

XHTML

Introduzione a XHTML

- XHTML 1.0 è stata la prima grande modifica allo standard HTML 4.0 (rilasciato nel 1997).
- Le specifiche sono reperibili all'indirizzo <http://www.w3.org/TR/2000/REC-xhtml1-20000126>.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Obiettivi

- Maggiore flessibilità
- Possibilità di creare pagine *web* più ricche su una gamma di *browser* e *display* in espansione
- Modularità
- Compatibilità all'indietro

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: breve introduzione

- XML è un nuovo linguaggio creato a partire dallo standard SGML al fine di garantire maggior flessibilità e generalità di utilizzo per implementare diversi tipi di dati.
- È stato sviluppato dallo XML Working Group (originariamente noto come SGML Editorial Review Board) costituitosi all'interno del W3C nel 1996.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: obiettivi (1)

- Creare un linguaggio utilizzabile con facilità in Internet ma più flessibile di HTML.
- Compatibilità con applicazioni differenti.
- Facilità di progettazione ed elaborazione dei documenti.
- Necessità di disporre di un linguaggio che permettesse di descrivere e strutturare i dati.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: obiettivi (2)

- Comprensibilità elevata dei documenti grazie all'utilizzo di elementi aventi nomi non arbitrari o convenzionali ma che rispecchino il loro contenuto (ad esempio si potrebbe definire un elemento *author* in cui vada inserito il nome dell'autore di un libro, nel contesto di un sito creato per la vendita o la consultazione di libri), rendendo anche più facile la catalogazione dei documenti per eventuali ricerche.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: obiettivi (3)

- Permettere la pubblicazione online di documenti indipendenti dal tipo di dispositivo che vi avrà accesso.
- Permettere alle industrie di sviluppare protocolli indipendenti dalle piattaforme per lo scambio dei dati.
- Permettere agli sviluppatori di visualizzare l'informazione nel modo desiderato, grazie al supporto dei fogli di stile.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: definizione della sintassi (1)

- La sintassi di XML può essere **definita dall'utente a seconda del tipo di documento da creare.**
- E' possibile definire delle classi di documenti significative (chiamate DTD – Document Type Definition ossia definizione di tipo del documento) e associarvi particolari proprietà mediante un foglio di stile esterno, che può essere realizzato in formato CSS o XSL.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XML: definizione della sintassi (2)

- L'utente può scegliere se utilizzare delle DTD già create (ad esempio sono state create delle DTD per uniformare i documenti creati per particolari settori, come MathML-Mathematical Markup Language).

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Differenze con HTML

- È **obbligatorio**, all'inizio del *file*, dichiarare il tipo di modello di documento (DTD) utilizzato.
- Si deve includere il riferimento al XML *namespace*.
- E' *case-sensitive*, per cui i nomi dei *tag* e degli attributi devono essere scritti in lettere minuscole.
- Si devono indicare gli elementi vuoti.
- Tutti i valori degli attributi devono essere racchiusi tra virgolette.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Compatibilità con HTML

- Perché XHTML sia compatibile all'indietro con HTML sono necessari alcuni accorgimenti:
 - Gli elementi vuoti devono essere scritti con uno spazio tra il nome del *tag* e lo *slash* (ad es. `
`).
 - Nel caso in cui si utilizzino elementi senza contenuto è opportuno ricorrere alla notazione standard (ad es. `<head>` `</head>`) e non a un *tag* vuoto (`<head />`).

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Compatibilità con HTML (2)

- Non si possono omettere le sezioni `<head>` e `<body>`
- Il primo elemento della sezione `<head>` deve essere `<title>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Compatibilità con HTML (3)

- Non lasciare linee vuote o spazi tra i valori degli attributi perché differenti *user-agent* potrebbero interpretarli in modi diversi.
- Utilizzare sia l'attributo `id` sia `name` per definire il nome di un elemento. Infatti l'attributo `name` è stato rimpiazzato dall'attributo `id` ma non tutti i *browser* lo interpretano ed è quindi opportuno utilizzare notazione doppia.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Well Formed

- I documenti XHTML devono essere ben formati ossia:
 - Tutti gli elementi devono avere un tag di apertura e uno di chiusura.
 - Tutti gli elementi devono essere annidati in modo corretto.
 - Ci deve essere un unico elemento radice (`<html>`).

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Struttura di base

- La struttura di base di un documento XHTML è costituita da:
 - Dichiarazione versione XML (`<?xml version="1.0" ?>`)
 - Dichiarazione DOCTYPE (`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd)`
 - Sezione `<head>`
 - Sezione `<body>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

DTD XHTML

- **Strict DTD**, la versione base.
- **Transitional DTD**, supporta gli stessi *tag* della *Strict* ma con attributi aggiuntivi.
- **Frameset DTD**, aggiunge alle versioni precedenti il supporto di elementi (`<frame>` e `<frameset>`) per la creazione di *frame* all'interno della pagina.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

`<head>`

- La sezione `<head>` può contenere:
 - `<title>`, per dare un titolo al documento
 - `<style>`, per inserire informazioni di stile interne (CSS)
 - `<link>`, per inserire un link a un foglio di stile esterno
 - `<script>`, per inserire script
 - `<meta>`, per inserire meta informazioni sul documento
 - `<base>`, per definire l'URL di base del sito
 - `<object>`, per inserire oggetti

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

`<body>`

- La sezione `<body>` di un documento può contenere:
 - Titoli
 - Paragrafi
 - Liste
 - Tabelle
 - Immagini
 - Commenti
 - Codice di script

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Attributi di <body>

- Gli attributi di <body> sono:
 - `Background="URL"`, per definire un'immagine di sfondo
 - `Bgcolor="name | hex code"`, per definire un colore di sfondo
 - `text="name | hex code"`, per definire il colore di base del testo

Attributi di <body> (2)

- `link="name | hex code"`, per definire il colore dei link
- `alink="name | hex code"`, per definire il colore dei link attivi
- `alink="name | hex code"`, per definire il colore dei link visitati

Commenti

- I commenti sono introdotti (come in HTML, SGML e XML) dai caratteri speciali:
`<!-- commenti -->`

Elementi block

- Gli elementi block, ossia che definiscono dei blocchi all'interno del documento sono:
 - `<p>`
 - `<div>`
 - `<blockquote>`
 - `<pre>`
 - `<h1>...<h6>`
- Hanno un significato semantico

Paragrafi

- Il tag per introdurre un nuovo paragrafo è:
`<p> paragrafo </p>`
- Per l'allineamento si utilizza l'attributo:
`align="left | right | center"`

Livelli

- Il tag per introdurre livelli è:
`<div> scritte </div>`
- Per l'allineamento si utilizza l'attributo:
`align="left | right | center"`

Titoli

- I tag per definire i titoli sono:
`<h1> titolo </h1>`
`<h2> titolo </h2>`
`<h3> titolo </h3>`
`<h4> titolo </h4>`
`<h5> titolo </h5>`
`<h6> titolo </h6>`
- Possono definire titoli fino al 6 livello.

Citazioni

- Il tag per inserire una citazione è:
`<blockquote> ... </blockquote>`
- Viene generalmente utilizzato per inserire citazioni lunghe.

Testo preformattato

- Il tag per inserire un testo preformattato è:
`<pre> ... </pre>`

Elementi descrittivi

- Sono elementi utilizzati per descrivere a livello semantico il loro contenuto.
- Elementi inline

Abbreviazioni

- Il tag utilizzato per descrivere un testo di tipo abbreviazione è:
`<abbr> ... </abbr>`
- Ha un significato puramente semantico.
- Nella maggior parte dei browser non ha nessun effetto a livello di stile.

Citazioni

- Il tag utilizzato per inserire il riferimento di una citazione è:
`<cite> ... </cite>`
- Generalmente è visualizzato in italico.
- Per inserire invece il contenuto di una citazione breve si usa:
`<q> ... </q>`
- Nella maggior parte dei browser non ha nessun effetto a livello di stile.

Codice

- Il tag utilizzato per inserire codice è:
`<code> ... </code>`
- Generalmente visualizzato come carattere di macchina da scrivere (ex. Courier new)

Enfasi

- Il tag utilizzato per esprimere l' enfasi di un elemento è:
` ... `
- Generalmente visualizzato in italico.

Importanza

- Il tag utilizzato per sottolineare l' importanza semantica di un elemento è:
` ... `
- Generalmente è visualizzato in grassetto.

Elementi stilistici

- Elementi di tipo inline.
- Descrivono l'apparenza stilistica del loro contenuto.
- Non hanno alcun valore semantico (non danno significato al testo cui si applicano).

Elementi stilistici (2)

- Testo in grassetto:
` ... `
- Testo di dimensioni maggiori:
`<big> ... </big>`
- Testo italico:
`<i> ... </i>`
- Testo sottolineato
`<u> ... </u>`

Elementi stilistici (3)

- Testo come macchina da scrivere:
`<tt> ... </tt>`
- Testo di dimensione ridotta:
`<small> ... </small>`
- Testo "barrato":
`<strike> ... </strike>`

Elementi stilistici (4)

- Apici:
`^{...}`
- Pedici:
`_{...}`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Elementi stilistici (5)

- ``, permette di modificare l'aspetto del testo.
- Principali attributi:
 - `size="int"`, per definire le dimensioni (da 1 a 7)
 - `color="name | hex code"`, per definire il colore
 - `face="font-family"`, per definire il font da utilizzare

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Elementi stilistici (6)

- `<basefont>` permette di definire l'aspetto di base del testo per tutto il documento.
- Ha gli stessi attributi di ``.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Nuova riga

- Il tag per introdurre una nuova riga è:
`
`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Linea

- Il tag per inserire una linea orizzontale è:
`<hr />`.
- Principali attributi:
 - `align="right | left | center"`, per definire l'allineamento.
 - `noshade="noshade"`, per impostare la linea come non ombreggiata.
 - `size="pixel"`, per definire lo spessore della linea.
 - `width="int | perc"`, per definire la larghezza della linea.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Liste

- In XHTML è possibile creare 3 tipi di liste:
 - Ordinate
 - Non ordinate
 - Di definizione

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Liste ordinate

- Il tag che introduce una lista ordinata è:
` ... `
- Gli elementi della lista sono introdotti dal tag:
` ... `

Liste ordinate (2)

- `` Principali attributi:
 - `type="1 | a | A | i | I "`, per definire il tipo di numerazione degli elementi.
 - `start="int"`, per definire il numero da cui inizia la lista.

Liste non ordinate

- Il tag che introduce una lista non ordinata è:
` ... `
- Gli elementi della lista sono introdotti dal tag:
` ... `

Liste non ordinate (2)

- `` ha come principale attributo:
 - `type="square | disc | circle"`, per definire il tipo di marcatore della lista.

Liste di definizione

- Una lista di definizione è introdotta dal tag:
`<dl> ... </dl>`
- Il termine da definire è introdotto dal tag
`<dt> ... </dt>`
- che contiene la definizione, introdotta dal tag:
`<dd> ... </dd>`

Link

- I link vengono realizzati tramite il tag:
`<a> ... `
- Principali attributi:
 - `href="URL"`, per definire la risorsa a cui ci si collega
 - `name="text"`, per etichettare una sezione del documento
 - `target="name"`, nome del frame in cui caricare la risorsa o parola chiave

Link (2)

- Il valore dell'attributo **href** può essere:
 - Una URL esterna, per cui deve essere indicato il protocollo:
 - `href="http://www.di.unito.it"`.
 - `href="mailto:andres@di.unito.it"`.
 - Un file:
 - `href="index.htm"`.
 - Una sezione del documento:
 - `href="#sez"`.

Link (3)

- Il valore di **target** può essere una parola chiave:
 - `_blank`, per aprire la risorsa in una nuova finestra.
 - `_top`, per aprire la risorsa a piena pagina.
 - `_self`, per aprire la risorsa nello stesso frame in cui si trova il link.
 - `_parent`, per aprire la risorsa nel frame parent.

Immagini

- Le immagini da inserire in un documento XHTML devono essere in un formato supportato:
 - GIF
 - JPG
 - PNG
- Nota: XHTML e HTML non supportano l'utilizzo di immagini BMP.

Inserire un'immagine

- Il tag per inserire un'immagine è:
``
- Principali attributi obbligatori:
 - `src="URL"`, per indicare l'indirizzo dell'immagine da caricare
 - `alt="text"`, per definire un testo alternativo all'immagine

Attributi dell'immagine

- Principali attributi facoltativi:
 - `width="int | pixel"`, per definire la larghezza dell'immagine
 - `height="int | pixel"`, per definire l'altezza dell'immagine
- Questi attributi sono importanti per velocizzare il caricamento delle pagine in cui l'immagine è inserita.

Attributi dell'immagine (2)

- Altri attributi facoltativi:
 - `hspace="pixel"`, per definire un margine verticale.
 - `vspace="pixel"`, per definire un margine orizzontale.
 - `border="pixel"`, per definire lo spessore del bordo.

Attributi dell'immagine (3)

- `Align="left | right | middle | top | bottom | baseline"`, per definire l'allineamento orizzontale e verticale dell'immagine
- Attenzione! Per centrare l'immagine è necessario utilizzare il tag `<center>` o un livello!!!

Attributi dell'immagine (4)

- `usemap="#name"`, per indicare una sezione con i comandi di mappatura
- `ismap="ismap"`, per utilizzare l'immagine come image map server side

Image map

- mappa cliccabile che consente, all'interno di una singola immagine, di creare differenti aree di ritaglio, ad ognuna delle quali è possibile associare un link differente.
- Esistono due tipi di image map:
 1. Client side
 2. Server side

Coordinate di un'immagine

- Per trovare le coordinate di un'immagine è possibile:
 - caricarla utilizzando la normale sintassi (tag ``) con l'aggiunta dell'attributo `ismap`.

```
<a href="#">

</a>
```
 - aprirla con un programma di grafica o con un editor web.

Image map client side

- Per inserire nel codice un image map client side si utilizza la seguente sintassi:

```
<IMG SRC="menu.gif" usemap="#menu">
```

Image map client side (2)

- Il codice che associa ogni area ritagliata dell'immagine a un link è contenuto all'interno del tag

```
<map> ... </map>
```
- I comandi di mappatura sono introdotti dal tag :
 - `<area>` ... `</area>`

<area>

- Il tag <area> Principali attributi:
 - `shape="rect"` | `"circle"` | `"poly"` | `"default"`, per indicare la forma dell'area ritagliata
 - `alt="text"`, testo alternativo
 - `coords="x1,y1,x2,y2"` {per i rect, in alto a sin e in basso a destra} | `"x, y, r"` {per i circle, coordinate centro e raggio} | `"x1,y1, x2,y2, x3,y3, ..."` {coordinate dei vertici, in ordine}

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

<area> (2)

- Principali attributi:
 - `href="URL"`, per indicare l'URL associata all'area
 - `nohref="nohref"`, per impostare come non attiva una determinata area
 - `target="name"`, nome del frame in cui caricare la risorsa o parola chiave

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Tabelle

- Le tabelle vengono utilizzate soprattutto per realizzare documenti con un layout preciso.
- La tabella é introdotta dal tag `<table> ... </table>`
- Si costruiscono poi le righe `<tr> ...</tr>`
- Dentro le righe si inseriscono le celle, o colonne `<td> ... </td>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Esempio tabella

```
<TABLE>
<TR>
<TD> <TD> <TD>
<TD> <TD> <TD>
<TR>
<TD> <TD> <TD>
<TD> <TD> <TD>
<TR>
</TABLE>
```


Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

<table>

- Può contenere i tag:
 - `<caption>`
 - `<thead>`
 - `<tfoot>`
 - `<tbody>`
 - `<tr>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

<table> (2)

- Principali attributi:
 - `width="pixel|percent"`, per definire la larghezza della tabella
 - `border="int"`, per specificare il bordo della tabella
 - `cellspacing="int"`, per specificare lo spazio tra le celle (in pixel)
 - `cellpadding="int"`, per specificare lo spazio tra il bordo e il contenuto della cella (in pixel)

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

<table> (3)

- `align="left | center | right"`, per definire l'allineamento della tabella nella pagina
- `bgcolor="name|hex code"`, per specificare un colore di sfondo della tabella
- `summary="Text"`, per descrivere lo scopo o la struttura della tabella. Utile per migliorare l'usabilità del documento.

<tr>

- Principali attributi:
 - `align="left | center | right"`, per definire l'allineamento orizzontale del contenuto delle celle nella tabella
 - `valign="top | middle | bottom"`, per definire l'allineamento verticale del contenuto delle celle nella tabella

<td>

- Principali attributi:
 - `align="left | center | right"`, per definire l'allineamento orizzontale del contenuto della cella
 - `valign="top | middle | bottom"`, per definire l'allineamento verticale del contenuto della cella
 - `bgcolor="name|hex code"`, per definire un colore di sfondo della cella
 - `rowspan="int"`, per definire il numero di righe occupate dalla cella
 - `colspan="int"`, per definire il numero di colonne occupate dalla cella

<caption>

- `<caption>` permette di inserire una didascalia per la tabella.
- ha come principale attributo:
 - `align = "top|bottom|left|right"`, per specificare l'allineamento della didascalia rispetto alla tabella.

Strutture aggiuntive

- E' possibile dividere la tabella in:
 - Una sezione head `<thead>`
 - Una sezione body `<tbody>`
 - Una sezione foot `<tfoot>`
- Utile per accedere agli elementi della tabella con linguaggi di scripting.

Frame

- I frame sono un modo per suddividere una schermata in più sezioni ognuna delle quali contenente un documento HTML (o un'immagine o altro tipo di file) differente e indipendente.
- Nel codice XHTML il tag `<body>` viene sostituito dal tag `<frameset>`
- Ogni frame è introdotto dal tag:
`<frame> ... </frame>`

<frameset>

- Principali attributi:
 - `cols = "percent | pixel| **"`, per definire un numero di colonne in cui deve essere suddivisa la pagina
 - `rows = "percent | pixel| **"`, per definire un numero di righe in cui deve essere suddivisa la pagina
- E' possibile utilizzare contemporaneamente gli attributi `cols` e `rows` qualora si desideri creare una struttura speculare.

<frame>

- Principali attributi:
 - `src="URL"`, per definire il file o l'indirizzo web caricato nel frame
 - `name="text"`, per dare un nome logico al frame
 - `scrolling="yes" | "no" | "auto"`, per indicare la presenza o meno di scrollbar associate al frame
 - `noresize="noresize"`, per impedire il ridimensionamento del frame da parte dell'utente

<frame> (2)

- `marginwidth="int"`, per specificare la dimensione in pixel dei margini in larghezza del frame
- `marginheight="int"`, per specificare la dimensione in pixel dei margini in altezza del frame
- `frameborder="1|0"`, per specificare la presenza (1) o assenza (0) di bordo nel frame

Form

- I form sono dei moduli, inseriti all' interno del codice XHTML, per permettere all'utente di inserire dati.
- Sono introdotti dal tag:
`<form> ... </form>`

<form>

- Il tag `<form>` Principali attributi:
 - `action="URL"`, per specificare una risorsa a cui inviare i dati
 - `method="post | get"`, per indicare il metodo di invio dei dati
 - `name="text"`, per dare un nome logico al form
 - `target="name"`, nome del frame in cui caricare la risorsa o parola chiave

Campi di un form

- I campi di un form sono introdotti dal tag:
`<input> ... </input>`

<input>

- Principali attributi:
 - `type="text | password | file | submit | reset | button | image | checkbox | radio | hidden"`, per definire il tipo di campo
 - `name="text"`, per dare un nome logico al campo
 - `value="text | int"`, per associare un valore al campo.
 - `disabled="disabled"`, per disabilitare un campo

<textarea>

- Il tag `<textarea>` permette di introdurre un campo di testo di grandi dimensioni.
- Principali attributi:
 - `rows="int"`, per definire le righe del campo
 - `cols="int"`, per definire le colonne del campo

Menu di selezione

- E' possibile definire menu di selezione tramite il tag `<select> ... </select>`.
- Ogni opzione del menu è racchiusa all' interno di un tag `<option> ... </option>`

<select>

- Principali attributi:
 - `size="int"`, per definire il numero di opzioni visibili. Se l'attributo è omissso si crea un menu pop-up.
 - `multiple="true | false"`, per permettere la scelta multipla
 - `name="text"`, per associare un nome logico al menu

<option>

- Il tag `<option>` Principali attributi:
 - `name="text"`, per associare un nome logico all'opzione
 - `value="text | int"`, per associare un valore alla scelta.