



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

HTML (I parte)

Fabio Vitali

CdS in Informatica

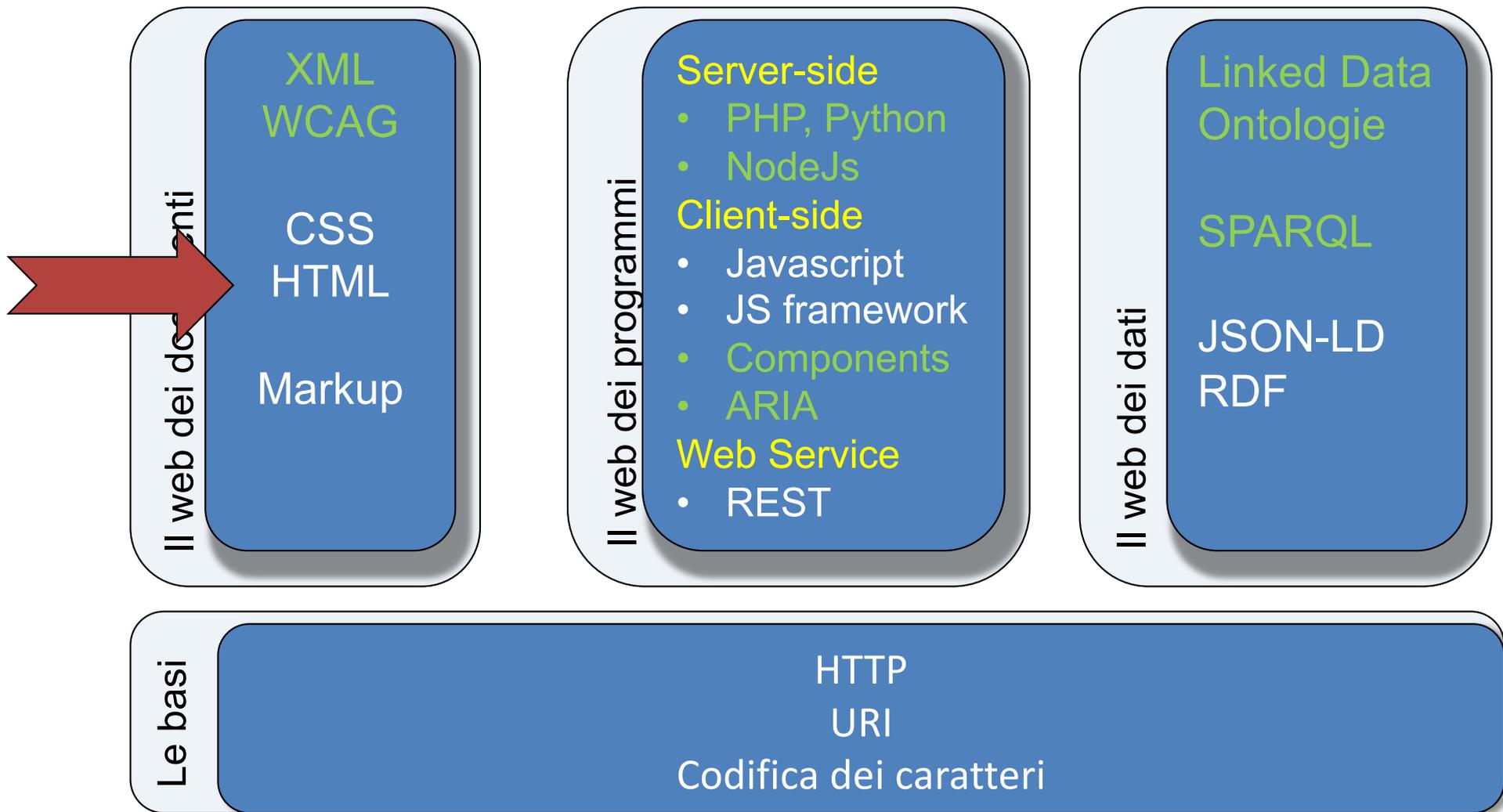
CdS in Informatica per il Management

Introduzione

Oggi esaminiamo in breve:

- La storia e le caratteristiche di HTML
- HTML (concetti base)

Argomenti delle lezioni



Contenuti per tutti
Contenuti solo per 9CFU



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Breve storia di HTML

HTML oggi: una tragedia?

HTML è diventato qualcosa di diverso rispetto a ciò che è stato per molti anni: da linguaggio di markup per documenti ipertestuali e multimediali ad un vero e proprio linguaggio per applicazioni

E' il World Wide Web che sta cambiando e l'attenzione è sempre più su interattività, dinamicità e manipolazione sofisticata (client-side) dei contenuti

Il browser non è più soltanto uno strumento di lettura (senza dimenticare che inizialmente era anche un editor!) ma diventa un motore generico per l'esecuzione di applicazioni

E HTML cambia sia nella semantica che nella sintassi, rendendo vani molti degli sforzi fatti dal W3C negli anni.

Anche perché c'è ancora confusione tra diverse versioni dello standard, proposte parallele e gli stessi gruppi di lavoro...



Due HTML paralleli ?

A oggi esistono due linee di sviluppo parallele di HTML:

- HTML 5.2 è una Recommendation del W3C dal 14 Dicembre 2017 sospesa il 28 gennaio 2021.
<https://www.w3.org/TR/html52/>
- “HTML Living Standard” del WHATWG, con ultimo aggiornamento del 16 Marzo 2021
<https://html.spec.whatwg.org/>

Cosa succede?



HTML 4

Il linguaggio HTML un tipo di documenti SGML (esiste un DTD di HTML, anzi più di uno) progettato per marcare documenti ipertestuali. Tramite HTML è possibile realizzare documenti con una semplice struttura, con aspetti grafici anche sofisticati, con testo, immagini, oggetti interattivi e link.

La cosiddetta “guerra dei browser” (1994-98) ha portato alla creazione di numerosi elementi proprietari, estensioni, effetti sofisticati e caratteristiche puramente presentazionali

Dopo diverse versioni intermedie, HTML 4.0 (1997) “chiude i giochi” e aggiunge il supporto l’internazionalizzazione, per gli style sheet, per i frame, tabelle molto più ricche, etc. HTML 4.01 (1999) apporta modifiche minori

Da allora lo standard è rimasto sostanzialmente invariato fino al 2008: HTML 4 è il linguaggio in cui è stata scritta la stragrande maggioranza delle pagine Web di contenuti.



Da HTML a “tag soup”

Molte pagine HTML tuttavia sono diventate “tag soup”, ossia un insieme di elementi non conformi allo standard. I browser infatti si sono preoccupati poco della correttezza sintattica o strutturale dei documenti HTML.

Questo significa che esistono differenze anche sensibili tra un documento HTML corretto e un documento HTML visualizzabile da un browser Web

Ma come fa il browser ad interpretare i documenti sintatticamente non corretti? Ognuno fa a modo suo!

Non è mai stato infatti standardizzato (prima di HTML 5) il modo in cui fare il parsing di questi documenti.

Inoltre per gestire sia le pagine conformi allo standard che quelle non compatibili (proliferate dopo la “guerra dei browser”), sono stati introdotti due modelli di rendering: quirks mode e strict mode



Quirks mode e strict mode

Molte pagine HTML sono state scritte usando un modello scorretto e locale ad uno specifico browser. Queste stesse pagine non funzionano con i browser che implementano correttamente lo standard.

E' necessario capire se il browser dovesse interpretare una pagina secondo lo standard oppure no.

La soluzione è avere due stili di interpretazione di HTML e CSS:

- ***quirks mode***, compatibile con il passato e più permissivo
- ***strict mode***, compatibile con le specifiche ufficiali

Se la pagina non specifica niente, il browser adotta il modo compatibile (*quirks mode*), altrimenti se l'autore lo richiede esplicitamente attiva il modo restrittivo e corretto.

HTML 5 cambia le regole standardizzando le regole di parsing (anche e soprattutto in caso di errori)



XHTML 1.0

XHTML 1.0 è una Recommendation del 2000. È una riformulazione di HTML 4 come un'applicazione di XML 1.0.

L'elenco e la semantica di elementi e attributi non è assolutamente cambiata rispetto a HTML 4.

I documenti debbono essere ben formati, in particolare l'annidamento deve essere corretto.

Altre differenze con HTML 4 (alcune):

- I nomi di elementi e attributi sono minuscoli (XML è case-sensitive)
- Il tag finale è obbligatorio per elementi non vuoti
- Gli elementi vuoti debbono seguire la sintassi XML
- I valori degli attributi debbono sempre avere le virgolette



(X)HTML 5

Nel 2004, Firefox e Opera proposero al W3C la riapertura del Working Group su HTML per lo sviluppo di nuove versioni del linguaggio. La proposta, che ignorava volutamente XHTML e la rigida sintassi di XML, venne bocciata dal W3C.

Venne così creata una comunità aperta chiamata WHATWG (Web Hypertext Application Technology Working Group) finanziata e supportata da Mozilla, Opera e Apple, che ha iniziato a lavorare ad una versione “intermedia” di HTML, “Web Application 1.0” (WA1), meno ambiziosa di XHTML 2.0. Il chair, Ian Hickson, è ora un project manager a Google.

Nel 2007 il W3C dovette ammettere che queste modifiche avevano un impatto innegabile (“he who ships working code wins”) riaprì il working group con tutti i membri del WHAT per creare una nuova versione del linguaggio, (X)HTML 5.



(~~X~~)HTML 5

WA1 accetta l'amara realtà dei browser odierni che dicono che non è mai esistita una grammatica SGML per HTML, ma che il linguaggio è sempre stato

- O un'applicazione di XML (XHTML *)
- Oppure una tag soup in cui i browser accettano ogni sorta di “porcheria” e fanno il meglio che possono

WA1 è più interessato a mettere a posto le cose che certamente non vanno piuttosto che lavorare su un linguaggio nuovo.

E il W3C deve adeguarsi, anche perché WA1 ha supporto da tutti i browser e di tutta l'industria delle applicazioni Web

“(X)HTML 5” diventa quindi solo “HTML 5” e successivamente solo “HTML”



HTML Living Standard

Dal 2011 il “nuovo” HTML è, per scelta del WG, una specifica perennemente in sviluppo (“living standard”):

Questa instabilità diventa una feature:

Because the specification is now a living document, we are today announcing two changes:

- *The HTML specification will henceforth just be known as "HTML".*
- *The WHATWG HTML spec can now be considered a "living standard".*

It's more mature than any version of the HTML specification to date, so it made no sense for us to keep referring to it as merely a draft.

Ian Hickson, chair del WHATWG, 19 gennaio 2011

Cambia completamente il modello di sviluppo del linguaggio che si allontana dall’approccio sistematico e democratico di evoluzione degli altri standard W3C. E’ la vittoria indiscussa dei produttori di browser: sia per le caratteristiche del linguaggio sia per il modo in cui è sviluppato.



La fine della divisione su HTML

Nel linguaggio esistevano due “anime” incompatibili, con obiettivi diversi:

- La componente W3C voleva “stabilizzare” le versioni di HTML5, per farle diventare Recommendation, con test di conformità e modello approvato con il processo W3C.
- La componente WHATWG voleva continuare lo sviluppo continuo di HTML e di altre tecnologie collegate, in maniera fortemente collegata ad implementazioni esistenti.

Nel 2011 i due gruppi si separano. La licenza di “HTML Living Standard” prevede il riuso di parte delle specifiche. Il W3C quindi pubblica un sottoinsieme delle specifiche di “HTML Living Standard” come “HTML5.x”

- Alcune feature sono omesse, altre sono pubblicate in specifiche separate (es. Canvas 2D)

Negli anni il W3C approva un HTML 5.1, un HTML 5.2 e un draft iniziale di HTML 5.3, progressivamente divergenti da HTML Living Standard. Non c'è però supporto dai browser.

Per questo motivo, nel maggio 2019 il W3C annuncia ufficialmente che le nuove release del linguaggio saranno in carico al WHATWG e che cesserà la produzione di un linguaggio diverso da quello.



Novità di Living Standard e HTML 5

- Il linguaggio di markup è stato rivisto globalmente, introducendo nuovi elementi più specifici per usi e abitudini ormai consolidate (e.g., `<nav>`, `<section>` e `<article>` invece di `<div>`, `<audio>` e `<video>` invece di `<object>`, ecc.)
- L'uso di script per la modifica dinamica *in memory* del documento (DOM) è diventata parte integrante del linguaggio, e ne costituisce un elemento prevalente sulla sua linearizzazione in stringa HTML.
- La sintassi basata su SGML è definitivamente abbandonata, e quella basata su XML è solo facoltativa. HTML è sufficientemente importante da non dover dipendere da una meta-sintassi terza per le sue caratteristiche.
- Oltre al markup, HTML definisce e descrive un certo numero di API per la caratterizzazione del DOM più sofisticata di quanto possibile col markup.
- XHTML5 è una linearizzazione facoltativa di HTML basata sulla sintassi XML. Ha poche ma importanti differenze da HTML.





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

HTML – le basi

HTML – le basi

Adesso parliamo di:

- La struttura di un documento HTML
- Elementi inline
- Elementi di blocco e di lista
- Elementi generici
- Elementi di struttura
- *Link ed immagini*
- *Tipi di dati*
- *Entità predefinite*
- *Peculiarità sintattiche*
- *Tabelle*
- *Form*
- *Il contenuto dell'elemento HEAD*
- *Il DOM*



La struttura di un documento HTML

`<!DOCTYPE html>`: segnala il tipo di markup usato nel documento e la sua versione, con molte complessità e variazioni.

Case-insensitive, ma se si usa XHTML e XHTML 5, allora DOCTYPE è maiuscolo e tutti i tag sono minuscoli.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Document title</title>
```

```
</head>
```

```
<body>
```

```
<p>Text of a paragraph</p>
```

```
</body>
```

```
</html>
```

`html`: La radice dell'albero.

`head`: Informazioni globali sul documento

`body`: Il contenuto del documento



Un primo esempio di HTML

```
<?DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<html>
```

```
  <head>
```

```
    <title> Document title
```

```
  </head>
```

```
  <body>
```

```
    <h1> Major Header </h1>
```

```
    <p>This is a complete  
write until I fill in seve  
it wraps automatically. Su  
document.</p>
```

```
    <p>Did you expect <b>p
```

```
    <p>Here you can see a  
      split by a &lt;br&g
```

```
  <hr>
```

```
  <p> A list of importan
```

```
  <ul>
```

```
    <li>Spaces, tabs and
```

```
    <li>Document type de
```

```
    <li>Document structu
```

```
    <li>Nesting and clos
```

```
  </ul>
```

```
</body>
```

```
</html>
```

Major Header

This is a complete paragraph of a document. I write and write until I fill in several lines, since I want to see how it wraps automatically. Surely not a very exciting document.

Did you expect **poetry**?

Here you can see a paragraph split by a **
**

A list of important things to remember:

- Spaces, tabs and returns
- Document type declaration
- Document structure
- Nesting and closing tags

Elementi inline (1)

Gli elementi *inline* (o di carattere) non spezzano il blocco (non vanno a capo) e si includono liberamente l'uno dentro all'altro. Non esistono vincoli di contenimento.

Si dividono in elementi *fontstyle* e elementi *phrase*.

I tag *fontstyle* forniscono informazioni specifiche di rendering. Molti sono deprecati, e si suggerisce comunque sempre di usare gli stili CSS.

- TT (TeleType, font monospaziato, ad es. Courier),
- I (corsivo),
- B grassetto,
- U (sottolineato - deprecato),
- S e STRIKE (testo barrato - deprecato),
- BIG, SMALL (testo più grande e più piccolo)



Elementi inline (2)

```
<body class="container-fluid">
  <h1>Inline elements</h1>
  <p>Fontstyle elements define specific re
  <ul>
    <li>text in <i>italics</i>.</li>
    <li>text in <b>bold</b>.</li>
    <li>text in <tt>teletype</tt> (i.e.,
    <li>text in <u>underlined</u>.</li>
    <li>text in <s>strike-through</s> an
    <li>text in <sup>superscript</sup> a
    <li>text in <big>bigger</big> and <s
  </ul>
  <p>The phrase elements define specific m
  <ul>
    <li><em>emphasis</em></li>
    <li><strong>major emphasis</strong><
    <li><dfn>definition</dfn></li>
    <li><code>program fragment</code></l
    <li><kbd>text entered by the user</k
    <li><var>program variable</var></li>
    <li><abbr>abbreviation</abbr> and <a
    <li><span>generic inline element</sp
    <li><cite>short quote</cite></li>
    <li><q>long quote</q>.</li>
  </ul>
</body>
```

F Fontstyle elements define specific rendering.

- text in *italics*.
- text in **bold**.
- text in teletype (i.e., monospaced font).
- text in underlined.
- text in ~~strike-through~~ and also ~~strike-through~~.
- text in ^{superscript} and _{subscript}.
- text in **bigger** and smaller.

T The phrase elements define specific meanings.

- *emphasis*
- **major emphasis**
- *definition*
- **program fragment**
- **text entered by the user**
- *program variable*
- abbreviation and acronym
- generic inline element
- short quote
- "long quote".

Elementi di blocco

I tag di blocco definiscono l'esistenza di blocchi di testo che contengono elementi inline.

Elementi fondamentali:

- `<p>`: paragrafo,
- `<div>`: Blocco generico,
- `<pre>`: blocco preformattato,
- `<address>`: l'autore della pagina,
- `<blockquote>`: Citazione lunga

Blocchi con ruolo strutturale

- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`: header



Elementi blocco (2)

```
<body class="container-fluid">
  <h1>Block elements</h1>
  <p>The block elements define text
  <h2>Basic elements: </h2>
    <p>paragraph. White space
    is          <b>not</b> meanin
    <div>generic block</div>
    <pre>preformatted block. Whit
    is          meaningful</pre>
    <address>the author of the p
    <blockquote>This is a long q
  <h2>Blocks with a structural role
    <h1>First level heading</h1>
    <h2>Second level heading</h2>
    <h3>Third level heading</h3>
    <h4>Fourth level heading</h4>
    <h5>Fifth level heading</h5>
    <h6>Sixth level heading</h6>
</body>
```

Block elements

The block elements define text blocks that contain inline elements.

Basic elements:

paragraph. White space is **not** meaningful

generic block

```
preformatted block. White space
                           is          meaningful
```

the author of the page

This is a long quote

Blocks with a structural role

First level heading

Second level heading

Third level heading

Fourth level heading

Fifth level heading

Sixth level heading

Elementi di lista

Le liste sono contenitori di elementi omogenei

Liste più importanti sono:

- ``: Unordered list: lista non ordinata di elementi ``
 - Attributi: `type` (disc, square, circle)
- ``: ordered list: lista ordinata di elementi ``
 - Attributi: `start` (valore iniziale) e `type` (1, a, A, i, I).
- `<dl>`: lista di definizioni, composta di molti elementi `<dt>` (definition term) e `<dd>` (definition data)



Elementi di lista (2)

```
<body class="container-fluid">
  <h1>List elements</h1>
  <ul>
    <li>First element</li>
    <li>Second element</li>
    <li>Third element</li>
  </ul>
  <ol>
    <li>First element</li>
    <li>Second element</li>
    <li>Third element</li>
  </ol>
  <dl>
    <dt>First term</dt><dd>First definition
  </dd>
    <dt>Second term</dt><dd>Second definition
  </dd>
    <dt>Third term</dt><dd>Third definition
  </dd>
  </dl>
</body>
```

List elements

S

- First element
- Second element
- Third element

1. First element
2. Second element
3. Third element

First term

First definition

Second term

Second definition

Third term

Third definition

Elementi di lista (3)

```
<body class="container-fluid">
  <h1>List elements</h1>
  <ul>
    <li>First element</li>
    <li type="square">Second element</li>
    <li type="circle">Third element</li>
  </ul>
  <ol>
    <li type="1">First element</li>
    <li type="i">Second element</li>
    <li type="A">Third element</li>
  </ol>
</body>
```

List elements

- First element
 - Second element
 - Third element
-
1. First element
 - ii. Second element
 - C. Third element

Elementi generici

`<div>` e `` sono cosiddetti *elementi generici*: privi di caratteristiche semantiche o presentazionali predefinite, assumono quelle desiderate con l'aiuto dei loro attributi (`style`, `class`, `lang`).

`<div>` mantiene la natura di elemento blocco, e `` la natura di elemento inline, ma ogni altra caratteristica è neutra.

Es.: identificare con elementi appositi tutti i titoli di film può realizzarsi con `<titolofilm>`, ma si esce da HTML, oppure con ``.



Piccoli effetti grafici

`<hr>` (horizontal rule): una piccola riga orizzontale attraverso lo schermo. Si può controllarne la larghezza e lo spessore.

`
` (break): una andata a capo forzata (non spezza logicamente il paragrafo, impone semplicemente il cambio riga)



Elementi di struttura

HTML 4 fornisce pochi elementi per strutturare i contenuti in sezioni e sottosezioni: il contenitore gerarchico `<div>` (cui associare un ruolo tramite l'attributo `@class`) e gli elementi `<h1>`, ..., `<h6>` (che però non danno “profondità” al documento e devono essere usati in uno specifico ordine)

HTML 5 introduce nuovi elementi con una semantica precisa, per organizzare il documento in contenitori che possono essere più facilmente manipolati, esportati, modificati, etc.:

`<main>`: la parte principale della pagina, ovvero un contenitore di testo o di sottosezioni che rappresenta la parte più importante della pagina. Al suo interno possiamo trovare:

- `<section>`: un contenitore generico annidabile.
- `<article>`: una parte del documento *self-contained* e pensata per essere ri-usata e distribuita come unità atomica (post di un blog, news, feed, etc.)

`<aside>`: una sezione collegata al testo ma separata dal flusso principale (note a margine, sidebar, incisi, pubblicità, etc.)

`<header>` e `<footer>`: elementi iniziali e finali di un documento

`<nav>`: liste di navigazione



HTML 4 corretto ma...

```
<h1>Benvenuto nella mia home!</h1>
```

```
<p>Si divide in due parti: short-bio e post  
giornalieri...</p>
```

```
<h2>Bio</h2>
```

```
<p>Nato a, laureato il, dottorato il, etc.</p>
```

Benvenuto nella mia home!

```
<h2>9 marzo 2017</h2>
```

```
<p>Che giornata grigia.</p>
```

Si divide in due parti: short-bio e post giornalieri...

```
<h2>13 marzo 2017</h2>
```

Bio

```
<p>Una bella giornata di sole</p>
```

Nato a, laureato il, dottorato il, etc.

9 marzo 2017

Che giornata grigia.

13 marzo 2017

Una bella giornata di sole.



...molto meglio (HTML 4)

```
<div id="main">
  <h1>Benvenuto nella mia home!</h1>
  <p>Si divide in due parti: short-bio e post
    giornalieri...</p>

  <div id="bio">
    <h1>Bio</h1>
    <p>Nato a, laureato il, dottorato il, etc.</p>
  </div>

  <div id="posts">
    <div id="09-03-17">
      <h1>09 marzo 2017</h1>
      <p>Che giornata grigia.</p>
    </div>

    <div id="13-03-17">
      <h1>13 marzo 2017</h1>
      <p>Una bella giornata di sole.</p>
    </div>
  </div>

</div>
```



E in HTML 5...

```
<main>
  <h1>Benvenuto nella mia home!</h1>
  <p>Si divide in due parti: short-bio e post
    giornalieri...</p>

  <section id="bio">
    <h1>Bio</h1>
    <p>Nato a, laureato il, dottorato il, etc.</p>
  </section>

  <section id="posts">
    <article id="09-03-17">
      <h1>9 marzo 2017</h1>
      <p>Che giornata grigia.</p>
    </article>

    <article id="13-03-17">
      <h1>13 marzo 2017</h1>
      <p>Una bella giornata di sole.</p>
    </article>
  </section>
</main>
```



Header e footer

HTML 5 definisce inoltre due specifici elementi usati sia per contenuti ripetuti (nel caso di paginazione) e sia per contenuti che solitamente sono visualizzati all'inizio o alla fine di una sezione, sottosezione, articolo, etc.

<header>: contiene l'intestazione della sezione corrente o dell'intera pagina ed è solitamente usato per tabelle di contenuti, indici, form di ricerca, intestazioni.

- Può essere anche usato come contenitore degli headings H1, .., H6.

<footer>: contiene la “parte conclusiva” della sezione corrente o dell'intera pagina. E' usato principalmente per mostrare informazioni (testuali, non metadati) sugli autori della pagina, copyright, produzione, licenze

- Non deve essere visualizzato necessariamente a fondo pagina
- Può essere usato anche per contenere intere sezioni come appendici, allegati tecnici, colophon, etc.



Liste di navigazione

HTML 5 riprende da XHTML 2.0 anche le “liste di navigazione” ossia particolari sezioni dedicate a raggruppare link alla pagina corrente (o a sezioni di) o ad altre pagine

Si usa l’elemento `<nav>` molto spesso in combinazione con `<header>` e `<footer>`.

Le sezioni `<nav>` sono molto utili per l’accessibilità, in quanto possono essere più facilmente identificate e accedute anche da utenti disabili (tramite screenreaders o altri ausili)

Possono essere usate anche per attivare/disattivare le funzionalità di navigazione in base allo user-agent (browser) che sta accedendo alla pagina.



Esempio

```
<header>
  <nav>
    <h1>Navigation</h1>
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/standards/">Standards</a></li>
      <li><a href="/participate/">Participate</a></li>
      <li><a href="/membership">Membership</a></li>
      <li><a href="/Consortium/">About W3C</a></li>
    </ul>
  </nav>
</header>
<main>... </main>
<footer>
  <nav>
    <h1>Contact W3C</h1>
    <ul>
      <li><a href="/Consortium/contact">Contact</a></li>
      <li><a href="/Help/">Help and FAQ</a></li>
      <li><a href="/Consortium/sup">Donate</a></li>
      <li><a href="/Consortium/siteindex">Site Map</a></li>
    </ul>
  </nav>
</footer>
```



Conclusioni

In questa lezione abbiamo cercato di fare chiarezza tra:

- HTML
- (X)HTML 5
- HTML 4.x
- XHTML 1.0



Riferimenti

The HTML Living Standard:

<http://www.whatwg.org/html>

W3C HTML5:

<http://www.w3.org/TR/html5/>

D. Raggett, A. Le Hors, I. Jacobs, *HTML 4.01 Specification*, W3C Recommendation 24 December 1999,

<http://www.w3.org/TR/html401>

M. Altheim et alii, *Modularization of XHTML*, W3C Recommendation 10 April 2001, W3C Recommendation 10 April 2001,

<http://www.w3.org/TR/xhtml-modularization>

M. Altheim, S. McCarron, *XHTML™ 1.1 - Module-based XHTML*, W3C Recommendation 31 May 2001,

<http://www.w3.org/TR/xhtml11>

J. Axelsson et alii, *XHTML™ 2.0*, W3C Working Draft 31 January 2003,
<http://www.w3.org/TR/2003/WD-xhtml2-20030131>





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fabio Vitali

Corso di tecnologie web

Fabio.vitali@unibo.it

www.unibo.it