

Applicazioni web

Parte 1 HTML, XHTML

Alberto Ferrari

Sommario

- HTML
 - Elementi principali ed esempi
 - Collegamenti, liste e tabelle
- Form HTML e campi di input
 - Azione e metodo
 - Coppie nome/valore
 - Campi di input
- XHTML

Alberto Ferrari



World Wide Web

- Assieme alla posta elettronica, il World Wide Web (WWW o Web) è uno dei modi più diffusi di utilizzare la rete Internet
- Il Web permette agli utenti di Internet di pubblicare e accedere a documenti Html via Http
- Il Web si basa su due programmi
 - Il Web server
 - Il Web client (browser)

Alberto Ferrari



World Wide Web (storia)

- 1990: World Wide Web (Html+Http)
 - Sistema per la condivisione di informazioni in ipertesto
 - Sviluppato da Tim Berners-Lee
 - CERN (Centro Europeo per la ricerca nucleare)
- 1993: Mosaic, primo browser grafico
 - Marc Andreessen
- 1994: Netscape Navigator
- 1995: Internet Explorer

Alberto Ferrari

HTML

HyperText Markup Language

- Standardizzato dal W3C (<http://www.w3.org>)
 - Linguaggio per pubblicare informazioni
 - Una sorta di lingua madre che tutti i computer sono potenzialmente in grado di capire
- Html fornisce agli autori degli strumenti per:
 - Pubblicare documenti online con titoli, testo, tabelle, foto ecc.
 - Recuperare informazioni online attraverso collegamenti ipertestuali
 - Disegnare moduli per condurre transazioni con servizi remoti, da usare per ricerca di informazioni, prenotazioni, ordini ecc.
 - Includere nei documenti oggetti di media diversi

Alberto Ferrari

Html

Tag ed elementi

- Html dichiara *tipi di elementi* che permettono di strutturare i documenti (paragrafi, collegamenti ipertestuali, liste, tabelle, immagini ecc.)
- Ogni dichiarazione di tipo di elemento generalmente descrive tre parti: un *tag di apertura*, un *contenuto* e un *tag di chiusura*
 - Testo normale. `Testo in grassetto.` Normale.
- La maggior parte dei tag permette la definizione di *attributi*
 - Materiale didattico
` da scaricare`.
- I *tag semplici* non hanno un contenuto
 - ``

Alberto Ferrari



W3C

World Wide Web Consortium

- Nell'ottobre del 1994 Tim Berners Lee, padre del Web, fondò al MIT (Massachusetts Institute of Technology), in collaborazione con il CERN (il laboratorio dal quale proveniva), un'associazione di nome World Wide Web Consortium (abbreviato W3C), con lo scopo di migliorare gli esistenti protocolli e linguaggi per il WWW e di aiutare il Web a sviluppare tutte le sue potenzialità.
 - Nell'aprile del 1995 l'INRIA (Istituto Nazionale di Ricerca Informatica ed Automatica francese) divenne il primo membro europeo dell'organizzazione.
 - Nel 1996 approdò anche l'Asia, grazie alla Keio University of Japan.
 - Nel 2003 l'ERCIM (Il Consorzio Europeo di Ricerca in Informatica e Matematica) prese il ruolo di host europeo del W3C dall'INRIA (che è l'istituzione francese dell'ERCIM).

Wikipedia

Alberto Ferrari



Html - Struttura di una pagina

- Anatomia di una pagina Html
 - Tipo di documento – All'inizio e alla fine del file
 - Intestazione – Informazioni descrittive, come il titolo
 - Titolo – Deve essere nell'intestazione
 - Corpo – Parte principale della pagina, contenuto del documento

```
<html>
<head>
  <title>Hello, world</title>
</head>
<body>
  Hello, world.
</body>
</html>
```

[Alberto Ferrari](#)



Html

Tag di formattazione testo

- Grassetto e corsivo
 - `testo in grassetto`
 - `testo in corsivo`
- Titoli
 - `<h1>Il titolo più grande</h1>`
 - ...
 - `<h6>Il titolo più piccolo</h6>`
- Paragrafi
 - `<p>Questo è un paragrafo.</p>`

[Alberto Ferrari](#)



Html

Tag di formattazione testo

- Testo preformattato
 - `<pre>viene usato un font a larghezza fissa, gli spazi e i ritorni a capo sono preservati</pre>`
- Riga orizzontale di separazione
 - `<hr />`
- Interruzioni di linea
 - `Prima linea.
Seconda linea.`
- Commenti
 - `<!-- un commento html -->`

[Alberto Ferrari](#)

Html - Url

- Url è un acronimo per *Uniform Resource Locator* ed è un riferimento (indirizzo) per una risorsa sul Internet



- Il *nome della risorsa* è il suo indirizzo completo e dipende interamente dal *protocollo* – Per Http include:
 - *Nome dell'host* – Il nome della macchina su cui risiede la risorsa
 - *Numero di porta* – Num porta a cui collegarsi (di solito omissso)
 - *Nome del file* – Percorso completo della risorsa sulla macchina
 - *Frammento* – Riferimento ad una sezione con id univoco all'interno della risorsa, ad una porzione di testo (di solito omissso)
 - <http://www.ietf.org:80/rfc/rfc2732.txt>

Alberto Ferrari

Html - Collegamenti (link)

- Collegamenti ad altre pagine
 - `AOTLab`
- Collegamenti su immagini
 - ``
 - Se la risorsa (immagine o documento) si trova nella stessa cartella della pagina, allora basta un url relativo
- Collegamenti a indirizzi e-mail
 - Scrivi al `presidente` cosa pensi.

Alberto Ferrari

Html - Collegamenti

- Un collegamento può puntare ad uno specifico *elemento* della stessa pagina, o di un'altra pagina
 - Si deve assegnare all'elemento puntato un nome univoco con l'attributo *id*, oppure porre dove richiesto un elemento *anchor*
 - Si deve aggiungere un *frammento* alla url del collegamento

```
<html>
<body>
<p id="paragraph1">Il primo paragrafo. <a href="#paragraph2">Vai
  al secondo paragrafo.</a></p>
<p><a name="paragraph2">Il secondo paragrafo.</a> <a
  href="#paragraph1">Vai al primo paragrafo.</a></p>
<p>Il terzo paragrafo. <a
  href="http://somehost.com/anotherpage.html#frag1">Vai ad un
  frammento in un'altra pagina.</a></p>
</body>
</html>
```

[Alberto Ferrari](#)

Html - Liste non ordinate

```
<ul>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ul>
```

- First item
- Second item
- Third item

[Alberto Ferrari](#)



Html - Liste ordinate

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

1. First item
2. Second item
3. Third item

[Alberto Ferrari](#)



Html - Liste di definizione

```
<dl>
  <dt>First term</dt>
  <dd>Definition of first term.</dd>
  <dt>Second term</dt>
  <dt>Third term</dt>
  <dd>Definition of second and third term.</dd>
  <dd>Alternate definition of second and third term.</dd>
</dl>
```

First term
 Definition of first term.
Second term
Third term
 Definition of second and third term.
 Alternate definition of second and third term.

[Alberto Ferrari](#)

Html - Tabelle

- Le tabelle html permettono di mostrare dati
- Le tabelle sono definite dal tag `<table>`
- Il tag `<table>` contiene righe di celle, definite dal tag `<tr>`
- Ogni tag `<tr>` contiene celle di dati, definite dal tag `<td>`
- Le celle di titolo sono definite dal tag `<th>`
- Le righe procedono dall'alto verso il basso
- Le celle procedono da sinistra a destra
- Ogni cella di dati può contenere tutto ciò che si vuole – collegamenti, immagini, liste e anche altre tabelle

Alberto Ferrari

Html - tabella semplice

```
<table border="1">
  <tr>
    <td>northwest</td>
    <td>northeast</td>
  </tr>
  <tr>
    <td>southwest</td>
    <td>southeast</td>
  </tr>
</table>
```

northwest	northeast
southwest	southeast

Alberto Ferrari

Html - tabella con celle unite

```

<table border="1">
<caption><i>A test table with
merged cells</i></caption>
<tr>
  <th rowspan="2"></th>
  <th colspan="2">Average</th>
  <th rowspan="2">Red 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>

```

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Alberto Ferrari

Html Elementi di raggruppamento

- Gli elementi *div* e *span*, assieme agli attributi *id* e *class*, offrono un meccanismo generico per aggiungere struttura ai documenti
 - L'elemento *span* Lo `` è un contenitore generico. E' un elemento inline, non va a capo e continua sulla stessa linea del tag che lo include.
 - L'elemento *div* definisce un blocco di testo e va a capo, ma, a differenza del paragrafo, non introduce spazi prima e dopo la sua apertura
- Non impongono nessun vincolo di presentazione al contenuto
 - Si possono usare questi elementi congiuntamente a fogli di stile per definire le caratteristiche di presentazione dei documenti html

Alberto Ferrari



Markup semantico

Linee guida generali

- Definire un ricco vocabolario di classi semantiche
 - ... la cui presentazione può essere specificata in fogli di stile validi per tutto il sito web
 - Permette di scrivere documenti complessi senza introdurre elementi di presentazione
 - Prende il posto dei vecchi template di presentazione
 - La presentazione è ora definita nei fogli di stile, permettendo al markup di rimanere semantico
- Riferimenti per il markup di mozilla.org
 - <http://www.mozilla.org/contribute/writing/markup>
- Tecniche e linee guida per l'accessibilità dei contenuti web
 - <http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Alberto Ferrari



Form - Introduzione

- Un form html è una sezione di documento che contiene
 - Testo normale e markup
 - Elementi speciali chiamati *controlli* (checkbox, bottoni radio, menu ecc.)
- Gli utenti di solito “completano” questi controlli
 - Inserimento testo, selezione voci di menu ecc.
- Poi il form viene inviato (submit)
 - Elaborazione remota su web server

Alberto Ferrari



Form - action and method

- I form sono racchiusi dai tag `<form>` e `</form>`
- Il tag `<form>` ha
 - Un attributo *action* – url a cui inviare i dati del form
 - Un attributo *method* – metodo http da usare per sottomettere i dati del form (*get* o *post*)
- Esempio
 - ```
<form
action="http://www.myhost.com/myscript.cgi"
method="post">
```

Alberto Ferrari



## Form - Coppie nome/valore

- Ogni campo di input in un form ha:
  - Un **nome**, definito dall'attributo *name* (o *id*) del tag `<input>`, `<select>`, o `<textarea>`
  - Un **valore**, che l'utente imposta immettendo testo o cliccando col mouse
- I dati del form vengono inviati (submit) allo script sul server come un insieme di coppie nome/valore
  - Campi di testo vuoti sono inviati come una coppia nome/valore, in cui il valore è una stringa vuota
  - Checkbox e pulsanti radio non selezionati non vengono inviati

Alberto Ferrari



## Form - Tag per i campi

- **<input>**
  - Campi per l'immissione di testo, checkbox, bottoni radio, o bottoni normali
- **<select>**
  - Menu a discesa e box di selezione
- **<textarea>**
  - Campi per l'immissione di testo su righe multiple

Alberto Ferrari



## Form - Il tag input

- Il tag **<input>** crea diversi tipi di campi, a seconda del valore dell'attributo *type*
- **text** – Campo normale per l'immissione di testo (default)
- **password** – Identico a text, ma nasconde il testo inserito
- **checkbox** – Per semplici valori on/off
- **radio** – Pulsante per scegliere una tra diverse opzioni
- **submit** – Bottone per inviare i dati inseriti nel form
- **reset** – Bottone che riporta tutti i campi al valore iniziale
- **image** – Come submit, ma immagine come bottone
- **hidden** – ulteriori coppie nome/valore da inviare al server, ma non visualizzate all'utente

Alberto Ferrari

## Form – Attributi del tag input

- Insieme di attributi variabile a seconda dell'attributo *type*
- **text, password**
  - *value* – Valore di default per il campo
  - *size* – Dimensione visualizzata del campo
  - *maxlength* – Quantità di dati che si può inserire
- **checkbox, radio**
  - *value* – Valore del campo quando è selezionato; default = "on"
  - *checked* – Se il campo per default è selezionato
- **submit, reset**
  - *value* – Etichetta del bottone
- **image**
  - *src* – Url dell'immagine da visualizzare
  - maggior parte degli attributi del tag *img*

Alberto Ferrari

## Form - Esempio con tag input

```
<form action="http://myhost.com/myscript.cgi" method="post">

State: <input type="text" name="state" value="IT" size="2"
 maxlength="2" />

Password: <input type="password" name="password" />

<input type="checkbox" name="moreinfo" value="yes"
 checked="checked" />Send me more info.

Select your gender below:

<input type="radio" name="gender" value="F" />Female

<input type="radio" name="gender" value="M" />Male

<input type="submit" name="okbutton" value="OK, submit!"
 />

<input type="reset" value="Whoops - erase that" />
<input type="image" src="hand.gif" />
<input type="hidden" name="totalsofar" value="1290.65" />
</form>
```

Alberto Ferrari

## Form - Esempio con tag input

```

<form action="http://myhost.com/myscript.cgi" method="post">

State: <input type="text" name="state" value="IT" size="2"
maxlength="2" />

Password: <input type="password" name="password" />

<input type="checkbox" name="moreinfo" value="yes"
checked="checked" />Send me more info.

Select your gender below:

<input type="radio" name="gender" value="female" />

<input type="radio" name="gender" value="male" />

<input type="reset" value="OK, submit!" />
<input type="image" value="Whoops - erase that" />
<input type="hidden" value="" />
</form>

```

[Alberto Ferrari](#)

## Form - Il tag select

```

Choose your favorite color:
<select name="favcolor">
 <option>green</option>
 <option>aquamarine</option>
 <option selected="selected">emerald</option>
 <option>turquoise</option>
 <option>aqua</option>
 <option value="green2">green</option>
 <option value="green3">green</option>
</select>

```

[Alberto Ferrari](#)

## Form - Il tag textarea

- Il tag `<textarea>` crea aree per l'inserimento di testo su più linee, con barre di scorrimento
- Tutto ciò che si trova tra i tag di apertura e chiusura costituisce il contenuto iniziale del campo

```
<textarea name="stuff" rows="5" cols="10">Enter stuff
here</textarea>
```



Enter  
stuff  
here

Alberto Ferrari

## XHTML

- L'XHTML (acronimo di eXtensible HyperText Markup Language, Linguaggio di marcatura di ipertesti estensibile) è un linguaggio di marcatura che associa alcune proprietà dell'XML con le caratteristiche dell'HTML: un file XHTML è un pagina HTML scritta in conformità con lo standard XML.
- Il linguaggio prevede un uso più restrittivo dei tag HTML sia in termini di validità che in termini di sintassi, in modo da descrivere solo la struttura logica della pagina, mentre il layout e la resa grafica sono imposti dai fogli di stile a cascata (Cascading Style Sheets, CSS).
- L'XHTML è nato ufficialmente il 26 gennaio 2000 come standard del World Wide Web Consortium (W3C).

*Wikipedia*

Alberto Ferrari



## Versioni di XHTML

- **XHTML 1.0 Transitional:** nato per favorire la migrazione dalla vecchia versione HTML 3.2 o per uso insieme a link e frame in-line. Accetta come validi anche i tag HTML che sono stati definiti come deprecati in XHTML ed è tollerante rispetto ad alcune non conformità sintattiche.
- **XHTML 1.0 Strict:** rispetto alla versione Transitional non accetta i tag HTML definiti deprecati, non è tollerante a non conformità sintattiche e prevede controlli più rigorosi anche rispetto al valore di alcuni attributi dei tag (per esempio, l'attributo id deve avere valori univoci all'interno dello stesso documento).
- **XHTML 1.0 Frameset:** nato per motivi di compatibilità per suddividere la finestra visualizzata dal browser in diversi frame (sottofinestre), pratica un tempo diffusa ma ora deprecata dal World Wide Web Consortium.
- L'ultima versione di XHTML è la **XHTML 1.1**, che è una riformulazione dell' XHTML Strict, quindi ha una sola DTD. Non accetta i tag e gli attributi più criticati della precedente versione (come la struttura a frameset). Questa versione invece supporta i tag del cosiddetto markup «ruby», che consente di visualizzare i set di caratteri delle lingue orientali.

Alberto Ferrari

## Validazione (<http://validator.w3.org/>)

- Un documento per essere validato deve contenere un elemento chiamato DTD (Document Type Definition o Document Type Declaration), da posizionare preferibilmente all'inizio del documento.
- I tipi più comuni di DTD, e le relative sintassi dichiarative, sono:
- XHTML 1.0 Strict
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
- XHTML 1.0 Transitional
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- XHTML 1.0 Frameset
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`
- XHTML 1.1
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">`

Alberto Ferrari



## XHTML - Sintassi

- Scrivere Xhtml richiede una chiara sintassi html
- Alcune regole di sintassi xhtml
  - Chiusura degli elementi
  - Nomi di elementi e attributi in minuscolo
  - Valori degli attributi tra doppi apici
  - Minimizzazione degli attributi vietata
  - Attributo *id* sostituisce *name*
  - Elementi obbligatori

Alberto Ferrari



## XHTML - Chiudere gli elementi

- Un tag di chiusura per ogni tag di apertura
  - Tag `<p>` e `</p>` per aprire e chiudere paragrafi
  - Lo stesso vale per `<i>` e tutti gli altri tag
  - Per compatibilità con i vecchi browser, meglio aggiungere uno spazio prima del simbolo `"/` nei tag semplici (senza corpo)
- No
  - Paragrafo 1.`<p>`Paragrafo 2.
  - `<b><i>This is wrong</b></i>`
  - `<br>`
- Si
  - `<p>`Paragrafo 1.`</p><p>`Paragrafo 2.`</p>`
  - `<b><i>This is right</i></b>`
  - `<br />`

Alberto Ferrari



## XHTML - Lettere minuscole

- No
  - `<HTML>`
- Si
  - `<html>`
  
- No
  - `<a HREF="http://w3.org">`
- Si
  - `<a href="http://w3.org">`

Alberto Ferrari



## XHTML - Valori degli attributi

- No
  - `<a href=http://w3.org>`
- Si
  - `<a href="http://w3.org">`

Alberto Ferrari

# XHTML

## Minimizzazione degli attributi

### ■ No

- `<input checked>`
- `<input readonly>`
- `<input disabled>`
- `<option selected>`
- `<frame noresize>`

### ■ Si

- `<input checked="checked" />`
- `<input readonly="readonly" />`
- `<input disabled="disabled" />`
- `<option selected="selected" />`
- `<frame noresize="noresize" />`

Alberto Ferrari

# XHTML - Attributi id e name

- L'attributo *id* sostituisce l'attributo *name*
  - HTML 4.01 definisce un attributo *name* per molti elementi:
    - *a*, *applet*, *frame*, *iframe*, *img*, *map*
  - In XHTML l'attributo *name* è deprecato, sostituito da *id*
- No
  - ``
- Si
  - ``
- Transitoriamente, per interoperare con i vecchi browser, si può assegnare lo stesso valore a *name* e *id*
  - ``

Alberto Ferrari



## XHTML - Attributo lang

- L'attributo *lang* si applica a quasi tutti gli elementi *xhtml*
  - Specifica la lingua del contenuto di un elemento
  - L'attributo *xml:lang* è lo standard per xml, ma per compatibilità con I browser meglio definire entrambi
  - `<div lang="no" xml:lang="no">Heia Norge!</div>`

Alberto Ferrari



## XHTML - Elementi obbligatori

- La dichiarazione *DOCTYPE* deve essere presente
  - Non è considerata parte del documento *xhtml* stesso
  - Non è un elemento *xhtml* e non deve avere un tag di chiusura
- Gli elementi *html*, *head* e *body* devono essere presenti
- L'elemento *title* deve essere presente dentro *head*
- L'attributo *xmlns* nel tag *html* è necessario
- ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title goes here</title>
  </head>
  <body>
  </body>
</html>
```

Alberto Ferrari



XHTML

■ DOCTYPE

- perché un documento possa essere validato, deve contenere un elemento chiamato DTD (Document Type Definition o Document Type Declaration)
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

■ *xmlns*

- Namespace XML e lingua del documento nel tag html
- `<html xmlns="http://www.w3.org/1999/xhtml" lang="it">`

Alberto Ferrari