

Trama del territorio in forma(to) HTML5 (Story plot in HTML5). Hyper Text Markup Language/History Telling Media Layout

Autore: Gianni Messina

Introduzione

HTML 5 è un insieme di nuovi standard per lo sviluppo di contenuti web, e non solo,.

Le tecniche presentate in questo studio di caso sono il frutto di un lavoro di ricerca e testing delle nuove specifiche di aggiornamento del linguaggio html alla versione 5, non ancora rilasciata nella sua versione definitiva. L'attività è finalizzata a rispondere ad un'esigenza emersa in altri ambiti disciplinari: al fine di coadiuvare i colleghi di italiano e Lingue straniere impegnati in un progetto di digital storytelling, gli studenti vengono guidati nella realizzazione di un sito web in HTML5 in grado di ospitare contenuti video.

Descrizione della situazione di partenza

Descrizione del contesto

Nella classe in cui insegno la disciplina informatica gli studenti hanno iniziato a svolgere un'attività di digital storytelling. I colleghi/e di italiano e di lingue straniere

hanno progettato un percorso didattico-formativo basato sul paradigma della narrazione video-digitale.

Il progetto prevede la realizzazione di brevi video (1-3 minuti) per raccontare un evento storico, culturale, artistico o ambientale del territorio accompagnati da una scheda descrittiva funzionale ad inquadrare semanticamente il video in uno di questi contesti.

Durante una riunione alla quale vengo invitato i colleghi, esponendomi quanto stavano realizzando, chiedono il mio contributo facendo emergere due bisogni:

- Creare delle pagine web per permettere la fruizione agevole dei video prodotti in seno al progetto sul sito scolastico.
- Creare una struttura adeguata per la redazione delle schede di accompagnamento previste per la descrizione di ogni video.

Inizialmente, preso dalla frenesia della risposta-soluzione, propongo la creazione di un profilo e il caricamento dei video su un servizio di video-sharing quale "youtube" o "vimeo" e la restituzione degli stessi su una pagina appositamente creata sul sito della scuola. Nulla di difficile sarebbe bastato creare una semplice pagina html e incorporare il codice di condivisione (embed) che tali sistemi mettono a disposizione.

La pagina da realizzare avrebbe restituito un preview dei video prodotti con in calce il link alla scheda descrittiva anch'essa scritta in html o scaricabile in formato testo (pdf, doc, etc.).

Ma riflettendo un pò sulle loro richieste e avendo previsto fra gli argomenti da trattare in classe con gli studenti le specifiche della nuova versione HTML5, decido di intraprendere una strada diversa.

Creare un'occasione di studio per gli allievi del corso di informatica per:

- apprendere i nuovi elementi di markup (marcatori) definiti in HTML5 per descrivere i contenuti delle pagine;
- utilizzare i nuovi tag multimediali per la restituzione dei filmati;
- testare la compatibilità dei browser per il supporto ad HTML5.

Già nel primo semestre del secondo anno lo studio del linguaggio HTML e la redazione di semplici pagine Web rappresenta un argomento fondamentale.

Il laboratorio a disposizione della classe composta da 20 studenti ha in dotazione 10 calcolatori abbastanza recenti di cui 5 calcolatori hanno il sistema operativo Windows Xp e 5 il sistema operativo Linux nella Distro Fedora core 13. Rispetto alle sei ore totali settimanali di lezione previste, Il laboratorio viene utilizzato per tre ore settimanali in compresenza del collega Istruttore Tecnico Pratico.

Svolto durante le ore di laboratorio, lo studio del linguaggio ipertestuale a marcatori (HTML= Hyper Text Markup Language) permette di acquisire un insieme di semplici regole sintattiche e di comprendere consapevolmente ciò che accade quando un browser (il programma che usiamo abitualmente per navigare in internet) aprendo una pagina web ne esamina il contenuto (parsing) e ne restituisce la forma multimediale che viene presentata all'utente.

Il processo di appropriazione delle regole lessicali, sintattiche e semantiche del linguaggio HTML permette agli studenti di approcciare in modo semplice gli

argomenti legati alla programmazione che verranno poi trattati nel triennio con livelli crescenti di complessità.

Inoltre l'immediata spendibilità di quanto appreso, permettendo la creazione di pagine web o piccoli siti web statici, favorisce l'interesse degli studenti che possono sperimentare lo strumento, oltre che per le finalità legate all'apprendimento disciplinare proprio della materia, mettendolo a servizio di altre attività disciplinari o per scopi personali.

Uno degli obiettivi del corso di informatica consiste infatti nell'acquisizione della consapevolezza dei vantaggi e dei limiti dell'uso degli strumenti, dei metodi, dei linguaggi informatici a servizio delle altre aree disciplinari e delle conseguenze culturali e sociali di tale uso.

In tale contesto lo studio della versione 5 del linguaggio viene proposto come strumento funzionale alla risoluzione di un'esigenza reale mostrata dai colleghi per rispondere al loro bisogno di pubblicazione sul web del lavoro svolto all'interno del progetto di digital storytelling.

La metodologia messa in atto, in questo caso, è quindi quella del problem solving di un caso reale attraverso attività laboratoriali di implementazione del servizio richiesto.

La classe che viene coinvolta in questa attività è composta da studenti che hanno già acquisito le regole fondamentali del linguaggio HTML e sono in grado di scrivere pagine web utilizzando editor testuali e "wysiwyg".

Rispetto al lavoro di insegnamento da svolgere ho fatto riferimento al sito del W3C (<http://www.w3c.it/>) ed al manuale trovato sul web all'indirizzo: _____ che mi

hanno permesso l'acquisizione delle nuove regole del linguaggio (versione 5) (Tali regole si trovano nelle schede allegate relativamente ai tag necessari per svolgere questo lavoro, mentre in bibliografia trovate il riferimento alla sintassi completa.

Definizione del problema

Le ore di laboratorio di informatica vengono abitualmente utilizzate per sperimentare praticamente le nozioni teoriche trattate durante le ore di lezione in classe.

Gli studenti vengono coinvolti in attività funzionali alla sperimentazione di quanto appreso nella realizzazione di soluzioni di casi reali che potrebbero in futuro rappresentare le attività con le quali si confronteranno una volta inseriti nel mondo del lavoro.

Questa strategia, unita alla presentazione di casi e problemi calibrati per rispondere agli interessi peculiari della loro età, favorisce l'attenzione e la motivazione.

In questo specifico caso la possibilità di poter realizzare personalmente un piccolo browser video e di acquisire le nozioni di pubblicazione degli stessi, rappresenta un lavoro che identificano immediatamente come un servizio di video-sharing (youtube e simili).

Frequentano abitualmente tali servizi, non sarà necessario spiegare loro quale sarà il risultato atteso, piuttosto bisognerà individuare cosa è e cosa non è possibile implementare con le conoscenze le competenze e le risorse che al momento hanno a disposizione.

In questo quadro l'implementazione di funzionalità aggiuntive al sistema iniziale che si realizzerà sarà l'obiettivo da raggiungere quando verranno acquisite le altre competenze previste nel corso di studio.

Gli strumenti necessari alla realizzazione del lavoro consistono di tre elementi fondamentali:

- un editor delle pagine html5.
- i browser più diffusi per verificare il risultato e il livello di compatibilità nella restituzione di tali pagine.
- un web server locale per il test della pubblicazione del servizio.

Come editor è stato scelto "Scite" <http://www.scintilla.org/SciTE.html> un editor di testo con supporto ad una gran varietà di linguaggi di programmazione. Scite offre funzionalità di editing molto semplici, come la colorazione sintattica e un semplice sistema di completamento automatico per le parole chiave. Si distingue per la rapidità di funzionamento, davvero notevole, anche quando si aprono più schede contemporaneamente. Scite include il supporto a buona parte dei tag html5.

Un secondo editor con funzioni avanzate di completamento che si suggerisce di provare è Komodo edit <http://www.activestate.com/komodo-edit>.

I browser da tenere in considerazione per la lettura delle pagine, come detto i più diffusi, sono stati: Internet Explorer, Mozilla Firefox, Google Chrome e Safari.

Mentre come web server sono stati utilizzati IIS (Internet information Server presente in Windows XP professional e Apache caricato sui sistemi Linux Fedora).

Per la configurazione di IIS su Windows 7 si faccia riferimento alla scheda pratica 1.

Quindi si parte con il lavoro per il quale si prevede di impiegare due settimane : tre ore per lo studio del linguaggio e 9 ore di laboratorio per la realizzazione pratica.

Narrazione del caso

L'organizzazione dell'attività è stata così progettata:

1. 3 ore di lezione teorica che consistono nella consultazione delle schede di lettura proposte:
2. **Scheda di lettura 1:** "Che cos'è HTML5"
Un primo approccio alla versione 5 di HTML per delineare un quadro di contesto;
3. **Scheda di lettura 2:** "Quali sono le novità di Html5"
Una panoramica sui nuovi tag introdotti suddivisi per area
4. **Scheda di lettura 3 :** Differenze fra HTML 4 e HTML5
Un breve report di elementi deprecati e ratio dei nuovi tag.
5. 9 ore di attività pratica in tre blocchi da tre ore nelle quali si procederà all'implementazione del servizio secondo la scaletta sotto riportata.
6. **Laboratorio 1: I tag multimediali**
7. **Laboratorio 2 : la semantica delle pagine**
8. **Laboratorio 3 : assembliamo il tutto**

Scheda di lettura 1 : Che cosa è HTML5?

HTML5 sarà il nuovo standard per HTML, XHTML, e il [DOM](#) HTML .

HTML5 è una cooperazione tra il World Wide Web Consortium ([W3C](#)) e il Web Hypertext Application Tecnologia Working Group ([WHATWG](#)).

La pagina ancora in draft delle specifiche può essere consultata su

<http://dev.w3.org/html5/spec/Overview.html> ancora in draft (bozza) in quanto

HTML5 è un work in progress che diventerà definitivo nella seconda metà del 2012, tuttavia le ultime versioni dei browser più diffusi hanno un supporto ad alcune delle nuove specifiche.

Alcune regole definite dai progettisti per lo sviluppo di HTML5:

- Nuove funzionalità basate su HTML, CSS, DOM e JavaScript;
- Riduzione della necessità di ricorrere a plugin esterni (come Flash);
- Inserimento di meccanismi per una migliore gestione degli errori nelle form;
- Definizione di markup (marcatori del linguaggio comunemente chiamati;
- TAG) per sostituire funzioni delegate allo scripting;
- Indipendenza dal dispositivo utilizzato;
- Processo di sviluppo visibile al pubblico.

Alcune delle caratteristiche più interessanti in HTML5:

1. Introduzione dell'elemento canvas per il disegno;
2. Introduzione di elementi video ed audio per la riproduzione multimediale;
3. Definizione di nuovi contenuti specifici, come l'articolo, piè di pagina, intestazione, nav, sezione
4. Definizione di nuovi controlli per la redazione e il controllo di moduli di raccolta dati (form), come calendario, data, ora, email, url, ricerca.

Supporto del browser

HTML5 non è ancora uno standard ufficiale, e nessun browser fornisce il pieno supporto ma tutti i principali browser (Safari, Chrome, Firefox, Opera, Internet Explorer) continuano ad aggiungere nuove funzionalità HTML5 alle loro ultime versioni.

Scheda di lettura 2: Nuovi elementi in HTML5

Il mondo del web è cambiato molto dal 1999 anno in cui la versione HTML 4.01 diventa uno standard.

Per rispondere alle nuove esigenze, HTML5 include un insieme corposo di nuovi elementi per il disegno, i contenuti multimediali, il miglioramento della struttura delle pagine e la gestione delle form.

Di seguito vengono elencati tutti i nuovi elementi divisi per tipologia.

Nuovi elementi di Markup (Tag)

Miglioramento della struttura di una pagina.:

Tag	Descrizione
<article>	Indica un contenuto autoconsistente quali una notizia, un articolo una news, un blog post, che possono essere letti indipendentemente dal resto della pagina.
<aside>	L'elemento aside rappresenta la parte della pagina, contenente del testo ed affiancata ai contenuti principali del sito, separato dai contenuti stessi. Tale sezione è da considerarsi come una barra laterale nella pagina web.
<command>	Un bottone o un radiobox o un checkbox
<details>	Indica i dettagli di un documento o di parti di un documento
<summary>	Una descrizione del sommario dell'elemento details
<figure>	Raggruppa sezioni pensate per contenere immagini o video.
<figcaption>	Etichetta di una sezione figure
<footer>	Piè di pagina di un documento o di una sezione, può essere il nome dell'autore, la data del documento, le informazioni di contatto o di copyright
<header>	Intestazione di un documento, può includere il menu di navigazione
<hgroup>	Per l'inserimento di elementi titolo quali <h1> .. <h6> ,
<mark>	Evidenziazione di testo
<meter>	Inserimento di misure su un range conosciuto.
<nav>	Per inserire menu di navigazione di una section
<progress>	Lo stato di avanzamento del lavoro

<ruby>	Per annotazioni del framework ruby (Chinese characters)
<rt>	Testo descrittivo di ruby
<rp>	Testo da visualizzare se il browser non supporta ruby
<section>	Sezione di un documento. Si può inserire in headers, footers, o incapsulare in altre sezioni. Definisce semanticamente il contenuto
<time>	Definisce ora o data o entrambi.
<wbr>	Testo a capo

Nuovi elementi per il Multimedia

HTML5 fornisce per la prima volta uno standard per la gestione di audio video e plug-in in una pagina web:

Tag	Descrizione
<audio>	Inserimento di suoni, musica o altri stream audio
<video>	Inserimento di video interni o di stream video
<source>	Definisce le risorse audio, video o stream che devono essere restituite
<embed>	Permette di incapsulare risorse esterne quali i plug-in
<track>	Per inserire sottotitoli testuali al player video

L'elemento CANVAS

Canvas utilizza JavaScript per permettere di disegnare su una pagina web.

Tag	Descrizione
<canvas>	Permette il disegno grafico con l'uso di script

Nuovi elementi per i Form

Tag	Descrizione
<datalist>	Inserisce una lista di opzioni per una casella di testo input
<keygen>	Generatore di chiavi per l'autenticazione degli utenti
<output>	Utilizzato per la gestione di diverse tipologie di output tra i quali i valori ritornati da uno script

Nuovi attributi per le caselle di testo <input>

Tali attributi permettono la gestione semplificata per il controllo dei dati inseriti dall'utente prima che questi vengano inviati al server.

Tipo	Descrizione
tel	Numero telefonico
search	Campo di ricerca
url	Il valore di input deve essere un URL
email	Il valore di input deve essere una o più email
datetime	Il valore di input deve essere una data e/o un ora
date	Il valore di input deve essere un data
month	Il valore di input deve essere un mese
week	Il valore di input deve essere una settimana
time	Il valore di input deve essere un ora
datetime-local	Il valore di input deve essere un data o un ora locale
number	Il valore di input deve essere un numero
range	Il valore di input deve essere un numero compreso nell'intervallo
color	Il valore di input deve essere un colore espresso in esadecimale del tipo #FF8800
placeholder	Visualizza una finestra che suggerisce il tipo di valore atteso

Scheda di lettura 3: differenze fra HTML 4 e HTML5

HTML5 garantisce la **retrocompatibilità con HTML 4.01 o XHTML 1.0**.

Gli elementi e gli attributi non più inclusi in HTML 5, continueranno ad essere supportati ancora per diverso tempo dai browser web, per consentire di adeguare i siti web al nuovo standard in modo graduale.

Comunque è utile fare riferimento al documento ["HTML5 differences from HTML4"](#) (in inglese) per avere il quadro completo delle differenze fra le due versioni e fornisce la ratio delle modifiche apportate o delle novità introdotte.

Html 5 vuole rafforzare la semantica delle pagine web, e per farlo considera obsolete una serie di abitudini degli sviluppatori:

- elementi e/o attributi con significato di stile. Per questo ci sono i CSS
- elementi e/o attributi che hanno causato problemi di accessibilità e usabilità (i famosi frame, frameset, & co.)

Fra i tag esiliati elenco alcuni dei più utilizzati per categoria:

1. **Inserimento di codice esterno:** applet;
2. **Formattazione del testo:** blink, center, dir, font, index, listing, marquee, multicol, plaintext, strike;
3. **Pagine multiple:** frame, frameset, noframesnobr;

Per il resto rimando [alla lista completa dal sito del W3C](#) dove vengono distinti i tag obsoleti ma ancora conformi da quelli invece banditi come non più conformi.

I tag non conformi, pur rimanendo leggibili dai browser per la retrocompatibilità, non permetteranno la convalida della pagina come documento HTML5.

Laboratorio 1: I tag multimediali

Alla luce di quanto descritto nelle schede di lettura emerge che, per il progetto che si intende realizzare, le nuove specifiche sembrano essere proprio quello che serve.

In particolare il supporto multimediale con l'introduzione degli elementi audio e video permetteranno di creare agevolmente il nostro browser video

Iniziamo a sperimentare la prima caratteristica creando una semplice pagina che permette di visualizzare un video.

```
<video>
```

```
  <source src="movie.mp4" type="video/mp4" />
```

```
  <source src="movie.ogg" type="video/ogg" />
```

Il tuo browser non supporta il tag video di HTML5.

```
</video>
```

Come potete notare immediatamente la restituzione di un video rispetto all'inserimento di <Object>, <Embed> e dei vari attributi a corredo necessari nella precedente versione di html è veramente semplice ed intuitivo.

Si tratta dell'inserimento di un semplice tag <video>. All'interno del tag **video** vengono definiti i tag **source** che puntano al file video che si vuole visualizzare.

Possiamo notare qui una caratteristica tipica delle specifiche XML che consiste nell'incapsulamento in relazione gerarchica dei tag:

```
<video> apro tag-padre video  
  <source attributi /> tag-figlio compatibile al tag-padre  
  <source attributi /> " " "  
</video> chiudo tag-padre video
```

Per una descrizione completa del tag video fate riferimento alla scheda **Guida al**

Linguaggio: tag video

Passiamo al test <http://www.teleconsulto.com/video.htm>

La pagina indicata nel link contiene il codice sopra riportato e mostra un breve video.

I browser testati visualizzano correttamente i video solo se questi sono caricati in formato **ogg** ed **mp4**.

Per la codifica dei vostri video in un formato supportato potete utilizzare i software che trovate su questa pagina <http://www.webmproject.org/tools/>

o se utilizzate mac l'ottimo [miro video converter](#)

Il tag video può essere personalizzato con l'uso di attributi specifici:

autoplay	Specifica se il video deve partire non appena è pronto
controls	Permette di mostrare i comandi di riproduzione (play, soto, pause)
height	Specifica l'altezza in pixel del lettore
width	Specifica la larghezza in pixel del lettore
loop	Permette di far eseguire il video a ciclo continuo

preload Permette di decidere di far iniziare il video solo quando è stato caricato interamente. Questo attributo è ignorato se la funzione autoplay è attiv

src Url del percorso in cui si trova il video

poster consente di specificare una immagine che verrà utilizzata all'interno dell'area di riproduzione prima che il video venga eseguito.

Impostiamo quindi larghezza e altezza (width e height) inseriamo una immagine da visualizzare (poster) abilitiamo i controlli (controls) e disabilitiamo il preload per avere un caricamento veloce della pagina visto che prevediamo di inserire diverse finestre video.

```
<video width="320" height="240" controls="controls"
```

```
poster="poster.png" preload="none">
```

```
<source src="movie.mp4" type="video/mp4" />
```

```
<source src="movie.ogg" type="video/ogg" />
```

Il tuo browser non supporta il tag video di HTML5.

```
</video>
```

Prima di passare alla stesura definitiva della nostra pagina prendiamo in considerazione un'altra caratteristica molto interessante del tag video ovvero la possibilità di inserire dei sottotitoli.

Per iniziare bisognerà predisporre un file di testo che contiene i sottotitoli.

Le specifiche legate all'elemento in questione non sono però, ad oggi, supportate da nessuno dei principali browser in commercio; l'estensione di file (.SRT) usata per i sottotitoli è praticamente priva di documentazione ufficiale relativamente all'uso in html5.

Le sole informazioni che si riescono a recuperare parlano di questo tipo di formato:

WEBSRT	• Tipo file
0	• progressivo di testo
00:00:00,1 --> 00:00:03	• intervallo temporale
Ciao	• testo
1	
00:00:03 --> 00:00:06	
Questo breve video	
2	
00:00:06 --> 00:00:10	
per testare il tag video	
3	
00:00:10 --> 00:00:20	
In HTML5	

Salvando questo file con il nome "sottotitoli.vtt" e inserendo questo codice all'interno del tag <Video>

```
<track src="sottotitoli.vtt" kind="subtitles" srclang="it"  
label="Italian">
```

Dovremmo poter visualizzare quanto descritto.

In effetti ciò non accade per la mancata implementazione, al momento, di questa funzione sui browser.

Ricorriamo allora all'uso di librerie molto diffuse quali le [Jquery](#) per ottenere quanto desideriamo:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <title>HTML5: audio e video</title>  
  
  <script type="text/javascript" src="jquery-1.5.2.min.js"> </script>  
  
  <script type="text/javascript" src="mediaelement-and-player.min.js"> </script>  
  
  <link rel="stylesheet" href="mediaelementplayer.min.css" type="text/css" />  
  
</head>  
  
<body>
```

```
<video width="600" height="256" controls="controls"
poster="screenshot.png" preload="none" id="video">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />

  <track src="test.srt" srclang="it" kind="subtitles" label="Italiano" />
```

Il tuo browser non supporta il tag video di HTML5.

```
</video>
<script type="text/javascript">
  $("#video").mediaelementplayer();
</script>
</body>
</html>
```

Teniamo per buono il codice evidenziato in rosso che implementa la funzione di restituzione dei sottotitoli tramite le jquery, senza entrare nel dettaglio che esula da questa lezione,

Il codice evidenziato in azzurro permette di richiamare il file dei sottotitoli. È possibile inserire più tracce per la restituzione in diverse lingue tipo:

```
<track src="test.srt" srclang="it" kind="subtitles" label="Italiano" />
<track src="test_en.srt" srclang="en" kind="subtitles" label="Inglese" />
```

Per testare il tutto l'esempio completo può essere scaricato da

<http://www.teleconsuto.com/public/esempio1/video1.zip>

Prima di concludere l'argomento video, visto che il nostro servizio dovrà funzionare anche nel caso di browser che non supportano ancora questi tag, possiamo ricorrere a un altro set di librerie open source: Video.JS.

[Video.js](#) è una libreria JavaScript e CSS che implementa un "Video Player HTML5" superando i limiti attuali dei browser. Inoltre nel caso di browser datati senza supporto HTML5 permette la restituzione attraverso un player di Flash. La libreria fornisce anche un consistente API JavaScript per l'interazione con il video.

Potete scaricare da

<http://www.teleconsuto.com/public/esempio2/video2.zip>

Il codice completo e le librerie, mentre per una demo potete [cliccare qui](#)

Non ci resta che inserire in una struttura a griglia nella pagina per contenere i video-clip prodotti in seno al progetto di digital storytelling e il nostro primo compito si conclude qui concluso.

Note: è utile conoscere i limiti di HTML5 per quanto riguarda la riproduzione video.

- la riproduzione Full-screen attualmente non rientra nella specifica HTML5;
- i menu forniti di default dal player sono essenziali e dar loro un aspetto più sofisticato o coordinato con un brand aziendale, richiede un certo sforzo utilizzando Java-script;
- non c'è nulla nelle specifiche che permetta uno streaming omogeneo o adattivo, che invece rappresenta una caratteristica essenziale per un sito professionale di riproduzione video;
- la specifica non include, i Digital Rights Management (DRM). Questo significa che i contenuti protetti da copyright non potranno essere distribuiti attraverso HTML5 se non dal detentore dei diritti
- i [codec](#) video pronti sono solo MP4, OGG e WebM

per avere un quadro comparativo generale delle funzionalità implementate sui più diffusi player video potete consultare [questa pagina](#)

Laboratorio 2: La semantica

In HTML5 la definizione di tag specifici, come articolo, intestazione, piè di pagina, sezione permettono di definire le schede a corredo dei clip video utilizzando una struttura sintattica nuova.

HTML5 introduce questi nuovi elementi di struttura e i relativi attributi, con il preciso scopo di rafforzare il valore semantico del documento. I tag in HTML5, infatti, perdono qualsiasi implicazione stilistica e vengono utilizzati per contrassegnare esclusivamente la struttura della pagina.

Vediamo cosa succede nel dettaglio:

Tutto comincia con una semplice istruzione:

```
<!DOCTYPE html>
```

Dichiarazione decisamente minimale, rispetto alle precedenti finora utilizzate, come ad esempio:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Non è più necessario fare riferimento ai diversi modelli che definivano le specifiche HTML4 e XHTML 1, il nuovo standard include tutte le specifiche.

Passiamo alla scrittura della pagina introducendo i nuovi elementi, che verranno utilizzati per realizzare lo scheletro del documento.

Vengono introdotti i nuovi tag <header>, <nav>, <footer>, <section>, <article>, <aside>, <figure>.

Elemento <header>

Contrassegna l'intestazione di una qualsiasi sezione del documento e il suo contenuto deve rappresentare materiale introduttivo (titolo di un paragrafo, note speciali, link di supporto, etc).

```
<header>
```

```
  <h1>Titolo</h1>
```

```
  <h2>sottotitolo </h2>
```

```
  <p>Pubblicato il ...</p>
```

```
</header>
```

Elemento <nav>

Questo tag dovrebbe essere utilizzato per racchiudere link di navigazione o collegamenti che rappresentano un ausilio alla fruizione dei contenuti del documento.

```
<nav>
```

```
  <ul>
```

```
    <li> <a href="/">Home</a> </li>
```

```
    <li> <a href="/schede">Consulta le altre schede</a> </li>
```

```
    <li> <a href="/contatti">Contatti</a> </li>
```

```
  <li> <a href="/chisiamo">Chi siamo</a> </li>
```


</nav>

Elemento <footer>

L'elemento footer viene usato per demarcare la fine (anche in modo NON sequenziale) di una qualsiasi sezione, come per l'elemento header.

Il suo utilizzo più classico potrebbe essere quello relativo al vero e proprio footer di pagina, con indicazioni sul copyright, sull'anno di produzione e sull'eventuale autore. Esempio:

<footer>

<p>© All rights reserved 2012 digitalstorytelling.it</p>

<p>si ringraziano...</p>

</footer>

Ma questo tipo di elemento potrebbe altresì essere riportato all'inizio di una sezione, giusto per definire alcune note ridondanti, che identificano quella determinata sezione, presente magari in altri punti del documento.

Elemento <section>

Questo nuovo elemento ha una definizione non è del tutto chiara che si presta a diverse interpretazioni. La [specifica ufficiale](#) recita:

The section element represents a generic section of a document or application. A section, in this context, is a thematic grouping of content, typically with a heading.

In realtà la sua applicazione è abbastanza facile da individuare: pensiamo alle schede che vogliamo preparare suddivise in sezione tematiche (storia, eventi, ...etc).

Può contenere altri section, non deve essere utilizzato per formattare graficamente i

contenuti, ma usato per raggruppare concettualmente delle aree del documento stesso. Il suo imiego può essere compreso nell'esempio successivo riportato per descrivere l'elemento "article" article.

Elemento <article>

Si usa il tag article quando si vogliono racchiudere porzioni indipendenti di testo, che possono essere estratte dal documento per altri scopi. Quindi pensiamo agli articoli relativi alle nostre schede.

L'elemento article può essere annidato, con l'ammonizione di farlo quando sussiste un legame tra i nodi della catena: esempio i commenti riportati sotto ad un articolo.

Vediamo un esempio:

```
<article>
```

```
<header>
```

```
<h1>Le otto rappresentazioni del Genio</h1>
```

```
</header>
```

```
<section>
```

```
<h2>La Storia</h2>
```

```
<p>Il Genio di Palermo è il nume tutelare della città, o genius
```

```
loci,complementare a Santa Rosalia, protettrice della città. ...
```

```
</p>
```

```
</section>
```

<section>

<header>

<h3>Le otto rappresentazioni del Genio: Gli Eventi</h3>

</header>

<h4>Evento 1</h4>

<p>testo ...</p>

<h4>Evento 2</h4>

<p>testo ...</p>

</section >

<article>

<header>

<h5>Le otto rappresentazioni del Genio</h5>

</header>

Commenti

...

</article>

...

</article>

In questo caso abbiamo utilizzato i tag section all'interno di un article per suddividere varie parti o sezioni dell'articolo.

Non sempre questa è la regola generale. In particolare ricordiamo che article rappresenta il tag utilizzato per racchiudere un contenuto indipendente dal resto della pagina e quindi estrapolabile come elemento autoconsistente dai motori di ricerca. Mentre section viene utilizzato per raggruppare porzioni di contenuto omogenea. Quindi l'esempio sopra riportato potrebbe essere riscritto invertendo section con article, ma questo è ancora materia di discussione per la definizione della regola ufficiale da utilizzare.

Elemento <aside>

Il tag aside, secondo la specifica html5, deve limitarsi ad arricchire il documento con contenuti che potenzialmente possono essere rimossi, senza alterare la completezza della pagina.

Questo nuovo elemento semantico può essere utilizzato per riportare un testo "virgolettato", introdurre uno spazio pubblicitario o aggiungere note a margine.

Esempio:

```
<aside>
```

```
  <header>
```

```
    <h1>La storia del genio di palermo</h1>
```

```
  </header>
```

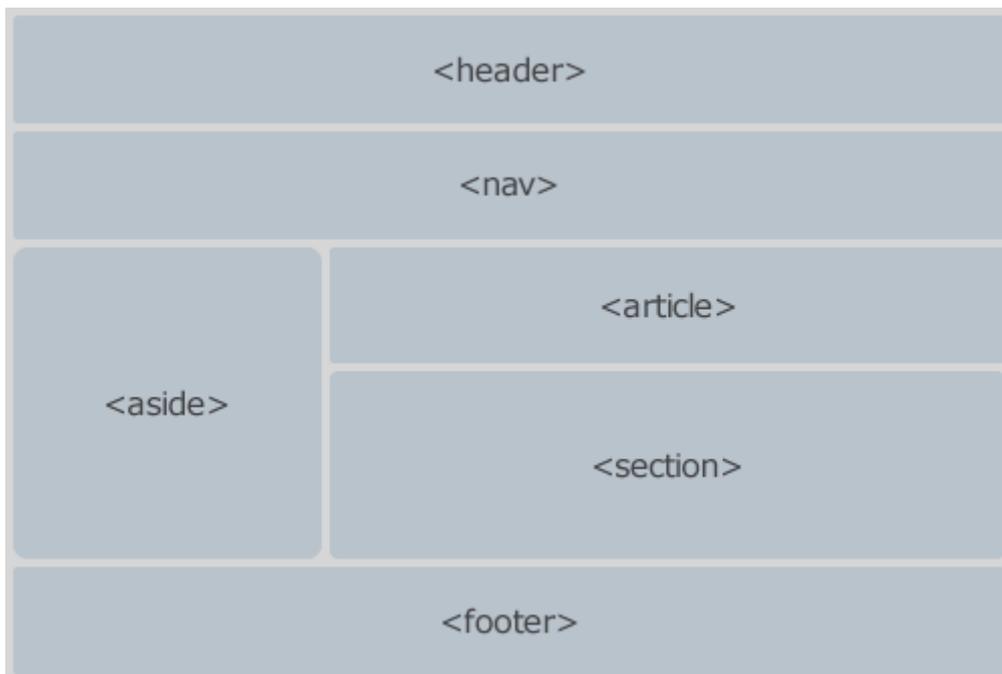
```
  <p>Palermo è famosa per la presenza di statue che raffigurano dei geni .....</p>
```

`<aside>Sei anche tu un genio, iscriviti a Didatec ...</aside>`

`</aside>`

Con un tocco di ironia, ecco un pratico utilizzo di aside. L'informazione può essere rimossa e non costituisce uno scempio al resto dell'articolo.

Ecco quindi in conclusione la struttura tipica di una pagina HTML5 sintatticamente e semanticamente ben formata:



```
<!doctype html>

<html lang="it" >

<head>

</head>

<body>

  <header>

    <div style="width:500px; border:#CCCCCC 1px solid; padding:20px; background-
color:#FFFF66;">

      --- Titolo e Testata del video ---

    </div>

  </header>

<nav>

  <div style="width:500px; border:#CCCCCC 1px solid; padding:20px; background-
color:#0099CC;" >

    --- Voci di Menu --- <br />

    <a href="Index.html">Home</a> | Federico II | Palermo su Treccani

  </div>

</nav>

  <article>

    <div style="width:500px; height:400px; border:#CCCCCC 1px solid;
padding:20px;">

      --- Article: Il genio di Palermo ---

    </div>
```

```
</article>

<section>

<div style="width:500px; border:#CCCCCC 1px solid; background-color:#CCFFFF;
padding:20px;">

    --- Section: I luoghi del genio --- <p> ...</p>

<article>

    <div style="width:500px; height:200px; border:#CCCCCC 1px solid; background-
color:#FFFFCC">

        --- Article: Piazza Rivoluzione ---

    </div>

</article>

<article>

    <div style="width:500px; height:300px; border:#CCCCCC 1px solid; background-
color:#FFFFCC">

        --- Article: La Vucciria ---

    </div>

</article>

</div>

</section>

</body>

</html>
```

Pronti allora per la stesura delle nostre pagine definitive.

Laboratorio 3 : assembliamo il tutto

Bene è ora di creare la struttura definitiva delle pagine e renderle disponibili.

Per questo livello di realizzazione le pagine saranno statiche e per l'aggiornamento delle stesse quindi bisognerà modificare il codice creato.

Iniziamo con il predisporre la pagina demandata a contenere l'elenco di video in una struttura a griglia di questo tipo:

Per fare ciò ricorriamo alla scrittura di un foglio di stile CSS.

Infatti in html5 come già avviene in xhtml 2 le tabelle sono state bandite e al loro posto si indica l'uso di CSS per tutto ciò che riguarda la parte stilistica di formattazione delle pagine.

Esula da questa lezione trattare l'argomento CSS e quindi viene fornita già pronta una struttura adeguata allo scopo.

```
<!-- Griglia di 2 righe e tre colonne ogni casella contiene un Video -->
```

```
<style type="text/css">
```

```
<!--
```

```
.tab {float:left;      • Allinea a Sinistra senza andare a capo  
padding:20px;      • il contenuto del riquadro ha un margine di 20 px  
margin:10px;      • distanza di 10 px ogni riquadro dall'altro  
border:#000000 1px solid • crea il bordo del riquadro  
}
```

```
.newrow {clear:left; • Elimina l'allineamento a sinistra andando a capo  
padding:20px;  
}
```

```
.container{text-align:center; • Crea il contenitore generale  
width:940px;  
  
}
```

```
-->
```

```
</style>
```

```
<!-- Prima riga contenente tre video -->
```

```
<div class="tab">
```

```
Video 1 – Scheda 1
```

```
</div>
```

```
<div class="tab">
```

```
Video 2 – Scheda 2
```

```
</div>
```

```
<div class="tab">
```

Video 3 – Scheda 3

</div>

<!-- Seconda riga contenente tre video -->

<div class="newrow"><hr></div>

<div class="tab">

Video 4 – Scheda 4

</div>

<div class="tab">

Video 5 – Scheda 5

</div>

<div class="tab">

Video 6 – Scheda 6

</div>

Anche questa pagina viene organizzata utilizzando i tag descritti nel laboratorio 2

quindi definiamola in questo modo:

<header>

<h1>La storia, l'architettura e i monumenti</h1>

<h2>raccontati attraverso la leggenda </h2>

<p>Pubblicato il ...</p>

```
</header>
```

```
<article>
```

```
<h3>Federico II</h3>
```

```
<figure>
```

codice per restituire il Video

```
</figure>
```

```
<p> <a href="..."> Guarda la scheda completa </a> </p>
```

```
</article>
```

Una domanda sorge spontanea: quando utilizzare i div?

Se vogliamo organizzare i contenuti per lo styling, allora occorre usare un `<div>`. Il significato del div non è cambiato tra le versioni di HTML e quindi neanche in HTML5.

Se serve un elemento per raggruppare i contenuti e impostare lo stile (larghezza, altezza, font, colori etc.), senza la pretesa di collegamenti o correlazioni di tipo semantico, allora il tag `<div>` è il tag giusto.

Le cose cambiano quando vogliamo che i contenuti abbiano un significato semantico, in questo caso dobbiamo usare i tag che HTML5 ci mette a disposizione.

Allora:

L'elemento <section> rappresenta una generica sezione di un documento o applicazione. Una sezione, in questo contesto, è un raggruppamento tematico di contenuti, in genere con un titolo.

L'elemento [article](#) indica un contenuto autoconsistente che può essere estrapolato dalla pagina mantenendone il significato.

Ricordare inoltre che l'elemento section può essere incluso in un elemento article e viceversa.

Per la redazione delle schede che ci sono state fornite del gruppo che si occupa del progetto di digital storytelling, si ricorre allora all'uso dei nuovi tag utilizzati nel corso del laboratorio 2 per definire senso e organizzare il contenuto dello schema fornito.

Bene, abbiamo portato a termine il nostro lavoro, dopo aver configurato in nostro Web Server come indicato nella scheda pratica 1 possiamo verificarne il corretto funzionamento.

Per scaricare il progetto completo [clicca qui](#)

Per visualizzare un anteprima [clicca qui](#)

Esercizio 1

Sostituire gli elementi video in modo che questi siano compatibili con qualsiasi browser utilizzando il codice fornito nell'esempio 2 del laboratorio 1

<http://www.teleconsuto.com/public/esempio2/video2.zip>

Esercizio 2

Creare una scheda descrittiva per il video e i suoi contenuti e implementarla in html5 seguendo le indicazioni fornite nel laboratorio 2 utilizzando i nuovi Tag di aggregazione (<header> <footer> <nav> <article> e <section>).

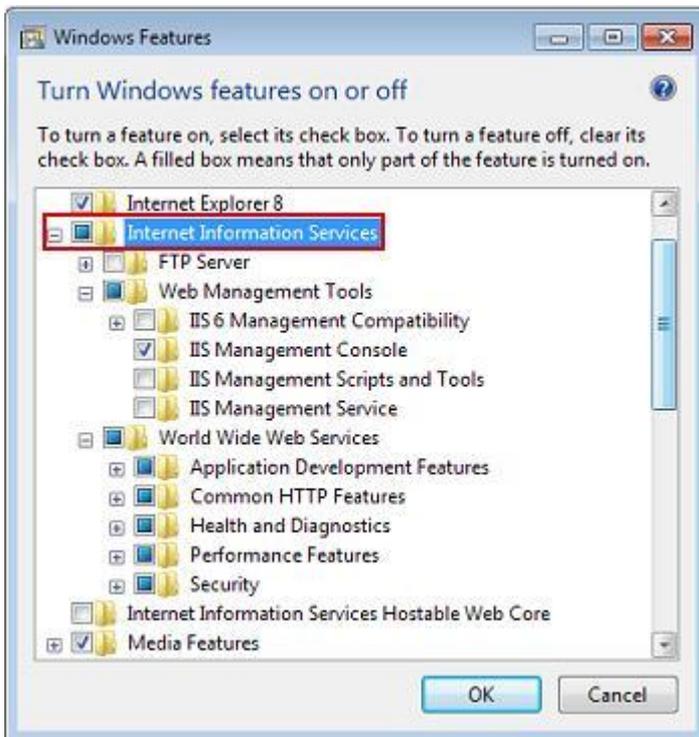
Scheda pratica 1: Installare IIS su Windows 7

Anche se presente su tutte le versioni di Microsoft Windows 7, IIS non viene installato d'ufficio. L'operazione è come sempre molto veloce, a patto di sapere dove mettere le mani.

Clicchiamo su Start -> Control panel -> selezioniamo Programs e successivamente la voce Turn Windows features on or off presente sotto Program and Features.



Spuntiamo ed espandiamo la voce Internet Information Services. Possiamo osservare che non tutte le voci figlie sono state selezionate in automatico. Possiamo farlo adesso o in seguito nel qual caso ce ne fosse bisogno. Clicchiamo su OK per confermare.



Dopo un paio di minuti la procedura avrà termine e e potremo verificare il corretto funzionamento del componente appena installato digitando localhost nella barra degli indirizzi del proprio browser.



Scheda pratica 2: installare Apache web server su Linux Fedora 13 in tre passi

✦ Aprire una finestra terminale e autenticarsi come utente root:

utente#: **su root**

password:

Inserita la password il prompt mostrerà di essere pronto ad eseguire i comandi

come root

root#:

Apache Sever viene scaricato e installato digitando questo comando, se il pacchetto non è presente nella vostra istallazione locale verrà scaricato da internet, quindi è preferibile essere connessi

yum install httpd

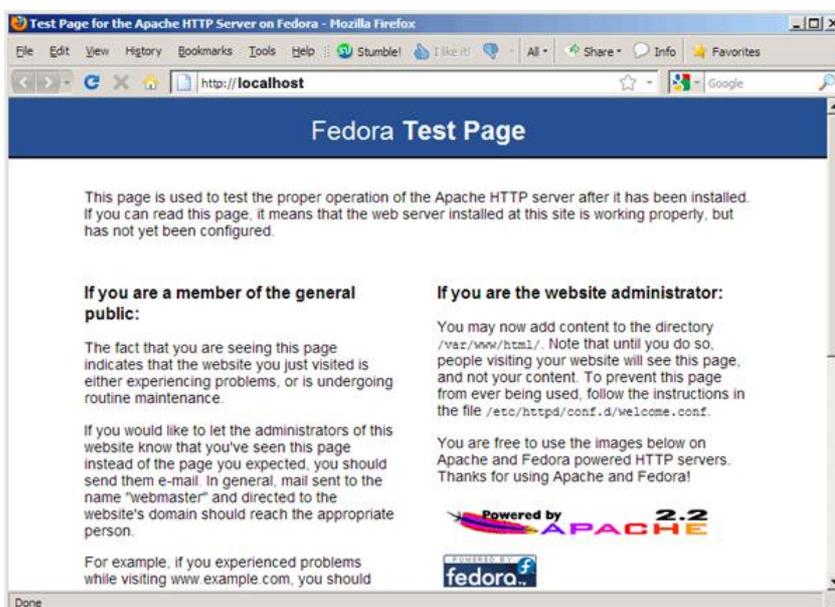
✦ Impostare Apache per essere avviato automaticamente ad ogni riavvio del computer

chkconfig --levels 235 httpd on

✦ Avviare Apache per la prima volta o riavviarlo se già in esecuzione.

service httpd restart

Per testare Apache basta aprire la pagina **localhost** con il browser abitualmente utilizzato



se viene visualizzata questa pagina tutto è andato a buon fine

Altrimenti potete trovare informazioni per risolvere il problema sulla guida in italiano di fedora <http://fedoraproject.org/it/get-help> o su quella completa in inglese <http://fedoraproject.org/en/get-help>

Guida al linguaggio

<VIDEO>

Differenze fra HTML 4.01 e HTML5

<video> tag è un nuovo elemento definito per la prima volta in HTML5.

Note

Qualsiasi testo inserito all'interno dei tag <video> e </video>

Verrà visualizzato dai browser che non supportano il tag

Attributi opzionali

Attributo	Valore	Descrizione
autoplay	autoplay	Il video verrà eseguito all'apertura della pagina
controls	controls	Specifica la visualizzazione della barra di controllo (bottoni play/pause etc).
height	pixels	Altezza in pixel della finestra video
loop	loop	Esecuzione continua in loop
muted	muted	Disabilita l'audio
poster	URL	Immagine di preview visualizzata prima di avviare il video
src	URL	Specifies the URL of the video file
width	pixels	Larghezza in pixel della finestra video

Formati video e support browser

Al momento sono supportati 3 formati video: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer 9	SI	NO	NO
Firefox 4.0	NO	SI	SI
Google Chrome 6	SI	SI	SI
Apple Safari 5	SI	NO	NO
Opera 10.6	NO	SI	YES

- MP4 = MPEG 4 con H264 video codec e AAC audio codec

- WebM = WebM con VP8 video codec e Vorbis audio codec
- Ogg = Ogg con Theora video codec e Vorbis audio codec

Glossario

Codec: è un programma o un dispositivo che si occupa di codificare e/o decodificare digitalmente un [segnale](#) analogico (tipicamente audio o video), affinché possa essere salvato su un supporto di memorizzazione o richiamato per la sua lettura o riproduzione oppure trasmesso a distanza su un [canale](#) di comunicazione.

(Wikipedia)

attualmente sono 5 i codec e i relativi formati video più diffusi con cui è possibile presentare un video sul web e si è ancora lontani dalla definizione di uno standard unico:

- **AVI**

E' un vecchio formato inventato da Microsoft ai tempi in cui trovare contenuti video sul web era molto raro. Ad oggi non viene più ufficialmente supportato dai player più diffusi sebbene la maggior parte di essi sia in grado di leggere filmati .avi.

- **MPEG4**

E' un formato video basato su QuickTime di Apple e, per questo, ampiamente supportato dai dispositivi della Mela oltre che da Explorer su

Windows.

- **FLV**

E' il formato predefinito utilizzato da Flash ed è quindi fondamentale avere il relativo plugin installato sul client. A partire dalla versione 9 il Flash Player supporta, oltre a questo formato, anche i video .mp4. Sebbene gli equilibri inizino a cambiare con l'avvento di HTML 5, il formato .flv è ancora il più diffuso per i filmati sul web.

- **OGG (Theora e Vorbis)**

E' un formato opensource supportato nativamente da browser come Firefox, Chrome e Opera. Nella versione video è conosciuto come Theora mentre nella versione audio come Vorbis. È supportato da tutte le maggiori distribuzioni Linux mentre su Mac e Windows può essere letto installando il plugin per QuickTime. E' destinato ad essere soppiantato dal formato WebM.

- **WEBM**

E' il nuovo formato video opensource sponsorizzato da Google che si prefigge l'arduo obiettivo di diventare uno standard capace di mettere d'accordo tutti. Chrome e Opera già supportano nativamente questo formato; Firefox lo farà a partire dalla versione 4; Adobe ha annunciato che, in un prossimo futuro, il Flash Player sarà in grado di leggere i filmati WebM. Apple e Microsoft, invece, non hanno ancora chiarito del tutto le loro

posizioni: la casa di Cupertino sembra non voler appoggiare i video WebM perchè in netto contrasto con Mpeg4 (come già accaduto per Theora e Vorbis) mentre l'azienda di Bill Gates sembra fornirà un supporto a partire da IE9 ma senza aver specificato se sarà nativo o avrà bisogno di un plugin.

La compatibilità con tutti i browser riguardo ad HTML5 si raggiunge proponendo un singolo video in 3 formati diversi: mp4, ogg e webm. WebM si sta ancora ritagliando un proprio spazio.

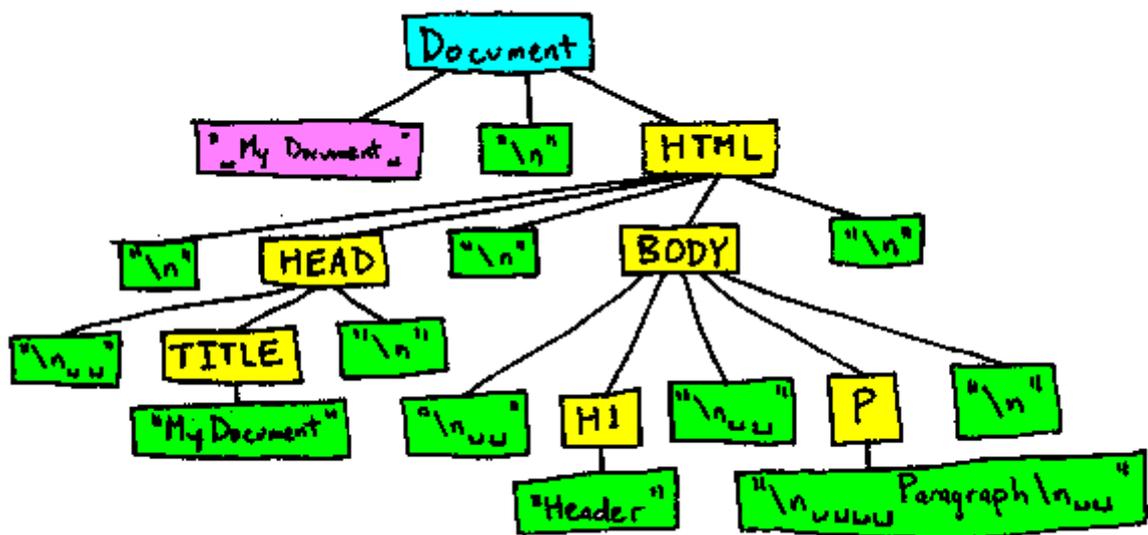
L'ordine dei formati all'interno del tag <video> di HTML5 non risulta casuale e serve per assicurare la migliore compatibilità con tutti i dispositivi.

La sequenza Mp4 - Ogg - WebM raggiunge il massimo della compatibilità con le funzionalità implementate ad oggi sui browser.

Jquery: è una libreria di funzioni (un cosiddetto software framework) javascript, cross-browser per le applicazioni web, che si propone come obiettivo quello di semplificare la programmazione lato client delle pagine HTML. Il framework fornisce metodi e funzioni per gestire al meglio aspetti grafici e strutturali come posizione di elementi, effetto di click su immagini, manipolazione del Document Object Model e quant'altro ancora, mantenendo la compatibilità tra browser diversi e standardizzando gli oggetti messi a disposizione dall'interprete javascript del browser. Sito web: <http://jquery.com/>

DOM Document Object Model: letteralmente modello a oggetti del documento, è lo standard ufficiale del W3C per la rappresentazione dei documenti html come modello orientato agli oggetti.

Un classico esempio di albero DOM è quello generato da un browser nell'interpretazione di un documento HTML mostrando la gerarchia dei tag che compongono la pagina.



Parser: In generale un parser è un software in grado di effettuare l'analisi sintattica di un testo rispetto ad un determinato linguaggio. Nello specifico, un parser HTML è un software in grado di effettuare l'analisi sintattica di un documento HTML ed è in grado di verificare se il documento HTML è ben formato.

CSS : (Cascading Style Sheets o Fogli di stile) è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML. Le regole per comporre il CSS sono contenute in un insieme di direttive (Recommendations) emanate a partire dal [1996](#) dal [W3C](#). L'introduzione del CSS si è resa necessaria per separare i contenuti dalla [formattazione](#) e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine HTML che per gli utenti.

Le specifiche CSS 3 versione che verrà rilasciata insieme ad HTML5 non sono state ancora pubblicate, sebbene il W3C pubblichi costantemente [informazioni](#) sulle novità in fase di sviluppo. I CSS 3 dovrebbero presentare soluzioni per la correzione di alcuni [bug](#) di interpretazione di Internet Explorer, migliorie nella gestione degli sfondi e una soluzione per realizzare i bordi arrotondati la cui realizzazione affligge i webdesigner da tempo.