` impostato su `tel`) e gli indirizzi e-mail (`<input>` con il loro `type` impostato su `email`):

```
<form>
  <p><label>Nome: <input></label></p>
  <p><label>Telefono: <input type=tel></label></p>
  <p><label>Email: <input type=email></label></p>
  <fieldset>
    <legend> Grandezza della pizza</legend>
    <p><label> <input type=radio name=size> Piccola </label></p>
    <p><label> <input type=radio name=size> Media </label></p>
    <p><label> <input type=radio name=size> Grande </label></p>
  </fieldset>
  <fieldset>
    <legend> Condimenti </legend>
    <p><label> <input type=checkbox> Bacon </label></p>
    <p><label> <input type=checkbox> Extra formaggio </label></p>
    <p><label> <input type=checkbox> Cipolla </label></p>
    <p><label> <input type=checkbox> Funghi </label></p>
  </fieldset>
</form>
```

Web Forms 2.0

Possiamo utilizzare `<input>` con il suo `type` impostato su `time` per richiedere un tempo di consegna. Molti di questi controlli hanno attributi per controllare esattamente quali valori possono essere specificati; in questo caso, tre attributi di particolare interesse sono `min`, `max`, e `step`. Questi impostano il tempo minimo, il tempo massimo e l'intervallo tra i valori consentiti (in secondi).

Questa pizzeria consegna solo tra le 11:00 e le 21:00 e non promette niente di meglio di incrementi di 15 minuti, che possiamo contrassegnare come segue:

```
<p><label>Ora di consegna richiesta:  
    <input type=time min="11:00" max="21:00" step="900">  
</label></p>
```

Web Forms 2.0

L'elemento `<textarea>` può essere utilizzato per fornire un controllo di testo multilinea.

In questo caso, lo utilizzeremo per fornire uno spazio in cui il cliente può fornire istruzioni di consegna:

```
<p><label>Istruzioni per la consegna: <textarea></textarea></label></p>
```

Infine, per rendere inviabile la `form` utilizziamo l'elemento `button`:

```
<p><button>Invia ordine</button></p>
```

Web Forms 2.0

Gli invii di moduli sono esposti ai server in vari modi, più comunemente come richieste HTTP GET oppure POST.

Per specificare il metodo esatto viene utilizzato l'attributo `method` della `form`. Tuttavia, questo non specifica come vengono codificati i dati del modulo; per specificarlo utilizziamo l'attributo `enctype`.

Occorre anche specificare l' URL del servizio che gestirà i dati inviati utilizzando l'attributo `action`.

```
<form method="post"  
      action="https://pizza.example.com/ordine.php">  
  ...
```

Web Forms 2.0

```
<form method="post" action="https://pizza.example.com/ordine.php">
```

```

<p><label>Nome: <input name="custname"></label></p>
<p><label>Telefono: <input type=tel name="custtel"></label></p>
<p><label>Email: <input type=email name="custemail"></label></p>
<fieldset>
  <legend> Grandezza della pizza </legend>
  <p><label> <input type=radio name=size value="small"> Piccola </label></p>
  <p><label> <input type=radio name=size value="medium"> Media </label></p>
  <p><label> <input type=radio name=size value="large"> Grande </label></p>
</fieldset>
<fieldset>
  <legend> Condimento </legend>
  <p><label> <input type=checkbox name="topping" value="bacon"> Bacon </label></p>
  <p><label> <input type=checkbox name="topping" value="cheese"> Extra formaggio </label></p>
  <p><label> <input type=checkbox name="topping" value="onion"> Cipolla </label></p>
  <p><label> <input type=checkbox name="topping" value="mushroom"> Funghi </label></p>
</fieldset>
<p><label>Ora di consegna richiesta
  <input type=time min="11:00" max="21:00" step="900" name="delivery"></label></p>
<p><label>Istruzioni per la consegna<textarea name="comments"></textarea></label></p>
<p><button>Invia l'ordine</button></p>

```

```
</form>
```

Web Forms 2.0

Ad esempio, se il cliente ha inserito " Pippo Pelo" come nome, " 3470112233" come numero di telefono, non ha specificato un indirizzo e-mail, ha chiesto una pizza di media grandezza, ha selezionato i condimenti Extra Formaggio e Funghi, inserito un tempo di consegna di 19:00 e lasciato vuoto il controllo del testo delle istruzioni di consegna, la stringa che verrà inviata alla pagina ricevente sarà:

```
custname=Pippo+Pelo&custtel=3470112233&custemail=&size=medium&topping=cheese&topping=mushroom&delivery=19%3A00&comments=
```

%3A → :

Per rendere obbligatorio l'inserimento dei dati in un determinato campo di <input> si usa l'attributo required:

```
<p><label>Nome: <input name="custname" required></label></p>
```

Web Forms 2.0

Di seguito elencato i diversi tipi di <input>

```
<input type="button">  
<input type="checkbox">  
<input type="color">  
<input type="date">  
<input type="datetime-local">  
<input type="email">  
<input type="file">  
<input type="hidden">  
<input type="image">  
<input type="month">  
<input type="number">  
<input type="password">  
<input type="radio">  
<input type="range">  
<input type="reset">  
<input type="search">  
<input type="submit">  
<input type="tel">  
<input type="text">(valore di default)  
<input type="time">  
<input type="url">  
<input type="week">
```


Esercitazione

Utilizzando le nozioni che abbiamo appreso proviamo a realizzare un sito web composto della sola pagina index.html (chi vuole anche più di una, aggiungendo anche una pagina di contatti con un modulo – form - di richiesta informazioni) contenente le nostre informazioni ed eventualmente che proponiamo ad un possibile cliente.

Tempo disponibile per la prova: **60 minuti.**

Inviare il file prodotto rinominato come **cognome.html** ed inviarlo alla mail:
info@synthesi.net



CSS

CSS (da Wikipedia)

Il **CSS** (sigla di **Cascading Style Sheets**, in italiano **fogli di stile a cascata**), in [informatica](#), è un linguaggio usato per definire la [formattazione](#) di documenti [HTML](#), [XHTML](#) e [XML](#), ad esempio i [siti web](#) e relative [pagine web](#). Le regole per comporre il CSS sono contenute in un insieme di direttive (*Recommendations*) emanate a partire dal 1996 dal W3C.

L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una [programmazione](#) più chiara e facile da utilizzare, sia per gli autori delle pagine stesse sia per gli utenti, garantendo contemporaneamente anche il [riutilizzo di codice](#) ed una sua più facile manutenzione.

HTML con CSS

Con questo abbiamo creato la nostra prima pagina html con dei contenuti di prova. Dal punto di vista grafico lascia veramente a desiderare, vedremo ora come migliorare la situazione applicando uno stile grafico con i Css. Il sito avrà un foglio di stile CSS esterno condiviso in tutte le pagine. Partiamo dalla seguente struttura html.

```
<!doctype html>
<html>
  <head>
    <title>Home</title>
  </head>
  <body>
    <h1>Il mio Sito</h1>
    <ul>
      <li> Home </li>
      <li><a href="pagina-2.html">Pagina 2</a></li>
      <li><a href="pagina-3.html">Pagina 3</a></li>
    </ul>
    <h2>Home</h2>
    <p>Home page del mio sito</p>
  </body>
</html>
```

HTML con CSS

Nel codice abbiamo utilizzato una lista `` per creare il menu di navigazione. All'interno della lista `` abbiamo definito 3 pagine. Di queste pagine due hanno il link (Pagina 2 e Pagina 3) e una no (Home), home non necessita di link perché quando visualizziamo questa pagina (la home page) siamo già nella home `index.html`.

I link utilizzati nel menu hanno definito nell'attributo `href=""` percorsi relativi (non hanno un percorso completo `http://www.miosito.com/miapagina.html`). I percorsi relativi funzionano se le pagine che stiamo "linkando" sono nella stessa cartella della home (`index.html`).

HTML con CSS

Per creare la seconda pagina del nostro sito duplichiamo index.html, la rinominiamo in pagina2.html. Fatto questo apriamo pagina-2 nell'editor, cambiamo il title, l'<h2> e il link attivo nel menu, ottenendo il seguente html:

```
<!doctype html>
<html>
  <head>
    <title>Pagina 2</title>
  </head>
  <body>
    <h1>Il mio Sito</h1>
    <ul>
      <li><a href="index.html"> Home </a> </li>
      <li>Pagina 2 </li>
      <li><a href="pagina-3.html">Pagina 3</a> </li>
    </ul>
    <h2>Pagina 2</h2>
    <p>Pagina interna del sito</p>
  </body>
</html>
```

HTML con CSS

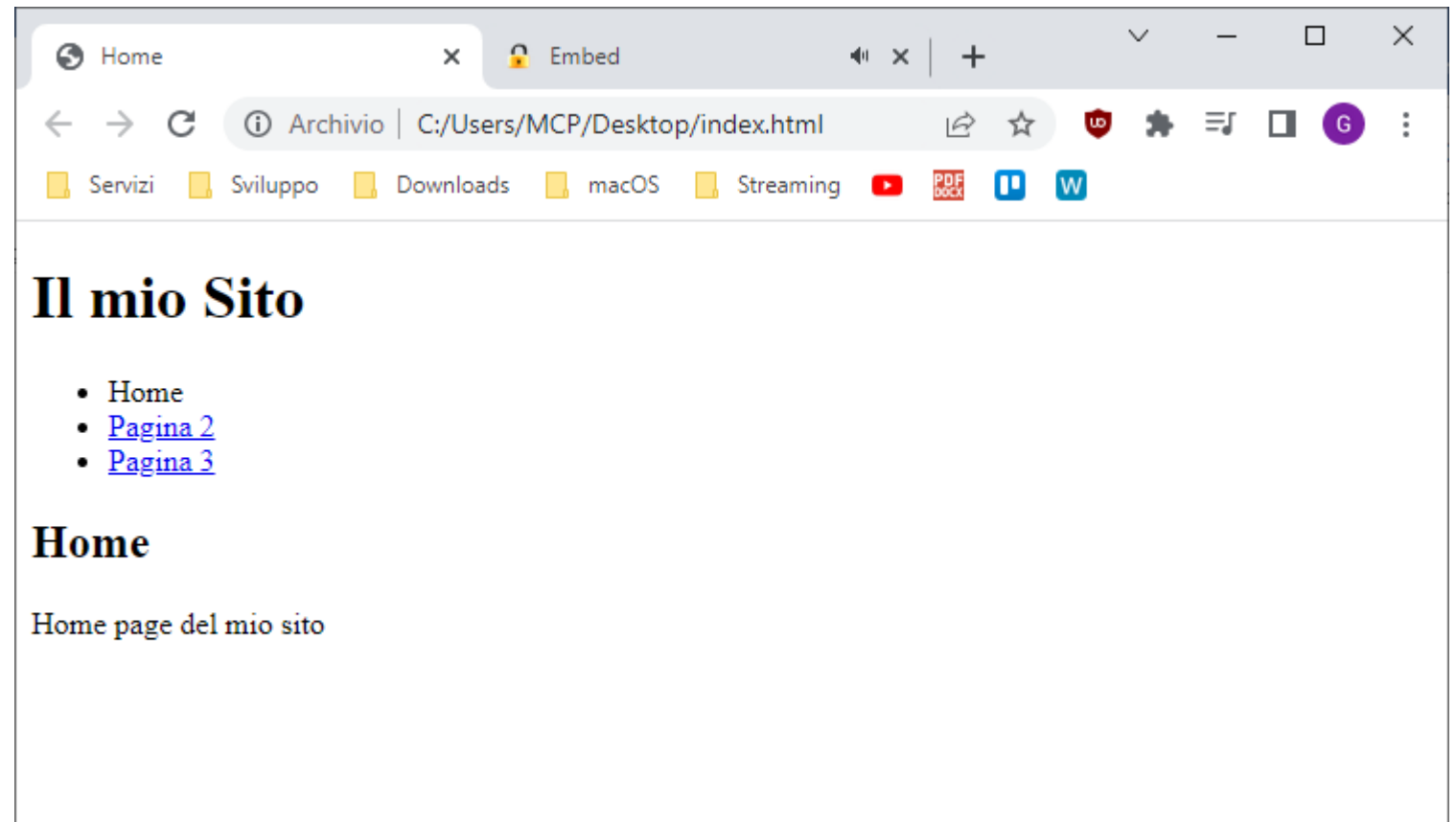
Ripetiamo l'operazione per la pagina 3:

```
<!doctype html>
<html>
<head>
  <title>Pagina 3</title>
</head>
<body>
  <h1>Il mio Sito</h1>
  <ul>
    <li> <a href="index.html"> Home </a> </li>
    <li> <a href="pagina-2.html">Pagina 2 </a></li>
    <li> Pagina 3 </li>
  </ul>
  <h2>Pagina 3</h2>
  <p>Pagina interna del sito</p>
</body>
</html>
```

HTML con CSS

Se abbiamo fatto tutto correttamente cliccando sulle voci del menu dovremmo navigare tra le 3 pagine.

La pagina corrente (nella quale siamo) non deve avere il link attivo.



Ricerca i termini

- Cos'è un **DNS**
- Cosa sono i **Cookie**
- Cos'è **un processore**

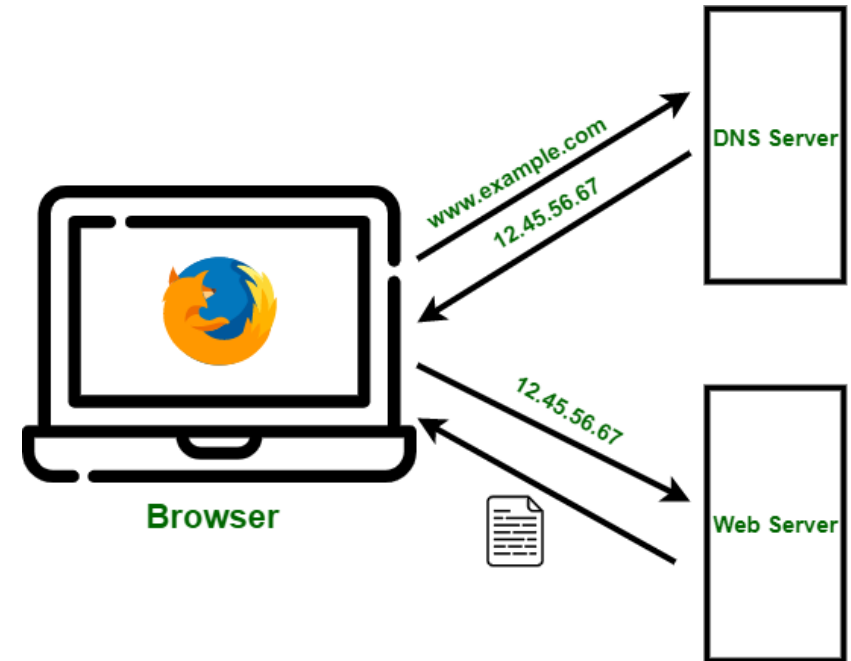


Tempo disponibile 15 minuti

DNS

Il DNS, abbreviazione di **Domain Name System**, è uno dei componenti più comuni ma meno compresi del panorama web. In parole semplici, il DNS aiuta a dirigere il traffico su Internet collegando i nomi di dominio con server web reali. In sostanza, prende una richiesta di un nome di dominio come andoria.it e lo traduce in un indirizzo IP del server, come 82.165.202.42

Dato che il DNS riguarda la ricerca di indirizzi e dispositivi di connessione, molte persone chiamano il DNS «la rubrica di Internet».



Processore

Un processore è composto da due parti principali: L'Unità di

Controllo (UC)

Interpreta le istruzioni del programma memorizzate nella m
controlla l'esecuzione delle istruzioni.

L'Unità Aritmetico Logica (ALU)

Effettua le operazioni aritmetiche e logiche.

L'ALU contiene anche dei registri (o register) ossia degli spazi di memoria in cui memorizzare i dati parziali durante le operazioni di calcolo.



I cookie

I cookie (“biscotto” in inglese) sono dei piccoli file di testo necessari affinché il server del sito web che li ha installati possa ottenere informazioni sulla specifica attività che l'utente compie su quelle pagine web del tipo chi si è collegato a quel sito e che cosa vi ha fatto.

Ogni volta che quel dispositivo si ricollega al sito gli rimanda il cookie e così è possibile riconoscere e tracciare l'attività a distanza di tempo.

Detti file possono essere temporanei “cookie di sessione” e cancellarsi al termine della singola sessione oppure “cookie permanenti” possono rimanere “nascosti” nei meandri delle cartelle del nostro pc, rientrando in collegamento con l'applicazione web ogni volta che l'utente si riconnette al medesimo server remoto.

HTML con CSS

Ora che abbiamo la nostra struttura html base, possiamo pensare a renderla più accattivante grazie ad uno foglio di stile Css.

Apriamo il nostro editor, creiamo un nuovo file e lo chiamiamo stile.css. Apriamo il file appena creato e scriviamo la nostra prima istruzione css.

```
body{  
    background: yellow;  
}
```

Un istruzione CSS è formata dal selettore (nel nostro caso body) e dalla regola (nel nostro caso background:yellow).

Tutte le regole sono racchiuse tra parentesi graffe e vengono applicate al selettore di riferimento che precede le parentesi. Le regole css hanno la prima parte detta proprietà (nel nostro caso background) e la seconda parte detta valore (nel nostro caso yellow).

Nell'esempio yellow è il valore della proprietà background.

HTML con CSS

Per applicare lo stile che abbiamo appena creato al nostro sito dobbiamo includerlo dentro il tag head di tutte le pagine.

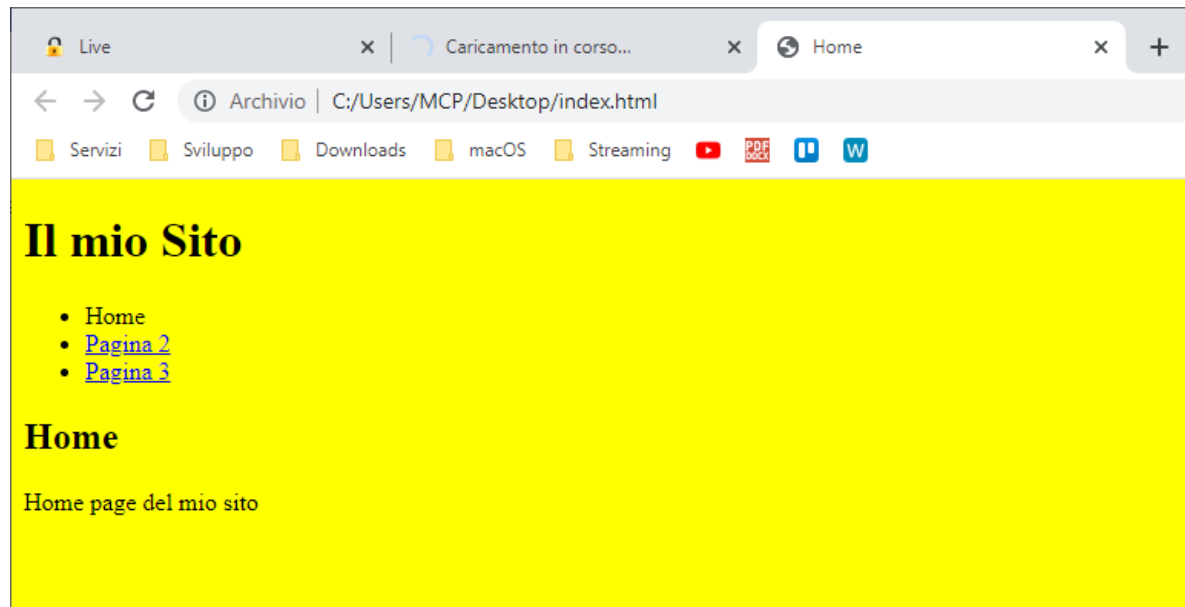
```
<!doctype html>
<html>
<head>
  <title>Home</title>
  <link href="stile.css" rel="stylesheet" type="text/css"/>
</head>
<body>
  <h1>Il mio Sito</h1>
  <ul>
    <li> Home </li>
    <li> <a href="pagina-2.html">Pagina 2</a> </li>
    <li> <a href="pagina-3.html">Pagina 3</a> </li>
  </ul>
  <h2>Home</h2>
  <p>Home page del mio sito</p>
</body>
</html>
```

HTML con CSS

Introduciamo un nuovo tag, il tag `<link>` che serve per inserire risorse esterne nelle pagine web nel nostro caso lo stile:

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

Il tag va inserito nell'header di tutte le pagine del nostro sito quindi anche in pagina-2.html e pagina-3.html. Così avremo lo stile contenuto in style.css condiviso e applicato in tutte le pagine. Se aggiorniamo il browser dovremmo vedere il colore dello sfondo giallo.



HTML con CSS

Le proprietà css a differenza dei tag html sono molte, partiamo da quelle fondamentali e man mano andiamo a vedere quelle più complesse. Come prima cosa impostiamo un carattere (font) per tutte le pagine e aggiustiamo il colore di sfondo.

```
body{  
    background:#dfdfff;  
    font-family: Arial, Verdana, Tahoma;  
}
```

Ora stiamo utilizzando come selettore body ovvero il tag html <body>.

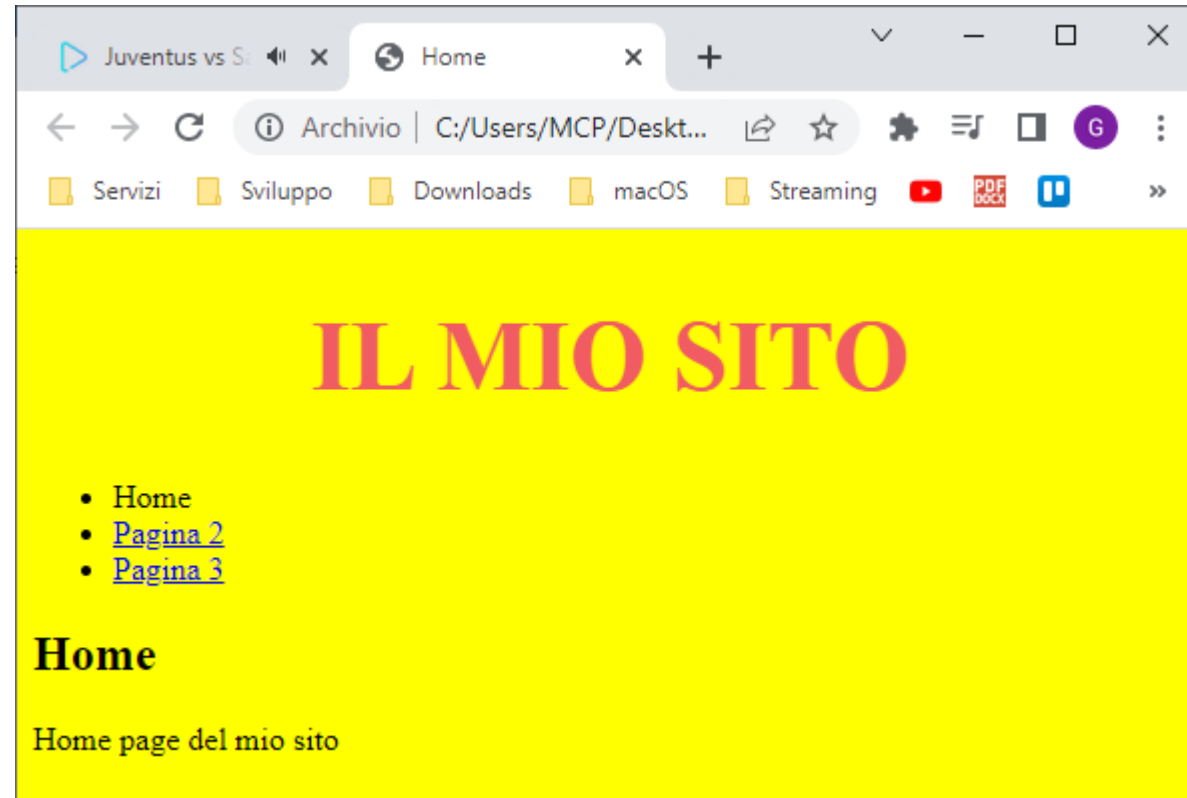
Utilizzando come selettori i tag html possiamo applicare uno stile per tutti i titoli <h1>.

```
h1{  
    font-size: 50px;  
    color: #f15c62;  
    text-transform: uppercase;  
    text-align: center;  
}
```

Con questo codice definiamo la dimensione del titolo con “font-size: 50px” e il colore con “color: #f15c62”. Con “text-transform: uppercase” imposto tutte le lettere del titolo come maiuscole e con “text-align: center” centro il testo.

HTML con CSS

Il risultato è un qualcosa di questo tipo:



Il codice del colore che abbiamo utilizzato è esadecimale #f15c62, possiamo trovare il codice di questo colore con photoshop o altri software di grafica come sketch e illustrator.

CSS – Tipi di grandezza

La grandezza indicata in alcuni proprietà come **width**, **font-size**, **height**, ecc. può essere fissa se indicata in pixel (**px**) oppure relativa. La tabella seguente mostra tutte le **lunghezze relative** usate per ridimensionare al meglio il selettore rispetto allo schermo:

Unità	Descrizione
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

* Viewport = la dimensione della finestra del browser.
Se la finestra è larga 50 cm, 1vw = 0,5 cm.

CSS – Tipi di grandezza

Pixel

Partiamo dall'unità più usata nel web. Fa riferimento alla dimensione dello schermo ed essendo un'unità di misura assoluta, risulta la più semplice da capire e usare. Spesso l'unica usata da molti Web Designer.

Pro: Semplice la gestione e l'interpretazione delle dimensioni sullo schermo.

Contro: Non si sposa bene con i layout responsive, costringendo lo sviluppatore a ridefinire le misure nei dispositivi mobili mediante media-query CSS.

Utilizzo: Potenzialmente ovunque, è effettivamente il sistema più veloce e comodo per montare un sito web, ma ricordatevi che probabilmente dovrete andare a riscrivere queste misure nei layout mobile.

CSS – Tipi di grandezza

Unità assolute “pt, pc, cm, mm, in”

Altre unità assolute, usate esclusivamente per la stampa. Quindi se avete necessità di creare pagine HTML/CSS per grafiche cartacee (ad esempio per la generazione di PDF) possono fare al caso vostro, altrimenti lasciatele stare. Rispettivamente corrispondono a: Punti (pt), Pica (pc, cioè 12 punti), centimetri (cm), millimetri (mm) e pollici (in).

Pro: Fondamentali per la generazione di documenti cartacei

Contro: Inutili per le pagine web.

Utilizzo: Come detto poco fa, solo per la creazione di layout di stampa.

CSS – Tipi di grandezza

Unità relativa “%”

La prima unità di misura relativa. Queste tipologie di unità sono molto più potenti e flessibili delle unità assolute viste poco fa, ma questo rende la loro lettura e interpretazione non sempre immediata.

Nel caso della percentuale, normalmente, il risultato si calcola rispetto al valore della stessa proprietà presente nell'elemento padre. Quindi se abbiamo un tag padre con `width:600px;` andando a creare un elemento figlio al suo interno con `width:50%;` avremo effettivamente creato un elemento con larghezza del contenuto pari a 300px. Se decidessimo di aumentare la larghezza del padre a 800px, in automatico aumenterebbe anche il figlio a 400px (che per l'appunto è sempre il 50%).

CSS – Tipi di grandezza

Questa definizione presenta però delle eccezioni, che vi elenco qui sotto:

- **margin/padding:** fanno riferimento al “width” del padre
- **line-height:** viene calcolato relativamente al font-size dell’elemento stesso, e non del padre
- **width/height:** con elemento in “position:fixed;” fanno riferimento alla dimensione della finestra del browser (viewport)

CSS – Tipi di grandezza

Pro: Unità molto comoda per gestire le dimensioni del box-model, soprattutto per quanto riguarda width/height. Permette di creare elementi fluidi, utili soprattutto nel layout smartphone. Possiamo infatti adattare i contenuti ad ogni schermo senza conoscere a priori la sua dimensione.

Contro: Meno indicato nella gestione delle dimensioni carattere e con eccezioni da tenere a mente. Altro elemento negativo è che non sempre avere layout fluidi semplifica l'organizzazione grafica della pagina.

Utilizzo: Molto utile nel dimensionamento fluido del box-model (width, height, margin e padding). Soprattutto quando non conoscete a priori la dimensione del contenitore in cui andrete a lavorare.

CSS – Tipi di grandezza

Unità relativa “em”

Questa rappresenta l'unità di misura relativa più usata per i font. Infatti 1em rappresenta il 100% del font-size dell'elemento padre. Quindi se ho un elemento padre con “font-size:16px;” e imposto l'elemento figlio con “font-size:1.5em;”, avrò che l'elemento figlio avrà caratteri la cui dimensione sarà 24px. Anche qui l'eccezione è il “line-height” che calcola il valore relativo analizzando il font-size dell'elemento stesso.

Pro: Strumento molto valido per mantenere la proporzione tra titoli, sottotitoli e testi, così da poter ridimensionare in fase di responsive questi elementi, mantenendo però coerente la loro relazione.

Contro: Come per tutte le misure relative, la difficoltà maggiore è tradurre il valore calcolato, cioè la dimensione finale che quell'elemento assume.

Utilizzo: Strumento molto comodo da usare per il dimensionamento dei caratteri, Sconsigliato invece per la gestione del box-model.

CSS – Tipi di grandezza

Unità relativa “rem”

Il “rem” si può vedere come un’evoluzione dell’unità “em”. Molto simile a quest’ultima, presenta però una particolarità fondamentale: 1rem rappresenta il 100% del font-size definito nell’elemento root della pagina, cioè “<html>”. Quindi non è un’unità relativa “a cascata”, ma tutti i rem fanno riferimento ad un unico valore definito in <html>.

Pro: Rispetto ad “em” è semplificata la gestione relativa delle misure, soprattutto quando abbiamo em impostati su elementi inclusi tra di loro.

Contro: Anche qui le difficoltà maggiori sono quelli di traduzione del reale valore.

Utilizzo: Vale lo stesso discorso fatto per gli “em”. L’unica differenza è che qui, se non ridefiniamo il font-size dell’html, il nostro utente avrà un sito totalmente accessibile: potrà infatti gestire la dimensione di tutti i caratteri del sito tramite impostazioni del browser, mantenendo comunque coerenza tra titoli e testo.

CSS – Tipi di grandezza

Le unità relative “vh, vw, vmin, vmax”

Le unità “vh” e “vw”, sono acronimi rispettivamente di “viewport height” e “viewport width” e fanno parte degli aggiornamenti portati con il CSS3. In pratica 1vh rappresenta 1% dell'altezza della finestra del browser e 1vw l'1% della larghezza della finestra del browser. Queste 2 unità di misura risultano utilissime quando vogliamo dimensionare elementi in base allo spazio fornito dal browser stesso (ad esempio mettere un banner full-screen). Le versioni vmin/vmax (non pienamente compatibili), permettono di scegliere un valore in base al valore minimo (o massimo) tra larghezza e altezza della finestra.

Pro: L'unico strumento CSS a disposizione per avere un chiaro riferimento il relazione alla dimensione del viewport, cioè la finestra del browser.

Contro: Unità di misura nuova e non del tutto compatibile, soprattutto con browser precedenti. Unità utile in casi ben specifici.

Utilizzo: Strumento molto utile per lavorare alle dimensioni del box-model il relazione alla finestra del browser.

CSS – Tipi di grandezza

Le unità relative “ch, ex”

Sono unità relative che impostano la loro dimensioni sulla base della larghezza del carattere “0” (zero) nel caso di “ch”, e dell’altezza del carattere “x” nel caso di “ex”. Sono unità di misura usate molto raramente perché poco utili a livello pratico.

L’unica utilità che posso immaginare è data da “ch” quando si utilizzano font “monospace”.

Occhio però alla retro-compatibilità (novità CSS3).

CSS – Tipi di grandezza

Compatibilità

Ora che abbiamo visto tutte le unità di misura disponibili è giusto vedere la loro compatibilità. Anche perchè alcune di loro nascono con il CSS3 e questo ne limita l'uso sui browser più datati.

Qui sotto una tabella riassuntiva presa direttamente dal sito [W3schools](http://W3schools.com):

Length Unit					
em, ex, %, px, cm, mm, in, pt, pc	1.0	3.0	1.0	1.0	3.5
ch	27.0	9.0	1.0	7.0	20.0
rem	4.0	9.0	3.6	4.1	11.6
vh, vw	20.0	9.0	19.0	6.0	20.0
vmin	20.0	9.0*	19.0	6.0	20.0
vmax	26.0	Not supported	19.0	7.0	20.0

Note: Internet Explorer 9 supports vmin with the non-standard name: vm.

CSS – Tipi di grandezza

Come gestire il line-height

Questo elemento infatti (line-height) serve a definire l'altezza della riga di un testo. Potete usare delle unità di misura assolute o relative. Ma potete anche usare numeri senza alcuna unità di misura. In questo caso, il valore sarà un moltiplicatore del font-size dell'elemento stesso.

Molti pensano che questo sia la stessa cosa che succede usando le unità relative come em, rem o %. Ma non è così. Nel caso di unità relative, infatti, tale valore calcolato, si ripercuoterà così com'è anche agli elementi figlio, indipendentemente dal font-size.

Mentre, usando un valore “senza unità”, avrete un normalissimo moltiplicatore che sarà usato così com'è anche agli elementi figli.

Quindi, il mio consiglio è di cercare di usare un line-height SENZA unità di misura.

Esercitazione

Create una piccola pagina html con del testo e formattatelo in grandezza utilizzando i diversi tipi per vedere come si comportano.

Esempio

```
<p style="font-size: 16px">Testo di esempio</p>
```

Tempo per la prova: 15 minuti

CSS – display

La proprietà **display** specifica il comportamento di visualizzazione di un elemento.

Valore	Descrizione
inline	Displays an element as an inline element (like). Any height and width properties will have no effect
block	Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
contents	Makes the container disappear, making the child elements children of the element the next level up in the DOM
flex	Displays an element as a block-level flex container
grid	Displays an element as a block-level grid container
inline-block	Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values
inline-flex	Displays an element as an inline-level flex container
inline-grid	Displays an element as an inline-level grid container
inline-table	The element is displayed as an inline-level table
list-item	Let the element behave like a element
run-in	Displays an element as either block or inline, depending on context
table	Let the element behave like a <table> element
table-caption	Let the element behave like a <caption> element
table-column-group	Let the element behave like a <colgroup> element
table-header-group	Let the element behave like a <thead> element
table-footer-group	Let the element behave like a <tfoot> element
table-row-group	Let the element behave like a <tbody> element
table-cell	Let the element behave like a <td> element
table-column	Let the element behave like a <col> element
table-row	Let the element behave like a <tr> element
none	The element is completely removed
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

HTML con CSS

Possiamo definire uno stile sempre utilizzando i tag html come selettori per i paragrafi <p> e i titoli <h2>

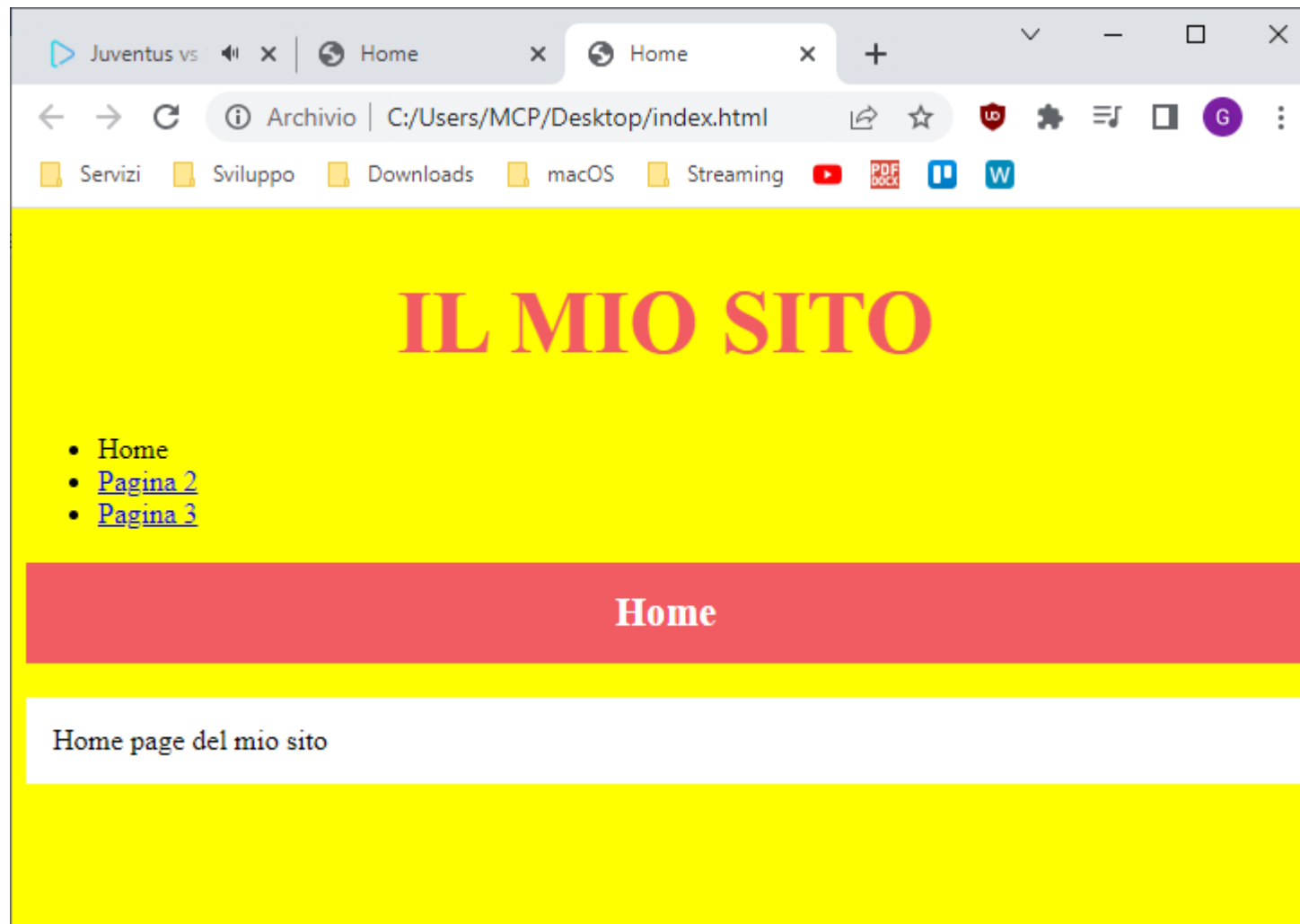
```
p{  
    background: #fff;  
    padding:15px;  
}  
h2{  
    font-size: 22px;  
    color: #fff;  
    background: #f15c62;  
    padding:15px;  
    text-align: center;  
}
```

In questo codice utilizziamo le proprietà viste prima (color, font-size) e ne introduciamo di nuove. La proprietà background viene utilizzata per definire il colore dello sfondo (nel nostro caso del paragrafo).

Padding viene utilizzata per dare una spaziatura interna di 15px al nostro paragrafo.

HTML con CSS

00



HTML con CSS

Una tecnica spesso utilizzata è quella di applicare proprietà a più elementi utilizzando più selettori separati da virgola, per esempio:

```
h1, a{color: #f15c62;}
```

Con questa istruzione sto applicando il colore **#f15c62** a tutti i titoli `<h1>` e a tutti i link `<a>` con un'unica istruzione.

Questo ci permette di definire lo stile degli elementi in maniera veloce e intelligente.

HTML con CSS

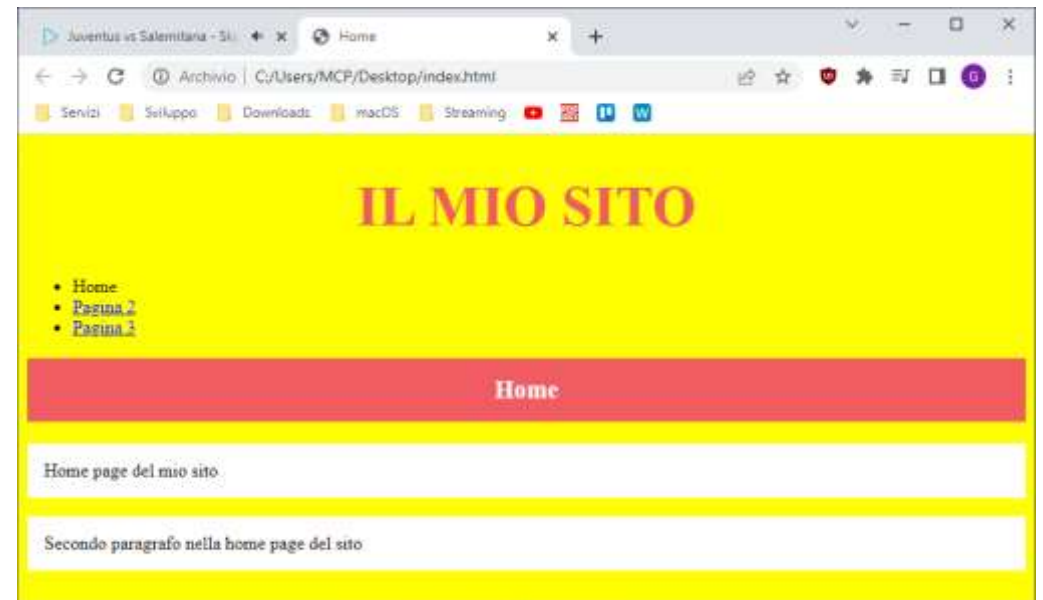
Abbiamo applicato un stile a tutti i paragrafi se ne duplico uno quindi dovremmo vedere lo stesso stile applicato al secondo <p>

<h2>Home</h2>

<p>Home page del mio sito</p>

<p>Secondo paragrafo nella home page del sito</p>

Con il seguente risultato nel browser:



HTML con CSS

Se volessimo però avere uno stile differente solo per un paragrafo dobbiamo utilizzare una classe.

Per creare una classe prima la dobbiamo dichiarare nel foglio di stile con la seguente e sintassi:

```
.dark {  
    background: #333;  
    color:#fff;  
}
```

Con *.nome-classe* definiamo la classe, all'interno delle parentesi graffe definiamo le proprietà. Fatto questo applichiamo la classe all'elemento nel file html.

```
<p>Home page del mio sito</p>  
<p class="dark">Secondo paragrafo nella home page del sito</p>
```

HTML con CSS

Di seguito come verrà visualizzata la pagina:

Selezionando gli elementi con le Classi siamo più specifici rispetto la selezione con i tag html.

Questo vuol dire che la classe .dark applicata al elemento p sovrascrive le proprietà di <p>.

```
p{  
  background: #fff;  
  padding:15px;  
}  
.dark{  
  background: #333;  
  color:#fff;  
}
```



HTML con CSS

Le classi sono più specifiche dei tag html nel css, se avessimo invece due selettori uguali e di pari livello:

```
p{  
    background: #fff;  
    padding:15px;  
}  
p{  
    background: #333;  
    color:#fff;  
}
```

Le proprietà che vengono applicate sono le ultime background:#333 e color:#fff.

Questo perché i fogli di stile CSS sono processati dall'alto verso il basso (in cascata), a parità di specificità vengono applicate le proprietà dell'ultimo selettore.

I selettori CSS e gli ID

Le classi possono essere applicate a più elementi nella stessa pagina per esempio:

```
<p>Home page del mio sito</p>
```

```
<p class="dark">Secondo paragrafo nella home page del sito</p>
```

```
<p class="dark">Terzo paragrafo nella home page del sito</p>
```

Gli id invece sono univoci possono essere applicati a un solo elemento.

```
<p id="colorfull">Quarto paragrafo nella home page del sito</p>
```

Il selettore id è più specifico delle classi, nel foglio di stile ha la seguente sintassi.

```
#colorfull{  
    background: #f15c62;  
    color:#333;  
}
```

I selettori CSS e gli ID

Il cancelletto è seguito dal nome del id. Il risultato nel browser sarà il seguente:



I selettori CSS (esempi)

Nello sviluppo di siti web si tende a utilizzare più le classi rispetto che gli id, questo perché l'id è molto specifico. **Nella nostra pagina di esempio possiamo avere un solo id** id="colorfull". Se avessimo l'esigenza di utilizzare quello stile su un altro elemento non potremmo farlo.

Per uno stile ripetibile su più elementi devo utilizzare una classe.

Un esempio classico dell'utilizzo di un id è il menu di navigazione. Nella nostra struttura html inseriamo (nella lista utilizzata per creare il menu) l'id navigazione.

```
<ul id="navigazione">  
  <li> Home </li>  
  <li> <a href="pagina-2.html">Pagina 2</a> </li>  
  <li> <a href="pagina-3.html">Pagina 3</a> </li>  
</ul>
```

I selettori CSS (esempi)

A seguire nel foglio di stile definisco le proprietà dell'id.

```
#navigazione{text-align: center;}  
#navigazione li {display: inline;}
```

Con questo codice allineo al centro la lista #navigazione. Con il selettore combinato **#navigazione li** vado ad applicare la proprietà **display:inline** a tutti gli elementi della lista che **id=navigazione**.

Il risultato nel browser è un menu centrato.



TEST - Domanda

Ora facciamo una piccola verifica per vedere se hai capito la base dei selettori CSS.

Se volessi applicare un colore a tutti i link `<a>` del nostro sito che sintassi CSS dovrei utilizzare?

TEST - Risposta

`a{color: #codice-colore}`

per esempio --> `a{color: #f15c62;}`



Ricerca i termini

- Cos'è un **DATABASE**
- Elenco di almeno 10 **linguaggi** di programmazione



Tempo disponibile 10 minuti

Le proprietà

Una caratteristica molto interessante dei CSS sono le proprietà (o variabili) personalizzate, che consentono a framework come Bootstrap di utilizzare un unico valore definito ovunque e ovunque.

Per essere chiari, mentre i CSS le chiamano proprietà personalizzate, ho sempre pensato a queste definizioni come costanti perché è quello che sono veramente. Un valore viene assegnato a un nome e quel valore sarà rappresentato da quel nome per tutta la durata del documento.

Ad esempio, potrei creare una proprietà personalizzata chiamata "primary" che assomiglia a questa:

```
:root {  
  --primary: #0d6efd  
}
```

Le proprietà

Quindi, quando voglio usare quel colore, diciamo come sfondo per un pulsante, lo farei riferimento in questo modo:

```
button {  
  background-color: var(--primary)  
}
```

Questo mi dà la possibilità di apportare una modifica al CSS e rifletterla ovunque venga utilizzato il colore e rende la personalizzazione dei progetti molto più semplice che se il valore del colore fosse impostato in ogni elemento CSS per ogni luogo in cui volevo usarlo.

Esercitazione

Utilizzando le nozioni che abbiamo appreso proviamo a realizzare un sito web composto da tre pagine con un header che contiene un menu le cui voci, poste in orizzontale, che richiamano le altre pagine. Le voci del menu devono avere un fondo colorato che cambia al passaggio del mouse (a:hover). Tutte le pagine devono avere una intestazione, un corpo ed un footer, un file style.css per il codice css e dovranno essere:

- 1) **Home** con due sezioni con del testo
- 2) **Chi siamo** con delle informazioni su eventuali servizi offerti, ecc.
- 3) **Contatti** con un modulo di richiesta informazioni (nome, cognome, email, telefono) che richiami la pagina <https://www.andoria.it/risultato.php>

Tempo disponibile per la prova: **60 minuti.**

Inviare il file prodotto rinominato come ***cognome-css.html*** ed inviarlo alla mail:
info@synthesi.net

Ricerca i termini

- Cos'è la metodologia **AGILE**
- Cos'è il **CLOUD**
- Cosa sono i **BIG DATA**



Tempo disponibile 10 minuti

5

BEM

Block Element Modifier

Introduzione

Partiamo dalle basi: BEM non è un framework ma una metodologia, un modo particolare di scrivere e organizzare il codice CSS che possiamo definire “BEM Philosophy”, un gioco di parole per spiegare i risultati che si possono raggiungere attraverso l'utilizzo di questa tecnica.

Abbiamo visto come SASS ci permetta di migliorare la stesura dei fogli di stile. Per rendere ancora più leggibile il nostro codice possiamo implementare lo standard BEM (Block Element Modifier).

L'idea che sta alla base della **metodologia BEM** è quella di definire alcune regole per la definizione delle classi degli elementi basandosi sulle proprietà degli elementi stessi. La definizione delle classi si basa su tre componenti: il blocco, l'elemento e la variante.

.block : rappresenta l'elemento padre, può essere innestato in altri blocchi ma non deve essere dipendente da essi;

.block__element : elemento, è dipendente dal blocco;

.block-modifier : serve a modificare lo stile di base, può essere applicato sia a un blocco che ad un elemento.

Il blocco

Il blocco è il contenitore o il contesto in cui l'elemento si trova.

È un'entità indipendente che può essere semplice o composta da ulteriori blocchi.

L'elemento

Il secondo componente della metodologia BEM è l'elemento. L'elemento è un singolo componente inserito in un blocco.

Esempi di elementi sono un logo, un input, un link.

La classe di ogni elemento viene definita da un nome semplice, chiaro e univoco preceduto dal blocco e da due underscore:

`blocco__elemento`

La classe identificativa del logo sarà, ad esempio:

`header__logo`

mentre quella dei link di navigazione sarà:

`header__link`

La variante

L'ultimo componente da valutare è la variante che definisce versioni differenti di un elemento.

La variante andrà specificata come terzo parametro della classe e verrà separata dall'elemento con due trattini. Possiamo ad esempio identificare i link di navigazione come:

`header__link`

e i link selezionati come:

`header__link-selected`

Con queste tre semplici componenti andremo a definire le classi di tutti i tag della nostra pagina HTML.

SAS e BEM

Spesso la variante modifica solamente una proprietà di un elemento: colore, font, background. Supponiamo di aver definito una classe per i link presenti nell'header:

```
.header__link {  
    font-size: 14px;  
    font-weight: bold;  
    text-decoration: none;  
    color: #000;  
}
```

Abbiamo ora la necessità di definire una variante di questi link con colore del testo rosso. Seguendo la metodologia BEM dovremmo creare una variante:

```
.header__link--red {  
    color: #FF0000;  
}
```

SAS e BEM

Dovremmo poi arricchire il nostro elemento con entrambe le classi:

```
<a href="http://html.it" class="header__link header__link--red">Link</a>
```

In alternativa possiamo sfruttare SASS ed in particolare la funzione di @extend e modificare il precedente codice in questo modo:

```
.header__link {  
  font-size: 14px;  
  font-weight: bold;  
  text-decoration: none;  
  color: #000;  
}
```

```
.header__link--red {  
  @extend .header__link;  
  color: #FF0000;  
}
```

```
<a href="http://html.it" class="header__link--red">Link</a>
```

Con questa soluzione, ogni modifica sulla classe `header__link` andrà a modificare anche le varianti.

Esempio

```
<div class="nav">
  <ul class="nav__list">
    <li class="nav__list_item nav__list_item--visited"><a href=""></a></li>
    <li class="nav__list_item"><a href=""></a></li>
    <li class="nav__list_item"><a href=""></a></li>
    <li class="nav__list_item"><a href=""></a></li>
  </ul>
</div>
```

```
.header{background-color: blue; min-height: 80px; display: flex; justify-content: space-between; align-items: center;}
.header__logo{padding: 0 15px; color: white;}

.nav{background-color: brown;}
.nav__list{padding: 20px; list-style-type: none;}
.nav__list_item{display: inline-block;}
.nav__list_item a{padding: 0 20px; color: white; text-decoration: none;}
.nav__list_item--visited{background-color: red;}
```

Esercitazione

Provare a modificare il file style.css utilizzato per il precedente esercizio utilizzando la metodologia BEM.

Tempo disponibile per la prova: **15 minuti.**

Links

<https://www.color-hex.com/>

<https://css-tricks.com/>

<https://developer.mozilla.org/en-US/>

<https://marketplace.visualstudio.com/>

<https://caniuse.com/>

<https://docs.emmet.io/>