

Programmazione Web

– HTML e CSS–

Francesco Tiezzi



Scuola di Scienze e Tecnologie

Sezione di Informatica
Università di Camerino

Anno accademico 2015-2016

Pagina Web

- Quando si fa riferimento ad un documento HTML spesso si utilizza il termine *pagina Web*
- Può contenere
 - ▶ collegamenti (*link*) ad altre pagine Web
 - ▶ immagini (fisse o in movimento)
 - ▶ suoni
 - ▶ scenari tridimensionali interattivi
 - ▶ codice eseguibile localmente
- Utilizzo
 - ▶ L'utente legge il testo della pagina Web, vede le immagini, ascolta la musica, . . .
 - ▶ L'utente seleziona un link e la pagina di partenza viene sostituita sullo schermo da quella relativa al link selezionato
 - ★ la nuova pagina può provenire da qualunque altra parte del mondo

WWW: World Wide Web

- Assieme alla posta elettronica, il World Wide Web (in breve Web) è uno dei servizi di Internet più utilizzati e conosciuti
 - ▶ Il Web è uno spazio elettronico e digitale destinato alla pubblicazione di contenuti multimediali nonché uno strumento per implementare servizi (es. download di software, web mail, streaming, . . .)
- Sistema di indirizzamento basato su URL
 - ▶ permette di far riferimento alle entità indirizzabili nel Web, che possono essere documenti o programmi
- Linguaggio HTML
 - ▶ linguaggio di definizione delle pagine Web
- Protocollo HTTP
 - ▶ utilizzato da client e server per comunicare

Uniform Resource Locator

- Viene usato per indirizzare una risorsa (testo, immagine, . . .) nel Web
- Specifica
 - ▶ il protocollo per accedere al server
 - ▶ il nome del server
 - ▶ la locazione (*path*) di un file su quel server
- Esempio: `http://cs-research.unicam.it/people`
- Nelle pagine dinamiche (sviluppate con PHP, ASP, ecc.), gli URL possono anche specificare una query-string
 - ▶ serve per trasmettere informazioni alla pagina seguente (es. pagina successiva a una form)
 - ▶ preceduta da `?` ed intervallata da `&` (se ne è presente più di una)
 - ▶ esempio: `https://www.google.it/?q=unicam`

Sintassi di un URL

```
protocol://hostname[:port]/path/filename#sec
```

protocol: il protocollo utilizzato nella comunicazione

- file, ftp, http, mailto, ...

hostname: è il nome del server che fornisce la risorsa

- indirizzo IP del server o nome logico

port: questo parametro (opzionale) indica la porta da utilizzare nella comunicazione TCP/IP (es. 8080 nel caso di servlet)

path: indica il direttorio nel server

filename: il file corrispondente alla risorsa

#sec: è un riferimento ad uno specifico punto all'interno del documento

Linguaggio HTML

- HTML è l'acronimo di **H**yper **T**ext **M**arkup **L**anguage
- Non è un linguaggio di programmazione, ma di *markup*
 - ▶ descrive il contenuto di una pagina Web tramite una *marcatatura descrittiva* che definisce la sua struttura logica
- Sviluppato da Tim Berners-Lee al CERN di Ginevra
- È di pubblico dominio e la sua sintassi è stata stabilita dal World Wide Web Consortium (W3C)
- È basato su SGML (Standard Generalized Markup Language)
 - ▶ standard per descrivere linguaggi di *markup*
 - ▶ anche XML è basato su SGML

Documento HTML

- Un documento ipertestuale scritto in HTML deve essere contenuto in un file con estensione `.htm` oppure `.html`
- Un documento HTML può essere creato con un semplice editor di testo (es. NotePad su Windows, TextEdit su MAC, . . .)
- Un documento HTML può essere letto con vari browser (es. Internet Explorer, Firefox, Opera, . . .)
- Un buon modo per imparare HTML consiste nel vedere come altre persone hanno codificato le proprie pagine Web
 - ▶ Firefox: selezionare *Visualizza* nel menu e quindi *Sorgente pagina*
 - ▶ Internet Explorer: selezionare *Pagina* nel menu e quindi *HTML*

Tag

- Il concetto chiave di HTML è quello di *tag*
- I tag sono delle etichette che servono a marcare componenti fondamentali della struttura di un testo
 - ▶ intestazioni, tabelle, liste, ...
- Sintassi: un tag `<tagname>` è una sequenza di caratteri racchiusa da due parentesi angolari
 - ▶ non case-sensitive, cioè `` e `` sono lo stesso tag
 - ▶ coppia di tag per definire inizio e fine `<tagname>...</tagname>`
 - ▶ *attributi* nel tag di apertura
 - ★ `<tagname attr1="value" attr2="value" >`
 - ★ forniscono informazioni aggiuntive sul contenuto del tag
 - ★ il valore di un attributo deve essere racchiuso da una coppia di " oppure di '

Tag obbligatori

- Ogni documento HTML dovrebbe
 - ▶ *iniziare* con l'indicazione della definizione di tipo del documento (DTD)

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

la quale dice al browser l'URL delle specifiche HTML utilizzate dal documento (indicando quindi, implicitamente, quali tag ed attributi possiamo utilizzare)

NB: in HTML 5 `<!DOCTYPE html>`

- ▶ *contenere* i seguenti elementi
 - ★ `<html>...</html>`
 - ★ `<head>...</head>`
 - ★ `<title>...</title>`
 - ★ `<body>...</body>`

- Alcuni browser (ma non tutti) permettono di non specificarli

I quattro tag fondamentali

html: indica che l'informazione contenuta nel file è un documento HTML;

head: identifica la parte del documento che contiene il titolo e altre informazioni sul documento (stile, script, . . .)

title: contiene il titolo del documento ed identifica il suo contenuto in un contesto globale

body: contiene la parte del documento visualizzata all'interno della finestra del browser

Esempio

```
<html>
  <head>
    <title>Un semplice esempio...</title>
  </head>
  <body>
    <H1>HTML &grave; facile da imparare</H1>
    <P>Benvenuti in HTML. Questo &grave;
      il primo paragrafo!</P>
    <P>E questo &grave; il secondo paragrafo </P>
  </body>
</html>
```

Intestazioni

- Abbiamo 6 possibili livelli di intestazione
 - ▶ livello 1, indica l'intestazione più generale;
 - ▶ livello 6, la più dettagliata.
- Il testo nelle intestazioni viene mostrato in forma più grande ed in grassetto
- Sintassi
 - ▶ `<Hn>testo</Hn>`, dove `n` è compreso tra 1 e 6
- Buona norma (ma non obbligatorio)
 - ▶ chiudere i tag delle intestazioni

Paragrafi

- In HTML i caratteri di fine riga non significativi e gli spazi multipli vengono compressi in un singolo spazio
- Un nuovo paragrafo inizia con il tag `<P>`
 - ▶ senza tag `<P>` il documento diviene un unico paragrafo
 - ▶ il tag di terminazione `</P>` può essere omissso
- Buona norma
 - ▶ usare linee bianche per la leggibilità del sorgente HTML

Liste numerate e non

- Il tag `` apre la lista **numerata** (*ordered list*)
- Il tag `` apre la lista **puntata** (*unordered list*)
- Ogni elemento della lista viene individuato dal tag ``
- Il tag di fine lista è necessario
- Il tag `` può essere omesso

Esempio

```
<html>
  <head>
    <title>Le liste...</title>
  </head>
  <body>
    <ul>
      <li>Aldo</li>
      <li>Giovanni</li>
      <li>Giacomo</li>
    </ul>
    <ol>
      <li>Aldo</li>
      <li>Giovanni</li>
      <li>Giacomo</li>
    </ol>
  </body>
</html>
```

Liste di definizione

- il tag `<dl>` apre la lista
- il tag `<dt>` individua l'etichetta
- il tag `<dd>` individua il testo
- non sono necessari i tag di chiusura `</dt>` e `</dd>`
- il tag `</dl>` chiude la lista

Esempio

```
<html>
  <head>
    <title>Le liste di definizione...</title>
  </head>
  <body>
    <dl>
      <dt>UNICAM</dt>
      <dd>Universit&agrave; di Camerino</dd>
      <dt>SST</dt>
      <dd>Scuola di Scienze e Tecnologie</dd>
      <dt>SI</dt>
      <dd>Sezione di Informatica</dd>
    </dl>
  </body>
</html>
```

Testo preformattato

- Il tag `<pre>` consente di generare testo con un tipo di font a larghezza fissa
 - ▶ rende significativi spazi multipli e ritorno di carrello
- Utile per includere listati di programma
- L'attributo `width` specifica il numero massimo di caratteri per linea
- Buona norma
 - ▶ non utilizzare dei tag HTML dentro una sezione `<pre>`

Esempio

```
<html>
  <head>
    <title>Testo preformattato...</title>
  </head>
  <body>
    Quello che segue &egrave; un testo preformattato:
    <pre>
// HelloDate.java
import java.util.*;

public class HelloDate {
    public static void main(String[] args) {
        System.out.println("Hello, it's: ");
        System.out.println(new Date());
    }
}
    </pre>
  </body>
</html>
```

Citazioni, indirizzo ed a capo

- `<blockquote>`, per quotare citazioni lunghe
- `<address>`, per specificare l'autore del documento e la data dell'ultima modifica
- `
`, per forzare l'interruzione di linea

Esempio

```
<html>
  <head><title>Citazioni e altro...</title></head>
  <body>
    <blockquote>
      <p>I've seen things you people wouldn't believe.</p>
      <p>Attack ships on fire off the shoulder of Orion.</p>
      <p>I watched C-beams glitter in the dark near the
        Tannhauser gate.</p>
      <p>All those moments will be lost in time, like tears
        in rain.</p>
      <p>Time to die.</p> --Roy (Blade Runner)
    </blockquote>
    <address>
      Francesco Tiezzi, francesco.tiezzi@unicam.it,
      ultima modifica 18.35
    </address>
    Univ. Camerino<br>Via del Bastione, 1<br>62032 Camerino
  </body>
</html>
```

Linee orizzontali

- Il tag `<hr>` permette di includere una riga orizzontale (Horizontal Rule)
 - ▶ utile per separare il testo
- L'attributo `size` permette di regolare l'altezza della, linea (1= la più sottile)
- L'attributo `width` permette di regolare la lunghezza della linea
 - ▶ si può esprimere sia in punti (numero assoluto) che in percentuale (rispetto alla larghezza della pagina)

Esempio

```
<html>
  <head>
    <title>Linee orizzontali...</title>
  </head>
  <body>
    <hr>
    <hr width="50%">
    <hr width="25%">
    <hr width=200>
    <hr size=4>
    <hr size=8>
    <hr size=16>
  </body>
</html>
```

Formattazione dei caratteri

- Abbiamo due tipi di tag di stile
 - ▶ *logici*: indicano il significato del testo (ad es. ``)
 - ▶ *fisici*: indicano l'aspetto del testo (ad es. `<i>`)
- Vantaggi dei tag logici:
 - ▶ modifica dello stile fisico di un tag logico più efficiente
 - ▶ forzano la consistenza nel documento
- Buona norma
 - ▶ essere coerenti: usare stesso tipo di tag (logici o fisici)

Esempio

```
<html>
  <head>
    <title>Stili logici...</title>
  </head>
  <body>
    <dfn>Definizione</dfn><br>
    <em>Enfaticizzato</em><br>
    <cite>Java Network Programming</cite><br>
    <code>program pippo</code><br>
    <kbd>querty</kbd>
    <samp>Command line</samp><br>
    <strong>Nota</strong><br>
    <var>hostname</var><br>
  </body>
</html>
```

Stili fisici

- Alcuni tag di stile fisico

- ▶ ``, testo in grassetto
- ▶ `<i>`, testo in italico
- ▶ `<tt>`, testo di larghezza fissa
- ▶ `<u>`, testo sottolineato
- ▶ `<strike>`, testo barrato da una linea
- ▶ `<sub>`, testo formattato come pedice
- ▶ `<sup>`, testo formattato come apice

Sequenze di *escape*

- Servono per includere nel testo caratteri usati per altro scopo (ad esempio <) oppure caratteri non inclusi in ASCII
- Iniziano con & terminano con ;
 - ▶ ` ` = non-breaking space
 - ▶ `<` = <
 - ▶ `>` = >
 - ▶ `&` = &
 - ▶ `ö` = ö
 - ▶ `ñ` = ñ
 - ▶ `à` = à
 - ▶ ...
- Sono case-sensitive

Collegamenti

- I collegamenti (*link*) sono fra gli elementi fondamentali di HTML
- Il tag `<a>` consente:
 - ▶ con l'attributo `href` di fare riferimento all'URL di un documento
 - ▶ con l'attributo `name` di definire la sezione di un documento
- Buona norma
 - ▶ usare, quando possibile, URL relativi

URL relativi

- Ereditano protocollo, nome del server e cammino del documento nel quale appaiono

- Vantaggi:
 - ▶ scrivere meno
 - ▶ i link interni al documento sono indipendenti alla posizione del *sito* sul server

Esempio

```
<html>
  <head>
    <title>I collegamenti...</title>
  </head>
  <body>
    <a href="books.html#jnp">JNP</a>
  </body>
</html>
```

Nel file `books.html`

```
...
<a name="jnp">Java Network Programming</a>
...
```

Immagini

- Per includere immagini si utilizza il tag ``
 - ▶ l'attributo `src` individua l'url (tra virgolette) dell'immagine da visualizzare
 - ▶ gli attributi `width` e `height` individuano, rispettivamente, larghezza e altezza dell'immagine
 - ▶ l'attributo `align`, con valori `bottom`, `left` o `center`, indica il posizionamento del testo circostante
 - ▶ l'attributo `alt` indica la stringa (tra virgolette) da mostrare al posto dell'immagine
- Buona norma
 - ▶ selezionare con cura le immagini da includere in un documento ed il loro numero specificando sempre l'attributo `alt`

Esempio

```
<html>
  <head>
    <title>Le immagini...</title>
  </head>
  <body>
    
    Testo allineato in basso<br><br>
    
    Testo allineato al centro<br><br>
    
    Testo allineato in alto<br><br>
  </body>
</html>
```


Le tabelle: elementi

- `<table>`, definisce una tabella
 - ▶ se presente l'attributo `border` la tabella ha un bordo
- `<caption>`, definisce il titolo della tabella
 - ▶ l'uso degli attributo `align` con valore `bottom` indica il posizionamento del titolo dopo la tabella
 - ▶ il titolo può contenere qualsiasi tag
- `<tr>`, definisce una riga della tabella
- `<th>`, definisce una cella di intestazione della tabella
 - ▶ di default in grassetto e centrata
- `<td>`, definisce il contenuto di una cella di una riga della tabella
 - ▶ di default allineata a sinistra e centrata verticalmente

Tabelle: attributi delle celle...

- align
 - ▶ left, center, right
- valign
 - ▶ top, middle, bottom
- colspan
 - ▶ numero delle colonne coperte da una cella
- rowspan
 - ▶ numero delle righe coperte da una cella
- nowrap
 - ▶ disabilita il *wrapping* (a capo automatico)
- Gli attributi di `<th>` e `<td>` sovrascrivono quelli di `<tr>`

Esempio

```
<html>
  <head><title>Le tabelle...</title></head>
  <body>
    <table border="1" width="640">
      <tr>
        <th width="100%" colspan="3">
          <h5 align="center">Appelli di Laurea 2015 – 2016</h5></th>
        </tr>
        <tr><th width="33%">Data sessione</th>
          <th width="33%">Termine per la presentazione</th>
          <th width="34%">Termine per la consegna della tesi</th>
        </tr>
        <tr><td width="33%" align="center">20 Luglio 2016</td>
          <td class="33%" align="center">20 Giugno 2016</td>
          <td class="34%" align="center">5 Luglio 2016</td></tr>
        </table>
      </body>
    </html>
```

Buone norme

- Non intrecciare i tag:
`Errore<i>moltograve</i>`
- Non inserire tag all'interno di un'ancora
- Essere sicuri della formattazione e dei collegamenti
 - ▶ esistono appositi servizi di validazione (es. `validator.w3.org`)
- Aggiornare i propri file
- Non legarsi ad un solo browser
- Commentare il codice
`<!-- commento -->`

I moduli

- Uno dei fattori che ha decretato il successo del Web è la possibilità di interazione
 - ▶ possibilità di inviare dati
 - ▶ possibilità di iscriversi a servizi di vario tipo (es. mailing list)
- Sintassi dei moduli:
 - ▶ `<form>`, definisce un modulo
 - ▶ `<input>`, definisce un campo (non ha tag finale corrispondente)
 - ★ a seconda del valore dell'attributo `type` assume forme diverse
 - ▶ `<select>`, definisce una lista di selezione
 - ★ `<option>`, definisce una scelta
(attributo `selected` per indicare la selezione di default)
 - ▶ `<textarea>`, definisce un'area di testo di più linee

Attributi del tag `form`

- `action`, URL a cui sottomettere il modulo
- `method`, metodo HTTP da usare per sottomettere il modulo
 - ▶ `get`, il contenuto è appeso all'URL
 - ▶ `post`, il contenuto è inviato in un campo dati a parte
- `enctype`, tipo di codifica del contenuto
 - ▶ i dati passati alla pagina di risposta, che si trova sul server, vengono codificati dal browser per evitare errori (es. gli spazi vengono convertiti in "+")
 - ▶ per l'invio di testo semplice non è necessario specificare la codifica (il valore di default è `enctype="text/plain"`)
 - ▶ per l'invio di immagini: `enctype="multipart/form-data"`

Attributi del tag `input`

- `type`
 - ▶ `text`, `password`, `checkbox`, `radio`, `submit`, `reset`
- `name`, nome simbolico
 - ▶ **obbligatorio** tranne che per `submit` e `reset`
- `value`, contenuto di default
 - ▶ per `submit` e `reset` indica l'etichetta
- `checked`
 - ▶ **solo per** `checkbox` e `radio`
- `size`
 - ▶ **solo per** `text` e `password`
- `maxlength`, massimo numero di caratteri
 - ▶ **solo per** `text` e `password`

Attributi del tag `select`

- `name`, nome simbolico
 - ▶ obbligatorio

- `size`
 - ▶ 1, mostra la selezione come menù a tendina
 - ▶ > 1, mostra la selezione come una lista a scorrimento, il valore indica il numero di linee visibili

- `multiple`, permette scelte multiple
 - ▶ forza la selezione ad essere mostrata come una lista a scorrimento

Attributi del tag `textarea`

- `name`, nome simbolico
- `rows`, numero di righe nell'area di testo
- `cols`, numero di colonne dell'area di testo
- il testo di default viene incluso tra il tag iniziale e quello finale

Esempio: inizio

```
<html>
<head>
  <TITLE>Un esempio di modulo</TITLE>
</head>
<body>
  <FORM ACTION="iscrizione.html" METHOD="POST">
    <P>Nome <INPUT NAME="nome" TYPE="text" SIZE="30">
  </P>
    <P>Cognome <INPUT NAME="cognome" TYPE="text" SIZE="30">
  </P>
    <P>Residenza
      <INPUT NAME="residenza" TYPE="text" SIZE="30"
        VALUE="Camerino">
  </P>
    <P>Telefono
      <INPUT NAME="tel" TYPE="text" SIZE="30"
        VALUE="0737">
  </P>
```

Esempio: fine

```
<P>
<SELECT SIZE="3">
  <OPTION SELECTED>Informatica (Laurea Triennale)
  <OPTION>Informatica (Laurea Magistrale)
  <OPTION>Boh?
</SELECT>
</P>
<P>Commenti
<TEXTAREA NAME="commenti" ROWS="7" COLS="27">
  Commento di default
</TEXTAREA>
</P>
<P>
  <INPUT TYPE="submit" VALUE="Submit"></P>
</FORM>
</body>
</html>
```

I Fogli di stile

- Un foglio di stile (*Cascading Style Sheets, CSS*) consente di definire un insieme di regole per determinare la veste tipografica di un documento HTML
- Permette di separare i contenuti dalla formattazione
- Vantaggio: impone una programmazione più chiara e facile
- Ogni regola ha la seguente forma:
 - ▶ `H1 { color: red; }`
- La semantica delle regole viene formalmente definita dal **W3C**, il consorzio che si occupa di *mantenere* gli standard del Web; versioni:
 - ▶ CSS 1
 - ▶ CSS 2
 - ▶ CSS 2.1
 - ▶ CSS 3

HTML e CSS

- I riferimenti al foglio di stile vengono posti nell'intestazione:

- ▶ direttamente all'interno del tag `<STYLE>`:

```
...  
<head>  
  
...  
<STYLE>  
H1 { color: red; }  
</STYLE>  
</head>  
  
...
```

- ▶ riferimento all'interno del tag `<STYLE>`:

```
<STYLE TYPE="text/css">  
    @import "foglio_di_stile.css";  
</STYLE>
```

- ▶ riferimento all'interno del tag `<LINK>`:

```
<LINK REL="stylesheet" TYPE="text/css"  
      HREF="foglio_di_stile.css">
```

Le regole

- Struttura di una regola: `H1 { color: red; }`

- ▶ `H1`: *selettore*
- ▶ `{ color: red; }`: *dichiarazione*
- ▶ `color`: *proprietà*
- ▶ `red`: *valore*

- Le regole sono raggruppabili per:

- ▶ *selettore*:

```
H1, H2 {color: red;}
```

- ▶ *dichiarazione*:

```
H1 {color: red; background:white;}
```

- ▶ *entrambi*:

```
H1, H2 {color: red; background:white;}
```

I valori

- Possiamo catalogare i possibili valori in:
 - ▶ colori
 - ▶ unità di lunghezza
 - ▶ valori percentuale
 - ▶ le URL

I colori

- Sono identificabili *direttamente* per mezzo dei nomi:
 - ▶ non tutti i browser implementano gli stessi colori;
 - ▶ non tutti i browser implementano ugualmente colori con lo stesso nome
 - ▶ è però identificabile un sottoinsieme coerente:
 - ★ aqua, black, blue, fucsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
- Sono identificabili per mezzo delle componenti RGB:
 - ▶ percentuale di colore (`rgb(25%, 25%, 25%)`)
 - ▶ interi da 0 a 255 (`rgb(25, 253, 87)`)
 - ▶ colori esadecimali (`#FF0054;`)

Unità di lunghezza

- Assolute:

- ▶ *Inches* (`in`)
- ▶ *Centimetri* (`cm`), $2.54\text{cm} = 1\text{in}$
- ▶ *Millimetri* (`mm`), $10\text{mm} = 1\text{cm}$
- ▶ *Punti* (`pt`), $72\text{pt} = 1\text{in}$
- ▶ *Pica* (`pc`), $12\text{pt} = 1\text{pc}$

- Relative:

- ▶ *em-height*, (`em`)
- ▶ *x-height*, (`ex`)
- ▶ *Pixel*, (`px`)

Selettori di classe e di ID

- I CSS permettono di definire regole (i selettori) che non sono direttamente associabili ad uno specifico tag
- Caratterizzano aspetti tipografici caratteristici di alcuni elementi di testo
 - ▶ esempio: messaggi di errore/attenzione
- `class` e `id` sono due attributi applicabili a tutti gli elementi HTML
 - ▶ se non si utilizza CSS, non hanno alcun senso e non modificano la presentazione della pagina

I selettori di classe

- La regola viene specificata inserendo come selettore un identificativo preceduto dal “.”:

```
.attenzione { font-weight: bold; }
```

- Al momento di utilizzare un tag viene aggiunto l'opportuno attributo CLASS:

```
<P CLASS="attenzione">...</P>
```

- Il selettore di classe è personalizzabile in funzione di un dato tag:

```
P.attenzione { font-weight: bold; }
```

```
EM.attenzione { font-style: italic; color: red; }
```

I selettori di ID

- La regola viene specificata indicando un identificativo che *dovrebbe* essere unico nel documento:

```
#intestazione { color: red; }
```

- Al momento di utilizzare il tag viene aggiunto l'opportuno attributo ID:

```
<H1 ID="intestazione">...</P>
```

- Stesso meccanismo dei selettori di classe ma con una sola differenza: in un documento HTML l'attributo ID è usato per identificare in modo **univoco** un elemento
 - ▶ la maggior parte dei browser non controlla l'unicità nell'uso degli ID

Elementi `span` e `div`

- Vengono usati per strutturare un documento HTML
- Sono elementi neutri
 - ▶ di per se non aggiungono niente al documento
- Hanno un ruolo fondamentale in CSS quando usati in combinazione con gli attributi `class` e `id`
 - ▶ Possono aggiungere effetti visivi a parti specifiche del testo all'interno del documento

Raggruppare con span

- Agisce come blocco all'interno di un elemento

`<P>`Tutti i cittadini hanno pari dignità sociale e sono eguali davanti alla legge, senza distinzione di sesso, di razza, di lingua, di religione, di opinioni politiche, di condizioni personali e sociali.`</P>`

- Per evidenziare in rosso le categorie distintive

`<P>`Tutti i cittadini hanno pari dignità sociale e sono eguali davanti alla legge, senza distinzione di ``sesso``, di ``razza``, di ``lingua``, di ``religione``, di ``opinioni politiche``, di ``condizioni personali e sociali``.`</P>`

regola CSS: `span.categoria {color: red;}`

Raggruppare con `div`

- Raggruppa una o più elementi di livello blocco

```
<div id="professori">
  <ul>
    <li>Emanuela Merelli</li>
    <li>Francesco Tiezzi</li>
    <li>Luca Tesei</li>
  </ul>
</div>
<div id="postDoc">
  <ul>
    <li>Mario Rossi</li>
    <li>Andrea Margheri</li>
    <li>Lucia Miccoli</li>
  </ul>
</div>
```

regole CSS:

```
#professori {background: blue;}
#postDoc {background: red;}
```



Principali novità:

- *Strutturazione di pagina semplificata*: nuovi elementi (header, nav, section, article, aside, footer) **anziché** div

<body>

```
<header>...</header>
```

```
<nav>...</nav>
```

```
<article>
```

```
  <section>
```

```
  ...
```

```
  </section>
```

```
</article>
```

```
<aside>...</aside>
```

```
<footer>...</footer>
```

</body>

- *Multimedialità*: nuovi elementi video e audio

Esercitazione

- Installare Apache Tomcat

- ① Scaricare Apache Tomcat versione 7

- ★ <http://tomcat.apache.org>

- ② Creare una cartella `programmazioneWeb` e decomprimervi il file

- ③ Impostare le variabili di ambiente:

- ★ `JAVA_HOME = ../jdk1.7.xxx`

- ★ `CATALINA_HOME = ../apache-tomcat-7.0.68`

- ④ Creare dentro la cartella `webapps` una cartella con il nome del gruppo e contenente la cartella `WEB-INF`, contenente a sua volta la cartella `classes`

- ⑤ Copiare i file HTML nella cartella col nome del gruppo

Esercitazione

- Creare una pagina Web del proprio gruppo usando un semplice editor (es. NotePad, TextEdit, ecc.) cercando di usare la maggior parte degli elementi HTML presentati
 - ▶ titolo, intestazione, paragrafi, liste, linee, immagini (prese dal Web), tabelle, ...
 - ▶ usare CSS per la veste grafica (sfondo, margini, link, ...)
- Creare una seconda pagina Web contenente una form con tutti i possibili tipi di input (es. inserimento dati di uno studente)
 - ▶ aggiungere dei link da una pagina all'altra
- Aggiungere dei link alle pagine Web dei vostri vicini di postazione
 - ▶ usare `ifconfig` (Mac OS X, Linux), `ipconfig` (Windows)