

Linguaggi

*Corso M-Z - Laurea in Ingegneria Informatica
A.A. 2008-2009*

Alessandro Longheu

<http://www.dit.unict.it/users/alongheu>

alessandro.longheu@dit.unict.it

- lezione 12 -

HTML - 2

1

A. Longheu – Linguaggi M-Z – Ing. Inf. 2008-2009

HTML

- Introduzione all'HTML,
- Struttura di un documento HTML
- TAG di **organizzazione** del testo
 - Intestazioni
 - Paragrafi e ritorni a capo
 - HR
 - Blocchi
 - Rientri
- TAG di **formattazione** del testo, rappresentazione colori, caratteri speciali
- Link: esterni, interni, etichettati
- Immagini
- Liste
- Tabelle
- Frames
- Mappe
- Tool per creazione pagine

2

HTML - Immagini

- Nei documenti HTML è possibile inserire immagini ed i formati più comunemente utilizzati sono GIF (*Graphic Interchange Format*) e JPEG (*Joint Photographic Experts Group*); i file corrispondenti hanno rispettivamente le estensioni «.gif» e «.jpg» o «.jpeg».
- Il formato GIF è stato sviluppato da Comuserve alla fine degli anni '80 ed è particolarmente adatto per linee, icone, immagini generate dal computer e con colori netti, non sfumati; inoltre grazie a specifici software, le immagini GIF possono anche essere rese trasparenti e animate.
- Il formato JPEG si è diffuso dal 1993 ed è progettato per le fotografie e per altre immagini con colori sfumati.
- Tra i due il formato GIF è senz'altro il più diffuso per il fatto che le immagini GIF sono solitamente poco «pesanti» e quindi si trasferiscono in tempi brevi. Si deve infatti notare che, quando viene richiesta ad un server Web una pagina contenente immagini, quest'ultimo invia in risposta il file HTML corrispondente alla pagina e, separatamente, i file dove sono memorizzate le immagini; se questi ultimi sono molto grandi (e in presenza di collegamenti lenti alla rete) avremo dei tempi di attesa abbastanza lunghi per il completamento a video della pagina richiesta.

HTML - Immagini

- A vantaggio delle immagini JPEG c'è la maggiore qualità (sono "a 24 bit" cioè possono avere 16 milioni di colori contro gli "8 bit" e 256 colori delle GIF) e la compattezza: la stessa immagine in formato JPEG occupa circa un quarto dello spazio in byte del formato GIF.
- Di solito comunque le immagini JPEG hanno un maggiore occupazione di memoria perché create con apparecchi fotografici, mediante acquisizione dallo scanner o da altre fonti ad alta qualità e definizione.
- Nel 1995 il W3C ha definito un nuovo formato di immagini per la rete, il PNG (*Portable Network Graphics*) che, almeno nelle intenzioni, avrebbe dovuto sostituire gli altri formati; infatti PNG prevede il supporto da 256 colori (con PNG-8 che può essere considerato il sostituto di GIF) fino a 16,7 milioni di colori (con PNG-24 che dovrebbe sostituire JPEG).
- Nella realtà il formato PNG, pur utilizzato, non ha mai sostituito del tutto gli altri formati. Di solito in un documento Web si trovano immagini di tutti i formati.
- Esistono altri formati (.tiff, .tga), il loro supporto da parte del browser è variabile, quindi non sono sempre visualizzate

HTML - Immagini

- L'elemento che si utilizza per l'inserimento di immagini è ``; è un elemento di testo, prevede l'attributo `'src'` per specificare il file contenente l'immagine e l'attributo `'alt'` per indicare un testo alternativo nel caso quest'ultima non possa essere visualizzata. Esempio:
 - ``
 - In questo modo si inserisce l'immagine contenuta nel file di nome `'topolino.gif'` residente nella directory corrente; il testo alternativo è *Topolino*.
- I motivi per cui può essere necessario il testo alternativo?:
 - il *browser* non supporta la grafica;
 - il *browser* è programmato per pronunciare testo al posto di immagini (per utenti non vedenti);
 - il *browser* è stato configurato solo per il testo in modo da velocizzare il caricamento dei documenti Web;
 - dare la possibilità al *browser* di visualizzare il testo alternativo mentre l'immagine viene caricata;
 - l'immagine da visualizzare non viene trovata dal server Web.

5

HTML - Immagini

- Le immagini possono essere posizionate con l'attributo `'align'` del tag ``; i valori possibili sono "top", "bottom" e "middle" (esistono anche "texttop", "absmiddle" e "absbottom") per l'allineamento verticale e "left" e "right" per l'allineamento orizzontale.
- Il valore per difetto è "top" grazie al quale il bordo superiore dell'immagine è allineato con la riga di testo di cui essa fa parte (si ricordi che l'elemento `` è di testo). Invece con "bottom" è il bordo inferiore dell'immagine ad essere allineato con la riga di testo, mentre con "middle" l'allineamento è con la parte centrale dell'immagine.
- "texttop" allinea il bordo superiore dell'immagine con il bordo superiore della riga di testo, "absmiddle" allinea il centro dell'immagine con il centro dell'oggetto più esteso nella riga, "absbottom" allinea il bordo inferiore dell'immagine con il bordo inferiore dell'oggetto più basso nella riga.
- Con le scelte "right" e "left" l'immagine viene posizionata rispettivamente al margine destro o sinistro della pagina ed il testo che segue l'elemento `` viene visualizzato affiancato all'immagine stessa.
- Se si vuole interrompere la visualizzazione del testo affiancato e far posizionare gli elementi successivi sotto l'immagine si deve usare l'interruzione di linea con l'attributo `'clear'` con `<br clear="left/ right/ all/">` rispettivamente nel caso che l'immagine sia sul bordo sinistro o sul bordo destro o ci siano immagini su entrambi i lati.

6

HTML - Immagini

```

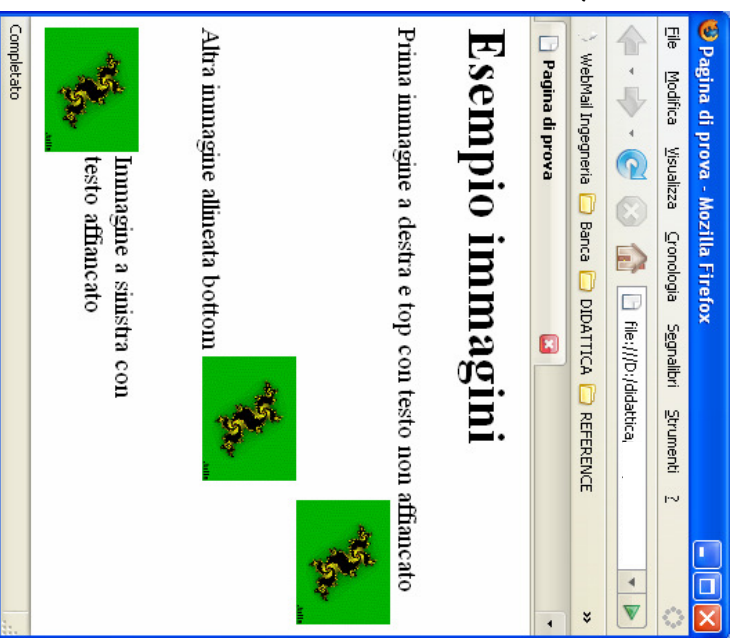
</html>
</head>
<title>Esempio immagini</title></head>
<body>
<h1>Esempio immagini</h1>
Prima immagine a destra
e top con testo non affiancato

<br clear="right">
Altra immagine allineata bottom

<p><br/>

Immagine a sinistra con
<br/> testo affiancato</p>
</body>
</html>

```



A. Longheu – Linguaggi M-Z – Ing. Inf. 2008-2009

HTML - Immagini

- Attraverso l'uso di altri attributi del tag `` si possono dimensionare le immagini, impostare lo spessore del bordo e stabilire quanto spazio vuoto lasciare tra esse e il testo che le circonda.
- Gli attributi `'width'` e `'height'` permettono di specificare rispettivamente la larghezza e l'altezza di un'immagine espressa in *pixel*. Esempio:


```

```
- L'attributo `'border'` permette di specificare un bordo di un certo spessore espresso in *pixel*, il valore di difetto e zero che significa nessun bordo. Esempio:


```

```
- Si noti che in questo caso l'immagine avrà una larghezza totale di 120 *pixel* e un'altezza totale di 220 *pixel*.
- Con gli attributi `'hspace'` e `'vspace'` si indica infine lo spazio vuoto, sempre espresso in *pixel*, da lasciare a sinistra o destra (con `'hspace'`) e sopra e sotto (con `'vspace'`) all'immagine
- E' possibile usare immagini al posto del testo o anche insieme al testo come àncore per i collegamenti. Occorre ricordare che alcuni navigatori aggiungono automaticamente un bordo blu intorno ad una immagine collegamento per evidenziare la presenza di quest'ultimo, quindi non è opportuno definire bordi personalizzati per tali immagini.

HTML - Tabelle

- Le tabelle sono elementi molto usati per la realizzazione di documenti in HTML in quanto permettono di organizzare i dati in strutture formate da righe e colonne ma soprattutto perché con esse si può impostare l'aspetto di una pagina disponendo i paragrafi in colonne, oppure creando dei margini, o ancora distribuendo testo ed immagini in modo più vario.
- Questo modo di usare le tabelle, pur ancora molto frequente, è da considerare deprecato a vantaggio del posizionamento ottenuto attraverso i fogli di stile.
- Nelle tabelle di un documento HTML, come in qualsiasi tabella si identificano i seguenti componenti:
 - il titolo, opzionale, della tabella;
 - le celle, cioè le intersezioni tra righe e colonne, che contengono i dati;
 - le intestazioni delle righe o colonne, cioè delle celle contenenti le etichette che identificano i tipi di dati delle righe o colonne corrispondenti.
- La tabella più piccola che si può creare contiene una sola cella; non ci sono invece restrizioni sul numero massimo di celle definibili (esiste però un limite dettato da ragioni pratiche)

A. Longheu – Linguaggi M-Z – Ing. Inf. 2008-2009

HTML - Tabelle

```

<table border>
<caption> Tabella di
prova </caption>
<tr>
<th> Alunno </th>
<th> Voto orale </th>
<th> Voto scritto </th>
<th> Media </th>
</tr> <tr>
<td> Pippo </td>
<td> 5 </td>
<td> </td>
<td> &nbsp; </td>
</tr> <tr>
<td> Pluto </td>
<td> 6 </td>
<td> 8 </td>
</tr>
</table>
  
```



HTML - Tabelle

- richiede il tag `</table>` alla fine della definizione della tabella stessa (altrimenti la tabella non viene visualizzata!)
- Con `<caption>` e `</caption>` si può inserire il titolo della tabella.
- Il tag `<tr>` definisce le varie righe, suddivise in celle il cui contenuto è definito con `<td>` per le celle normali, o con `<th>` per le intestazioni. I tag `</tr>`, `</td>`, `</th>` non sono obbligatori
- All'interno del titolo, cioè tra `<caption>` e `</caption>` si possono inserire solo elementi di testo. Con l'attributo `'align'` si può decidere se il titolo sta sopra (impostazione per difetto) o sotto la tabella

11

HTML - Tabelle

```

<table border
  cellpadding="9">
  <caption> Tabella
    2</caption>
  <tr>
    <th>Matematica</th>
    <td>Prof. Pippo</td>
    <td>Ore: 5</td>
  </tr>
  <tr>
    <th>Storia</th>
    <td>Prof. Pluto</td>
    <td>Ore: 3</td>
  </tr>
</table>

```



12

HTML - Tabelle

- Gli attributi degli elementi per la definizione delle tabelle sono tutti opzionali; se non se ne specifica alcuno si ottiene una tabella allineata a sinistra e senza bordi.
- Iniziamo a considerare gli attributi del marcatore `<table>` che sono: `'align'`, `'border'`, `'cellspacing'`, `'cellpadding'`, `'bgcolor'`, `'width'`.
- Con l'attributo `'align'` si specifica l'allineamento della tabella; i valori possono essere "left" (valore per difetto), "center" e "right".
- Attraverso `'bgcolor'` si specifica il colore dello sfondo della tabella.
- L'attributo `'border'` permette di aggiungere i bordi a tutte le celle della tabella; se si specifica anche un valore in pixel si ottiene il dimensionamento dei bordi, ma solo di quelli esterni.
- Con `'cellspacing'` e `'cellpadding'` si può specificare di quanti pixel distanziare rispettivamente una cella da un'altra cella e il bordo delle celle dal loro contenuto.
- Con `'width'` si imposta la grandezza della tabella, in percentuale rispetto alla larghezza della finestra del browser, oppure in base a un valore espresso in pixel.
- Se questo attributo non viene usato, la tabella è dimensionata in base al contenuto della sue celle; a questo proposito è opportuno sottolineare, come emerge anche dagli esempi, che la larghezza di una colonna di una tabella è determinata dalla larghezza della cella più grande appartenente alla colonna stessa; allo stesso modo l'altezza di una riga si adegua all'altezza della cella più grande presente in essa

13

HTML - Tabelle

- Il tag `<tr>`, `<th>`, `<td>` prevedono gli attributi `'align'`, `'valign'`, `'bgcolor'`; esistono inoltre gli attributi `'rowspan'`, `'colspan'`, `'width'` che si applicano solo a `<th>` e `<td>`.
- Con `'align'` si imposta l'allineamento orizzontale dei dati in una cella o in tutte le celle di una riga a seconda che tale attributo sia abbinato a `<td>` e `<th>` oppure a `<tr>`; il valore per difetto è "left" e gli altri possibili sono "center" e "right".
- Con `'valign'` invece si imposta l'allineamento verticale dei dati; i valori possibili sono: "top", "middle" (valore per difetto), "bottom", "baseline". Il valore "baseline" è simile a "top" e imposta quindi un posizionamento in alto nella cella; la differenza è nel fatto che con "baseline" la parte inferiore della prima riga di testo di ogni cella è sempre allineata qualunque sia la dimensione dei caratteri usati nelle varie celle.
- L'attributo `'bgcolor'` permette di specificare il colore di sfondo di una riga o di una cella e questa impostazione prevale su quella dell'intera tabella
- Gli attributi `'colspan'` e `'rowspan'` servono ad unire orizzontalmente e verticalmente più celle di una tabella;
- `'width'` specifica la larghezza in *pixel* di una cella indipendentemente dal contenuto;
- `'nowrap'` impone di non fare andare a capo il testo entro una cella

14

HTML - Tabelle

```

<table border>
<caption> Tabella n. 4</caption>
<tr>
<th rowspan="2">Milano</th>
<td>Minima</td>
</tr>
</tr>
<tr>
<td>Massima</td>
</tr>
<td>Massima</td>
</tr>
<td>10</td>
</tr>
</tr>
<tr>
<th rowspan="2">Roma</th>
<td>Minima</td>
</tr>
<tr>
<td>Massima</td>
</tr>
<td>4</td>
</tr>
<td>13</td>
</tr>
</table>

```



15

HTML - Tabelle

- In una cella si può inserire praticamente tutto ciò che può essere definito nella sezione 'body' di un documento: testo; elementi di blocco (paragrafi, elenchi, altre tabelle, ecc.) elementi di testo (font, collegamenti, immagini, ecc.)
- Grazie a questo, come detto in precedenza, le tabelle vengono anche usate per definire la disposizione degli elementi all'interno della pagina; ovviamente in questo caso si evita di definire i bordi, le celle di intestazione, e tutto quello che può rendere «visibile» la tabella stessa.
- Il documento HTML dell'esempio seguente è appunto «impaginato» usando una tabella che qui serve a posizionare con precisione tre piccole immagini allineate orizzontalmente e le relative didascalie sotto di esse
- Il risultato che si ottiene inserendo i vari elementi come contenuto delle celle di una grossa tabella è solitamente accettabile, ma è utile ribadire che lo stesso risultato (se non migliore) si ottiene in modo più appropriato ed «elegante» con i fogli di stile

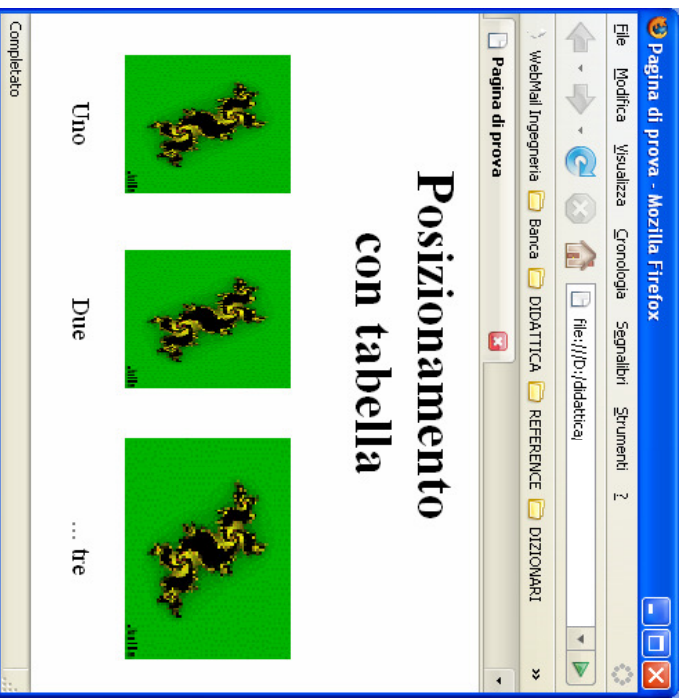
16

HTML - Tabelle

```

<table cellpadding="10">
<tr>
<td width="200"
  align="center"></td>
<td width="200"
  align="center"></td>
<td width="200"
  align="center"></td>
</tr>
<tr>
<td align="center">Uno</td>
<td align="center">Due</td>
<td align="center">... tre</td>
</tr></table>

```



17

HTML - Tabelle

Una tabella è nidificata quando viene inserita in una cella di un'altra tabella. L'uso di tabelle nidificate è anche un modo per ottenere due tabelle affiancate; basta infatti inserire le due tabelle in due celle di una tabella senza bordo

```

<table align="center">
<caption> Tabella n. 5</caption>
<tr><td><table border="10">
<caption> I Quadrimestre</caption>
<tr><td></td><th>Orale</th><th>Scritto</th><th>Pratico</th>
<tr><td><td>Pippo</td><td>6</td><td>7</td><td>7</td></tr>
<tr><td><td>Pippo</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td><td>Pluto</td><td>7</td><td>8</td><td>8</td></tr>
</table></td>
<td><table border="10">
<caption> II Quadrimestre</caption>
<tr><td><td></td><th>Voto finale</th>
<tr><td><td>Pippo</td><td>7</td>
<tr><td><td>Pluto</td><td>8</td>
</table></td>
</tr></table>

```

I Quadrimestre			II Quadrimestre
Orale	Scritto	Pratico	Voto finale
Pippo	6	7	7
Pippo	7	7	7
Pluto	7	8	8

HTML - Frame

- L'uso dei frame è deprecato a meno che non si faccia riferimento alla variante «Frameset» dell'HTML; qui vengono trattati perché permettono in modo semplice di gestire più documenti HTML in una singola pagina.
- I riquadri sono delle suddivisioni della finestra del *browser* in cui possono essere visualizzati documenti HTML diversi.
- Un esempio in cui possono essere utili è se si hanno sitiWeb con molti livelli di pagine: in tal caso si potrebbe pensare di usare un riquadro per visualizzare un indice del sito in modo permanente ed un altro per contenere le varie pagine che si richiamano durante la navigazione.
- Per definire una pagina che deve contenere riquadri si usa un documento HTML particolare chiamato '**frameset**' che si differenzia da quelli visti finora in quanto contiene l'elemento '**<frameset>**' al posto di '**<body>**'.

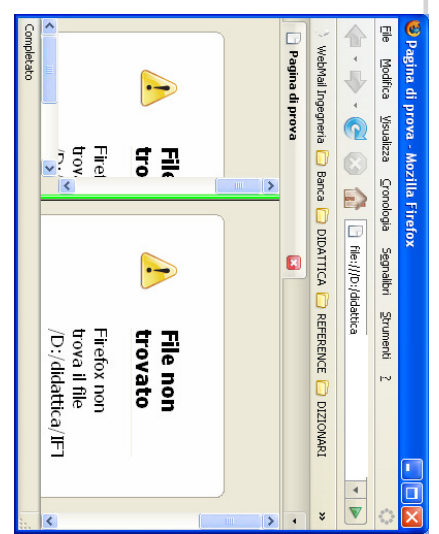
19

HTML - Frame

```

<html>
<head><title>Esempio con riquadri</title>
</head>
<frameset cols="35%,65%"
bordercolor="#00FF00">
<frame src="doc1.html"/>
<frame src="doc2.html" name="frame2"/>
</frameset>
<body>

```



Il vostro browser non gestisce i frame.

Per visualizzare i documenti cliccare sui link:

- `Primo documento`
- `Secondo documento`
- `</body></noframes>`
- `</frameset></html>`
- Bisogna ovviamente creare doc1 e doc2 come file html separati

20

HTML - Frame

- Il bordo dei riquadri viene impostato di colore verde con l'attributo `'bordercolor'` (il default è il grigio); grazie all'attributo `'cols'` con valore "35%,65%" si ha una suddivisione in due colonne contenenti rispettivamente i documenti 'doc1.html' e 'doc2.html'.
- La parte del documento compresa tra i marcatori `'<noframes>'` e `'</noframes>'` serve a rendere fruibile la pagina anche ai programmi di navigazione che non gestiscono i riquadri; essa è alternativa al `'frameset'` e contiene una sezione `'body'` come i «normali» documenti.
- Volendo ottenere una suddivisione in riquadri orizzontali si deve usare l'attributo `'rows'` al posto di `'cols'`; è anche possibile usare entrambi gli attributi per definire delle "griglie" di riquadri, ad esempio:
 - `<frameset cols="300,*" rows="25%,50%,25%">`
- La grandezza può essere specificata in pixel (300 nell'esempio); «*» significa che la colonna occupa lo spazio restante; `cols="2*,*"` significa che la prima riga deve essere il doppio della seconda.
- Esiste anche la possibilità di usare riquadri nidificati continuando a dividerli in più parti fino ad un massimo di nove riquadri.
- Si deve però tenere presente che l'uso contemporaneo di più di due o tre riquadri in una finestra può favorire errori nella realizzazione del documento e può creare disorientamento nell'utente che lo visualizza.

21

HTML - Frame

```

<html>
<head>
<title>Riquadri nidificati</title>
</head>
<frameset cols="250,*">
<frame src="doc1.html"/>
<frameset rows="50%,50%">
<frame src="doc2.html" name="frame2"/>
<frame src="doc3.html" name="frame3"/>
</frameset>
</frameset>
</html>

```



- In questo modo si ottiene una finestra suddivisa in due colonne, una larga 250 pixel, l'altra la restante parte dello spazio, a sua volta suddivisa in due righe di uguale grandezza.

22

HTML - Frame

- L'attributo **'name'** del tag **<frame>** serve ad assegnargli un nome; questo è utile nel caso si voglia fare in modo che i collegamenti attivati su un riquadro abbiano "effetto" su un altro.
- Ad esempio possiamo avere in 'doc1.html' il collegamento seguente in cui si fa uso dell'ulteriore attributo **'target'**:
- `Cliccare per vedere documento 4`
- Se viene attivato questo collegamento, il riquadro che lo contiene non si altera mentre in quello chiamato "frame2" si ha la visualizzazione della pagina 'doc4.html'
- Nel caso in un documento ci siano molti collegamenti tutti con stesso **'target'** si può evitare di scrivere l'attributo per ognuno di essi ricorrendo all'elemento **<base>** da inserire nella sezione **'head'** nel seguente modo

```
<head>
<title> Titolo </title>
<base target="nome_frame">
</head>
```

23

HTML - Frame

- Esistono quattro nomi di destinazioni "speciali" che è possibile assegnare all'attributo **'target'** e precisamente:
- **"_blank"**: indica di caricare il documento aprendo una nuova finestra del navigatore;
- **"_parent"**: fa riferimento al **'frameset'** di appartenenza (può essere utile se si usano riquadri nidificati);
- **"_self"**: indica di caricare il documento nel riquadro corrente (questa è l'impostazione per difetto e quindi ha senso usarla solo nel caso si voglia cambiare la destinazione stabilita con un **'base target'** precedente);
- **"_top"** è forse l'opzione più utile e permette di rimuovere tutti i riquadri presenti nella finestra.

24



HTML - Frame

- Altri attributi del tag `<frame>` sono:
 - `frameborder`, con valori "0" oppure "1" (per difetto è "1"): con "0" si elimina il bordo di separazione tra un riquadro e quello adiacente; si possono usare anche i valori "no" per avere un bordo normale e "yes" per averlo tridimensionale e si può anche impostare lo spessore del bordo con l'attributo `border` del tag `<frameset>`; questi attributi comunque sono estensioni HTML fuori dallo standard;
 - `borderwidth`, `borderheight`, si possono usare per stabilire quanto spazio in *pixel* deve essere inserito tra i bordi del riquadro (rispettivamente i bordi destro e sinistro oppure superiore e inferiore) ed il suo contenuto;
 - `noresize`, permette di impedire il ridimensionamento del riquadro da parte dell'utente;
 - `scrolling`, con valori "yes" oppure "no" oppure "auto"; il valore per difetto è "auto" ed indica che le barre di scorrimento vengono aggiunte al riquadro automaticamente quando necessario; il valore "yes" le fa inserire sempre ed il valore "no" mai (sconsigliato)

25



HTML - Frame

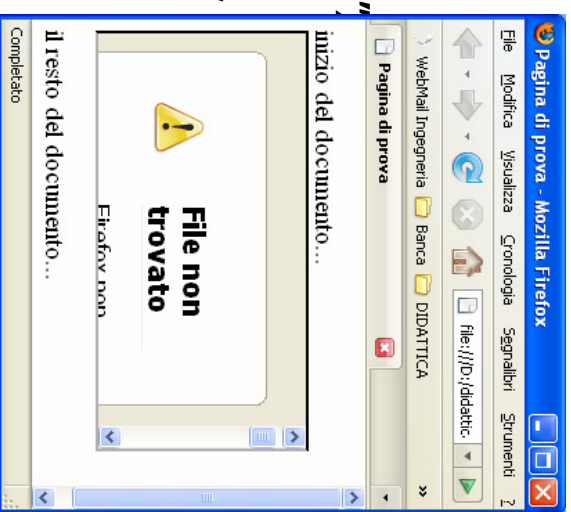
- I **frame in linea** sono elementi di testo che permettono di visualizzare un documento separato come parte di una pagina; i marcatori da usare sono `<iframe>` e `</iframe>` con gli stessi attributi del tag `<frame>` ed in più `width`, `height` e `align`, con ovvio significato.
- Un riquadro in linea dovrebbe sempre contenere anche del testo alternativo da visualizzare se il browser non supporta tale elemento.
- Anche ai riquadri in linea può essere assegnato un nome, esattamente allo stesso scopo visto per i riquadri: riferendosi all'esempio, è possibile definire in un'altra pagina un collegamento che, grazie all'attributo `target` con valore "riquadro", fa aprire un documento all'interno del riquadro in linea
 - essendo in linea, questi frame devono essere utilizzati all'interno di un body

26

HTML - Frame

Esempio di Frame inline

```
<body>  
<div align="left">  
<p>inizio del documento...  
<iframe src="doc2.html" name="frame">  
<p> Testo alternativo.</p>  
<p><a href="doc1.html">  
Cliccare qui per il documento</a></p>  
</iframe>  
<p>  
il resto del documento...  
</div>  
</body>
```



27

HTML - Form

- Tra i documenti del Web esiste una importante suddivisione: ci sono pagine «statiche» e «dinamiche»
- Un documento è statico quando viene definito una volta per tutte e messo a disposizione degli utenti in una forma che non concede loro la possibilità di alterazioni; il contenuto e l'aspetto della pagina possono essere variati solo dalle persone responsabili del sito di cui essa fa parte.
- Un documento è invece dinamico quando viene creato, tutto o in parte, in seguito ad operazioni interattive svolte dall'utente. Un esempio classico è quello di una persona che richiede sul sito Internet di una azienda informazioni su un prodotto da essa commercializzato inserendo i parametri di ricerca in una pagina definita a tale scopo; i dati vengono inviati al server Web dell'azienda che interroga la base di dati dei prodotti e costruisce «al volo» la pagina HTML di risposta da inviare al *browser* dell'utente.

28

HTML - Form

- La tecnologia che per prima ha permesso la realizzazione di documenti Web dinamici è stata la CGI (*Common Gateway Interface*); si tratta di un protocollo standard che descrive come un server Web possa invocare programmi esterni per elaborare dati ricevuti dall'utente ed inserirli in appositi moduli definiti in pagine HTML. Ulteriori possibilità di creazione di pagine Web dinamiche sono offerte dalle «applet» 'Java', dagli «script attivi» scritti con linguaggi di *scripting* come 'JavaScript' dal linguaggio 'PHP'. Vediamo una prima classificazione di questi strumenti in base alle loro caratteristiche:
 - una applicazione CGI può essere scritta con uno qualsiasi dei linguaggi di programmazione esistenti, anche quelli più «tradizionali» come 'C', o 'Fortran', e «lato server» (*server-side*) in quanto viene eseguita sulla macchina in cui risiede il server Web ed è «esterna», cioè fisicamente scritta in un file separato da quello contenente il codice della pagina HTML;
 - le applet e gli script attivi sono «lato cliente» (*client-side*) cioè eseguiti sulla macchina in cui è in esecuzione il programma di navigazione; le prime sono esterne i secondi invece sono «incorporati» (*embedded*) in quanto il loro codice è scritto all'interno dello stesso file che contiene il sorgente HTML;
 - gli script PHP, infine, sono lato server e incorporati.
 - Anche il linguaggio HTML mette a disposizione degli strumenti utili alla definizione di documenti dinamici: i *form* o moduli, grazie ai quali possono essere raccolti i dati immessi dagli utenti. 29

HTML - Form

- I moduli si definiscono in un documento HTML con il tag '<form>', ma la loro gestione, o meglio, la gestione dei dati in essi contenuti, non avviene con gli strumenti dell'HTML ma grazie a dei programmi esterni chiamati programmi o script CGI. Occorre quindi esaminare due aspetti distinti:
 - definizione in HTML degli elementi che compongono il modulo;
 - scrittura, in linguaggio di programmazione, del programma che deve gestire i dati immessi con il modulo.
- Più in dettaglio, la gestione di un modulo passa attraverso le seguenti fasi:
 - inserimento nella pagina HTML dei tag per la definizione del modulo;
 - indicazione, tramite un attributo dell'elemento '<form>', del nome del programma esterno che deve elaborare i dati;
 - attivazione, da parte del server Web, del programma indicato; ad esso vengono forniti i dati ricevuti dal modulo ed avviene l'elaborazione dei dati;
 - restituzione dei risultati al programma di navigazione, solitamente tramite la creazione di una nuova pagina Web da visualizzare.
- I CGI si differenziano in due categorie in base al linguaggio di programmazione utilizzato per realizzarli:
 - programmi CGI, se si usano linguaggi compilati come 'C', 'C++', 'Java';
 - script CGI, se si usano linguaggi interpretati come 'Perl', 'shell' di Unix, 'Python',

HTML - Form

- I moduli iniziano con il tag `<form>` e terminano con `</form>`; si tratta di un elemento di blocco che può contenere qualsiasi altro elemento ma non altri moduli; i suoi attributi sono:
 - `action`, per specificare l'URI a cui inviare i dati del form;
 - `method`, per specificare l'azione da svolgere sui dati; i valori possibili sono "get" o "post";
 - `enctype`, per indicare il tipo di MIME usato per il trasferimento delle informazioni;
 - `target`, per definire una nuova finestra dove visualizzare la risposta elaborata dal CGI.
 - Esistono poi i cosiddetti attributi di evento come `onmouseover`, `onclick`, `onsubmit`, `onreset` che però vengono solitamente gestiti con gli script attivi (JavaScript)
 - Gli attributi più importanti sono `action` e `method`.
 - Il valore di `action` corrisponde all'indirizzo a cui inviare i dati; può essere o un indirizzo di posta elettronica o il nome del programma CGI (completo di percorso) che dovrà ricevere ed elaborare in qualche modo i dati del form.
 - Il valore di `method` può essere "get" oppure "post"; nel primo caso i dati vengono accordati all'URI indicato in `action`, nel secondo caso i dati vengono passati attraverso lo «standard input» al servernte.

31

HTML - Form

- Riguardo all'attributo `enctype` è opportuno soffermarsi brevemente sui tipi MIME (*Multipurpose Internet Mail Extension*) creati originariamente per descrivere gli allegati di posta elettronica e ora usati, più generalmente, per riconoscere la natura dei file presenti sul Web.
- Sono composti da due parti: «tipo-principale/tipo-specifico», come ad esempio: *image/gif* (immagini di tipo gif), *text/plain* (testo puro) e molti altri.
- Elenchi completi di tipi MIME si trovano facilmente in Internet, ad esempio su <http://www.iana.org>
- Il tipo di MIME da usare per trasferire i dati dal modulo al servernte Web è *application/x-www-form-urlencoded* e deve essere specificato solo nel caso si usi il metodo "post"; questo valore è comunque quello attribuito per diretto all'attributo `enctype`.
- In pratica serve a comunicare al servernte che i dati saranno inviati attraverso lo standard input ma codificati alla stessa maniera di qundo sono inviati accordati all'URI (ad esempio, gli spazi sono convertiti in %20

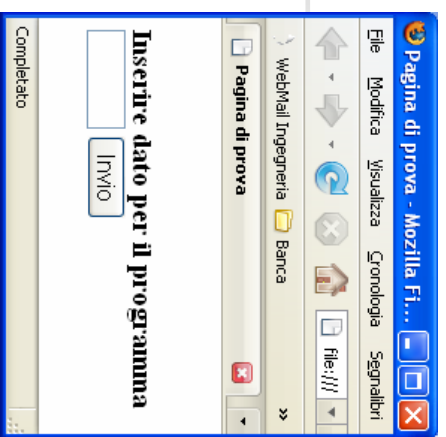
32

HTML - Form

- Gli elementi specifici usati all'interno dei moduli sono tag di testo chiamati «controlli»; i più importanti sono:
 - `<input>`, per creare vari tipi di input diversi;
 - `<select>`, per creare menu a scorrimento le cui opzioni sono indicate con il tag `<option>`;
 - `<textarea>`, per l'immissione di testo esteso;
 - `<fieldset>`, per raggruppare più controlli di un modulo.
- I primi tre prevedono l'attributo `'name'` che permette di indicare il nome della variabile o del campo che contraddistingue l'elemento in questione.
- Talvolta si utilizza anche l'attributo `'value'` per assegnare un valore alla variabile o al campo; più spesso però il valore assunto corrisponde a ciò che l'utente ha digitato o scelto in corrispondenza di quel campo.
- Il nome ed il valore dei campi sono naturalmente di fondamentale importanza per il programma CGI che deve elaborare i dati inviati dal modulo; in pratica siamo in presenza della distinzione tra nome e contenuto di una variabile.

33

HTML - Form



- Un primo esempio:


```
<form action="/cgi-bin/cgi-prova.sh" method="get">
</form>
<br/>
<b>Inserire dato per il programma&nbsp;  </b>
<input type="text" name="datoin" size="5" maxlength="3"/>
<input type="submit" value="Invio"/>
</form>
```
- In questo esempio si chiede l'inserimento di un dato che viene passato con metodo "get" al programma "cgi-prova.sh" (presumibilmente scritto con i comandi della shell di GNU/Linux o Unix, data l'estensione `.sh`) residente nella directory `/cgi-bin`.
- Il dato viene identificato come "datoin" dal programma, viene immesso in una casella lunga cinque caratteri ma può essere lungo al massimo tre caratteri.
- L'invio dei dati avviene premendo sul pulsante definito dal controllo "submit"; in questo esempio tale controllo non ha l'attributo `'name'` che può servire nel caso³⁴ si debbano distinguere più pulsanti di invio presenti in uno stesso modulo.

HTML - Form

- Tra gli attributi di **<input>** il più importante è **'type'**, il cui valore permette di definire vari tipi di controlli:
- "submit", crea un pulsante da usare per l'invio dei dati;
- "reset", crea un pulsante per la reinizializzazione dei campi del modulo;
- "text", crea un campo di testo di una sola riga;
- "radio", crea un pulsante di opzioni;
- "checkbox", crea una casella di selezione
- "image", come "submit" ma con una immagine al posto del pulsante;
- "hidden", crea un elemento che non appare nel modulo ma ha un nome e un valore;
- "password", crea un campo per l'immissione di *password* in cui i dati immessi sono visualizzati come asterischi;
- "button", crea un pulsante generico definito e gestito dal programmatore;
- "file", usato per inviare un file insieme ai dati del modulo.

35

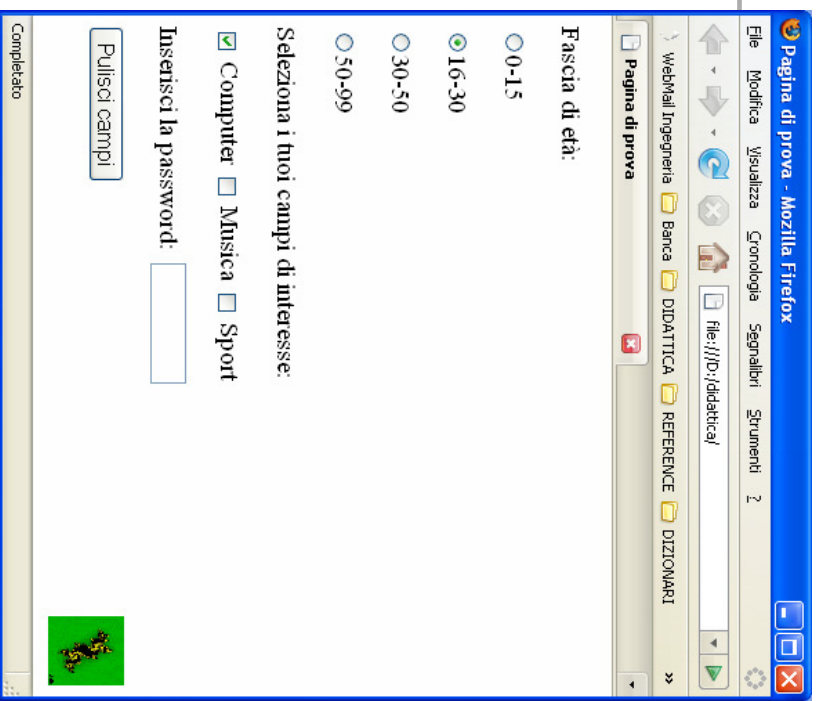
HTML - Form

```
<form action="/cgi-bin/cgi-prova.sh" method="get">
<input type="hidden" name="nascosto" value="aaa"/>
<p> Fascia di età:</p>
<p><input type="radio" name="fascia" value="0-15"/>0-15</p>
<p><input type="radio" name="fascia" value="16-30" checked/>16-30</p>
<p><input type="radio" name="fascia" value="30-50"/>30-50</p>
<p><input type="radio" name="fascia" value="50-99"/>50-99</p>
<p/>Seleziona i tuoi campi di interesse:<p/>
<input type="checkbox" name="int" value="computer" checked/>&nbsp;Computer
<input type="checkbox" name="int" value="musica"/>&nbsp;Musica
<input type="checkbox" name="int" value="sport"/>&nbsp;Sport
<p/>Inserisci la password:&nbsp;
<input type="password" name="psw" size="8" maxlength="8"/>
<p/>
<input type="image" src="julia.gif" width="50" height="55"
title=""Premi qui per inviare i dati" alt="Invio" align="right"/></p>
<p><input type="reset" value="Pulisci campi"/></p>
</form>
```

36

HTML - Form

- Risultato della form:
- Alcuni degli attributi utilizzati in questo esempio sono di immediata comprensione, anche perché già visti in precedenza (come `src`, `alt`, `align`).
- L'attributo `checked` viene usato in caso di controlli che prevedono più alternative per preimpostarne una o alcune.
- I controlli "hidden" possono essere utili nel caso si debbano inviare dei dati fissi al programma CGI senza che l'utente possa vederli e tantomeno variarli.
- L'invio dei dati del modulo avviene cliccando sull'immagine che appare in fondo a destra.

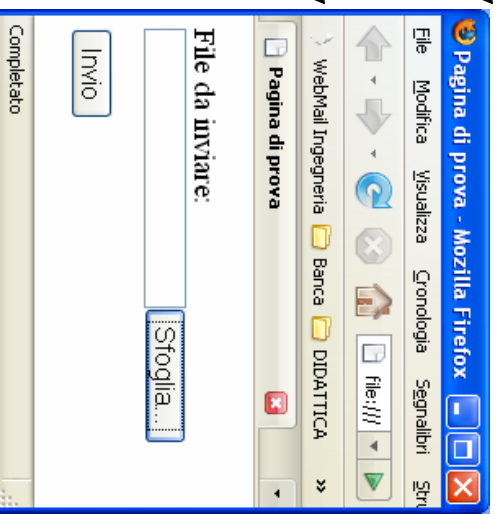


HTML - Form

- Upload di file:
- ```

<form enctype="multipart/form-data"
action="/cgi-bin/cgi-prova.sh">
<p> File da inviare: </p>
<input type="file" name="file"/> <p/>
<input type="submit" size="5"
value="Invio"/>
</form>

```
- Il controllo "file" permette di inviare un file al programma CGI; presenta un bottone "Browse" o "Sfoglia", inserito automaticamente, per permettere la ricerca del file nell'ambito della macchina locale.
  - Si noti che in questo contesto si deve valorizzare l'attributo `enctype` con "multipart/formdata".



# HTML - Form

- Con il tag `<select>` si crea un menu a cascata le cui opzioni sono indicate ognuna con un tag `<option>` e la cui definizione si conclude con `</select>`.
- Il tag `<select>` prevede obbligatoriamente l'attributo `name`; altri attributi utilizzabili in questo contesto sono:
  - `size` (attributo di `select`), per indicare il numero di voci di menu visibili contemporaneamente;
  - `multiple` (attributo di `select`), per dare la possibilità all'utente di selezionare più di una voce;
  - `selected` (attributo di `option`), per indicare una voce selezionata per difetto;
  - `value` (attributo di `option`), per indicare un testo, associato alla scelta effettuata, da inviare al programma CGI; in assenza di tale attributo, il testo inviato coincide con il nome dell'opzione
- Con il tag `<textarea>` si può creare una casella di testo contenente più righe.
- Si deve chiudere con `</textarea>` ed è obbligatorio l'attributo `name`. Ci sono anche gli attributi `rows` e `cols` con i quali si indica l'ampiezza della casella di testo senza però limitare la quantità di caratteri inseribili; quasi tutti i *browser* infatti inseriscono le barre di scorrimento a sinistra ed in basso in caso la quantità di caratteri immessi superi la dimensione della casella.
- Si può inoltre definire un testo pre-inserito scrivendolo tra i tag `<textarea>` e `</textarea>`.

39

# HTML - Form

```
<form action="/cgi-bin/cgi-prova.sh"
method="post">
```

Componenti da acquistare: <p>

```
<select name="comp" multiple size="6">
```

```
<option selected>Main board</option>
```

```
<option selected>Cpu</option>
```

```
<option selected>Ram</option>
```

```
<option>Floppy disk 120 Mb</option>
```

```
<option>Floppy disk 1,44 Mb</option>
```

```
<option>Hard disk IDE</option>
```

```
<option>Hard disk SCSI</option>
```

```
<option>Scheda video</option>
```

```
<option>Scheda audio</option></select>
```

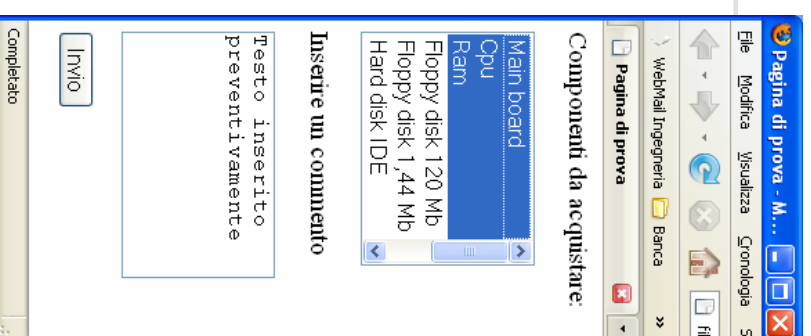
```
<p/>Inserire un commento<p/>
```

```
<textarea name="commento" rows="5" cols="15">
```

Testo inserito preventivamente

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

```
<input type="submit" value="Invio"/></form>
```



40



## HTML - Form

- Con l'elemento `<fieldset>` si possono raggruppare, incorniciandoli automaticamente, più controlli di un modulo, tutti quelli presenti fino al tag di chiusura `</fieldset>`; all'interno si può usare il tag `<legend>` per aggiungere una legenda posizionandola con l'attributo `align`.
- Resta infine da segnalare l'attributo `'tabindex'`, utilizzabile per tutti i controlli di un modulo, con il quale si può cambiare l'ordine di selezione degli stessi assegnadogli un valore intero. Valori alti hanno la precedenza rispetto a quelli bassi o negativi; in caso di assenza di questo attributo, l'ordine di selezione dei controlli corrisponde a quello di inserimento all'interno del modulo: dall'alto in basso e da sinistra verso destra.

41



## HTML - Mappe

- In un documento può essere utile rendere **active** delle porzioni di immagini, ossia realizzare con delle parti di immagini dei link ipertestuali. Esistono due tipi di image maps:
  - **client-side** (interpretate dal browser)
  - **server-side** (richiedono che l'utente abbia accesso al server e sono utili per browser di vecchio tipo)
- Per creare una mappa è necessario avere:
  - l'immagine
  - una specifica di quali parti sono attive e di quali azioni devono essere svolte quando si clicca su di esse
- L'immagine viene caricata sempre con il tag `<IMG>` ma usando gli attributi **ISMAP(server)** o **USEMAP(client)**

# HTML - Mappe

- Per realizzare mappe client-side:
- `<IMG src="pathname" USEMAP="specifica mappatura">`
- Per scrivere la specifica della mappatura si può :
- Creare un file con estensione .map che contiene la mappatura dell'immagine e ha una struttura come quella seguente
  - **<MAP** name="mappa"
  - comandi di mappatura
  - **</MAP>**
  - e scrivere nel file .html le istruzioni `<IMG src="pathname" USEMAP="/><path>/mappa.map">`
- Scrivere la specifica della mappatura al fondo del documento, alla fine del body, e usare una label per far riferimento alla mappatura `<IMG src="pathname" USEMAP="#mappa">`
- il nome del file .map e il nome dentro il comando `<MAP>` devono coincidere

43

# HTML - Mappe

Comandi di mappatura: `<AREA>`

- Si specificano delle aree dell'immagine e ad ogni area si associa la URL corrispondente. Sono presenti 3 attributi:

```
<AREA
```

```
 shape="rect" | "circle" | "polygon" | "default"
```

```
 {forma dell'area; default indica l'azione associata all'immagine per le aree non coperte dalla mappatura}
```

```
 coord="x1,y1,x2,y2" |
```

```
 {per i rect, in alto a sin e in basso a destra}
```

```
 "x, y, r" |
```

```
 {per i circle, coordinate centro e raggio}
```

```
 "x1,y1, x2,y2, x3,y3, ... "
```

```
 {coordinate dei vertici}
```

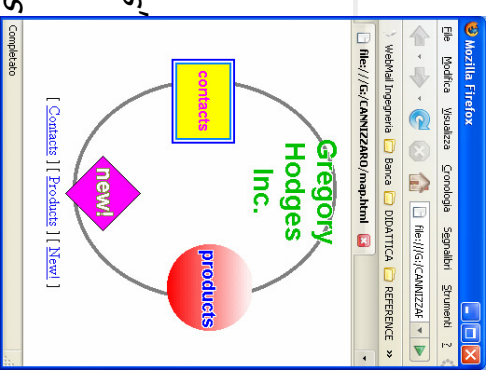
```
 href="URL" {URL associata all'area}
```

```
 nohref {per aree non attive}
```

```
>
```

44

# HTML - Mappe



- **Esempio:**

```
<MAP NAME="map1">
<AREA
 HREF="contacts.html" ALT="Contacts" TITLE="Contacts"
 SHAPE=RECT COORDS="6,116,97,184">
<AREA
 HREF="products.html" ALT="Products" TITLE="Products"
 SHAPE=CIRCLE COORDS="251,143,47">
<AREA
 HREF="new.html" ALT="New!" TITLE="New!"
 SHAPE=POLY COORDS="150,217, 190,257, 150,297,110,257">
</MAP>
<IMG SRC="testmap.gif"
 ALT="map of GH site" BORDER=0 WIDTH=300 HEIGHT=300
 USEMAP="#map1">

```

# HTML - Mappe

- **Server-side image maps**
- L'immagine viene caricata con l'opzione ISMAP
- <IMG src="pathname" .... ISMAP>
- L'immagine deve essere all'interno di un comando <A> che contiene l'indirizzo del file della image map
- <A href="pathname image map (.map)">
- <IMG src="pathname" .... ISMAP>
- </A>

- Il file .map come quello del caso precedente ma senza i comandi AREA. E costituito da un insieme di righe del tipo:  
*shape URL coordinate*
- E' meglio usare sempre mappe di tipo clientside con mappatura in fondo al file HTML in quanto oggi tutti i browser sono in grado di interpretarle

# HTML - Oggetti multimediali

- Per inserire oggetti multimediali (suoni e filmati) in un documento HTML si usa principalmente l'elemento di testo `<object>`. Grazie alla tecnica dello *streaming* è anche possibile ascoltare suoni e vedere filmati in tempo reale (cioè durante il trasferimento, senza dover attendere che il file relativo all'oggetto multimediale sia stato completamente scaricato dalla rete) ma sono necessari appositi programmi come **'RealPlayer'**. I formati più comuni:
  - .au: Unix audio;
  - .wav: Microsoft Wave;
  - .aif: Macintosh audio;
  - .ra o .ram: RealAudio;
  - .mid o .midi: MIDI (*Music Instrument Digital Interface*);
  - .mp3: standard MPEG-1 audio layer III.
- I primi quattro contengono suoni digitalizzati con una qualità non eccezionale. I file MIDI invece contengono "istruzioni musicali" che vengono eseguite dalla scheda audio; sono piccoli e offrono una qualità che dipende dalla scheda
- Passando ai formati video possiamo citare:
  - .mpg o .mpeg: MPEG (*Moving Pictures Expert Group*);
  - .avi: Microsoft AVI (*Audio Video Interleave*);
  - .mov o .qt: QuickTime.
- Tutti i tipi di file video supportano anche l'audio associato ma occupano una grande quantità di memoria (almeno 1 Mb per 30 secondi di filmato). I migliori sono i filmati MPEG che offrono anche una buona qualità di riproduzione.

# HTML - Oggetti multimediali

- L'elemento `<object>` prevede gli attributi `'data'`, `'type'` e molti degli attributi già visti per le immagini (per definire larghezza, altezza, allineamento, bordo, ecc.).
- Non è invece previsto l'attributo `'alt'` in quanto il testo alternativo si inserisce semplicemente tra i tag `<object>` e `</object>`.
- Il valore di `type` permette una specifica esplicita del tipo di oggetto, in modo che il browser sappia quale applicativo invocare per visualizzare l'oggetto; se manca, il browser utilizzerà il valore presente nell'HTTP content-type e specificato dal server da cui l'oggetto viene scaricato. In ogni caso, il valore è uno fra quelli definiti dallo standard MIME (Multipurpose Internet Mail Extension)
  - `<object data="symphony.mid" type="audio/midi">`
  - `Sinfonia in formato midi</object>`
  - `<object data="meteo.mov" type="video/quicktime" align="left">`
  - `Filmato da satellite meteo</object>`
  - `<object data="symphony.mid" type="audio/midi">`
  - `<param name="autostart" value="true">`
  - `Sinfonia in formato midi con autostart </object>`



# HTML - Tool

- Siti con riferimenti ai numerosissimi editor HTML
- <http://download.html.it/categorie/start/113/windows/editor-html-visuali/>
  - freeware indica un software che viene distribuito in modo gratuito, mentre I programmi shareware, nonostante siano distribuiti in maniera simile a quelli freeware, richiedono un pagamento al termine di un periodo di prova o per attivarne tutte le funzionalità.
  - [WYSIWYG Web Builder 5.0.3](#)
  - Shareware (16 Gennaio 2008) a pagamento ma molto recente
  - [Koala edit 6.2](#)
  - Freeware (31 Ottobre 2007) recente
  - [Amaya 9.55](#)
  - Open Source (26 Settembre 2007)
  - [Blue Voda Website Builder 7.6](#)
  - Freeware (13 Dicembre 2005), vecchio ma molto diffuso
  - [Dreamweaver CS3](#)
  - Shareware (21 Dicembre 2007), a pagamento ma molto diffuso
  - [WebSite X5](#)
  - Shareware (18 Maggio 2007), a pagamento ma molto diffuso
  - [Visual Web Developer 2008 Express Edition](#)
  - Freeware (08 Febbraio 2008)

49

# HTML - Tool

- <http://download.html.it/categorie/start/114/windows/editor-html-testuali>
  - [Ultraedit-32 12.10](#)
  - Shareware (18 Luglio 2006), molto diffuso
  - [PSPad 4.5.3](#)
  - Freeware (14 Dicembre 2007)
  - [WebBuilder 2007 8.2](#)
  - Shareware (05 Dicembre 2007)
- [http://it.wikipedia.org/wiki/Lista di editor HTML](http://it.wikipedia.org/wiki/Lista_di_editor_HTML) altro elenco
- <http://www.w3.org/Tools/> elenco di tool (non solo editor)

50