



Università di Cagliari  
Dipartimento di Matematica e Informatica

---

# HTML

Lucio Davide Spano

[davide.spano@unica.it](mailto:davide.spano@unica.it)

# Cosa è l'HTML



- Hyper Text Markup Language
- Un linguaggio di markup descrive la struttura di un documento utilizzando un insieme di tag
  - Si separa il documento in sé
  - Dal codice per formattarlo
- Questo mix di codice e testo forma una pagina web
- Il browser interpreta i tag “disegnando” il documento
- N.B. Ogni browser lo fa un po' a modo suo
- Importante testare su diversi browser





# Cosa definiamo con l'HTML

- Ipertesti (testi collegati fra loro)
- Il web è un grafo
  - 14.5 miliardi di nodi
  - 10 link mediamente per pagina
- L'HTML viene usato per definire la **struttura di un documento**
- **Non** deve essere utilizzato per la **presentazione (layout)**
- Es. Questa slide ha questa struttura
  - Un titolo
  - Un elenco puntato per contenuto
- La presentazione definisce
  - Il colore rosso per il titolo
  - Il nero per il testo
  - Il bianco per lo sfondo

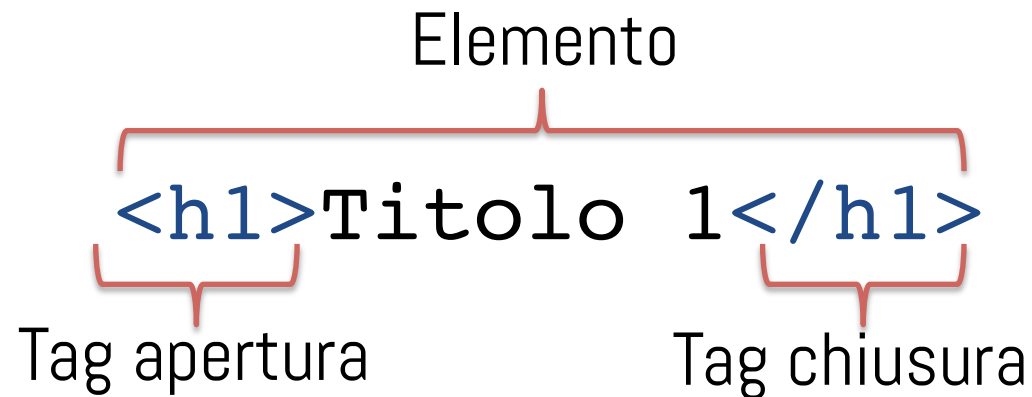
# HTML tags



- Sono delle parole chiave utilizzate per marcare parti del testo
- Sono contenute fra parentesi angolari
- Di solito accoppiati
  - Un tag di apertura
  - Un tag di chiusura
- Quello di chiusura ha uno slash dopo la prima parentesi angolare
- Esempio

```
<h1>Titolo 1</h1>
```

- **Tag** ed **Elemento** erroneamente usati come sinonimi
- Un tag è ciò che è compreso fra parentesi angolari
- Un elemento è tutto ciò che è contenuto fra il tag di apertura e quello di chiusura (anche nulla)
- Possono essere nidificati (il contenuto di un elemento può essere formato da uno o più elementi)
- Possono essere chiusi direttamente nel tag di apertura  
es. `<br/>`
- Il nome del tag è meglio scriverlo in minuscolo



# Attributo



- Gli elementi HTML possono avere degli attributi
- Forniscono **informazioni aggizionali** su un elemento
- Sono sempre specificati nel tag di apertura
- Sono formati da **coppie nome-valore**
- I valori vanno sempre messi tra virgolette
- È meglio scriverli in minuscolo

Attributo

<a href="http://www.w3schools.com">Esempio di link</a>

Nome                      Valore

# Attributi comuni



Attributo	Descrizione
class	Uno o più nomi di classi associate ad un elemento (ci torneremo con i CSS)
id	Un identificatore unico per l'elemento
style	Uno stile CSS specificato come attributo (ci torneremo con i CSS)
title	Informazione supplementare su un elemento (visualizzata come tooltip)

# Un po' di storia



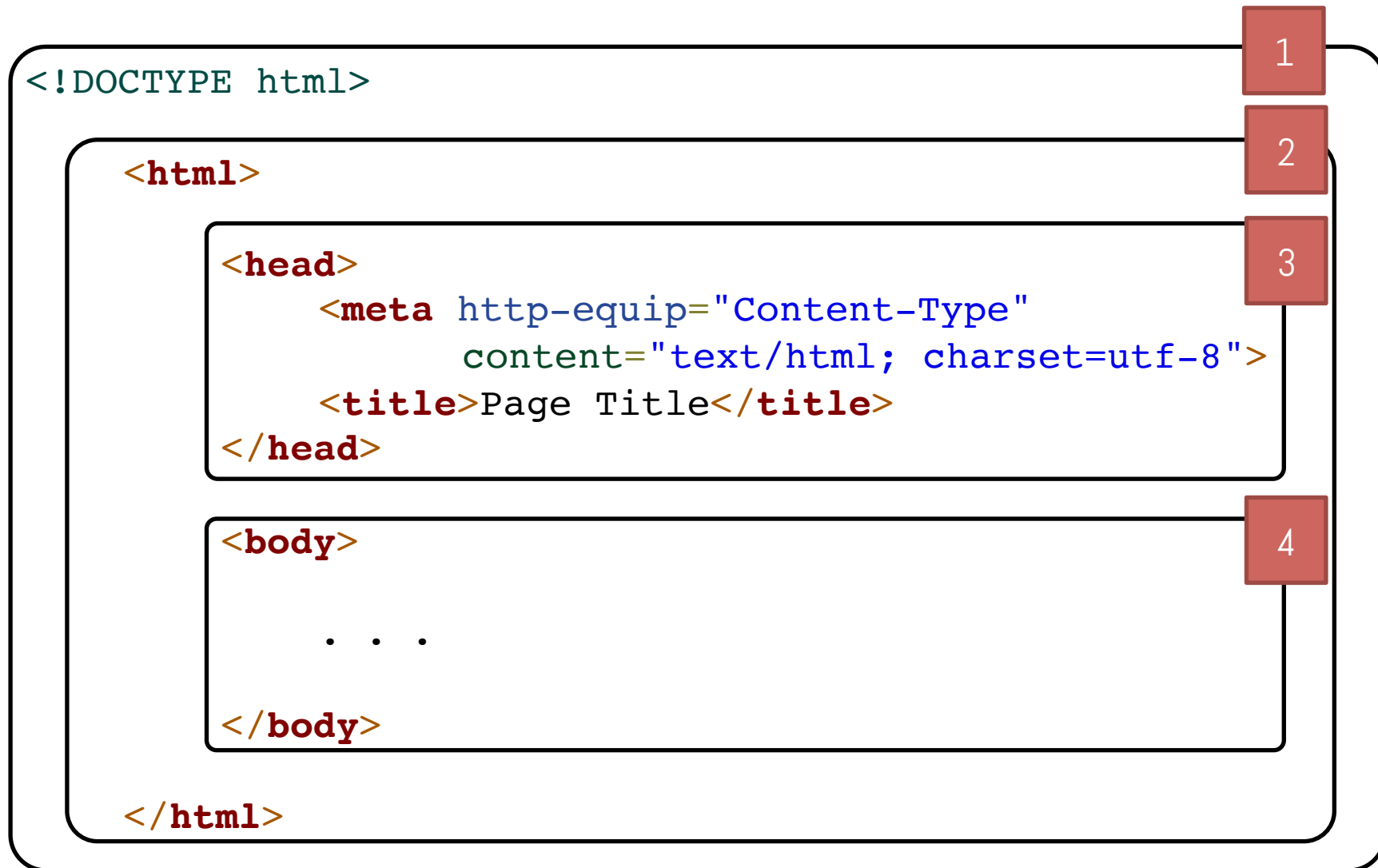
- All'inizio non aveva una definizione rigorosa
- Primo standard HTML 2.0 (1993)
- Oggi è mantenuto dal W3C
- Versioni degne di nota
  - HTML 4.0 Transitional e Strict
    - Con e senza tag legati alla presentazione
  - XHTML 1.0 Transitional e Strict
    - Riscrittura dell'HTML con schema XML
  - HTML 5.0
    - Multimedia
    - Migliore semantica delle parti del documento
    - Migliore supporto per lo sviluppo di applicazioni web
- Noi utilizzeremo HTML5
- Ma per questioni di tempo non sfrutteremo tutte le potenzialità



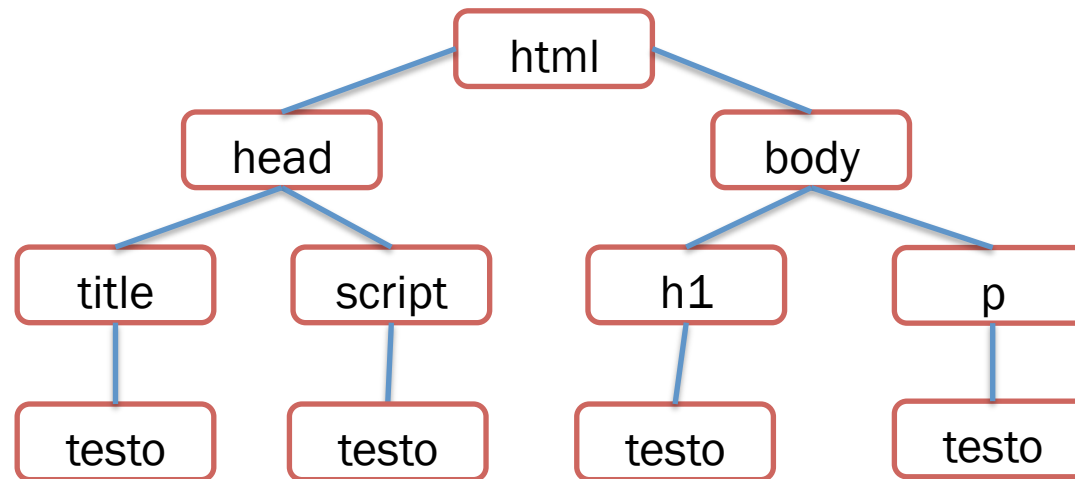
- Nuova versione dello standard
- È diventata una Recommendation W3C il 28 Ottobre 2014
- Introduce molti miglioramenti all'HTML
  - Che risultava un po' datato
  - Eliminando la necessità di ricorrere a javascript per avere dei controlli «decenti»



# Struttura di una pagina HTML



# L'albero del documento HTML



- La struttura nidificata degli elementi crea un albero
- È importante per diverse funzionalità
- Applicazione di stili
- Comportamento della pagina con javascript

# Il doctype



- Definisce quale versione dell'HTML vogliamo usare
- La correttezza sintattica è legata al doctype
- Anche se il browser visualizza la pagina non è detto che sia corretta

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01

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

XHTML 1.0

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

- Servono... a chi sviluppa
- Per capire quanto sono importanti, provate a riprendere una pagina complicata dopo un anno...

```
<!-- un commento in HTML -->
```

# Head



- Il tag head contiene elementi con informazioni generali sulla pagina
- Il titolo della pagina
- Dove trovare fogli di stile (CSS)
- Includere script
- Fornire meta-informazioni

```
<head>
  <title>Title</title>
  <meta charset="utf-8"/>
  <style type="text/css"></style>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  <script type="text/javascript"></script>
</head>
```

# Title



- Definisce il titolo del documento
- È necessario in tutti i documenti HTML
- Definisce il titolo della finestra del browser
- Fornisce il titolo quando si aggiunge una pagina ai preferiti
- Fornisce il titolo per i risultati sui motori di ricerca

```
<head>  
    <title>La mia pagina</title>  
    ...  
</head>
```

# Uniform Resource Locator



`<protocollo>://<nomehost>:<porta><percorso><querystring>#<fragment>`

- Il *protocollo* applicativo da utilizzare per reperire la risorsa descritta (HTTP, HTTPS, FTP, MMS)
- Il *nome host* è costituito o da un nome di dominio o direttamente da un indirizzo IP
- La *porta* viene specificata quando il protocollo non utilizza quella standard (opzionale).
- Il *percorso* è il path locale della risorsa richiesta all'interno del server
- La *querystring* è una serie di coppie nome-valore con il formato  
`<nome1>=<valore1>&... &<nomeN>=<valoreN>`  
che sono usati dal server come parametri
- Il *fragment* viene utilizzato per specificare una posizione particolare all'interno della risorsa

<http://spano.sc.unica.it:80/amm.php?anno=2013&alunno=spano#titolo1>





# Collegamenti assoluti e relativi

- Un collegamento assoluto è formato da una URL che inizia con l'identificatore del protocollo es.  
`http://www.w3schools.com/images/img1.png`
- Un collegamento relativo fa riferimento invece alla cartella (locale o remota) che contiene la risorsa corrente
  - Risorsa corrente: `/main/index.html`
  - Il collegamento relativo `images/img1.png` riferisce la risorsa  
`http://www.w3schools.com/main/images/img1.png`
  - La URL base: `/main`
  - Se invece scriviamo `/images/img1.png` la URL è sempre relativa, ma parte dal dominio (che è la parte della URL di base immediatamente dopo il protocollo e prima dello /), qui
  - La cartella corrente si identifica con un punto  
Es. `./images/img1.png` equivale alla URL relativa  
`images/img1.png`
  - La cartella superiore si identifica con un doppio punto `../`

- L'elemento `base` permette di ridefinire la URL di base per tutti i collegamenti relativi nella pagina

```
<head>  
    <base href="http://www.w3schools.com/images/">  
</head>
```



- Definisce un relazione fra il documento corrente ed una risorsa esterna
- Di solito è un file CSS (li vedremo in seguito)

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

- Definisce le informazioni riguardo alla presentazione del documento
- Si usa la sintassi dei CSS
- Sono preferibili file esterni

```
<style type="text/css">
  body {
    background-color: yellow;
  }

  p {
    color: blue;
  }
</style>
```

- Contiene informazioni sul documento HTML
- Descrizione, keywords, autore, ultima modifica ecc.
- Non sono visualizzati, ma sono utilizzati da programmi
- Sono utilizzati dai browser, motori di ricerca e altri servizi

Keywords

```
<meta name="keywords" content="HTML, CSS, PHP, MySQL, AMM">
```

Descrizione

```
<meta name="description" content="Slides del corso di AMM">
```

Autore

```
<meta name="author" content="Davide Spano">
```

Ricaricare la pagina ogni 30 secondi

```
<meta http-equiv="refresh" content="30">
```

- È il tag che contiene la definizione della struttura del documento
- In poche parole, contiene quello che si vede della pagina

```
<body>
```

```
    <h1>Un titolo</h1>
```

```
    <p>Un paragrafo molto interessante</p>
```

```
</body>
```



# Caratteri speciali e accentati

- I caratteri accentati vengono visualizzati in modo errato se utilizzati con un browser che usa un charset (modo di codificare i caratteri) diverso.
- Esistono due possibilità
  1. Specificare il charset esplicitamente con un tag meta (nelle specifiche HTML5 è il default)

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

2. Utilizzare le HTML entities, dei codici speciali per indicare delle lettere non ASCII

Es. a con accento grave (à) `&agrave;`  
a con accento acuto (á) `&acute;`



# Headings (Titoli)

- Definiscono i titoli nei documenti HTML
- Elementi `<h1>` ... `<h6>`
  - 1 è il più importante
  - 6 è il meno importante
- È importante usare questi elementi solo per i titoli
  - Non per fare il testo più grande
  - Gli utenti leggono la pagina velocemente
  - Anche i motori di ricerca li usano per capire la struttura dei documenti

`<body>`

`<h1>Capitolo</h1>`

`<h2>Sezione</h2>`

`<h3>Sotto-sezione</h3>`

`</body>`

**Capitolo**

**Sezione**

**Sotto-sezione**



# Paragrafi



- Elemento `<p>`
- I browser aggiungono una linea vuota prima e dopo il testo contenuto nel tag
- Non contate sugli spazi che mettete nel testo. Il browser non li considera
- Potete usare il tag `<br />` per andare a capo
- Potete usare il tag `<hr />` per disegnare una linea orizzontale

```
<body>
  <p>Il mio primo paragrafo</p>
  <p>Un paragrafo      con spazi e
accapo</p>
<p>
Un paragrafo con il tag <br />
      per andare a capo
</p>
<p>Una riga <hr /> orizzontale</p>
</body>
```

Il mio primo paragrafo

Un paragrafo con spazi e accapo

Un paragrafo con il tag  
per andare a capo

Una riga

---

orizzontale



# Formatting tags

- Servono per formattare il testo
  - Grassetto
  - *Corsivo*
  - Sottolineato
  - ...
- Alcuni sono di solito renderizzati allo stesso modo
  - `<strong>` e `<b>` di solito visualizzano il testo in grassetto
  - Ma `<strong>` è un tag semantico, indica un testo più importante
  - `<b>` invece indica semplicemente il grassetto
- Ricordate che i tag semantici valgono anche per modalità non visuali
  - Ad esempio chi legge la pagina con uno screen reader (audio) sentirà pronunciare il testo in modo diverso



## Formatting tags (2)

- Lista completa
- [http://www.w3schools.com/html/html\\_formatting.asp](http://www.w3schools.com/html/html_formatting.asp)

<body>

<p><b>Testo in grassetto</b></p>

<p><strong>Testo importante</strong></p>

<p><i>Testo corsivo</i></p>

<p><em>Testo con enfasi</em></p>

<p><code>Codice di programma</code></p>

<p>Testo<sub> pedice</sub> e

<sup>apice</sup></p>

</body>

**Testo in grassetto**

**Testo importante**

*Testo corsivo*

*Testo con enfasi*

Codice di programma

Testo<sub>pedice</sub> e <sup>apice</sup>

- Un link è costituito una o più parole o immagini che possono essere cliccate per cambiare documento
- Tag `<a>`
- L'attributo `href` indica verso quale documento porta il collegamento
- L'attributo `target`, con valore `_blank` apre il documento in una nuova finestra
- È possibile definire collegamenti all'interno dello stesso documento
  - Si definisce un segnalibro, marcando una parte del documento con un tag `<a>` e dando un valore all'attributo `id`
  - Il collegamento si effettua definendo un altro tag `<a>` e dando all'attributo `href` il valore dell'attributo `id` del segnalibro preceduto dal carattere `#`

## Link (2)



[Link esterno](#)  
[Link esterno in nuova finestra](#)

# Capitolo 1

[Collegamento al capitolo 1](#)

```
<a href="http://people.unica.it/davidespano/">
    Link esterno
</a>
<br />
<a href="http://people.unica.it/davidespano/" target="_blank">
    Link esterno in nuova finestra
</a>
<a id="segnalibro">
    <h1>Capitolo 1</h1>
</a>
<a href="#segnalibro">Collegamento al capitolo
</a>
```

# Immagini



- Sono definite dal tag `<img>`
  - È un tag vuoto (non si può mettere un altro elemento all'interno)
- L'immagine è una risorsa **esterna** al documento
  - È caricata da un'altra richiesta HTTP
  - La sua URL è specificata nell'attributo `src`
- Se non fosse disponibile viene visualizzato il testo alternativo
  - Specificato nell'attributo `alt`, obbligatorio
  - È necessario anche per i non vedenti (screen reader)
- È buona pratica specificare altezza e larghezza
  - `height` e `width`
  - Utilizzate dal browser per calcolare lo spazio prima di caricarla



```

```

- Sono definite dal tag `<table>`
  - `<th>` per le intestazioni
  - `<tr>` per le normali righe
  - `<td>` per le celle
- Vedremo come rifare il look con in CSS

```
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

- Due tipi
  - Ordinate `<ol>`
  - Non-ordinate `<ul>`
- Gli elementi all'interno si taggano con `<li>`
- Anche loro hanno stili CSS dedicati

```
<h2>Lista ordinata</h2>
<ol>
  <li>Primo valore</li>
  <li>Secondo valore</li>
  <li>Terzo valore</li>
</ol>
```

```
<h2>Lista non ordinata</h2>
<ul>
  <li>Primo valore</li>
  <li>Secondo valore</li>
  <li>Terzo valore</li>
</ul>
```

## Lista ordinata

1. Primo valore
2. Secondo valore
3. Terzo valore

## Lista non ordinata

- Primo valore
- Secondo valore
- Terzo valore



- Contiene degli elementi che consentono l'input da parte dell'utente
  - Text fields, check-box, drop-down lists, pulsanti ecc.
- Sono utilizzati per inviare dati ai server
  - Con una get o una post
  - La url è specificata tramite l'attributo *action*
  - Il metodo http è specificato dall'attributo *method* (get o post)
  - Ne ripareremo per il lato server

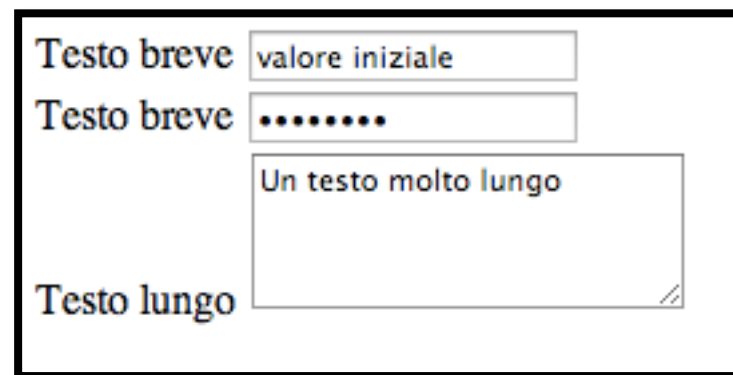
```
<form action="pagina.php" method="post">  
  <!--  
    Inserire qui gli elementi  
    di input  
  -->  
</form>
```

- Insieme di elementi utilizzati per ricevere un valore dall'utente
- Sono solitamente contenuti negli elementi `form`
- Ne esistono vari tipi
  - Differenziati dal valore dell'attributo `type` del tag `<input>`
  - Altri elementi come `<select>` e `<textarea>`
  - Si devono scegliere con criterio a seconda del tipo di input
- L'input viene fornito come coppia chiave-valore
  - La chiave è specificata dall'attributo `name`
  - Il valore è specificato dall'attributo `value`
    - Viene specificato nell'html il valore `iniziale`
    - Alcuni non possono essere modificati
- È possibile specificare una etichetta testuale tramite l'elemento `<label>`
  - Specificare nell'attributo `for` l'`id` dell'elemento `input`

# Input testuale

- Buono per testo e numeri “liberi”
  - `<input type="text">` per valori brevi
  - `<textarea>` per valori più lunghi
  - `<input type="password">` per le password
- HTML 5 ne ha introdotto molti altri per tipi diversi tipi di dato
  - Numeri, date, email ecc.
  - Faremo una breve carrellata

```
<form action="demo_form.asp" method="get">
  <label for="corto">Testo breve</label>
  <input type="text" name="corto"
    id="corto" value="valore iniziale" />
  <br />
  <label for="pswd">Testo breve</label>
  <input type="password" name="pswd"
    id="pswd" value="oscurato" />
  <br />
  <label for="lungo">Testo lungo</label>
  <textarea rows="4" cols="20"
    name="lungo" id="lungo">
    Un testo molto lungo
  </textarea>
</form>
```












































Testo breve valore iniziale

Testo breve .....

Testo lungo Un testo molto lungo

## Valori input HTML5

Input type	Descrizione	Supporto
color		    
date	<input type="text" value="dd/mm/yyyy"/>	    
email	E-mail: <input type="text"/>	    
file	<input type="button" value="Choose File"/> No file chosen	    
number	<input type="text"/>	    
range		    
time	<input type="text" value="--:--"/>	    
url	<input type="text"/>	    

- Esempio online  
<http://spano.sc.unica.it/amm2014/davide/html5/inputHTML5.html>
- Più informazioni  
[http://www.w3schools.com/html/html\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html_form_input_types.asp)

- Si possono inserire utilizzando due tag diversi:
  - `input`
  - `button`
- Entrambi ammettono i seguenti valori per l'attributo `type`:
  - `submit` invia il form al server
  - `button` bottone generico da utilizzare con Javascript
  - `reset` cancella gli input dell'utente da tutti i campi della form
- Differiscono essenzialmente nella specifica del contenuto
  - `input` ammette solo testo nell'attributo `value`, e non può contenere elementi figli
  - `button` può contenere gli elementi figli, quindi è possibile mescolare immagini e testo all'interno dell'etichetta
- Input ha anche un valore `image` per l'attributo `type` che consente di usare una sola immagine come etichetta



# Pulsanti: esempi

```
<input type="submit" value="Submit con tag input" />
```

```
<button type="submit">Submit con button</button>
```

```
<button type="submit" >
```


```
    Ok !
```

```
</button>
```

```
<input type="image" src="button_ok.png" />
```

Submit con tag input

Submit con button

 Ok !





# Selezione a scelta singola

- Diversi controlli in base a quante scelte sono disponibili
- Bassa cardinalità ( $7 \pm 2$ )
  - Radio button `<input type="radio">`
- Media cardinalità (tra 10 e 20)
  - Drop down list `<select>`
- Alta cardinalità (più di 20)
  - List box `<select size="n">`
- Non sono regole fisse
  - Ma aiutano a scegliere
- Come vedremo, tecniche che utilizzano ajax offrono una usabilità superiore



## Selezione a scelta singola (2)

```
<h2>Cardinalit&agrave; bassa</h2>
<p>
  <label for="uomo">Uomo</label>
  <input type="radio" name="sesso"
    id="uomo" value="uomo"><br>
  <label for="donna">Donna</label>
  <input type="radio" name="sesso"
    id="donna" value="donna"><br>
</p>
<h2>Cardinalit&agrave; media</h2>
<p>
  <select name="marca">
    <option value="volvo">Fiat</option>
    <option value="saab">Alfa Romeo</option>
    <option value="opel">Opel</option>
    <option value="audi">Audi</option>
  </select>
</p>
<h2>Cardinalit&agrave; alta</h2>
<p>
  <select name="lista" size="3">
    <option value="val1">Valore 1</option>
    <option value="val2">Valore 2</option>
    <option value="val3">Valore 3</option>
    <option value="val4">Valore 4</option>
  </select>
</p>
```

**Cardinalità bassa**

Uomo ☒

Donna ☐

**Cardinalità media**

Fiat

**Cardinalità alta**

Valore 1

Valore 2

Valore 3

Valore 4





# Selezione a scelta multipla

- Diversi controlli in base alle scelte disponibili
- Medio-bassa cardinalità (circa 15)
  - Check box `<input type="checkbox">`
  - Va bene anche per i booleani (uno solo)
- Medio-alta cardinalità (più di 15)
  - Listbox `<select multiple>`
- Sempre regole non teologiche
- Quando la cardinalità è alta, è meglio utilizzare dei metodi più interattivi (e.s. drag & drop)



## Selezione a scelta multipla (2)

```
<h2>Cardinalit&agrave; medio-bassa</h2>
<p>
  <input type="checkbox" name="vehicle" value="Bike">
  Ho una bicicletta
  <br>
  <input type="checkbox" name="vehicle" value="Car">
  Ho un'automobile
  <br>
  <input type="checkbox" name="vehicle" value="Moto">
  Ho una moto
  <br>
</p>

<h2>Cardinalit&agrave; medio-alta</h2>
<p>
  <select name="lista" size="3" multiple>
    <option value="val1">Valore 1</option>
    <option value="val2">Valore 2</option>
    <option value="val3">Valore 3</option>
    <option value="val4">Valore 4</option>
  </select>
</p>
```

**Cardinalità medio-bassa**  
☒ Ho una bicicletta  
☐ Ho un'automobile  
☒ Ho una moto

**Cardinalità medio-alta**  

Valore 1
Valore 2
Valore 3
Valore 4



- Le sezioni nei documenti HTML vengono utilizzate per due motivi principali
- Il primo è separare i contenuti
- Il secondo è permettere di definire layout
- Due elementi:
  - `<div>`
    - Definisce un blocco del documento separato dagli altri
  - `<span>`
    - Definisce una sezione inline (non separata dalle altre)
- Ne parleremo diffusamente introducendo il block model



# Sezioni: esempi

```
<div class="input-form">
  <h3>Filtro</h3>
  <form method="post" action="docente_reg_esami.html">
    <label for="matricola">Matricola</label>
    <input name="matricola" id="matricola" type="text"/>
    <label for="nome">Nome e Cognome</label>
    <input name="nome" id="nome" type="text"/>
    <label for="data">Data</label>
    <input name="data" id="data" type="text"/>
    <input type="submit" value="Filtra"/>
  </form>
</div>
```

## Filtro

Matricola

Nome e Cognome

Data

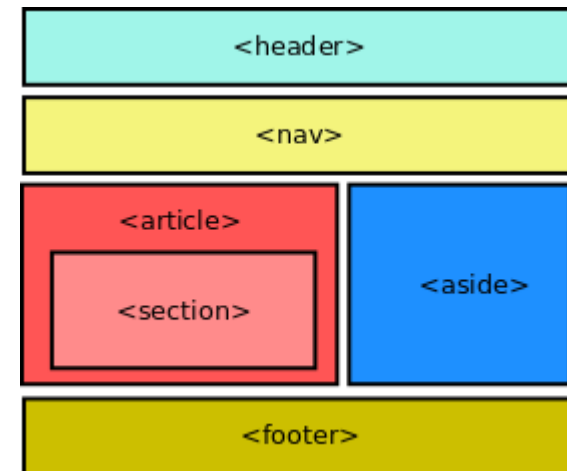
Filtra

- Sezioni specifiche per
  - Parti di testo
    - Articoli
    - Sezioni
    - Note a margine
  - Componenti di una pagina HTML
    - Header
    - Footer
    - Menu di navigazione
- Dal punto di vista visuale non presentano alcuna differenza con una sezione generica
- Ma hanno una semantica
- Se usate correttamente aiutano il browser a comprendere il tipo di contenuto

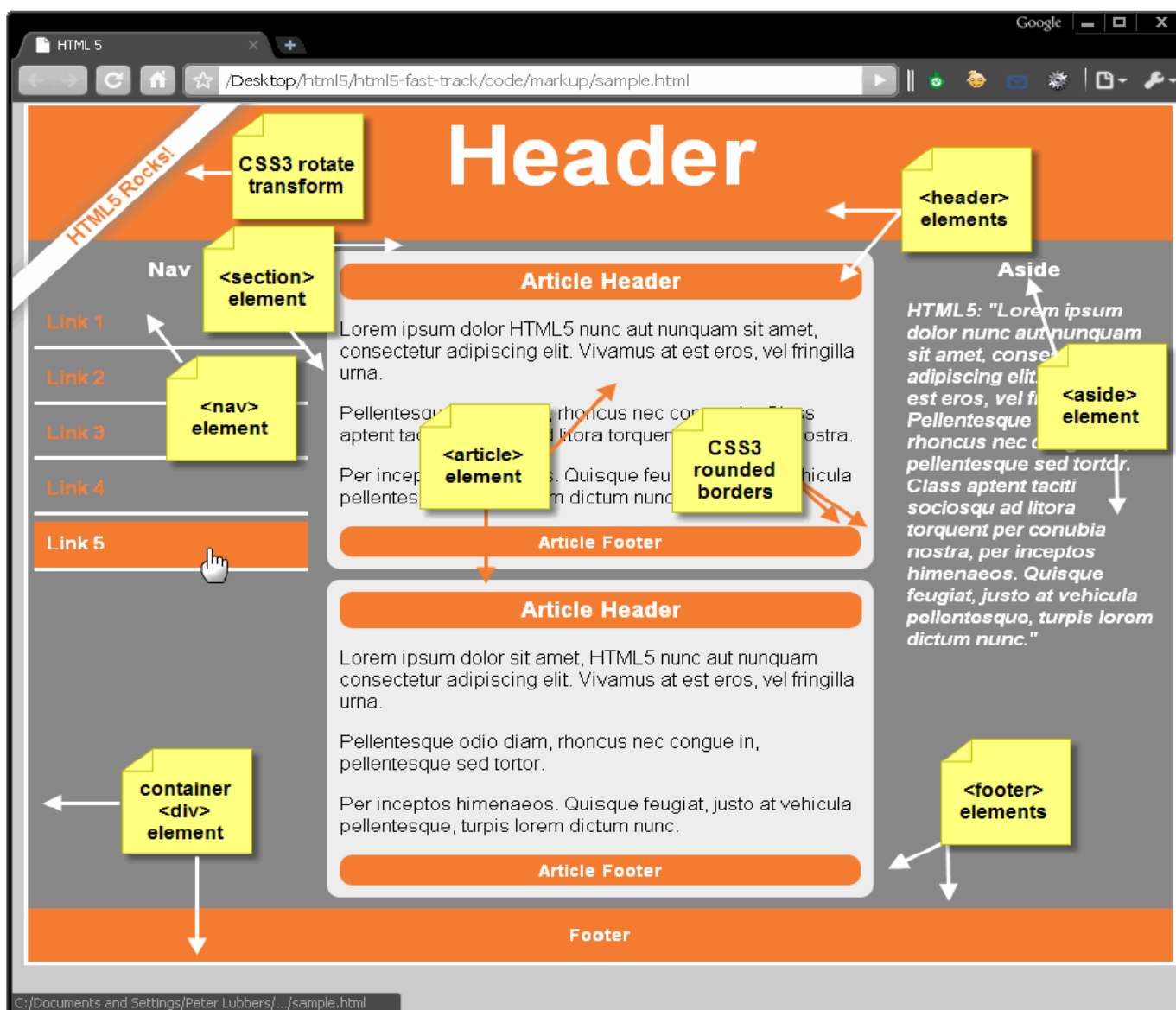
# Sezioni HTML (2)



- **<header>**
  - Sezione che contiene la testata della pagina
  - Sezione che contiene i titoli dell'articolo
- **<nav>**
  - Sezione che contiene i link per navigare il sito
- **<article>**
  - Sezione della pagina che contiene un determinato contenuto, identificabile rispetto agli altri (es. una notizia)
- **<section>**
  - Definisce una sezione all'interno di un articolo
- **<aside>**
  - Sezione di approfondimento (note a margine) o con parti non direttamente collegate al contenuto (sponsor, link esterni ecc.)
- **<footer>**
  - Sezione di chiusura della pagina, con informazioni su autore contatti ecc.
- Ce ne sono altre, ma queste sono le più importanti



# Sezioni HTML 5: Esempio





# Curiosità fuori programma



# Canvas

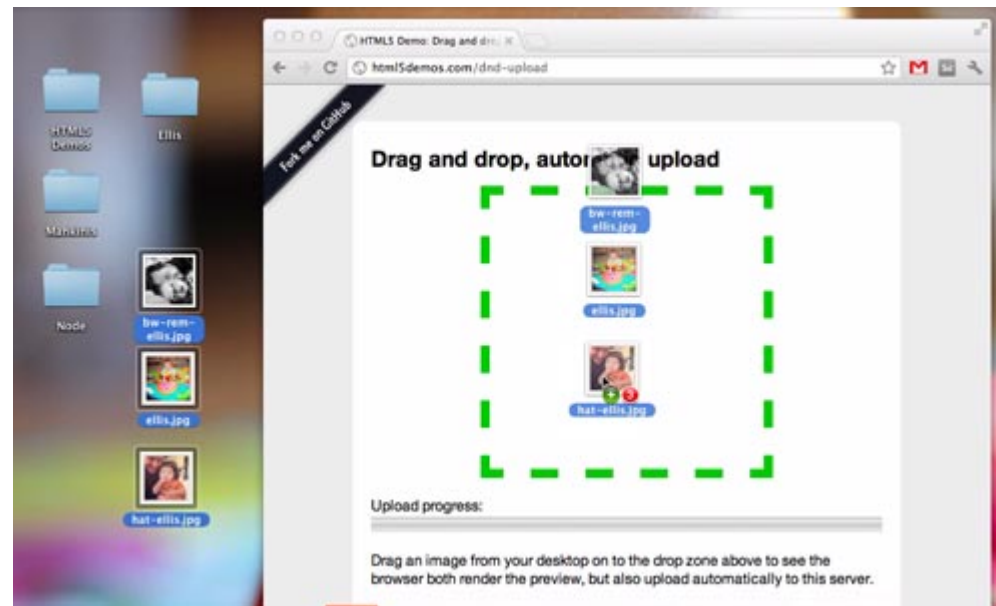


- Permette di definire una sezione dove è possibile disegnare grafica 2D
- Direttamente tramite codice javascript
- Molto utile per applicazioni come i giochi
- Ma anche per la visualizzazione di dati
- Diminuisce la necessità di utilizzare plugin di terze parti (es. Flash)



# Drag & Drop

- HTML5 aggiunge il supporto all'interazione Drag & Drop
- È possibile specificare che alcuni blocchi accettino dati trascinati da altri
- Era possibile anche prima, ma era necessario implementare tutto il supporto da zero



# Geolocation



- Quante volte vi è capitato che una applicazione web mobile vi chiedesse la vostra posizione?
- La nuova versione dell'HTML standardizza l'API



# Video & Audio



- Elementi pensati per includere file video e audio in una pagina web
- Come si fa per le immagini
- Prima si utilizzava un plugin esterno con l'elemento `<object>`





- <http://www.w3schools.com/html/>