

ITS INFORMATION AND COMMUNICATIONS TECHNOLOGY Academy

WEB1

ANDREA DIMITRI

Anno Accademico
2023-2024



INFORMATION AND COMMUNICATIONS TECHNOLOGY



HTML e CSS3

andrea.dimitri@uniroma2.it





HTML: let's start with a simple page

`<!doctype html>`

`<html lang="it">`

`<head>`

`<title>Ciao Mondo!</title>`

`</head>`

`<body>`

`<h1>Ciao Mondo!</h1>`

`<p>Questa è la nostra prima
HTML!</p>`

`</body>`

`</html>`

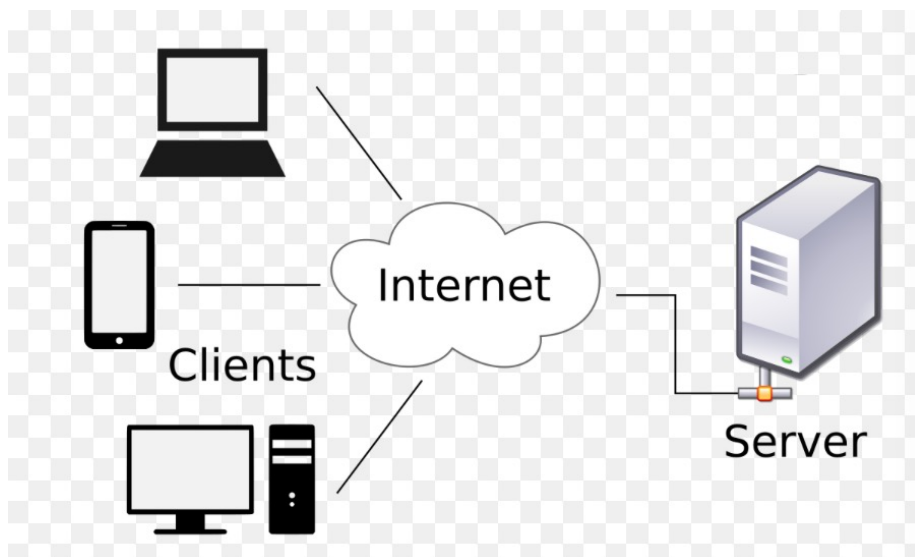
Tag	Descrizione
<code><!DOCTYPE html></code>	Serve semplicemente a dire che il file è una pagina HTML e in particolare che si tratta di un documento secondo lo standard HTML5. Lo si mette punto e basta.
<code><html></code>	È il tag che racchiude tutta la pagina e ci permette ad esempio di definire quale sia la lingua della pagina, come nel nostro caso in cui abbiamo definito che il documento è in Italiano grazie all'attributo <code>lang="it"</code>
<code><head></code>	Questo tag contiene una serie di informazioni utili per la gestione della pagina, nel nostro caso abbiamo definito ad esempio quale sarà il titolo che apparirà sui motori di ricerca e sulle linguette del browser, grazie al tag <code><title></code> , ma questa sezione è destinata a contenere molto di più: dall'inserimento di librerie alla definizione dell'insieme di caratteri. Vedremo tutto in seguito.
<code><body></code>	Contiene gli elementi della pagina, tutto il contenuto e tutti i relativi tag che saranno poi resi a video vengono inseriti qui dentro, come abbiamo fatto nel nostro caso con il titolo e il paragrafo.

- 1) In your working directory create a folder to host your first html page
- 2) Create a new file called index.html and copy the red text in this slide .
- 3) Open the file with your favorite browser.

We created our first HTML page !!



Client-Server Architecture



Un server, spesso, processa più richieste in parallelo. Nella figura la nuvola definisce la rete Internet, ma per essere precisi dobbiamo scrivere una generica rete.

Il web ha un'architettura client-server.

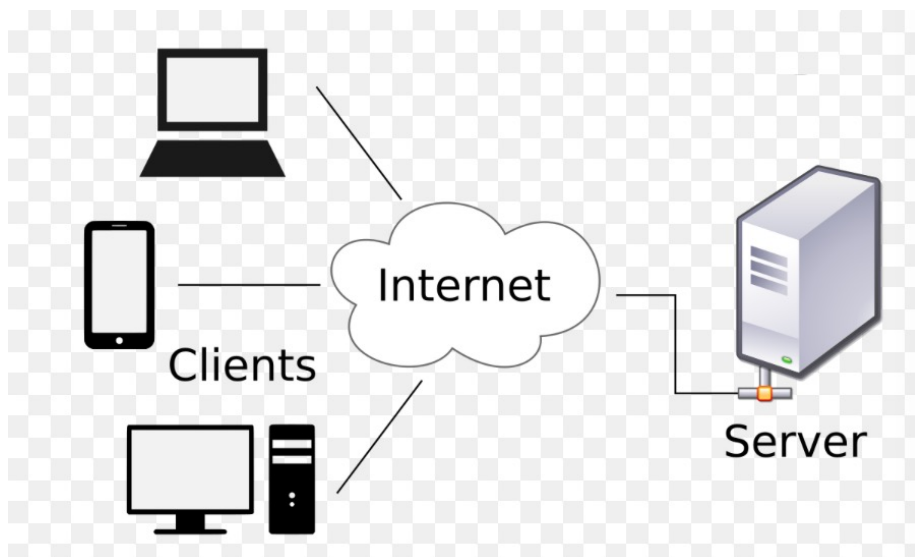
Clients significa programmi che, usando la rete (usualmente la rete internet, ma non necessariamente) inviano dati al server e aspettano una risposta.

Un server è un'applicazione che, all'avvio, si lega ad un device di rete e quindi aspetta che arrivino richieste.

Quando arriva una richiesta il server la processa e produce una risposta che invia al client. La risposta è inviata al client usando la rete.



Client-Server Architecture



I nostri client:

- Il browser (installato su PC oppure su un altro dispositivo, es. uno smartphone)
- Un'APP che ha delle librerie in grado di leggere e tradurre graficamente uno stream HTML. Es. programma scritto in uno dei linguaggi che abbiamo studiato.

Per ora non useremo un server. Scriveremo pagine HTML e verificheremo il funzionamento/effetto grafico delle stesse usando un browser.

Quando parleremo delle FORM html introdurremo il server.



HTML: insert an image in the page

```

```

For details and other features:

https://www.w3schools.com/tags/tag_img.asp



HTML: LINKs

```
<a href="https://www.uffizi.it/gli-uffizi">Visit  
  Florence !</a>
```

For details:

https://www.w3schools.com/tags/tag_a.asp



PUT IT ALL TOGETHER

You can add to your page an image that is, at the same time, an hyperlink:

```
<a href="https://www.w3schools.com">  
    
</a>
```




HTML: LISTS

<p>The ol element defines an ordered list:</p>

Coffee

Tea

Milk

<p>The ul element defines an unordered list:</p>

Coffee

Tea

Milk



Exercise

Let's build a simple web site with :

- A starting page with a list of links
- For each link a html page with a title and an image and a text description.
- In every page there are links to go to the starting page.



Html tables

Let's read together:

https://www.w3schools.com/html/html_tables.asp

Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Try it Yourself »



Html tables: an exercise

Consider the following link and import the referred table in your PC:

https://colorlib.com/etc/tb/Table_Responsive_v1/index.html

Steps:

- 1) Open a new html file with a header and a body
- 2) Using the browser get the header part of the page and copy it in your file
- 3) Using the browser get the body part of the page and copy it in your file
- 4) Read all files referred in the head of the html and download them in your pc
- 5) Redefine local links and put all together.



HTML IFRAME

Let'S READ TOGETHER:

https://www.w3schools.com/tags/tag_iframe.ASP



Exercise ONE

Using tag `iframe` with a table, create an html page to see contemporarely 4 videos, for example 4 football matchs.



Exercise TWO

Using tag `iframe` with a table, create an html page to see a page with a press review.





CSS: i fogli di stile

Cascading Style Sheets (CSS) è usato per definire il layout di una pagina web. Con i CSS, puoi controllare il colore, il font, la dimensione dei testi, lo spazio tra elementi della tua pagina, come i vari elementi della pagina sono posizionati, le immagini o i colori in background, scegliere differenti formati per differenti devices e screen sizes, e molto altro ancora!

CSS-Esempio1-PaginaBase mostra i tre modi in cui i CSS possono essere usati all'interno di una pagina html (pagina11.html)



SELETTORI (1)

I selettori sono usati per indicare a quali elementi di una pagina html vuoi applicare un certo stile.

```
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

p è il selettore: indica che le due proprietà (text-align:center e color:red) si applicano a tutti i paragrafi della pagina html, cioè a tutti gli elementi inclusi in <p> </p>



SELETTORI (2)

I selettori sono usati per indicare a quali elementi di una pagina html vuoi applicare un certo stile.

```
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
```

si applica a <p **id="para1"**>Hello world</p>

Cioè a tutti gli
elementi che hanno
id="para1"

```
<style>
.center {
  text-align: center;
  color: red;
}
</style>
```

si applica a <h1 **class="center"**>



Selettori semplice

All CSS Simple Selectors

Selector	Example	Example description
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>element.class</u>	p.intro	Selects only <p> elements with class="intro"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element,...</u>	div, p	Selects all <div> elements and all <p> elements

Quelli che abbiamo visto sono i selettori semplici, perchè si applicano a singoli elementi della pagina html.

Vedremo nella prossima slide selettori che si applicano a combinazioni di più elementi della pagina html.

Vedi:
Esempio2-CssSelettori



SELETTORI combinati

I selettori sono usati per indicare a quali elementi di una pagina html vuoi applicare un certo stile.

```
<style>
div p {
  background-color: yellow;
}
</style>
```

```
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
```

Cioè a tutti gli elementi che hanno **<div>** e dentro **<p>**

All CSS Combinator Selectors

Selector	Example	Example description
<u><i>element element</i></u>	div p	Selects all <p> elements inside <div> elements
<u><i>element>element</i></u>	div > p	Selects all <p> elements where the parent is a <div> element
<u><i>element+element</i></u>	div + p	Selects the first <p> element that are placed immediately after <div> elements
<u><i>element1~element2</i></u>	p ~ ul	Selects every element that are preceded by a <p> element



CSS: facciamo un esercizio(1)

```
<!doctype html>
<html>
<head>
<title>I miei quadri</title>
<link href="style.css" type="text/css"
      rel="stylesheet">
</head>
<body>
</body>
</html>
```

Facciamo un esercizio un po' più complesso per capire la potenzialità dei css. Creiamo un contenitore di file img solo con css. Il primo step è creare una pagina html standard nel file index.html.

Da notare il riferimento al file style.css che dobbiamo creare nella stessa cartella.

Vedi:
Esempio3-EsercizioCss per
l'esercizio step by step.



CSS: facciamo un esercizio (2)

```
<!--Step1-->
<div class="main-container">
  <div class="image-container">
    
    
    
    
    
    
  </div>

  <div class="button-container">
    <a href="" class="button-css">b1</a>
    <a href="" class="button-css">b2</a>
    <a href="" class="button-css">b3</a>
  </div>
</div>
<!--Fine Step1-->
```

Creiamo un body fatto di un container (il div più esterno) Che contiene al suo interno due container. Il primo è una sequenza di immagini. Il secondo è una sequenza di bottoni.

Ciascun **div** ha associata una classe che ne definisce le caratteristiche in termini di stile. Le classi sono definite nel file css. Dobbiamo definire nel file style.css 5 classi:

- main-container
- image-container
- image-css
- button-container
- button-css



CSS: facciamo un esercizio (3)

Osserviamo il file style.css e le 4 classi ognuna per ogni div che abbiamo creato.

Usiamo il link <https://www.w3schools.com/css/default.asp> per andare a vedere il significato delle varie property.

I bottoni non hanno associata nessuna azione in questo html.

Adesso modifichiamo l'href dei bottoni aggiungendoci il contenuto come nella slide che segue.

Quello che faremo, come risultato finale dello step 2, è:

- Ogni bottone, premuto cambia il valore di "image-container" e quindi il background color associato.



CSS: facciamo un esercizio (4)

I bottoni che abbiamo creato non hanno associata nessuna azione. Usiamoli per cambiare lo stile dei containers che compongono la nostra pagina. Nel file index.html aggiungiamo due parti:

```
<div class="slider-container">
  <span id="slider-image-1"></span>
  <span id="slider-image-2"></span>
  <span id="slider-image-3"></span>

  <div class="image-container">
    
    
    
    
    
    
  </div>

  <div class="button-container">
    <a href="#slider-image-1" class="button-css">b1</a>
    <a href="#slider-image-2" class="button-css">b2</a>
    <a href="#slider-image-3" class="button-css">b3</a>
  </div>
</div>
```

Aggiungiamo tre tag `` e modifichiamo l'href dei bottoni.

Modifichiamo l'href dei bottoni in modo da richiamare lo `` dell'html che ha l'unica funzione di richiamare il riferimento CSS associato (vedi la prossima slide)



CSS: facciamo un esercizio (5)

I bottoni che abbiamo creato non hanno associata nessuna azione. Usiamoli per cambiare lo stile dei containers che compongono la nostra pagina. Nel file index.html aggiungiamo due parti:

```
<div class="slider-container">
  <span id="slider-image-1"></span>
  <span id="slider-image-2"></span>
  <span id="slider-image-3"></span>

  <div class="image-container">
    
    
    
    
    
    
  </div>

  <div class="button-container">
    <a href="#slider-image-1" class="button-css">b1</a>
    <a href="#slider-image-2" class="button-css">b2</a>
    <a href="#slider-image-3" class="button-css">b3</a>
  </div>
</div>
```

``
è un link che inizia con # e
quindi punta ad un certo
elemento della pagina che
diventa il **target element**.

Questo significa che,
cliccando b1 il target
element è

``



CSS: facciamo un esercizio (6)

I bottoni che abbiamo creato non hanno associata nessuna azione. Usiamoli per cambiare lo stile dei containers che compongono la nostra pagina. Nel file index.html aggiungiamo due parti:

```
<div class="slider-container">
  <span id="slider-image-1"></span>
  <span id="slider-image-2"></span>
  <span id="slider-image-3"></span>

  <div class="image-container">
    
    
    
    
    
    
  </div>

  <div class="button-container">
    <a href="#slider-image-1" class="button-css">b1</a>
    <a href="#slider-image-2" class="button-css">b2</a>
    <a href="#slider-image-3" class="button-css">b3</a>
  </div>
</div>
```

Se il target element è

allora

#slider-image-1:target ~ .image-container

diventa il selector attuale del <div class="image-container">

:target è il selettore che identifica il target element.



CSS: facciamo un esercizio (7)

Per ciascun id associato allo creiamo l'apposito riferimento nel file style.css in modo da associare ad ogni bottone una funzione diversa.

```
#slider-image-1:target ~ .image-container  
{  
  background-color: brown;  
}
```

Con questa sintassi associamo al bottone un'azione che si effettua sul target (.image-container) che cambierà colore.



CSS: facciamo un esercizio (8)

Nel file scommentiamo **step3** per vedere un altro modo per creare una galleria di immagini con il css.

Poi scommentiamo **step4** per i bottoni

Float:left regola l'allineamento degli elementi

Per interrompere tale allineamento usiamo

Clear=both



CSS: creiamo slide scorrevoli

Usando la versione 3 di CSS (CSS3) è possibile usare la proprietà **animation** e creare delle pagine html dinamiche con il solo CSS. L'esempio:

E:\MOV\html-css\Esempio1-PaginaBase\css_esercizio_avanzato2

mostra questo.

<div class="slide-wrapper"> identifica la classe in cui è usata la proprietà **animation** con il nome **EffettoSlide**.

```
.slide-wrapper {  
  width: 2912px;  
  animation: EffettoSlide 10s ease infinite; /*CSS3 animation*/  
}
```

Il termine **ease** definisce un timing personalizzato che è poi definito nella struttura **EffettoSlide** che si trova in style.css.

L'esempio è da completare con le immagini che devono essere messe su tutte le slide. Tale completamento può essere fatto per esercizio.



Interagire con l'utente: le form html

Una form HTML è usata per raccogliere dati attraverso l'interazione con l'utente. I dati inseriti possono essere inviati per il processamento:

- ad una funzione javascript locale
- al server remoto

Vedi HTML-FormHtml per alcuni esempi di form. Parti da [formSemplice.html](#)



formSemplice.html(1)

```
body {  
    margin-left: 30%;  
    margin-right: 30%;  
    font-family: Arial;  
    font-size: 22px;  
    text-align: center;  
    background: #a9d6bf;  
}
```

Queste sono le impostazioni di stile riferite al `<body>.....</body>`

Possiamo regolare:

- La posizione degli oggetti attraverso `margin-left` e `margin-right`
- Le caratteristiche del testo (font family e font size)
- L'allineamento del testo (`text-align`)
- Il colore di sfondo



formSemplice.html(2)

```
p {  
    text-align: left;  
    color: red;  
}
```

Il primo stype p{ si riferisce ai paragrafi. Rende il colore del testo rosso e allinea il testo a sinistra e non centrato come definito nel body.

```
#btnid {  
    text-align: center;  
}
```

Il secondo fa si che i bottoni siano centrati.

```
input[type = submit], input[type =  
    reset] {  
    background-color: orange;  
    font-size: 22px;  
}
```

Definisce le impostazioni dei due bottoni che sono orange e hanno una dimensione definita da 22px



formSemplice.html(3)

```
<form action="stat.php" method="post">
```

`action="stat.php"`

Fa riferimento al programma CGI sul server che viene richiamato e che riceverà i dati per il processamento e la creazione della pagina HTML di risposta.

`method="post"`

Fa riferimento alla modalità usata per inviare i dati e quindi al comando HTTP.

I due metodi principali sono:

GET

POST

Provare i due metodi e vedere le differenze. Nel caso delle GET notare il legame tra il formato della stringa inviata e l'html (<input> tag).



Html Form: i tipi di dato

Le form html servono per:

- Permettere all'utente di inserire i dati **in modo semplice**
- Evitare **errori e ambiguità** nei dati stessi.

Es. mettere DataNascita come campo di testo può creare errori e ambiguità

Es. mettere in un unico campo Nome e Cognome può creare ambiguità



Html Form: i tipi di dato

Per evitare **errori e ambiguità** le form sono **costruite** in modo da guidare l'utente nell'inserimento dei dati.

Vedi form[Intermedia.html](#)

Quando studieremo il javascript, vedremo ulteriori strumenti per validare il contenuto dei dati immesso dall'utente attraverso una form.



Leggere le pagine HTML degli altri

Possiamo sempre leggere il codice HTML e in generale il codice sorgente di una pagina HTML che scarichiamo da Internet. (es. scarichiamo la pagina di www.subito.it e riscriviamo il titolo).

Questo può darci una metodologia di lavoro. Scegliamo un sito che ci piace e poi lo personalizziamo.

Tuttavia questa cosa non è facile. Proviamo a capire come si fa e quali sono i limiti di questo approccio.

Proviamo a rifare l'esercizio di cui sopra, ognuno con un sito web a sua scelta. Cioè scarichiamo la pagina e cambiamo una componente.



La libreria jquery-mobile

Ormai le pagine html sono complesse. Questo significa che non lavoreremo mai con l'HTML di base, ma con delle librerie che implementano queste complessità e ci aiutano.

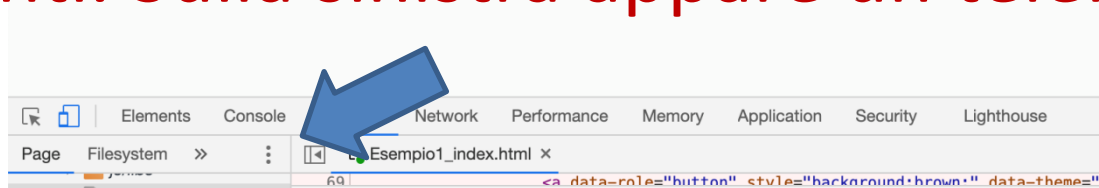


Lavorare con il browser

Una volta scritta una pagina html, il browser ci permette di visualizzarla e quindi anche verificare quanto scritto.

Il browser ci da una serie di strumenti che utilizzeremo. Per esempio ci permette di verificare che le nostre pagine siano “responsive”.

Con Chrome premere F12 per accedere a questi strumenti. Sulla sinistra appare un telefonino.





Iniziamo a lavorare.....

(Considerare sequenza di esempi nel file
file:///Users/andrea/Progetti/AppPertiniFalcone3/www/Esempio1_index.html)

+ lista.html

+ paginaBaseConTable.html

a partire dal sito degli esempi di jquery mobile



Il tag VIDEO (1)

```
<video width="320" height="240" controls>  
<source src="VIDEO.mp4" type="video/mp4">  
<source src="VIDEO.avi" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

L'attributo **controls** aggiunge i controlli video attraverso i tasti play, pause, e volume.

E' possibile specificare più video (**source**) alternativi. Il browser sceglierà il primo che riesce a leggere.

Il testo (parte in blu) è visualizzata solo se il browser non supporta nessuno dei formati video messi a disposizione.

Oltre all'attributo **controls** si possono aggiungere gli attributi **autoplay** per avviare il video autonomamente e **muted** per eseguire il video in silenzioso.



Il tag **VIDEO**(2)

In javascript `mydocument.getElementById` permette il liking all'elemento **VIDEO** per un miglior controllo dello stesso.

Vedi esempio: EsempioVideo.html

Vedi esempio: EsempioVideo3.html (mostrare come si costruisce un esempio dalla sorgente web)