

Metodologie Informatiche Applicate al Turismo

9. HTML 5

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa

<http://www.di.unipi.it/~milazzo>

milazzo@di.unipi.it

Corso di Laurea in Scienze del Turismo

A.A. 2012/2013

Sommario

- 1 Introduzione
- 2 Caratteristiche principali di HTML 5
- 3 Differenze rispetto ad HTML 4
- 4 Nuove funzionalità di HTML 5
 - Tag semantici
 - Canvas
 - SVG
 - Audio e video
 - Novità riguardo ai form
 - Altre novità per i programmatori

Introduzione

- Negli ultimi tempi la progettazione delle nuove versioni di HTML ha preso una nuova piega
- Se fino a qualche tempo fa l'impressione era che si volesse convergere verso una sintassi con regole più forti (in stile XHTML), ora si sta facendo "marcia indietro"
- La nuova versione 5 di HTML, anzichè cercare di contrastare le **bizzarrie sintattiche** concesse dai vari browser, **le include nel linguaggio**
- Inoltre, HTML5 prevede **nuovi tag** per una migliore strutturazione della pagina e orientati a includere nativamente una più ampia gamma di contenuti multimediali



Un po' di storia... (1)

- I linguaggi per il web (HTML, CSS, e tanti altri) sono gestiti e mantenuti dal consorzio **W3C** (World Wide Web Consortium)
 - ▶ Il W3C è stato fondato nel 1994 da **Tim Berners Lee** (uno degli “inventori” del Web) il quale si occupa tutt'ora di organizzare e indirizzare le attività del consorzio
 - ▶ Tutte le principali aziende informatiche, università e istituzioni di ricerca sono membri del W3C (oltre 350 in totale)
 - ▶ Il W3C utilizza metodi “democratici” per apportare aggiornamenti ai vari linguaggi gestiti.
 - ▶ Il W3C è organizzato in **working groups** tematici: per ogni specifico linguaggio esiste (più o meno) un working group dedicato che se ne occupa
 - ▶ Periodicamente, il W3C rilascia **documenti di specifica ufficiali** (recommendations) delle nuove versioni dei linguaggi per il web

Un po' di storia... (2)

Un po' di cronologia “recente” sulle attività del W3C (e non solo):

- **1997** Rilasciata la recommendation di **HTML 4** (leggermente modificata nel 1999 – HTML 4.01)
- **2000** Rilasciata la recommendation di **XHTML 1**
 - ▶ come HTML 4, ma con regole sintattiche più stringenti
 - ▶ **l'idea**: se un solo tag è scritto male il browser non deve far vedere nulla!
 - ▶ ciò implica che: i vari produttori di browser non possono più aggiungere funzionalità proprietarie al linguaggio (tag/attributi che funzionano solo sul proprio browser)
- **2002** Pian piano inizia il lavoro su XHTML 2.0 (all'interno del working group di HTML)
 - ▶ La nota ufficiale recita: “XHTML 2.0 is a next generation markup language, intended for **rich, portable web-based applications**. Note that while the ancestry of XHTML 2 comes from HTML 4, XHTML 1.0, and XHTML 1.1, **it is not intended to be backward compatible** with its earlier versions.”

Un po' di storia... (3)

- **2004** Apple, Mozilla Foundation e Opera Software (produttori di alcuni tra i principali browser) fondano un gruppo di lavoro **indipendente** denominato WHATWG (Web Hypertext Application Technology Working Group) a cui si associerà anche Google
 - ▶ I produttori di browser sono preoccupati che l'abbandono di HTML per il più rigido XHTML **impedisca loro di poter competere** sulle funzionalità dei loro prodotti
 - ▶ Inoltre, temono che la **non retro-compatibilità** di XHTML 2.0 richieda agli sviluppatori di siti web di dover riscrivere completamente i propri siti
 - ▶ Il WHATWG inizia lo sviluppo di **HTML 5**, aggiornamento di HTML 4 senza le rigide regole di XHTML e molto orientato alla realizzazione di applicazioni web (tanti aspetti di programmazione)

Un po' di storia... (4)

- **2006** Viene **creato** il working group di XHTML 2.0 (con l'idea che questo sostituisca il working group di HTML).
- **2007** Il WHATWG **chiede** al W3C di supportare ufficialmente il linguaggio HTML 5
- **2009** Sulla base del successo ottenuto da HTML 5 (sebbene ancora in via di sviluppo i browser iniziano a supportarlo e gli sviluppatori di siti web iniziano ad utilizzarlo) il W3C **accoglie la richiesta** del WHATWG
 - ▶ Il working group di XHTML 2.0 viene **chiuso** e il progetto interrotto
 - ▶ Il working group HTML viene riaperto con l'obiettivo di realizzare una recommendation per HTML 5 (prevista nel 2014)
 - ▶ Il WHATWG **continua a lavorare in parallelo** ad una specifica di HTML in continua evoluzione in cui gli sviluppatori possono liberamente proporre nuove funzionalità (e competere su esse)

Caratteristiche principali di HTML 5 (1)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titolo del documento</title>
  </head>

  <body>
    ... il contenuto del documento ...
  </body>
</html>
```

- Un documento HTML 5 è nella struttura di base molto simile a un documento HTML 4
- L'unica novità da rilevare è il doctype `<!DOCTYPE html>` che dice al browser che deve aspettarsi un documento HTML 5

Caratteristiche principali di HTML 5 (2)

- Con la versione 5 il linguaggio HTML diventa sempre più legato ai linguaggi
 - ▶ CSS
 - ▶ JavaScript
- Il linguaggio CSS (già visto) diventa **l'unico modo per gestire gli aspetti di presentazione** della pagina web
- Il linguaggio JavaScript (vedremo brevemente) è un vero e proprio **linguaggio di programmazione** che consente di **animare** il contenuto delle pagine web
 - ▶ molte delle nuove caratteristiche di HTML 5 per essere utilizzate richiedono di scrivere codice JavaScript (quindi di programmare!!)
 - ▶ un frammento di codice JavaScript (tanto per rendere l'idea...)

```
var f = document.getElementById("myform");
for ( var i=0; i<10 ; i++) {
  var nuovo = document.createElement("input");
  nuovo.setAttribute("type","text");
  nuovo.setAttribute("name","v"+i);
  f.appendChild(nuovo);
}
```

Differenze rispetto ad HTML 4 (1)

- I tag e gli attributi deprecati in HTML 4 (ad esempio quelli relativi agli aspetti di presentazione e ai frames) **non fanno parte di HTML 5**
- Ad esempio, sono stati rimossi:
 - ▶ I tag `<big>`, `<center>`, ``, `<u>`, `<frame>`, `<frameset>`, ...
 - ▶ Gli attributi `align`, `valign`, `bgcolor`, `border`, `cellpadding`, `cellspacing`, ...

Differenze rispetto ad HTML 4 (2)

- Le principali nuove funzionalità del linguaggio sono le seguenti:
 - ▶ Nuovi **tag semantici** per strutturare meglio il documento e organizzare meglio la navigazione
 - ★ tag <section>, <article>, <header>, <footer>, <nav>, e altri...
 - ▶ La possibilità di scrivere codice HTML (e JavaScript) per **disegnare** nella pagina web
 - ★ tag <canvas> + codice JavaScript
 - ★ incorporamento del linguaggio svg per la grafica vettoriale
 - ▶ La possibilità di gestire flussi **audio e video** in maniera semplice
 - ★ tag dedicati <audio> e <video>
 - ▶ Nuovi **elementi di input** per i form
 - ★ calendari, indirizzi email, sliders, tastiere per i dispositivi mobili, e altri...
 - ▶ Inoltre ci sono molte altre novità utilizzabili dai programmatori tramite JavaScript
 - ★ possibilità di **salvare informazioni** nei computer degli utenti, possibilità di usare applicazioni web anche in modalità **off-line**, eseguire operazioni in **background**, usare il **taglia-incolla** per interagire con gli altri programmi dell'utente, e molto altro

Differenze rispetto ad HTML 4 (3)

- Complessivamente:
 - ▶ HTML 4 era prevalentemente un linguaggio di **descrizione di siti web**
 - ▶ HTML 5 (con CSS e JavaScript) si propone come infrastruttura per la **realizzazione di applicazioni web**

Tag semantici (1)

- In HTML 4 si può usare il tag `<div>` per raggruppare i tag delle varie aree di un documento
- Tramite CSS si può poi descrivere l'aspetto di tali aree

```
<body>
  <h1>La mia Home Page</h1>

  <h2>Biografia</h2>
  <p>Sono nato a....</p>

  <h2>Notizie</h2>
  <p>Notizie aggiornate:</p>

  <h3>Lez. del 14 maggio</h3>
  <p>Lezione su HTML 5</p>

  <h3>Lez. del 7 maggio</h3>
  <p>Lezione su CSS</p>
</body>
```

```
<body>
  <div id="titolo">
    <h1>La mia Home Page</h1>
  </div>

  <div class="sezione">
    <h2>Biografia</h2>
    <p>Sono nato a....</p>
  </div>

  <div class="sezione">
    <h2>Notizie</h2>
    <p>Notizie aggiornate:</p>

    <div class="articolo">
      <h3>Lezione del 14 maggio</h3>
      <p>Lezione su HTML 5</p>
    </div>

    <div class="articolo">
      <h3>Lezione del 7 maggio</h3>
      <p>Lezione su CSS</p>
    </div>
  </div>
</body>
```

Tag semantici (2)

- In HTML 5 invece che usare l'**insignificante** `<div>` si possono usare nuovi tag che hanno un senso all'interno del documento

```
<body>
  <div id="titolo">
    <h1>La mia Home Page</h1>
  </div>

  <div class="sezione">
    <h2>Biografia</h2>
    <p>Sono nato a....</p>
  </div>

  <div class="sezione">
    <h2>Notizie</h2>
    <p>Notizie aggiornate:</p>

  <div class="articolo">
    <h3>Lezione del 14 maggio</h3>
    <p>Lezione su HTML 5</p>
  </div>

  <div class="articolo">
    <h3>Lezione del 7 maggio</h3>
    <p>Lezione su CSS</p>
  </div>
</body>
```

```
<body>
  <header>
    <h1>La mia Home Page</h1>
  </header>

  <section>
    <h2>Biografia</h2>
    <p>Sono nato a....</p>
  </section>

  <section>
    <h2>Notizie</h2>
    <p>Notizie aggiornate:</p>

  <article>
    <h3>Lezione del 14 maggio</h3>
    <p>Lezione su HTML 5</p>
  </article>

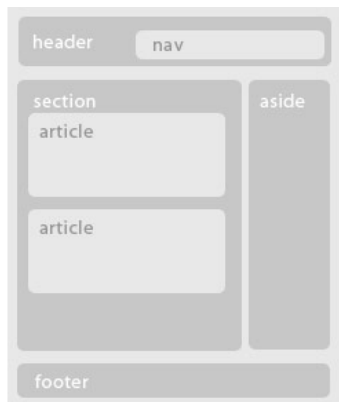
  <article>
    <h3>Lezione del 7 maggio</h3>
    <p>Lezione su CSS</p>
  </article>
</section>
</body>
```

Tag semantici (3)

- I principali nuovi **tag semantici** sono i seguenti:
 - ▶ `<header>` raggruppa elementi introduttivi (e.g. titoli) dell'intera pagina, di una sezione, ecc...
 - ▶ `<footer>` raggruppa elementi conclusivi (e.g. autore, link vari, ecc...) dell'intera pagina, di una sezione, ecc...
 - ▶ `<section>` rappresenta una sezione del documento
 - ▶ `<article>` rappresenta una parte di contenuto autonoma e ripubblicabile (spesso contenuta in una sezione)
 - ▶ `<nav>` contiene link di navigazione all'interno della pagina stessa o verso pagine esterne
 - ▶ `<aside>` raggruppa informazioni correlate ai contenuti principali (spesso visualizzate di lato)
 - ▶ `<hgroup>` raggruppa titoli multipli (es. titolo + sottotitolo) in un'unico elemento
 - ▶

Tag semantici (4)

- Tipica disposizione delle aree corrispondenti ai vari tag semantici
- La disposizione non è automatica, deve essere realizzata tramite CSS



Tag semantici (5)

Vedere esempio: `semantic_tags.html` (e `semantic_tags.css`)

Tag semantici (6)

- Esiste un algoritmo ben definito per estrarre automaticamente un indice della pagina web dai titoli contenuti nelle varie aree opportunamente indicate tramite i tag semantici
- Tale algoritmo (outliner) al momento può essere utilizzato solo installando opportune estensioni nei principali browser
- L'effetto che si ottiene è come in figura

The screenshot shows a web browser window with the following content:

- Page title: I diari di viaggio
- URL: localhost/~sandropaganotti/HTML5/sectioning.html
- Section 1: [I diari di viaggio:](#)
 - 1. [Un week-end a Barcellona](#)
 - 2. [Europa](#)
 - 1. [Giro turistico della Bretagna](#)
 - 2. [Cracovia e la Polonia misteriosa](#)
 - 3. [Africa](#)
 - 1. [Alla scoperta del Kenya](#)

The page content visible below the outliner includes:

- I diari di viaggio**
- A spasso per il mondo
- Un week-end a Barcellona**
- lorem ipsum..
- Europa**
- Giro turistico della Bretagna**

Disegnare nella pagina web: canvas

- Una delle più importanti innovazioni di HTML 5 è la possibilità di **disegnare** direttamente sulla pagina web
- Questo risultato è ottenuto tramite:
 - ▶ il nuovo tag `<canvas>`, che definisce l'area della pagina su cui disegnare (rettangolo visto come un piano cartesiano)
 - ▶ nuove funzionalità JavaScript che consentono di disegnare nell'area definita dal canvas
- La larghezza e l'altezza del canvas possono essere specificati tramite gli attributi `width` ed `height` del tag `<canvas>`
- Le coordinate $(0,0)$ del piano cartesiano corrispondono all'angolo in alto a sinistra

Disegnare nella pagina web: canvas (2)

Un esempio di utilizzo di <canvas>

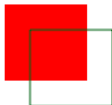
- Codice HTML:

```
<canvas id="c1" width="175" height="175" onLoad="draw();" >  
</canvas>
```

- Codice JavaScript:

```
function draw(){  
  var canvas = document.getElementById('c1');  
  var ctx = canvas.getContext('2d');  
  ctx.fillStyle = "rgb(256,0,0)";  
  ctx.fillRect (20, 20, 65, 60);  
  ctx.strokeStyle = "rgb(0, 100, 20)";  
  ctx.strokeRect (40, 40, 65, 60);  
}
```

- Risultato



Disegnare nella pagina web: canvas (3)

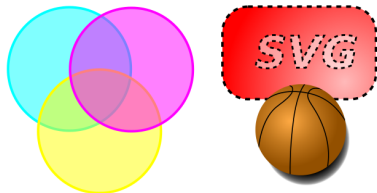
Vedere esempio: `canvas.html`

Disegnare nella pagina web: canvas (4)

- Grazie alle funzionalità di JavaScript i canvas sono particolarmente potenti
- In particolare, esistono funzionalità per la grafica 3D molto evolute
- Inoltre, grazie alla possibilità di catturare l'input da tastiera e mouse si possono creare applicazioni sofisticate che funzionano all'interno del browser
- Alcuni esempi:
 - ▶ <http://zwibbler.com/>
 - ▶ <http://www.benjoffe.com/code/demos/canvascape/>
 - ▶ <http://alteredqualia.com/cubeout/>

Disegnare nella pagina web: SVG (1)

- SVG è un linguaggio di markup per la **grafica vettoriale**
- La grafica vettoriale è una forma di rappresentazione delle immagini come insieme un **insieme di forme di base** molto semplici (linee, quadrati, rettangoli, testo, ...)
- Una caratteristica tipica di questo tipo di grafica è che le immagini possono essere **ingrandite a volontà** senza perdere di definizione
- inoltre, SVG consente di **interagire direttamente** con gli elementi che formano l'immagine
- piccoli esempi di cose che si possono fare con SVG



Disegnare nella pagina web: SVG (2)

Vedere esempio: [svg.html](#)

canvas vs SVG (3)

canvas	SVG
Molto potente in quanto programmabile	Disegni piuttosto semplici
Poco scalabile (zoomando si perde definizione)	Molto scalabile (si può zoomare a volontà)
L'interazione con la figura va programmata	Facile interazione con gli elementi facile

Aggiungere contenuti audio/video (1)

- In HTML 4 i contenuti audio/video possono essere inseriti solo utilizzando estensioni (plug-in) di terze parti
 - ▶ Flash (di Adobe)
 - ▶ QuickTime (di Apple)
 - ▶ Silverlight (di Microsoft)
 - ▶ Real Player (di RealNetworks)
 - ▶ ...
- Ciò lo si ottiene utilizzando i tag `<object>` e `<param>` il cui modo di utilizzo varia a seconda del plugin e del tipo di contenuto (e anche del browser).
- Ad esempio:

```
<object width="420" height="360"  
        classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"  
        codebase="http://www.apple.com/qtactivex/qtplugin.cab">  
  <param name="src" value="liar.wav">  
  <param name="controller" value="true">  
</object>
```

Aggiungere contenuti audio/video (2)

- In HTML 5 sono invece stati aggiunti dei tag **specifici** e **standardizzati** per la riproduzione di contenuti audio e video
 - ▶ `<audio>` per i contenuti audio
 - ▶ `<video>` per i contenuti video
- Ad esempio, per includere un video è sufficiente inserire un frammento di HTML simile al seguente (il funzionamento di `<audio>` è analogo):

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
</video>
```

- Gli attributi `width` e `height` impostano le dimensioni dell'area in cui riprodurre il video
- L'attributo `controls` (senza valore) indica che devono essere messi a disposizione i **pulsanti di controllo** (play, pause, ecc..)
- I vari tag `<source>` specificano **versioni alternative** dello stesso video (secondo diverse codifiche) tra le quali il browser può scegliere

Aggiungere contenuti audio/video (3)

- Quando si inserisce un video (o audio) in una pagina web è necessario fornirne versioni alternative in quanto **non esiste una codifica universalmente accettata** per i contenuti di questo tipo.
- I vari produttori di browser non sono riusciti a trovare un accordo sul formato da usare per i video
 - ▶ Apple vuole H.264 (ossia mp4) in quanto è supportato da iPhone, iPod e Mac OSX, e in quanto detiene alcuni brevetti su questo formato
 - ▶ Mozilla vuole Ogg Theora in quanto è un formato libero (e non intende pagare la licenza di H.264)
 - ▶ Google usa H.264, ma supporta anche un suo formato proprietario WebM
- La soluzione **per ora** è quindi permettere l'uso di diverse codifiche...

Aggiungere contenuti audio/video (4)

Formati video e funzionamento (al momento) sui browser principali:

Browser	MP4	WebM	Ogg
Internet Explorer 9+	SI	NO	NO
Chrome 6+	SI	SI	SI
Firefox 3.6+	NO	SI	SI
Safari 5+	SI	NO	NO

Novità riguardo ai form (1)

- HTML 5 introduce molte novità per **velocizzare**, **semplificare** e **controllare** l'inserimento dei dati da parte dell'utente
- Il tag `<input>` è arricchito con due attributi che permettono di controllare il focus e aggiungere suggerimenti agli utenti:
 - ▶ **placeholder** contiene una frase che sarà visualizzata in un campo di testo se quel campo non ha il focus (ossia non è attivo, non è stato selezionato cliccandoci sopra)
 - ▶ **autofocus** (senza valore assegnato) indica il campo sul quale posizionare il focus al caricamento del form.
- Inoltre sono introdotti molti **nuovi tipi di input** (alcuni mostrati tra poco...).
- Il supporto dei browser tuttavia è ancora molto **eterogeneo** (alcuni funzionano su un browser, altri su un altro, ma nessun browser li ha ancora implementati tutti...)

Novità riguardo ai form (2)

- Alcuni nuovi tipi di input (valore dell'attributo `type` di `<input>`)
 - ▶ **email** casella di testo in cui si assume di dover inserire un indirizzo email. Il browser verifica se il testo inserito contiene il simbolo @ e il dominio è (sintatticamente) corretto. Inoltre, sui **dispositivi mobili** di solito fa comparire la tastiera (virtuale) adatta a scrivere indirizzi email.
 - ▶ **url** casella di testo in cui si assume di dover inserire un URL. Il browser verifica che il testo inserito segua le specifiche degli URL
 - ▶ **number** casella di testo per l'inserimento di un numero con bottoni per incrementare/decrementare il valore. E' possibile specificare il valore minimo, massimo e l'unità di modifica in altri attributi del tag `<input>`. Inoltre, sui **dispositivi mobili** di solito fa comparire la tastiera (virtuale) numerica.
 - ▶ **range** fa visualizzare uno slider (una leva da trascinare lungo una barra di scorrimento) per incrementare o decrementare un valore numerico. E' possibile specificare il valore minimo, massimo e l'unità di modifica.
 - ▶ **date** fa visualizzare un calendario in cui selezionare una data. Esistono vari tipi collegati come `month`, `week`, `time`.
 - ▶ **color** usato per mostrare una tavolozza di colori, da cui selezionare un codice RGB
 - ▶

Novità riguardo ai form (3) - Esempi

```
<form>
  <p>Nome:
    <input name="nome" type="text"
      placeholder="Inserisci qui il tuo nome"
      size="40" autofocus></p>
  <p>Cognome:
    <input name="cognome" size="40" type="text"
      placeholder="Inserisci qui il tuo cognome"></p>
  <p>Email:
    <input name="mail" type="email"></p>
  <p>Lezioni:
    <input type="number" min="5"
      max="25" step="1" value="1"></p>
  <p>Livello:
    <input type="range" name="level"
      min="1" max="10" step="1" value="3"></p>
  <p>Data Inizio:
    <input name="birth" type="date"
      value="2011-04-01"></p>
  <p><input type="submit"
    value="Iscriviti"></p>
</form>
```

Nome:

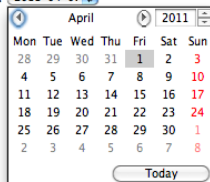
Cognome:

Email:

Lezioni:

Livello:

Data Inizio:



Novità riguardo ai form (4)

- HTML prevede che i valori inseriti nei vari campi del form vengano controllati e validati dal browser (senza inviare nessun dato al server) non appena si preme il bottone submit
- Se qualche dato immesso nel form non è scritto correttamente il browser visualizza un messaggio di errore
- Se tutti i dati sono (sintatticamente) corretti, si procede con l'invio al server
- E' inoltre possibile indicare i campi obbligatori in un form tramite l'attributo `required` del tag `<input>`
- Infine, l'attributo `novalidate` può essere usato per indicare di non controllare la correttezza dei dati dell'intero form o di un singolo campo prima dell'invio

```
<input name="mail" type="email" required>  
<input name="url" type="url" novalidate>
```

Altre novità per i programmatori

- HTML5 (così come definito dal WHATWG) include anche molte novità per gli sviluppatori di programmi applicativi con interfaccia web
- Tutte queste novità sono sostanzialmente nuove funzionalità del linguaggio JavaScript
- Ad esempio:
 - ▶ **Off-line web applications** permette alle applicazioni di funzionare anche in assenza di connessione alla rete
 - ▶ **Copia/incolla** funzioni che consentono di copiare e incollare contenuti (testi, immagini, file, ecc...) da/verso altre applicazioni in esecuzione sul computer (e.g. il desktop)
 - ▶ **Web Workers** consentono di eseguire processi in background senza interrompere l'interazione con l'utente
 - ▶ **Geolocation** consente di recuperare ed utilizzare le informazioni sulla localizzazione geografica del dispositivo utilizzato (PC, smartphone, ecc...). Esempio: calcolare la propria posizione su una mappa
 - ▶ **CSS3** Nuova versione di CSS con tantissimi nuovi effetti
 - ▶ **Tante altre cose...** vedere ad esempio
<http://slides.html5rocks.com> (usando Chrome!)