

Applicazioni web

Parte 2 DHTML - CSS

Alberto Ferrari

Sommario

- DHTML
 - Elementi di raggruppamento
 - Markup semantico
- CSS
 - Sintassi
 - Proprietà

Alberto Ferrari



DHTML

Dynamic HTML

- Dynamic Html non è uno standard definito dal World Wide Web Consortium (W3C) – il consorzio no-profit che definisce alcuni standard, tra cui l'html
 - Termine di marketing usato da Netscape e Microsoft per descrivere le tecnologie supportate dalle versioni 4.x dei loro browser
- Dynamic Html è una combinazione di tecnologie per creare pagine web dinamiche
- Secondo il W3C:
 - Dynamic HTML è un termine usato da alcune aziende per descrivere una combinazione di html, fogli di stile e script che permette di ottenere documenti animati

Alberto Ferrari



Markup semantico

Linee guida generali

- Definire un ricco vocabolario di classi semantiche
 - ... la cui presentazione può essere specificata in fogli di stile validi per tutto il sito web
 - Permette di scrivere documenti complessi senza introdurre elementi di presentazione
 - Prende il posto dei vecchi template di presentazione
 - La presentazione è ora definita nei fogli di stile, permettendo al markup di rimanere semantico
- Riferimenti per il markup di mozilla.org
 - <http://www.mozilla.org/contribute/writing/markup>
- Tecniche e linee guida per l'accessibilità dei contenuti web
 - <http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Alberto Ferrari



Mozilla Foundation

- Mozilla Foundation (Fondazione Mozilla) è una organizzazione non-profit fondata per supportare e fornire l'organizzazione e la direzione del progetto open source Mozilla.
- L'organizzazione coordina il rilascio delle versioni dei software Mozilla, stabilisce le linee guida alla base del processo di sviluppo, mantiene l'infrastruttura base necessaria e dà lavoro a diversi programmatori pagati.

Wikipedia

Alberto Ferrari

Contenuto

Markup semantico

Alberto Ferrari

Sezioni

- Usare `<p>` per marcare i paragrafi
 - Usare i tag per racchiudere i paragrafi, non per separarli
- Usare `<div class="section">` per marcare le sezioni all'interno di un documento
 - Racchiudere sia il titolo che il contenuto della sezione
 - Le sezioni sono tipicamente un po' rientrate rispetto al testo vicino

```
<div class="section">
  <h2 id="structure">General Structure</h2>
  <p>...</p>
</div>
```

Alberto Ferrari

Citazioni

- Usare `<blockquote>` con `<address>`
 - Paragrafi o altri contenuti di livello blocco
 - Si può indicare una attribuzione con `<address>` proprio prima del tag di chiusura
 - Epigrafi: "epigraph" come classe

```
<blockquote cite="http://www.faqs.org/rfcs/rfc1866.html">
  <p>The BLOCKQUOTE element contains text quoted from another
  source.</p>
  <p>A typical rendering might be a slight extra left and right
  indent, and/or italic font. The BLOCKQUOTE typically provides
  space above and below the quote.</p>
  <address><a href="http://www.faqs.org/rfcs/rfc1866.html">HTML
  2.0</a></address>
</blockquote>
```

Alberto Ferrari

Linee guida mozilla.org

Note di vario tipo

- Commenti dell'autore o dell'editor (in una bozza)
 - `<div class="remark">`
 - ``
- Note generiche
 - ``
 - `<p class="note">`
 - `<div class="note">`
- Note chiave
 - `<div class="key-point">`
- Avvisi importanti (cioè da leggere)
 - `<* class="important">`

Alberto Ferrari

Linee guida mozilla.org

Figure ed esempi

- Figure
 - ```
<div class="figure">

 <div class="legend">Caption</div>
</div>
```
- Esempi
  - `<div class="example">`
  - `<div class="good example">`
  - `<div class="bad example">`


Alberto Ferrari



## Linee guida mozilla.org Computer

- Codice inline: `<code>`
- Nomi di file e url: `<code class="filename">`
- Blocchi di codice: `<pre><code>`
- Input da tastiera: `<kbd>`
- Variabili e costanti: `<var>`
- Output inline: `<samp>`
- Catture di schermate testuali (blocchi di output)  
`<pre><samp>`

Alberto Ferrari



## Linee guida mozilla.org Navigazione

- Indice (table of contents):  
`<ol class="toc">` e `<ul class="toc">`
- Navigazione locale (short o sectional):  
`<ul class="snav">`
- Breadcrumb (navigazione gerarchica) :  
`<p class="crumbs">`
- Riferimenti esterni al sito:  
`<a class="ex-ref">`

Alberto Ferrari

# Forma (stile)

## CSS

Alberto Ferrari

## Cascading Style Sheet

- I fogli di stile a cascata (detti anche semplicemente fogli di stile) vengono usati per definire la **rappresentazione** di documenti HTML, XHTML e XML.
- Le regole per comporre i fogli di stile sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C.
- L'introduzione dei fogli di stile si è resa necessaria per **separare** i contenuti dalla formattazione.

*Wikipedia*

Alberto Ferrari



## Css

# Cascading Style Sheets

- Gli stili definiscono come visualizzare gli elementi html
- Gli stili sono di solito raggruppati in fogli di stile
- Gli stili sono stati aggiunti ad html (v4.0) per risolvere un problema
- I fogli di stile esterni fanno risparmiare lavoro
- I fogli di stile esterni sono memorizzati in file css
- Definizioni di stile multiple sono trattate a cascata in un'unica definizione

Alberto Ferrari



## Css

# Gli stili risolvono un problema

- I tag html erano stati in origine studiati per definire il contenuto di un documento html
- Il layout del documento doveva essere preso in carica dal browser, senza usare tag di formattazione

Alberto Ferrari





## Css

### Gli stili risolvono un problema

- I due browser principali – Netscape e Internet Explorer – hanno continuato ad aggiungere nuovi tag e attributi (come il tag `<font>` e l'attributo `color`) alle specifiche html originali
  - Sempre più difficile creare siti web dove il contenuto fosse separato dal layout di presentazione
- Per risolvere questo problema, il W3C ha creato gli **stili** in aggiunta ad Html 4.0
  - Gli stili in Html 4.0 definiscono la visualizzazione degli elementi, sostituendo `<font>` e `color` di Html 3.2
  - Sia Netscape 4.0 che Internet Explorer 4.0 supportano i Cascading Style Sheets

Alberto Ferrari



## Css

### Gli stili fanno risparmiare lavoro

- Css segna una svolta nei progetti web perché permette agli sviluppatori di controllare in un sol colpo lo stile e la disposizione di molte pagine web
  - Stili normalmente in un file esterno alle pagine html
  - Uno sviluppatore web può definire uno stile per ciascun elemento html ed applicarlo a tutte le pagine che vuole
- Modifica globale - es. modificare il font o il colore di tutti i titoli in tutte le pagine di un sito
  - Cambiando lo stile tutti gli elementi nel sito si aggiornano automaticamente

Alberto Ferrari



# Css

## Come inserire un foglio di stile

- Fogli di stile esterni
- Fogli di stile interni
- Stile inline

Alberto Ferrari



# Css

## Fogli di stile esterni

- Foglio esterno: ideale se l'applicazione ha molte pagine
  - Si può cambiare l'aspetto di un intero sito modificando un solo file
- Ogni pagina deve essere collegata al foglio di stile
  - Tag `<link>` all'interno della sezione `head`
  - ```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css" />  
</head>
```
 - Il browser leggerà le definizioni di stile dal file `mystyle.css`, e formatterà il documento di conseguenza
- Un foglio di stile esterno può essere scritto in qualsiasi editor di testo
 - Il file non deve contenere alcun tag html
 - Il foglio di stile deve essere salvato con estensione `.css`
- Un esempio di foglio di stile:
 - ```
body {background-image: url("images/back40.gif")}
p {margin-left: 20px}
```

Alberto Ferrari

## Css

### Fogli di stile interni

- Un foglio di stile interno può essere usato quando un documento html ha uno stile unico
- Un foglio di stile interno si definisce nella sezione *head* usando il tag `<style>`
  - `<head>`

```

<style type="text/css">
 hr {color: sienna}
 p {margin-left: 20px}
 body {background-image: url("images/back40.gif")}
</style>
</head>

```
- Il browser leggerà le definizioni di stile, e formatterà il documento in accordo ad esse

[Alberto Ferrari](#)

## Css

### Stile inline

- Uno stile inline fa perdere molti dei vantaggi dei fogli di stile, perché mischia il contenuto con la presentazione
  - Metodo da usare con parsimonia, quando uno stile deve essere applicato ad una sola occorrenza di un elemento in un documento
- Per definire uno stile inline si usa l'attributo *style* dell'elemento da formattare
  - L'attributo *style* può contenere qualsiasi proprietà css
  - L'esempio mostra come cambiare il colore e il margine sinistro di un singolo paragrafo
  - `<p style="color: sienna; margin-left: 20px">`  
     This is a paragraph  
   `</p>`

[Alberto Ferrari](#)



## Css

### Ordine di priorità (cascata)

- I fogli di stile permettono di specificare informazioni di stile in vario modo
  - Gli stili possono essere specificati in un singolo elemento html, nella sezione *<head>* di una pagina html, o in un file css esterno
  - È possibile fare riferimento pure a diversi fogli esterni in una stessa pagina html
  - Quale stile sarà usato quando è specificato più di uno stile per un elemento html?
- Tutti gli stili vengono raccolti *a cascata* in un nuovo foglio di stile *virtuale* secondo le seguenti regole
  - Default del browser (priorità più bassa)
  - Foglio di stile esterno
  - Foglio di stile interno (nella sezione *<head>*)
  - Stile inline (nell'elemento html – priorità più alta)

Alberto Ferrari



## Sintassi Css

### Regole

- La più semplice regola css è composta di tre parti: un *selettore*, una *proprietà* ed un *valore*:
  - `selector {property: value}`
  - Il **selettore** può essere l'elemento html che si vuole definire
  - La **proprietà** è l'attributo che si vuole cambiare
  - Alla proprietà è associato un **valore**
  - La proprietà e il valore sono separati da due-punti e racchiusi tra parentesi graffe
  - `body {color: black}`

Alberto Ferrari



## Sintassi Css

### Regole

- Se il valore è composto da più parole, bisogna racchiuderlo tra virgolette doppie
- Se si vuole specificare più di una proprietà, bisogna separarle con un punto-e-virgola
- L'esempio mostra come definire un paragrafo con allineamento centrato e colore del testo rosso

```
□ p {
 text-align: center;
 color: red;
 font-family: "sans serif";
}
```

Alberto Ferrari



## Sintassi Css

### Raggruppare i selettori

- Si possono raggruppare i selettori: bisogna separare i selettori con una virgola
- Nell'esempio sotto, gli elementi di titolo sono raggruppati; ogni elemento di titolo sarà verde

```
□ h1, h2, h3, h4, h5, h6 {
 color: green;
}
```

Alberto Ferrari



## Sintassi Css

### Selettore di classe

- Con il selettore di classe si possono definire stili diversi per uno stesso tipo di elemento html
  - Es. due tipi di paragrafo: uno allineato a destra e uno allineato al centro
  - ```
p.signature {text-align: right}
p.important {text-align: center}
```
 - Bisogna impostare l'attributo *class* nel doc html
 - ```
<p class="signature">
 This paragraph will be right-aligned.
</p>
<p class="important">
 This paragraph will be center-aligned.
</p>
```

Alberto Ferrari



## Sintassi Css

### Selettore di classe

- Si può specificare *un solo attributo class* per ciascun elemento html
- Ma questo attributo può contenere una *lista di classi*, separate da spazi
  - ```
<div class="side-note new">
  This is a side-note about the main article.
  It's new for the current user.
</div>
```

Alberto Ferrari

Sintassi Css

Selettore di classe

- Si può anche omettere il nome di tag nel selettore per definire uno stile da applicare a tutti gli elementi html di una certa classe
 - Allineare al centro tutti gli elementi di classe *important*
 - `.important {text-align: center}`
 - Sia l'elemento *h1* che quello *p* sono di classe *important*
 - Entrambi gli elementi seguiranno le regole del selettore *.important*
 - ```
<h1 class="important">
 This heading will be center-aligned
</h1>
<p class="important">
 This paragraph will also be center-aligned.
</p>
```

Alberto Ferrari

## Sintassi Css

### Selettore di id

- Il selettore *id* è diverso dal selettore *class*
  - Mentre un selettore *class* può applicarsi a **molte** elementi su una pagina, il selettore *id* si applica ad **un solo** elemento
  - Un attributo *id* deve essere unico all'interno del documento
- La regola di stile qui sotto sarà applicata ad un elemento *p* che ha il valore di *id* posto a *para1*
  - ```
p#para1 {
  text-align: center;
  color: red;
}
```
- La regola di stile qui sotto sarà applicata all' elemento avente come *id* il valore *wer345*
 - ```
#wer345 {color: green}
```
  - ```
<h1 id="wer345">Some text</h1>
```

Alberto Ferrari

Sintassi Css

Commenti

- Per spiegare le regole, si possono inserire nel css dei commenti, che possono essere d'aiuto quando il codice sorgente deve essere in seguito modificato
- I commenti vengono ignorati dal browser
- Un commento comincia con `/*` e termina con `*/`

```

□ /* This is a comment */
p {
  text-align: center;
  /* This is another comment */
  color: black;
  font-family: arial;
}

```

Alberto Ferrari

Proprietà Css

Sfondo

- **background-color** – Imposta il colore di sfondo di un elemento – *rgb(r,g,b), color-hex, color-name, transparent*
- **background-image** – Imposta una immagine come sfondo – *url(...), none*
- **background-repeat** – Imposta la ripetizione di una immagine di sfondo – *repeat, repeat-x, repeat-y, no-repeat*
- **background-attachment** – Immagine di sfondo fissa o che scorre con il testo – *scroll, fixed*
- **background-position** – Posizione di partenza di una immagine di sfondo – *top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right, x-% y-%, x-pos y-pos*
- **background** – Imposta tutte assieme le proprietà dello sfondo
- Css1 definisce 16 nomi di colori (palette VGA di Windows)
 - *aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow*

Alberto Ferrari

Proprietà Css Testo

- **color** – Colore del testo – *color*
- **direction** – Direzione del testo – *ltr, rtl*
- **text-align** – Allineamento – *left, right, center, justify*
- **text-decoration** – Decorazione – *none, underline, overline, line-through, blink*
- **text-indent** – Indentazione della prima riga – *length, %*
- **text-shadow** – Ombra – *none, color length length length*
- **text-transform** – Controlla le lettere – *none, capitalize, uppercase, lowercase*
- **white-space** – Trattamento degli spazi bianchi – *normal, pre, nowrap*

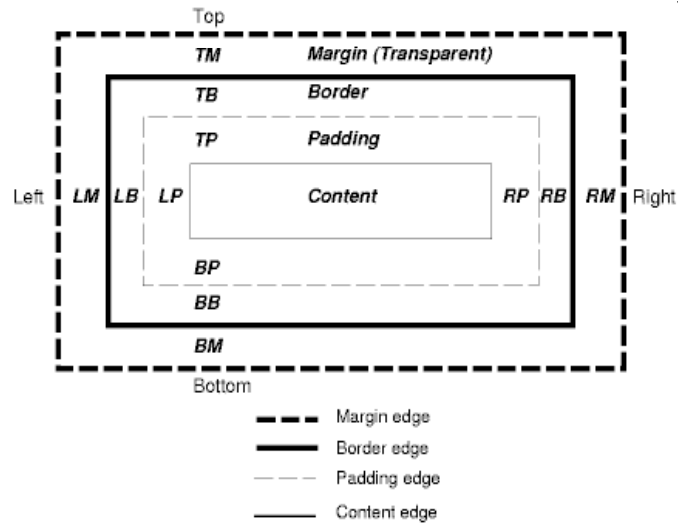
Alberto Ferrari

Proprietà Css Font

- **font-style** – Stile, corsivo – *normal, italic, oblique (elaborazione)*
- **font-variant** – Maiuscoletto o normale – *normal, small-caps*
- **font-weight** – Peso, grassetto – *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*
- **font-size** – Dimensione – *xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %*
- **font-family** – Lista con priorità di nomi di famiglie di font e/o nomi di famiglie generiche – *family-name, generic-family (serif, sans-serif, cursive, fantasy, monospace)*
- **font** – Imposta tutte le proprietà del font
- Dimensioni
 - px: misura in pixel (fissa rispetto a preferenze utente e risoluzione monitor)
 - pt: punti tipografici, 1/72 pollice (fissa rispetto a preferenze utente)
 - em: rispetto alla dimensione di font attuale per l'elemento
 - %: rispetto alla dimensione di font dell'elemento genitore

Alberto Ferrari

Proprietà Css Box model



Proprietà Css Bordi

- ***border-width*** – Ampiezza dei quattro bordi – 1-4 valori ()
- ***border-style*** – Quattro bordi – 1-4 valori – *none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset*
- ***border-color*** – Colore – 1-4 valori – *color*
- ***border*** – Imposta tutte le proprietà (uguali) per i quattro bordi
 - `.important { border: 2px dashed red; }`
- ***border-top, border-right, ...*** – Imposta tutte le proprietà per il bordo superiore, destro, ...
 - `blockquote { border-left: 2px solid black; }`
- I bordi *top, right, bottom, left* possono essere cambiati indipendentemente usando proprietà separate: es. *border-top-width*

Proprietà Css

Margini esterni e interni

- **margin** – Spazio attorno agli elementi – 1-4 valori
 - È possibile usare valori negativi per sovrapporre il contenuto
 - I margini *top*, *right*, *bottom*, *left* possono essere cambiati indipendentemente usando proprietà separate, es. *margin-top*
 - Possibile usare scorciatoia *margin* per impostare tutti i margini
- **padding** – Spazio tra il bordo e il contenuto degli elementi – 1-4 valori
 - Non sono ammessi valori negativi
 - I padding *top*, *right*, *bottom*, *left* possono essere cambiati indipendentemente usando proprietà separate, es. *padding-top*
 - Possibile usare la scorciatoia *padding* per impostare tutti i padding

Alberto Ferrari

Proprietà Css

Dimensioni

- **width** – Ampiezza – *auto*, *length*, %
- **height** – Altezza – *auto*, *length*, %
- IE5 bug (?)
 - Se si vuole un box largo 100 pixel con 10 pixel di padding e 10 pixel di bordo, bisogna impostare una regola così:
 - `width: 100px; padding: 10px; border: 10px`
 - Per ottenere lo stesso effetto in Internet Explorer 5.x, si dovrebbe alterare la regola di stile a questo modo, per evitare di ottenere un box di 40 pixel più piccolo
 - `width: 140px; padding: 10px; border: 10px`
- IE6 è finalmente conforme alle specifiche Css sulle dim del box
 - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">`
- **box-sizing** – Proprietà Css3 per specificare se le dimensioni del box comprendono bordi e margini o no – *border-box*, *content-box*

Alberto Ferrari



Proprietà Css Posizione

- **bottom** – Distanza del limite inferiore di un elemento rispetto all'elemento che lo contiene – *auto, %, length*
- **left** – Sinistra ... – *auto, %, length*
- **right** – Destra ... – *auto, %, length*
- **top** – Superiore – *auto, %, length*
- **vertical-align** – Allineamento verticale – *top, middle, bottom, baseline, sub, super, text-top, text-bottom, length, %*
- **position** – Modalità di posizionamento – *static, relative, absolute, fixed, inherit*
- **float** – Elementi posizionati a lato, circondati da testo – *left, right, none, inherit*

[Alberto Ferrari](#)



Proprietà Css Visibilità

- **z-index** – Ordine nello stack (rilievo, valori più alti sono in primo piano) – *auto, number*
- **overflow** – Impostazione per contenuto più ampio dell'area disponibile – *visible, hidden, scroll, auto*
- **visibility** – Visibilità – *visible, hidden (occupa spazio)*
- **display** – Visualizzazione – *none (non occupa spazio), block, inline, ...*

Alberto Ferrari