

Università Degli Studi di Ferrara

Corso di Laurea in Informatica - A.A. 2023 - 2024

Tecnologie Web

Lez. 04 - HTML

Cap. 1

Tecnologie Server Side

In questa lezione...

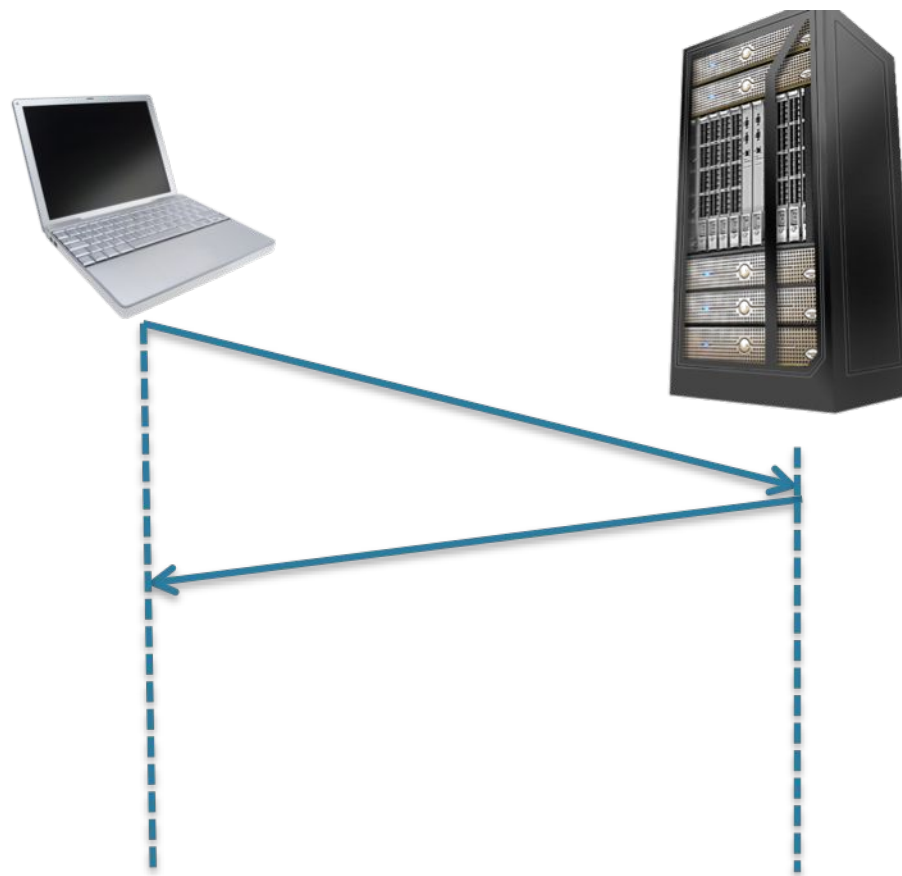
- La struttura del documento HTML
- I principali tag HTML
- I fogli di stile CSS

Modello Client-Server

Il modello di comunicazione client-server è basato su due entità:

Client (Browser)

Effettua la richiesta di accesso (visita) di una pagina web al Web Server



Web Server

Risponde alla richiesta inviando un documento HTML

HTML

Hyper**T**ext **M**arkup **L**anguage

NON È: Un linguaggio di programmazione

È: Un linguaggio a marcatori

HTML

Lo scopo del linguaggio HTML è quello di **rappresentare** (cioè visualizzare) le informazioni.

Esistono altri linguaggi a marcatori che hanno scopi diversi dalla rappresentazione delle informazioni, come ad esempio l'XML che è orientato alla **descrizione** della struttura dei dati.

I Tag HTML

I **Tag** sono gli elementi fondamentali del linguaggio HTML

Sono:

delle parole chiave racchiuse fra parentesi angolari < ... >

Usati:

(solitamente) in “coppia” come ad esempio: <p> e </p>

Consentono :

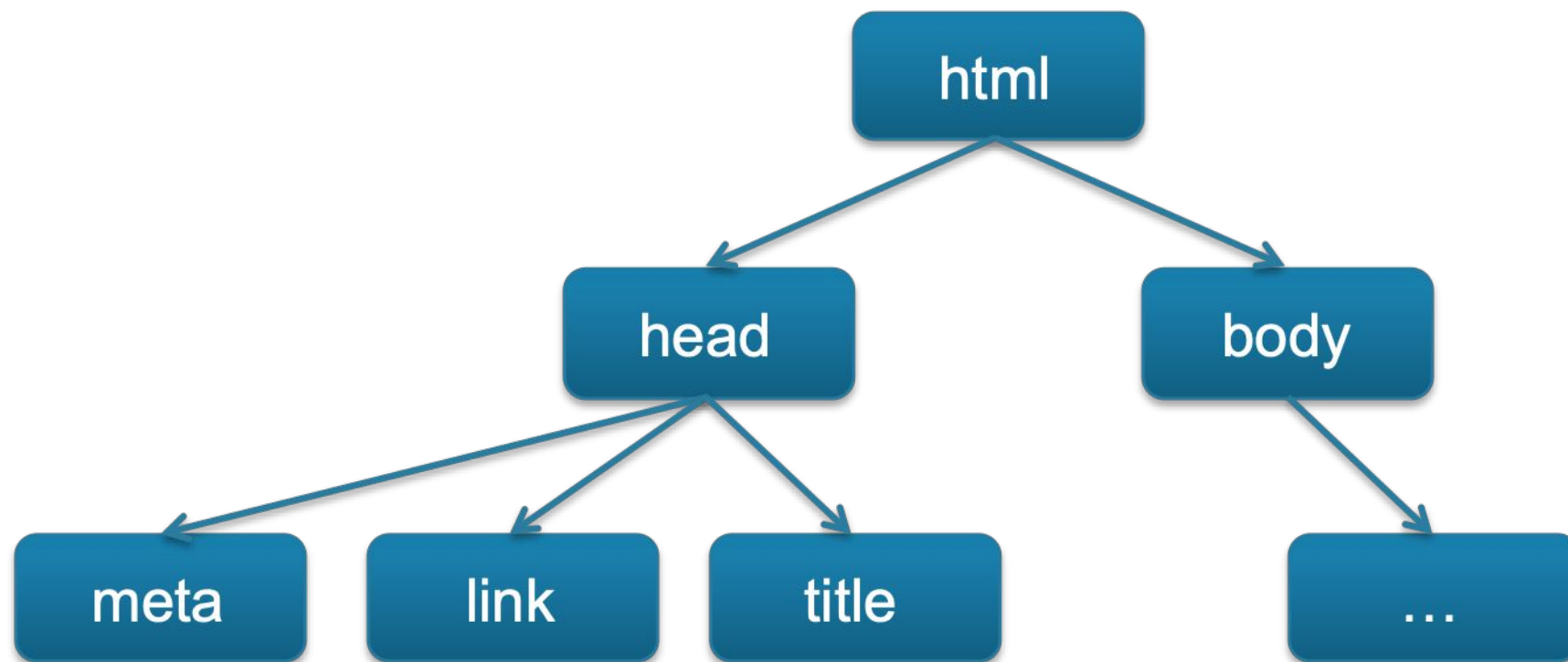
- Formattare il testo (h, p, strong, ...);

- Definire hyperlink (a);

- Raccogliere informazioni dall'utente (form).

HTML

Un documento HTML ha una struttura ad **albero** i cui elementi (i tag) sono annidati creando una gerarchia



Tag di struttura

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

`<html>` è il tag radice

`<head>` identifica la sezione header

`<body>` identifica il corpo del documento

Tag di blocco e liste

```
<body>
```

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

```
<hX></hX>
```

```
<a></a>
```

```
<div></div>
```

```
<ol></ol>
```

```
<ul>
```

```
  <li></li>
```

```
</ul>
```

```
<img />
```

```
<br />
```

```
</body>
```

Tutti i tag di blocco o di lista vanno racchiusi all'interno del tag `body`

I tag `ol` ed `ul` vengono utilizzati per gli elenchi, i primi per gli elenchi ordinati, i secondi per gli elenchi non ordinati

Il tag `li` viene utilizzato per gli elementi degli elenchi e devono essere all'interno dei tag `ul` o `ol`

I tag `img` e `br`, sono tag “speciali” che possono non avere il tag di chiusura

Attributi dei Tag

Tutti i Tag HTML ammettono attributi.

Gli attributi sono elementi che vengono inseriti nel tag di apertura per meglio definire il comportamento dei tag e sono inseriti con la seguente formato:

```
attributo="valore"
```

Ogni tag ammette uno specifico insieme di attributi, ad esempio il tag `<a>` ammette l'attributo `href`:

```
<a href="http://www..."> Io sono un link </a>
```

Note generali

Tutti i tag occupano, se non specificato diversamente dagli **attributi** (*width* e *height*) o modificatori **css**, il 100% della dimensione (in larghezza) del suo contenitore e lo spazio necessario a visualizzare il loro contenuto in altezza.

Tutti i tag “rompono” la riga, ovvero, vengono sempre posti uno sotto l’altro ... tranne alcune eccezioni.

Paragrafo

`<p>Questo è un paragrafo</p>`

Si utilizza il tag `
` per andare a capo.

Titoli

`<h1>Questo è il titolo principale</h1>`

`<h2>Questo è un Sottotitolo</h2>`

`<h3>Questo è un Sottotitolo</h3>`

`<h4>Questo è un Sottotitolo</h4>`

`<h5>Questo è un Sottotitolo</h5>`

`<h6>Questo è un Sottotitolo</h6>`

All'interno di una pagina, ci deve essere un solo titolo h1, ma è possibile inserire più titoli h2. È importante che i titoli h2 ... h6 siano ben gestiti gerarchicamente all'interno della pagina.

Ancore

```
<a href=".." target=".." title="..">Questo  
è un link</a>
```

Il tag `<a>` ha 3 attributi:

- `href`: Specifica l'indirizzo a cui rimandare l'utente;
- `target`: Specifica il comportamento per l'apertura;
- `title`: Specifica il titolo dell'ancora.

Contenitori

```
<div>  
  <p> ... </p>  
  <div> ... </div>  
</div>
```

Il tag `<div>` (Content Division) è utilizzato per suddividere la pagina in blocchi o sezioni.

I tag contenitore sono frequentemente caratterizzati da attributi **id** e **class** per identificarne lo scopo e il ruolo.

Contenitori in HTML5

In HTML5 i contenitori `<div>` sono stati sostituiti dai tag `<section>` e da moltissimi altri tag che ne descrivono meglio il ruolo come i tag `<nav>` e `<footer>`

Le liste

```
<ol>  
  <li> ... </li>  
  <li> ... </li>  
</ol>
```

```
<ul>  
  <li> ... </li>  
  <li> ... </li>  
</ul>
```

I tag lista possono essere di 2 tipi:

- ordinate ``
- non ordinate ``

La differenza è che di base, gli elementi di una lista ordinata sono preceduti da un numero progressivo, mentre gli elementi della lista non ordinata, da un bullet (•) chiamato “disc”.

Immagini

```

```

Il tag `` ha 2 attributi principali:

- `src`: Specifica il percorso dell'immagine sul server o sul web
- `alt`: Specifica un titolo da utilizzare al posto dell'immagine (didascalia)

Il tag `` inoltre ammette altri attributi che ne modificano la visualizzazione come la larghezza (`width`) e l'altezza (`height`).

Altri tag

**
**

Il tag `
` "rompe" la riga di testo e manda a capo la visualizzazione.

`<p>` Questa è una riga interrotta dall'andata **`
`**
a capo`</p>`

<hr/>

Il tag `<hr/>` visualizza una riga di separazione.

`<h1>` Titolo `</h1>`

`<p>` Paragrafo del capitolo `</p>`

<hr/>

`<p>` Paragrafo del capitolo preceduto da una riga
di separazione`</p>`

Accorgimenti

`<h1>...</h1>`

Si

`<h1>...</h1>`

No

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

Si

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

No

`<div><p>...</p></div>`

Si

`<p><div>...</div></p>`

No

Web semantico



```
<body>
```

```
<header></header>
```

```
<footer></footer>
```

```
    <aside></aside>
```

```
    <menu></menu>
```

```
    <nav></nav>
```

```
<section></section>
```

```
</body>
```

HTML5 introduce altri tag di blocco e liste allo scopo di rendere il Web “semantico”

`<header>` e `<footer>` per le intestazioni e i piè di pagina.

`<aside>` per le sidebar

`<menu>` e `<nav>` per il menù di navigazione principale e altri menù

`<section>` e `<article>` per le sezioni e gli articoli oppure per blocchi che si ripetono nel documento

Web semantico



`<header>`

`<section>`

`<article>`

`<aside>`

`<footer>`

Web semantico



Lo scopo principale di HTML 5 è quello di rendere il Web “semantico”, cioè dare un significato alla composizione della pagina in modo da rendere meglio identificabile cosa rappresentano le varie parti della pagina.

Questo non è indirizzato all’utente finale (chi legge la pagina) ma più alle macchine (motori di ricerca in primis) che analizzando la pagina sono in grado poi di fornire un miglior servizio all’utilizzatore finale.

HTML 4.x VS HTML5

Nell'utilizzare HTML, cercate sempre di utilizzare i tag più appropriati al “Concetto Generale” di ciò che state costruendo ad esempio, un menù:

```
<div>
  <div><a href="..."> Home      </a></div>
  <div><a href="..."> Contatti  </a></div>
  <div><a href="..."> ...        </a></div>
</div>
```

HTML 4.x VS HTML5

Un menù è un **elenco di link** ...

```
<ul>
  <li><a href="#"> Home      </a></li>
  <li><a href="#"> Contatti  </a></li>
  <li><a href="#"> ...        </a></li>
</ul>
```

HTML 4.x VS HTML5

Nell'utilizzare HTML5, cercate sempre di utilizzare i tag più appropriati al “Concetto Generale” di ciò che state costruendo e dategli poi un Significato

```
<nav>
  <ul>
    <li><a href="#" > Home      </a></li>
    <li><a href="#" > Contatti  </a></li>
    <li><a href="#" > ...        </a></li>
  </ul>
</nav>
```

Tag in linea

`<body>`

``

``

``

`</body>`

Si chiamano tag in linea perchè a differenza di altri non “rompono” la riga.

Tutti i tag inline vanno racchiusi all’interno di altri tag di blocco o liste per definire il comportamento di alcune porzioni di blocchi

Tag in linea

<p>Questa riga contiene una parte in grassetto</p>

Questa riga contiene una parte **in grassetto**

<p>Questa riga contiene una parte in corsivo</p>

Questa riga contiene una parte *in corsivo*

<p>Questa riga contiene una parte speciale</p>

Questa riga contiene una parte **speciale**

Attributi dei Tag

Tutti i Tag, in genere, ammettono l'uso di alcuni attributi di qualificazione globali detti anche Core Attributes (**coreattrs**) fra cui: `id`, `class`, `style`.

Attributo	Nome	Specifiche
<code>id</code>	Identificatore	Specifica l'identificatore di un elemento (deve essere unico nel documento)
<code>class</code>	Classe	Specifica la classe (o insieme) di appartenenza
<code>style</code>	Stile	Permette di specificare lo stile dell'elemento "in linea"

Attributi dei Tag

Alcuni esempi

```
<h1 id="titolo"> Io sono il titolo </h1>
```

```
<li class="elemento"> Io sono un punto di un elenco </li>
```

```
<a href="..." style="color:red;"> Io sono un link rosso </a>
```

HTML

Potete trovare maggiori informazioni sui tag HTML, le loro funzioni e le loro opzioni su:

<http://www.w3schools.com/html/default.asp>

Tag <head>

<html>

<head>

<title> il titolo del documento

<link> documenti da includere

<script> file di script (javascript)

<style> codice css

<meta> informazioni e metadati

</head>

Maggiori informazioni su:

...

http://www.w3schools.com/html/html_head.asp

</html>

Tag <meta>

```
<html>
  <head>
    <meta name="" content="" />
    ...
  </head>
  ...
</html>
```

Il tag <meta> richiede la coppia di attributi:

- name
- content

Oppure altri attributi come:

- charset
- author

e altri ...

Tag <meta>

```
<meta charset="UTF-8">  
<meta name="description" content="Corso di Tecnologie Web">  
<meta name="keywords" content="HTML,CSS,XML,JavaScript">  
<meta name="author" content="Giacomo Piva">
```

I tag <meta> nell'head di una pagina possono essere più di uno e sono particolarmente importanti perché vengono letti e utilizzati dai motori di ricerca.

Tag <meta>

```
<meta name="description" content="Facebook è una ... ">
```

Ti diamo il benvenuto su Facebook: accedi, iscriviti o scopri maggiori ...

<https://it-it.facebook.com/> ▼

Facebook è una piattaforma sociale che ti consente di connetterti con i tuoi amici e con chiunque lavori, studi e viva vicino a te. Puoi usare **Facebook**...

Esempio 1

<html>

Tag <form>

Il tag <form> viene utilizzato per costruire un modulo di raccolta dati che vengono inviate al server per poi essere elaborate, ad esempio per essere salvate su un database.

Attraverso i form, il browser:

- Raccoglie i dati dall'utente
- Crea una connessione HTTP con il server all'URL specificato nell'attributo `action` e invia i dati al server

Tag <form>

```
<form action="form.php" method="POST" >
```

Il tag <form> possiede 2 attributi principali ed indispensabili:

- `action`: l'indirizzo della pagina a cui inviare i dati
- `method`: Il metodo (o verbo) HTTP da utilizzare

Tag <form>

Il tag <form> racchiude un gruppo di altri tag che servono a raccogliere le informazioni:

```
<input>  
<textarea>  
<select> + <option>  
<button>
```

HTML5 inoltre introduce nuovi tag:

```
<datalist> + <option>  
<fieldset>  
<label>
```


Tag <input>

```
<input type="..." />
```

- text
- password
- radio
- checkbox
- submit

HTML5 inoltre introduce nuovi input:

- date
- color
- email
- number

Costruzione di un Form

```
<form action="form.php" method="POST">
  Nome:<br/>
  <input type="text" name="nome">
  <br/>
  Cognome:<br/>
  <input type="text" name="cognome">
  <br>
  <input type="submit" value="Invia">
</form>
```

Il form viene quindi costruito racchiudendo tutti i campi `<input>` nel tag `<form>` avendo cura di aggiungere:

- gli attributi `action` e `method` al tag `<form>`
- di includere un tag `<input>` di tipo `submit`

Ricevere i dati di un form

Nel nostro esempio il nostro form invia i dati raccolti alla pagina form.php.

```
<form action="form.php" method="POST" >
```

Ma come vengono poi raccolti i dati all'interno della nostra pagina? Il linguaggio PHP mette a disposizione 2 Array che contengono i dati che sono stati passati dalla chiamata HTTP:

```
$_GET[] e $_POST[]
```

I dati saranno disponibili all'interno di questo array alla chiave (posizione) che porta il nome dell'input:

```
$name = $_POST['nome'];  
$surname = $_POST['cognome'];
```

... ma lo vedremo nelle lezioni dedicate a PHP

Tag <form>

Le informazioni riguardanti i form sono davvero tantissime, in questo corso preferisco affrontarle in maniera più pratica durante la realizzazione del progetto.

Potete comunque trovare tutte le informazioni aggiornate sulla pagina

http://www.w3schools.com/html/html_forms.asp

Esempio 2

<form>

Tag <table>

L'elemento probabilmente più utilizzato per visualizzare i dati, è il tag <table> che definisce una tabella.

I tag <tr> e <td> definiscono rispettivamente le righe e le colonne della tabella.

Le <table> sono costruite per righe:

```
<tr>  
  <td>Colonna 1</td>  
  <td>Colonna 2</td>  
  <td>Colonna 3</td>  
</tr>
```

Tag <thead>

Il tag <thead> definisce la testata di una tabella. Può contenere più righe di intestazione che si definiscono con il solito <tr> ma per le colonne dell'intestazione è bene usare il tag <th>

```
<table>
  <thead>
    <tr>
      <th>Titolo Colonna 1</th>
      <th>Titolo Colonna 2</th>
      <th>Titolo Colonna 3</th>
    </tr>
  </thead>
</table>
```

Tag <tbody>

Il tag <tbody> definisce il corpo della tabella. Può contenere più righe che si definiscono con i tag <tr>.

Ogni riga a sua volta può contenere più colonne definite con il tag <td>.

```
<table>
  <thead> ... </thead>
  <tbody>
    <tr>
      <td>Colonna 1</td>
      <td>Colonna 2</td>
      <td>Colonna 3</td>
    </tr>
  </tbody>
</table>
```


Tag <tfoot>

Il tag <tfoot> definisce il footer della tabella. Può contenere più righe che si definiscono con i tag <tr> che a sua volta può contenere più colonne definite con il tag <td>.

```
<table>
  <thead> ... </thead>
  <tbody> ... </tbody>
  <tfoot>
    <tr>
      <td>Colonna 1</td>
      <td>Colonna 2</td>
      <td>Colonna 3</td>
    </tr>
  </tfoot>
</table>
```

Esempio 3

<table>

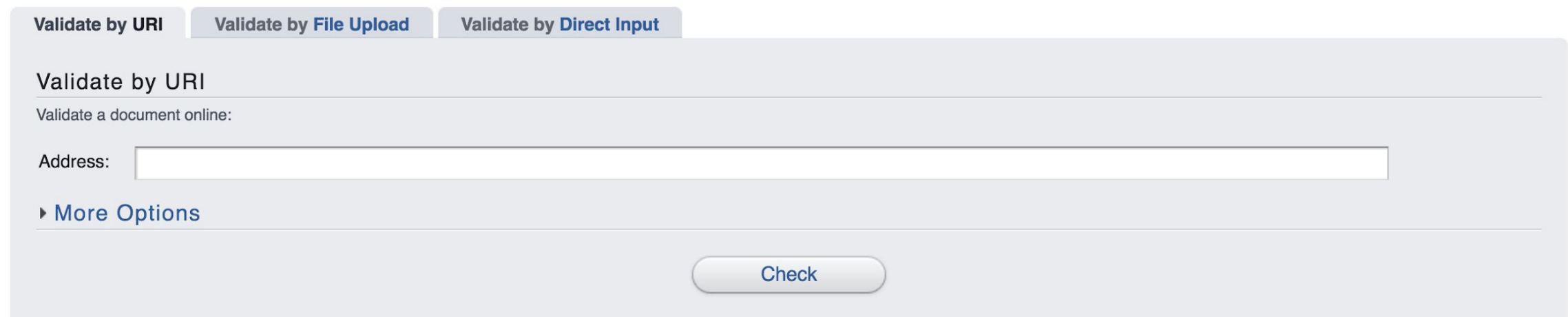
Correttezza dell'HTML

L'autorità che definisce lo standard HTML e quindi “certifica” la correttezza dell'HTML prodotto dai vari sviluppatori è il W3C (<http://www.w3.org>)

Il W3C mette a disposizione uno strumento per verificare la validità del vostro codice.

Questo strumento è reperibile all'indirizzo:
<http://validator.w3.org/>

Correttezza dell'HTML



The image shows the W3C HTML Validator interface. At the top, there are three tabs: 'Validate by URI' (selected), 'Validate by File Upload', and 'Validate by Direct Input'. Below the tabs, the 'Validate by URI' section is active. It contains the text 'Validate a document online:' followed by a label 'Address:' and a text input field. Below the input field is a link 'More Options'. At the bottom right of the section is a 'Check' button.

È possibile inviare il codice per ottenerne la validazione attraverso tre modalità:

- Fornendo un indirizzo raggiungibile dal web
- Caricando un file HTML
- Copiando/Incollando il codice

Fogli di stile

CSS (**C**ascading **S**tyle **S**heet) è un linguaggio che definisce lo stile di un documento HTML.

Attraverso i fogli di stile è possibile definire l'aspetto dei vari elementi del codice HTML ad esempio:

- Colore, Font, interlinea di un titolo o di un paragrafo
- Altezza, larghezza, bordo di un contenitore o di una sezione
- Allineamento, bulleting, spaziatura di un elenco puntato

Maggiori informazioni su:

<http://www.w3schools.com/css/>

Fogli di stile

Il codice CSS può essere inserito all'interno del documento HTML in 3 modi:

- Nel Tag `<style>` nell'`<head>` del documento
- In linea come attributo dei Tag
- Organizzato in file css ed inclusi attraverso il tag `<link>`

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

Regole CSS

Le regole CSS hanno la seguente forma:

h1 { **font-size**: **25px**; }

Selettore **Proprietà** **Valore**



Selettori CSS

Le regole CSS vengono definite con dei selettori:

- Nome tag: h1...h6, p, a, form, li, ol, ul ...
- Classi: .nome-classe
- Identificatori: #nome-id

Esempio 4

CSS

Integrazione

È possibile raggruppare il codice CSS in un file e poi includerlo come risorsa esterna all'interno della pagina HTML attraverso il tag `<link>`

Esempio 5

file .css

Cascading

Il codice CSS potrebbe avere “conflitti” nella sua stesura.

Il significato di “Cascading” si può tradurre in: la regola più specifica viene scelta e applicata.

- La regola più specifica “vince”
- A parità di specificità vince chi sta più in alto nella Gerarchia

Gerarchia

Meno
importante

Più
importante



- 1 User Agent defaults
- 2 Fogli esterni inclusi
- 3 Documento
- 4 Inline

Esempio 6

Gerarchia

Gerarchia

Esiste il modificatore **!important** che consente di “scavalcare” la gerarchia di un gradino:

```
p.p2 { color: red !important; }
```

Altri esempi

Fate attenzione agli spazi!

`section .importante`

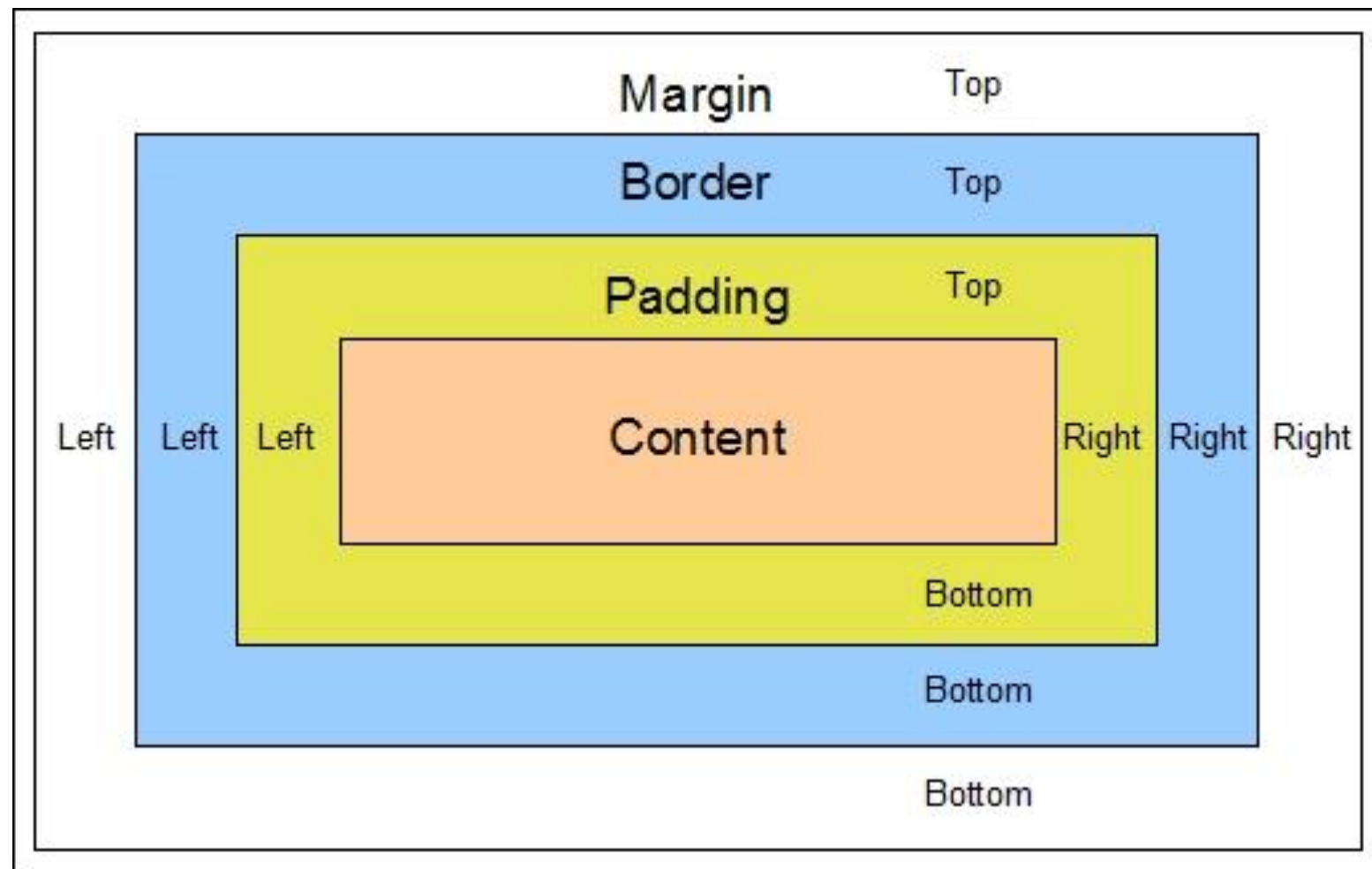
è diverso da

`section .importante`

Esempio 7

Selettori

Margini e Padding



Esempio 8

Margin e Padding

Altri selettori

Esistono anche dei selettori speciali per identificare lo stato di **alcuni** tag.

Per identificare lo stato di un tag (per quelli che lo consentono) usiamo i : seguiti dal nome della pseudo-classe:

- a:hover
- a:visited

Esercizio 1

Sperimentate le regole CSS usando il file di esempio che potete scaricare qui ...

Applicate ad ogni elemento qualche regola di stile.

Trovare un completo elenco di regole di stile sul sito:
<http://www.w3schools.com/css/>

Twitter Bootstrap

<http://getbootstrap.com/>

Container

```
<div class="container"> ... </div>
```

```
<div class="container-fluid"> ... </div>
```

Row

```
<div class="container">
```

```
  <div class="row"> ... </div>
```

```
</div>
```


Column

```
<div class="container">
```

```
  <div class="row">
```

```
    <div class="col-xx-y"> ... </div>
```

```
  </div>
```

```
</div>
```

Column

```
<div class="col-xx-y"> ... </div>
```

xx è la dimensione del breakpoint
y è il numero di colonne (in dodicesimi)

| Breakpoint | Pixels | Dispositivo |
|------------|------------------|-------------|
| xs | < 576px | Smartphone |
| sm | ≥ 576px < 768px | Smartphone |
| md | ≥ 768px ≤ 991px | Tablet |
| lg | ≥ 992px ≤ 1199px | Tablet |
| xl | ≥ 1200px | Desktop |
| xxl | ≥ 1400px | Desktop |

Column

```
<div class="row">  
  <div class="col-md-6"> ... </div>  
  <div class="col-md-6"> ... </div>  
</div>
```

Le due colonne occupano entrambe i 6/12 della riga (cioè metà) e resteranno allineate sulla stessa riga in schermi più grandi di 768px

Quando le dimensioni dello schermo si riducono a 767px le due colonne si disporranno una sotto l'altra.

Esempi 1, 2 e 3

Bootstrap Grid

hidden-* visible-*

Le classi `visible-*` e `hidden-*` nascondono o mostrano l'elemento al quale sono applicate nelle varie dimensioni degli schermi.

Es:

```
<div class="hidden-sm">...</div>
```

Risulterà non visibile quando lo schermo sarà più piccolo di 768px

Esempio 4

Bootstrap Grid

Consigli finali

Prendetevi il tempo di guardare la documentazione di Bootstrap per scoprire più funzioni possibile.

Usate il più possibile le classi pre-configurate di Bootstrap per realizzare un certo comportamento.

Date particolare importanza alle tabelle:

<https://getbootstrap.com/docs/4.1/content/tables/>

Esercizio 1

Partendo dall'esercizio 3 della lezione 5, aggiungete al file PHP il foglio di stile di Bootstrap e applicate le classi alla tabella.

Esercizio 2

Costruite una pagina categoria per un negozio online.

La pagina dovrà contenere un elenco di articoli con:

- Nome
- Immagine (usate <http://via.placeholder.com/200x200.jpg> come src)
- Prezzo

Le informazioni dovranno essere memorizzate in un array associativo di almeno 12 elementi

La pagina dovrà mostrare gli elementi in una griglia di 4 articoli per riga.

Domande?