

# HTML5

Prof. Raffaele Montella, PhD

[raffaele.montella@uniparthenope.it](mailto:raffaele.montella@uniparthenope.it)

# Sommario

- Storia
- Evoluzione
- Dettagli
- Applicazioni

# Hyper Text Markup Language 5

- E' una specifica XML i cui elementi permettono di rappresentare i componenti di una pagina web
- HTML5 deriva da HTML4 & XHTML
  - Ne eredita gli elementi
  - Ha una definizione più formale

# Pagine Web

- Sono documenti di testo
  - Organizzazione in moduli
  - Paragrafi di testo
  - Tabelle
  - Elenchi
- NOTA: nessun riferimento è fatto alla formattazione del documento, ovvero come esso appare all'utente

# Storia

- Nei primissimi anni 90 Tim Barners-Lee inventa il protocollo HTTP e l'HTML: nasce il world wide web.
- HTML si evolve fino alla versione 4.01, specifica congelata al 1999
- Nel 2000 è proposta la specifica XHTML in versione 1.0 con i seguenti obiettivi:
  - Separare la definizione del documento dalla sua formattazione
  - Mantenere una definizione formale rigorosa dei documenti HTML

# Storia

- HTML 4.01 è stato utilizzato per lungo tempo (2003-2013).
- XHTML (ultima versione è la 1.1 del 2001) consente di allargare la compatibilità a differenti dispositivi client ed è un XML Schema che definisce HTML.
- HTML 5 accoglie le caratteristiche fondamentali di XHTML ed è fortemente orientata ai contenuti multimediali, alle pagine dinamiche lato client e alla grafica prodotta client side.

# I moduli di HTML5

- L'HTML5 ha una struttura modulare:
  - Core: Structure, Text, Hypertext, List
  - Text extensions: Presentation, Edit, Bi-directional text
  - Forms: Basic forms, Forms
  - Tables: Basic tables, Tables
  - Images: Image, Client-side image map, Server-side image map
  - Programming: Scripting, Object
  - Heading: Style sheet, link, base, meta
  - Events: Intrinsic events
  - Deprecated: Applet, Style attribute
- Alcuni moduli sono pensati per l'uso con dispositivi aventi risorse limitate come gli smartphone

# Conformità

- Deve essere valido nei confronti delle specifiche DTD XHTML 1.1
- L'elemento radice deve essere `<html>`
- Nell'elemento radice deve essere specificato l'attributo:  
`xmlns="http://www.w3.org/1999/xhtml"`
- Nell'intestazione deve essere presente il DOCTYPE:  
`<!DOCTYPE html>`
- XML Schema open source: <https://github.com/stefangrunert/XHTML5-XML-Schema>



# Conformità

- La definizione del DOCTYPE consente al browser di gestire il documento nella maniera opportuna
- Il testo deve essere sempre racchiuso in un elemento come, ad esempio <p>
- I browser spesso accettano sintassi fuori dello standard
- Questo può provocare problemi nella visualizzazione delle pagine
  - Non tutti i browser si comportano nello stesso modo
  - I browser per dispositivi scarsamente dotati in termini di risorse possono non funzionare

# Attributi di HTML5

- HTML5 definisce attributi comuni a tutti gli elementi in modo da farvi riferimento:
  - id: Identificazione del singolo elemento
  - class: Associazione di elementi anche differenti in una classe comune
- Esempio:
  - Accesso da programma lato client mediante linguaggio di scripting
  - Definizione di caratteristiche di formattazione tramite i Cascade Style Sheet (CSS)

# Contenuto degli elementi

- Specifica degli elementi che possono comparire all'interno di un altro elemento
- Elementi a livello di blocco (block):
  - Possono apparire nel corpo del documento
  - Possono contenere altri elementi a livello di blocco o elementi inline
  - Possono contenere testo
- Elementi inline:
  - Devono apparire all'interno di elementi di blocco
  - Possono contenere testo o altri elementi inline

# Struttura del documento(Structure)

- `<html>`: elemento radice  
deve contenere un elemento `<head>` ed un elemento `<body>`
- `<head>`: intestazione  
deve contenere l'elemento `<title>` che contiene il titolo del documento; può contenere fogli di stile, parole chiave, script lato client
- `<body>`: corpo del documento  
Può contenere gli elementi che andranno visualizzati nella finestra del browser

# Hello HTML5 World!

- Creare un file di testo chiamandolo helloworld.html
- Usare l'editor di sistema (notepad, vi, textedit)
- Attenzione alla gestione delle estensioni!!!
- Dopo averlo salvato, aprire il file con un browser

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello HTML5 World</title>
  </head>
  <body>
    <!-- Corpo del documento -->
    Questo &egrave; il mio primo documento HTML5<br/>
  </body>
</html>
```

# Controllo del testo (Text)

- `<abbr>`, `<acronym>` - inline
  - Definiscono una porzione di testo come abbreviazione ed acronimo
  - L'attributo `title` può contenere la definizione estesa
- `<address>` - block
  - Parte del documento contenente un indirizzo postale
  - Nessuna specifica è fornita sul formato dell'indirizzo
- `<blockquote>`, `<q>` - block, inline
  - Parte di testo definita come citazione
  - L'attributo `cite` contiene la URL della citazione

# Controllo del testo (Text)

- `<br/>` - inline
  - Elemento vuoto che forza il ritorno a capo
- `<cite>`
  - Identifica la fonte della citazione
  - Utile per i riferimenti bibliografici
- `<code>`, `<kbd>`, `<samp>`, `<var>`: inline
  - Usati in documenti di tipo tecnico
  - Parti di documento contenenti codici di programmi

# Controllo del testo (Text)

- `<dfn>`
  - Marca la definizione del termine incluso
- `<div>`: blocco
  - Raggruppa altri elementi semanticamente correlati
  - Raggruppa altri elementi che devono condividere caratteristiche di visualizzazione
  - E' identificabile mediante attributi `id` e `class`
- `<em>`, `<strong>`
  - Parti di testo enfatizzate o molto enfatizzate



# Controllo del testo (Text)

- `<h1> ... <h6>`: block
  - Titolo di un paragrafo
  - Gli indici 1...6 indicano il livello di sottoparagrafo
- `<p>`
  - Paragrafo di testo
  - Il testo è visualizzato su di una nuova linea
  - Al testo è seguito un ritorno a capo
- `<pre>`
  - Marca il testo come preformattato

# Controllo del testo (Text)

- `<span>`: inline
  - Simile a `<div>`
  - Raggruppa testo o altro senza una semantica specificata
- Quando non esplicitamente indicato gli elementi HTML5 sono considerati come all'interno dell'elemento `<body>`

# Controllo del testo (Text)

- Creare un file di testo chiamandolo text.html
- Dopo averlo salvato, aprire il file con un browser

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
<html lang="it">
  <head>
    <title>Controllo del Testo</title>
  </head>
  <body>
    <h1>...</h1>
    <p>... <acronym title="...">...</acronym> ... <strong>...</strong></p>
    <h2> ... </h2>
    <p> ... </p>
  </body>
</html>
```

# Iper testo (Hypertext)

- Il modulo hypertext delle specifiche HTML5 definisce un unico elemento <a> (anchor)
  - Permette di definire sia la sorgente che la destinazione del collegamento
  - L'attributo href contiene l'URI della risorsa connessa

- Esempio:

```
<a id="qui"/>
```

```
<p>Universit&agrave; di Napoli Parthenope: <a  
href="http://www.uniparthenope.it">[web]</a></p>
```

```
<a href="http://informatica.uniparthenope.it#qui">Corso di Laurea in  
Informatica</a>
```

# Immagini (images)

- Il modulo Images delle specifiche HTML5 definisce un solo elemento `<img>`
  - Elemento vuoto di tipo inline
  - Permette di inserire un'immagine all'interno del documento XHTML
- Attributi:
  - `src`: specifica l'URI dell'immagine
  - `alt`: descrizione testuale dell'immagine (obbligatoria)
  - `width`, `height`: dimensioni dell'immagine

# Immagini (images)

- Creare un file di testo chiamandolo image.html
- Dopo averlo salvato, aprire il file con un browser

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
<html lang="it">
  <head>
    <title>Controllo del Testo</title>
  </head>
  <body>
    <p>
      <br/>
      Logo dell'Universit  Parthenope
    </p>
  </body>
</html>
```

# Liste ed Elenchi (Lists)

- Il modulo lists definisce tre tipi di elenchi
  - Non ordinati
  - Ordinati
  - Liste di definizioni
- `<li>`
  - Elemento dell'elenco
  - Può contenere altri elementi
- `<ol>`, `<ul>`
  - Liste ordinate, liste non ordinate
  - Contiene elementi `<li>`

# Liste ed Elenchi (Lists)

- `<dt>`
  - E' il termine da definire in una lista di definizioni
- `<dd>`
  - E' la definizione del termine da definire in `<dt>`
- `<dl>`
  - Lista di definizioni
  - Contiene elementi `<dt>`



# Liste ed Elenchi (Lists)

- Le liste possono essere annidate
- Le liste annidate possono essere di tipo differente
- L'indentazione è gestita automaticamente dal browser

- Esempio:

```
<ul>
```

```
  <li>Programmazione III</li>
```

```
  <li>Tecnologie Web</li>
```

```
  <li>Programmazione su Rete</li>
```

```
  <li>Informatica per applicazioni ambientali</li>
```

```
</ul>
```

# Liste ed Elenchi (Lists)

- Esempio:

```
<dl>
```

```
<dt>I figli dei Simpson</dt>
```

```
<dd>Questa &grave; la lista ordinata per et&grave; decrescente:
```

```
<ol>
```

```
<li>Bart</li>
```

```
<li>Lisa</li>
```

```
<li>Maggie (<em>la pi&ugrave; giovane</em>
```

```
<ul>
```

```
<li>Non parla</li>
```

```
<li>Ha il ciuccio</li>
```

```
</ul>
```

```
</li>
```

```
</ol>
```

```
</dd>
```

```
</dl>
```

# Conclusioni

- HTML5 è un linguaggio di descrizione di pagina
- E' una specifica XML relativa equivalente ad HTML
- Questo comporta uno standard HTML più restrittivo
- HTML5 è stato pensato come evoluzione di HTML 4.01 & XHTML 1.1
- HTML 5.0 è lo standard de facto, ha continuità con XHTML 1.1

# Esercizi

- Creare la propria pagina personale in HTML5
- La pagina deve avere le sezioni
  - Home
  - Vita
  - Attività
  - Hobby
  - Contatti
- Utilizzare almeno gli elementi HTML5 descritti in questa presentazione

# Docker

```
mkdir -p $HOME/TW202122/html
```

```
docker volume create mysql_data
```

```
docker volume create mysql_log
```

```
docker volume create apache_data
```

```
docker volume create httpd_data
```

# Docker

```
docker run -d --name mylespLAMP18.04 \  
    -p 80:80 \  
    -v mysql_data:/var/lib/mysql/ \  
    -v mysql_log:/var/log/mysql/ \  
    -v apache_data:/etc/apache2/ \  
    -v $HOME/TW202122/html:/var/www/html/ \  
    -v httpd_data:/var/log/httpd/ \  
    --restart unless-stopped \  
    mylesp/dockerlamp:latest
```