

HTML (HyperText Markup Language)

Antonio Lioy
< lioy@polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

Un po' di storia

- **HTML 2.0 (nov'95 = RFC-1886)**
 - codifica lo standard de facto del 1994
- **HTML 3.2 (1996)**
 - compatibile con 2.0
 - aggiunge table, applet, apici, pedici, testo attorno alle immagini, ...
- **HTML 4.01 (dic' 97 – apr' 98 – dic' 99)**
 - ultima (?) versione di HTML
- **XHTML 1.0 (gen'00 – ago'02)**
 - riscrittura di HTML 4.01 con XML
 - strict / transitional / frameset

Un documento HTML

- è un normale testo US-ASCII
 - non sono quindi ammesse lettere accentate ed altri caratteri "estesi"
- ... arricchito con puntatori ipertestuali e ipermediali
- ... e con moderate capacità di formattazione del testo
- tutte queste capacità aggiuntive sono ottenute mediante annotazioni espresse tramite i *tag*

I tag

- sono racchiusi tra i simboli "minore" e "maggiore"
- di solito sono accoppiati (start tag – end tag)
`<h1> ... </h1>`
ma possono essere anche soli
`
` (`
` in XHTML)
- regola generale: il tag finale è identico a quello iniziale, preceduto dal simbolo `/`
- sono *case insensitive* in HTML e *lowercase* in XHTML
 - meglio quindi scriverli sempre minuscoli

Gli attributi

- tramite gli attributi è possibile caratterizzare meglio un tag
- ogni attributo è costituito da una variabile a cui viene assegnato un valore, posta all'interno del tag di apertura (es. `<hr width="90%">`)



I browser

- per visualizzare documenti HTML (e per navigarli) occorre un programma apposito: un *browser* HTML
- un browser è un interprete:
 - legge codice sorgente (HTML + estensioni)
 - cerca di capirlo (sperando non contenga errori ...)
 - fa del suo meglio per visualizzare quanto descritto dal codice sorgente
- **attenzione! non tutti i browser visualizzano un documento allo stesso modo**
- browser testuale: Lynx
- browser grafici: Firefox, SeaMonkey, Chrome, Opera, Internet Explorer, Netscape, ...

La guerra dei browser

- non esiste un unico browser più usato degli altri (preferenze personali, piattaforma di uso, ...)
- bisogna cercare di scrivere pagine HTML adatte a qualunque browser
- alcune statistiche:
 - www.w3schools.com/browsers/browsers_stats.asp
 - (jan-10) IE=27%, FX=43%, Chrome=24%, Safari=4%, Opera=2%
 - www.upsdell.com/BrowserNews/stat.htm
 - grande variabilità
 - www.pgts.com.au/pgtsj/pgtsj0212d.html
 - difficile identificare i browser con certezza

Struttura generale di un documento HTML

```
<!DOCTYPE HTML PUBLIC ...>
<html>
  <head>
    <title> titolo </title>
    ... altre intestazioni ...
  </head>
  <body>
    testo del documento
  </body>
</html>
```

Document type declaration (DTD)

- **necessario per capire il tipo di HTML (da 4.01):**
 - strict.dtd = tutti gli elementi non deprecati
 - loose.dtd = include elementi deprecati
 - frameset.dtd = loose + uso di frame

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

Esempio

```
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Esempio di pagina HTML</title>
</head>
<body>
Qui posso inserire il testo del mio documento
che, se non uso i tag di formattazione, viene
visualizzato come semplice testo.
</body>
</html>
```

Osservazioni

- **il browser non segnala gli errori: li ignora!**
- **spazi e ritorni a capo:**
 - spazi multipli vengono trattati come uno spazio singolo
 - i fine linea non hanno alcun effetto di formattazione
- **molto importante il titolo (ed in generale i dati dell'head) perché è l'elemento più usato dai sistemi automatici di indicizzazione**
- **HTML è un linguaggio estensibile**
 - spesso si aggiungono nuovi tag
 - il browser ignora i tag (o gli attributi) non riconosciuti ... ma visualizza il testo racchiuso dal tag

Meta-dati

- **nella parte di HEAD**
- **dati utili per:**
 - indicizzare la pagina HTML
 - fornire informazioni al server web e/o al browser
- **sintassi:**

```
<meta name="author" content="Antonio Liroy">
<meta name="keywords" content="html">

<meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
<meta http-equiv="Expires"
  content="Sun, 28 Feb 2010 23:59:00 GMT">
```

Internazionalizzazione di HTML

- pagine HTML < 4 scritte con codifica ISO-8859-1
- HTML-4 incorpora RFC-2070 che specifica le regole di internazionalizzazione (abbreviato "i18n")
- HTML-4 adotta lo standard ISO/IEC:10646 (UCS)
 - nota: ISO-8859-1 = byte 0..255 di UCS e Unicode
- UA determina la codifica usata in base ai seguenti elementi (da priorità più alta a più bassa):
 - HTTP response header "Content-Type: charset=xxx"
 - specifico tag META nell'header HTML
<meta http-equiv="Content-Type" charset=xxx" ...>
 - attributo charset di un elemento che punta ad una risorsa esterna

Il tag "link"

- nella parte di HEAD
- collegamento logico ad altri documenti correlati in qualche modo a quello corrente
- possibili più tag LINK
- attributi:
 - href=URL
 - rel=alternate lang=...
 - rel=alternate media=...
 - rel=stylesheet
 - rel=start / contents / prev / next / ...
 - type=MIME-type

Esempio di LINK

```
<head>
  <title>Chapter 2</title>
  <link rel="contents" href="../toc.html">
  <link rel="next" href="chapter3.html">
  <link rel="prev" href="chapter1.html">
  <link rel="stylesheet"
        type="text/css" href="mystyle.css">
</head>
```

Strumenti di controllo per HTML

- **<http://validator.w3.org>**
 - permette di verificare se una pagina è scritta rispettando completamente la sintassi ufficiale
 - può fornire spiegazioni dettagliate sugli errori e come correggerli
- **<http://tidy.sourceforge.net>**
 - “ripulisce” il codice HTML e lo trasforma in una versione più recente
 - installabile localmente o usabile via rete
<http://cgi.w3.org/cgi-bin/tidy>
- **problemi con HTML generato da un server dinamico (non validabile una pagina sorgente ASP o PHP)**

Strumenti di controllo per HTML

- per validare pagine create dinamicamente bisogna fare l'operazione sul client invece che sul server
- occorre quindi:
 - un plugin speciale per il browser
 - visitare manualmente tutte le pagine da validare
- un ottimo plugin per FireFox:
 - <http://users.skynet.be/mgueury/mozilla/index.html>
 - configurarlo in modalità "SGML parser" per avere gli stessi risultati di validator.w3.org
- attenzione ad alcuni errori tipici (soprattutto con gli script client-side):
 - www.htmlhelp.com/tools/validator/problems.html

Commenti

- inseribili in qualunque punto del testo
- possono occupare varie righe
- racchiusi tra `<!-- e -->`
- esempi:

```
<!-- questo è un commento -->

<!--
questo commento
occupa quattro righe
-->
```

I titoli

- **esistono sei livelli di titolo o intestazione:**
 - `<h1> . . . </h1>`
 - `<h2> . . . </h2>`
 - `<h3> . . . </h3>`
 - `<h4> . . . </h4>`
 - `<h5> . . . </h5>`
 - `<h6> . . . </h6>`
- **usarli per il significato logico, non per ottenere una certa formattazione**
- **in particolare è scorretto usare `<hN>` se non preceduto da `<hN-1>`**

Blocchi di testo

- `<p> . . . </p>`
 - inizia e termina un paragrafo
 - al termine di un paragrafo il browser va a capo (e può anche lasciare un piccolo spazio verticale)
- `
` (HTML)
`
` (XHTML)
 - inserisce un ritorno a capo
- `<hr>` (HTML)
`<hr/>` (XHTML)
 - inserisce una retta orizzontale

Rette orizzontali (tag <HR>)

- è possibile specificare degli attributi:
 - `size= n_pixel` (spessore)
 - `width= n_pixel` (larg. assoluta)
 - `width= percentuale` (larg. in % del contenitore)
 - `align=left / right / center`
- per default la linea è centrata e con larghezza 100%

Elenchi e liste

- lista non ordinata:
 - ` ... `
- lista ordinata:
 - ` ... `
- directory (deprecato):
 - `<dir> ... </dir>`
- menù (deprecato):
 - `<menu> ... </menu>`
- un elemento di una (qualsiasi) lista:
 - ` ... `

Opzioni per le liste

- tipo di simbolo per le liste non ordinate:
 - `type=disc / circle / square`
- numerazione delle liste ordinate:
 - `start= indice_primo_elemento`
 - `type=A / a / I / i / 1`
 - ossia:
 - lista alfabetica (maiuscola o minuscola)
 - numeri Romani (maiuscoli o minuscoli)
 - numeri decimali
 - specificabile sia sulla lista (ol) sia sul singolo elemento (li)

Esempio di elenco

Per superare l'esame 01ENY:

```
<ol type="I">  
<li>frequentare  
le lezioni</li>  
<li>svolgere  
le esercitazioni  
di laboratorio</li>  
</ol>
```

browser
(notare l'indentazione)

Per superare l'esame 01ENY:
I. frequentare le lezioni
II. svolgere le esercitazioni
di laboratorio

Definizioni

- elenco di definizioni:

```
<dl>
  <dt> termine 1 </dt>
  <dd> definizione 1 ... </dd>
  <dt> termine 2 </dt>
  <dd> definizione 2 ... </dd>
  .....
</dl>
```

Formattazione del testo

- una porzione di testo può essere caratterizzata per la *funzione* che assume nel documento (stile logico) ...
- ... oppure per il modo in cui si desidera *fisicamente* visualizzarla (stile fisico)
- si tende a preferire gli stili logici per lasciare più libertà all'utente finale nel definire come il testo deve apparire su video
- con XHTML (strict) sono scomparsi i tag di formattazione (si deve obbligatoriamente usare CSS)

Formattazione: stili fisici

- ` ... `
 - testo in grassetto (bold)
- `<i> ... </i>`
 - testo in corsivo (italico)
- `<u> ... </u>`
 - testo sottolineato
- `<tt> ... </tt>`
 - testo a spaziatura fissa (tipo macchina da scrivere)
- `<blink> ... </blink>`
 - testo lampeggiante

Formattazione: stili fisici

- `^{...}`
 - testo sopraelevato (apice)
- `_{...}`
 - testo sottoelevato (pedice)
- `<s> ... </s>`
■ `<strike> ... </strike>`
 - testo sbarrato orizzontalmente

Formattazione: stili logici

- `<cite>` *citazione* `</cite >`
- `<code>` *codice (programma)* `</code>`
- `` *enfasi* ``
- `<kbd>` *tastiera* `</kbd>`
- `<samp>` *esempio* `</samp>`
- `` *rinforzo* ``
- `<var>` *variabile* `</var>`
- `<dfn>` *definizione* `</dfn>`



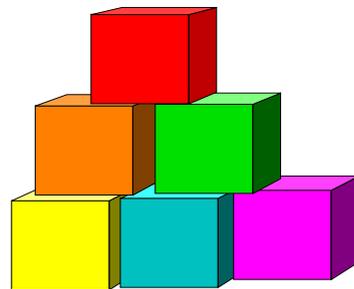
Altri stili logici

- `<big>` *testo grande* `</big>`
- `<small>` *testo piccolo* `</small>`
- si possono usare ripetutamente in modo annidato per ottenere effetto maggiore:

```
<big> <big> testo molto grande </big> </big>
```

Formattazione: blocchi di testo

- `<address> ... </address>`
 - indirizzo (tipicamente di e-mail)
- `<blockquote> ... </blockquote>`
 - grosse citazioni
- `<center> ... </center>`
 - testo centrato
- `<pre> ... </pre>`
 - testo preformattato (la spaziatura viene rispettata)



Riferimenti a caratteri non US-ASCII

- HTML normalmente scritto in US-ASCII con MSB=0
- caratteri ASCII > 127 vanno codificati, come pure caratteri con significato speciale
- prestare attenzione al ";" finale

<i>per avere ...</i>	<i>si scrive ...</i>	
<	<code>&lt;</code>	} importanti perché sono caratteri riservati di HTML
>	<code>&gt;</code>	
&	<code>&amp;</code>	
"	<code>&quot;</code>	
É	<code>&Egrave;</code>	
é	<code>&eacute;</code>	
©	<code>&copy;</code>	

Riferimenti a caratteri non US-ASCII

- sezione 24 (pag. 299) dello standard HTML 4.01
- comprende:
 - caratteri estesi di ISO-8859-1 – es. » = »
 - simboli matematici – es. ∃ = \exists
 - lettere greche – es. α = α
 - simboli internazionali – es. € = €

I collegamenti (hyperlink)

- utilizzando un collegamento (detto *àncora* in HTML) è possibile spostarsi automaticamente da una risorsa ad un'altra
- il tag che identifica la presenza di un collegamento è l'ancora, indicata con `<a>`

Come inserire un hyperlink

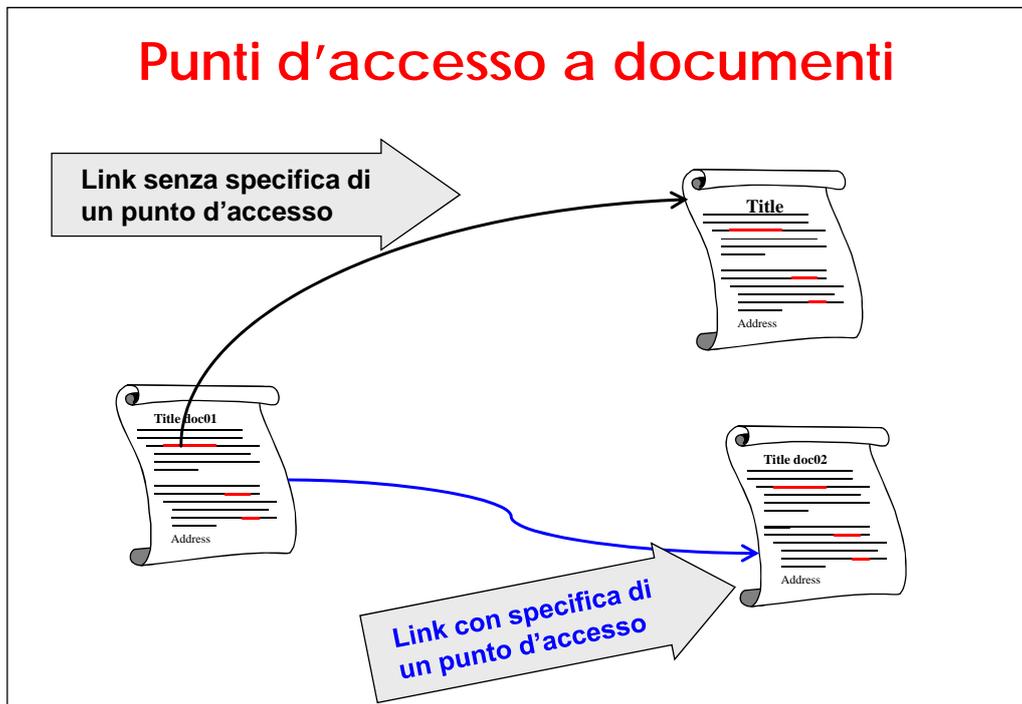
- aprire il tag di inizio àncora: `<a`
- inserire uno spazio
- inserire l'URL della risorsa, preceduto da `href=` e racchiuso tra virgolette
- chiudere il tag di apertura con `>`
- inserire il testo da evidenziare (quello associato all'àncora, detto "hot word")
- chiudere l'àncora: ``

```
<a href="http://www.polito.it">POLITO</a>
```

Link assoluti e relativi

- possibile omettere parti della URL
 - in questo caso si parla di link "relativo"
 - le parti mancanti assumono il valore della pagina corrente
- esempi di link relativi (supposti presenti nella pagina <http://www.lioy.it/01eny/esame.html>)

<i>link relativo</i>	<i>link assoluto</i>
<code>biblio.html</code>	<code>http://www.lioy.it/01eny/biblio.html</code>
<code>../cv.html</code>	<code>http://www.lioy.it/cv.html</code>
<code>ris/a1.html</code>	<code>http://www.lioy.it/01eny/ris/a1.html</code>



Punti d'accesso a documenti

- nel documento “bersaglio” definire il punto d'accesso tramite un'ancora con attributo NAME

```
<a name="cuc_ita">
La cucina italiana
</a>
```

- nel documento di partenza, includere nell'URL il nome del punto di accesso

```
<a href="doc2.html#cuc_ita">
```

- il punto d'accesso può anche essere un qualsiasi elemento identificato tramite il suo "id"

```
<h1 id="cuc_ita">
La cucina italiana
</h1>
```

Immagini

- ``
 - inserisce l'immagine contenuta nel file polito.gif
- ``
 - inserisce l'immagine polito.gif, ma, se il browser non prevede l'uso della grafica, viene visualizzato il testo Foto del Politecnico
- **differenza tra inserimento o link ad un'immagine:**
 - `` (inserisce l'immagine nella pagina)
 - `` (seguendo il link si visita una pagina che contiene solo l'immagine)

Posizionamento reciproco di testo e immagini

```
<img align=left ...>
<img align=right ...>
<img align=top ...>
<img align=texttop ...>
<img align=middle ...>
<img align=absmiddle ...>
<img align=baseline ...>
<img align=bottom ...>
<img align=absbottom ...>
```

Formato delle immagini

- ``
 - dimensione dell'immagine
 - permette visualizzazione più rapida della pagina (non occorre aver caricato tutta l'immagine per sapere quale spazio occupa)
- ``
 - distanze minime tra testo e immagine
- ``
 - dimensione del bordo

Font

- ` ... `
 - scelta del font per il blocco di testo incluso tra i tag
 - uso sconsigliato (usare CSS)
- **attributi:**
 - `size=` *dimensione*
 - `color=` *colore*
 - `face=` *font-family* (es. Arial, Helvetica, Times)
 - ...
- **la dimensione può essere data in varie unità:**
 - N (=1...7, default=3), +N, -N
 - consigliabili +N -N

Colori

- alcuni colori predefiniti ed accessibili per nome:

Black, White, Gray, Silver,
Yellow, Red, Purple, Fuchsia,
Maroon, Green, Lime, Olive, Aqua,
Teal, Blue, Navy

- altri colori sono specificabili mediante il loro codice RGB in esadecimale (# rr gg bb)

- esempio:

- `Bianco!`

Colori standard

	black = #000000		green = #008000
	silver = #C0C0C0		lime = #00FF00
	gray = #808080		olive = #808000
	white = #FFFFFF		yellow = #FFFF00
	maroon = #800000		navy = #000080
	red = #FF0000		blue = #0000FF
	purple = #800080		teal = #008080
	fuchsia = #FF00FF		aqua = #00FFFF

Tabelle

- `<table ... > ... </table>`
- **attributi:**
 - `align= left / center / right`
 - `border= dimensione`
 - `width= dimensione (n_pixel o %)`
 - `cellspacing= dimensione`
 - `cellpadding= dimensione`
 - `summary= testo`
 - `frame= void / above / below / hside / lhs / rhs / vside / box / border`
 - `rules = none / groups / rows / cols / all`

Dati in tabella

- `<tr ... > ... </tr>`
 - una riga della tabella
 - contiene celle normali (`<td>`) o di intestazione (`<th>`)
- `<th ... > ... </th>`
`<td ... > ... </td>`

un dato (oppure un'intestazione = heading) della tabella, che può occupare più celle in orizzontale o verticale

 - `colspan= numero-colonne`
 - `rowspan= numero-righe`

Elementi (opzionali) di una tabella

- **<thead>**
 - l'intestazione
- **<tbody>**
 - il contenuto
- **<tfoot>**
 - il footer
- **<caption>**
 - la didascalia

Table: attributi di riga, header e dati

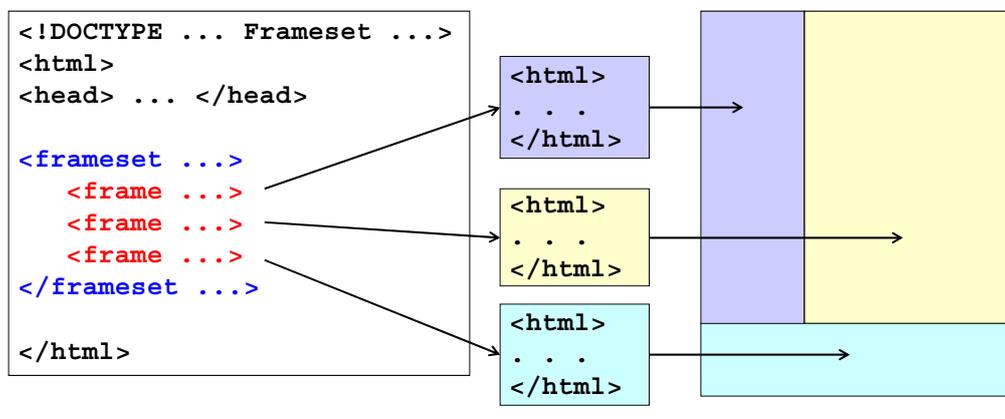
- **align= allineamento-orizzontale**
 - left, center, right
- **valign= allineamento-verticale**
 - top, middle, bottom
 - baseline
- **bgcolor= colore**

Table: gruppi di colonne

- **<colgroup span=n width=... align=... valign=...>**
 - gruppo strutturale di n colonne, ciascuna con gli attributi specificati
- **<col span=n width=... align=... valign=...>**
 - definizione di attributi per una o più colonne

I frame

- **divisione di una pagina in zone il cui contenuto è specificato da altri file HTML**
- **uso sconsigliato (usare CSS e “include”)**



Frameset e Frame

- la struttura di un pagina organizzata in frame è simile a quella tradizionale, sostituendo `<body>` con `<frameset>`
- è possibile annidare `FrameSet` per creare suddivisioni complesse della pagina
- il contenuto di ciascun frame è specificato tramite:
 - `<frame src=URI name=...>`
- usare il tag `<noframe>` per testo da visualizzare sui browser che non supportano i frame

Spazio occupato dai frame

- possibile specificare la porzione di pagina occupata da ciascun frame, usando:
 - percentuale (dello spazio disponibile)
 - valore assoluto (in pixel)
 - "*" per indicare di usare tutto lo spazio rimasto
- in caso di "overflow" si attivano le scrollbar (H e V)
- esempio (suddivisione in tre frame orizzontali):
 - `<frameset rows="20%,50%,30%">`
- esempi (suddivisione verticale):
 - `<frameset cols="20%,80%">`
 - `<frameset cols="100,*,100">`

Navigazione dei frame

- nei link è bene indicare in quale frame (o finestra) deve essere visualizzata la pagina bersaglio:
 - ` ... `
- valori speciali del target:
 - `"_blank"` (nuova finestra)
 - `"_self"` (nello stesso frame) = default
 - `"_parent"` (frameset di ordine superiore)
 - `"_top"` (occupare l'intera finestra)

Esempio di frame (I)

```
<!-- pagina iniziale -->
<frameset rows="80%,20%">
  <noframe>
    <p>Pagina non visualizzabile</p>
  </noframe>

  <frameset cols="100,*">
    <frame src="menu.html">
    <frame src="p1.html" name="content">
  </frameset>

  <frame src="footer.html">

</frameset>
```

Esempio di frame (II)

```
<!-- menu.html -->
<html>
<head> . . . </head>
<body>
<p><a href="p1.html" target="content">Pag.1</a></p>
<p><a href="p2.html" target="content">Pag.2</a></p>
<p><a href="p3.html" target="content">Pag.3</a></p>
</body>
</html>
```

Inline frame (iframe)

- un frame trattato come un singolo oggetto (es. come un'immagine)
- collocabile quindi in qualunque punto della pagina
- inizialmente supportato solo da IE
- sintassi:
 - <iframe src=uri ...> ... </iframe>
 - height=... width=... name=... frameborder=... marginwidth=... marginheight=... scrolling=yes/no/auto align=... vspace=... hspace=...
 - testo all'interno del tag viene ignorato dai browser che capiscono iframe e visualizzato dagli altri (usarlo per segnalazione di errore)

DIV e SPAN

- introdotti in HTML 4.0
- servono a raggruppare parti per poter applicare più facilmente una formattazione
- DIV identifica un blocco (tipicamente i browser vanno a capo alla fine del blocco)
- SPAN identifica una parte all'interno di un blocco
- molto usati per creare (con un opportuno CSS) un layout di pagina senza usare table o frame
- "id" e "class" permettono i riferimenti al CSS

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

```
<span id="..." class="..."> ... </span>
```

Attributi generali dei tag HTML

- **id = "stringa"**
 - àncora per un link
 - riferimento ad un elemento da parte di uno script
 - riferimento ad un campo di un form
 - riferimento per uno stile specifico in CSS
- **class = "class1 class2 ..."**
 - elenco di classi CSS da applicarsi all'elemento
- **title = "titolo"**
 - visualizzato quando si punta l'elemento
- **lang = "lingua"**
 - per lettura automatica del testo (valori: en it fr de ...)

Favourite icon

- **la piccola icona presente vicino alla URL**

- un'immagine 16 x 16 pixel

- **vecchi browser:**

- solo in formato MS icon
- in posizione fissa e con nome fisso = /favicon.ico

- **primo passo verso lo standard:**

```
<link rel="shortcut icon" href="/icons/my.ico"
      type="image/vnd.microsoft.icon">
```

- **nuovi browser supportano standard de-facto:**

```
<link rel="icon"
      type="image/png" href="/icons/my.png">
```