

# Linguaggi per pagine web intro a **HTML 5**

Lorenzo Sommaruga

Corso Applicazioni Web 1 C02049

Nov. 2018



This work is licensed under a Creative Commons  
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

# Contenuti ed Obiettivi

## Contenuti

- Introduzione al linguaggio HTML5

## Obiettivi

- (BI-1) Conoscere l'evoluzione dei linguaggi standard
- (BI-2) Comprendere la struttura di una pagina HTML5 e i nuovi elementi
- (BI-2) Comprendere le basi per scrivere pagine web

# Contenuti

- HTML5
- Storia
- New elements
  - Canvas
  - Audio
  - Video
  - Local storage
  - Elementi Semantici

# HTML 5

- HTML5 è nato come collaborazione tra W3C insieme a WHATWG (Web Hypertext Application Technology Working Group)
- Dopo XHTML 1: WHATWG lavorava su web forms/ applications, W3C su XHTML 2 con varie proposte fino a Nov. 2009, quando viene abbandonato in favore di HTML 5
- HTML5 ha ancora “lavori in corso”, è già supportato da molti browser
- Ultime ref. W3C:
- HTML5 - A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014, <http://www.w3.org/TR/html5/>
- HTML 5.1 2nd Edition, W3C Recommendation 3 October 2017, <https://www.w3.org/TR/html/>
- HTML 5.2, W3C Recommendation 14 December 2017, <https://www.w3.org/TR/html5/>
- ... tutta la storia su Ref.: <https://www.w3.org/TR/html/introduction.html#introduction-history>

# HTML 5 - Desiderata

- Alcune regole per definire HTML5:
  - Nuove caratteristiche dovrebbero basarsi su HTML, CSS, DOM, e JavaScript
  - Ridurre plugins esterni (e.g. Flash)
  - Miglior gestione errori
  - Scripting sostituito maggiormente da markup
  - Indipendenza HTML5 da devices
  - Sviluppo linguaggio aperto e pubblico

# HTML 5 – Novità

- Alcuni elementi di HTML 4.01 ormai obsoleti, vengono cancellati o ridefiniti in HTML5
- Intenzione di aumentare interoperabilità in user agents (browsers)
- Adattamento ad uso odierno di web
- Considerazione di nuovi elementi per
  - Strutturare meglio
  - Grafica/layout
  - Contenuti multimediali
  - Gestione form + input types

# HTML 5 – Nuovi elementi

- Principali nuove caratteristiche interessanti:
  - <!DOCTYPE...> semplificata
  - Elemento <canvas> per grafica
  - Elemento <audio>
  - Elemento <video> per riproduzione media
  - Miglior supporto per local offline storage
  - Elementi specifici per contenuti: article, footer, header, nav, section
  - Form controls: date, time, date-time, etc., email, url, search
  - Alcuni sono ancora poco supportati dai browsers

# HTML 5 – Tipi di Elementi

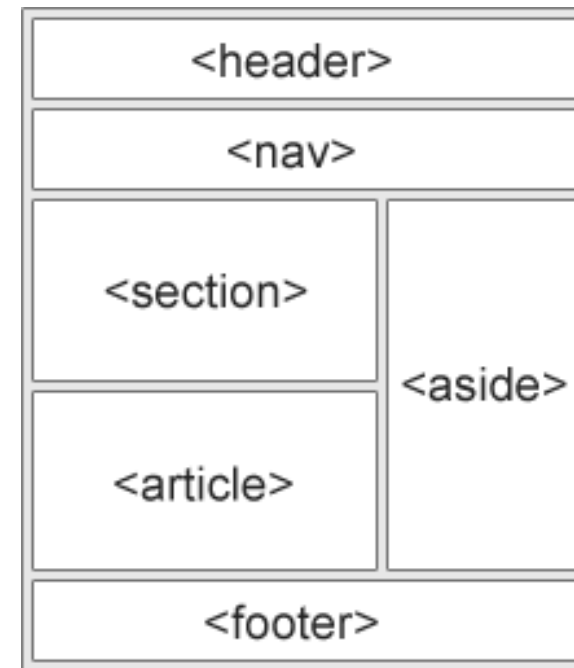
- 5 tipi di *elementi*: [void elements](#), [raw text elements](#), [escapable raw text elements](#), [foreign elements](#), e [normal elements](#)
- **Void elements**: [area](#), [base](#), [br](#), [col](#), [embed](#), [hr](#), [img](#), [input](#), [keygen](#), [link](#), [meta](#), [param](#), [source](#), [track](#), [wbr](#)
- **Raw text elements**: [script](#), [style](#)
- **Escapable raw text elements**: [textarea](#), [title](#)
- **Foreign elements**: elementi da [MathML namespace](#) e [SVG namespace](#)
- **Normal elements**: tutti gli altri elementi [HTML](#) sono elementi normali
- I *Tags* sono usati per delimitare l'inizio e la fine degli elementi nel markup.
- Gli start e end tags di certi [elements](#) possono essere omessi (e.g. elementi vuoti, <br>).
- Ref.: sec 8.1.2 Elements, <https://developer.mozilla.org/en-US/docs/Learn>



# HTML 5 - Elementi semantici

- HTML usava: `<div id="nav">` `<div class="header">` `<div id="footer">` per navigation, header, e footer
- HTML5 offre nuovi elementi per definire parti in una pagina web

- `<header>`
- `<nav>`
- `<section>`
- `<article>`
- `<aside>`
- `<figure>`
- `<figcaption>`
- `<footer>`
- `<details>`
- `<summary>`
- `<mark>`
- `<time>`



# HTML 5 – Elementi Semantici

Tag	Descrizione
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

# HTML 5 – <!DOCTYPE HTML>

- Breve dichiarazione di pagina HTML5 su prima linea documento
- Tag speciale, senza chiusura
- Esempio di primo semplice documento

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <title>Titolo del documento</title>
```

```
  </head>
```

```
  <body>
```

```
    Contenuto del documento ...
```

```
  </body>
```

```
</html>
```

# HTML 5 – Encoding

- In HTML5 semplificato con
- `<meta charset="UTF-8" />`
- In HTML c'era  
`<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`
- Esempio con altro encoding molto usato

```
<!DOCTYPE HTML>  
<meta charset="ISO-8859-1" />  
<html>  
...  
</html>
```

# HTML 5 – <canvas>

- Un canvas è un'area rettangolare di cui può essere controllato ogni suo pixel
- Usato per disegnare grafica al volo in una pagina web
- Utilizza JavaScript per disegnare
- Quando è caricato nel browser corrisponde ad un oggetto con diversi metodi predefiniti per disegnare percorsi, rettangoli, cerchi, etc., e mettere immagini

# HTML 5 – creazione <canvas>

Si definiscono: id, width, e height dell'elemento:

```
<canvas id="mioCanvas" width="300"  
height="200"></canvas>
```

# HTML 5 – Esempio <canvas> + JS

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<canvas id="mioCanvas" width="300" height="200" style="border:1px solid #cccccc;">
```

Your browser does not support the canvas element. (Fallback content)

```
</canvas>
```

```
<script type="text/javascript">
```

```
var c=document.getElementById("mioCanvas");
```

```
var contesto=c.getContext("2d");
```

```
contesto.fillStyle="#00FF00";
```

```
contesto.fillRect(10,10,200,120);
```

```
</script>
```

```
</body>
```

```
</html>
```

*Il canvas è identificato tramite id*

*Si prende il contesto, che  
espone le API per disegnare*

*disegna un rettangolo verde*

*...*

- Provare altri esempi in [http://www.w3schools.com/html/html5\\_canvas.asp](http://www.w3schools.com/html/html5_canvas.asp)

# HTML 5 – <audio>

- HTML5 fornisce uno standard per riprodurre un contenuto audio in una pagina web
- Prima si usavano plugins
- Può fare play di audio file o di stream audio
- Alcuni formati supportati (Ogg, Wav, MP3)
- Inserimento di un audio

Si definiscono: sorgente e controlli

```
<audio src="song.ogg" controls="controls">
```

Your browser does not support the audio element.

```
</audio>
```



# HTML 5 – <audio> Attributi

- Possibili attributi del tag <audio *attributo*="...">

<i>attributo</i>	<b>Valore</b>	<b>Descrizione</b>
autoplay	autoplay	L'audio inizierà la riproduzione non appena pronto
controls	controls	I controlli saranno visualizzati, e.g. il comando play
loop	loop	L'audio re-inizierà la riproduzione dopo il termine (looping)
preload	preload	L'audio sarà caricato (e pronto) al caricamento della pagina. Ignorato con autoplay
src	<i>url</i>	Specifica l'URI dell'audio da riprodurre

# HTML 5 – Esempio <audio> con <source>

- Non tutti i formati supportati da tutti i browser
- Si possono specificare più sorgenti <source>
- Il primo riconosciuto viene usato

```
<audio controls="controls">
```

```
  <source src="song.ogg" type="audio/ogg" />
```

```
  <source src="song.mp3" type="audio/mpeg" />
```

```
Your browser does not support the audio element.
```

```
</audio>
```

# HTML5 - <video>

- HTML5 fornisce uno standard per riprodurre un contenuto video in una pagina web
- Prima si usavano plugins
- Alcuni formati supportati (Ogg, MPEG4)
- Inserimento di un video

Si definiscono: sorgente e controlli, ev. dimensioni video player

```
<video src="movie.ogg" width="320" height="240"  
      controls="controls">
```

Your browser does not support the video tag.

```
</video>
```

Ref.: <http://www.w3.org/TR/html5/video.html#video>

# HTML 5 – <video> Attributi

- Possibili attributi del tag <video *attributo*="...">

<i>attributo</i>	<b>Valore</b>	<b>Descrizione</b>
autoplay	autoplay	Il video inizierà la riproduzione non appena pronto
controls	controls	I controlli saranno visualizzati, e.g. il comando play, pause, stop
loop	loop	Il video re-inizierà la riproduzione dopo il termine (looping)
preload	preload	Il video sarà caricato (e pronto) al caricamento della pagina. Ignorato con autoplay
src	<i>url</i>	Specifica l'URI del video da riprodurre
height	<i>pixels</i>	Specifica l'altezza del video player
width	<i>pixels</i>	Specifica la larghezza del video player

# HTML 5 – Esempio <video> con <source>

- Non tutti i formati supportati da tutti i browser
- Si possono specificare più sorgenti <source>
- Il primo riconosciuto viene usato



```
<video width="320" height="240" controls="controls">  
  <source src="movie.ogg" type="video/ogg" />  
  <source src="movie.mp4" type="video/mp4" />  
  Your browser does not support the video tag.  
</video>
```

# HTML 5 – Local Storage

- Web app possono mantenere dati localmente nel browser: uso pagine con dati offline e performance con risposte rapide (dati locali)
- Cookies erano inclusi nelle richieste al server
- Con local storage dati salvati localmente danno > sicurezza e maggior spazio
- HTML5 fornisce due oggetti per salvare dati sul client:
  - `window.localStorage` - dati senza expiration date
  - `window.sessionStorage` - dati per una session (persi quando si chiude la pagina)

# Scrittura pagine HTML – trovare errori

- Errori comuni si possono trovare tramite verifiche di validità e di conformità
- Strumenti di conformance checkers (*validators*)
- E.g. online di W3C: [Nu Markup Validation Service](https://validator.w3.org/nu/)

<https://validator.w3.org/nu/>

The screenshot displays the Nu HTML Checker interface. At the top, a blue header reads "Nu Html Checker". Below it, a small note states: "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". The main section is titled "Showing results for contents of text-input area". Under "Checker Input", there are tabs for "source" (selected), "outline", "image report", and "Options...". The "Check by" dropdown is set to "text input". The input area contains the following HTML code:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Titolo del documento</title>
</head>
<body>
Contenuto del documento ...
</body>
</html>
```

Below the input area is a "Check" button. Underneath, a note says: "Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings." followed by a "Message Filtering" button. A green banner indicates: "Document checking completed. No errors or warnings to show." Below this, the "Source" section shows a numbered list of the HTML code lines. At the bottom, it states: "Used the HTML parser. Total execution time 2 milliseconds."

# Scrittura pagine HTML – Markup di presentazione

- La maggior parte di *presentational markup* non è più permesso in HTML
- Portava a problemi di:
  - accessibilità (usare markup semantico appropriato), costi di mantenimento/cambi (e.g. `<font color="">`)
  - Errori comuni si possono trovare tramite verifiche di validità e di conformità
  - Ridondanza con grandezza documenti (size)
- Unico markup di presentazione rimasto in HTML è l'attributo `style` e il tag `<style>`
- Da non usare in produzione
- Utili per rapid prototyping, poi le regole definite possono essere spostate direttamente in fogli di stile separati



# Riferimenti HTML5

- Specifica del W3C:  
HTML 5.2, W3C Recommendation 14 December 2017,  
<https://www.w3.org/TR/html5/>
- HTML MDN <https://developer.mozilla.org/it/docs/Learn/HTML>
- Reference Elementi di HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- HTML5 tutorial [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- Libri
  - *Learning PHP, MySQL, JavaScript, CSS & HTML5* – 16 giu 2014
  - *HTML5 Solutions* – 2011 Essential Techniques for HTML5 Developers, Casario, M., Elst, P., Brown, C., Wormser, N., Hanquez, C.

