

2

Informazione, informatica e risorse culturali digitali sul web: guida all'uso dei linguaggi ipertestuali

UniMc – BBCC 12/13 – II semestre

Laboratorio di informatica
avanzata per i beni culturali

prof. Pierluigi Feliciati
pierluigi.feliciati@unimc.it

Ripartiamo dai portali...

- <http://www.presadiretta.rai.it/dl/portali/site/puntata/ContentItem-f2be1f06-4a10-4e>
- Ragioniamo sulla percezione comune dell'uso dell'ICT per i beni culturali vedendo insieme la parte della trasmissione RAI *Preso Diretta* riguardante Culturaitalia.
- Che ne pensate?
- I temi non sembrano limitarsi alla soddisfazione degli utenti ma al senso stesso del portale... capolavori o beni culturali?

I'ARCHITETTURA del WEB

Gli aspetti che caratterizzano una pagina Web sono quattro:

1. il contenuto

che rappresenta l'informazione che si vuole comunicare all'utente, che come è facile intuire è l'aspetto che maggiormente contribuisce a determinare il valore che l'utente attribuisce alla pagina

2. la struttura

intesa sia come struttura di navigazione, costituita dall'insieme di link che consentono all'utente di raggiungere le varie sezioni di un sito, eventuali form di ricerca e login, ecc. sia come struttura logica del contenuto, cioè l'organizzazione di questo in paragrafi, intestazioni, elenchi puntati e numerati, ecc.

I'ARCHITETTURA del WEB

Gli aspetti che caratterizzano una pagina Web sono:

3. la presentazione

che rappresenta la modalità con cui la struttura viene presentata all'utente. La presentazione può essere di tipo visuale (grafica), testuale e non visuale, per gli utenti che usano tecnologie assistive ed alcuni browser alternativi

4. il comportamento

che consente di alterare la struttura e la presentazione in risposta ad eventi generati dall'utente (pressione di un tasto del mouse, movimento del puntatore, pressione di un tasto della tastiera, ecc.) o ad eventi e condizioni indipendenti dall'utente

I'ARCHITETTURA del WEB

Gli aspetti che caratterizzano una pagina Web

I primi tre aspetti (**contenuto**, **struttura**, **presentazione**) definiscono una pagina come entità statica, l'ultimo (**comportamento**) ne determina la dinamicità.

Premesso che il contenuto ed un minimo di struttura logica sono sempre presenti, non tutte le pagine presentano tutti e quattro gli aspetti.

Molte pagine, infatti, non presentano nessuna struttura di navigazione, altre non presentano nessun comportamento ed altre ancora, per la verità piuttosto rare, non presentano alcuna presentazione.

l'architettura del web a strati



Lo strato di comportamento poggia sia su quello di presentazione, che su quello di contenuto e struttura in quanto agisce su entrambi quello di presentazione poggia unicamente su quello di contenuto e struttura.

Ogni strato agisce esclusivamente su quello sottostante (quelli sottostanti nel caso del comportamento) e deve totalmente ignorare la presenza di quelli sovrastanti.

le risorse digitali e il WWW

Type:text

E' il primo e più universale dei formati secondo la classificazione standard MIME. Si tratta di dati codificati con combinazioni 1:1 di mappe caratteri – charset – anche per esprimere eventuali formattazioni.

Subtype and extensions:

plain	txt, text	Cfr. RFC 2046
html	htm, html	
sgml	sgml	http://www.w3.org/MarkUp/SGML/
xml	xml, dtd, x	http://www.w3.org/MarkUp/
rtf	rtf	www.soft-land.org/documenti/rtf.html

le risorse digitali e il WWW

I linguaggi di markup

Il termine **markup** (o **marcatura**) deriva dall'ambiente tipografico dove si usava marcare con annotazioni le parti del testo che andavano evidenziate o corrette.

La tecnica di composizione di un testo con l'uso di marcatori (o espressioni codificate) richiede una serie di convenzioni, ovvero di un **linguaggio a marcatori** di documenti. In generale un linguaggio di markup **descrive i meccanismi di rappresentazione** (strutturali, semantici o presentazionali) **del testo** che, utilizzando convenzioni standardizzate, sono utilizzabili su più supporti di lettura.

I *linguaggi di marcatura di tipo descrittivo* lasciano la scelta del tipo di rappresentazione da applicare al testo al software che di volta in volta lo riprodurrà

HTML

I linguaggi di markup: HTML

HTML è un linguaggio usato per scrivere i documenti ipertestuali per il Web, ossia **descrive il contenuto, testuale e non, di una pagina web**. E' un linguaggio standard, di pubblico dominio, la cui sintassi è stabilita dal **World Wide Web Consortium (W3C)**

Durante gli anni l'HTML ha subito molte revisioni e miglioramenti: attualmente l'ultima versione disponibile è la versione 4.01, resa pubblica il 24 dicembre 1999.

Da allora, da parte del W3C non è stata manifestata alcuna intenzione di apportare ulteriori modifiche all'HTML, poiché verrà presto sostituito dai nuovi linguaggi **XHTML** ed **XML** di cui si dirà.

HTML

Ogni documento HTML deve essere contenuto in un **file**, la cui **estensione** deve essere **.htm** o **.html**.

L'atomo principale della sintassi di questo linguaggio è **l'elemento**. Gli elementi sono le **strutture del linguaggio** a cui è delegata la funzione di formattare i dati o indicare al Web browser delle informazioni. Ogni **elemento** è racchiuso all'interno di **tag, uno di apertura ed uno di chiusura**, quest'ultimo, per certi elementi opzionale, è indicato apponendo il carattere **slash (/)** dopo la parentesi angolare aperta. I tag sono **marcatori (markup)** costituiti da una sequenza di caratteri racchiusa da due parentesi angolari, cioè i segni minore e maggiore.

HTML

la struttura dei documenti

Ogni pagina HTML, nella sua interezza, costituisce essa stessa **un elemento** di nome **HTML** e contiene **due sottoelementi** distinti: **HEAD** e **BODY**.

I tag iniziali e finali che delimitano tutti e tre i suddetti elementi sarebbero opzionali, ma è buona norma specificarli comunque.

<HEAD> contiene l'insieme di elementi che consentono di specificare informazioni generali sul documento che non ne costituiscono il contenuto informativo vero e proprio, insomma i suoi **metadati**.

<BODY> delimita gli elementi che costituiscono il **contenuto informativo** vero e proprio del documento.

HTML

HEAD e metadati

Oltre all'elemento **TITLE**, l'elemento **META** viene inserito nella sezione HEAD e consente di specificare informazioni (o proprietà) non specificabili mediante altri elementi. I METADATI, che possono essere estratte ed **utilizzate dai server, dai browser e dai motori di ricerca**. META riconosce i seguenti attributi:

NAME per specificare il valore della proprietà che si sta definendo
CONTENT per attribuire un nome alla proprietà che si sta definendo:

```
<META NAME="Author" CONTENT="Pierluigi Feliciati">
```

```
<META NAME="Description" CONTENT="Pagina che mostra l'uso  
del tag META">
```

```
<META NAME="keywords" CONTENT="HTML, META">
```

HTML

HEAD e metadati

Per non lasciare alla discrezionalità di ogni sito la definizione dei **NAME** dei **META**, è stato elaborato come abbiamo detto lo standard **DUBLIN CORE**. Vediamone un esempio di uso (si noti la dichiarazione dello schema con LINK) per l'home page dell'AIB:

```
<link rel="schema.DC" href="http://dublincore.org" />
<meta name="DC.Title" content="AIB-WEB (Associazione italiana
    biblioteche)" />
<meta name="DC.Creator" content="Associazione italiana
    biblioteche" />
<meta name="DC.Subject" content="Associazione italiana biblioteche,
    AIB, Italian library association" />
<meta name="DC.Description" lang="it" xml:lang="it" content="Home
    page dell'Associazione italiana biblioteche" />
<meta name="DC.Date.Created" content="1997-03-01" />
<meta name="DC.Language" scheme="RFC1766" content="it" />
```

HTML

la struttura dei documenti HTML

Riassumendo, quindi, ogni pagina HTML dovrebbe avere questa struttura, che si raccomanda di definire prima dell'inserimento dei contenuti:

```
<HTML>  
  <HEAD>  
    <TITLE>Titolo</TITLE>  
  </HEAD>  
  <BODY>  
    <P> Paragrafi... </P>  
  </BODY>  
</HTML>
```

HTML

- Provate a sviluppare una pagina HTML che abbia come contenuto visibile (BODY) solo un PARAGRAFO con qualche parola di prova, ma che abbia come intestazione ovvero metadati (HEAD) almeno TITOLO, AUTORI, DATA, PAROLE CHIVE, DESCRIZIONE
- Per sviluppare un file di testo si usa un semplice editor testuale come BLOCCO NOTE, ma per interpretare i tag HTML si deve visualizzare con un Browser.
- Attenzione! Per salvare il file scegliete Salva con nome > altri file > prova.html NON SALVATE IL FILE COME TXT

HTML

BODY: i tag

I tag (elementi) di BODY possono essere **strutturali o formali**

Un esempio base di elemento formale è `<br>` che serve per indicare **un ritorno a capo**, che non ha bisogno di tag di chiusura. Quasi sempre **le informazioni su cui "agisce" il tag** devono essere racchiuse fra un tag di apertura ed uno di chiusura:

prima, `<b>testo testo testo</b>` e dopo

marca la parte di testo che deve essere visualizzato **in grassetto**:

prima, **testo testo testo** e dopo

ci sono molti altri elementi che agiscono a livello di fonte carattere, per ottenere corsivo `<i>`, sottolineato `<u>`, etc.

HTML

Gli attributi

Gli attributi che hanno lo scopo di **specificare le proprietà degli elementi**, se presenti, sono caratterizzati dal seguente formato:

nome_attributo = valore

Non tutti gli attributi necessitano di un valore. I valori, se presenti, possono essere racchiusi tra apici singoli (') o doppi (").

Esempio: l'elemento **SIZE** consente di impostare il livello di **dimensione dei caratteri** esplicitamente **<SIZE="n">** oppure come incremento **<SIZE="+n">** o decremento **<SIZE="-n">** rispetto al livello base

HTML

HTML

In HTML esistono **6 livelli di intestazioni** specificati mediante gli elementi **H?** (con ? che può assumere i valori da 1 a 6).

Ogni intestazione è visualizzata su una riga separata utilizzando una fonte carattere ed uno stile che dipendono dal browser.

La dimensione dei caratteri utilizzati, comunque, diminuisce al crescere del livello dell'intestazione.

Livello 1

Livello 2

Livello 3

Livello 4

Livello 5

Livello 6

HTML

Struttura: le liste

Sono previste **tre tipologie di liste** in HTML:

- **di definizioni**, che consentono di rappresentare elenchi di termini con le relative definizioni;
- **ordinate**, che consentono di rappresentare elenchi numerati;
- **non ordinate**, che consentono di rappresentare elenchi i cui elementi vengono contrassegnati con un simbolo grafico (generalmente un pallino riempito).

HTML

Le liste non ordinate vengono delimitate dall'elemento `<UL>`. I loro elementi, invece, vengono specificati mediante l'elemento `<LI>` (il cui tag finale è opzionale).

`<UL>`

`<LI> Elemento 1`

`<LI> Elemento 2`

`<LI> Elemento 3`

`<LI> Elemento 4`

`<LI> Elemento 5`

`</UL>`

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4
- Elemento 5

HTML

Le liste ordinate vengono delimitate dall'elemento ****. I loro elementi, invece, vengono specificati mediante l'elemento **** (il cui tag finale è opzionale).

** Elemento 1**

** Elemento 2**

** Elemento 3**

** Elemento 4**

** Elemento 5**

1. Elemento 1
2. Elemento 2
3. Elemento 3
4. Elemento 4
5. Elemento 5

HTML

È consentito **annidare liste** anche di tipo diverso.
Quando si annidano liste dello stesso tipo il browser utilizzerà contrassegni diversi per gli elementi dei vari livelli e in ogni caso ogni livello sarà più indentato del precedente.

```
<UL>
```

```
  <LI>Verdura
```

```
    <OL>
```

```
      <LI> Lattuga
```

```
      <LI> Pomodori
```

```
      <LI> Carote
```

```
    </OL>
```

```
</UL>
```

- Verdura
 - 1. Lattuga
 - 2. Pomodori
 - 3. Carote

HTML

i link

Ogni link è caratterizzato da **due estremità**, denominate **anchor**, ed **una direzione**. Un link ha **inizio dall'anchor di origine e punta a quello di destinazione** che può essere una qualsiasi risorsa individuabile mediante una URL.

Nella definizione di un link oltre ad indicarne l'anchor di destinazione è anche possibile specificare **il tipo di relazione tra questo e l'anchor di origine**.

I link – la base dell'ipertesto - vengono comunemente utilizzati per accedere mediante la **selezione di una componente testuale e/o grafica del documento alla risorsa a cui puntano**.

HTML

i link

L'elemento `<A>` viene utilizzato per definire link e anchor di destinazione all'interno del corpo dei documenti (sezione BODY). Per funzionare utilizza l'attributo `HREF` che consente di definire **un link il cui anchor di origine è il contenuto dell'elemento A e quello di destinazione è la URL che costituisce il suo valore.**

L'utente avrà accesso a tale URL selezionando il testo e/o le componenti grafiche che costituiscono il contenuto di A.

Per evidenziare la presenza del link **il testo viene visualizzato sottolineato** mentre **le immagini appaiono bordate**, utilizzando un colore specificabile mediante gli attributi di `<BODY>`.

HTML

i link

È possibile definire un link specificando un valore per **HREF**, ma non è consentito annidare un elemento **A** all'interno di un altro elemento dello stesso tipo.

Vediamo un esempio di definizione di un link in cui la pagina *prova.html* viene associata alla parola “selezionami”:

```
<A HREF="ApproccioWeb.html">Selezionami</A> per  
    accedere ad una semplice pagina di prova che ti  
    mostrerà l'uso dei link.
```

ottiene:

Selezionami per accedere ad una semplice pagina di prova che ti mostrerà l'uso dei link.

HTML

Link assoluti e link relativi

nella sintassi delle URL, è possibile separare idealmente la parte che si riferisce al nome del server (dominio) e al percorso gerarchico per arrivare al nome del file (detta **URL base**) dal nome del file. Ecco un link assoluto:

```
<A HREF="www.w3c.org/Consortium/Offices/role.html">
```

HTML consente di inserire nel documento link relativi, che **omettono la parte di indirizzo già dichiarata** al browser, oppure uguale a quella del documento. Se volessi rinviare al file `role2.html` che si trova nella stessa cartella dello stesso server, sarebbe infatti sufficiente:

```
<A href="role2.html">
```

HTML

dichiarazione URL base

è possibile dichiarare nella **<HEAD>** la **URL base** tramite l'elemento **<BASE>**.

Nel caso mancasse **BASE** la **URL base** sarà quella che individua la **directory contenente il documento** di origine dei link.

BASE riconosce ovviamente l'attributo **HREF** per specificare la URL:

```
<HEAD>
```

```
...
```

```
<BASE HREF="www.w3c.org/Consortium/Offices/">
```

```
</HEAD>
```

HTML

Link per l'invio di e-mail

Utilizzando sempre l'elemento `<A>` ma con schema `mailto` (`mailto:nome_utente@macchina?subject="oggetto"`) possono essere utilizzate per creare un link la cui selezione fornisce all'utente la possibilità di **inviare un messaggio di posta elettronica ad un qualsiasi indirizzo e-mail** (come ad esempio quello dell'autore o del responsabile della pagina).

Per maggiori informazioni contattare: `<A HREF="mailto:pierluigi.feliciati@unimc.it">`

`Pierluigi Feliciati</A>`

ottiene:

Per maggiori informazioni contattare: [Pierluigi Feliciati](mailto:pierluigi.feliciati@unimc.it)

HTML

le immagini

Per inserire un'immagine (ovvero un **file immagine**) in una pagina HTML si utilizza l'elemento **IMG**, con l'attributo **SRC** che consente di specificare la URL dell'immagine da visualizzare e alcuni altri attributi per specifiche di posizione e formattazione dell'immagine. La sorgente:

Questo simbolo `<IMG SRC="smile.gif">` viene spesso utilizzato per sottolineare che una frase scritta e' uno scherzo.

restituisce (se **smile.gif** è un link corretto):

Questo simbolo  viene spesso utilizzato per sottolineare che una frase scritta e' uno scherzo.

HTML

le form

Le form sono maschere attraverso le quali l'utente può inserire dei dati. Questi potranno essere processati mediante una **CGI** (Common Gateway Interface), uno script scritto in uno dei linguaggi che è possibile "immergere" nelle pagine HTML (tipicamente JavaScript) o una combinazione dei due.

All'interno delle form, oltre agli elementi che ne definiscono le componenti (*controlli*), possono essere contenute tutte le entità HTML fatta eccezione per altre form. Non è possibile quindi annidare form, anche se una pagina ne può contenere più d'una.

La definizione di una form avviene mediante l'elemento **FORM**, con gli attributi **ACTION** e **METHOD**

HTML

un esempio di form

```
<FORM ACTION="" METHOD="POST">
  Nome: <INPUT TYPE="text" NAME="nome" SIZE="25"
  MAXLENGTH="30">
  <P>Cognome: <INPUT TYPE="text" NAME="cognome" SIZE="25"
  MAXLENGTH="30">
  <P>Indirizzo: <INPUT TYPE="text" NAME="indirizzo" SIZE="25"
  MAXLENGTH="30">
  <P>Sesso: <INPUT NAME="sesso" TYPE=RADIO VALUE="f">femminile
    <INPUT NAME="sesso" TYPE=RADIO VALUE="m" CHECKED>maschile
  <P>Che genere musicale preferisci?<BR><SELECT
  NAME="genere_musicale">
    <OPTION> rock
    <OPTION> pop
    <OPTION> musica classica </SELECT> <P>
  <INPUT TYPE=SUBMIT VALUE="Invia">
  <INPUT TYPE=RESET VALUE="Cancella">
</FORM>
```

HTML

Ecco il nostro esempio di form

Nome:

Cognome:

Indirizzo:

Sesso: ☐ femminile ☒ maschile

Che genere musicale preferisci?



HTML

HTML e i fogli di stile

Ci siamo limitati qui a vedere e sperimentare le funzioni principali del metalinguaggio HTML, prevedendo che ogni pagina web debba avere, oltre ai contenuti, le indicazioni riguardo alla struttura e al formato di visualizzazione.

In realtà, da tempo si è superato questo metodo, preferendo **separare i contenuti dagli stili di visualizzazione**, attraverso i **fogli di stile a cascata (CSS - Cascading Style Sheets)**, una tecnica che permette di fissare gli stili (per es. tipo di carattere, colori e spaziature) da applicare ai documenti HTML e XHTML. Le regole per comporre i fogli di stile sono contenute in un insieme di direttive emanate a partire dal 1996 dal W3C.

HTML

HTML e i fogli di stile

L'inserimento di codice CSS nelle pagine web può essere effettuato in due modi:

- 1- Inserendo nel tag <head> della pagina un **collegamento ad un foglio di stile esterno**, cioè un file contrassegnato dall'estensione **.css** come nell'esempio seguente:

```
<html>
  <head>
    <title>Esempio</title>
    <link rel="stylesheet" type="text/css"
href="foglio_di_stile.css">
  </head>
  <body> ... </body>
</html>
```

HTML

HTML e i fogli di stile

2 - Inserendo, sempre all'interno dell'<head> tra gli specifici tag **<style>** e **</style>** le dichiarazioni CSS:

```
<html>
  <head>
    <title>Esempio</title>
    <style type="text/css">
      comandi css
    </style>
  </head>
    <body> ... </body>
</html>
```