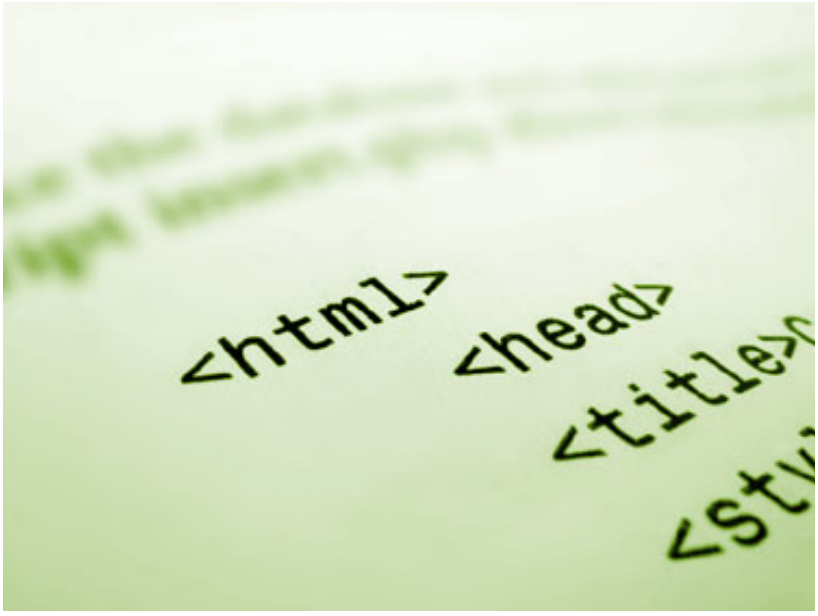

CSS

Corso di Web Design
Fabio Pittarello, Università Ca' Foscari Venezia - DAIS pitt@unive.it

CSS

- I fogli di stile (CSS o Cascading Style Sheets) permettono di applicare regole di presentazione agli elementi che caratterizzano una pagina XHTML.
- La parola cascading è riferita al fatto che l'informazione di stile può essere definita a più livelli, generando alla fine un unico foglio di stile virtuale nella quale si compongono tutte le regole provenienti da origine diverse e con specificità differenziate.

Vantaggi e svantaggi

- Vantaggi
  - Separazione di stile e struttura
  - Maggiore controllo sui caratteri e sul layout della pagina
  - Produzione di documenti potenzialmente più piccoli
  - Manutenzione più facile del sito
  - Facilità di apprendimento
- Svantaggi
  - Nel passato il supporto dei browser
    - CSS non supportati in browsers Explorer anteriori alla versione 3 e in Netscape anteriori alla 4
    - CSS supportati male in Netscape 4
    - Explorer 5.5 e Netscape 6 supportano quasi tutta la specifica CSS1 e parte della specifica CSS2.

L'evoluzione dei fogli di stile

- Prima specifica CSS1 nel 1996
- Specifica CSS2 nel 1998
  - Include proprietà addizionali e metodi avanzati per il posizionamento degli elementi, in grado di sostituire completamente tabelle e frames
- CSS 2.1 Giugno 2011
- CSS 3 (selettori e colori) hanno raggiunto lo stato finale (W3C Recommendation) Giugno 2011
- Altre parti di CSS sono ancora in corso di sviluppo e si trovano nello stato di Candidate Recommendation o di Working Draft
- Maggiori informazioni in http://www.w3.org/style/CSS

Regole di sintassi

- I fogli di stile consistono in una o più regole per descrivere come dovrebbe essere mostrato un elemento di una pagina
- Sintassi di una regola
  selettore {proprietà: valore}
- Le componenti principali sono il selettore, che identifica l'elemento che verrà condizionato dalla regola, e la dichiarazione, che specifica lo stile da applicare all'elemento.
- La dichiarazione, racchiusa tra parentesi grafe è composta da una proprietà e un valore, separate dal simbolo : seguito da uno spazio.
- Una dichiarazione può contenere una o più coppie proprietà/ valore, separate dal carattere ;
  - È opportuno utilizzare il carattere ; anche dopo l'ultima coppia, come rimedio per un bug che affligge i browsers più vecchi
- I valori sono dipendenti dalle proprietà
  - In alcuni casi sono misure di lunghezza, nomi di colori, oppure valori da una lista predefinita

Selettori CSS1 e 2

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div,p	Selects all elements and all <p> elements	1
element element	div p	Selects all <p> elements inside elements	1
element>element	div>p	Selects all <p> elements where the parent is a element	2
element+element	div+p	Selects all <p> elements that are placed immediately after elements	2
[attribute]	[target]	Selects all elements with a target attribute	2
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"	2
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"	2
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"	2
:link	a:link	Selects all unvisited links	1
:visited	a:visited	Selects all visited links	1
:active	a:active	Selects the active link	1
:hover	a:hover	Selects links on mouse over	1
:focus	input:focus	Selects the input element which has focus	2
:first-letter	p:first-letter	Selects the first letter of every <p> element	1
:first-line	p:first-line	Selects the first line of every <p> element	1
:first-child	p:first-child	Selects every <p> element that is the first child of its parent	2
:before	p:before	Insert content before the content of every <p> element	2
:after	p:after	Insert content after every <p> element	2
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"	2

Selettori CSS3

Selector	Example	Example description	CSS
element1~element2	p~ul	Selects every element that are preceded by a <p> element	3
[attribute^=value]	a[src^="https"]	Selects every <a> element whose src attribute value begins with "https"	3
[attribute$=value]	a[src$=".pdf"]	Selects every <a> element whose src attribute value ends with ".pdf"	3
[attribute=value]*	a[src*="w3schools"]	Selects every <a> element whose src attribute value contains the substring "w3schools"	3
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent	3
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent	3
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent	3
:only-child	p:only-child	Selects every <p> element that is the only child of its parent	3
:nth-child(*n*)	p:nth-child(2)	Selects every <p> element that is the second child of its parent	3
:nth-last-child(*n*)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child	3
:nth-of-type(*n*)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent	3
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child	3
:last-child	p:last-child	Selects every <p> element that is the last child of its parent	3
:root	:root	Selects the document's root element	3
:empty	p:empty	Selects every <p> element that has no children (including text nodes)	3
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)	3
:enabled	input:enabled	Selects every enabled <input> element	3
:disabled	input:disabled	Selects every disabled <input> element	3
:checked	input:checked	Selects every checked <input> element	3
:not(*selector*)	:not(p)	Selects every element that is not a <p> element	3
::selection	::selection	Selects the portion of an element that is selected by a user	3

Selettori di tipo e contestuali

- I Selettori sono la parte della regola che identifica l'elemento a cui lo stile viene applicato.
  - Selettori di tipo
    - L'elemento XHTML viene indicato con il suo nome
      - Esempio h1 {color: blue}
    - I Selettori di tipo possono essere raggruppati in liste separate da virgole
      - Esempio h1, h2, p {color: blue}
  - Selettori contestuali
    - E' possibile specificare attributi di stile per gli elementi HTML basati sul contesto nel quale appaiono
      - Esempio li em {color: green}
    - I selettori contestuali (più specifici) hanno la precedenza sul selettore semplice
    - Come i selettori semplici, i selettori contestuali possono essere raggruppati in liste separate da virgole
      - Esempio h1 b, h2 b, h3 b {color: red}

Ereditarietà

- Le maggior parte delle proprietà di stile vengono trasmesse o ereditate dai livelli più alti nella gerarchia dei tag XHTML a quelli più bassi.
  - Es.1 se si stabilisce il colore del testo per una lista <ul>, il colore verrà ereditato da tutti gli elementi <li> di quella lista.
  - Es. 2 se si specifica il colore del testo nel tag <body>, tutti gli elementi del documento verranno visualizzati rossi.
- Tuttavia, gli stili applicati a elementi specifici del documento hanno la meglio sugli stili applicati a livello più alto.
- Es. se specifico il colore del testo associandolo al tag body, ma poi specifico il colore associandolo a ul, ha la prevalenza l'ultima specifica
- Ci sono tuttavia delle eccezioni, per le quali le proprietà di stile non vengono applicate ai livelli più bassi

Selettori class e id

- Possono essere applicati a tutti gli elementi XHTML, eccettuati <base>, <head>, <html>, <meta>, <script>, <style> e <titolo>. (Inoltre l'attributo class non può essere utilizzato in <basefont> e <param>).

Selettore class

- Selettore class
  - Identifica un insieme di elementi come parte di un gruppo concettuale. Gli elementi di una classe possono essere modificati con una singola regola di stile.
    - » Esempio <h1 class="importante">Attenzione</h>
    - » <p class="importante">Qualche cosa …</h>
    - » NOTA: i nomi di classe non possono contenere spazi
  - Per specificare lo stile è necessario aggiungere il nome della classe al selettore di tipo, separato da un punto (.)
    - » Esempio h1.importante {color: red}
  - Per applicare una proprietà a tutti gli elementi che condividono una stessa classe, va omesso il nome dei tag XHTML
    - » Esempio .importante {color: red}
  - E' possibile definire associare più classi ad un determinato elemento HTML, separando i valori di ogni classe con una spaziatura

Selettore id

- Selettore id
  - Viene utilizzato per identificare un elemento univoco nel file X/HTML.
    - » Es. <p id="d345">Qualche cosa …</h>
    - » NOTA: Il valore di un attributo id deve sempre iniziare con una lettera.
  - Per specificare lo stile è necessario aggiungere il nome dell'identificativo al selettore di tipo, separato da un cancelletto (#)
    - » Esempio p#d345 {color: red}
  - Il nome del tag XHTML può essere omesso
    - » Esempio #d345 {color: red}

Selettori class e id contestuali

- E' possibile utilizzare le classi e gli id per definire il contesto di applicazione di una certa regola
  - Es. #laterale p {color: red}
  - Es. 2 .fantasia td {color: red}

Pseudo-selettori

- La specifica CSS1 prevede un insieme di pseudo-elementi e pseudo-classi che non sono basati su elementi strutturali. Nel file X/HTML non appare alcun codice; essi vengono interpretati dal browser in base al contesto.
- I pseudo-selettori sono indicati dal carattere :
- Sono supportati in maniera affidabile a partire da IE 5.5 + (Win), IE 5 (Mac), Netscape 6 e Opera.

Pseudo-selettori

- Pseudo-elementi
  - In CSS1 sono first-line e first-letter. Possono essere usati per applicare stili alla prima riga o lettera di un elemento XHTML.
    - Es. p:first-line {letter-spacing: 8pt}
  - Possono essere combinati con l'attributo class, in modo da applicare gli effetti di presentazione solo in una certa classe di elementi.
    - Es. p.miaclasse:first-letter {font-size: 200%}
- Pseudo-classi
  - CSS2 specifica quattro pseudo-classi che possono essere applicate al tag <a>: link, visited, active, e hover. Si applicano alle ancore che contengono l'attributo href.
    - Es. a:link {color: red}
      a:visited {color: blu}
      a:hover {color: maroon}
      a:active {color: maroon}
  - Il supporto di Netscape alle pseudo-classi è inconsistente fino alla versione 6; consigliato l'ordine **l-v-h-a** (link-visited-hover-active) per evitare che alcuni browser ignorino le regole.

Aggiungere stili a un documento

- Sono possibili tre modalità:
  - Inline Styles
  - Embedded style sheets
  - Foglio di stile esterno collegato
    - con funzione import
    - con tag link

Aggiungere stili a un documento 1

- Inline Styles
  - L'informazione di stile può essere aggiunta a elementi individuali attraverso l'attributo style all'interno del tag XHTML.
  - Il valore dell'attributo stile corrisponde ad una o più dichiarazioni di stile standard
    - <h1 style="color: red">Titolo rosso</h1>
    - Contenuto
  - E' possibile utilizzare tag generici e per designare elementi condizionati da istruzioni di stile.
    - Esempio Contenuto
    - Esempio Paragrafo con paroleblu
    - Va rilevato comunque che l'associazione a tag div e span ha il difetto di non caratterizzare il contenuto da un punto di vista semantico, ma solo presentazionale e perciò va evitato se sono possibili soluzioni alternative
    - Esempio: da evitare Intestazione ; da sostiuire con <h1 style="font-size: 12pt">Intestazione </h1>
  - Sebbene l'uso di stili inline sia lecito, essi sono equivalenti al tag <font>, nel senso che *inquinano* il documento con informazioni di presentazione; inoltre qualsiasi cambiamento deve essere fatto in ogni tag nel quale è presente l'informazione di stile.
  - Quindi **gli stili inline vanno utilizzati soprattutto per gestire le eccezioni**, cioè per cambiare localmente il comportamento di regole generali di stile stabilite a livello più alto.

Aggiungere stili a un documento 2

- Embedded style sheets
  - Un metodo più compatto consiste nell'aggiungere un blocco con informazioni di stile all'inizio del documento XHTML utilizzando il tag <style> nella sezione <head> del documento.
  - Si possono inserire uno o più elementi di stile in un documento.
  - Attributi di <style>
    - media=screen|tty|tv|projection|handheld|print|braille|aural|all
      per specificare i media di destinazione a cui si applica il foglio di stile
    - type=content-type
      obbligatorio: specifica il linguaggio del foglio di stile; al momento l'unico valore disponibile è text/css, anche se W3C ha previsto la possbilità di linguaggi aggiuntivi; se questo attributo viene omesso, alcuni browser potrebbero ignorare il foglio di stile
    - title=text
      fornisce un titolo al tag di stile

Aggiungere stili a un documento 3

- <html>
  <head>
  <style type="text/css">
  <!--
  h1 {color: red}
  p {font-size: 12pt; font-family: Verdana, sans-serif;}
  -->
  </style>
  <title>Titolo documento</title>
  </head>
  …
  </html>

- NOTA: solitamente è necessario inserire il tag di commento intorno ai contenuti del tag <style>, per evitare che i browser che non capiscono l'informazione di stile mostrino le regole di stile come testo

Aggiungere stili a un documento 4

- Il modo più efficiente per utilizzare i fogli di stile è quello di raccoglierli in un documento di testo separato (con suffisso .css) e creare collegamenti ad esso dalle pagine XHTML.
- Questo documento di stile è un semplice file di testo che contiene una collezione di regole di stile.
- Ci sono due modalità per riferirsi ad un foglio di stile esterno:
  - tag <link>; è il metodo più supportato dai browser; è possibile collegare più di un foglio di stile ad un documento; ad es.

    <head><link rel="stylesheet" href="/pathname/fogliostile.css" type="text/css"></head>

  - Funzione @import; da utilizzare all'interno dell'elemento <style>; ad es:

    <style type="text/css">@import url(http://pathname/fogliostile.css);</style>
  - E' possibile rifersi o importare più di un foglio di stile in un documento. L'informazione dell'ultimo file importato ha la precedenza sulle altre regole.
  - @import è supportato solo dalle generazioni più recenti dei browser (es. Explorer 4 e sup. e da Netscape 6 e superiori)
- E' possibile utilizzare le due modalità in maniera combinata nello stesso file XHTML

La cascata

- L'informazione di stile possono essere specificate in luoghi diversi e con gradi diversi di specificità. Tutte le regole fluiscono, come una cascata, in un'unico foglio di stile virtuale dove però possono sorgere dei conflitti, dovuti al fatto che ad uno stesso elemento HTML sono attribuite regole di stile diverse.
- In questo caso come ci si deve regolare? Fortunatamente la specifica CSS stabilisce delle priorità

Per risolvere i conflitti ...

- Primo passo: raccogli tutte le regole che riguardano un determinato elemento HTML
- ORIGINE In caso di conflitto stabilisci la priorità in base all'**origine** delle regole (dove le regole sono state scritte), partendo dal default del browser (priorità più bassa) fino alla regole di stile inline (priorità più alta)
- SPECIFICITA' In caso di ulteriore conflitto considera la **specificità** del selettore CSS, considerando come più importanti (nell'ordine) il riferimento ad id e il loro numero, il riferimento ad attributi e pseudo-classi e il loro numero, il riferimento a elementi HTML e il loro numero
- ORDINE DI SPECIFICA In caso di ulteriore conflitto considera come più importante la regola che è stata scritta per ultima

Origine

- Gerarchia delle istruzioni di stile va dal generale allo specifico;
  gli ultimi elementi hanno maggior peso
  - Settaggi di default del browser
  - Foglio di stile dell'utente (se consentito dal browser)
  - Fogli di stili esterni collegati con tag <link>;
  - Fogli di stili esterni collegati con funzione @import;
  - Fogli di stile incorporati (embedded) nel documento;
  - Informazione di stile inline
  - attributi HTML (**nota:** vista l'importanza data agli attributi HTML, è bene che nella revisione di un layout esistente il codice HTML venga depurato da essi, in modo che la loro presenza non interferisca con le regole di stile CSS)

Origine

- Se si desidera che una regola non venga sopraffatta da una regola descritta ad un livello più specifico, è necessario utilizzare l'indicatore **!important** dopo il valore della proprietà
- Esempio:
- Se la regola **p {color: blue !important;}** venisse specificata in un file CSS collegato/importato oppure in una sezione incorporata nel documento XHTML e il documento stesso contenesse anche uno stile inline come <p style="color: red">, quest'ultima regola verrebbe ignorata (e quindi il testo del paragrafo verrebbe visualizzato comunque con il colore blu)
- L'indicatore viene supportato dalla maggior parte dei browser

Specificità

- Compariamo le specificità di
  "h2 em { color: green }" e "h2 em#foo { color: red }":
  - Specificità di "h2 em { color: green }" = 0-0-2 (or "2")
    nessun riferimento ad id, nessun riferimento ad attributi e pseudo-classi, 2 riferimenti a elementi HTML
  - Specificità di "h2 em#foo { color: red }" = 1-0-2 (or "102")
    1 riferimento ad id, nessun riferimento ad attributi e pseudo-classi, 2 riferimenti a elementi HTML
- Concludendo, "h2 em#foo { color: red }" ha una specificità maggiore e quindi ha la prevalenza sulla prima regola

Specificità, un altro esempio

- #id1 {xxx}
- UL UL LI.red {xxx}
- LI.red {xxx}
- LI {xxx}

Chi vince?

Specificità, un altro esempio

- **# id1 {xxx} /* a=1 b=0 c=0 --> specificity = 100 Winner */**
- UL UL LI.red {xxx} /* a=0 b=1 c=3 --> specificity = 013 */
- LI.red {xxx} /* a=0 b=1 c=1 --> specificity = 011 */
- LI {xxx} /* a=0 b=0 c=1 --> specificity = 001 */

Ordine di specifica ...

- Se due regole di peso uguale vengono utilizzate in uno stesso foglio di stile, ha la prevalenza l'ultima regola dichiarata.
- Allo stesso modo, se due regole di ugual peso vengono dichiarate in fogli di stile diversi (collegati con il tag <link> o importati), ha la prevalenza la regola dichiarata nell'ultimo foglio di stile