

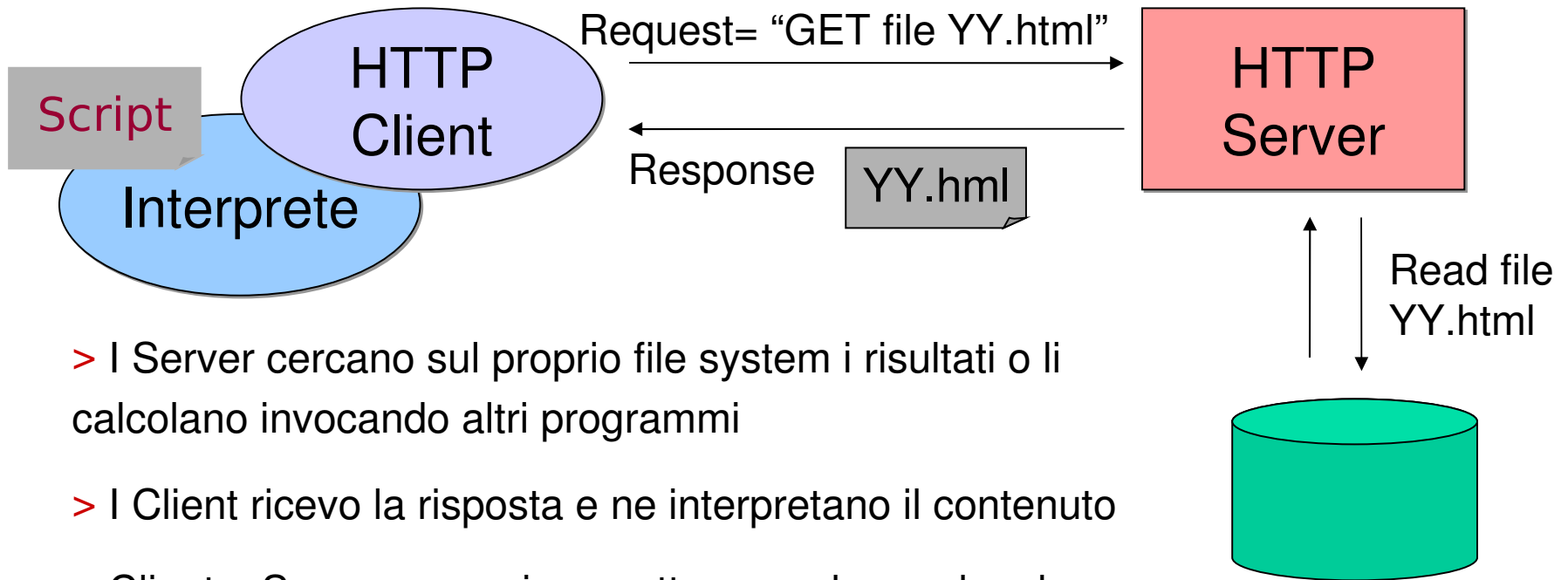
Anno Accademico 2007-2008

Corso di Tecnologie Web
HTML

<http://www-lia.deis.unibo.it/Courses/TecnologieWeb0708/>

HyperText Transfer Protocol

- > Le pagine web sono rese disponibili da computer che eseguono i Web Server
- > I computer connessi alla rete eseguono i Web Client (es: browser) per formulare richieste ai Web Server e visualizzarne le risposte



- > I Server cercano sul proprio file system i risultati o li calcolano invocando altri programmi
- > I Client ricevono la risposta e ne interpretano il contenuto
- > Client e Server comunicano attraverso le regole e le procedure che costituiscono il protocollo HTTP

`http://myserver/YY.html`



Il linguaggio HTML

> HTML

HyperText Markup Language

Possibili riferimenti e navigazione tra diversi documenti

- Iper testo richiama il concetto di (salto nel) iperspazio, per passare istantaneamente da un punto all'altro di uno o più documenti

> Sviluppato al CERN dal 1989

Tim Berners-Lee: dare ai ricercatori un modo per condividere informazioni e pubblicazioni all'interno della rete dell'istituto

Basato su SGML: standard internazionale per il markup di documenti ai fini della loro visualizzazione su terminali eterogenei

> Pubblicato nel 1991

Primo Browser: Mosaic

> Web rende i servizi esistenti obsoleti

WAIS, Hytelnet, Gopher

Il linguaggio HTML: Tag

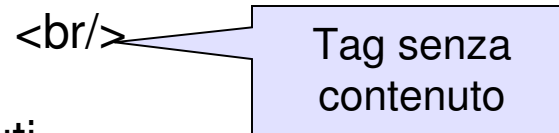
- > I tag HTML sono usati per definire il *mark-up* di **elementi** HTML.
- > I tag HTML sono preceduti e seguiti rispettivamente da due caratteri “<” e “>” (parentesi angolari).
- > I tag HTML sono normalmente accoppiati; un esempio è dato da: e , rispettivamente “start tag” e “end tag”. L’apertura e la chiusura di tutti i tag, anche quelli relativi ad elementi privi di contenuto, è una buona prassi di programmazione in quanto è un requisito XHTML (XHTML è uno specifico vocabolario XML e, come tale, rispettoso delle regole sintattiche XML). Inoltre, vi sono dei vincoli legislativi per tutti i siti della PA: questi siti devono rispettare XHTML Strict (per garantirne l’accessibilità).
- > Il testo tra start e end tag è detto *contenuto dell’elemento*.
- > I tag HTML non sono *case sensitive*.
- > Il W3C raccomanda di usare *lowercase* tag (HTML 4 recommendation). XHTML richiede espressamente il *lowercase*.

Il linguaggio HTML: Tag - Esempi

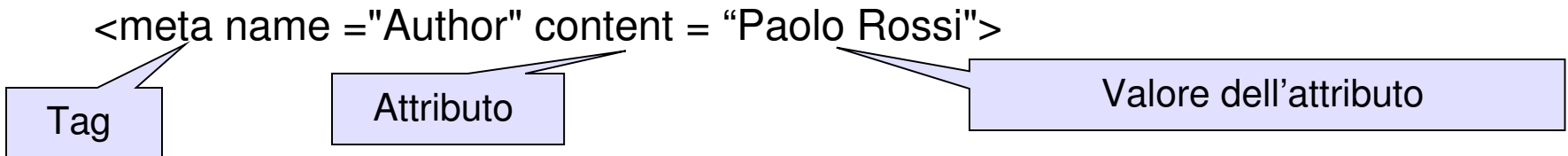
- > Linguaggio basato su marcatori che definiscono diverse proprietà del testo.



- > Marcatori di formattazione senza testo. Apertura e chiusura in forma compatta



- > Un tag può avere degli attributi



- > Gli attributi e i corrispondenti valori sono anch'essi *case insensitive* ma, in osservanza alle raccomandazioni del W3C, è preferibile usare il *lowercase*. Gli attributi vanno sempre racchiusi tra *quotes* " ", eventualmente singoli ' '.

Il linguaggio HTML: Struttura e presentazione

> Struttura

Marcare le parti di un documento per identificarne il tipo

- Paragrafi, liste, tabelle, titoli, didascalie, ...

> Presentazione

Definire come le parti di un documenti sono riprodotte da diversi dispositivi

- Schermi di computer, stampanti, ... → palmari e telefonini (oggi)!

> Perché questo?

Chi scrive non è, normalmente, anche un buon designer...

La manutenzione del contenuto di un documento deve poter prescindere dalla manutenzione della sua visualizzazione

I cambiamenti nello stile adottato per la presentazione devono avere il minimo impatto possibile sui documenti e poter influenzare immediatamente tutti i documenti interessatii

> Eppure...

Il linguaggio HTML: Evoluzione storica e concettuale

- > L' HTML non impose fin dall'inizio una distinzione così netta tra struttura e presentazione (e c'è ancora un sacco di vecchio HTML là fuori...)

Internet diventa una realtà commerciale: i Web author richiedono strumenti più potenti per rendere attraenti i propri lavori

Le compagnie che sviluppano i browser rispondono introducendo nuovi “elementi” del linguaggio, non standard, per ottenere vantaggio dal maggiore supporto per la presentazione

Il neonato W3C (metà anni '90) tenta una prima standardizzazione: entrano a far parte dello standard anche gli elementi presentation-oriented.

- > Prime versioni di HTML:

HTML 1.0 → niente sfondi, né immagini di sfondo, né controllo sui font; pagine in sola lettura: non previsto l'invio di dati da parte dell'utente!

HTML 2.0 → sfondi e tabelle; form: l'interazione Web diventa bidirezionale!

HTML 3.2 → standardizzazione dei vari elementi del linguaggio introdotti dai diversi browser: enorme dizionario di specifiche (non esiste un HTML 3.0 perché non vi fu mai accordo tra tutti!); primi fogli di stile (ma i browser di allora non li supportano in modo efficace)

Il linguaggio HTML: oggi

> HTML 3.2 rende impossibile la manutenzione dei documenti

Tag e attributi per il rendering intercalati nel contenuto del documento

Le modifiche relative alla presentazione obbligano ad intervenire in punti multipli

> HTML 4.0 e 4.01:

Ripensamento generale delle funzionalità del linguaggio

Deprecati quasi tutti i tag e gli attributi per il rendering

Informazioni di presentazione veicolate attraverso la definizione di stili da applicare ai documenti HTML veri e propri → gli stili possono essere cambiati e modificati a prescindere dal contenuto a cui si applicano

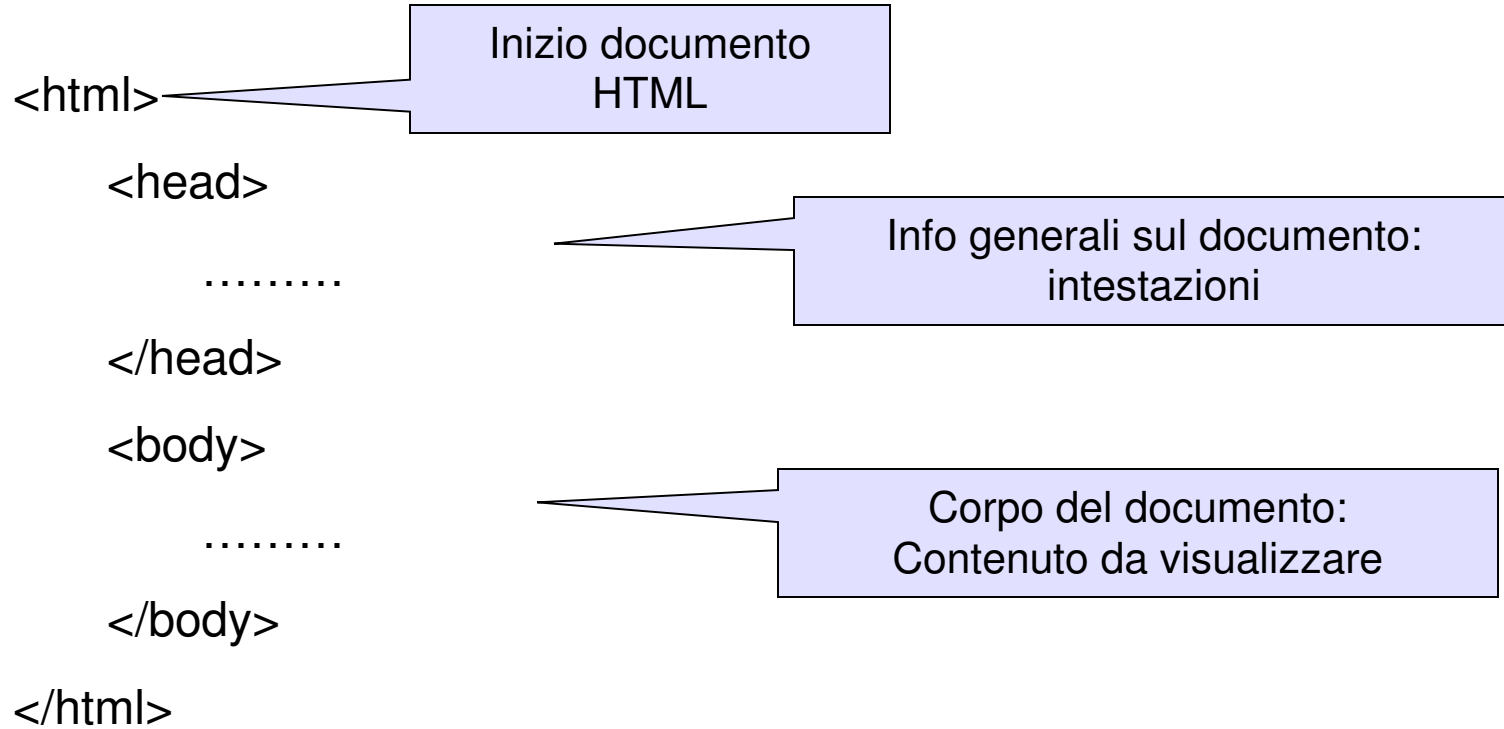
> XHTML 1.0:

Extensible HyperText Markup Language

Documenti ben formati e numero di tag disponibili molto ridotto

Obbligo di separazione tra struttura (contenuto, in pagine XHTML) e presentazione (stile, possibilmente in “fogli” CSS)

Il linguaggio HTML: Struttura del Documento

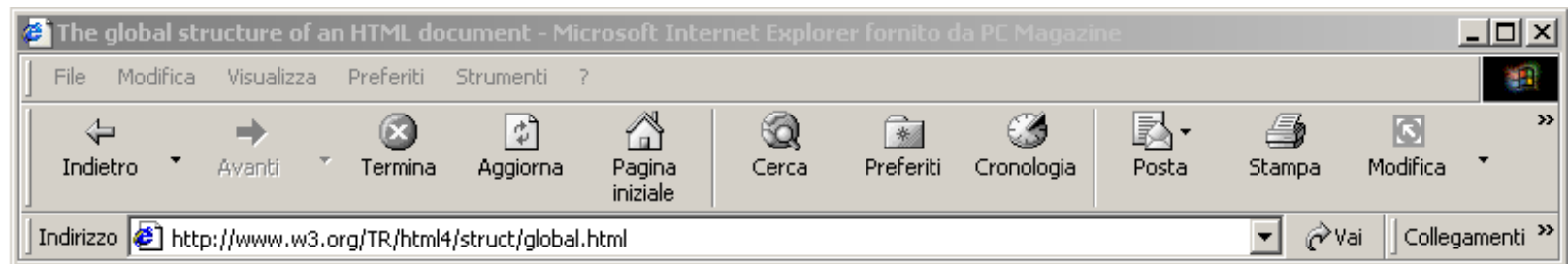


Il linguaggio HTML: Head

- > Il tag <head> è opzionale
- > Il tag <head> può contenere:

Il Tag <title>

<title>The global structure of an HTML Document</title>



Più Meta Tag <meta>

<meta name="Author" content="Paolo Rossi">

Il tag <base>

<base href="http://deis.unibo.it" /> utilizzato per risolvere i path di URI gerarchiche

Più tag <link>

<link rel="stylesheet" type="text/css" href="theme.css" /> utilizzato per collegare due documenti stabilendo una relazione tra di essi.

Il linguaggio HTML: Meta Tags

- > Definizione di coppie proprietà-valore:

`<meta name ="Author" content ="Paolo Rossi">`

- > Due tipi di utilizzo:

Definizione di una coppia nome valore generica

`<meta name="PropertyName" content="PropertyValue">`

Definizione di una coppia nome valore equivalente ad un header nell'HTTP Response

`<meta http-equiv="Expires" content ="Tue, 20 Aug 1996 14:25:27 GMT">`

Header equivalente → Expires: Tue, 20 Aug 1996 14:25:27 GMT

(RFC2616)

Il linguaggio HTML: Body

> Il tag <body> delimita il corpo del documento.

> Alcuni Attributi:

`background = uri`

Definisce l'URI di una immagine da usare come sfondo per la pagina.

`text = color`

Definisce il colore del testo.

`bgcolor= color`

In alternativa a background definisce il colore di sfondo della pagina

Il Colore: “#RRGGBB”



Red = “#FF0000”



Black = “#000000”



Blue = “#0000FF”



Yellow = “#FFFF00”

Il linguaggio HTML: Esempio

<html>

<head>

<title>A study of population dynamics</title>

</head>

<body bgcolor="white" text="black" link="red"

alink="fuchsia" vlink="maroon">

... *document body*...

</body>

</html>

Il linguaggio HTML: Heading

- > Creazione di un testo strutturato su più livelli:

`<h1>Titolo più significativo</h1>`

`<h2>Un po' meno significativo </h2>`

`<h3> Un po' meno significativo </h3>`

`<h4> Un po' meno significativo </h4>`

`<h5> Un po' meno significativo </h5>`

`<h6> Titolo meno significativo </h6>`

- > Attributi:

`align = left|center|right|justify`

Il linguaggio HTML: Testo Strutturato

> Elementi della frase:

`Testo da enfatizzare`

`Enfasi ancor più forte`

`<cite>Frase o citazione</cite>`

`<code>static void main(String[] args) { }</code>`

`a²+b₀` → a^2+b_0

Il linguaggio HTML: Font Style

> Formattazione del testo:

<code><tt>monospaced text</tt></code>	→	monospaced text
<code><i>italic text</i></code>	→	<i>italic text</i>
<code>bold text</code>	→	bold text
<code><big>big text</big></code>	→	big text
<code><small>small text</small></code>	→	small text
<code><u>underlined text</u></code>	→	<u>underlined text</u>

Il linguaggio HTML: Tag Font

> Il tag permette di formattare il testo. E' deprecato in HTML 4.01 e non supportato in XHTML 1.0 Strict DTD.

> Attributi:

size = [+/-]n

Definisce le dimensioni del testo (1-7 o relative)

color = color

Definisce il colore del testo

face = text

Definisce il font del testo

> Il tag <basefont> definisce le impostazioni di default di un documento.

Il linguaggio HTML: Horizontal Rule

> Il tag <hr> serve ad inserire una riga di separazione.

> Attributi:

`align = {left|center|right}`

Definisce l'allineamento della riga rispetto a ciò che la circonda

`size = pixels`

Definisce l'altezza della riga

`width = length`

definisce la larghezza della riga in modo assoluto o in percentuale delle dimensioni di ciò che la contiene

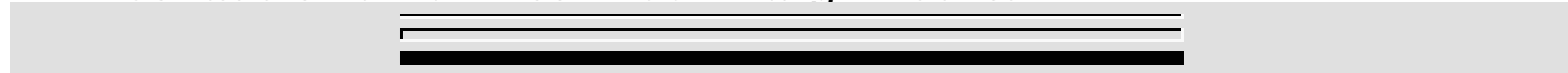
`noshade`

definisce se la riga deve essere “solida” o con un effetto di ombreggiatura.

```
<hr width="50%" align="center">
```

```
<hr size="5" width="50%" align="center">
```

```
<hr noshade size="5" width="50%" align="center">
```



Il linguaggio HTML: Paragrafi

> Il Tag <p> delimita un paragrafo:

<p>Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte;
 e il ponte, che ivi congiunge le due rive, par che renda ancor piú sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda ricomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni.</p>

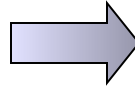
> Attributi del Tag <p>:

align = left|center|right|justify

Il linguaggio HTML: Liste

- Liste non ordinate: Type: disc,circle,square

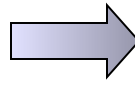
```
<ul type="disc">  
  <li>Unordered information.</li>  
  <li>Ordered information.</li>  
  <li>Definitions.</li>  
</ul>
```



- Unordered information.
- Ordered information.
- Definitions.

- Liste ordinate: Type: 1 (1,2,..), a (a,b,..), A (A,B,..), i (i,ii,..), I (I,II,..)

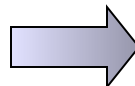
```
<ol type="I">  
  <li>Unordered information.</li>  
  <li value="4">Ordered information.</li>  
  <li>Definitions. </li>  
</ol>
```



- I. Unordered information.
- II. Ordered information.
- III. Definitions.

- Liste di definizione:

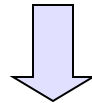
```
<dl>  
  <dt><strong>UL</strong></dt>  
  <dd>Unordered List.</dd>  
  <dt><strong>OL</strong></dt>  
  <dd>Ordered List.</dd>  
</dl>
```



- UL**
Unordered List.
- OL**
Ordered List.

Il linguaggio HTML: Tabelle

```
<table border="1" >
  <caption align="top"><em>A test table with merged cells</em></caption>
  <tr>
    <th rowspan="2"></th>
    <th colspan="2">Average</th>
    <th rowspan="2">Red<br/>eyes</th>
  </tr>
  <tr><th>height</th><th>weight</th></tr>
  <tr><th>Males</th><td>1.9</td><td>0.003</td><td>40%</td></tr>
  <tr><th>Females</th><td>1.7</td><td>0.002</td><td>43%</td></tr>
</table>
```



<i>A test table with merged cells</i>			
	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Il linguaggio HTML: <table>

> Tag che racchiude la tabella.

> Attributi:

`align = "{left|center|right}"`

allineamento della tabella rispetto alla pagina;

`width="n/n%"`

larghezza della tabella (anche in percentuale rispetto alla pagina);

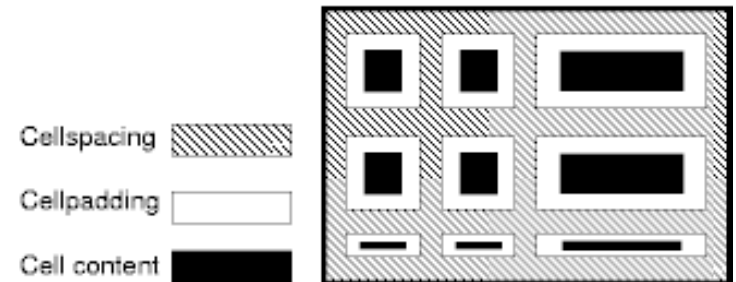
`bgcolor="#xxxxxx"`

colore di sfondo della tabella;

`border="n"`

spessore dei bordi della tabella (0 = tabella senza bordi);

`cellspacing, cellpadding`



Il linguaggio HTML: <tr> <th> <td>

> <tr> è il tag che racchiude ciascuna riga della tabella.

> Attributi:

align = "{left|center|right|justify}"

allineamento del contenuto delle celle della riga;

valign = "{top|middle|bottom|baseline}"

allineamento verticale del contenuto delle cella della riga;

bgcolor="#xxxxxx"

colore di sfondo della riga.

> <th> e <td> sono i tag che racchiudono la cella.

> Attributi:

Gli stessi di <tr>;

width,height = "*length/length%*"

specifica le dimensioni (larghezza e altezza) della cella,
dimensione assoluta (pixels) o valore percentuale;

rowspan,colspan = n

indica su quante righe,colonne della tabella si estende la cella.

Il linguaggio HTML: Tabelle - Esempio

**regali aziendali**vai a "regali personali"

Step 2: selezione dei prodotti

➔ Help

My Jakala

Il mio carrello 

Dati aziendali

Dati personali

Rubrica

I Miei Regali Preferiti

Catalogo Aziendale

Gestione Uffici

Gestione Utenti

Campagne

Gestione Ordini

Vuoi parlare con noi?
Jakala ti risponde in tempo reale!
 **Live Chat**

Le Campagne della tua azienda
Campagna Natale 2000

Le selezioni della campagna

➔ Catalogo Jakala

Oppure accedi alla selezione prodotti completa
➔ per Marca ➔ per Settore

➔ Torna scelta della modalita' di accesso

Vuoi avere dei consigli su come utilizzare il servizio Regali e Premi?
 **Consigli dell'Esperto**

I dati della Campagna:
Data inizio : 01/01/2001
Data fine : 06/09/2001
Data Consegna: 12/09/2001
Budget : € 70.000,00
Budget Ufficio: € 0,00
Budget Residuo: € 0,00

 **Contattaci**

Servizio Clienti: lun-ven 9-18 Tel. 02 89 255 555 - Fax 02 67 07 16 76
info@jakala.com

Italian Web Site
Copyright Jakala - Tutti i diritti riservati

➔ Home ➔ Usability Info ➔ Credits

Il linguaggio HTML: Tabelle - Esempio

		regali aziendali		vai a "regali personali"			
Step 2: selezione dei prodotti							
<div>➔ Help</div> <div>My Jakala</div> <div>Il mio carrello </div> <div>Dati aziendali</div> <div>Dati personali</div> <div>Rubrica</div> <div>I Miei Regali Preferiti</div> <div>Catalogo Aziendale</div> <div>Gestione Uffici</div> <div>Gestione Utenti</div> <div>Campagne</div> <div>Gestione Ordini</div> <div>Vuoi parlare con noi? Jakala ti risponde in tempo reale!  Live Chat</div>		<div>Le Campagne della tua azienda Campagna Natale 2000</div> <div>Le selezioni della campagna</div> <div>➔ Catalogo Jakala</div> <div>Oppure accedi alla selezione prodotti completa ➔ per Marca ➔ per Settore</div> <div>➔ Torna scelta della modalita' di accesso</div>		<div>I dati della Campagna:</div> <div>Data inizio : 01/01/2001</div> <div>Data fine : 06/09/2001</div> <div>Data Consegna: 12/09/2001</div> <div>Budget : € 70.000,00</div> <div>Budget Ufficio: € 0,00</div> <div>Budget Residuo: € 0,00</div>		<div>Vuoi avere dei consigli su come utilizzare il servizio Regali e Premi?</div> <div> Consigli dell'Esperto</div>	
 Contattaci		Servizio Clienti: lun-ven 9-18 Tel. 02 89 255 555 - Fax 02 67 07 16 76 info@jakala.com					
Italian Web Site Copyright Jakala - Tutti i diritti riservati		➔ Home ➔ Usability Info ➔ Credits					

Il linguaggio HTML: Tabelle - Esempio

 **regali aziendali** vai a "regali personali"

Step 2: selezione dei prodotti

➔ Help

My Jakala

Il mio carrello 

Dati aziendali

Dati personali

Rubrica

I Miei Regali Preferiti

Catalogo Aziendale

Gestione Uffici

Gestione Utenti

Campagne

Gestione Ordini

Vuoi parlare con noi?

Jakala ti risponde in tempo reale!

 **Live Chat**

Le Campagne della tua azienda

Campagna Natale 2000

Le selezioni della campagna

➔ Catalogo Jakala

Oppure accedi alla selezione prodotti completa

➔ per Marca ➔ per Settore

➔ Torna scelta della modalita' di accesso

Vuoi avere dei consigli su come utilizzare il servizio Regali e Premi?

 **Consigli dell'Esperto**

I dati della Campagna:

Data inizio : 01/01/2001

Data fine : 06/09/2001

Data Consegna: 12/09/2001

Budget : € 70.000,00

Budget Ufficio: € 0,00

Budget Residuo: € 0,00

 **Contattaci**

Servizio Clienti: lun-ven 9-18 Tel. 02 89 255 555 - Fax 02 67 07 16 76
info@jakala.com

Italian Web Site
Copyright Jakala - Tutti i diritti riservati

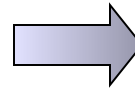
➔ Home ➔ Usability Info ➔ Credits

Il linguaggio HTML: div e span

- > I div sono raccomandati negli standard web come elementi per la presentazione dei dati. In altri termini, sostituiscono le tabelle nella definizione del layout di pagina. Si parla di principi di *grafica liquida* (<http://www.sitepoint.com/article/liquid-design/2>).
- > I div definiscono una sezione di un documento
- > I div vengono normalmente utilizzati per definire un blocco di elementi da formattare secondo determinati stili di presentazione.
- > **Note:** I browser usualmente inseriscono un line break prima e dopo il div

This is some text

```
<div id="me" style="color:#FF0000;">  
  <h4>This is a header in a div section</h4>  
  <p>This is a paragraph in a div section</p>  
</div>
```



This is some text

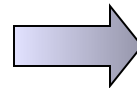
This is a header in a div section

This is a paragraph in a div section

Il linguaggio HTML: div e span

- > il tag `` è usato per raggruppare elementi *inline* in un documento
- > Differentemente dai `div` non definiscono dei blocchi
- > Consentono, come i `div`, di formattare secondo determinati stili di presentazione gli elementi *inline* raggruppati.

```
<p>This is a paragraph  
  <span style="color:#0000FF;">  
    This is a paragraph  
  </span>  
This is a paragraph  
</p>  
<p>  
  <span style="color:#00DD45;">  
    This is another paragraph  
  </span>  
</p>
```



This is a paragraph This is a paragraph
This is a paragraph
This is another paragraph

Il linguaggio HTML: Link e Anchor

- > Il Link è il costrutto di base dell'ipertesto.
- > Il Link è una connessione fra una risorsa Web ed un'altra.
- > Concetto semplice, ma chiave per l'affermazione dell'HTML
- > Link → due estremi detti anchors e una direzione
- > source anchor → destination anchors
- > Il destination anchors può essere una qualsiasi risorsa web
(un immagine, un video, un eseguibile, un documento HTML)
- > La risorsa di destinazione si ottiene visitando il link

Il linguaggio HTML: Link - Esempio

```
<body>
  ...some text...
  <p>
    You'll find a lot more in
    <a href="chapter2.html">chapter two</a>.
    See also this
    <a href = "../images/forest.gif">
      map of the enchanted forest.
    </a>
  </p>
</body>
```

Il linguaggio HTML: Link – Destination Anchors

.....

```
<h1>Table of Contents</h1>
```

```
<p>
```

```
  <a href="#section1">Introduction</a><br/>
```

```
  <a href="#section2">Some background</a><br/>
```

```
  <a href="#section2.1">On a more personal note</a><br/>
```

```
  ...the rest of the table of contents... ...the document body...
```

```
</p>
```

```
<h2><a name="section1">Introduction</a></h2>
```

```
  ...section 1...
```

```
<h2><a name="section2">Some background</a></h2>
```

```
  ...section 2...
```

```
<h3><a name="section2.1">On a more personal note</a></h3>
```

```
  ...section 2.1...
```

.....

Il linguaggio HTML: Link – Destination Anchors

.....

```
<h1>Table of Contents</h1>
```

```
<p>
```

```
  <a href="#section1">Introduction</a><br/>
```

```
  <a href="#section2">Some background</a><br/>
```

```
  <a href="#section2.1">On a more personal note</a><br/>
```

```
    ...the rest of the table of contents... ...the document body...
```

```
</p>
```

```
<h2 id="section1">Introduction</h2>
```

```
  ...section 1...
```

```
<h2 id="section2">Some background</h2>
```

```
  ...section 2...
```

```
<h3 id="section2.1">On a more personal note</h3>
```

```
  ...section 2.1...
```

.....

Il linguaggio HTML: Immagini

> Il tag consente di inserire immagini in un documento HTML

``

> Attributi:

`src = uri`

specifica l'indirizzo dell'immagine (required)

`alt = text`

specifica un testo alternativo nel caso fosse impossibile visualizzare l'immagine

`align = {bottom|middle|top|left|right}` (deprecato in HTML 4)

definisce la posizione dell'immagine rispetto al testo che la circonda

`width,height = length (pixels)`

definisce larghezza e altezza dell'immagine

`border = pixels` (deprecato in HTML 4)

definisce lo spessore del bordo dell'immagine (0 = nessun bordo)

`hspace,vspace = pixels` (deprecato in HTML 4)

definisce lo spazio in orizzontale e in verticale fra l'immagine e ciò che la circonda.

Uniform Resource Identifier: definizione

- > Il riferimento a una URI in forma di stringa presenta la seguente sintassi:

`[scheme:]scheme-specific-part[#fragment]`

- > Una URI che specifica uno schema è detta **assoluta**; in caso contrario **relativa**
- > Le URI sono anche classificate come **opache** o **gerarchiche**:

opaca: una URI assoluta in cui scheme-specific-part **non inizia** con “/”. Non è soggetta ad ulteriori operazioni di parsing. Esempi:

mailto:paolo.rossi@deis.unibo.it

urn:isbn:096139210x

gerarchica: una URI assoluta il cui scheme-specific-part **inizia** con “/”, oppure una URI relativa (cioè, in altri termini, che non specifica lo schema). Esempi:

http://deis.unibo.it/

docs/guide/collections/designfaq.html#28

../../../../lab/examples/ant/build.xml

file:///~/calendar

Uniform Resource Identifier: definizione (2)

> Una URI gerarchica è sottoposta ad un ulteriore parsing secondo la sintassi:

```
[scheme:][//authority]path[?query][#fragment]
```

- La componente **authority**, se specificata, può essere **server-based** o **registry-based**.
 - Una *authority server-based* subisce il seguente processo di parsing:
[user-info@]host[:port]
 - Una *authority non server-based* è detta *registry-based*.
- La componente **path** di una URI gerarchica è detta assoluta se inizia con “/”, relativa in caso contrario.
 - Il *path* di una URI gerarchica assoluta o che specifica una *authority* è sempre assoluto.

Uniform Resource Identifier: operazioni su un'istanza URI

- > **Normalizzazione** è il processo di rimozione dei segmenti "." e ".." dal *path* di una URI gerarchica. La normalizzazione su una URI opaca non ha effetto.
- > **Risoluzione** è il processo che a partire da una URI originaria porta all'ottenimento di una URI risultante. La URI originaria viene *risolta* basandosi su una terza URI, detta *base URI*. Specifiche definite da RFC 2396. Esempi:

<http://deis.unibo.it/docs/guide/collections/designfaq.html#28>

Original URI: docs/guide/collections/designfaq.html#28

Base URI: <http://deis.unibo.it/>

<http://deis.unibo.it/lab/examples/ant/build.xml>

Original URI: ../../../../lab/examples/ant/build.xml

Base URI: <http://deis.unibo.it/docs/guide/collections/designfaq.html#28>

[lab/examples/ant/build.xml](http://deis.unibo.it/lab/examples/ant/build.xml)

Original URI: ../../../../lab/examples/ant/build.xml

Base URI: docs/guide/collections/designfaq.html#28

Uniform Resource Identifier: operazioni su un'istanza URI

(2)

> **Relativizzazione** è il processo inverso alla risoluzione. Per ogni 2 URI normalizzate u e v vale:

$$1) \quad u.\text{relativize}(u.\text{resolve}(v)).\text{equals}(v)$$

$$2) \quad u.\text{resolve}(u.\text{relativize}(v)).\text{equals}(v)$$

Il linguaggio HTML: Client Side Maps

- > Le Mappe Client Side consentono di assegnare funzioni diverse a diverse regioni di una immagine
- > Esempio:

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

...the rest of the page here...

```
<map name="map1">  
  <area href="guide.html" alt="Access Guide"  
    shape="rect" coords="0,0,118,28"/>  
  <area href="shortcut.html" alt="Go"  
    shape="circle" coords="184,200,60"/>  
  <area href="top10.html" alt="Top Ten"  
    shape="poly" coords="276,0,276,28,100,200,50,50,276,0"/>  
  <area nohref shape="default">  
</map>
```

Il linguaggio HTML: <area>

> Il tag area associa un'area dell'immagine ad un link

> Attributi:

shape = default|rect|circle|poly

specifica la forma dell'area da definire

coords = coordinates

rect: left-x, top-y, right-x, bottom-y

circle: center-x, center-y, radius

poly: x1, y1, x2, y2, ..., xN, yN

href = uri

destination anchor per l'area definita

nohref

se specificato indica che nessun link è associato all'area definita

alt = text

specifica un testo alternativo per l'area definita

Il linguaggio HTML: Form

- > Sezione del documento HTML che contiene elementi di controllo con cui l'utente può inserire dati o in generale interagire.
- > I dati inseriti possono essere poi inoltrati ad un agente che può processarli.
- > Gli elementi di controllo sono accessibili da parte di script client-side.
- > Gli elementi di controllo sono caratterizzati da un valore iniziale e da un valore corrente.
- > Gli elementi di controllo possono essere:

Bottoni

CheckBox (Switch on/off)

Radio Buttons (Switch mutuamente esclusivi)

Menu di selezione (Lista di opzioni)

Inserimento di testo

Oggetti nascosti (elementi di controllo valorizzati ma invisibili)

Il linguaggio HTML: Form

> Racchiude tutti gli elementi della form

> Attributi:

action = uri

specifica l'URI dell'agente che riceverà i dati della form

name = text

specifica il nome della form

method = {get|post}

specifica il modo in cui i dati vengono inviati

enctype = content-type

specifica il content type usato per l'*encoding* del data set del form (quando il metodo è *post*). Il default è: *application/x-www-form-urlencoded*.

```
<form action="http://somesite.com/prog/adduser" method="post">  
...form contents...  
</form>
```

Il linguaggio HTML: Input text

```
<form action="http://somesite.com/prog/adduser" method="post">  
  <p>  
    First name: <input type="text" name="firstname"/><br/>  
  </p>  
</form>
```



>Attributi:

name = text

specifica il nome del controllo

value = text

specifica un eventuale valore iniziale

size = n

lunghezza del campo specificata in numero di caratteri

maxlength = n

massima lunghezza del testo specificata in numero di caratteri

Il linguaggio HTML: Input file

```
<form action="http://somesite.com/prog/adduser" method="post"
      enctype="multipart/form-data" >
  <p>
    File: <input type="file" name="attach"/><br/>
  </p>
</form>
```



>Attributi:

- name = text*
specifica il nome del controllo
- value = content-type*
lista di MIME types per l'upload

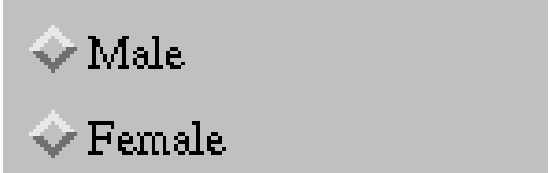
Il linguaggio HTML: Checkbox

```
<form action="http://somesite.com/prog/adduser" method="post">  
  <p>  
    <input type= "checkbox" name="food" value="pane"/>Pane<br/>  
    <input type= "checkbox" name="food" value="burro"/>Burro<br/>  
    <input type= "checkbox" name="drink"  
      value="acqua" checked="checked"/>Acqua<br/>  
  </p>  
</form>
```

☐ Pane
☐ Burro
☒ Acqua

Il linguaggio HTML: Radio

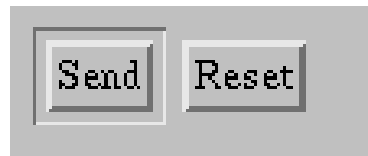
```
<form action="http://somesite.com/prog/adduser" method="post">  
  <p>  
    <input type="radio" name="sex" value="Male">Male<br/>  
    <input type="radio" name="sex" value="Female">Female<br/>  
  </p>  
</form>
```



Male
Female

Il linguaggio HTML: Button

```
<form action="http://somesite.com/prog/adduser" method="post">  
  <p>  
    <input type="submit" value="Send"><br/>  
    <input type="reset"><br/>  
    <input type="image" src="invia.gif alt="Invia"><br/>  
    <input type="button" name="check" value="Check"><br/>  
    <input type="hidden" name="status" value="view"><br/>  
  </p>  
</form>
```



Il linguaggio HTML: Drop down list

```
<form action="http://somesite.com/prog/component-select"
      method="post">
  <p>
    <select multiple size="3" name="component-select">
      <option selected value="Component_1_a">Component_1</option>
      <option selected value="Component_1_b">Component_2</option>
      <option>Component_3</option>
      <option>Component_4</option>
    </select>
    <input type="submit" value="Send">< input type="reset">
  </p>
</form>
```



Il linguaggio HTML: Frames

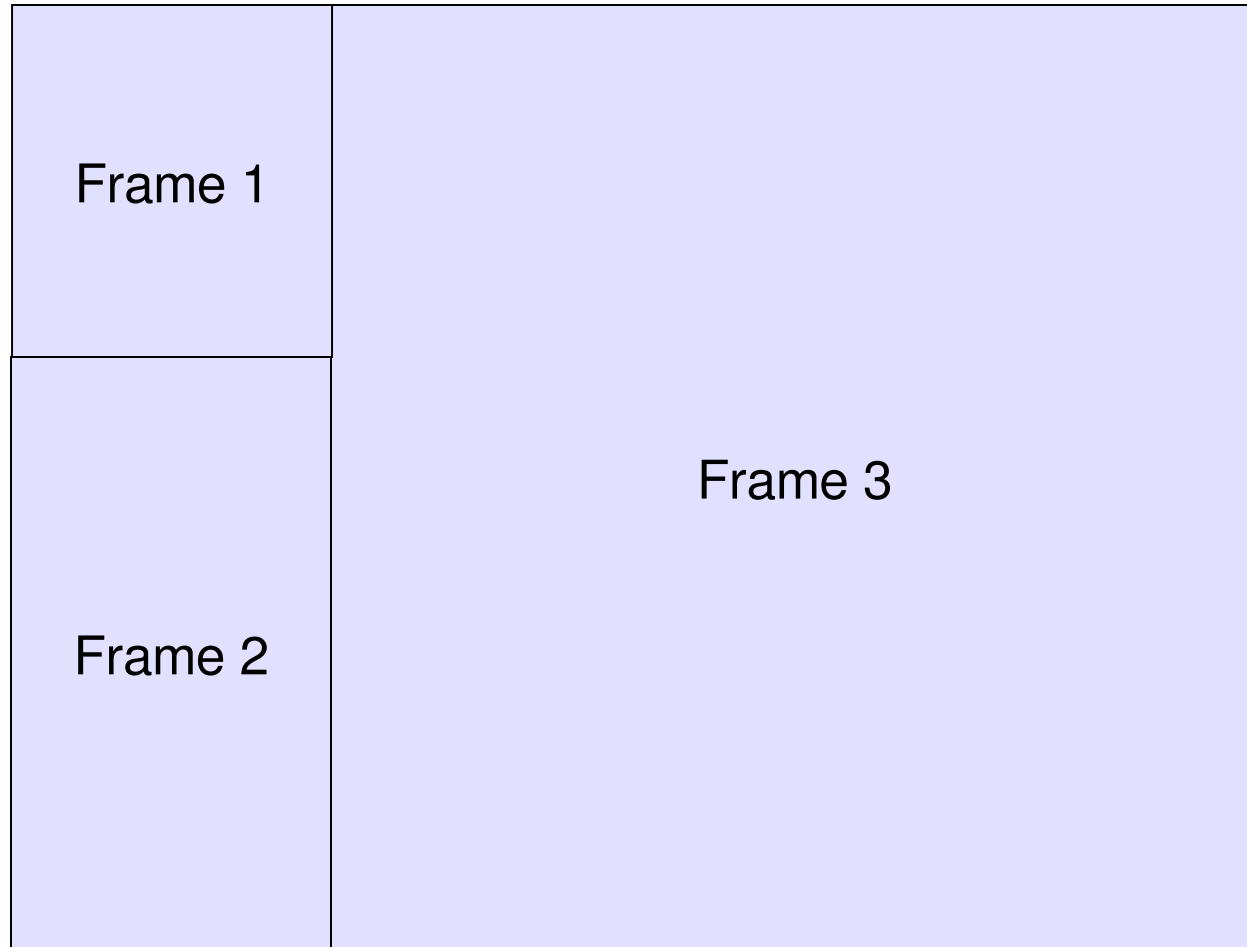
- > Strumento che consente di presentare i documenti mediante viste multiple.
- > E' possibile suddividere il documento in finestre o porzioni di finestra indipendenti.
- > Offre la possibilità di mantenere sezioni del documento sempre visibili.
- > Le porzioni del documento possono interagire e scambiare informazioni.

Il linguaggio HTML: Frames

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

```
<html>  
  <head>  
    <title>A simple frameset document</title>  
  </head>  
  <frameset cols="20%, 80%">  
    <frameset rows="100, 200">  
      <frame src="contents_of_frame1.html">  
      <frame src="contents_of_frame2.gif">  
    </frameset>  
    <frame src="contents_of_frame3.html">  
  </frameset>  
  <noframes>  
    .....  
  </noframes>  
</html>
```

Il linguaggio HTML: Frames



Il linguaggio HTML: Noframes

.....

<noframes>

<p>

This frameset document contains:

Some contents

Some other
contents

</p>

</noframes>

.....

Il linguaggio HTML: Noframes

> Specifica il layout della struttura a frame della finestra.

> Attributi:

rows = length,length,...

layout dei frame in senso orizzontale

cols = length,length,...

layout dei frame in senso verticale

```
<FRAMESET rows="50%, 50%">  
...the rest of the definition...  
</FRAMESET>
```

```
<FRAMESET cols="1*,250,3*">  
...the rest of the definition...  
</FRAMESET>
```

```
<FRAMESET rows="30%,70%"  
    cols="33%,34%,33%">  
...the rest of the definition...  
</FRAMESET>
```

```
<FRAMESET rows="30%,400,*,2*">  
...the rest of the definition...  
</FRAMESET>
```

Il linguaggio HTML: Nested Frameset

```
<html>
  <head>
    <title>A nested frameset document</title>
  </head>
  <frameset cols="33%, 33%, 34%">
    ...contents of first frame...
    <frameset rows="40%, 50%">
      ...contents of second frame, first row...
      ...contents of second frame, second row...
    </frameset>
    ...contents of third frame...
  </frameset>
</html>
```

Il linguaggio HTML: Tag Frame

> Definisce le caratteristiche di ciascun frame.

> Attributi:

`name = text` specifica il nome del frame

`src = uri` specifica il contenuto iniziale del frame

`noresize` se specificato indica che il frame non può essere ridimensionato

`scrolling = {auto|yes|no}` specifica se il frame debba essere dotato di scrollbar

`frameborder = 1/0` specifica se il frame debba avere dei bordi di separazione dei frame adiacenti

`marginwidth = pixels` quantità di spazio vuoto fra i bordi verticali del frame e il suo contenuto

`marginheight = pixels` quantità di spazio vuoto fra i bordi orizzontali del frame e il suo contenuto

Il linguaggio HTML: Tag Frame

```
<html>
  <head>
    <title>A frameset document</title>
  </head>
  <frameset cols="33%,33%,33%">
    <frameset rows="*,200">
      <frame src="contents_of_frame1.html" scrolling="no">
      <frame src="contents_of_frame2.gif"
        marginwidth="10" marginheight="15" noresize>
    </frameset>
    <frame src="contents_of_frame3.html" frameborder="0">
    <frame src="contents_of_frame4.html" frameborder="0">
  </frameset>
</html>
```

Il linguaggio HTML: Attributo Target

- > Specifica il frame dove un nuovo document verrà aperto

target = “*frame-target*”

- > Tag che supportano l'attributo target:

A

indica in quale frame deve essere aperto il destination anchor

AREA

indica in quale frame deve essere aperto il destination anchor

FORM

indica in quale frame deve essere aperto il documento di risposta al submit

Il linguaggio HTML: Frame target

> Nome del frame:

Stringa alfanumerica che inizia con a-zA-Z

> Frame speciali:

`_blank`

una nuova finestra

`_self`

il frame stesso

`_parent`

frame di ordine gerarchico immediatamente superiore

`_top`

la finestra originale che contiene la gerarchia di frame

Il linguaggio HTML: Target - Esempio

```
<html>
  <head>
    <title>A frameset document</title>
  </head>
  <frameset rows="50%,50%">
    <frame name="fixed" src="init_fixed.html">
    <frame name="dynamic" src="init_dynamic.html">
  </frameset>
</html>
```

```
<html>
  <head>
    <title>A document with anchors with specific targets</title>
  </head>
  <body>
    <p>Now you may advance to
      <a href="slide2.html" target="dynamic">slide 2.</a></br>
      You're doing great. Now on to
      <a href="slide3.html" target="dynamic">slide 3.</a>
    </p>
  </body>
</html>
```

Il linguaggio HTML: Iframe

> L'elemento `<iframe>` crea un frame *inline* che contiene un altro documento.

