Corso di Web Programming

4. HTML Parte II

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
http://www.di.unipi.it/∼milazzo
milazzo@di.unipi.it

Corso di Laurea in Informatica Applicata
A.A. 2010/2011

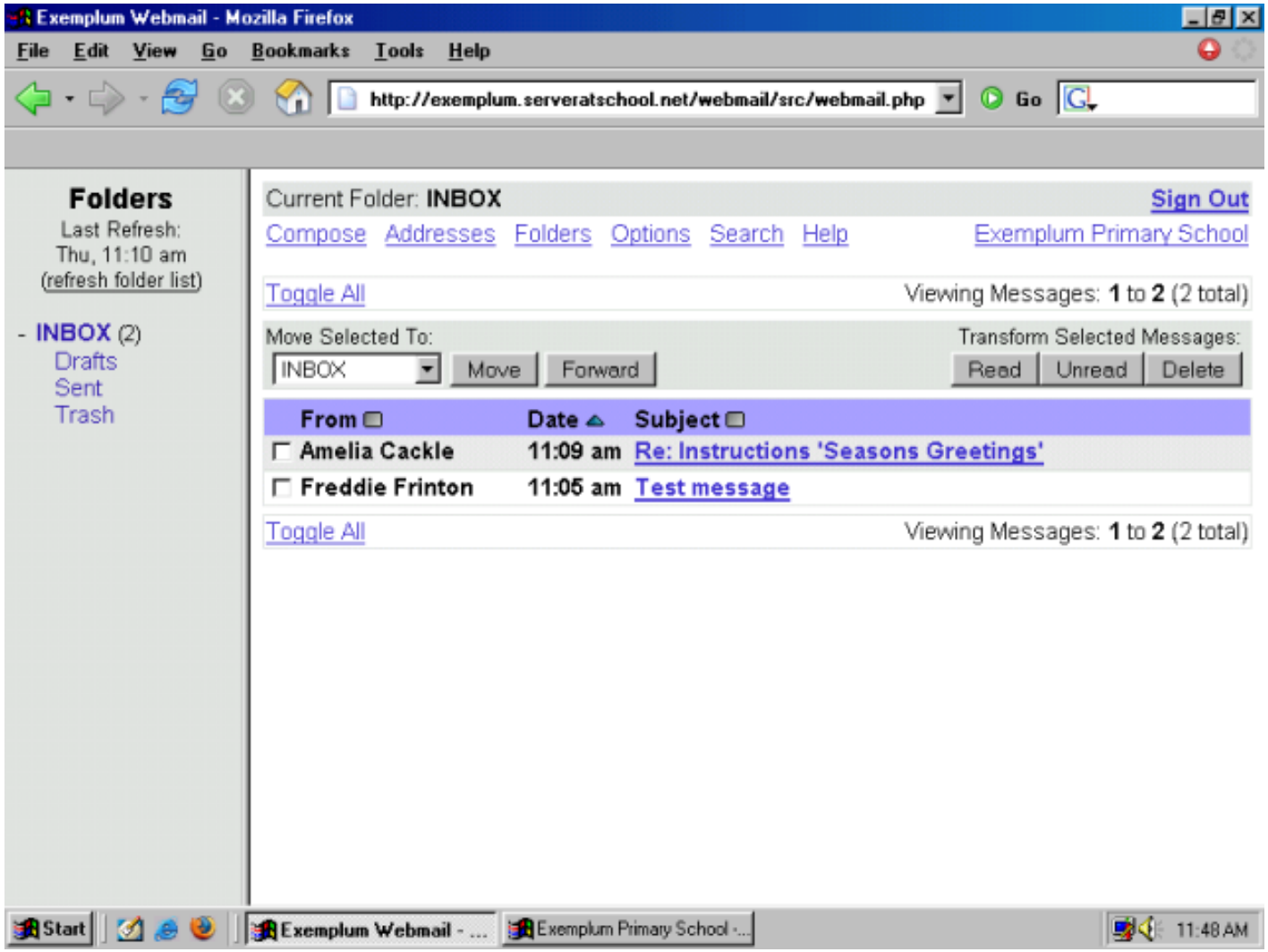
Sommario

Frames (1)

- HTML (versione 4) consente di visualizzare più documenti nella stessa finestra del browser attraverso l'uso di frames
- L'uso dei frame in HTML prevede la presenza di un documento principale (detto frameset document) che definisce la strutturazione in frames (sotto-aree) della finestra del browser e contiene i riferimenti delle pagine da richiamare nei vari frames
- Le pagine web richiamate nei vari frames sono normali documenti HTML
- E' prevista una minima forma di interazione tra i frames, tipicamente un link presente in un frame può essere aperto in un altro frame (tipico esempio: frame a lato della finestra con indice delle pagine del sito)

Frames (2)

Un classico esempio:



Frames (3)

- Il frameset document è un documento HTML che al posto del `<body>` ha un tag `<frameset>` che descrive la strutturazione in frames della pagina
- Il tag `<frameset>` può avere due attributi `rows` e `cols` che specificano le dimensioni di ogni riga e colonna. In questo modo si può strutturare la pagina come una tabella
- Le dimensioni possono essere espresse in pixel o in percentuale sulla dimensione della finestra del browser:
  - `<frameset cols="200,800">` indica una pagina con due colonne di larghezza 200 e 800 pixel
  - `<frameset cols="200,800,*">` indica una pagina con tre colonne in cui * rappresenta "tutto quello che avanza in larghezza"
  - `<frameset cols="30%,70%">` indica una pagina con due colonne
  - `<frameset cols="50%,50%" rows="50%,50%">` indica una pagina divisa in quattro parti uguali

Frames (4)

- Il contenuto del tag `<frameset>` è una sequenza di tag `<frame>`, uno per ogni frame della pagina
- Il tag `<frame>` ha due attributi principali: `name` che associa un nome al frame, e `src` che contiene l'url del documento HTML da caricare nel frame
- I vari frame vengono associati alle posizioni nella griglia definita dal tag `<frameset>` in maniera ordinata da sinistra a destra e poi dall'alto verso il basso
- All'interno del tag `<frameset>` ci può essere un tag `<noframes>` che contiene un body HTML da visualizzare se il browser non supporta i frames

Frames (5)

Un semplice esempio di documento HTML con frames:

```
<html>
  <head><title>Documento con frames</title></head>
  <frameset cols="300,*">
    <frame name="elenco" src="elenco.html"/>
    <frame name="notizie" src="http://www.repubblica.it"/>
  </frameset>
</html>
```

dove `elenco.html` potrebbe essere il seguente:

```
<html>
  <head><title>Elenco</title></head>
  <body>
    <h3>Elenco siti di notizie</h3>
    <ul>
      <li>Repubblica</li>
      <li>Corriere della Sera</li>
      <li>La Stampa</li>
    </ul>
  </body>
</html>
```

Frames (6)

Il risultato:

Frames (7)

- E' possibile fare in modo che i link presenti in un frame vengano aperti in un altro frame o nell'intera pagina
- Si usa l'attributo `target` di `<a>` specificando il nome del frame (attributo `name` di `<frame>`) in cui si vuole aprire il link
- Ad esempio, modifichiamo `elenco.html` come segue:

```
<html>
  <head><title>Elenco</title></head>
  <body>
    <h3>Elenco siti di notizie</h3>
    <ul>
      <li><a href="http://www.repubblica.it"
             target="notizie">Repubblica</a></li>
      <li><a href="http://www.corriere.it"
             target="notizie">Corriere della Sera</a></li>
      <li><a href="http://www.lastampa.it"
             target="notizie">La Stampa</a></li>
    </ul>
  </body>
</html>
```

Frames (8)

Il risultato:

Frames (9)

- I frame inline (o iframe) sono un modo per visualizzare all'interno di un documento HTML un documento HTML esterno
- Un iframe può essere definito tramite il tag `<iframe>`. Tale tag ha un attributo `src` che va associato all'URL della pagina da caricare nel frame
- Un iframe viene visualizzato all'interno del documento HTML che lo ospita più o meno come una figura
- Tramite gli attributi `width` e `height` è possibile impostare le dimensioni dell'iframe
- Tramite l'attributo `name` si può associare un nome all'iframe e fare in modo che i link del documento che lo ospita vengano aperti nell'iframe (come per i frames normali)
- Gli iframes vengono di solito visualizzati all'interno di un bordo che può essere rimosso settando l'attributo `frameborder` a 0

Frames (10)

Un esempio di uso di `<iframe>` in un normale documento HTML

```
<p>Ecco un inline frame:</p>
<iframe name="notizie" src="http://www.repubblica.it"
        width="300" height="200"/>
```

Risultato:

Frames (11)

E' bene notare che:

- I frames vengono in realtà sempre meno usati nelle pegine web:
  - dal punto di vista della strutturazione e disposizione dei contenuti è preferibile usare fogli di stile CSS
  - la separazione dei contenuti in documenti HTML diversi rende difficile far interagire i contenuti delle varie parti
- in HTML5 i frames non ci saranno più! (a parte i gli inline frames....)

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`

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

Titolo di studio
- Licenza Elementare
- Licenza Media
- Diploma
- Laurea

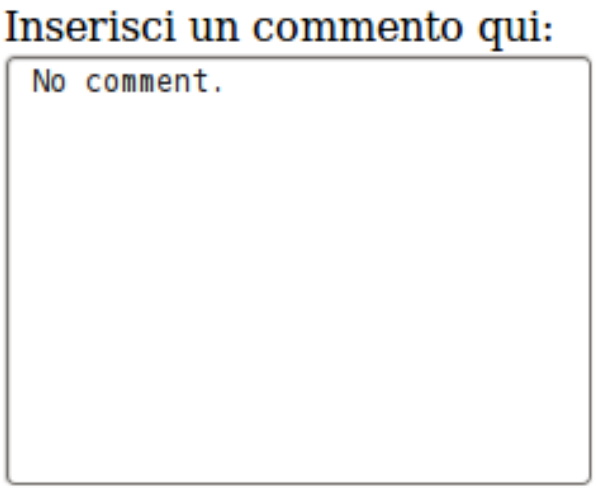
```
<form>
 Patenti di guida<br>
   <input type="checkbox" name="patenteA"
          value="A">Patente A</input><br>
   <input type="checkbox" name="patenteB"
          value="B">Patente B</input><br>
   <input type="checkbox" name="patenti"
          value="altre">Altre patenti</input>
</form>
```

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 visulizzato all'interno della casella di testo (ovviamente è modificabile)

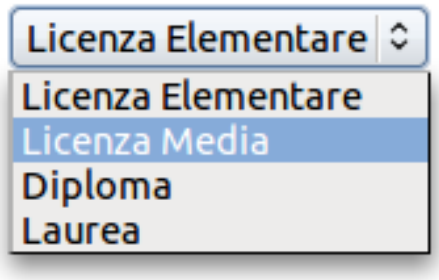
```
<form>
 Inserisci un commento qui:
 <textarea rows="10" cols="30">
 No comment.
 </textarea>
</form>
```



Forms (6)

- 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 possible 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 (7)

- 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>` speficicando 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 (8)

- 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 (9)

- Nell'esempio seguente (in cui si usa il metodo get), inserendo Paolo e Milazzo nei due campi e premendo Invia si redirige 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 (10)

- 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 (11)

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)

Versioni di HTML 4 (1)

- La versione correntemente più diffusa di HTML è la 4.01, di cui esistono tre varianti:
  - HTML 4.01 Strict non include i tag e gli attributi deprecati (quelli relativi ad aspetti di presentazione, ecc...) e non prevede l'uso di frames
  - HTML 4.01 Transitional include i tag e gli attributi deprecati, ma non i frames
  - HTML 4.01 Framset Come Transitional, ma con frames (da usare in pratica solo nei frameset document)
- Ogni variante è descritta formalmente in un documento detto DTD (Document Type Definition) fornito dal consorzio W3C

Versioni di HTML 4 (2)

- Ogni documento HTML dovrebbe essere preceduto da una dichiarazione `DOCTYPE` che specifica la variante di HTML utilizzata
- Utilizzare il `DOCTYPE` semplifica il lavoro del browser in alcuni casi (il browser può avere parser ottimizzati per le specifiche varianti)
- Una dichiarazione `DOCTYPE` è una linea di testo che deve essere posizionata all'inizio del documento (prima di `<html>`)
- Le dichiarazioni `DOCTYPE` per le tre versioni di HTML considerate (Strict, Transitional e Frameset) sono, nell'ordine, le seguenti:

```
<!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">
```

XHTML (1)

- è una diversa definizione di HTML4 in formato XML
- XML (eXtensible Markup Language) è una famiglia di linguaggi di markup basati su tag, ma destinati ad essere utilizzati dalle applicazioni
  - I linguaggi basati su XML sono molto meno permissivi dal punto di vista della sintassi rispetto ad HTML!
- In sostanza XHTML è pressocchè identico ad HTML, a patto che si rispettino alcuni vincoli sintattici. Ad esempio:
  - I tag e gli attributi devono sempre essere scritti in minuscolo
  - I tag (anche vuoti) devono sempre essere chiusi
    `<ul> <li>Uno <li>Due </ul>` $\Longrightarrow$ `<ul> <li>Uno</li> <li>Due</li> </ul>`
    `<br>` $\Longrightarrow$ `<br/>`
  - Le virgolette negli attributi non possono essere omesse
    `<table width=100%>` $\Longrightarrow$ `<table width="100%">`
  - La dichiarazione `DOCTYPE` e i tag `<html>`,`<head>`,`<title>` e `<body>` devono sempre essere presenti

XHTML (1)

- In XHTML abbiamo tre possibili dichiarazioni DOCTYPE che corrispondono a quelle di HTML 4.01

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Verso HTML5 (1)

- Negli ultimi tempi la progettazione delle nuove versioni di HTML ha preso una nuova piega
- Se fino a qualche tempo fa l'impressione era che si volesse convergere verso una sintassi con limiti più forti (in stile XHTML), ora si sta facendo "marcia indietro"
- La nuova versione 5 di HTML, anzichè cercare di contenere le bizzarrie sintattiche concesse dai vari browser, le includerà nella specifica del linguaggio
- Inoltre, HTML5 prevederà nuovi tag per una migliore strutturazione della pagina e orientati a includere nativamente una più ampia gamma di contenuti multimediali

Verso HTML5 (2)

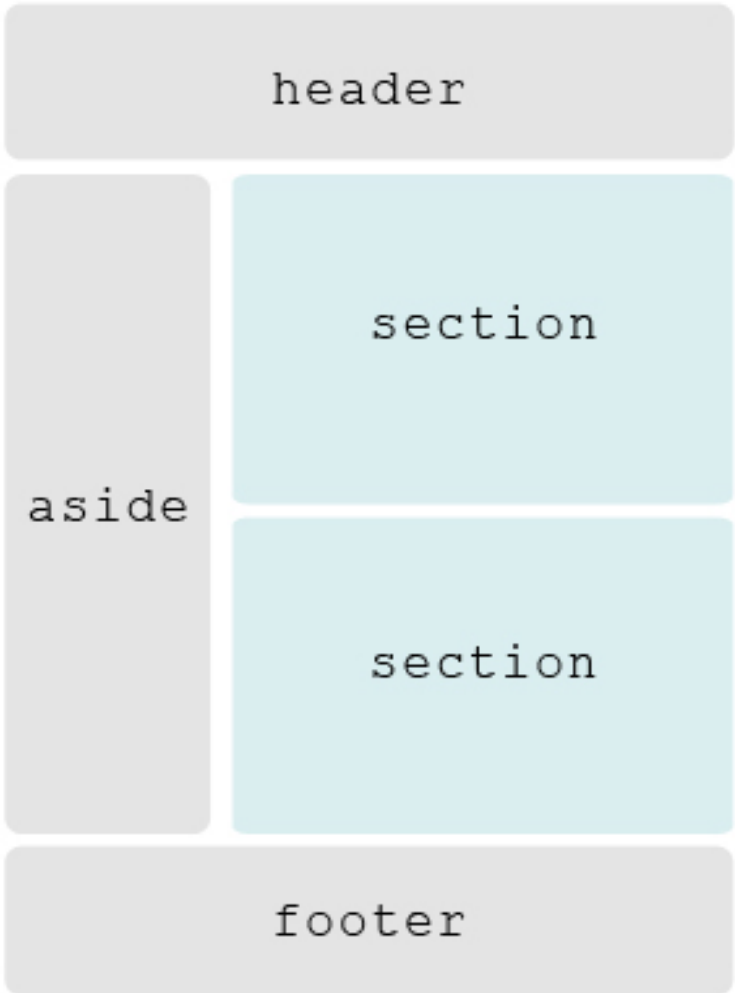
- Per quanto riguarda la sintassi generale, in HTML5 è possibili:
  - usare arbitrariamente il maiuscolo o il minuscolo
  - usare o meno le virgolette negli attributi
  - usare gli apici al posto delle virgolette negli attributi
  - evitare il tag di chiusura in alcuni casi particolari (ad esempio `<head>`,`<body>`,`<html>`,`<li>`,`<p>`
- L'elemento `DOCTYPE` da usare all'inizio del documento sarà semplicemente `<!DOCTYPE html>`

Verso HTML5 (3)

- I tag e gli attributi deprecati in HTML4 (ad esempio quelli relativi agli aspetti di presentazione e ai frames) non faranno parte di HTML5
- Ad esempio, sono stati rimossi:
  - I tag `<big>`,`<center>`,`<font>`,`<u>`,`<frame>`,`<frameset>`,...
  - Gli attributi `align`, `valign`, `bgcolor`, `border`,`cellpadding`, `cellspacing`, ...

Verso HTML5 (4)

- In HTML5 ci saranno nuovi tag per strutturare meglio la pagina web: `<header>`,`<section>`, `<article>`, `<aside>`, `<footer>`, ...
- Questi tag consentono di raggruppare megli i contenuti della pagina e consentono al browser di creare automaticamente un indice della pagina stessa (esempio nella prossima slide...)



Verso HTML5 (5)

```
<!DOCTYPE html>
<html>
<head><title>I diari di viaggio</title></head>
<body>
  <header><hgroup>
      <h1>I diari di viaggio:</h1>
      <h2>A spasso per il mondo alla scoperta di nuove culture:</h2>
    </hgroup></header>
  <section><h1>Europa</h1>
      <article>
        <h1>Giro turistico della Bretagna</h1>
        <p>lorem ipsum..</p>
      </article>
      <article>
        <h1>Cracovia e la Polonia misteriosa</h1>
        <p>lorem ipsum..</p>
      </article>
  </section>
  <section><h1>Africa</h1>
    <article>
      <h1>Alla scoperta del Kenya</h1>
      <p>lorem ipsum..</p>
    </article>
  </section>
  <p>tutti i viaggi sono completi di informazioni su alberghi e prezz
</body>
</html>
```

Verso HTML5 (6)

Verso HTML5 (7)

- HTML5 prevede anche nuove varianti del tag `<input>` che corrispondono a nuovi elementi e/o alla possibilità di specificare il formato dei dati da immettere
- Ad esempio, `<input type="email">` presenterà una casella di testo il cui contenuto inserito dall'utente dovrà avere la forma di un inirizzo email (altrimenti il browser segnalerà un'errore)
- Altri esempi di possibili valori dell'attributo `type`: `url`, `tel`, `datetime`, ...
- In casi particolari (ad esempio nei dispositivi mobili) il browser può usare speciali elementi di inserimento per questi nuovi tipi di input
  - ad esempio: un calendario per `datetime`, una tastiera telefonica per `tel`, ecc...

Verso HTML5 (8)

Infine, HTML5 fornisce strumenti per migliorare l'uso di strumenti multimediali e gli aspetti di dinamicità delle pagine web in (almeno) due modi:

- Tramite la definizione dei tag `<audio>` e `<video>` per i contenuti multimediali
- Tramite la definizione di un API utilizzabile tramite JavaScript e che consente di rendere più semplice l'uso nelle pagine web di funzionalità quali: grafica 2D, navigazione offline, drag-and-drop, ecc. . .