

Linguaggio di markup

- Linguaggio di markup modellato per rendere esplicita una particolare interpretazione di un testo.
- Per esempio, tutte quelle aggiunte al testo scritto che permettono di renderlo più .
- Oltre a rendere il testo più leggibile, il markup permette anche di specificare ulteriori usi del testo.
- Con il markup per sistemi informatici, specifichiamo le modalità esatte di utilizzo del testo nel sistema stesso.
- I linguaggi di markup sono i linguaggi più opportuni per strutturare e marcare i documenti in maniera indipendente dall'applicazione, favorendo la riusabilità, la flessibilità e la apertura ad applicazioni complesse

HyperText Markup Language (HTML)

HTML linguaggio per la creazione di pagine Web (è un file di testo)

- Standard per la rappresentazione
- Descrive la struttura della pagina
- Non è un linguaggio di programmazione, è un linguaggio di markup
- Descrive dati e regole su come mostrarli utilizzando poche e semplici regole sintattiche

Caratteristiche

Le loro caratteristiche sono Iper testo, Multimedia, Ipermedia

Browser- 1

- Programma per la navigazione nel Web
- Richiede risorse attraverso il web (all'indirizzo richiesto) e le mostra nella finestra
- Legge e interpreta i documenti HTML, CSS, JavaScript, immagini sulla base di specifiche e standard
- Browser: Internet Explorer, Mozilla Firefox, Chrome, Opera, Safari

Browser- 2

Contiene tool per gli sviluppatori, Debug pagina, Analisi stili e sorgenti dei file, Visualizzazione HTTP request e response, Accesso a cookie e storage, Molto altro...

- Layout Engine – riceve input dal browser (URL bar, search box, mouse clicks and key presses) e li passa al rendering engine
- Rendering Engine – riceve il codice HTML e lo interpreta visualizzandolo a video. Ad esempio, testo in bold
- User Interface – controlli del browser, ad esempio, bottoni per andare avanti e indietro, bookmark
- JavaScript Engine – engine che si prende cura di parsare ed eseguire codice JavaScript per poi ritornare il risultato dell'esecuzione

- Network Layer – gestisce funzioni di rete, ad esempio, criptazione, richieste http e tutte le configurazioni di rete
- Storage – porzione di memoria dove il browser memorizza cached file, cookie e oggetti e dati create con JavaScript
- Operating System Interface – componente che interagisce con il sistema operativo per disegnare diversi elementi e per gestire la finestra (chiudi, massimizza, minimizza)

Browser- 3

Ha una mentalità multi-browser ed il suo obiettivo è rendere visibile la propria pagina su tutti i browser ed rendering

Scrivere una pagina Web

- Creazione file .html o .htm
- Editor Testuale vs Editor Visuale
- Visualizzazione pagina tramite browser

Storia degli standard web

→ **Working Draft (WD)**: Documento che il W3C pubblica per ottenere una revisione dalla comunità (include W3C Member, pubblico e altre organizzazioni tecniche)

→ **Candidate Recommendation (CR)**: Documento revisionato in dettaglio che il W3C ritiene soddisfatti i requisiti tecnici del Group. W3C pubblica una Candidate Recommendation per raccogliere esperienza implementativa.

→ **Proposed Recommendation (PR)**: Un technical report maturo che, dopo una dettagliata revisione che valuta l'implementabilità e la correttezza della soluzione, W3C invia al W3C Advisory Committee per un endorsement finale.

→ **W3C Recommendation (REC)**: Una specifica o insieme di guideline che, dopo aver raccolto un ampio consenso, ha ricevuto l'endorsement dei W3C Member e del W3C Director. W3C raccomanda il deployment e l'utilizzo della Recommendation. W3C Recommendation sono simili a standard pubblicati da altre organizzazioni.

Storia degli standard web

- Ultima parte del 1991: prima descrizione di HTML (HTML tag)
- 1993-94: HTML 2.0 pubblicato
- Marzo 1995: HTML 3.0 pubblicato come internet draft
- Dicembre 1997: HTML 4.0 is published by the W3C
- Febbraio-Marzo 1998: XML 1.0 is published
- Dicembre 1999-Gennaio 2000: ECMAScript 3rd Edition, XHTML 1.0 (HTML tag riformulati in XML) e HTML 4.01 recommendation pubblicate
- Maggio 2001: XHTML 1.1 recommendation pubblicata
- Agosto 2002: XHTML 2.0 first working draft rilasciato
- Dicembre 2002: XHTML 2.0 second working draft pubblicato
- Gennaio 2008: First W3C working draft di HTML5 pubblicato

Struttura della pagina HTML

HTML- Tag

- Tag sono marcatori che identificano porzioni di testo
- Permettono la personalizzazione della pagina, e di identificare e processare le porzioni di testo contrassegnate da una certa marcatura
- Nome tag = nome funzione
- Struttura tag: Tag contenuto tra parentesi triangolari <Tag di inizio, tag di fine>
Contenuto del tag: contenuto <tag attributi > (contenuto que puede ser omitible)
</tag>

Struttura della pagina – 1

- Riga di intestazione <!DOCTYPE>...
- Indichiamo le specifiche W3C adottate
- Dice al browser quale linguaggio è stato usato per il rendering

Struttura della pagina- 2

Il tag indica l'inizio di una pagina HTML } Tutto ciò compreso all'interno del tag è il codice HTML <html> </html>

Struttura della pagina- 3

Documento HTML diviso in due

Testa:

(resumen)

- La testa contiene informazioni non immediate
- Descrive come il documento deve essere letto e interpretato
- Contiene meta-tag, script, stili
- Racchiuso tra il tag <HEAD>

(diapositiva principal)

Contiene informazioni/metadati relativi al documento. Ad esempio, titolo, keyword, altri dati non considerati contenuto del documento.

Browser di solito non presentano elementi che compaiono nella testa (HEAD)

Rendono disponibili queste informazioni attraverso altri meccanismi

Elemento title

Elemento <title>→Indica il titolo della pagina

Viene visualizzato in alto a sinistra nei browser più datati o nella scheda nei browser più moderni.

Metadati

Elemento <meta>→ Definisce informazioni relative alla pagina piuttosto che contenuto vero e proprio. Contiene tre attributi principali: Name, content, http-equiv

- Name→ contiene il nome della proprietà ,Author, Description, Copyright, Generator, Language, Keywords
- Content→ contiene il valore della proprietà
- http-equiv→ specifica caratteristiche dell'HTTP response header

esempi: <META NAME ="author" CONTENT="Claudio">

Meta Tag- Keywords

- Indica alcuni informazioni sul contenuto del sito
- Separate da virgola, punto e virgola o spazio
- Sono chiavi di ricerca
- Motore di ricerca ricava chiavi di indicizzazione
- Attenzione nella scrittura delle keywords: Evitare termini generici e particolari, riportare termine italiano e inglese, singolare e plurale
- Evitare di ripetere le stesse parole e usare Keyword astute

Attributo HTTP-EQUIV

Informazioni sulla comunicazione server-browser → può essere usato al posto dell'attributo name

Attributo http-equiv fornisce un header HTTP per informazioni/valore dell'attributo content → Può essere usato per simulare un HTTP response header.

HTTP server usano questo attributo per ottenere informazioni circa l'HTTP response header. Alcuni valori: content-type, default-style, refresh.

Corpo

(resumen)

Contiene il documento vero e proprio → tutti i tag per la realizzazione del sito Web
Racchiuso tra il tag <BODY>

(diapositiva principal)

Il corpo di un documento contiene il contenuto vero e proprio

Il browser presenta il contenuto in diversi modi:

- Browser visuali: è una tela su cui disegnare testi immagini colori...
- Agenti audio: lo stesso contenuto è parlato

Contiene tutte le informazioni per la visualizzazione della pagina.

Siccome gli style sheet sono il modo suggerito per specificare come verrà presentato un documento gli attributi del BODY sono stati deprecati.

Indentazione

Buona norma utilizzare caratteri di tabulazione → Aumenta leggibilità && Diminuisce tempi di modifica (Quiere decir q tabulemos bien el codigo para que sea mas legible)

Annidamento

Caratteristica HTML: tag annidati Formattazioni successive Los tags se pueden anidar

Commenti

Aumenta la leggibilità del codice. Buona norma commentare SOLO le parti significative.

Permettono insieme all'indentazione di orientarsi all'interno di un grosso documento.

<!-- =inizio commento

-- > = fine commento

<!--questo è un commento-->

Maiuscolo o minuscolo → Case insensitive → Consigliabile usare carattere minuscolo

Non è sensibile agli spazi && Non è sensibile alle linee vuote

Tag interni al body

Elementi inline vs block

Alcuni elementi HTML nel body sono "block-level" altri "inline" ("text level")

Content model:

- Elementi block-level possono contenere elementi inline e altri elementi block-level
- Elementi inline contengono solo dati e altri elementi inline

Formatting:

- Elementi block-level sono formattati in maniera differente dagli elementi inline
- Elementi block-level iniziano su una linea nuova, elementi inline no

Testo – Tag predefiniti

Tag-contenitori di testo (elementi block-level, unica eccezione)

<h1>,<h2><p><div>

Titoli – Heading

Sono previste sei grandezze predefinite per i titoli.

Si va da <h1> che è la grandezza maggiore fino ad <h6> che è la grandezza minore.

Paragrafo

<p> Unità base di suddivisione del testo → lascia una riga vuota prima e dopo il testo

Blocco testo

<div> è un blocco contenitore.

Blocco testo va a capo ma non lascia righe di spazio

Non impone rappresentazioni di default (a parte il fatto di essere un elemento block-level)

Struttura della pagina con tag <div>



È uno tra i tag più utilizzati nella creazione di pagine Web, soprattutto nella creazione di layout



Fornisce un vero e proprio elemento strutturale della pagina



Suddivide gli spazi in zone per progettare il sito in modo semplice e dettagliato



Molto utile quando usato insieme a fogli di stile

Facciamo subito un esempio: immaginando di dover costruire un sito pensiamo a come esso debba essere strutturato, tipicamente abbiamo

1. Un contenitore (container)
2. Una parte alta (header)
3. Un corpo centrale (middle)
4. Un menù (navigation)
5. Una parte bassa (footer)

6. Spesso anche un pannello laterale (sidebar)



```
<div id="container">
  <div id="header">
    <div id="navigation"></div><!--#navigation-->
  </div><!--#header-->
  <div id="main"></div><!--#main-->
  <div id="sidebar"></div><!--#sidebar-->
  <div id="footer"></div><!--#footer-->
</div><!--#container-->
```

Contenitore

Contenitore generico • Può essere annidato • Elemento inline (continua sulla stessa riga del tag che lo contiene (ad es.)) • Non va a capo • Non impone rappresentazioni di default (a parte il fatto di essere un elemento inline)

Testo – Stili

Due tipi di stili

- Fisici: definiscono lo stile grafico (tutti elementi inline)

<bold>bold :formatta il testo in grassetto

<i>italic: formatta il testo in corsivo

<u>: sottolinea il testo

<strike> testo barrato usato per correzioni (tachado)

- Logici: forniscono informazioni sul ruolo svolto dal testo
- Opzionale: impongono uno stile grafico

- `<abbr>` abbreviazione: non comporta nessun cambiamento grafico
- `<address>` indirizzo, `<cite>` citazioni: testo in corsivo
- `<samp>` esempio: testo a spaziatura fissa

Testo – Font

Font = colore, dimensione e tipo di carattere:

- Carattere predefinito Times new roman
- Poco leggibile
- Meglio Verdana, Arial o Helvetica

E' buona norma:

- Utilizzare caratteri sicuri (sicuramente visualizzabili)
- Non indicare un solo carattere
- Solo come ultima spiaggia sarà usato il Times

Dimensione del testo

Attributo size del tag font

- Due modi per impostare la dimensione
 - Valore tra 1 e 7
 - Valori relativi alla dimensione base del font

Dimensione del font di base • Tag `<basefont>` permette di cambiare il font di base

Elenchi

`<elenco>`

`<elemento>` primo elemento

`<elemento>` secondo elemento

`</elenco>`

- Chiusura elemento opzionale
- Sintassi degli elenchi di definizione leggermente diversa

Elenchi ordinati: (Numerazione degli elementi)

- Numerazione progressiva
- Tag elenco ordinato `` (ordered list)

- Tag elemento (list item)

Elenchi non ordinati:

- Nessuna numerazione
- Elenchi puntati
- Tabella tag (unordered list)

Elenchi di definizioni

- Elenco di definizione <dl> (definition list)
- Termine da definire <dt> (definition term)
- Definizione del termine <dd> (definition description)

```

<p> Titolo
<dl>
  <dt>&ltp&gt
    <dd>apertura paragrafo
  <dt>&ltdiv&gt
    <dd>apertura blocco di testo
  <dt>&ltspan&gt
    <dd>apertura elemento inline
  Altri tag...
</dl>
▶ Esempio13.html

```

Iperestualità (Link)

Link

- Testi vs ipertesti
- Iperestesti caratteristica che ha reso grande il web
- Link = ponte tra un testo e un altro, sono formati da due componenti
 - Il contenuto (testo o immagine) che nasconde il collegamento
 - La risorsa puntata
- Sintassi
 - Clicca qui per visualizzare il collegamento.
 - La testa del link è qui
 - La coda è indirizzo
- Coda → Pagina HTML, immagine, documenti, altri file

Destinazioni

- Pagina HTML: aperta nel browser
- Immagine (.gif, .jpeg): aperta nel browser
- Altri documenti (.doc, .pdf): → Visualizzata nel browser ed Richiesta di salvataggio
- Altri file (.zip, .exe) richiesta di salvataggio

Percorsi assoluti: Viene indicato il percorso per esteso

- Leggi l'esempio ` qui `
- Nel caso di file sull'hard disk
- Leggi l'esempio ` qui `

Percorsi relativi: Fanno riferimento alla posizione del file a cui si riferisce al link a partire dalla posizione della pagina che stiamo sviluppando

- `Clicca qui` → Indica al browser di cercare il file nella stessa directory
- `Clicca qui` → Indica al browser di cercare il file nella sottodirectory sottodir
- `Clicca qui` → Indica al browser di cercare il file nella directory Padre

Considerazioni

- **Nomi dei file**
 - Evitare gli spazi (`mio_file.html`)
 - Ricordarsi che maiuscole e minuscole fanno la differenza
- `` non funziona quando caricate la pagina sul vostro sito
- Allora diventerà
 - ``

Link interni o ancore

- Permette di creare un indice
- Ancora
 - ` àncora `
- Riferimento all'ancora
 - ` vai all'àncora `
- Riferimento ad inizio pagina
 - ` torna su `
- `ancora.html`

Tablelle

Definizione tramite l'elemento `<table>`

- Ogni riga definita con il tag <tr> contenuto in table
- Ogni cella definita con il tag <td> contenuto in <tr>
- Una riga può essere anche divisa in table heading usando l'elemento <th> Di solito mostrati centrati e in grassetto
- Elemento <caption> per definire la caption della tabella

Bordi

Attributo border → Se non specificato nessun bordo <table border="">

Celle collassate in una riga/colonna

→ Attributo rowspan

```
<th rowspan="2">Telephone</th> <td rowspan="2">Telephone</td>
```

→ Attributo colspan

```
<th colspan="2">Telephone</th> <td colspan="2">Telephone</td>
```

FORM

Introduzione

Form usate per collezionare input utente, basato sull'elemento. Principio del formulario. Specifica diversi elementi: input, checkbox, radio button, submit button...

Elemento input

Elemento più importante → Diverse varianti che dipendono dall'attributo type

- Text: definisce input testo normale
- Radio: definisce un input radio button (per selezionare una tra diverse scelte)
- Submit: definisce un bottone submit (per sottomettere la form)

Text input

- <input type="text">
 - Definisce un campo testo di una linea
 - Esempio1.html
 - <input type="password"> un tipo speciale di campo testo dove il valore inserito viene nascosto

```
<form>
First name:<br>
<input type="text" name="firstname">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

Text Area

- `<textarea>`
 - Definisce un campo testo a linee multiple
 - Esempiof1.html

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

Radio Button input

- `<input type="radio">`
 - Definisce un radio button che permette **una** scelta tra diverse opzioni
 - Esempiof2.html

```
<form>
<input type="radio" name="sex" value="male"
checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>
```

Checkbox input

- `<input type="checkbox">`
 - Definisce una serie di checkbox che permettono una scelta **multipla** tra diverse opzioni
 - Esempiof2.html

```
<form>
<input type="checkbox" name="vehicle1" value="Bike"> I
have a bike
<br>
<input type="checkbox" name="vehicle2" value="Car"> I
have a car
</form>
```

Drop-down list

- `<select>`
 - Definisce un menù a tendina
 - Elemento `<option>` definisce le possibili opzioni
 - `<option value="fiat" selected>Fiat</option>`
 - Esempiof3.html

```
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
```

Submit button

- `<input type="submit">`
 - Definisce un bottone per sottomettere la form ad un handler
 - Handler è una pagina lato server che processa i dati della form
 - Handler è definito con l'attributo `action`
 - Esempio4.html

```
<form action="action_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey">  
<br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse">  
<br><br>  
<input type="submit" value="Submit">  
</form>
```

- Permette anche di eseguire semplici azioni
 - `<button type="button" onclick="alert('Hello World!')>Click Me!</button>`

Action attribute

- Definisce le azioni che devono essere fatte quando la form è inviata
- Sottomissione fatta solitamente con il bottone submit
- La pagina viene inviata a una pagina web lato server
 - Ad esempio uno script lato server gestisce la form inviata
 - `<form action="action_page.php">`
 - Se l'attributo `action` è omesso si considera la pagina corrente di default

Method attribute

- Specifica l'operazione HTTP (GET o POST) da usare per inviare la form
 - `<form action="action_page.php" method="get">`
 - `<form action="action_page.php" method="post">`
- GET (default) usata per invio di form passive (ad es., search engine query) e senza informazioni sensibili
- I dati inviati tramite la form sono visibili nell'indirizzo
 - `action_page.php?firstname=Mickey&lastname=Mouse`
- GET è ottima per piccola quantità di dati

- Specifica l'operazione HTTP (GET o POST) da usare per inviare la form
 - `<form action="action_page.php" method="get">`
 - `<form action="action_page.php" method="post">`
- GET (default) usata per invio di form che aggiornano dati
- POST fornisce maggiore sicurezza perchè i dati non sono visibili nell'indirizzo (ad es., password)

Name attribute

- Ogni campo input deve avere un nome

```
<form action="action_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey">  
<br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse">  
<br><br>  
<input type="submit" value="Submit">  
</form>
```

Altri attributi

- Attributo value specifica il valore iniziale
 - `<input type="text" name="firstname" value="John">`
- Attributo readonly specifica che il valore non può essere cambiato
 - `<input type="text" name="firstname" value="John" readonly>`
- Attributo disabled specifica un campo disabilitato e non utilizzabile (non viene inviato)
 - `<input type="text" name="firstname" value="John" disabled>`

XHTML

XHTML = Extensible HyperText Markup Language

XHTML è quasi identico, è più stringente di HTML.

XHTML è supportato dai principali browser

Perché XHTML?

Molte pagine web contengono HTML errato. Il seguente codice viene visualizzato bene nella maggior parte dei casi anche se non segue le regole HTML

Diversi browser che eseguono su diversi device

Interpretare un markup sbagliato richiede maggiori risorse o potenza computazionale (più complicato per i device più piccoli)

XML è un linguaggio di markup che esploreremo più avanti in questo insegnamento dove i documenti devono essere ben formati (markup corretto)

XHTML combina HTML e XML

XHTML è HTML ridisegnato come XML

XHTML definito per essere visualizzabile dai browser per HTML

HTML vs X-HTML

Struttura documento:

- XHTML DOCTYPE è obbligatorio
- L'attributo xmlns in <html> è obbligatorio
- <html>, <head>, <title>, e <body> sono obbligatori

Elementi XHTML

Elementi XHTML devono essere innestati correttamente, devono sempre essere chiusi, devono essere minuscoli ed devono avere un'unico elemento radice

Attributi XHTML

- Nomi di attributi devono essere minuscoli
- Valori di attributi devono essere quoted
- Minimizzazione degli attributi vietata

Come convertire HTML in X-HTML

- Aggiungere un XHTML <!DOCTYPE> nella prima linea di ogni pagina
- Aggiungere un attributo xmlns all'elemento html di ogni pagina
- Cambiare tutti i nomi di elemento in minuscolo
- Chiudere tutti gli elementi vuoti
- Cambiare tutti i nomi di attributo in minuscolo
- Quotare tutti i valori di attributi

Validazione

W3C segue l'evoluzione del Web, fornisce linee guida e regole per la specifica delle pagine

W3C fornisce un meccanismo per la validazione delle pagine web <http://validator.w3.org>