

# **HTML5 (Parte 2)**

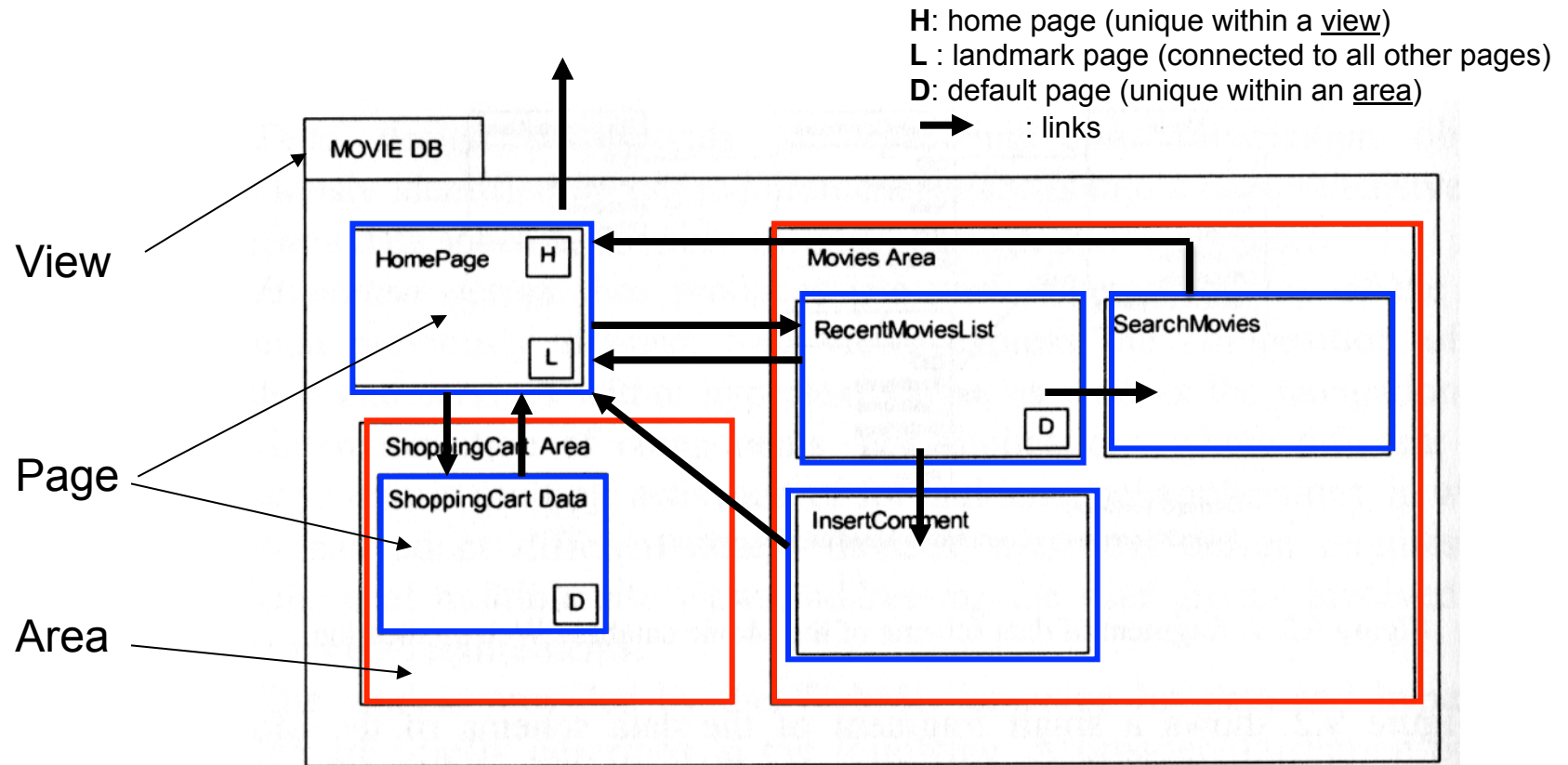
Prof. Elio Toppano – Università di Udine  
Tecnologie Web e Laboratorio (a.a. 2020-2021)

**COLLEGAMENTI IPERTESTUALI**

# Tipi di collegamenti

- Il linguaggio HTML5 permette di descrivere collegamenti ipertestuali:
  - tra pagine appartenenti a siti/portali web diversi
  - tra pagine diverse dello stesso sito web
  - tra parti diverse di una stessa pagina
- Inoltre è possibile specificare se la pagina di destinazione deve essere visualizzata nella stessa finestra del browser della pagina di partenza o in una finestra diversa

# Strutturazione di una *vista* di un sito web



Example of site view modularization based on areas and pages.

**Nota:** un sito è tipicamente formato da una o più viste (view)

# Anatomia di un collegamento ipertestuale

- Un collegamento ipertestuale viene creato mediante l'elemento *a* (anchor)
- La pagina (o la parte di documento) a cui ci si vuole collegare è specificata mediante l'attributo *href* (hypertext reference). Il riferimento può essere **assoluto o relativo**
- Il testo contenuto tra il tag di apertura e quello di chiusura rappresenta *l'àncora* del collegamento (detta anche link test). Cliccando su di essa si richiede la trasmissione della pagina
- Esempi
  - <a href="<http://www.uniud.it>"> Università degli Studi di Udine </a>
  - <a href="contatti.html"> Contatti </a>
  - <a href="musica/index.html"> Articoli musicali </a>

# Riferimenti assoluti e relativi

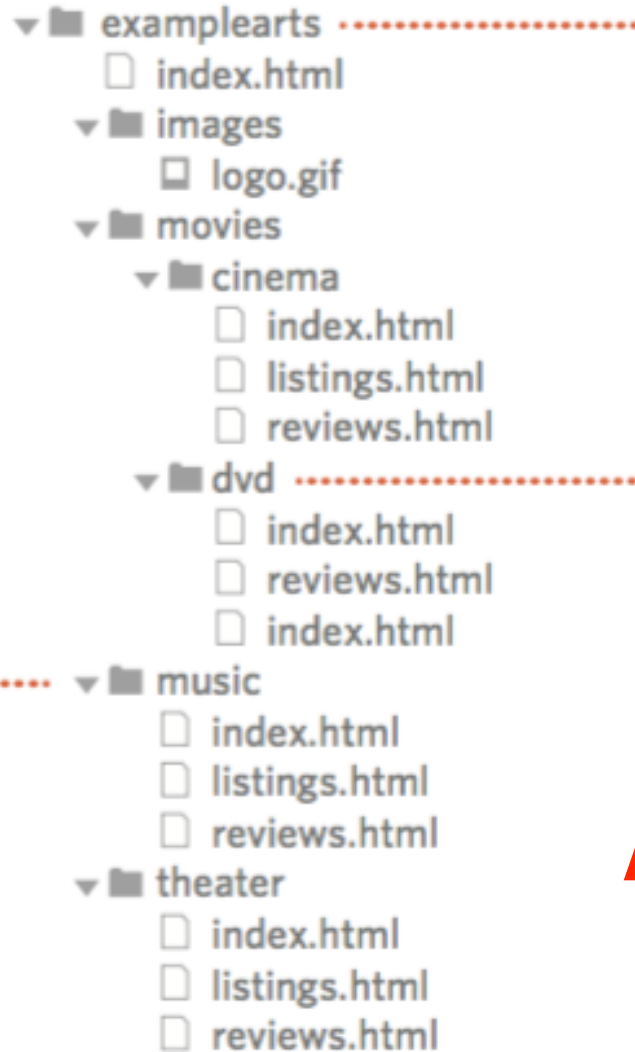
- Un ***riferimento assoluto*** specifica l'URI della risorsa referenziata (o parte di essa) che include: i) il protocollo; ii) il nome del dominio, iii) il percorso (pathname) della risorsa ed eventualmente iv) l'identificatore del frammento di risorsa referenziata (es. #idfragm) . Se non è presente alcun percorso si intende la homepage del sito (es. <http://www.uniud.it>). Viene usato tipicamente per i collegamenti a pagine appartenenti ad altri siti
- Un ***riferimento relativo*** specifica la posizione della risorsa referenziata (o parte di essa) rispetto alla posizione della pagina in cui ci si trova (quella in cui si inserisce il collegamento!). E' una notazione abbreviata rispetto al riferimento assoluto usata per i collegamenti a pagine dello stesso sito.

# Note sui riferimenti relativi

- Se tutte le pagine del sito si trovano nella stessa cartella allora il riferimento è semplicemente il nome del file html che descrive la pagina
- Se le pagine del sito sono distribuite su più cartelle (folder) allora bisogna indicare dove si trova la risorsa rispetto alla pagina corrente. Il percorso di localizzazione della pagina (o di una altra risorsa) dipende da come sono organizzate le cartelle che contengono i diversi files nella cartella principale.
- La organizzazione delle risorse in cartelle a diversi livelli gerarchici permette di rappresentare la struttura delle diverse aree del sito web. I file contrassegnati col nome *index.html* sono le homepage delle diverse aree del sito. Si veda l'esempio seguente

## PARENT

The *examplearts* folder is a parent of the *music* folder.



## CHILD

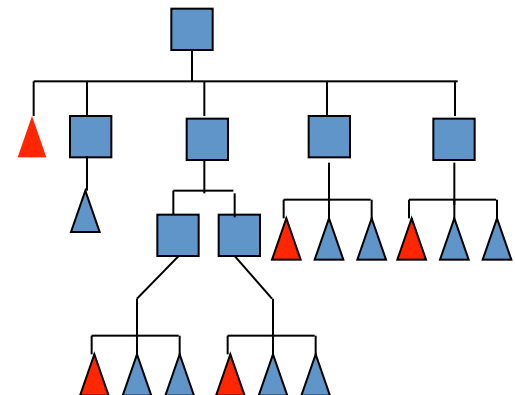
The *music* folder is a child of the *examplearts* folder.

## GRANDPARENT

The *examplearts* folder is a grandparent of the *dvd* folder.

## GRANDCHILD

The *dvd* folder is a grandchild of the *examplearts* folder.





## RELATIVE LINK TYPE

## EXAMPLE (from diagram on previous page)

### SAME FOLDER

To link to a file in the same folder, just use the file name. (Nothing else is needed.)

To link to music reviews from the music homepage:

```
<a href="reviews.html">Reviews</a>
```

### CHILD FOLDER

For a child folder, use the name of the child folder, followed by a forward slash, then the file name.

To link to music listings from the homepage:

```
<a href="music/listings.html">Listings</a>
```

### GRANDCHILD FOLDER

Use the name of the child folder, followed by a forward slash, then the name of the grandchild folder, followed by another forward slash, then the file name.

To link to DVD reviews from the homepage:

```
<a href="movies/dvd/reviews.html">Reviews</a>
```

### PARENT FOLDER

Use `../` to indicate the folder above the current one, then follow it with the file name.

To link to the homepage from the music reviews:

```
<a href="../index.html">Home</a>
```

### GRANDPARENT FOLDER

Repeat the `../` to indicate that you want to go up two folders (rather than one), then follow it with the file name.

To link to the homepage from the DVD reviews:

```
<a href="../../index.html">Home</a>
```

# Risoluzione dei riferimenti relativi

- I riferimenti relativi vengono risolti in riferimenti assoluti utilizzando degli **URI base**
- Un URI base può essere:
  - quello del documento corrente (in cui è citato il riferimento relativo)
  - un URI specificato come base nella intestazione del documento HTML  
`<head> ... <base ref="URI" .... /> ... </head>`

Esempio:

Se la base è <http://www.uniud.it/varie/doc.html>, il riferimento relativo  
“../pippo.html” diventa <http://www.uniud.it/pippo.html>

Significa salta alla cartella immediatamente sopra a quella attuale (la cartella in cui si trova doc.html cioè varie) e inserisci pippo.html

# Osservazioni sulle àncore

- E' importante che l'àncora sia significativa ossia suggerisca all'utente in maniera chiara cosa contiene la pagina di destinazione. Evitare quindi testi tipo "Clicca qui" o simili. L'utente deve sentire "il profumo" della informazione contenuta nella pagina collegata (information scent)

*J.M.Spool, C. Perfetti, D. Brittan (2004) [Designing for the Scent of Information](#). UIE Fundamentals*

- Alcuni siti chiamati ASK Systems anzichè specificare il contenuto di destinazione preferiscono specificare **il ruolo o la funzione retorica** che il contenuto della pagina di destinazione gioca rispetto al contenuto della pagina di partenza. La navigazione assomiglia ad una "conversazione"
- *R.Schank. Engines for education. <http://www.engines4ed.org/hyperbook/>*

# Osservazione

Inserire l'informazione sulla destinazione del link all'interno dell'elemento a non fuori!

```
1 | <p><a href="https://firefox.com/">  
2 |   Click here  
3 | </a>  
4 | to download Firefox</p>
```

**NO!**

```
1 | <p><a href="https://firefox.com/">  
2 |   Download Firefox  
3 | </a></p>
```

**SI**

# Visualizzazione della pagina destinazione

- Per visualizzare la pagina collegata in una nuova finestra del browser si usa l'attributo *target* con valore *\_blank*.

Esempio

```
<a href="http://www.imdb.com" target="_blank"> Internet Movie  
Database</a>
```

- Il motivo principale per visualizzare la pagina collegata in una nuova finestra è permettere all'utente di riprendere la navigazione dalla pagina origine del collegamento una volta visitata la pagina destinazione. E' buona regola avvertire l'utente che la pagina collegata si aprirà in una nuova finestra!

# Esempio

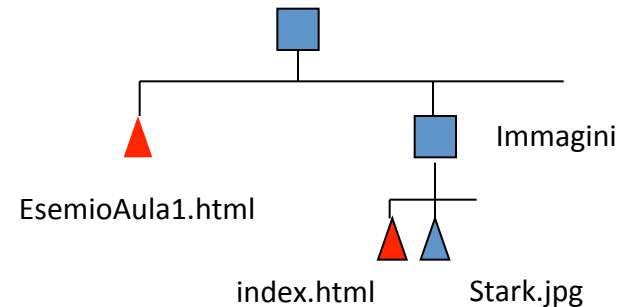
## File: *EsempioAula1.html*

```
<!DOCTYPE html>
<html>
<head>
<title> Titolo del documento </title> <meta charset="utf-8" />
</head>
<body>
<h2> P. Starck </h2>
<p> P. Starck è un progettista industriale che ....bla bla bla. Tra i diversi oggetti che ha progettato
c'è la <a href="Immagini/index.html" >Sedia Luigi XX </a>
che rappresenta un eccellente esempio ..bla bla bla <p>
</body>
</html>
```

## File: *index.html* (nella cartella Immagini)

```
<!DOCTYPE html>
<html>
<head>
<title> Titolo del documento </title> <meta charset="utf-8" />
</head>
<body>
<h3> Sedia Luigi XX di P. Starck </h3>

<p> Torna al <a href=" ../EsempioAula1.html"> testo principale </a></p>
</body>
</html>
```



# Collegamenti a parti specifiche della pagina corrente

- In una pagina "lunga" è utile poter accedere a diverse parti senza dover scorrere il contenuto.
- Per far questo bisogna:
  - identificare le parti cui si vuole accedere (usando, per esempio, l'attributo *id* all'interno di un paragrafo *p* o all'interno di una intestazione *h* )
  - creare un collegamento alla parte desiderata usando come valore di *href* nell'elemento *a* il valore dell'identificatore corretto preceduto dal simbolo '#'. Esempio:

....

```
<a href="#conclusioni"> Conclusioni</a>
```

....

```
<h3 id="conclusioni"> Conclusioni</h3>
```

```
<p> ...testo delle conclusioni.... </p>
```

# Esempio tipico

**<h1 id="top">Film-Making Terms</h1>**

**<a href="#arc\_shot">Arc Shot</a><br />**

**<a href="#interlude">Interlude</a><br />**

**<a href="#prologue">Prologue</a><br /><br />**

**<h2 id="arc\_shot">Arc Shot</h2>**

**<p>A shot in which the subject is photographed by an encircling or moving camera .... </p>**

**<p><a href="#top">Top</a></p>**

**<h2 id="interlude">Interlude</h2>**

**<p>A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film .... </p>**

**<p><a href="#top">Top</a></p>**

**<h2 id="prologue">Prologue</h2>**

**<p>A speech, preface, introduction, or brief scene preceding the the main action or plot of a film; contrast to epilogue .... </p>**

**<p><a href="#top">Top</a></p>**



# Note

- Per collegarsi a parti specifiche di altre pagine dello stesso sito o di siti diversi si usa la stessa tecnica

```
<a href="http://www. htmlandcssbook.com/#bottom"> ... </a>
```

permette di accedere alla parte contrassegnata con l'identificatore 'bottom' nella *home page* del sito [www.htmlandcss.com](http://www.htmlandcss.com)

- E' possibile anche collegarsi a frammenti di risorse multimediali, per esempio a segmenti di video o audio oppure a parti di immagini. Non lo vedremo!

# Collegamento alla posta elettronica

- Per creare un collegamento al programma di posta elettronica si usa il seguente formato

**`<a href="mailto:elio.toppano@uniud.it">Email Toppano</a>`**

**IMMAGINI**

# Immagini

- Esistono diversi tipi di immagini: marchi, fotografie, diagrammi, disegni, mappe, schemi, ecc.
- I formati (bitmap) che vengono usati più frequentemente su web sono: jpeg, png e gif. Usare jpeg per immagini ricche di particolari e colori (es. fotografie); usare png e gif per immagini semplici con pochi colori, parti estese monocromatiche (es. marchi, diagrammi).
- Le immagini possono essere create con dispositivi (es. fotocamere digitali, scanner); applicazioni specifiche (es. Adobe Photoshop, Adobe Fireworks, PaintShop Pro) o acquistate e scaricate da archivi specializzati (es. da Getty Bank o da Corbis di Bill Gates). Vedi: <https://www.gettyimages.it>. Attenzione ai problemi di copyright!

D. Machin. *Building the world's visual language: the increasing global importance of image banks in corporate media*. Visual Communication, Vol 3(3), pp. 316-336, 2004.

# Nota

- Una immagine può avere tre funzioni principali (e relativi significati):
  - **Rappresentazionale.** L'immagine viene usata per : i) denotare una entità specifica (es. un oggetto, una persona, un luogo, un evento) oppure ii) evocare un'idea o un concetto astratto (es. un tipo di persona, una atmosfera, un valore). Esempio: si usa una immagine per descrivere come è fatto un oggetto/processo , per rappresentare una azione/evento o per esprimere una sensazione di contentezza/gioia
  - **Interpersonale.** L'immagine viene usata per far fare, far provare, far credere, ecc. qualche cosa all'utente. Esempio: si utilizza una immagine per attirare l'attenzione, focalizzare l'interesse o per far provare delle emozioni (tristezza, gioia)
  - **Composizionale.** L'immagine viene usata per distinguere, delimitare, separare, integrare, ecc. parti di una composizione visiva nello spazio e nel tempo. Esempio: si utilizza una immagine di sfondo per delimitare e distinguere una area di un sito rispetto alle altre o per stabilire delle correlazioni tra parti di un documento (attraverso la coordinazione del colore)

C. Harrison. *Visual Social Semiotics: understanding how still images make meaning*. Technical Communication, Vol. 50 (1), 2003.



**I WANT YOU  
FOR U.S. ARMY**

NEAREST RECRUITING STATION

# L'elemento *img*

- Per inserire una immagine in un documento HTML5 si può usare l'elemento ***img*** (image). L'elemento *img* è un elemento vuoto.
- L'elemento *img* è un **elemento inline**: se l'immagine viene inserita all'interno di un paragrafo (che è un elemento di tipo blocco) viene visualizzata nel punto in cui ci si trova (segue il flusso del testo senza andare a capo!). Se due elementi *img* sono inseriti in successione le relative immagini vengono visualizzate una di fianco all'altra e non una sopra l'altra.
- L'elemento *img* è accompagnato dall'attributo ***src*** che specifica *dove* si trova l'immagine (es. l'URI della immagine) e dall'attributo ***alt*** che fornisce una descrizione testuale (alternativa) della immagine se l'immagine per qualche motivo non viene visualizzata. L'attributo *alt* è importante, per esempio, quando l'utente usa un lettore di schermo (screen reader) perchè è diversamente abile.

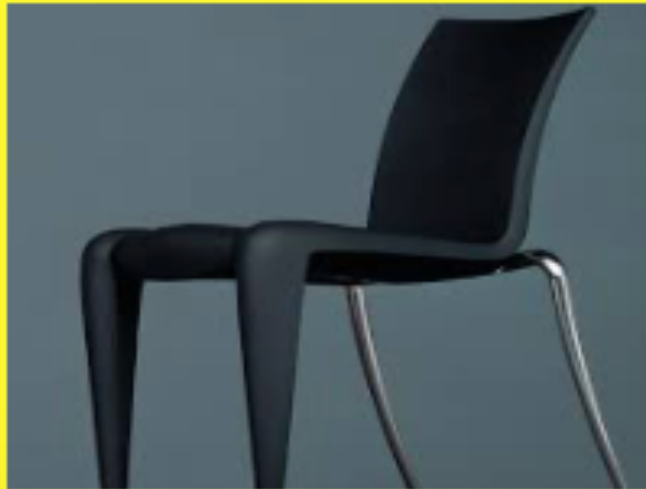
# Gli attributi: title, width e height

- E' possibile associare un titolo (***title***) alla immagine. Il titolo viene visualizzato quando si passa sopra l'immagine col cursore (tooltip)
- La dimensione della immagine (larghezza e altezza) viene specificata mediante l'attributo *style* e le proprietà *width* e *height*. I valori delle proprietà possono essere specificati in valore assoluto (in pixel) o relativo (in %). Usare fogli di stile!
- Esiste un attributo (*align*) per posizionare l'immagine nel documento. L'attributo può assumere i valori: left, right, top, middle, bottom (default). Non usarlo, usare fogli di stile!



# Esempio-0

**Sedia Luigi XX di P. Starck**



...

**<h3> Sedia Luigi XX di P. Starck </h3>**

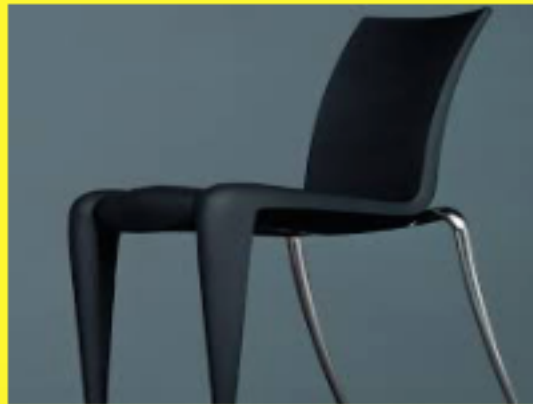
****

...

Nota: si ignori il colore giallo di sfondo

# Esempio-0 bis

**Sedia Luigi XX di P. Starck**



La fotografia seguente

illustra il modello Luigi XX ....

...

**<h3> Sedia Luigi XX di P. Starck </h3>**

**<p> La fotografia seguente  illustra il modello Luigi XX .... </p>**

...

# Gli elementi *figure* e *figurecaption*

- HTML5 consente di raggruppare una immagine (*img*) e la relativa didascalia (*figurecaption*) in un elemento di **tipo blocco** (*figure*)

- Esempio:

```
<figure id="f1" >
```

```

```

```
<figurecaption> <p> Figura3. Sedia Luigi XX di P. Starck </p> </figurecaption>
```

```
</figure>
```

Nota: in questo caso, se due elementi *figure* sono inseriti in successione le relative immagini-didascalie vengono visualizzate una sopra l'altra e non una di fianco all'altra

Cosa succede se si toglie l' elemento <p> in <figurecaption>?Provare...

# Aree attive all'interno di una immagine

- Per realizzare un'area attiva (un hotspot) all'interno di una immagine si usa l'attributo *usemap* (di *img*) il cui valore è il nome (preceduto dal carattere '#') di un elemento di tipo mappa (*map*) che descrive la forma (*shape*) e la posizione (*coords*) dell'area cliccabile
- Un elemento di tipo mappa (*map*) ha un nome (*name*) ed eventualmente un identificatore (*id*). Contiene uno o più elementi area (*area*) con gli attributi principali: *shape*, *coords*, *href* e *target*.
- L'attributo *shape* può assumere i seguenti valori: rettangolare (*rect* o *rectangle*), circolare (*circ* o *circle*), poligonale (*poly* o *polygon*). L'attributo *coords* permette di impostare, a seconda della figura geometrica scelta, le rispettive coordinate in pixel. Le coordinate sono riferite al vertice superiore sinistro della immagine di coordinate (0,0).

# Esempio

...

```

```

```
<map id="m1" name="mappa1">
```

```
<area shape="rect" coords="0,0,100,50" href="http://www.starck.com"  
title="Sito ufficiale di Starck" alt="Ancora al sito di Stark" target="_blank" />
```

```
</map>
```

...

Cliccando sull'area attiva di forma rettangolare e specificata dalle coordinate di 2 vertici (superiore sinistro 0,0; e inferiore destro 100,50), ci si collega alla pagina web ufficiale di Starck che viene visualizzata in una nuova finestra. Per le aree circolari si specificano le coordinate del centro (x,y) e il valore del raggio (r); per le aree poligonali si fornisce una successione di (coordinate di) punti che ne descrivono il perimetro

**MODULI**

# Moduli

- Un modulo (*form*) serve per raccogliere informazioni dall'utente, esprimere scelte o avviare azioni specifiche. Le informazioni raccolte possono essere usate per eseguire delle ricerche in basi di dati, modificare i dati di una base di dati, creare pagine dinamiche.
- Il modulo può presentare diversi *controlli grafici* di interfaccia (widget) dedicati a raccogliere informazioni/azioni specifiche. Esempi:
  - inserimento di una parola o di una breve linea di testo in chiaro (*text input*)
  - inserimento di una parola o una linea di testo nascosta (*password input*)
  - inserimento di messaggi o commenti su più righe di testo (*text area*)
  - scelta obbligatoria di una tra N voci alternative (*radio buttons*)
  - selezione e deselezione di una o più voci tra N alternative (*checkboxes*)
  - scelta obbligatoria di una o più voci tra N alternative presenti in una lista (*drop down boxes*)
  - invio dei dati dal modulo ad un'altra pagina (*submit button* e *image button*)
  - caricamento di file (*file upload*)

A user fills in a form and then presses a button to submit the information to the server.

Come funziona?





A form may have several form controls, each gathering different information. The server needs to know which piece of inputted data corresponds with which form element.

NAME  
username—Ivy  
VALUE

To differentiate between various pieces of inputted data, information is sent from the browser to the server using name/value pairs. In this example, the form asks for the visitor's username and also for their favorite jazz musician. The name/value pairs sent to the server are:

username—Ivy

If the form control allows the user to enter text, then the value of the form control is whatever the user has typed in.

vote—Herbie

If the form control allows you to choose from a fixed set of answers (e.g. radio buttons, checkboxes or a drop down list), the web page author will add code that gives each option an automatic value.

You should never change the name of a form control in a page unless you know that the code on the server will understand this new value.

# L'elemento *form*

- Per creare un modulo si usa l'elemento *form*  
**<form> .... </form>**
- Gli attributi più importanti dell'elemento *form* sono:
  - *action*: specifica l'indirizzo (URL) del programma a cui si debbono inviare i dati forniti dall'utente
  - *method*: specifica come inviarli (es. get o post)
  - *id*: associa un identificatore al form che è utile per richiamarlo in scripts o in regole CSS

**<form action="url del programma", method="get" id="form1"> .... </form>**

# I metodi *get* e *post*

- Col metodo ***get*** i dati (es. nome=valore) vengono concatenati all'indirizzo (URL) specificato dall'attributo *action* (si usa il carattere '?' per identificare l'inizio della stringa di dati e il carattere '&' per separare i dati tra di loro). Il metodo produce una stringa che è visibile nella barra degli indirizzi del browser. Questo metodo dovrebbe essere usato quando i dati da trasmettere sono pochi (nota: c'è un limite alla lunghezza della stringa che dipende da browser!), non sono dati binari, non costituiscono informazione sensibile, non modificano lo stato di data base esistenti (sono usati per ricerca-recupero non per modifica)
- Col metodo ***post*** i dati vengono inseriti nel corpo del messaggio HTTP di richiesta. Dovrebbe essere usato quando i dati da trasmettere sono molti (es. un intero file), possono essere dati binari, non costituiscono informazione sensibile, o quando i dati modificano lo stato di un data base esistenete (es. per l' inserimento, la sostituzione, o la eliminazione di dati)

# Elementi del modulo (1)

- **input di tipo testo** (<input type="text" />): rappresenta un campo, a riga singola, destinato a ricevere un dato di tipo testo. Ha un *nome* (che è il nome della variabile che conterrà il dato), e un *valore* (il dato digitato dall'utente)
- **input di tipo password** (<input type="password" />): rappresenta un campo, a riga singola, destinato a ricevere una password. Ha un *nome* (es. password), e un valore (il dato digitato dall'utente che però non viene visualizzato in chiaro)
- **input di tipo invia** (<input type="submit" />): si crea un pulsante che permette di inviare i dati alla pagina destinazione attraverso il metodo scelto. L'attributo *name* non è necessario, l'attributo *value* permette di specificare una etichetta associata al controllo

## ESEMPIO:

```
<form action= "prova.php" method="post" id="form1">
<p> <strong>Nome: </strong> <input type="text" name="nome" /> </p>
<p> <strong>Cognome: </strong><input type="text" name="cognome" /> </p>
<p> <strong>Password: </strong><input type="password" name="password" /> </p>
  <input type="submit" name="submit" value="invia" />
</form>
```

- **Pulsante** (<input type="button" />): permette di creare un comune pulsante con una etichetta testuale (tramite l'attributo *value*) a cui può essere associato del codice (es. in JavaScript) da eseguire quando si fa clic su di esso.

```
<input type="button" onclick="alert('Ciao a tutti!)" value="Premi qui" />
```

- Nota: per inviare i dati si può anche usare l'**elemento** <button>  
<button type="submit" > Premi qui </button>

# Elementi di un modulo (2)

- **textarea**: rappresenta un campo destinato a ricevere più linee di testo. Ha un nome (che è il nome della variabile che conterrà il dato), e un valore (il testo digitato dall'utente). Può contenere un testo iniziale

Esempio (provare):

```
<form action="prova.php" method="post" id="form2">  
<p> <strong>Inserire un commento:</strong></p>  
<textarea name="commento" rows="4" cols="40" > Qui va il commento </textarea>  
<p><input type="submit" name="submit" value="invia" /></p>  
</form>
```

- **input di tipo radio** (<input type="radio" />): rappresenta un controllo che permette all'utente di esprimere una scelta tra una o più alternative e la contestuale deselezione di un altro controllo precedentemente selezionato
- **input di tipo checkbox** (<input type="checkbox" />): rappresenta un controllo che permette all'utente di esprimere una o più scelte (non obbligatorie) tra N alternative disponibili.

# Elementi di un modulo (3)

- Esempi (provare):

```
<form action="prova.php" method="post" id="form3">
<p><strong> Scegliere uno tra i seguenti generi alternativi: </strong></p>
<p> Romanzo <input type="radio" name="genere" value="romanzo" />
Film <input type="radio" name="genere" value="film" />
Gioco <input type="radio" name="genere" value="gioco" checked="checked"/>
</p><p><input type="submit" name="submit" value="invia" /></p>
</form>
```

Nota: nell'esempio sopra si usa l'attributo booleano checked per esprimere una scelta di default (gioco). Scegliendo un altro genere quello di default viene deselezionato.

```
<form action="prova.php" method="post" id="form4">
<p> <strong>Scegliere tra i seguenti generi: </strong><br/>
<input type="checkbox" name="genere" value="romanzo" checked="checked" /> Romanzo
<input type="checkbox" name="genere" value="film" /> Film
<input type="checkbox" name="genere" value="gioco" /> Gioco </p><p><input
type="submit" name="submit" value="invia" /></p> </form>
```

# Elementi di un modulo (4)

- **selezione di una opzione:** rappresenta un controllo che permette all'utente di selezionare una tra N alternative disponibili visualizzate mediante una lista (drop down list box)

Esempio:

```
<form action="prova.php" method="post" id="form5">
<p> <strong>Scegliere uno tra i seguenti generi: </strong></p>
<select name="genere" >
<option value="romanzo" > romanzo </option>
<option value="film" selected="selected"> film </option>
<option value="gioco"> gioco </option>
</select><p><input type="submit" name="submit" value="invia" /></p>
</form>
```

**Nota:** è possibile selezionare dalla lista più voci. Nell'elemento select, si usa l'attributo booleano *multiple* per attivare la selezione multipla. Si usa il tasto Shift per indicare l'intervallo di scelta, il tasto CTRL per le scelte a scacchiera. L'attributo *size* serve per specificare il numero di scelte visualizzate



# Elementi di un modulo (5)

- **selezione di un file da caricare sul server** (<input type="file" />): rappresenta un controllo che permette all'utente di scegliere il file da trasmettere e avviarne la trasmissione (input type submit)

Esempio:

```
<form action="prova.php" method="post" id="form6">  
<p> <strong>Scegliere il file da trasmettere: </strong></p>  
<input type="file" name="miofile" />  
<input type="submit" name="submit" value="invia" />  
</form>
```

- **input di tipo immagine** (<input type="image" />): permette di usare una immagine come bottone di invio. Si possono usare gli attributi soliti per le immagini (*alt, title, width, height*)

```
<input type="image" src="..." alt=".." width=".." height=".." />
```

# Elementi di un modulo (6)

- **etichette** : per rendere accessibile il modulo è necessario che ciascun controllo abbia una sua etichetta. Si usa l'elemento *label*. **ATTENZIONE**: è importante usarle sempre!

Esempio:

...

```
<p>Gender:</p>
```

```
<p> <label for="female">Female</label>
```

```
<input id="female" type="radio" name="gender" value="f" /></p>
```

```
<p><label for="male">Male</label>
```

```
<input id="male" type="radio" name="gender" value="m" /></p>
```

...

**Nota**: cosa accade se l'utente clicca sulla etichetta? Provare!

L'elemento label si può usare anche per raggruppare un testo descrittivo e il relativo controllo di input:

Esempio

```
<label>Nome: <input type="text" name="nome" /></label>
```

è buona pratica, comunque, usare sempre l'attributo *for*:

```
<label for="c1">
```

```
Nome: <input type="text" name="nome" id="c1" />
```

```
</label>
```

# Elementi di un modulo (7)

- **elementi *fieldset* e *legend*:** *fieldset* è usato per raggruppare più controlli di un form disegnando attorno ad essi un box rettangolare; *legend* consente di assegnare una didascalia o un titolo ad un elemento *fieldset*. L'elemento *legend* va immediatamente dopo il tag di apertura di *fieldset*
- Esempio:

```
<form action="prova.php" method="post" id="form6">  
<fieldset>  
<legend>Contatti</legend>  
<p><label for="c1">Email:</label> <input type="text" name="email" id="c1" /></p>  
<p><label for="c2">Cell:</label><input type="text" name="mobile" id="c2"/></p>  
<p> <label for="c3">Telefono:</label> <input type="text" name="telephone" id="c3"/></p>  
<input type="submit" name="submit" value="Invia" />  
</fieldset>  
</form>
```

# Osservazioni

- **Reset** (`<input type="reset" />`): questo elemento permette di creare un pulsante con etichetta testuale (tramite l'attributo `value`) che, quando attivato, consente di cancellare gli eventuali dati del relativo form
- **Hidden** (`<input type="hidden" />`): questo elemento non viene visualizzato. Permette di associare un valore ad una variabile che specifica per esempio dove l'utente si trova quando riempie il modulo  
`<input type="hidden" name="bookmark" value="prodotti" />`
- **Required**: con questo attributo si indica che il valore di un elemento di tipo `input`, `select` o `textarea` è obbligatoriamente richiesto per il corretto submit del form

# **Approfondimento su form**

# Inserimento di dati particolari (HTML5)

- Inserimento di una data nel formato anno-mese-giorno (YYYY-MM-DD):  
`<input type="date" name="data" />`
- Inserimento di un orario nel formato ore-minuti-secondi e frazioni di secondi (HH:MM:SS:FFF):  
`<input type="time" name="ora" />`
- Inserimento di un indirizzo di email (es. nome@dominio)  
`<input type="email" name="email" />`
- Inserimento di un indirizzo URL (es. http://www.uniud.it):  
`<input type="url" name="url" />`

**Nota:** i browser che supportano la validazione possono controllare se i dati inseriti sono nel formato corretto

# Inserimento di dati particolari ...cont..

- Inserimento di testo per effettuare delle ricerche

```
<input type="search" name="ricerca" placeholder="Introduci keyword" />
```

- Inserimento di un dato numerico

```
<input type="number" name="numero" min="5" max="10" step="1.5" />
```

- Inserimento di un colore

```
<input type="color" name="colore" value="#ffffff"/>
```

- Inserimento di un intervallo di valori:

```
<input type="range" name="valore" min="0" max="100" />
```



# Elenco completo dei tipi di *input*

## ▼ <input> types

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

 <input type="datetime">

<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">

<input type="time">

<input type="url">

<input type="week">

# **AUDIO E VIDEO (HTML5)**

# Osservazioni

- I nuovi elementi di HTML5 per visualizzare video e audio sono supportati solo dai browser più recenti!
- I formati dei video maggiormente usati sono:
  - MP4
  - WebM
  - Ogg
- I formati audio maggiormente usati sono:
  - MP4
  - MP3
  - Ogg Vorbis
- E' necessario verificare quali formati sono supportati da quali browser (e da quale versione in poi). Si veda: [www.w3school.com](http://www.w3school.com)

Format	File	Description	Formati video
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML.	
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.	
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.	
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.	
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers.	
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.	
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.	
WebM	.webm	WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML.	
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube.	

# Formati audio

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. Supported by all browsers.

# L'elemento *video*

- Per inserire un video nella pagina si usa l'elemento `<video> .. </video>` che ha diversi attributi. Alcuni esempi:
  - ***src***: specifica dove si trova il file video da visualizzare
  - ***poster***: permette di specificare una immagine che viene visualizzata mentre il video è scaricato o fintantochè l'utente non lo avvia
  - ***width e height***: specificano la dimensione della regione in cui visualizzare il video (in pixel)
  - ***preload***: indica al browser se scaricare il video quando viene caricata la pagina o aspettare che l'utente attivi il controllo play
  - ***controls***: si chiede al browser di fornire i controlli per la riproduzione del video
  - ***loop***: specifica che il video deve ripartire dall'inizio una volta terminato
  - ***autoplay***: specifica che il video deve partire automaticamente una volta caricato

# Esempio

```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <title> Esempio video </title></head>
```

```
<body>
```

```
<video controls width="300" height="300" autoplay loop muted  
  preload="auto">
```

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

```
  <p> Video non supportato da questo browser ...</p>
```

```
</video>
```

```
</body>
```

```
</html>
```

# Versioni multiple del video

- Per specificare più sorgenti si usa l'elemento `<source ../>` all'interno dell'elemento `<video>`

Esempio:

```
<video poster="images/puppy.jpg" width="400" height="320" preload controls loop>  
  <source src="video/puppy.mp4"  
    type='video/mp4;codecs="avc1.42E01E, mp4a.40.2"' />  
  <source src="video/puppy.webm"  
    type='video/webm;codecs="vp8, vorbis"' />  
  <p>A video of a puppy playing in the snow</p>  
</video>
```



# L'elemento *audio*

- Per inserire un file audio nella pagina si usa l'elemento `<audio> .. </audio>` che ha diversi attributi. Alcuni esempi:
  - *src*: specifica il file audio da riprodurre
  - *controls*: si chiede al browser di fornire i controlli per la riproduzione audio
  - *loop*: specifica che l'audio deve ripartire dall'inizio una volta terminato
  - *autoplay*: specifica che l'audio deve partire automaticamente una volta caricato

Esempio:

```
<audio src="song.mp3" controls autoplay>  
<p>This browser does not support our audio format.</p>  
</audio>
```

# Versioni multiple dell'audio

- Per specificare più sorgenti si usa l'elemento `<source ../>` all'interno dell'elemento `<audio>`

- Esempio:

```
<audio controls autoplay>
```

```
  <source src="audio/test-audio.ogg" />
```

```
  <source src="audio/test-audio.mp3" />
```

```
  <p>This browser does not support our audio format.</p>
```

```
</audio>
```

# Bibliografia

- Jon DUCKETT. [HTML & CSS: design and build websites](#). John Wiley & Sons, Inc., Indianapolis, Indiana, 2011. **Capitoli 4,5,9 e 17**
- Risorse su web
- **w3c school** : <https://www.w3schools.com>
- **mds web docs**: <https://developer.mozilla.org/it/>