

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Scienze di Internet

**HTML 5**

-

**evoluzione del linguaggio**

Tesi di Laurea in Ingegneria del software

**Relatore:**  
**Prof. Davide Rossi**

**Presentata da:**  
**Simone Di Benedetto**

**II Sessione**  
**Anno Accademico 2012/2013**



# Indice

<b>Elenco delle figure</b>	<b>5</b>
<b>Introduzione</b>	<b>7</b>
<b>1 Internet e gli strumenti per navigare</b>	<b>9</b>
1.1 Il World Wide Web . . . . .	9
1.2 Le componenti di una pagina web . . . . .	12
1.3 L'importanza dei browser . . . . .	14
1.4 Compatibilità tra browser . . . . .	16
<b>2 HTML: un linguaggio in evoluzione</b>	<b>21</b>
2.1 Un pò di storia . . . . .	21
2.2 Da CSS a CSS3 . . . . .	25
2.3 JavaScript . . . . .	29
2.4 Caratteristiche dell'HTML5 . . . . .	30
2.4.1 New Doctype e Charset . . . . .	31
2.4.2 Nuova struttura . . . . .	32
2.4.3 Nuovi elementi inline . . . . .	33
2.4.4 Nuove Dynamic Pages Support . . . . .	34
2.4.5 Nuovi form types . . . . .	35
2.4.6 Nuovi elementi . . . . .	36
2.4.7 Rimuove alcuni elementi . . . . .	36
<b>3 Il nuovo tag Canvas di HTML5</b>	<b>39</b>
3.1 Il nuovo tag canvas . . . . .	39
3.2 Utilizziamo canvas . . . . .	40
<b>4 Accessibilità</b>	<b>53</b>
4.1 Cos'è l'accessibilità . . . . .	53
4.2 Perché l'accessibilità del web è importante . . . . .	54
4.3 Rendere il Web accessibile . . . . .	55

4.3.1	Testo alternativo per le immagini . . . . .	56
4.3.2	Fornire contenuti organizzati e strutturati . . . . .	56
4.3.3	Fornire alternative testuali per contenuti non testuali . . . . .	57
4.3.4	Associare etichette a tutti i campi del modulo . . . . .	57
4.3.5	Consentire agli utenti di attivare / disattivare i contenuti audio e video . . . . .	57
4.3.6	Ottimizzare la leggibilità del testo . . . . .	58
4.3.7	Non usare il colore come unico indicatore di una possibile azione . . . . .	58
4.3.8	Consentire diversi modi di navigazione . . . . .	58
4.3.9	Prestare attenzione durante lo sviluppo di contenuti dinamici . . . . .	59
<b>5</b>	<b>Conclusioni</b>	<b>61</b>
	<b>Ringraziamenti</b>	<b>63</b>
	<b>Bibliografia</b>	<b>65</b>
	<b>Siti Web</b>	<b>69</b>

# Elenco delle figure

1.1	Percentuale utenti che utilizzando Internet anni 2005-2010 e 2012 . . . . .	11
1.2	Percentuale utenti che utilizzando Internet anni 2005-2010 e 2012 per Regioni . . . . .	12
1.3	Le componenti di una pagina web . . . . .	13
1.4	Browser Statistics and Trends W3C . . . . .	16
1.5	Logo dei principali browser in commercio . . . . .	18
1.6	Elenco browser che supportano CSS3 & HTML5 . . . . .	20
2.1	Logo HTML5 . . . . .	24
2.2	Logo CSS3 . . . . .	26
2.3	Browser che supportano le principali proprietà di CSS3 . . . . .	28
3.1	Area di disegno creta con il tag canvas . . . . .	41
3.2	Disegno di rettangolo rosso . . . . .	42
3.3	Indicazione delle coordinate x ed y . . . . .	43
3.4	Disegno di una linea all'interno di un rettangolo . . . . .	44
3.5	Disegno di un cerchio all'interno di un rettangolo . . . . .	45
3.6	Disegno del testo all'interno di un rettangolo con il metodo fillText() . . . . .	46
3.7	Disegno del testo all'interno di un rettangolo con il metodo strokeText() . . . . .	47
3.8	Sfumatura creata utilizzando il metodo createLinearGradient() . . . . .	48
3.9	Sfumatura creata utilizzando il metodo createRadialGradient() . . . . .	49
3.10	Disegno immagine con drawImage() . . . . .	51



# Introduzione

HTML5 è la quinta revisione ed ultima versione dello standard HTML. Esso offre nuove funzioni che non solo arricchiscono il supporto dei media, ma anche migliorano e semplificano la creazione di applicazioni web in grado di interagire con l'utente, con i suoi dati, rendendo tutto più facile ed efficace di quanto non fosse possibile in precedenza.

Il panorama di Internet è cambiato molto negli ultimi anni, passando dall'utilizzo da parte dell'utente per la fruizione di contenuti tipicamente testuali, alla presenza sempre più preponderante di applicazioni web, agevolate soprattutto dall'aumento della velocità di trasmissione dati.

Da queste necessità e dall'aumento di strutture ricorrenti, quali blog, articoli, menù di navigazione, testo a piè pagina ecc., è nata l'esigenza di sviluppare il linguaggio HTML potenziandolo e rendendolo allo stesso tempo più flessibile.

L'HTML5 nasce per risolvere questi problemi, dando la possibilità agli sviluppatori web di poter utilizzare un linguaggio creato per essere plasmato secondo le più recenti necessità. L'intento iniziale del suo sviluppo è stato quello di integrare nuovi comandi e funzionalità, fino ad allora ottenute mediante il ricorso a plug-in o a estensioni proprietarie dei vari browser.

HTML è il linguaggio che unifica il World Wide Web. Usando solo dei sem-

plici tag in esso contenuti, la razza umana ha creato una rete incredibilmente diversificata di documenti ipertestuali, da Amazon, eBay e Wikipedia, ai blog personali, ai siti web dedicati a specifici argomenti.

HTML5 è l'ultima iterazione di questa lingua franca. Lo possiamo definire come il più ambizioso cambiamento del linguaggio, questo non è la prima volta che HTML è stato aggiornato. L'HTML è un linguaggio in continua evoluzione, non rimane lo stesso per molto tempo.

Come per il web stesso, l'HyperText Markup Language è stato il frutto di Tim Berners-Lee. Nel 1991 ha scritto un documento chiamato Tag HTML in cui propose meno di due dozzine di elementi che potrebbero essere utilizzati per la scrittura di pagine web.

Nel presente lavoro di tesi, dopo una presentazione sulle principali novità e sulla semantica del linguaggio, è stato sperimentato in modo introduttivo l'aspetto riguardante la grafica utilizzando il nuovo tag canvas, elemento supportato da HTML5.



# Capitolo 1

## Internet e gli strumenti per navigare

### 1.1 Il World Wide Web

Internet è diventato così diffuso che è difficile immaginare la vita senza di esso. E' altrettanto difficile immaginare un mondo in cui www non è il prefisso di molte delle nostre attività online. Ma solo perché Internet e il World Wide Web sono saldamente intrecciati tra di loro, ciò non significa che siano sinonimi.

La creazione del World Wide Web (W3) è avvenuto grazie all'aiuto di un uomo di nome Tim Berners-Lee. Nel 1989, ha sviluppato la spina dorsale del World Wide Web. A quel tempo, Tim era un ingegnere software presso il CERN, il laboratorio di fisica delle particelle di grandi dimensioni nei pressi di Ginevra. Molti scienziati partecipavano per lunghi periodi di tempo ad esperimenti al CERN, poi tornavano presso i proprio laboratori dislocati in tutto il mondo. Questi scienziati erano desiderosi di scambiare dati e risultati, ma avevano grosse difficoltà nel farlo. Tim comprese questa necessità, ed il potenziale non realizzato di milioni di computer collegati tra loro attraverso

Internet.

Tim ha documentato quello che sarebbe diventato il World Wide Web con la presentazione di una proposta al CERN alla fine del 1989, questa proposta descriveva l'utilizzo di un insieme di tecnologie che avevano come obiettivo quello di rendere Internet veramente accessibile e utile alle persone. Nell'ottobre del 1990, aveva indicato le tre tecnologie importantissime che rimangono il fondamento del Web di oggi:

- **HTML:** HyperText Markup Language. Il formato di pubblicazione per il Web, tra cui la possibilità di formattare documenti e link ad altri documenti e risorse.
- **URI:** Uniform Resource Identifier. Una sorta di indirizzo che è unico per ogni risorsa sul web.
- **HTTP:** Hypertext Transfer Protocol. Consente il recupero le risorse collegate da tutto il web.

Nel 1991, le persone al di fuori del CERN si unirono alla nuova comunità web, nell'aprile del 1993, il CERN ha annunciato che il World Wide Web sarebbe stato reso disponibile per chiunque lo volesse utilizzare in modo gratuito. Da quel momento, il Web ha cambiato il mondo, è probabilmente diventato il più potente mezzo di comunicazione che il mondo abbia mai conosciuto. Attualmente solo il 25% della popolazione del pianeta sta utilizzando il Web, il Web ha cambiato il modo di insegnare e imparare, comprare e vendere, informare ed essere informati, condividere e collaborare, incontrare ed amare.

Tim Berners-Lee e gli altri si resero conto che perchè il Web raggiungesse il suo pieno potenziale, le tecnologie di base dovevano diventare standard

mondiali. Pertanto, nel 1994, Tim ha fondato il World Wide Web Consortium (W3C) con obiettivo di raggiungere il consenso attorno alle linee guida e per assicurare che il Web funzioni per tutti e che si evolve in maniera responsabile. La Fondazione Web supporta il lavoro del W3C per garantire che il Web e le tecnologie rimangano gratuito e aperte a tutti.

Qui di seguito viene riportato un grafico con il numero di utenti collegati alla rete. Gli utenti di Internet sono le persone che hanno utilizzato Internet negli ultimi 12 mesi da qualsiasi dispositivo, inclusi i telefoni cellulari, i dati sono espressi in percentuale e la fonte è International Telecommunications Unions (ITU), Geneva, 27 February 2013.

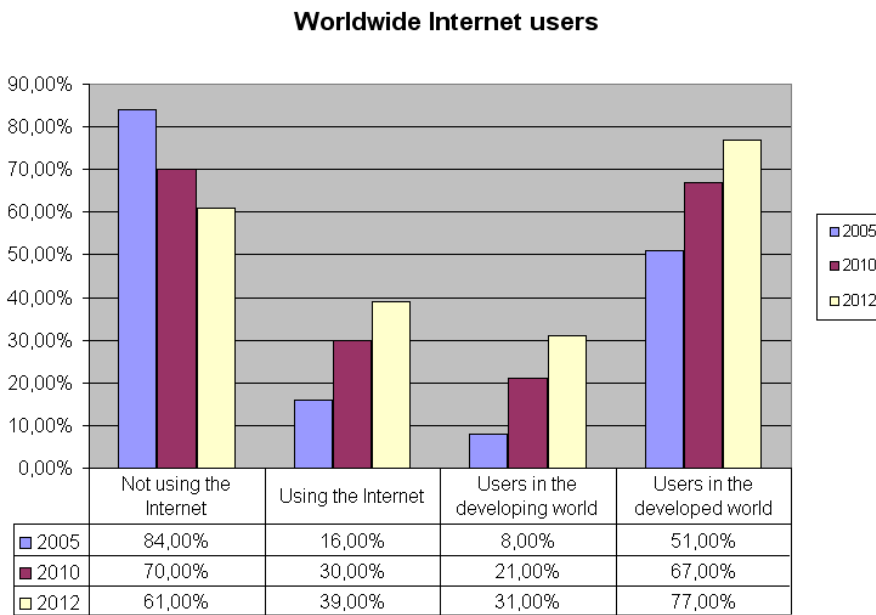


Figura 1.1: Percentuale utenti che utilizzando Internet anni 2005-2010 e 2012

Nel grafico che segue il dato è stato analizzato più in dettaglio, si riportano le percentuali di utenti collegati ad Internet suddivise per aree geografiche, la fonte è sempre International Telecommunications Unions (ITU), Geneva, 27 February 2013.

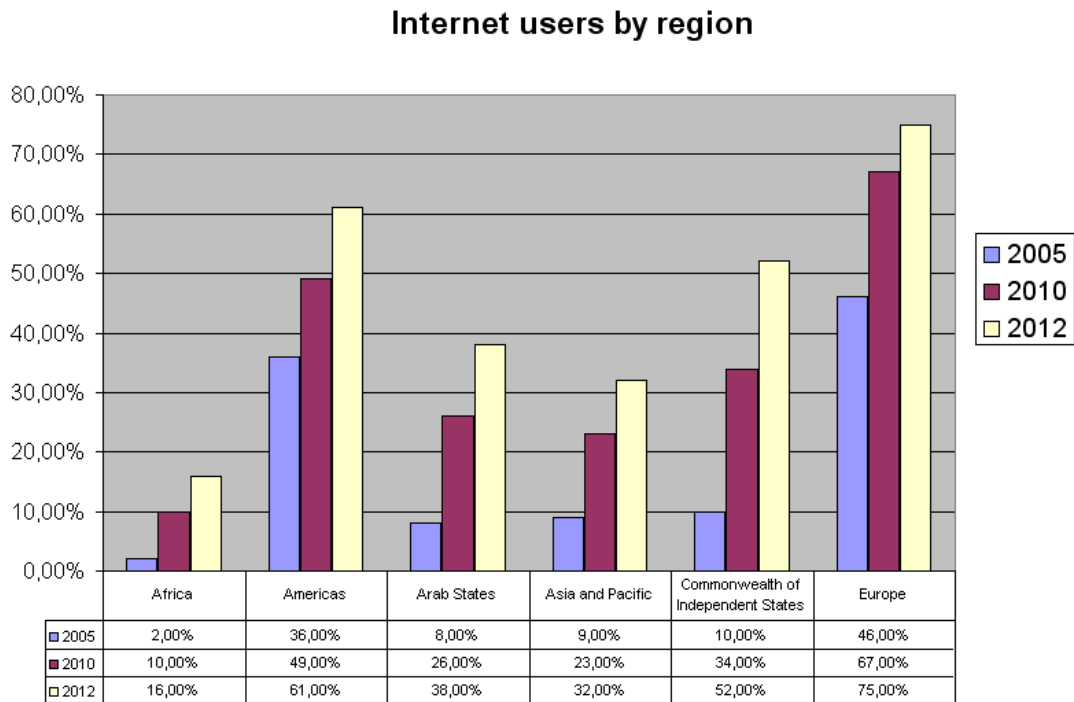


Figura 1.2: Percentuale utenti che utilizzando Internet anni 2005-2010 e 2012 per Regioni

## 1.2 Le componenti di una pagina web

Una pagina web la possiamo considerare come un documento che può essere presentato ai navigatori in modo esteticamente più o meno piacevole e che può o non può contenere informazioni che sono di loro interesse. Da un punto di vista tecnico, se analizziamo gli elementi utilizzati nella costruzione di un documento web possiamo individuare fino a tre livelli: contenuto, presenta-

zione e comportamento.

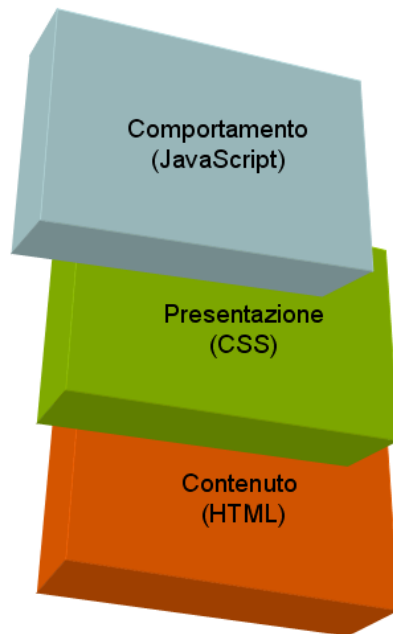


Figura 1.3: Le componenti di una pagina web

Il **livello di contenuto** è sempre presente. Comprende le informazioni che l'autore vuole trasmettere al pubblico, ed è racchiuso all'interno dei markup HTML o XHTML. La maggior parte dei contenuti sul web oggi è il testo, ma il contenuto può essere fornita attraverso immagini, animazioni, audio, video, e qualsiasi altra cosa si vuole pubblicare.

Il **livello di presentazione** definisce come il contenuto apparirà ad un utente che accede al documento. Il modo convenzionale di visualizzare una pagina web è utilizzando un browser web, ma questo è solo uno dei tanti metodi di accesso possibili. Il contenuto, ad esempio, può anche essere convertito in sintesi vocale per gli utenti che hanno problemi di vista o difficoltà di lettura.

Il **livello di comportamento** implica un'interazione in tempo reale dell'utente con il documento. Questa operazione viene in genere gestita da JavaScript. L'interazione può essere qualsiasi cosa, da una convalida banale che assicura un campo obbligatorio di un form di registrazione ad un'applicazione web sofisticata che potrebbe funzionare come un programma desktop.

E' possibile incorporare tutti i tre livelli all'interno dello stesso documento, ma tenendoli separati si ha un notevole vantaggio: siamo in grado di modificare o sostituire uno qualsiasi dei livelli senza dover cambiare gli altri.

Cascading Style Sheets, o CSS, è il metodo consigliato per controllare il livello di presentazione di un documento web.

### 1.3 L'importanza dei browser

Un browser web è definita come un software utilizzato per il recupero e la presentazione di risorse informative sul World Wide Web. Una pagina web, immagini, video e altri tipi simili di contenuti sono considerati risorse informative. Essi sono identificati da un URI (Uniform Resource Identifier). Gli utenti di Internet sono in grado di vedere i contenuti grazie a collegamenti ipertestuali presenti nelle pagine.

Negli ultimi dieci anni c'è stato un aumento del numero di browser che vengono utilizzati per la navigazione su web. Diversi browser interpretano HTML, CSS e JavaScript, con alcune differenze a volte sufficienti per non rendere fruibile un sito web. Dal momento che è difficile giudicare la scelta personale di ogni browser utilizzato dagli utente per visualizzare il sito web, diventa un

requisito fondamentale per un designer Web fare siti web che sono pienamente compatibili con la maggior parte dei browser in commercio oppure con i browser più diffusi.

I principali browser più utilizzati dalla maggior parte dei navigatori sono:

- Internet Explorer
- Mozilla Firefox
- Apple Safari
- Google Chrome
- Opera

Analizzando le statistiche del W3C per l'anno 2013, si vede che il trend di utilizzo del browser web, Internet Explorer (IE), Firefox e Chrome coprono più del 93% degli utenti di Internet. Chrome è il più utilizzato (più del 53%), seguito da Firefox con buon 27% e da IE che è usata da un 12% dei navigatori web.

### Statistiche browser web più utilizzati

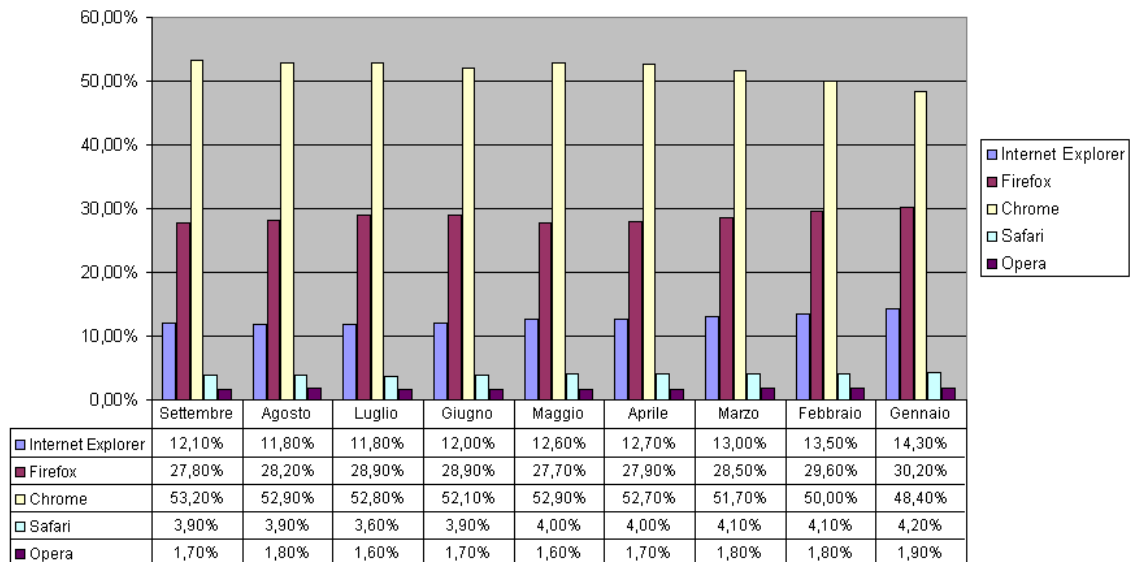


Figura 1.4: Browser Statistics and Trends W3C

Come detto in precedenza browser diversi hanno comportamenti diversi in quanto interpretano in modo diverso i codici HTML, IE è molto indulgente sulla lettura ed interpretazioni di codici HTML errati, ma non tutti i browser si comportano come IE.

## 1.4 Compatibilità tra browser

Una delle attività maggiormente critiche nello sviluppo di un sito web consiste nell'eseguire i test di compatibilità con i diversi browser in commercio. Le difficoltà sono molte e sono dovute al fatto che alcuni tra i browser più popolari, primo fra tutti Internet Explorer, interpretano HTML e CSS in modo diverso, ciascun browser ha delle proprie peculiarità nell'interpretazione.

La compatibilità tra i browser web è un campo minato fatto di conflitti che



si verificano tra i vari browser, ma anche tra browser sviluppati per le diverse piattaforme. Si potrebbero facilmente passare ore, se non giorni, ad ottimizzare le pagine per ogni browser esistente dalla prima versione di Mozilla fino all'ultima versione di Internet Explorer ma ciò non basterebbe a risolvere il problema della compatibilità in quanto basterebbe che uscisse una nuova versione che si dovrebbe nuovamente apportare delle modifiche al sito.

Costruire siti web che visualizzano correttamente le pagine web con i diversi browser è un dato di fatto che i web designer devono affrontare durante il loro lavoro. E' probabilmente uno degli aspetti più difficili da soddisfare, per fortuna ci sono disponibili degli strumenti che aiutano.

Il mercato offre numerosi browser che il singolo utente può utilizzare, tuttavia è inutile e privo di senso cercare di testare il sito su tutti i browser, anche perchè alcuni sono usati da una così bassa percentuale di utenti che non vale veramente la pena faticare per renderli compatibili. La soluzione è quella di scegliere quelli più popolari. A tal proposito il W3C mette a disposizione delle statistiche mensili sull'utilizzo dei browser che può rappresentare un ottimo punto di partenza.

Browser come Firefox, Google Chrome, Safari, Opera ed Internet Explorer sono molto diffusi in quanto sono semplici da trovare, scaricare ed installare, basta scrivere il nome su google e scaricare la versione più recente di ciascun software. E' molto importante che prima che il sito web sia rilasciato vengano eseguiti dei test. Scegliere il browser sul quale lavorare è molto importante non deve essere necessariamente quello più diffuso. La maggior parte dei web designer usa Mozilla Firefox come browser predefinito per la

realizzazione dei siti, anche se non risulta essere il browser più utilizzato. La ragione principale è che Firefox possiede degli addons dal valore inestimabile che sono firebug, pixelperfect, la web developer toolbar e HTML Validator. Inoltre, un sito che viene visualizzato correttamente su Firefox, il 99% delle volte si vede correttamente anche su Google Chrome, Safari e Opera poiché questi browser seguono gli standard del W3C alla lettera.

Il livello di compatibilità tra browser diventa ancora più rilevante quando escono delle nuove feature da poter usare nella programmazione delle pagine, in questo caso nasce spontanea la domanda “a che livello sia” la compatibilità tra browser per la funzionalità che si desidera utilizzare? Volendo entrare nello specifico dei nuovi elementi introdotti da HTML5, possiamo dividere i browser due gruppi:

- browser con alta predisposizione per gli elementi HTML5 (Firefox, Chrome, Opera, Safari)
- browser con bassa predisposizione per gli elementi HTML5 (Internet Explorer)



Figura 1.5: Logo dei principali browser in commercio

Tra i browser che hanno la più alta compatibilità verso i nuovi elementi HTML troviamo Firefox e Google Chrome che hanno in comune l’adattabilità con gli elementi più diffusi `<audio>`, `<video>`, `<canvas>` per poi differenziarsi

l'un l'altro con supporto per altre funzionalità più specifiche. Per analizzare meglio le varie funzionalità supportate per ogni browser è utile fare riferimento alle Feature lists che possiamo trovare tra la documentazione di ogni browser.

Qui si seguito riportiamo una tabella con le diverse funzionalità e le relative compatibilità dei browser distinguendo se piattaforma Windows (WIN) oppure Apple (MAC) fonte <http://www.findmebyip.com>.







	MAC						WIN										
																	
	CHROME	FIREFOX	OPERA	SAFARI	CHROME	FIREFOX	OPERA	6	7	8	9	10					
	25	20	12	15	5.1	6	25	15	12	6	7	8	9	10			
RGBA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%		
HSLA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%		
Box Sizing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	35%		
Background Size	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	90%		
Multiple Backgrounds	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	90%		
Border Image	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	85%		
Border Radius	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%		
Box Shadow	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%		
Text Shadow	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	76%		
Opacity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	91%		
CSS Animations	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	66%		
CSS Columns	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	85%		
CSS Gradients	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	84%		
CSS Reflections	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	46%		
CSS Transforms	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	90%		
CSS Transforms 3D	✓	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	41%		
CSS Transitions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	72%		
CSS FontFace	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	98%		
FlickBox	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	72%		
Generated Content	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	35%		
DataURI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	35%		
Pointer Events	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	29%		
Display: table	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	35%		
Overflow Scrolling	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	3%		
Media Queries	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	34%		

Figura 1.6: Elenco browser che supportano CSS3 & HTML5

## Capitolo 2

# HTML: un linguaggio in evoluzione

### 2.1 Un pò di storia

La storia di Hypertext Markup Language è una storia strana e interessante. HTML è l'abbreviazione appunto di Hyper Text Markup Language che è la lingua ufficiale del World Wide Web e fu concepito nel 1990. HTML è un prodotto di SGML (Standard Generalized Markup Language), che è una specifica tecnica che descrive i linguaggi di markup, in particolare quelli utilizzati nello scambio elettronico, nella gestione e nella pubblicazione di documenti.

HTML è stato originariamente creato per consentire a coloro che non sono stati specializzati in SGML di pubblicare e scambiare documenti tecnico scientifici e di altro. HTML facilita particolarmente questo scambio, integrando la capacità di unire i documenti tramite collegamenti ipertestuali . da qui deriva il nome di Hyper Text Markup Language.

Tuttavia, si è capito subito che l'HTML è relativamente facile da impara-

re, è autonomo e si prestava a una serie di altre applicazioni oltre che allo scambio di documenti. Con l'evoluzione del World Wide Web, l'HTML ha cominciato a proliferare e rapidamente.

Ben presto, le aziende hanno iniziato la creazione di browser, software necessario per visualizzare un documento HTML, questo ha dato origine alla concorrenza con altri browser web. Può sorprendere alcuni che alla fine del 1995, Netscape è stato il browser più utilizzato sul mercato. Infatti, Netscape fu il primo browser a supportare Javascript, gif animate e frame HTML.

Inizia così la cosiddetta guerra dei browser, ogni costruttore tendeva ad implementare propri elementi HTML che funzionavano solo con il proprio browser. Alcuni esempi di questi sono i

```
<marquee> ... </marquee>
```

tag (testo scorrevole) che in origine ha funzionato solo con Internet Explorer e la

```
<blink> ... </blink>
```

tag (testo lampeggiante) che ancora funziona solo con browser come Firefox.

Un effetto collaterale di tutto questo è che l'HTML era diventato eccessivamente frammentato e che le pagine di molti siti web si visualizzavano bene con un browser ma non con un altro. Quindi la creazione di una pagina web che fosse visibile in modo uniforme sui diversi browser richiedeva sempre più tempo. Questo fenomeno rimane in misura ridotta anche oggi.

Nel frattempo, un'organizzazione nota come World Wide Web Consortium

(W3C in breve) stava lavorando costantemente per standardizzare HTML. Diverse raccomandazioni sono state pubblicate dal W3C alla fine del 1990, che rappresentavano le versioni ufficiali di HTML e fornivano un riferimento completo per i webmaster. Così la nascita di HTML 2.0 nel settembre del 1995, HTML 3.2 nel gennaio 1997 e HTML 4.01 nel dicembre 1999.

Ormai, Internet Explorer (IE), aveva eclissato Netscape Navigator come il browser da utilizzare durante la navigazione in rete grazie alle sue superiori capacità, ma anche in gran parte a causa del fatto che l'IE era venduto in bundle con il sistema operativo Windows. In sostanza, quando la gente acquistava un computer che utilizzava il sistema operativo Windows, aveva Internet Explorer già installato.

HTML5 è l'ultima versione del linguaggio di markup destinato a sostituire non solo HTML 4.01, ma anche XHTML 1.1.

Come i suoi predecessori HTML 4.01 e XHTML 1.1, HTML5 è una risposta al fatto che il codice di uso comune sul World Wide Web è una miscela di caratteristiche introdotte da varie specifiche e rappresenta anche un tentativo di definire un unico linguaggio di markup che può essere scritto utilizzando la sintassi HTML o XHTML. Per le stesse ragioni, HTML5 è anche un potenziale candidato per applicazioni mobile cross-platform. Molte caratteristiche di HTML5 sono state costruite considerando il fatto che potessero funzionare su dispositivi a bassa potenza come smartphone e tablet.

HTML5 aggiunge molte nuove caratteristiche, tra queste i nuovi elementi `<video>`, `<audio>` e `<canvas>`, così come l'integrazione con grafica vettoriale scalabile SVG e MathML per le formule matematiche. Queste caratteristi-



Figura 2.1: Logo HTML5

che sono state progettate per rendere più semplici i contenuti sul web, senza dover ricorrere a plugin proprietari oppure ad API . Altri nuovi elementi, come `<section>` , `<article>` , `<header>` e `<nav>`, sono progettati per arricchire la semantica del contenuto dei documenti.

Spesso, erroneamente, HTML5 è paragonato a Flash, tuttavia le due tecnologie sono molto diverse tra loro, entrambi includono funzionalità per la riproduzione audio e video all'interno di pagine web, e per l'utilizzo di Scalable Vector Graphics ma l'HTML5 non può essere usato da solo per l'animazione e l'interattività, deve essere completato con CSS3 e JavaScript. Ci sono inoltre molte funzionalità di Flash che non hanno alcuna controparte in HTML5.



## 2.2 Da CSS a CSS3

CSS è l'acronimo di Cascading Style Sheets è un linguaggio di fogli di stile usato per descrivere la semantica di presentazione (l'aspetto e la formattazione) di un documento scritto in un linguaggio di markup. La sua applicazione più comune è nelle pagine web scritte in HTML o in XHTML.

CSS è stato progettato principalmente per consentire la separazione del contenuto del documento (scritto in un linguaggio di markup HTML o simile) dalla presentazione del documento, compresi elementi quali la struttura, i colori i font. Questa separazione può migliorare il tenore di accessibilità, fornire maggiore flessibilità e controllo consentendo a più pagine di condividere la formattazione e ridurre la complessità e la ripetizione nel contenuto strutturale.

CSS consente anche che la stessa pagina sia presentata in diversi modi sui diversi dispositivi di rendering, ad esempio un formato per lo schermo ed uno diverso per la stampa. Può anche essere utilizzato per consentire alla pagina web, di essere visualizzata in modo diverso a seconda delle dimensioni dello schermo o del dispositivo su cui viene visualizzato.

Le specifiche del CSS sono mantenute dal World Wide Web Consortium (W3C). CSS ha una semplice sintassi ed utilizza un certo numero di parole inglesi per specificare i nomi di diverse proprietà di stile. Un foglio di stile consiste di un elenco di regole. Ogni regola è composto da uno o più selettori.

Nei CSS, i selettori sono utilizzati per dichiarare quale parte del markup stile si applica a un tipo di testo. I selettori possono essere applicate a: tutti

gli elementi di un tag specifico oppure a determinati elementi attraverso l'utilizzo di un identificatore univoco id.

L'introduzione dei CSS è avvenuta ad opera del W3C fin dal 1996. Il CSS2 è apparso nel 1998 e da allora sono successe molte cose su Internet. La sua unica revisione è stata fatta nel 2011, CSS2.1, ma si trattava di una soluzione temporanea, giusto il tempo necessario per rilasciare la versione CSS3. Quello che molti non sanno è che lo sviluppo di CSS3 era iniziato solo un anno dopo la presentazione della sua versione precedente. Così il W3C ha iniziato a lavorare su questa versione migliorata a partire dal 1999, per più di 12 anni fino a quando è stata rilasciata la prima versione stabile CSS3.



Figura 2.2: Logo CSS3

Tutti i principali browser supportano le caratteristiche più diffuse di CSS3, come per HTML5, anche per i CSS3, il consorzio W3C considera

il progetto in continua evoluzione.

Forse la più grande differenza tra CSS2 e CSS3 è la separazione in moduli. Mentre nella versione precedente CSS2 era tutto un'unica grande specifica che definiva caratteristiche diverse, in CSS3 è suddiviso in diversi documenti che sono chiamati moduli. Ogni singolo modulo è dotato di nuove funzionalità, che non va ad inficiare la compatibilità con la versione precedente.

Le principali funzionalità CSS3 che sono supportate dalla maggior parte dei browser sono:

- **border-radius** Arrotonda gli angoli di un elemento con un valore di raggio indicato
- **text-shadow** Una proprietà CSS2.1 poi reintrodotta in CSS3 che aggiunge un'ombra di ipertesto, con opzioni per la direzione, la quantità di sfocatura e il colore dell'ombra.
- **box-shadow** Aggiunge un'ombra ad un elemento.
- **multiple background images** CSS3 aggiunge la possibilità di applicare più immagini di sfondo su di un elemento
- **opacity** definisce il livello di opacità di un elemento. Un valore 1 significa opacità massima, mentre un valore pari a 0 significa completamente trasparente.
- **RGBA** Non è una proprietà CSS, ma piuttosto un nuovo modello di colore introdotto in CSS3, aggiungendo la possibilità di specificare un livello di opacità con un valore di RGB

Nella tabella seguente, per ciascuna funzionalità vista vi è l'elenco dei browser con la relativa versione che la supportano.






























PROPERTY	SUPPORTED IN				
border-radius	 3+	 3+	 1+	 10.5+	 9 beta
text-shadow	 1.1+	 2+	 3.1+	 9.5+	
box-shadow	 3+	 3+	 3.5+	 10.5+	 9 beta
Multiple background images	 1.3+	 2+	 3.6+	 10.5+	 9 beta
opacity	 1.2+	 1+	 1.5+	 9+	 9 beta
RGBA	 3.2+	 3+	 3+	 10+	 9 beta

Figura 2.3: Browser che supportano le principali proprietà di CSS3

## 2.3 JavaScript

JavaScript è un linguaggio di scripting usato per rendere le pagine web interattive, la sua implementazione consente agli script lato client di interagire con l'utente. La sua sintassi è stata influenzata da Java. Anche se condivide molte delle caratteristiche e delle strutture del linguaggio Java, è stato sviluppato in modo indipendente. JavaScript può interagire con il codice sorgente HTML, consentendo di rendere dinamici i contenuti dei siti web. JavaScript è sostenuto da un certo numero di società di software ed è un linguaggio aperto chiunque può utilizzare senza l'acquisto di una licenza.

JavaScript è stato originariamente sviluppato da Brendan Eich, chiamato inizialmente LiveScript, prima dell'uscita su Netscape Navigator 2.0 nel settembre 1995 è stato rinominato JavaScript. La scelta finale del nome ha causato confusione, dando l'impressione che fosse uno spin-off del linguaggio di programmazione Java. JavaScript è stato formalizzato nel linguaggio standard ECMAScript ed è utilizzato principalmente come parte di un browser web (client-side JavaScript).

L'implementazione JavaScript di Microsoft è stata successivamente rinominata JScript per evitare problemi legati al marchio. JScript aggiunto nuovi metodi di data che in JavaScript sono basati sulla classe Java `java.util.Date`.

In HTML5, JavaScript fornisce il supporto per i tag multimediali ed i tag grafici. Utilizzando il tag `<video>` ed il tag `<audio>` è possibile visualizzare una serie di controlli per la riproduzione multimediale, tramite JavaScript si potrà avviare, stoppare l'esecuzione, ma anche riavvolgere la riproduzione multimediale. Questo rappresenta tuttavia il caso più comune di utilizzo

delle API ma si possono anche creare i propri controlli del player personalizzati, mantenendo i controlli di default nascosti. I controlli di default sono generalmente adatti per gli usi più comuni, ma se si vuole il pieno controllo l'utilizzo delle API JavaScript è necessario.

Anche il rendering dell'elemento `<canvas>` si basa quasi interamente su l'API JavaScript. Il tag `<canvas>` consente di definire vari comandi di disegno come linee e forme tutto tramite JavaScript, ad esempio si possono richiamare metodi come `lineTo()`, `rect()`, `circle()`. Una volta che si dispone di un gruppo di forme, è possibile iniziare a disegnare nella sezione, naturalmente è possibile applicare anche una varietà di stili, ad esempio per la linea è possibile stabilire la larghezza, la lunghezza, il colore, ecc. mentre per una circonferenza il colore di riempimento, la grandezza, ecc.

Così come è possibile disegnare su un tag, `<canvas>`, si può anche cancellare tutto o parte del disegno e disegnare di nuovo. In questo modo, `<canvas>` consente di creare animazioni basate sui fotogrammi dei disegni. Il tag `<canvas>` viene usualmente utilizzato per includere grafici, animazioni grafiche, giochi, visualizzazione audio, effetti video e molto altro.

## 2.4 Caratteristiche dell'HTML5

L'HTML è inizialmente nato come linguaggio per descrivere documenti scientifici, ma nel corso degli anni si è notevolmente diffuso diventando un standard per qualsiasi pagine pubblicata su Internet. Con il nuovo linguaggio HTML5 si cercherà di porre rimedio ad alcuni limiti che sono presenti. Inoltre HTML5 prevede il supporto per la memorizzazione locale di grosse quantità di dati scaricate dal browser e la nascita di elementi specifici per il multime-

dia, di cui Internet è molto ricca.

E' fondamentale ricordare che HTML5 è costruita sul successo della sua versione precedente, HTML 4.01, che è senza dubbio la versione migliore di sempre. HTML5 è stato creato per rendere il processo di codifica più facile e più logico. Grazie all'introduzioni di nuovi elementi, di contenuti multimediali e di grafica sul web saranno gestiti ed eseguiti più velocemente e senza bisogno di plugin o API.

Passiamo ora in rassegna le principali novità.

### 2.4.1 New Doctype e Charset

La sintassi di HTML5 richiede la dichiarazione di un doctype in testa al documento per assicurare che il browser renderizzi correttamente la pagina; la sintassi corretta è riportata nel codice minimale che segue:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Titolo documento</title>
  </head>
  <body>
    <p>paragrafo</p>
  </body>
</html>
```

L'esempio riportato mette in evidenza la semplificata dichiarazione del doctype

```
<!doctype html>
```

rispetto all'HTML 4.01 dove avevamo

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" \\ "http://www.w3.org/TR/html
```

almeno, questa volta è molto più semplice rispetto a prima ed è molto facile per ricordare e non dovremo cercarlo sul web o copiarlo da sito a sito. Si nota anche la semplificazione nel definire l'encoding della pagina semplicemente

```
<meta charset="UTF-8">
```

mentre in HTML 4.01 si doveva definire come

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

## 2.4.2 Nuova struttura

L'HTML5 riconosce che le pagine web hanno una struttura, proprio come i libri hanno una struttura. In generale, le pagine Web hanno una intestazione, una barra di navigazione, un corpo e le piè di pagina. HTML5 risolve questo problema fornendo un insieme di tag che definisce più chiaramente i grandi blocchi di contenuto che costituiscono un documento HTML. Indipendentemente dal contenuto finale visualizzato, la maggior parte delle pagine web sono costituiti da diverse combinazioni di sezioni ed elementi comuni.

- `<section>`  
definire sezioni di pagine
- `<header>`  
definisce l'intestazione di una pagina
- `<footer>`  
definisce il piè di pagina



- `<nav>`  
definisce la navigazione su una pagina
- `<article>`  
questo tag definisce un articolo, un commento utente o un post, quindi un elemento indipendente di contenute
- `<aside>`  
definisce i contenuti extra, come una barra laterale su una pagina
- `<figure>`  
definisce le immagini che annotano un articolo

### 2.4.3 Nuovi elementi inline

HTML utilizza elementi in linea per contrassegnare i dati come le citazioni, il codice di un programma, le immagini ed i link. Elementi inline non possono essere posti direttamente nel `<body>` ma devono essere annidati all'interno di un elemento di blocco, come ad esempio `<p>`.

I nuovi elementi inline sono:

- `<mark>`  
per indicare il contenuto che è segnato in qualche modo
- `<time>`  
per indicare il contenuto che è una data
- `<meter>`  
per indicare il contenuto che è una frazione di un intervallo noto

- `<progress>`

per indicare lo stato di avanzamento di una attività verso il completamento

#### 2.4.4 Nuove Dynamic Pages Support

HTML5 è stato sviluppato per aiutare gli sviluppatori di applicazioni Web, quindi ci sono molte nuove funzionalità per creare in modo più semplice pagine HTML dinamiche che sono:

- `context menus`

HTML5 favorisce la creazione e l'utilizzo di menu contestuali all'interno di pagine web e applicazioni

- `attributo async`

Questo si aggiunge al tag script per dire al browser che lo script deve essere caricato in modo asincrono in modo da non rallentare il caricamento e la visualizzazione del resto della pagina.

- `<details>`

fornisce dettagli su un elemento, sarebbe come tooltip delle applicazioni non web.

- `<datagrid>`

crea una tabella con dati provenienti da un database o da un'altra sorgente dinamica

- `<menu>`

consente di creare un sistema di menu sulle pagine web

- `<dommand>`

definisce le azioni che dovrebbe accadere quando si attiva un elemento dinamico

### 2.4.5 Nuovi form types

HTML5 introduce nuovi tipi di input per i moduli. Questi nuovi tipi di input hanno due vantaggi: il loro utilizzo significa meno tempo di sviluppo e una migliore esperienza utente. I nuovi tipi di input sono:

- `datetime`
- `datetime-local`
- `date`
- `month`
- `week`
- `time`
- `number`
- `range`
- `email`
- `url`
- `search`
- `tel`
- `color`

## 2.4.6 Nuovi elementi

Ci sono alcuni nuovi ed interessanti elementi in HTML5:

- `<canvas>`

un elemento per visualizzare uno spazio di disegno sulle pagine web. Può consentire di aggiungere immagini o grafici o semplicemente creare al volo grafici sulle pagine.

- `<video>`

consente di aggiungere video alle pagine web.

- `<audio>`

con questo semplice tag si potrà aggiungere l'audio alle pagine web.

## 2.4.7 Rimuove alcuni elementi

Ci sono anche alcuni elementi in HTML 4, che non saranno più supportati da HTML5. La maggior parte sono già deprecati e quindi non dovrebbe rappresentare una sorpresa la loro eliminazione dal nuovo linguaggio. I tag non più supportati sono:

- `acronym`

- `applet`

- `basefont`

- `big`

- `center`

- `dir`

- font
- frame
- frameset
- isindex
- noframes
- noscript
- s
- strike
- tt
- u



## Capitolo 3

# Il nuovo tag Canvas di HTML5

### 3.1 Il nuovo tag canvas

Il nuovo standard HTML5 introduce tra gli altri un nuovo tag `<canvas>` che permette di disegnare una “tela di bitmap” e di controllarne il contenuto utilizzando JavaScript (cioè si disegna sul `<canvas>` utilizzando JavaScript), la stessa tecnologia utilizzata dalle librerie RGraph per disegnare i grafici.

Una tela è un rettangolo nella pagina in cui è possibile utilizzare JavaScript per disegnare tutto quello che si vuole. Si potrebbe paragonare ad un pezzo di carta che fa parte della pagina web, su cui è possibile disegnare, grazie all'utilizzo di JavaScript, l'interazione con la tela è estremamente semplice.

Il tag `<canvas>` utilizza la metodologia di disegno fire and forget, non c'è DOM che viene comunque mantenuto, quindi se si vuole cambiare qualcosa probabilmente (ma non necessariamente) si deve ridisegnare l'intera tela. La mancanza di un DOM significa che la tela è veloce e molto reattiva questa caratteristica è importante quando si forniscono agli utenti grafici interattivi o animati.

Il tag canvas è stato originariamente introdotto da Apple nel 2004 per l'uso in Mac OS X WebKit per applicazioni dashboard all'interno del browser web Safari. Da allora è stato adottato da Mozilla ed Opera ed ora il W3C lo ha adottato nella specifica HTML5. Il nuovo tag è ora supportato da tutti i browser moderni, tra cui le versioni anche di Internet Explorer (a partire dalla versione 9). Se si utilizza Internet Explorer 7 e/o Internet Explorer 8 è necessaria la libreria ExCanvas che aggiunge il supporto per il tag che si traduce semplicemente nell'includere una libreria extra JavaScript nella pagina.

## 3.2 Utilizziamo canvas

Come abbiamo visto una delle caratteristiche più interessanti di HTML5 è l'elemento `<canvas>` che può essere utilizzato sia per disegnare la grafica vettoriale che per creare effetti sorprendenti o per produrre giochi interattivi e animazioni. In questo paragrafo vedremo un utilizzo del tag `<canvas>`.

Quando viene definito, questo tag specifica in proprio la larghezza e l'altezza dell'area di disegno ed un identificativo unico (id). L'attributo id in questo caso è anche un attributo di stile per aggiungere il bordo.

Il codice HTML è

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
```

Your browser does not support the HTML5 canvas tag.



```
</canvas>
</body>
</html>
```

graficamente il risultato è:

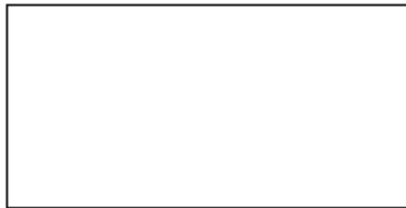


Figura 3.1: Area di disegno creta con il tag canvas

Il contenuto testuale tra i tag, “Your browser does not support the HTML5 canvas tag.” viene visualizzato se il browser non supporta il tag `<canvas>`.

Il disegno all’interno del tag `<canvas>` deve avvenire utilizzando JavaScript

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
Your browser does not support the HTML5 canvas tag.
</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,150,75);
</script>
```

```
</body>
</html>
```



Figura 3.2: Disegno di rettangolo rosso

inizialmente si recupera l'elemento `<canvas>` tramite l'id assegnato:

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

sull'oggetto ricavato, si invoca il suo metodo `getContext()`:

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

Il parametro (2D) del metodo `getContext` è previsto da HTML5, e consente di disegnare linee, rettangoli, cerchi, testi, immagini e molto altro.

le istruzioni che seguono definiscono il disegno all'intero dell'area delimitata dal rettangolo

```
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,150,75);
```

in particolare la proprietà `fillStyle` può essere un colore CSS o una sfumatura, il `fillStyle` predefinito è `#000000` (nero), mentre il metodo `fillRect(x, y, larghezza, altezza)` disegna un rettangolo riempito con l'attuale stile di riempimento.

La tela è una griglia bidimensionale, l'angolo superiore sinistro della tela ha coordinate (0,0), quindi, `fillRect()` che aveva come parametri (0,0,150,75), significa: inizia in alto a sinistra (0,0) e disegnare un rettangolo di 150x75 pixel.



Figura 3.3: Indicazione delle coordinate x ed y

Per disegnare linee rette su una tela, useremo i seguenti due metodi:

- `moveTo (x, y)` definisce il punto iniziale della linea
- `lineTo (x, y)` definisce il punto finale della linea

Per disegnare la linea in realtà, dobbiamo utilizzare uno dei metodi `stroke()`.

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
```

```

ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>
</body>
</html>

```



Figura 3.4: Disegno di una linea all'interno di un rettangolo

Per disegnare un cerchio su una tela, useremo il metodo seguente:

- `arc(x, y, r, start, stop)`

Per disegnare effettivamente il cerchio, dobbiamo usare uno dei metodi di `ink`, come la `stroke()` oppure `fill()`.

```

<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3"
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.beginPath();

```

```
ctx.arc(95,50,40,0,2*Math.PI);  
ctx.stroke();  
</script>  
</body>  
</html>
```

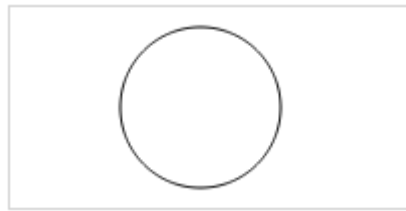


Figura 3.5: Disegno di un cerchio all'interno di un rettangolo

Per disegnare il testo su una tela, la proprietà ed i metodi più importanti sono:

- font - definisce le proprietà del font per il testo
- fillText (text, x, y) - disegna testo sulla tela con riempimento
- strokeText (text, x, y) - disegna il testo sulla tela senza riempimento

Utilizzando fillText():

```
<!DOCTYPE html>  
<html>  
<body>  
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">  
Your browser does not support the HTML5 canvas tag.</canvas>  
<script>  
var c=document.getElementById("myCanvas");
```

```

var ctx=c.getContext("2d");
ctx.font="30px Arial";
ctx.fillText("Hello World",10,50);
</script>
</body>
</html>

```

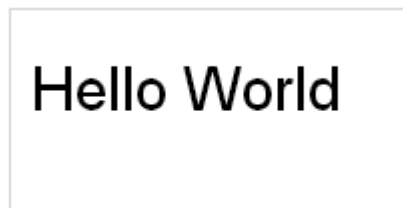


Figura 3.6: Disegno del testo all'interno di un rettangolo con il metodo `fillText()`

Utilizzando invece il metodo `strokeText()` abbiamo il testo senza riempimento

```

<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3; background-color:#d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.font="30px Arial";
ctx.strokeText("Hello World",10,50);
</script>

```

```
</body>  
</html>
```

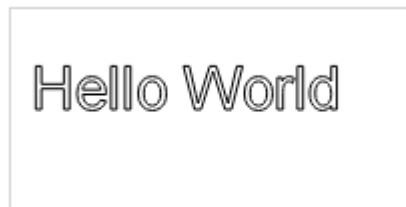


Figura 3.7: Disegno del testo all'interno di un rettangolo con il metodo `strokeText()`

Le sfumature possono essere utilizzate per riempire rettangoli, cerchi, linee, testo, ecc. Ci sono due diversi tipi di gradienti:

- `createLinearGradient (x, y, x1, y1)` crea un gradiente lineare
- `createRadialGradient (x, y, r, x1, y1, r1)` crea un gradiente circolare

Una volta creato un oggetto gradiente è necessario aggiungere due o più interruzioni di colore.

Il metodo `addColorStop()` specifica le interruzioni di colore, e la sua posizione lungo il gradiente. Posizioni gradiente può essere ovunque tra 0 e 1.

Per utilizzare il gradiente, impostare la proprietà `fillStyle` o `strokeStyle` al gradiente, e quindi disegnare la forma, come un rettangolo, un testo o una linea.

```
<!DOCTYPE html>  
<html>
```

```

<body>
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
// Create gradient
var grd=ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");
// Fill with gradient
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
</script>
</body>
</html>

```



Figura 3.8: Sfumatura creata utilizzando il metodo createLinearGradient()

Utilizzando invece un gradiente circolare abbiamo

```

<!DOCTYPE html>
<html>

```



```

<body>
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
// Create gradient
var grd=ctx.createRadialGradient(75,50,5,90,60,100);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");
// Fill with gradient
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
</script>
</body>
</html>

```



Figura 3.9: Sfumatura creata utilizzando il metodo `createRadialGradient()`

Per disegnare l'immagine su una tela, useremo il metodo seguente:

- `drawImage(immagine, x, y)`

```
<!DOCTYPE html>
<html>
<body>
<p>Image to use:</p>
<p>Canvas:</p>
<canvas id="myCanvas" width="250" height="300"
style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
var img=document.getElementById("scream");
ctx.drawImage(img,10,10);
</script>
</body>
</html>
```



Canvas:



Figura 3.10: Disegno immagine con drawImage()



# Capitolo 4

## Accessibilità

### 4.1 Cos'è l'accessibilità

Per accessibilità si intende che le persone con disabilità possono utilizzare il web senza alcuna barriera. Più precisamente, rendere un sito accessibile significa che le persone con disabilità possano percepire, comprendere, navigare e interagire con il Web. I benefici di un sito accessibili sono accolti anche da altri soggetti come gli anziani con poca abilità a causa dell'invecchiamento.

Accessibilità comprende tutte le disabilità che riguardano l'accesso al Web, tra cui quelle visive, uditive, fisiche, di parola, cognitive e neurologiche. Milioni di persone soffrono di disabilità, attualmente la maggior parte dei siti web hanno barriere di accessibilità che rendono difficile o impossibile la consultazione di molte persone con disabilità.

L'accessibilità dei siti web ha dei benefici anche per le persone che non hanno disabilità. Ad esempio, un principio chiave dell'accessibilità è progettare siti web che siano flessibili per soddisfare le diverse esigenze degli utenti. Questa flessibilità favorisce anche le persone senza disabilità, come ad esempio le persone che utilizzano una connessione Internet lenta, le persone con disabilità

temporanee, come un braccio rotto.

## 4.2 Perché l'accessibilità del web è importante

Il Web è una risorsa sempre più importante in molti aspetti della vita: l'educazione, l'occupazione, il governo, il commercio, la sanità. E' essenziale che il Web sia accessibile, al fine di garantire parità di accesso e pari opportunità a tutte le persone sia esse con disabilità e non. Un sito accessibile può anche aiutare le persone con disabilità a partecipare in modo più attivo alla società.

L'uso del web si sta diffondendo rapidamente in molte aree della società e della vita quotidiana. In molti paesi il Web è sempre più utilizzato per le informazioni governative, i servizi, l'istruzione, la formazione, il commercio, le notizie, la partecipazione alla vita sociale, la sanità, il tempo libero, l'intrattenimento e molto altro. In alcuni casi, il web sta sostituendo le risorse tradizionali e prestazioni di servizio.

Il Web è un mezzo importante per la ricerca di informazioni, nonché per fornire informazioni e interagire con la società. Questo diritto umano fondamentale è riconosciuto nella Convenzione delle Nazioni Unite sui diritti delle persone con disabilità, che menziona in particolare Internet e più in generale Information and Communication Technology (ICT).

Il web può rappresentare una opportunità, senza precedenti, per l'accesso alle informazioni da parte di persone con disabilità. Cioè, le barriere di accessibilità audio, video e di lettura, possono essere superate più facilmente attraverso le tecnologie web. Per esempio, se pensiamo che fino a qualche

anno fa l'unico modo per ottenere alcune informazioni era quello di andare in una biblioteca e leggere su carta, questo poteva rappresentare un notevole limite per molte persone con disabilità, in quanto dovevano recarsi sul posto, ottenere fisicamente la risorsa e leggere la risorsa.

Oggi la stessa informazioni può essere resa disponibile sul web in un formato accessibile rendendo così la sua fruizione alle persone molto più facile. Pertanto, le persone con disabilità possano avere in modo più efficace ed efficiente l'accesso alle informazioni attraverso siti web accessibili.

Il Web è un momento di incontro senza precedenti per le persone con disabilità, ad esempio, alcune disabilità possono limitare il tipo di lavoro che una persona può fare così che un web accessibile può contribuire ad aumentare le opportunità di lavoro.

### **4.3 Rendere il Web accessibile**

Gran parte dell'attenzione per la realizzazione di un sito web accessibile deve essere posta dagli sviluppatori. Tuttavia, il software Web ha anche un ruolo fondamentale nel campo dell'accessibilità Internet. Il software può aiutare gli sviluppatori a produrre e valutare i siti web accessibili e utilizzabili da persone con disabilità.

Uno dei ruoli della Web Accessibility Initiative ( WAI ) è quello di sviluppare le linee guida che descrivono le soluzioni di accessibilità per il software e per i siti web. Queste linee guida WAI sono considerati lo standard internazionale per l'accessibilità del web.

Ecco qui di seguito alcune linee guida di facile realizzazione per rendere le pagine web di un sito fruibili da persone con disabilità.

### **4.3.1 Testo alternativo per le immagini**

Tecnologie assistive come lettori di schermo (per gli utenti non vedenti o ipovedenti) convertono gli elementi testuali di una pagina web in un audio elettronico. Quando si incontra un'immagine, lo screen reader legge il contenuto dell'attributo alt. Il testo dell'alt dovrebbe descrivere in formato testuale il significato dell'immagine. Tutte le immagini devono pertanto avere un testo alternativo, tale testo dovrebbe essere il più sintetico possibile. Il testo alternato invece per le immagini puramente decorative dovrebbe essere vuoto `<alt="">` consentendo così agli screen reader di saltare l'immagine durante la lettura.

Evitare l'uso di immagini di testo in quanto i software di lettura non possono interpretare queste immagini, a meno che non venga fornito il testo alternativo, mentre gli utenti ipovedenti possono voler ingrandire il testo, cambiando il colore del testo, ma se si tratta di una immagine non possono manipolare il testo per soddisfare le loro esigenze.

### **4.3.2 Fornire contenuti organizzati e strutturati**

Un contenuto ben organizzato e strutturato è chiaramente un elemento critico di qualsiasi sito web, se così fosse, questo consentirebbe a tutti gli utenti di trovare le informazioni di cui hanno bisogno in poco tempo. Tutte le pagine dovrebbero avere un titolo e dei sotto-titoli visivamente distinguibili dal testo principale. Titoli e sotto-titoli dovrebbero essere chiaramente definiti utilizzando gli appositi tag html (h1, h2, h3, ecc), in questo modo i lettori



di schermo si affidano a questi identificatori per migliorare la scansione della pagina attraverso le varie sezioni.

### **4.3.3 Fornire alternative testuali per contenuti non testuali**

Molte tecnologie si basano su testo elettronico per fornire informazioni agli utenti non udenti. Se sul sito sono presenti audio e / o video allora è necessario fornire un testo come sostituto o se vogliamo il testo deve accompagnare questo tipo di contenuto multimediale. L'obiettivo è quello di fornire agli utenti informazioni basate su testo, che è equivalente a quello che è presentato in qualsiasi componente audio / video del sito. Le didascalie devono essere forniti per tutti i contenuti audio pre-registrati.

### **4.3.4 Associare etichette a tutti i campi del modulo**

Quando uno screen reader incontrano un campo di input, cerca l'etichetta corrispondente che deve descrivere cosa l'utente deve inserire nella casella. Se l'etichetta non è presente, l'utente non può capire quale informazione deve essere inserita. E' necessario quindi utilizzare l'attributo for con il tag label per creare la giusta associazioni fra etichetta testuali ed i campi corrispondenti.

Un esempio di codice è:

```
<label for="name"> Nome </ label> <input type="text" id="name" name="name" />
```

### **4.3.5 Consentire agli utenti di attivare / disattivare i contenuti audio e video**

La presenza di audio e video su un sito web può essere fonte di distrazione per molti utenti. Se il sito comprende contenuti visivi (come filmati in flash)

o contenuti audio che si avviano automaticamente e che durano più di cinque secondi, gli utenti devono avere la possibilità di mettere in pausa o disattivare questi contenuti.

#### **4.3.6 Ottimizzare la leggibilità del testo**

E' indispensabile che gli utenti possono facilmente distinguere il testo dallo sfondo. Assicurarsi quindi di aver scelto una combinazione di colori background / foreground che ha un elevato rapporto di contrasto.

#### **4.3.7 Non usare il colore come unico indicatore di una possibile azione**

Gli utenti non vedenti non possono basarsi su indicatori di colore, quindi è necessario che per indicare una azione dovrebbe essere utilizzato un indicatore secondario (come il testo sottolineato) oltre ad avere un colore diverso dal testo principale.

#### **4.3.8 Consentire diversi modi di navigazione**

Molti utenti con disabilità hanno accesso ad un solo metodo di input, il più delle volte è rappresentato da una tastiera. La navigazione di un sito web dovrebbe essere indipendente dal dispositivo, il navigatore non deve essere costretto ad utilizzare uno specifico dispositivo.

Un esempio è rappresentato dal menu di navigazione con sottomenu a popup che si apre quando il mouse passa sopra il titolo. Gli utenti che non possono utilizzare il mouse non saranno in grado di accedere alle pagine dei sottomenu.

### **4.3.9 Prestare attenzione durante lo sviluppo di contenuti dinamici**

Uso di JavaScript introduce diversi problemi di accessibilità. Il contenuto dinamico (tramite JavaScript, Flash, AJAX) non sempre possono essere rilevati dalle tecnologie assistive. Alcune tecnologie assistive accolgono il segnale di cambiamento, mentre altri possono solo parzialmente segnalare le modifiche o non riescono a segnalare il cambiamento del contenuto. Tecnologie che sono in grado di identificare la modifica dei contenuti possono avere difficoltà a determinare quali di questi cambiamenti devono essere segnalati all'utente. Problemi di accessibilità dovuti a JavaScript correlati possono essere evitati attraverso l'uso attento di soluzioni HTML e DOM-compliant o CSS



# Capitolo 5

## Conclusioni

Nel nostro tempo, in cui il confine tra desktop e mobile sta rapidamente svanendo, gli utenti si aspettano di ottenere le informazioni in qualsiasi luogo, in qualsiasi momento e vogliono che siano veloci. Soddisfare tutte queste aspettative, fornendo agli utenti quello che desiderano è l'unico modo per mantenere i clienti, focalizzarli sul nostro prodotto, servizio o sito web.

HTML5 come abbiamo visto è l'ultima versione del famoso linguaggio HTML, nato per affrontare le esigenze e le aspettative dei moderni siti web. Si occupa di cose come il markup semantico, che fornisce informazioni sul contenuto che descrive, è un nuovo standard per tutti gli sviluppatori web ed anche per i fornitori di browser.

L'utilizzo dell'HTML5 riduce i tempi di sviluppo e pone una particolare attenzione sugli ultimi browser. Infatti tutti i browser moderni supportano HTML5 e questo permette agli sviluppatori di utilizzare il nuovo linguaggio con tutte le nuove e interessanti funzioni che offre, consentendo così la creazione di risultati finali più dinamici.

Grazie all'utilizzo combinato di CSS3 e HTML5 oggi è possibile realizzare cose che fino a poco tempo fa non erano possibile creare soltanto con l'utilizzo di HTML ma che richiedevano plugin esterni come Adobe Flash o Microsoft Silverlight.

HTML5 è nato per lasciare una traccia. Ha il supporto della comunità dei produttori di browser ed è un ottimo modo per pubblicare i contenuti rendendoli vicino all'utente. Lavora con le più moderne tecnologie in un mercato mobile in rapida crescita.

I nuovi linguaggi come HTML5 e CSS3, se pur da un lato offrono agli sviluppatori incredibili possibilità per lo sviluppo di siti web ed applicazioni moderne, dall'altro lato creano dei problemi quando si vuole gestire la compatibilità cross-browser, infatti alcune funzionalità di ultima generazione offerte da HTML5 e CSS3 non sono ancora supportate alla stessa maniera da tutti i browser.

# Ringraziamenti

Al termine di questo lavoro di tesi, desidero ringraziare il prof.re Davide Rossi, relatore di questo lavoro di tesi, per la grande disponibilità e cortesia dimostratemi, e per tutto l'aiuto fornito durante la stesura.

Un sentito ringraziamento ai miei genitori, che, con il loro fondamentale sostegno economico e morale, mi hanno permesso di completare il percorso di studi e di raggiungere questo traguardo.

Desidero inoltre ringraziare i compagni di studi, per essermi stati vicini sia nei momenti difficili, sia nei momenti felici: sono stati per me più veri amici che semplici compagni





# Bibliografia

*HTML5 CSS3 JavaScript*, P. Principe, Apogeo, 2012

*Programming the World Wide Web*, R. W. Sebesta, Addison-Wesley Professional, 2012

*HTML5 Canvas for Dummies*, Don Cowan, 2012

*Internet and World Wide Web How to Program*, P. J. Deitel, Prentice Hall, 2011

*Tim Berners-Lee: Inventor of the World Wide Web*, S. Sammartino, Twenty-First Century Books, 2009

*HTML5. Canvas. Guida pocket*, D. Flanagan, Tecniche Nuove, 2011

*La storia del futuro. Le origini di Internet*, K. Hafner, M. Lyon, Feltrinelli, 1998

*HTML5 e CSS3. Per il World Wide Web*, E. Castro, Bruce Hyslop, Tecniche Nuove, 2012

*JavaScript per il World Wide Web*, T. Negrino, D. Smith, Pearson, 2000

*HTML5 e CSS 3*, G. Gigliotti, Apogeo, 2011

*HTML5 & CSS 3. Guida all'uso*, J. Hill, J.A. Brannan, Mondadori Informatica, 2012

*Storia di internet e il futuro digitale*, J. Ryan, Einaudi, 2011

*HTML5* B.Lawson, R.Sharp, Mondadori Informatica, 2011

*Accessibilità. Guida completa*, M. Diodati, Apogeo, 2007

*HTML5. Guida pratica*, F. Wempen, Mondadori Informatica, 2012

*I piani per l'accessibilità. Una sfida per promuovere l'autonomia dei cittadini e valorizzare i luoghi dell'abitare*, A. Lauria, Gangemi, 2012

*Sviluppare applicazioni con HTML5*, Z. Kessin, 2012, *Tecniche Nuove Web usability 2.0. L'usabilità che conta. Vol. 1*, J. Nielsen, H. Loranger, Apogeo, 2010

*Storia dell'informatica. Dai primi computer digitali all'era di internet*, P. E. Ceruzzi, Apogeo, 2005

*Web design. Bulletproof. Creare siti web standard e accessibili, basati su layout CSS liquidi, flessibili e che funzionano in tutti i browser.*, D. Cede-

rholm, L. Mondini L., Pearson, 2008

*HTML, XHTML e CSS per il World Wide Web*, E. Castro, Tecniche Nuove, 2007



# Siti Web

*<http://www.w3schools.com>*

*<http://www.findmebyip.com>*

*<http://en.wikipedia.org>*