

Metodologie Informatiche Applicate al Turismo

8. HTML Parte II

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa

<http://pages.di.unipi.it/milazzo>

milazzo@di.unipi.it

Corso di Laurea in Scienze del Turismo

A.A. 2016/2017

Sommario

1 Tabelle

2 Forms

Le tabelle (1)

- Nei documenti HTML si possono inserire dati che hanno una naturale rappresentazione come tabelle tramite il tag `<table>`;

Le tabelle (2)

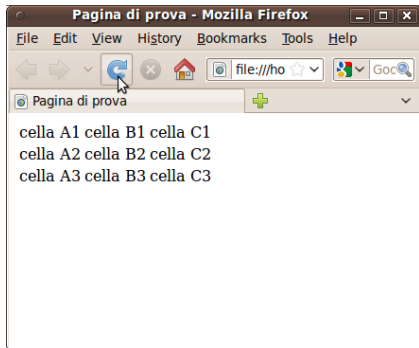
La struttura di base di una tabella è la seguente:

- L'intera tabella è racchiusa nel tag `<table>` che contiene:
- una lista di righe, ogniuna racchiusa nel tag `<tr>`, che a sua volta contiene:
- una lista di elementi (uno per ogni colonna) racchiusi nel tag `<td>`

```
<table>
<tr>
<td>cella A1</td>
<td>cella B1</td>
<td>cella C1</td>
</tr>

<tr>
<td>cella A2</td>
<td>cella B2</td>
<td>cella C2</td>
</tr>

<tr>
<td>cella A3</td>
<td>cella B3</td>
<td>cella C3</td>
</tr>
</table>
```



Le tabelle (3)

Alcune celle possono anche rappresentare titoli di righe o colonne:

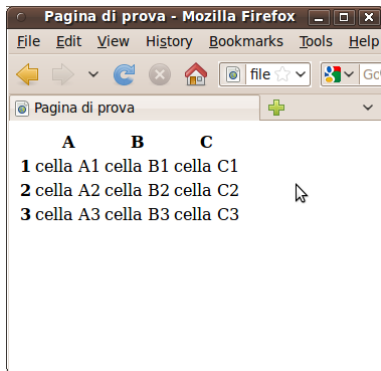
- per queste si usa `<th>` (table head) al posto di `<td>` (table data)
- la formattazione di queste celle è solitamente diversa dalle altre (a discrezione del browser)

```
<table>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>

<tr>
<th>1</th>
<td>cella A1</td> <td>cella B1</td>
<td>cella C1</td>
</tr>

.....

<tr>
<th>3</th>
<td>cella A3</td> <td>cella B3</td>
<td>cella C3</td>
</tr>
</table>
```



Le tabelle (4)

E' anche possibile specificare celle che spaziano su più righe o colonne:

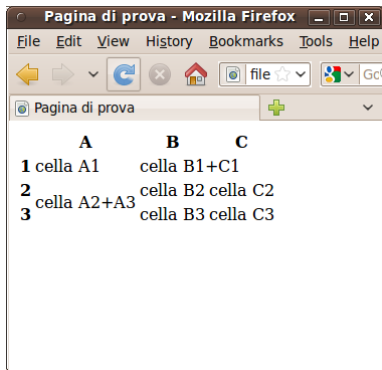
- usando gli attributi `rowspan` e `colspan` del tag `<td>`
- il valore degli attributi è il numero di righe o colonne su cui spaziare

```
<table>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>

<tr>
<th>1</th>
<td>cella A1</td>
<td colspan=2>cella B1+C1</td>
</tr>

<tr>
<th>2</th>
<td rowspan=2>cella A2+A3</td>
<td>cella B2</td> <td>cella C2</td>
</tr>

<tr>
<th>3</th>
<td>cella B3</td> <td>cella C3</td>
</tr>
</table>
```



Le tabelle: aspetti di formattazione (1)

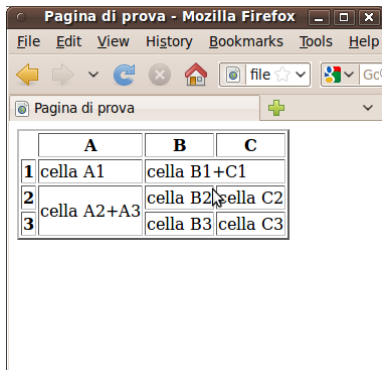
- E' possibile definire il bordo di una tabella tramite l'attributo border del tag <table>;
- Il valore dell'attributo è lo spessore del bordo espresso in pixel

```
<table border=2>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>

<tr>
<th>1</th>
<td>cella A1</td>
<td colspan=2>cella B1+C1</td>
</tr>

<tr>
<th>2</th>
<td rowspan=2>cella A2+A3</td>
<td>cella B2</td> <td>cella C2</td>
</tr>

<tr>
<th>3</th>
<td>cella B3</td> <td>cella C3</td>
</tr>
</table>
```



Le tabelle: aspetti di formattazione (2)

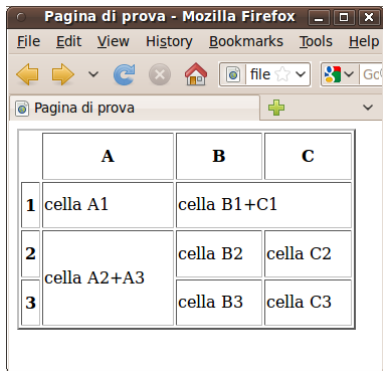
- Gli attributi `width` e `height` specificano larghezza e altezza della tabella (in pixel o in % sulle misure della finestra del browser)
 - ▶ Con un valore percentuale la tabella si ridimensiona automaticamente quando si ridimensiona la finestra del browser

```
<table border=2 width=95% height=200>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>

<tr>
<th>1</th>
<td>cella A1</td>
<td colspan=2>cella B1+C1</td>
</tr>

<tr>
<th>2</th>
<td rowspan=2>cella A2+A3</td>
<td>cella B2</td> <td>cella C2</td>
</tr>

<tr>
<th>3</th>
<td>cella B3</td> <td>cella C3</td>
</tr>
</table>
```



Le tabelle: aspetti di formattazione (3)

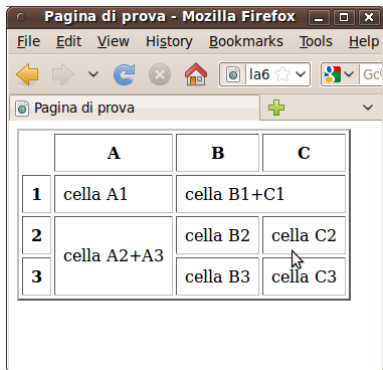
- L'attributo `cellspacing` specifica lo spazio tra le celle (in pixel)
- L'attributo `cellpadding` specifica lo spazio vuoto tra il bordo di una cella e il suo contenuto (in pixel)

```
<table border=2 cellspacing=3 cellpadding=8>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>

<tr>
<th>1</th>
<td>cella A1</td>
<td colspan=2>cella B1+C1</td>
</tr>

<tr>
<th>2</th>
<td rowspan=2>cella A2+A3</td>
<td>cella B2</td> <td>cella C2</td>
</tr>

<tr>
<th>3</th>
<td>cella B3</td> <td>cella C3</td>
</tr>
</table>
```



Le tabelle: aspetti di formattazione (4)

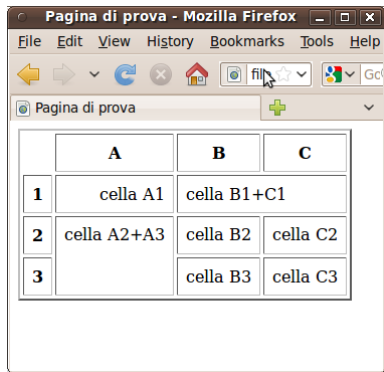
- Gli attributi `align` e `valign` possono essere usati in `<table>`, `<tr>` e `<td>` per specificare l'allineamento orizzontale e verticale del testo nella tabella, in una riga o in una cella;
 - ▶ Valori per `align`: `left`, `right`, `center`;
 - ▶ Valori per `valign`: `top`, `middle`, `bottom`;

```
<table border=2 cellspacing=3 cellpadding=8>
<tr>
<th><!--cella vuota in angolo--></th>
<th>A</th> <th>B</th> <th>C</th>
</tr>
```

```
<tr>
<th>1</th>
<td align="right">cella A1</td>
<td colspan=2>cella B1+C1</td>
</tr>
```

```
<tr align="center" valign="top">
<th>2</th>
<td rowspan=2>cella A2+A3</td>
<td>cella B2</td> <td>cella C2</td>
</tr>
```

```
<tr>
<th>3</th>
<td>cella B3</td> <td>cella C3</td>
</tr>
```



Le tabelle: aspetti di formattazione (4)

- Si può anche impostare la larghezza delle colonne e l'altezza delle righe di una tabella
 - ▶ Bisogna usare opportunamente gli attributi `width` e `height` del tag `<td>` (usano valori in pixel o in percentuale)
- E' bene ricordare che:
 - ▶ Una colonna è larga quanto la più larga delle sue celle, quindi si può impostare la larghezza di una colonna in una sola cella
 - ▶ Se non si specifica nessuna larghezza il browser adatterà le colonne ai contenuti e la larghezza della tabella verrà di conseguenza
 - ▶ Se si specifica solo la larghezza dell'intera tabella il browser imposterà la larghezza delle colonne "libere" (la cui larghezza non è specificata) di conseguenza
 - ▶ Se si specificano tutte le larghezze (tabella e tutte le colonne) è bene che la somma delle larghezze delle colonne sia pari alla larghezza della tabella
- Un discorso analogo vale per l'altezza delle righe.
 - ▶ l'altezza di una riga può essere specificata anche applicando l'attributo `height` al tag `<tr>`

Forms (1)

- Fino ad ora abbiamo visto metodi per visualizzare contenuti di varia natura (testo, immagini, liste, tabelle, ecc..)
- HTML prevede un metodo abbastanza semplice per inviare dati dal browser dell'utente al server: i **form**
- Un form non è altro che un modulo che può essere riempito dall'utente attraverso il browser
- Un form può essere fatto di caselle di testo da riempire, scelte multiple, bottoni, ecc...
- I dati raccolti tramite il form possono essere inviati ad un applicazione eseguita sul server oppure via email

Forms (2)

- Un form viene definito tramite il tag `<form>` che contiene, uno dopo l'altro, tutti gli elementi di cui è composto (caselle di testo, scelte multiple, ecc...)
- Un elemento di un form è definito tramite il tag `<input>` che ha un attributo `type` che viene usato per specificare di quale tipo di elemento si tratti
- I valori più comuni per l'attributo `type` sono i seguenti:
 - ▶ `text` - corrisponde a una casella di testo di una sola riga che può essere riempita dall'utente
 - ▶ `password` - simile a `text`, ma durante l'inserimento non visualizza i caratteri digitati (li sostituisce con asterischi o pallini)
 - ▶ `radio` - definisce una scelta singola tra un numero finito di alternative
 - ▶ `checkbox` - definisce una scelta multipla
- Altri tag consentono di definire caselle di testo multilinea (`<textarea>`) e caselle di selezione (`<select>`)

Forms (3)

- Il tag `<input>` serve in sostanza per assegnare un valore ad una variabile
- Il nome della variabile viene specificato tramite l'attributo `name`
- Nel caso di una casella di testo il valore assegnato alla variabile è il testo inserito dall'utente
- Nel caso degli strumenti di scelta tra varie alternative il valore da assegnare è inserito tramite l'attributo `value`
- Il tag `<input>` non prevede contenuto, quindi **non** è prevista la chiusura `</input>`

Forms (4)

```
<form>
  Username: <input type="text" name="user"> <br>
  Password: <input type="password" name="pwd">
</form>
```

Username:
Password:

```
<form>
  Titolo di studio<br>
  <input type="radio" name="titolo"
    value="elem">Licenza Elementare<br>
  <input type="radio" name="titolo"
    value="media">Licenza Media<br>
  <input type="radio" name="titolo"
    value="dipl">Diploma<br>
  <input type="radio" name="titolo"
    value="laurea">Laurea
</form>
```

Titolo di studio

- Licenza Elementare
- Licenza Media
- Diploma
- Laurea

```
<form>
  Patenti di guida<br>
  <input type="checkbox" name="patenteA"
    value="A">Patente A<br>
  <input type="checkbox" name="patenteB"
    value="B">Patente B<br>
  <input type="checkbox" name="patenti"
    value="altre">Altre patenti
</form>
```

Patenti di guida

- Patente A
- Patente B
- Altre patenti

Forms (5)

- Caselle di testo multi-linea possono essere definite tramite il tag `<textarea>`
- Il tag `<textarea>` prevede due attributi `rows` e `cols` che ne specificano le dimensioni della casella (numero di righe e di colonne)
- Il contenuto del tag viene visualizzato all'interno della casella di testo (ovviamente è modificabile)

```
<form>
  Inserisci un commento qui:
  <textarea rows="10" cols="30">
  No comment.
  </textarea>
</form>
```

Inserisci un commento qui:

No comment.

Forms (6)

```
Data di nascita (gg/mm/aaaa):  
<input type="text" name="gg"  
      size="2" maxlength="2">  
<input type="text" name="mm"  
      size="2" maxlength="2">  
<input type="text" name="aaaa"  
      size="4" maxlength="4">
```

Data di nascita (gg/mm/aaaa):

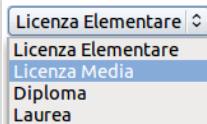
Note:

- Nei tipi "text" e "password" con l'attributo "maxlength" si può specificare il numero max di caratteri che possono essere inseriti
- Nei tipi "text" e "password" con l'attributo "size" si può specificare la larghezza (in caratteri) della casella di testo
 - ▶ la visualizzazione della larghezza non è uguale in tutti i browser
 - ▶ meglio usare il linguaggio CSS per questo aspetto
- Nel tipo "radio" bisogna usare lo stesso nome per tutte le opzioni che fanno parte della stessa scelta
 - ▶ Il browser capirà in questo modo quali sono le alternative tra le quali l'utente deve scegliere
 - ▶ Due gruppi di elementi di tipo "radio" distinti nella stessa pagina dovranno usare nomi diversi

Forms (7)

- L'esempio di scelta tra alternative visto prima con gli elementi di tipo radio può essere realizzato in alternativa con una casella di scelta multipla (drop-down menu)
 - ▶ Si usa il tag `<select>` che ha un attributo `name` per specificare il nome della variabile da assegnare
 - ▶ Il tag `<select>` contiene una lista di tag `<option>`, uno per ogni possibile valore tra cui scegliere
 - ▶ Ogni tag `<option>` ha un attributo `value` che ne specifica il valore corrispondente

```
<form>
  <select name="titolo">
    <option value="elem">Licenza Elementare</option>
    <option value="media">Licenza Media</option>
    <option value="dipl">Diploma</option>
    <option value="laurea">Laurea</option>
  </select>
</form>
```



Forms (8)

- Per ora abbiamo visto come raccogliere i dati dall'utente e associarli a variabili
- Vediamo ora come inviare tali dati ad una applicazione eseguita sul server (o via email)
- Per fare questo si utilizza un bottone “submit”, che si definisce tramite il tag `<input>` specificando il tipo `submit`
- La presenza del bottone `submit` richiede che nel tag `<form>` siano impostati un paio di attributi:
 - ▶ `action` - specifica l'URL dell'applicazione sul server (o l'indirizzo email) a cui inviare i dati
 - ▶ `method` - può essere impostato a `get` o `post` e specifica il tipo di messaggio HTTP da usare per inviare i dati.
 - ★ Nel caso di `get` i dati vengono aggiunti all'URL dell'applicazione
 - ★ Nel caso di `post` i dati vengono allegati nel corpo del messaggio HTTP

Forms (9)

- Quando un bottone submit è presente in un form e viene premuto dall'utente, tutti i dati inseriti vengono associati alle corrispondenti variabili e inviate all'applicazione (o indirizzo email) indicata dall'attributo `action`
- Nel caso il destinatario sia un'applicazione (e.g. PHP) si usa solitamente il metodo get, a meno che:
 - ▶ non si tratti di form molto complessi che renderebbero l'URL generata da get molto lunga
 - ▶ non si tratti di dati confidenziali: l'url di un pacchetto HTTP ha più visibilità del suo contenuto (e.g. rimane nella history del browser)

Forms (10)

- Nell'esempio seguente (in cui si usa il metodo get), inserendo Paolo e Milazzo nei due campi e premendo Invia si reindirige il browser alla pagina `elenco_telefonico.php?nome=Paolo&cognome=Milazzo`

```
<form action="elenco_telefonico.php" method="get">  
Nome: <input type="text" name="nome"/><br>  
Cognome: <input type="text" name="cognome"/><br>  
<input type="submit" value="Invia"/>  
</form>
```

Forms (11)

- Nell'esempio seguente (in cui si usa il metodo post), inserendo Paolo e Milazzo nei due campi e premendo Invia viene inviata una mail all'indirizzo (di fantasia) `dimmi_il_numero@paginegialle.it` contenente il seguente testo:
nome=Paolo
cognome=Milazzo
- Questo esempio mostra anche l'uso dell'attributo `enctype` di `<form>` che contiene il tipo MIME dei dati trasmessi. Omettendolo il contenuto dell'email sarebbe `nome=Paolo&cognome=Milazzo`

```
<form action="mailto:dimmi_il_numero@paginegialle.it"
      method="post" enctype="text/plain">
Nome: <input type="text" name="nome"/><br>
Cognome: <input type="text" name="cognome"/><br>
<input type="submit" value="Invia"/>
</form>
```

Forms (12)

Altri elementi che possiamo includere in un form tramite il tag `<input>`:

- Bottoni di che resettano il form (`type="reset"`)
- Elementi per l'upload di file (`type="file"`), con la possibilità di specificarne il tipo mime usando l'attributo `accept`
- Valori nascosti (`type="hidden"`)
- Altri bottoni (`type="button"`) che possono essere usati per eseguire parti dinamiche del documento, ossia script (che vedremo in seguito)

Uso di tabelle per strutturare i form

Per disporre gli elementi di un form in maniera ordinata nella pagina web si può utilizzare una tabella

- In particolare, una tabella consente di allineare gli elementi in verticale

```
<form action="prova.php" method=get">
<table>
<tr>
  <td align="right">Nome</td>
  <td>
    <input type="text" name="nome">
  </td>
</tr>
<tr>
  <td align="right">Cognome</td>
  <td><input type="text" name="cognome"></td>
</tr>
<tr>
  <td align="right">Età</td>
  <td><input type="text" name="eta"
    size="4" maxlength="3"></td>
</tr>
<tr>
  <td></td>
  <td><input type="submit" value="invia"></td>
</tr>
</table>
</form>
```

Nome	<input type="text"/>
Cognome	<input type="text"/>
Età	<input type="text"/>
	<input type="submit" value="invia"/>