

Lezione 5 e 6

La codifica del testo per il web

La codifica del testo consente di creare un linguaggio condiviso tra l'uomo e la macchina affinché quest'ultima possa *interpretare* alcune semplici istruzioni sulla **struttura logica e l'aspetto dei contenuti** di un testo.

`img`

HTML (HyperText Markup Language) venne ideato nel 1990 da Tim Berners-Lee, padre fondatore del web che conosciamo. HTML è considerato uno dei 3 standard dell'architettura web: * HTML, linguaggio per rappresentazione dei testi * HTTP, protocollo per lo scambio di informazioni nel web * URL, identificatore univoco di risorse nel web

Il linguaggio si è evoluto nel corso del tempo e attualmente siamo alla versione [HTML5](#).

HTML permette di **codificare/annotare/marcare** elementi della struttura logica di un testo (divisione in blocchi, titoli, immagini) e aspetti tipografici (corsivo, grassetto) mediante (1) l'utilizzo di un **vocabolario controllato** di termini e (2) una sintassi prestabilita per l'ordine dei termini (una sorta di grammatica).

Ad esempio, il seguente blocco di testo in HTML:

```
<p>Sono un paragrafo con un testo in <i>corsivo</i>.</p>
<p>Io sono un altro paragrafo.</p>
```

viene visualizzato da un browser come segue:

Sono un paragrafo con un testo in *corsivo*.

Io sono un altro paragrafo.

L'elemento `<p></p>` rappresenta un paragrafo, elemento di organizzazione logica della pagina, e l'elemento `<i></i>` racchiude un testo in corsivo.

1. HTML

Una **pagina HTML** è un documento di testo - che può essere redatto con qualsiasi editor di testo - che utilizza il vocabolario e la sintassi HTML per marcare i contenuti ed è salvato nel formato `.html`.

Vediamo un esempio di pagina HTML.

```
<!doctype html>
<html>
  <head>
    <title>La mia collezione di CD</title>
  </head>
  <body>
    <section>
      <h1>Romanza, Andrea Bocelli</h1>
      <p>titolo: <i>Romanza</i></p>
      <p>autore: Andrea Bocelli</p>
      <p>paese: EU</p>
      <p>etichetta: Polydor</p>
      <p>prezzo: 10.80 euro</p>
      <p>anno: 1996</p>
    </section> ... ..
  </body>
</html>
```

Questa pagina, visualizzata da browser, appare come segue:

Romanza, Andrea Bocelli

titolo: *Romanza*

autore: Andrea Bocelli

paese: EU

etichetta: Polydor

prezzo: 10.80 euro

anno: 1996

HTML fornisce un vocabolario prestabilito per descrivere gli elementi principali di una pagina web. Guardiamo il vocabolario usato nell'esempio: `<!doctype html>` è la dichiarazione del linguaggio utilizzato per redigere la pagina. Il browser interpreta automaticamente questa istruzione come **HTML5** * `html` elemento distintivo di una pagina web, racchiude **tutte** le informazioni della pagina * `head` intestazione della pagina web. I contenuti di questa sezione *non sono visibili nella pagina web*, ma includono informazioni fondamentali per la macchina (browser), e.g. il titolo della pagina `title`, che è visibile nel tab del browser ma non nella pagina stessa (*) * `body` corpo del testo della pagina, contiene tutto quello che può essere visualizzato in una pagina web * `section` divisione generica (blocco) di testo della pagina * `h1` (h = heading, 1 = level 1) intestazione/titolo di primo livello * `p` (p = paragraph) paragrafo, sotto-divisione del testo * `i` (i = italic), corsivo

(*) vedremo che `head` può contenere altre informazioni, come i `link` a risorse esterne, e.g. i fogli di stile `.css`, o metadati (`meta`), ovvero descrizioni dei contenuti della pagina per l'indicizzazione dei motori di ricerca.

Guardiamo la struttura e la sintassi dei contenuti della pagina: * **struttura gerarchica** dei contenuti: `body > section > p > i` * la gerarchia è evidenziata dall'utilizzo (consistente) di spazi all'inizio della linea, chiamato **indentazione** * la gerarchia è rappresentata mediante l'apertura e la chiusura di tag, e.g. `<p></p>` * la gerarchia rispetta delle regole di *semantica* dei termini, e.g. una sezione può contenere paragrafi (e.g. `section > p`) ma non viceversa (`p > section`) * gli elementi possono essere ripetuti, e.g. più paragrafi `p` in una sezione `section`

1.1 Markup: regole di sintassi HTML

1. Ogni documento HTML contiene sempre nella prima riga la **dichiarazione** del linguaggio usato, `<!doctype html>`

```
html
|
|_ head
|   |
|   |_ title
|
|_ body
|
|_ section
|
|   |_ h1
|   |_ p
|       |_ i
|   |_ p
|   |_ p
|   |_ p
|   |_ p
|   |_ p
```

2. Ogni documento HTML può essere visto come un albero (una struttura gerarchica) di nodi (detti **elementi HTML**). Esistono **nodi parenti** e **nodi discendenti**
3. Ogni documento HTML ha sempre un **nodo radice**, obbligatorio, ovvero l'elemento `html` che racchiude tutti gli altri elementi.

N.B. In HTML esistono alcuni **elementi obbligatori** (`html`, `head`, `title` e `body`). Tutti gli altri dipendono dallo sviluppatore.

4. Ogni elemento è composto da:

1. un tag di apertura, e.g. `<html>`
2. un tag di chiusura, e.g. `</html>`
3. il valore dell'elemento.

5. Il **valore** di un elemento può includere:

1. altri elementi (`<section><p>...</p></section>`)
2. semplice testo (`<p>autore: Andrea Bocelli</p>`)
3. altri Elementi e testo (e.g. `<p>titolo: <i>Romanza</i></p>`)

I rapporti tra nodi parenti e nodi discendenti si chiamano **annidamenti**. Un elemento HTML può contenere un altro elemento HTML **SOLO SE** lo include completamente.

```
<p>titolo: <i>Romanza</i></p> OK!
<p>titolo: <i>Romanza</p></i> NO!
```

6. Esistono **elementi vuoti**, detti anche *milestone*, che non contengono alcun valore. Gli elementi vuoti servono per inserire interruzioni, o multimedia (immagini, video). Gli elementi vuoti possono essere rappresentati con una sintassi sintetica.

Esempio:

```
<p>Sarei un paragrafo, ma se aggiungo "<br/>" obbligo il browser ad andare a capo prima del dovuto.</p>
```

In un browser:

Sarei un paragrafo, ma se aggiungo "
" obbligo il browser ad andare a capo prima del dovuto.

7. Ad ogni elemento HTML possono essere associati uno o più **attributi**. La funzione dell'attributo è specificare ulteriori caratteristiche dell'elemento a cui è associato. Il **valore dell'attributo** può essere unico all'interno della pagina o condiviso con altri elementi. L'attributo compare *solo* nel tag di apertura, non viene ripetuto nel tag di chiusura.

Riprendiamo l'esempio di prima con alcune modifiche.

```
<!doctype html>
<html lang="it">
<head>
  <title>La mia collezione di CD</title>
</head>
<body>
  <section>
    <h1>Romanza, Andrea Bocelli</h1>
    <p id="1" class="cd_info">titolo: <i>Romanza</i></p>
    <p id="2" class="cd_info">autore: Andrea Bocelli</p>
    <p id="3" class="cd_info">paese: EU</p>
    <p id="4" class="cd_info">etichetta: Polydor</p>
    <p id="5" class="cd_info">prezzo: 10.80 euro</p>
    <p id="6" class="cd_info">anno: 1996</p>
    <p id="7" class="link_video">
      <a href="https://www.youtube.com/watch?v=hVHTkpk-1sM">ascolta l'album su youtube</a>
    </p>
  </section> ... ..
</body>
</html>
```

- **@lang** (language) associato all'elemento `<html></html>` consente di specificare (al browser) la lingua della pagina o dell'elemento a cui viene associato. **Perché è utile?** (1) traduzione automatica e (2) accessibilità per reader non vedenti.
- **@id** (identifier) consente di attribuire un identificativo univoco (il valore dell'attributo) all'elemento a cui è associato. Il valore deve essere unico all'interno dello stesso sito.
- **@class** consente di attribuire un valore condiviso con altri elementi sulla base di esigenze di grafica, organizzazione dei contenuti etc.
- **@href** (Hypertext REFERENCE) associato all'elemento `<a>` (anchor) permette di creare un link ad una pagina

N.B. Gli attributi `@id` e `@class` saranno i nostri *punti di ancoraggio* per richiamare in CSS gli elementi HTML ai quali vogliamo associare uno stile grafico. Pertanto la scelta (degli elementi a cui associare un attributo, quale attributo associare e quale valore attribuire)

Altre note

1. HTML **non** è case sensitive.

```
<p>questo elemento</p>
<P>è uguale a questo</P>
```

Buona pratica è utilizzare *lowercase* per i tag e gli attributi.

2. **L'indentazione** del codice è una buona pratica (per la leggibilità del codice) ma non è obbligatoria.

```
<p>Questo è un paragrafo che finisce qui.</p><p>Qui inizia il paragrafo successivo, su una nuova linea, anche se non ho indentato il codice.</p>
```

In un browser vedremo:

Questo è un paragrafo che finisce qui.

Qui inizia il paragrafo successivo, su una nuova linea, anche se non ho indentato il codice.

N.B. Ciò significa che andando a capo nella scrittura del testo, il browser non andrà a capo a sua volta!! Solo specificando la fine dell'elemento - ed eventualmente gli stili per la grafica - si può *obbligare* il browser a mandare a capo.

3. Altra buona pratica è aggiungere commenti nel codice per annotare cosa si sta facendo o cosa si intende fare. Il contenuto dei commenti non viene visualizzato dal browser. I commenti sono inclusi tra `<!-- -->`

```
<p id="7" class="link_video">
<a href="https://www.youtube.com/watch?v=hVHTkpk-1sM">ascolta l'album su youtube</a>
<!-- TO DO: aggiungi link a Spotify -->
</p>
```

4. HTML non include aspetti di *semantica*, e.g. non è possibile specificare se si sta parlando di una persona, di un luogo o di un oggetto che è (potenzialmente) descritto in altri documenti web. Per questi aspetti esistono altre tecnologie, dette *Semantic Web*, per arricchire una pagina web con descrizioni di concetti (reali e astratti) e relazioni tra concetti (e.g. *avere autore*)

1.2 Elementi principali del vocabolario HTML

Struttura della pagina: `html`, `head`, `title`, `body`

`html` include *solo* i due elementi html che compongono la pagina web: `head` e `body`, che sono elementi *fratelli* (di pari livello).

I contenuti di una pagina HTML si dividono in `head`, che include informazioni e istruzioni (e.g. link) non visualizzate, e `body`, che include tutto ciò che può essere visualizzato dal browser.

Per il momento, sappiamo che `head` può contenere l'elemento `title`.

```
<html>
<head>
  <title>La mia collezione di CD</title>
</head>
<body>
  ...
</body>
</html>
```

Tutti i successivi elementi qui descritti vanno in `body`.

Intestazione e menu: `header`, `h1`, `nav`, `ul`, `li`, `a`

L'elemento `header` normalmente è il primo elemento incluso in `body` ed include tutti gli elementi che compongono l'intestazione della pagina web, e.g. titolo, logo, menu.

Il titolo principale della pagina va sempre espresso con l'elemento `h1`, (heading di livello 1).

L'elemento `nav` include il menu di navigazione primaria (o principale) del sito, che collega la pagina in cui ci si trova alle altre pagine che compongono il sito.

Normalmente l'elenco delle pagine che compongono il menu (o qualsiasi elenco all'interno di una pagina HTML) è incluso in un elemento `ul` (unordered list), che a sua volta include tanti elementi `li` (list item) quante sono le pagine del menu.

Ogni elemento della lista (ovvero ogni pagina a cui si vuole linkare dal menu) include un elemento `a` (anchor), il cui attributo `@href` include la *URL parziale* della pagina html a cui si vuole rimandare.

Per esempio:

```
<header>
  <h1>La mia musica</h1>
  <nav>
    <ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="about.html">ABOUT</a></li>
      <li><a href="cd_list.html">CD LIST</a></li>
    </ul>
  </nav>
</header>
```

Note: * **l'utilizzo e l'ordine** degli elementi `h1`, `nav`, `ul` non sono prescrittivi, ma sono una buona pratica utilizzata universalmente per lo sviluppo web. Infatti i **motori di ricerca** seguono questo schema per trovare e indicizzare le informazioni del sito web (e.g. cercano il titolo in `h1` per dedurre il tema della pagina – quindi le parole utilizzate sono importanti!). * le **URL** possono essere *assolute*, rimandando ad un sito esterno (e.g. `http://www.wikidata.org`), o *parziali* (e.g. `about.html`, `imgs/logo.jpg`), rimandando ad un'altra pagina dello stesso sito * la **home page** di ogni sito web *DEVE* sempre essere chiamata `index.html`. Questo perché il browser legge la lista di pagine html nella cartella del sito e interpreta sempre `index.html` come la pagina di partenza. * `href="about.html"` implica l'esistenza di una pagina html chiamata `about.html` contenuta all'interno della stessa cartella della home page `index.html`. Se la pagina `about.html` non esiste, il browser ritorna un errore (404 not found) * nel menu vanno inseriti **link a pagine interne** al sito. Non vanno mai inseriti link che rimandano a risorse esterne al sito (e.g. link a social network) e che quindi *buttano fuori* l'utente dal sito.

Contenuto della pagina: `h2`-`h6`, `main`, `aside`, `section`, `p`

I titoli successivi al titolo della pagina vanno espressi con headings di livello inferiore `h2`, `h3`, `h4`, `h5`, `h6`. I numeri crescenti non rappresentano *l'ordine dei titoli* all'interno della pagina, bensì il livello di sottotitolo. Normalmente questi elementi sono *figli* di altri elementi contenitori (vedi i successivi `main`, `aside`, `section`) e mai figli diretti di `body`.

Il contenuto principale della pagina (generalmente centrale o allineato a sinistra) può essere inserito nell'elemento `main`, che viene disposto allo stesso livello,

successivamente all'elemento `header` .

Contenuti secondari (e.g. menu di navigazione secondaria o contestuale, link a post su social network, note a margine) possono essere inseriti nell'elemento `aside` .

Il significato di `main` e `aside` è puramente *semantico*, poichè il comportamento all'interno della pagina (ovvero come vengono visualizzati) non cambia.

`main` e `aside` possono contenere blocchi di testo, chiamati `section` , che separano logicamente testi di natura diversa. `section` può *anche* essere utilizzato in alternativa a `main` e `aside` . `section` può contenere titoli `h1-h6` , paragrafi `p` e altri elementi.

Per esempio:

```
<main>
  <section>
    <h2>musica classica: opera</h2>
    <p>Di seguito una lista di CD di musica classica</p>
    <section>
      <h3>Bocelli</h3>
      <p>titolo: Romanza</p>
      ...
      <aside>compra il cd su <a href="http://amazon.com">amazon</a></aside>
    </section>
    <section>
      <h3>Verdi</h3>
      <p>titolo: Aida</p>
      ...
      <aside>compra il cd su <a href="http://amazon.com">amazon</a></aside>
    </section>

    ...
  </section>
</main>
```

Piede di pagina: `footer`

L'elemento `footer` racchiude le informazioni a piede di pagina. Si trova come ultimo elemento figlio di `body` . Può contenere testo strutturato in sezioni o paragrafi, o immagini (e.g. loghi di sponsor). Contiene informazioni e link sui contatti del sito (e.g. tecnico, commerciale, social network), i crediti del sito (e.g. grafica, contenuti), e l'attribuzione del sito (il proprietario del sito). Assieme a `header` , questi elementi sono generalmente fissi, ovvero si ritrovano identici in ogni pagina che compone il sito.

Per esempio:

```
<footer>
  <p>Sito realizzato da <a href="http://example.com">me</a></p>
</footer>
```

Multimedia: `img`

Immagini e altri multimedia (video, audio) possono essere contenuti a qualsiasi livello della pagina, sempre all'interno di elementi *figli* di `body` .

L'elemento `img` consente di inserire un file immagine all'interno della pagina. L'attributo `@src` , similmente a `@href` include la URL parziale al file che si vuole inserire in quel punto. L'elemento `img` è un elemento vuoto, che può essere rappresentato con la sintassi abbreviata.

Per esempio:

```

```

Note:

"imgs/logo.jpg" implica che l'immagine `logo.jpg` è contenuta in una cartella chiamata `imgs` , la quale si trova nella stessa cartella (quindi allo stesso livello) della pagina HTML in cui viene richiamata l'immagine (`index.html`).

Pertanto la **struttura ad albero** del sito web finora esemplificato corrisponde alla seguente organizzazione di file e cartelle:

```
sito_lezione5
|
|_ index.html
|_ about.html
|_ cd_list.html
|_ imgs
   |_ logo.jpg
```

Stile di carattere: `i` , `b` , `u`

Alcuni effetti tipografici possono essere rappresentati tramite appositi elementi html, e.g. corsivo `i` (o `em`), grassetto `b` (o `strong`), sottolineato `u`.

Note:

Altri effetti tipografici più sofisticati possono essere realizzati tramite l'utilizzo di CSS. Per esempio: `i` è una forzatura del font utilizzato rendendolo *obliquo*. In tipografia (web e tradizionale), il corsivo non corrisponde all'obliquo, ma ha un proprio disegno, con pesi e bilanciamento degli spazi e delle grazie studiato ad hoc. Tramite CSS è possibile specificare quando utilizzare il corsivo *vero* o l'obliquo.

2. Installazione Rich Text Editor

Esistono diverse soluzioni software per scrivere codice HTML, CSS, Javascript etc. In questo corso consigliamo l'utilizzo di un software free (gratuito) e *portable* (non richiede diritti di amministratore sul proprio computer per installarlo) chiamato [Sublime Text](#).

NB. Solo la versione 2 è free! Sublime 3 è a pagamento.

Installazione Sublime Text (portable)

Windows * cliccare [qui](#) per iniziare il download * cliccare sul file scaricato per aprirlo / decomprimerlo (file .zip) * spostare il file dell'applicazione sul Desktop per averlo sempre a portata di mano. * aprire l'applicazione

Mac * cliccare [qui](#) per iniziare il download * cliccare sul file scaricato per aprirlo / decomprimerlo (file .dmg) * spostare il file dell'applicazione nella cartella Applicazioni e creare il link nel dock per averlo sempre a portata di mano. * aprire l'applicazione

3. Creare e collegare documenti .html

- nel file system, scegliere/creare una cartella in cui salvare i propri lavori, e.g. `sito_lezione5`
- nell'editor, creare un documento HTML in Sublime text `File > Save as`
- nell'editor, salvare il file nella cartella scelta nominandolo `index.html`
- nel file `index.html`, inserire alcuni tra i principali elementi HTML
- nella cartella, cliccare col tasto destro sul file html > `Apri con` e selezionare il browser preferito (NO Internet Explorer)
- nella cartella, creare una sotto cartella per le immagini `imgs`
- nella sottocartella `imgs` inserire un'immagine a scelta
- nel file `index.html`, inserire il link all'immagine nel documento html in un punto a scelta
- Creare il menu di navigazione del sito ~~~~ HOME ABOUT GALLERY ~~~~
- Creare i link tra le pagine nel menu di navigazione
- Creare le pagine html menzionate nel menu

Link utili

Il miglior modo di imparare HTML è fare pratica e vedere tanti esempi.

Ecco una lista di tutorial (*mainly in english!*) per prendere confidenza con HTML: * [Mozilla - Introduction to HTML](#) spiegazione delle basi di HTML, esempi e *active learning* * [W3School HTML](#) sito con tutorial, esempi ed editor per interagire/sperimentare ad ogni lezione. * [learn HTML](#) spiegazioni con esempi e test interattivi

Impareremo *buona parte* degli elementi del vocabolario HTML strada facendo. Per una lista completa degli elementi HTML vedi la [Reference List di Mozilla](#), che include tutti gli elementi HTML raggruppati per organizzazione logica della pagina (head / body) e tipologia di contenuti (testo, multimedia)