

Linguaggi

*Corso M-Z - Laurea in Ingegneria Informatica
A.A. 2008-2009*

Alessandro Longheu

<http://www.dit.unict.it/users/alongheu>

alessandro.longheu@dit.unict.it

- lezione 25 -

CSS

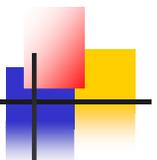
1

A. Longheu – Linguaggi M-Z – Ing. Inf. 2008-2009

Schema delle lezioni

- Modalità di utilizzo
- Cascading e inheritance
- Statement rules:
 - selectori (cosa formattare)
 - dichiarazioni (come formattare)
- Fogli di stile multipli (alternativi)
- CSS per tabelle
- Cursori
- Media e Stampa
- Software e validatori

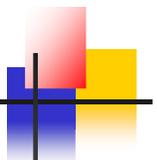
2



Cascading Style Sheets

- CSS è una raccomandazione del World Wide Web Consortium (W3C).
- I fogli di stile forniscono un mezzo per separare l'apparenza (aspetto grafico) di una pagina WEB dal contenuto della stessa, ponendo la prima nei fogli di stile, e lasciando il secondo entro il file HTML.
- Allo stato attuale esistono diversi livelli di CSS, CSS1(1996, 1999), CSS2(1998), CSS 2.1 (2007) e CSS3 (in sviluppo).
- E' bene accertarsi della versione di CSS supportata dai browser (tutti supportano almeno fino alla 2)

3



CSS - Generalità

- Ogni versione del Css (ovvero ogni raccomandazione W3C) è una grammatica che definisce quali statement possono essere presenti in un foglio di stile.
- Un foglio di stile è di fatto un semplice file di testo (con estensione .css) scritto in accordo ad una di tali grammatiche, ad esempio:

.more

```
{background-color: #003333;  
border-width: thin;  
border-color: black;  
border-style: ridge;  
color: white;  
font-family: verdana, geneva, sans-serif;  
font-size: .9em;  
vertical-align: text-bottom}
```

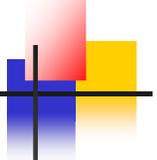
4



CSS - Modalità di utilizzo

- Esistono tre modalità di utilizzo del CSS:
 - Foglio di stile **in-line**
 - lo stile viene applicato “al volo” direttamente ad un tag
 - Foglio di stile **interno**
 - lo stile viene definito dentro <HEAD>
 - Foglio di stile **esterno**
 - lo stile è definito su un file che viene riferito dal codice HTML

5



CSS - Modalità di utilizzo

Stile in-line:

- Si possono inserire indicazioni di stile direttamente nel singolo TAG. Con questo metodo non si scrive nulla nell'head e si utilizza una sintassi di questo tipo:
`<h1 style="color:red;text-transform:capitalize">`
- Le indicazioni di stile sono presenti nei tag all'interno del BODY → è poco usato e poco conveniente!

6

CSS - Modalità di utilizzo

Stile interno:

- Il posizionamento del foglio di stile all'interno della pagina HTML prevede la seguente sintassi:

```
<head>  
  <style type="text/css">  
    H1 {font-size:medium; color:#FF00FF}  
  </style>  
</head>
```

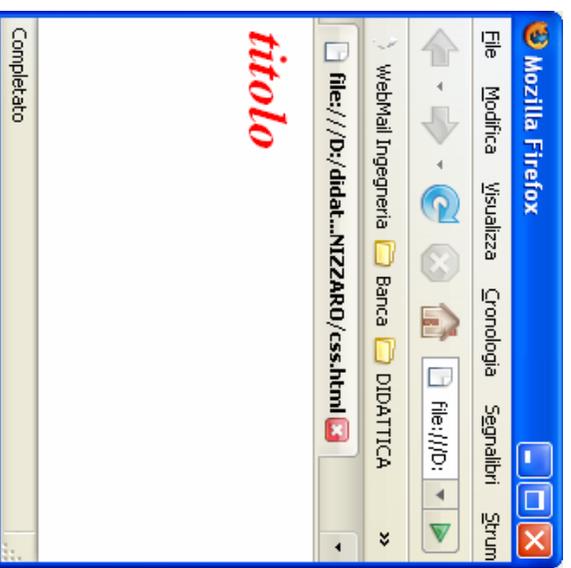
- Nel BODY si richiamano poi i TAG senza inserire attributi

7

CSS - Modalità di utilizzo

ESEMPIO:

```
<html>  
<head>  
<style type="text/css">  
  H1 {color:red;  
    font-style:italic}  
  </style>  
</head>  
<body>  
<h1> titolo</h1>  
</body>  
</html>
```



8

CSS - Modalità di utilizzo

Stile esterno:

- L'utilizzo di un foglio di stile esterno è l'unico modo per poter modificare un intero sito senza dover variare tutte le pagine HTML.
- Si inserisce nella sezione <head> del codice HTML, utilizzando il tag <link>.
- Il tag <link> ha come attributi:
 - rel, che indica il tipo di risorsa a cui ci si collega: rel="stylesheet"
 - type, che indica il tipo di foglio di stile: type="text/css"
 - href, che indica l'URL del file .css

9

CSS - Modalità di utilizzo

```
<html>  
<head>  
<link rel="stylesheet"  
      type="text/css"  
      href="stile.css">  
</head>  
...
```

- Le indicazioni di stile vengono inserite direttamente in un documento a parte chiamato STILE con estensione CSS (un file di testo)

10

CSS - Modalità di utilizzo

- In un documento HTML gli stili possono essere definiti a diversi livelli.
- In caso di ripetute definizioni di stile, l'ordine in cui vengono applicate (a priorit  crescente)  :
 1. Browser default
 2. External Style Sheet
 3. Internal Style Sheet (all'interno del tag <head>)
 4. Inline Style (all'interno dell'elemento HTML)

11

CSS – Cascading & Inheritance

- Uno stesso stile pu  essere realizzato tramite pi  fogli, che possono eventualmente essere collegati in modo da creare una **gerarchia**, in cui ad esempio un foglio contiene tutte le specifiche di stile generali del sito di un'organizzazione, mentre diversi fogli di livello pi  basso contengono specifiche aggiuntive per i vari gruppi all'interno della stessa organizzazione. Questo approccio consente una divisione pi  razionale degli stili, risolvendo il problema della gestione di un singolo file .css, che diventa difficile per siti di grosse dimensioni.
- Per costituire una gerarchia, ogni foglio di stile pu  specificare quali fogli di stile importa. Il foglio di stile che ne importa altri, si dice in cascata con essi, da cui **cascading** (style sheets).
- L'applicazione di un certo stile ad un tag Html x, viene automaticamente estesa a tutti quei tag che si trovano all'interno di x, a meno che per qualcuno di essi non venga esplicitamente indicato uno stile.
- Questo meccanismo, che viene applicato alla cosiddetta gerarchia di contenimento della pagina,   noto come **ereditariet ** del Css.¹²

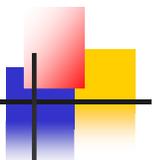
CSS - Utilità

- Il file .css contiene tutte le informazioni relative a come visualizzare i vari tag Html di un documento. **In assenza dei fogli di stile**, tali informazioni sono cablate nel file html stesso, approccio che costituisce una **forte limitazione** in quanto:
 - se entro lo stesso file html è usato un certo tag N volte, la specifica di visualizzazione deve essere scritta N volte (una in corrispondenza di ogni tag);
 - se esistono M pagine html (ad esempio, un intero sito), occorre ripetere le specifiche per ogni pagina;
 - cablare le specifiche non assicura la coerenza, sicchè se si vuole cambiare lo stile, si devono effettuare al più N*M modifiche.
- Un foglio di stile **risolve tutte queste problematiche**, in quanto la specifica di formattazione di un dato tag è fornita una volta sola entro il file .css ed automaticamente applicata a tutti i tag entro tutti i file html che siano stati collegati a quel foglio di stile.
- Un tale approccio assicura la coerenza dello stile, consente modifiche rapide e una facile condivisione di stili, molto utili¹³ all'interno di siti di grosse dimensioni.

CSS - Statement

- Le istruzioni di un file CSS sono chiamate Statements. Ve ne sono diversi tipi: Rules, @rules, @media rules, @page instructions, Commenti
- Uno **Statement di tipo Rules** è uno statement che suggerisce al browser come designare un dato elemento (tag) Html. E lo statement più importante. Una Rule ha due sezioni:
 - quella che identifica un elemento nella pagina (**selector**). Tale selector può essere il nome del tag, privato di < e >. Esistono anche altri tipi di selectori, ad esempio per selezionare un sottoinsieme (classe) di tag dello stesso tipo, o per altri compiti.
 - quella che suggerisce al browser come l'elemento deve essere rappresentato (**declaration**). Si parla di suggerimento in quanto il browser, a seconda della sua configurazione e/o di quella locale, potrebbe non soddisfare tutte le richieste. La declaration è una sequenza di nome attributo : valore racchiusa fra { e }.
- Esempio di rule:

```
BODY {
font-family: verdana, sans-serif;
font-size: 1em;
text-align: justify;
}
```



CSS - Statement

- @rules sono delle macro, ossia insiemi di istruzioni per compiti specifici. Nel Css1, l'unica ammessa è la @import, per importare fogli di stile. Nel Css2 esiste anche l'import condizionale a seconda del tipo di media usato per la rappresentazione.
- @media (Css2) consente di specificare parti del foglio di stile da usare solo in corrispondenza di un dato media (schermo, braille, stampa) usato per la rappresentazione, ad esempio

```
@media print { BODY {background-color: white} }
```

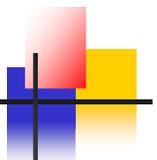
rende lo sfondo bianco solo quando la pagina sarà stampata. In generale, la sintassi è:

```
@media <lista media> { <lista regole> }
```
- @page (Css2) consente di fornire regole specifiche per la stampa, ad esempio

```
@page { margin: 5% }
```
- Comments. La sintassi è la tipica

```
/* ... */
```

15



CSS - Rules

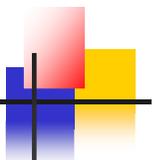
La sintassi per la definizione di una regola è:

- Gli attributi sono inseriti tra parentesi graffe { }
- Al posto di “=” vengono usati i due punti “:”
- Gli argomenti consecutivi sono separati dal segno “,”
- Gli attributi composti da più termini sono separati da un trattino “_”
- Se il valore dell'attributo è composto da più parole sono necessarie le virgolette “”, ad esempio:

```
p {font-family: "sans serif" }
```

```
p {font-family: "san serif", verdana}
```

16

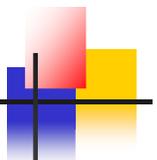


CSS - Rules - selectori

Definizione dei Selectori

- Un selettore può definire gli elementi da formattare nei seguenti modi:
 - in base al nome dell'elemento;
 - in base a raggruppamenti di elementi in classi e in ID;
 - in base al fatto che gli elementi abbiano certi attributi e valori;
 - in base al contesto cui l'elemento appartiene;
 - in base a pseudo-classi e a pseudo-elementi.
- La definizione può anche avvenire combinando uno o più dei succitati criteri.
- Inoltre è possibile applicare le stesse dichiarazioni a più selectori elencandoli separati da una “ , ”

17



CSS - Rules - selectori

Selectori basati sul nome di elementi:

- Questa categoria di selectori è molto semplice in quanto gli elementi cui si vogliono applicare le regole si identificano con il nome che hanno nel linguaggio HTML.
- Supponiamo di volere fare in modo che tutti i paragrafi e le intestazioni di livello tre abbiano testo di colore verde e giustificato:

```
p,h3 {color: green; text-align: justify;}
```

18

CSS - Rules - selectori

Selectori di classe e selectori ID

- Attraverso le classi e gli ID si possono specificare regole da applicare in modo indipendente dai marcatori HTML.
- Una classe si definisce utilizzando il simbolo «.» seguito dal nome della classe e preceduto (non obbligatoriamente) dal nome di un marcatore. Vediamo un esempio:


```
p.classe1 {color: red; text-align: left;}
.classe2 {font-weight: bold;}
```
- In questo modo vengono definita una classe *classe1* relativa al marcatore '**<p>**' e una classe *classe2* relativa a qualsiasi marcatore.
- Nel codice HTML si può fare riferimento a queste regole usando l'attributo '**class**' per i tag cui si vogliono applicare tali formattazioni, ad esempio nel modo seguente

19

CSS - Rules - selectori

```
<h1 class="classe2">
```

```
Titolo di classe2 </h1>
```

```
<p class="classe1">
```

```
Testo di colore rosso
```

```
e allineato a sinistra </p>
```

```
<p>
```

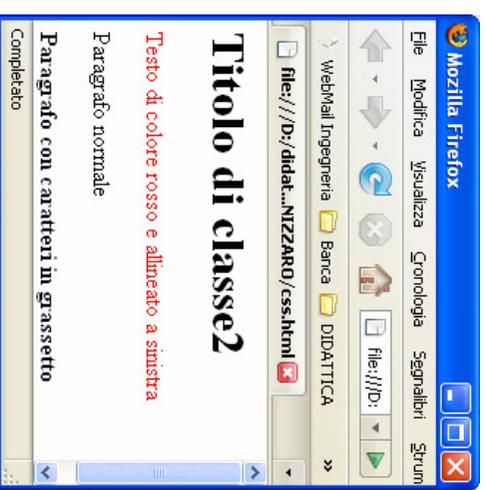
```
Paragrafo normale </p>
```

```
<p class="classe2">
```

```
Paragrafo con caratteri in grassetto </p>
```

- Come si vede la classe *classe2* si può associare a qualsiasi tag, mentre la classe *classe1* solo a marcatori '**<p>**'.

20



CSS - Rules - selectori

- Un ID si definisce utilizzando il simbolo «#» seguito dal nome della classe e preceduto (non obbligatoriamente) dal nome di un marcatore. Vediamo un esempio:


```
#identi {color: green; font-style: italic;}
```
- Nel codice HTML si può fare riferimento a questa regola usando l'attributo **'id'** per il tag cui si vuole applicare tali formattazioni, ad esempio nel modo seguente:


```
<h1 id="identi">Titolo con ID identi</h1>
```
- La differenza fondamentale tra i selectori di classe e i selectori ID è che i primi si possono usare in associazione a quanti elementi si vuole i secondi invece una sola volta all'interno di un documento HTML (identificano quindi un caso "unico" di formattazione). Questa regola non è imposta da tutti i browser, ma lo è nello standard XHTML, per cui è bene rispettarla anche se non controllata dal browser. Infine, un elemento HTML può subire lo stile specificato in più classi simultaneamente mentre un solo ID può essere applicato)

21

CSS - Rules - selectori

Selectori in base a attributi e valori

- La selezione di marcatori può avvenire anche in base alla presenza di un attributo oppure di un attributo con un dato valore (anche parziale); l'indicazione dell'attributo, semplice o valorizzato deve avvenire tra i simboli «[>» e «[>».
- L'esempio seguente mostra nell'ordine l'uso di un selettore di attributo semplice, di un selettore di attributo esatto e di un selettore di attributo parziale (si noti in questo caso l'uso dell'operatore «~=>»):

```
[align] {font-style: italic;}
```

```
a[href="www.google.com"] {font-weight: bold;}
```

```
a[href~="google"] {font-weight: bold;}
```

CSS - Rules - selectori

Selectori in base al contesto

- I selectori basati sul contesto tengono conto della struttura del documento HTML e permettono di fare riferimento a marcatori che sono «discendenti», «figli», «adiacenti» di altri marcatori.
- Un marcatore è discendente di un altro (che viene detto «antenato») quando è annidato in esso a un qualunque livello di profondità; è invece figlio di un marcatore (che viene detto «genitore») se è un discendente diretto;
- infine due marcatori si dicono adiacenti se hanno lo stesso genitore.
- Si tenga presente che in documento HTML il tag '<html>' è l'antenato di qualsiasi altro marcatore e per questo viene detto «elemento radice» (root).

23

CSS - Rules - selectori

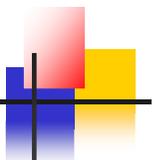
- Consideriamo il seguente codice HTML in cui abbiamo due tag '<h1>' e '<p>' adiacenti e figli di '<body>', e i tag '' e '' figli rispettivamente di '<h1>' e '<p>' (oltre che, ovviamente, discendenti di '<body>'):


```
<body>
<h1>Esempio per mostrare <em>struttura</em> </h1>
<p>Testo normale, <b>testo in grassetto</b></p>
</body>
```
- La selezione in base alla struttura avviene con la sintassi mostrata nell'esempio seguente:


```
h1 em {color: yellow;}
p > b {font-style: italic;}
h1 + p {text-align: justify;}

```
- I tre selectori dell'esempio si interpretano nel seguente modo:
 - seleziona i tag '' che siano discendenti di un '<h1>';
 - seleziona i tag '' che siano figli di un '<p>';
 - seleziona i tag '<p>' che siano adiacenti di un '<h1>';

24

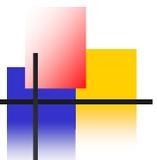


CSS - Rules - selectori

- Esiste anche la possibilità di **combinare le pseudo-classi** come mostrato di seguito:

```
a:link:hover {color: magenta;}  
a:visited:hover {color: red;}  
a:link:visited {color: black;}
```
- Nel primo caso si vogliono selezionare i collegamenti non visitati su cui è posizionato il puntatore del mouse; nel secondo caso stessa cosa per i collegamenti visitati; il terzo caso è assurdo (un collegamento non può essere allo stesso tempo visitato e non visitato) e viene solitamente ignorato dai browser

27



CSS - Rules - selectori

- con gli **pseudo-elementi** è possibile selezionare e applicare gli stili alla prima lettera o alla prima riga o anche prima e dopo un certo elemento.

```
p:first-letter {color: green;}  
p:first-line {color: red;}
```
- Molto interessanti sono gli pseudo-elementi `‘:before’` e `‘:after’` che usati congiuntamente alla regola `‘content’` permettono di inserire un contenuto (chiamato «contenuto generato») nel documento HTML prima o dopo un certo marcatore; esempio:

```
h1:before {content: "Nuovo capitolo"; display: block; color: green;}  
body:after {content: "Fine del documento"; font-style: oblique;}
```
- L'effetto è quello di inserire la stringa «Nuovo capitolo» di colore verde prima di ogni `<h1>` e la stringa «Fine del documento», obliqua, alla fine del testo visualizzato. Interessante è anche l'uso della proprietà `‘display’` che nell'esempio serve a inserire il testo generato come elemento di blocco e quindi a fare proseguire il testo contenuto nel tag `‘<h1>’` su un nuovo paragrafo

28



CSS - Rules

Dichiarazione di regole di formattazione

- Una volta visto come è possibile selezionare le parti del documento cui applicare gli stili passiamo ad elencare alcuni di quelli relativi alla formattazione del testo.
- L'elenco viene fornito con una tabella contenente le proprietà, il loro significato, i valori (almeno i più importanti) che esse possono assumere e se sono «ereditate» oppure no;
- Quando si fa riferimento a grandezze (es. misura di caratteri) si possono utilizzare varie **unità di misura**; alcune sono assolute:
 - pixel (px); punti (pt); picas (pc, pari a 12 pt)
 - millimetri (mm); centimetri (cm); pollici (in)
- Altre unità di misura sono relative, come:
 - *em-height* (em), altezza media dell'insieme di caratteri usato;
 - *ex-height* (ex), altezza del carattere «x» dell'insieme di caratteri usato.
- Esiste poi la possibilità di esprimere grandezze in percentuale, calcolata di solito in base alle dimensioni del tag genitore di quello considerato.

29



CSS - Rules

Dichiarazione di regole di formattazione

- **Altre unità di misura**, più infrequenti ma utili, sono:
 - Due proprietà relative ai dispositivi audio possono essere espresse come **angoli**. Le due proprietà sono azimuth e elevation (posizione ed elevazione della sorgente sonora). Le unità di misura sono:
 - deg (degree - grado): descrive l'ampiezza di un angolo (es. 90deg).
 - grad (gradians): descrive l'ampiezza di un angolo su una scala 1-400 (es. 100grad = 90deg)
 - rad (radians): descrive l'ampiezza di un angolo su una scala 1-pi greco
- Le **unità di tempo** trovano spazio solo negli stili audio. Sono usate in genere per impostare la pausa tra le parole lette da un sintetizzatore vocale. Si applicano solo alle proprietà: pause, pause-after, pause-before:
 - s (secondi) e ms (millisecondi)
- Le **unità di frequenza**, usate solo negli stili audio, definiscono la frequenza del segnale:
 - hz (Hertz) e khz (Kilohertz)

30

CSS - Rules

Regole di formattazione

- Una visione schematica delle regole:
 - colori
 - gestione dello sfondo
 - gestione del testo
 - box model: dimensione, margini, padding, bordi
 - posizionamento: termini, proprietà (display, float, clear), position, visibility, z-index

31

CSS - Rules

- I colori possono essere forniti tramite:
 - parole chiave, in particolare: black | navy | blue | maroon | purple | green | red | teal | fuchsia | olive | gray | lime | aqua | silver | yellow | white
- codice RGB, in tal caso si può utilizzare la notazione classica (#FF003A), ma anche dare i valori in % (rgb(10%,0%,35%)), o in decimale (rgb(255,255,0))
- Per ogni elemento si possono definire almeno tre colori:
 - il colore di primo piano (foreground color)
 - il colore di sfondo (background color)
 - il colore del bordo (border color)
- La proprietà color influisce sul colore di primo piano (quello del testo) e sul colore del bordo (a meno che non si imposti esplicitamente quest'ultimo con le proprietà border o border-color)
- Una buona norma dei CSS vuole che per un elemento di cui si sia definito il colore di primo piano si definisca anche e sempre un colore di sfondo (tramite la proprietà background-color).



32

CSS - Rules

- La lista delle proprietà, applicabili a tutti gli elementi, che permettono la **gestione dello sfondo** e la seguente:
 - background-color (colore)
 - background-image (immagine di sfondo)
 - background-repeat (se l'immagine è ripetuta o no nella pagina)
 - background-attachment (se l'immagine rimane fissa nello scroll)
 - background-position (posizionamento)
- Ciascuna di essa definisce un solo, particolare aspetto relativo allo sfondo di un elemento, novità del CSS rispetto all'HTML, in cui era possibile definire un colore o un'immagine di sfondo solo per il corpo principale della pagina (<BODY>) o le tabelle.
- Un'altra limitazione dell'HTML riguarda il comportamento delle immagini di sfondo: esse vengono infatti ripetute in senso orizzontale e verticale fino a riempire l'intera area della finestra, del frame o della tabella. Proprio questo comportamento ha fatto sempre propendere per la scelta di piccole textures in grado di dare la sensazione visiva dell'uniformità. Il CSS rimuove queste limitazioni

33

CSS - Rules

- **La lista delle proprietà del testo** è la seguente:
 - I font da usare, esempio:
 - `p {font-family: arial, Verdana, sans-serif;}`
 - Quando la pagina verrà caricata, il browser tenterà di usare il primo font della lista. Se questo non è disponibile userà il secondo. In mancanza anche di questo verrà utilizzato il font principale della famiglia sans-serif presente sul sistema. E' buona norma indicare sempre alla fine una famiglia; quelle possibili sono serif (Times New Roman), sans-serif (arial), cursive (Comic Sans), fantasy (Allegro BT), monospace (Courier)
 - la sua dimensione (**font-size**), assoluta (le sette parole chiave xx-small, x-small, small, medium, large, x-large, xx-large, oppure un valore espresso in punti o pixel. Si consiglia di usare la prima solo per CSS destinati alla stampa) o relativa (le parole chiave smaller e larger o un valore espresso in percentuale)
 - la sua consistenza (**font-weight**), con un valore numerico (100, 200, ... 900), o con le parole chiave normal (400), bold (700), bolder, lighter
 - lo stile (**font-style**), normal, italic, oblique (rendering corsivo)
 - l'interlinea tra i paragrafi (**line-height**), valore numerico o in percentuale
 - l'allineamento del testo (**text-align**), con valori left, right, center e justify
 - la sua decorazione (**text-decoration**), con valore underline, overline, line-through o blink

CSS - Rules

- Ancora per il testo, è possibile definire:
 - **font-size-adjust**, che può servire a migliorare la leggibilità di un carattere alternativo tra quelli indicati con la proprietà font-family. La proprietà è supportata solo dai browser Gecko-based (Mozilla, Netscape 6/7) ed è ereditata.
 - **font-stretch**, che consente di rendere il testo più o meno espanso del normale. Proprietà non supportata da nessun browser.
 - **font-variant** che consente di trasformare il testo in maiuscoletto
 - **text-indent**, che definisce l'indentazione della prima riga in ogni elemento contenente del testo.
 - **text-shadow** che crea un'ombreggiatura per il testo specificato. La proprietà non è supportata da nessun browser
 - **text-transform**, che serve a cambiare gli attributi del testo relativamente a tre aspetti: maiuscolo, minuscolo, prima lettera maiuscola.
 - **white-space** che serve a gestire il trattamento degli spazi bianchi presenti in un elemento, in maniera simile all'HTML.
 - **letter-spacing** aumenta lo spazio tra le lettere di una parola.
 - **word-spacing**, complementare a letter-spacing. Serve ad aumentare lo₅ spazio tra le parole comprese in un elemento.

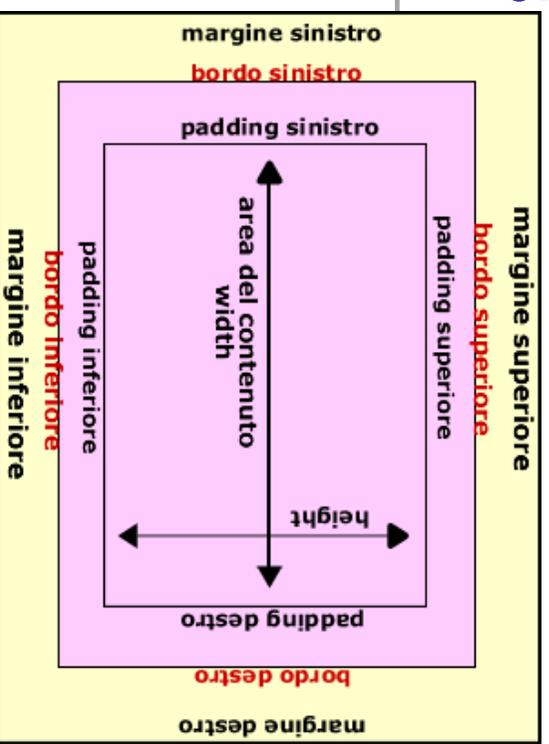
CSS - Rules

Box model e posizionamento con i fogli di stile

- Per parlare di posizionamento con i fogli di stile occorre prima di tutto chiarire come questi ultimi considerano e gestiscono gli elementi presenti in una pagina:
 - ogni elemento genera un **riquadro rettangolare (o box)** non visibile comprendente:
 - l'area del contenuto dell'elemento;
 - lo spazio tra il contenuto dell'elemento e i bordi dell'area (padding);
 - i bordi esterni dell'area (border);
 - lo spazio intorno ai bordi (margin).
- L'unica parte sempre presente è l'area del contenuto, le altre possono essere impostate ad ampiezza nulla. ³⁶

CSS - Rules

Figura esplicativa:



- Gli elementi presentati in figura non sono stati introdotti con i CSS, ma fanno parte del normale meccanismo di rendering di un documento ed è il browser ad applicare per alcune di queste proprietà le sue impostazioni predefinite. La novità è che con i CSS possiamo controllare con precisione al pixel tutti questi aspetti.

37

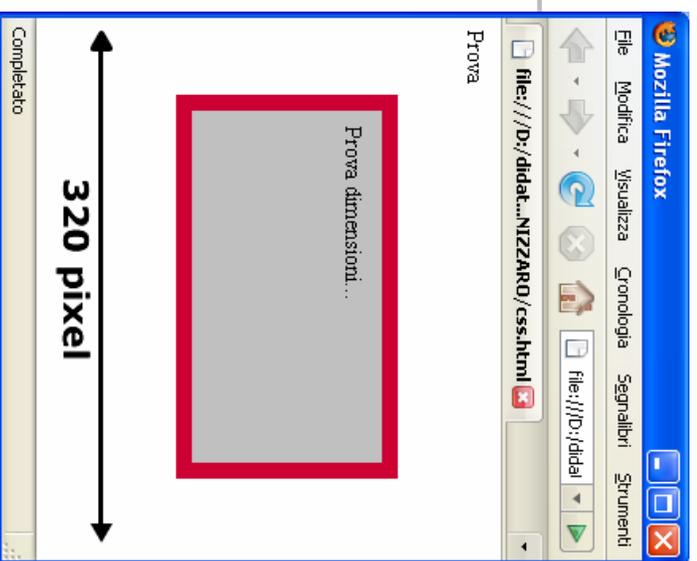
CSS - Rules

Esempio:

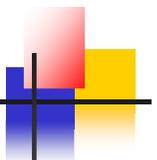
```
<html><head><style>
#div1 {
background: Silver;
margin : 40px;
border : 10px solid #CC0033;
padding : 10px;
width : 200px;
height : 100px;
}
</style>
```

```
<body>Prova<br>
<p id="div1"> Prova dimensioni... </p>

</body></html>
```

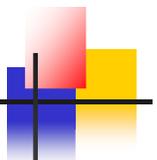


38



CSS - Rules

- Altezza: un primo concetto fondamentale: in genere **l'altezza di un elemento è determinata dal suo contenuto**. Più testo inserisco in box, in un paragrafo o in una cella di tabella più essi saranno estesi in senso verticale. Semplice. Le proprietà del CSS servono a fornire un meccanismo in grado di controllare o superare questo comportamento.
- **height** definisce la distanza tra il bordo superiore e quello inferiore di un elemento. Non è ereditata. Valori: numero, %, auto.
- L'uso di height va sempre valutato con attenzione e non pensando di farne una scorciatoia per avere layout precisi al pixel. Se il contenuto dovesse superare i limiti imposti, Explorer non fa altro che ignorare l'altezza impostata estendendola. Mozilla e Opera rispettano la regola, ma il contenuto eccedente va a sovrapporsi agli elementi vicini o sottostanti. Esiste un modo per gestire il contenuto che superi i limiti imposti con height: usare la proprietà overflow. Si applica a tutti gli elementi blocco e non è ereditata. Valori:
 - **visible**. Valore iniziale. Il contenuto eccedente rimane visibile.
 - **hidden**. Il contenuto eccedente non viene mostrato.
 - **scroll**. Il browser crea barre di scorrimento.
 - **auto**. Il browser gestisce il contenuto eccedente Di norma dovrebbe mostrare una barra di scorrimento laterale.
- **min(max)-height** imposta un'altezza minima (massima) per un elemento. Non è ereditata ³⁹



CSS - Rules

- La definizione della **larghezza** e più in generale la formattazione orizzontale degli elementi sono più problematiche rispetto all'altezza e alla formattazione verticale.
- Un errore macroscopico è ritenere che esse siano definite semplicemente dalla proprietà width. In pratica, dovete sempre distinguere tra la larghezza effettiva di un elemento (i pixel di spazio che occupa sulla pagina) e la larghezza dell'area del contenuto. La prima è data dalla somma di margine sinistro + bordo sinistro + padding sinistro + area del contenuto + padding destro + bordo destro + margine destro. È possibile ovviamente che i due valori coincidano. Accade quando di un particolare elemento non si impostino margini, padding e bordi; o quando il valore di tali proprietà sia uguale a 0. Explorer per Windows fino alla versione 5.5 interpreta male il concetto di width. Nel senso che con questa proprietà intende la larghezza globale dell'elemento, compresi margini, padding e bordi.
- Esistono anche min-width e max-width, i cui valori (insieme a width) sono definiti come per height

CSS - Rules

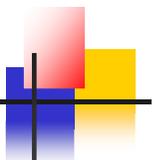
- **Margini:** Una delle maggiori limitazioni di HTML è la mancanza di un meccanismo interno per spaziare gli elementi di un documento. Per ottenere questo risultato si è ricorso negli anni ad una serie di trucchetti, magari efficaci, ma fuori dalla logica originaria del linguaggio. L'unica eccezione consisteva nella possibilità di definire una distanza tra le celle di una tabella con l'attributo cellspacing. I CSS risolvono il problema con l'uso di una serie di proprietà in grado di definire con precisione la distanza tra gli elementi. Possiamo infatti definire il margine come la distanza tra il bordo di un elemento e gli elementi adiacenti.
 - per il margine, esistono quattro proprietà: **margin-top**, **margin-bottom**, **margin-left** e **margin-right**, tutte con ovvio significato. I valori possono essere numerici, in %, o "auto". Se left e right sono auto, si ottiene la centratura orizzontale.
 - Per due box adiacenti in senso verticale che abbiano impostato un margine inferiore e uno superiore la distanza non sarà data dalla somma delle due distanze. A prevalere sarà invece la distanza maggiore tra le due. E il meccanismo del cosiddetto **margin collapsing**. Tale meccanismo non si applica ai box adiacenti in senso orizzontale.
 - Un altro modo per creare spazio intorno ad un elemento è quello di usare il **padding**, che rappresenta la spaziatura interna all'elemento. Come per il margine, esistono quattro valori.

41

CSS - Rules

- **Bordi:** l'impatto visivo da essi offerto è di sicuro effetto: la pagina acquista rigore e una suddivisione più logica, estetica e razionalità possono trovare un buon compromesso
- si possono definire per ciascun bordo (top, bottom, left e right) tre aspetti: colore, stile, spessore, in accordo alla sintassi `border-< lato >-< aspetto >: <valore>`, esempio :
 - `border-top-color: blue`
 - `border-left-style:dashed`
- I valori possibili per il colore sono **un qualsiasi colore** o **"inherit"**
- Lo stile di un bordo può invece essere espresso con una delle seguenti parole chiave: **none**, **hidden** (equivalente a none), **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**
- Lo spessore può essere modificato secondo i seguenti valori: **un valore numerico con unità di misura**, **thin**, **medium**, **thick** (la rappresentazione grafica di questi ultimi tre è a carico del browser)

42



CSS - Rules

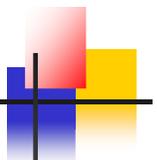
Qualche termine per comprendere il posizionamento dei box:

- **I box possono essere di blocco o in linea**, con le conseguenze che questo comporta
- **Gli elementi possono essere «rimpiazzati» o «non rimpiazzati»**; nel primo caso la larghezza ed altezza dell'elemento sono definite dall'elemento stesso e non da ciò che lo circonda (l'esempio più banale è '', ma anche textarea, input e select nelle form); Tutti gli altri elementi sono in genere considerati non rimpiazzati. La distinzione è importante perchè per alcune proprietà è diverso il trattamento tra l'una e l'altra categoria, mentre per altre il supporto è solo per la prima, ma non per la seconda.
- **Ogni elemento è contenuto in un «blocco contenitore»** :

```
<body><div><h1>Titolo</h1><p>Testo</p></div></body></html>
```

 i tag '<h1>' e '<p>' hanno '<div>' come contenitore ed esso è contenuto in '<body>'.
- Il **«flusso» di visualizzazione** di un documento HTML è da sinistra a destra e dall'alto verso il basso, questo è il comportamento di tutti i browser almeno per le lingue occidentali.

43



CSS - Rules

- **Una prima proprietà per il posizionamento è display.** Essa definisce il trattamento e la presentazione di un elemento. Fin quando non la si usa ognuno segue natura e comportamento standard, ma con display possiamo intervenire su di esso. La proprietà è ereditata.
 - I valori più comuni sono:
 - inline (default): l'elemento assume le caratteristiche degli elementi inline.
 - block. L'elemento viene trattato come un elemento blocco.
 - list-item. L'elemento si trasforma in un elemento lista.
 - compact. L'elemento viene piazzato al fianco di un altro. Non supportato da nessun browser.
 - marker. Questo valore fa sì che il contenuto generato con gli pseudo-elementi :before e :after sia trattato come un marcatore di liste. Non supportato da nessun browser.
 - none. L'elemento non viene mostrato. O meglio: è come se non fosse nemmeno presente nel documento, in quanto non genera alcun box. Diversa la proprietà visibility:hidden, che invece si limita a nascondere l'elemento lasciando lo spazio da esso occupato.

44



CSS - Rules

- L'utilizzo della proprietà `display` **si rivela utile** in diversi casi:
- Le immagini, per esempio, sono per loro natura elementi inline, si inseriscono nel testo ed è talvolta complicato gestirne il posizionamento. Se si volesse dedicare una zona separata per ogni immagine basterebbe impostare il `display` su `block`:
`img {display: block;}`
- Il valore `none` permette di realizzare una pagina alternativa fatta solo di testo, basta costruire un CSS alternativo dove si inserisce la regola
`img {display: none;}`

45



CSS - Rules

- Con la **proprietà float** è possibile rimuovere un elemento dal normale flusso del documento e spostarlo su uno dei lati (destro o sinistro) del suo elemento contenitore. Il contenuto che circonda l'elemento scorrerà intorno ad esso sul lato opposto rispetto a quello indicato come valore di float. La proprietà non è ereditata. Il float è un altro caso di funzionalità presenti in HTML solo per certi elementi che i CSS hanno esteso a tutti gli altri (come il padding). Il floating è infatti analogo all'attributo `align` delle immagini HTML; valori: `left`, `right` e `none`. Se si usa il float con le immagini non ci sono problemi perchè esse hanno una dimensione intrinseca che il browser riconosce. Ma se viene applicato ad altri elementi **si deve** esplicitamente impostare una dimensione orizzontale con la proprietà **`width`**.
- La **proprietà clear** serve a impedire che al fianco di un elemento compaiano altri elementi con il float. Si applica solo agli elementi bloccati e non è ereditata.
- L'origine di tale proprietà è questa: visto che il float sposta un elemento dal flusso normale del documento, è possibile che esso venga a trovarsi in posizioni non desiderate, magari al fianco di altri elementi che si vogliono tenere separati. clear risolve questo problema. I valori sono `none` (gli elementi con float possono stare a destra e sinistra dell'elemento), `left` (si impedisce il posizionamento a sinistra), `right`, `both` (si impedisce il posizionamento su entrambi i lati)

46

CSS - Rules

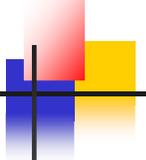
- **La proprietà position** specifica come un generico elemento dovrà essere posizionato. I valori sono: static, relative, absolute, fixed
- **statico** (default), l'elemento rimane nel flusso normale del testo;
- **relativo**: L'elemento viene posizionato relativamente al suo box contenitore. In questo caso il box contenitore è il posto che l'elemento avrebbe occupato nel normale flusso del documento. La posizione viene anche qui imposta con le proprietà **top, left, bottom, right**. che non indicano un punto preciso, ma l'ammontare dello spostamento in senso orizzontale e verticale rispetto al box contenitore.
- **assoluto**: L'elemento, o meglio, il box dell'elemento viene rimosso dal flusso del documento ed è posizionato in base alle coordinate fornite con le proprietà **top, left, right o bottom**. Il posizionamento avviene sempre rispetto al **box contenitore** dell'elemento. Questo è rappresentato dal primo elemento **antenato (ancestor)** che abbia un posizionamento diverso da **static**. Se tale elemento non esiste il posizionamento avviene in base all'elemento radice **HTML**, che in condizioni standard coincide con l'area del browser che contiene il documento e che ha inizio dall'angolo superiore sinistro di tale area. Un elemento posizionato in modo assoluto scorre insieme al resto del documento.

47

CSS - Rules

- **Ultimo valore possibile per position: fixed**, il cui significato è come per assoluto ma considerando come blocco contenitore la finestra del browser. Altra differenza: un box posizionato con **fixed** non scorre con il resto del documento. Rimane, appunto, fisso al suo posto. L'effetto è lo stesso che si può ottenere con l'uso dei frame, in cui una parte della pagina rimane fissa e il resto scorre. Solo che qui il documento è solo uno.
- In caso alcuni riquadri si sovrappongano, è anche possibile ordinarli in base all'asse perpendicolare all'area di visualizzazione, immaginando che il documento sia costituito da **livelli sovrapposti**.
- In ogni caso con i CSS si ottiene un tipo di **posizionamento «liquido»** in quanto tiene conto automaticamente del variare delle dimensione dell'area di visualizzazione della pagina (finestra del browser dell'utente)

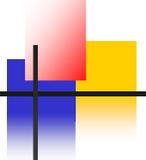
48



CSS - Rules

- Con **position** si stabilisce **come** posizionare un elemento. Le proprietà che definiscono invece **dove** un elemento posizionato sono top, bottom, left, right
- Il significato di top cambia secondo la modalità di posizionamento:
 - Per gli elementi posizionati con absolute o fixed definisce la distanza verticale rispetto al bordo superiore dell'elemento contenitore.
 - Per gli elementi posizionati con relative stabilisce invece l'ammontare dello spostamento rispetto al lato superiore della posizione originaria. In questo caso, usando valori positivi il box viene spostato sotto, mentre con valori negativi lo spostamento avviene verso l'alto
- Valori possibili di top: un valore numerico con unità di misura, un valore in percentuale (relativa all'altezza dell'elemento contenitore) o infine "auto"
- bottom, left e right hanno definizioni analoghe a top
- In genere non occorre usarle tutte e quattro, ne bastano solo due per assegnare una posizione, ad esempio top e left

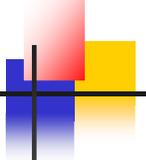
49



CSS - Rules

- **Le proprietà visibility, clip e z-index** influiscono infine sull'aspetto visuale del box definendone la visibilità e la relazione con gli altri box presenti nella pagina.
- **Visibility** determina se un elemento debba essere visibile o nascosto. Si applica a tutti gli elementi e non è ereditata. Valori possibili: visible (valore di default) e hidden, in cui l'elemento è nascosto, ma mantiene il suo posto nel layout dov'è apparirà come una zona vuota. In ciò è diverso dal valore none della proprietà display (che non assegna nemmeno una zona)
- **clip** definisce la parte di un elemento posizionato che deve essere visibile. Nella definizione che ne è stata data con CSS2 non è supportata da nessun browser.
- **z-index** infine imposta l'ordine di posizionamento dei vari elementi sulla base di una scala di livelli. E' un meccanismo simile a quello dei layer sovrapposti di Photoshop ed è utile nel contesto del posizionamento dinamico. In seguito ad un posizionamento, infatti, è possibile che un elemento si sovrapponga ad un altro rendendolo illeggibile. Impostando lo z-index è possibile modificare l'ordine di tutti gli elementi (chi starà in primo piano e chi no). Valori possibili: "auto", in cui l'ordine dei livelli è uguale per tutti gli elementi, o un valore numerico (un valore superiore indica un livello superiore)

50



CSS - Rules

- per utilizzare proficuamente il posizionamento è opportuno usare il tag '<div>' (riservarsi un blocco) o (per operare inline), in modo da definire le sezioni della pagina cui applicare le necessarie regole di stile. Questa non deve essere però una forzatura: una buona suddivisione del documento che tenga conto della sua struttura logica, sarà quasi sicuramente idonea anche all'uso degli stili

51



CSS - Rules

Esempio di posizionamento:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html><head><title>Posizionamento relativo con CSS</title>
<style type="text/css">
body { font-family: "DejaVu Sans", font-size: 14px; }
h1.titolo { color: green; font-style: italic; text-align: center; }
.box_immagini { position: relative; top: 0px; left: 50px; }
.box_fluttua { width: 350px; height: 100px; }
.fluttua { float: left; margin-top: 5px; margin-right: 15px; width: 140px; height: 120px;
}
.riga11 { position: relative; top: 0%; left: 0%; width: 120px; height: 135px; }
.riga12 { position: relative; top: 0%; left: 20%; width: 120px; height: 135px; }
.riga13 { position: relative; top: 0%; left: 40%; width: 180px; height: 135px; }
.riga21 { position: relative; top: 0%; left: 0%; margin-top: 10px; width: 120px; height:
15px; text-align: center; }
.riga22 { position: relative; top: -2em; left: 33%; margin-top: 10px; width: 120px;
height: 15px; text-align: center; }
.riga23 { position: relative; top: -4em; left: 66%; margin-top: 10px; width: 180px;
height: 15px; text-align: center; }
p.testo { width: 300px; height: 100px; overflow: auto; }</style></head>
```

52

CSS - Rules

```
</body>
```

```
<h1 class="titolo">Prova di posizionamento relativo con CSS</h1>
```

```
<p class="testo">
```

Le tre immagini seguenti con le relative didascalie sono posizionate con la proprietà position dei CSS e posizionamento relativo. L'immagine successiva è fluttuante a sinistra ed è inserita dopo la parola "fluttuante".

```
<div class="box_immagini">
```

```
 
```

```

```

```
</div>
```

```
<p class="riga21"> Uno</p>
```

Le ultime due didascalie sono posizionate correttamente in verticale ricorrendo a un posizionamento relativo negativo rispettivamente di -2 e -4 em cioè 2 volte e 4 volte l'ampiezza media del carattere usato; questo le fa allineare alla prima didascalia.

```
<p class="riga22"> due</p> <p class="riga23"> tre</p>
```

```
<div class="box_fluttua">
```

```
<p> Questa immagine è fluttuante, 
```

```
a sinistra; quindi il testo si affianca alla sua destra.</p>
```

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

53

CSS - Rules

Rendering

Posizionamento relativo con CSS - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti 2

file:///D:/didatice/inf13/CAMMINZZAR0/css.html

Posizionamento relativo con CSS

Prova di posizionamento relativo con CSS

Le tre immagini seguenti con le relative didascalie sono posizionate con la proprietà position dei CSS e posizionamento relativo. L'immagine successiva è fluttuante a sinistra ed è inserita dopo la parola "fluttuante".

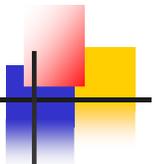
Uno

due

tre

Questa immagine è fluttuante, a sinistra, quindi il testo si affianca alla sua destra.

Completo



CSS - Rules

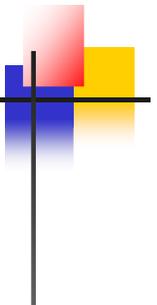
Sintesi delle regole di formattazione:

Proprietà	Significato	Valori	Ere- di- tata	Commento
font-family:	Famiglia di caratteri	<i>Arial</i> , <i>"Helvetica Bold"</i>	Si	Si applica a qualsiasi elemento; nomi multipli devono essere racchiusi tra virgolette; è possibile inserire più valori di famiglie di caratteri separati da «,» per avere una gerarchia di scelta; il valore per difetto dipende dal <i>browser</i> .

55

CSS Rules

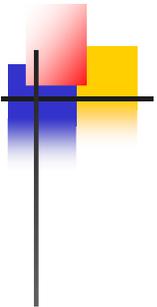
A. Longheu – Linguaggi M-Z – Ing. Inf. 2008-2009



Regole di formattazione

Proprietà	Significato	Valori	Ere- di- tata	Commento
font-size:	Grandezza del carattere	<i>xx-small</i> , <i>x-small</i> , <i>small</i> , <i>medium</i> , <i>large</i> , <i>x-large</i> , <i>xx-large</i> , <i>smaller</i> , <i>larger</i> , dimensione esatta in una delle unità di misura elencate in precedenza, dimensione in percentuale (n%)	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>medium</i> ; nel caso di valore in percentuale, o di valore « <i>em</i> », o di indicazione <i>smaller</i> o <i>larger</i> , viene presa come base la dimensione del carattere dell'elemento genitore; esempi: 15px, 20pt, 75%, 0.75em.
font-style:	Stile del carattere	<i>normal</i> , <i>italic</i> , <i>oblique</i>	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .
font-variant	Manuscoletto	<i>normal</i> , <i>small-caps</i>	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .
font-weight:	Peso del carattere	<i>normal</i> , <i>bold</i> , <i>bolder</i> , <i>lighter</i>	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .
line-height:	Interlinea	<i>normal</i> , valore intero da moltiplicare per la dimensione del carattere, valore in percentuale (n%) sulla dimensione del carattere, valore esatto espresso con una delle unità di misura previste.	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .

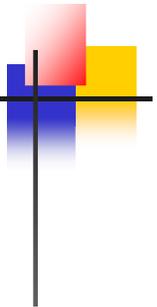
CSS Rules



Regole di formattazione

Proprietà	Significato	Valori	Ere-dita-tà	Commento
background-color:	Colore di sfondo	<i>transparent</i> , colore espresso in esadecimale (preceduto da «#») o con codice mnemonico (red, green,...)	No	Si applica a qualsiasi elemento; il valore per difetto è <i>transparent</i> .
background-image:	Immagine di sfondo	<i>none</i> , URI dell'immagine <i>center</i> , <i>left</i> , <i>right</i> , <i>top</i> , <i>bottom</i> , <i>top left</i> , <i>top right</i> , <i>bottom left</i> , <i>bottom right</i> , valori delle coordinate assolute o in percentuale rispetto all'angolo superiore sinistro	No	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> .
background-position:	Posizione di partenza dell'immagine di sfondo		No	Si applica a elementi di blocco; il valore per difetto è <i>0% 0%</i> .
background-repeat:	Modo di ripetizione dell'immagine di sfondo	<i>no-repeat</i> , <i>repeat</i> , <i>repeat-x</i> , <i>repeat-y</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>no-repeat</i> .
background-attachment:	Per far scorrere l'immagine di sfondo insieme all'elemento di cui è sfondo	<i>scroll</i> , <i>fixed</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>scroll</i> .

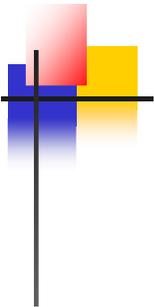
CSS Rules



Regole di formattazione

Proprietà	Significato	Valori	Ere-dita-tà	Commento
color:	Colore del testo	colore espresso in esadecimale (preceduto da «#») o con codice mnemonico (red, green,...)	Si	Si applica a qualsiasi elemento; il valore per difetto dipende dal <i>browser</i> .
text-align:	Allineamento del testo	<i>left</i> , <i>center</i> , <i>right</i> , <i>justify</i>	Si	Si applica a elementi di blocco; il valore per difetto dipende dal <i>browser</i> , di solito è <i>left</i> .
text-decoration:	Effetti «speciali» sul testo	<i>none</i> , <i>underline</i> , <i>overline</i> , <i>line-through</i> , <i>blink</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> .
text-indent:	Indentazione della prima riga	valore o percentuale specificati come per i valori in 'font-size'	Si	Si applica a elementi di blocco; il valore per difetto è <i>0</i> ; il valore in percentuale viene calcolato in base alla larghezza dell'elemento di blocco che contiene l'elemento considerato; si possono inserire anche valori negativi che permettono di avere rientri a sinistra del testo.
text-transform:	Capitalizzazione del testo	<i>none</i> , <i>uppercase</i> , <i>lowercase</i> , <i>capitalize</i>	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> ; con <i>capitalize</i> si fa in modo che la prima lettera di ogni parola sia maiuscola.

CSS Rules

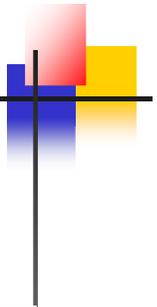


Regole di formattazione

Proprietà	Significato	Valori	Ere-dita	Commento
white-space:	Trattamento degli spazi vuoti	<i>normal</i> , <i>nowrap</i> , <i>pre</i>	No	Si applica a elementi di blocco; il valore per difetto è <i>normal</i> ; con <i>nowrap</i> si evita che gli spazi facciano spezzare una riga; con <i>pre</i> si fa in modo che gli spazi e le righe vuote vengano mantenute e non compatate; con <i>normal</i> si ha il comportamento consueto del programma di navigazione relativamente agli spazi e righe vuote.
word-spacing:	Controllo del <i>tracking</i> (spazio tra le parole)	<i>normal</i> , valore espresso con una delle unità di misura citate in precedenza	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .
letter-spacing:	Controllo del <i>tracking</i> (spazio tra le lettere)	<i>normal</i> , valore espresso con una delle unità di misura citate in precedenza	Si	Si applica a qualsiasi elemento; il valore per difetto è <i>normal</i> .

59

CSS Rules



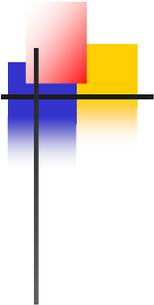
Regole di formattazione:

Proprietà	Significato	Valori	Ere-dita	Commento
vertical-align:	Allineamento verticale del testo	<i>baseline</i> , <i>sub</i> , <i>super</i>	No	Si applica a elementi in linea; il valore per difetto è <i>baseline</i> ; si noti che con <i>sub</i> e <i>super</i> si ottiene la trasformazione del testo in pedice e apice ma senza riduzione della dimensione dei caratteri, per la quale occorre intervenire con la proprietà ' <i>font-size</i> '.

- Nello specificare la proprietà 'font-family' è possibile che si imposti un tipo di carattere non disponibile sul client; in tal caso si può includere il carattere nella pagina:
 - @font-face {font-family: pippo; src:url(pippo.eot);}
 - h1 {font-family: pippo;}
- 'pippo.eot' è il file che deve essere presente nel server.

60

CSS Rules

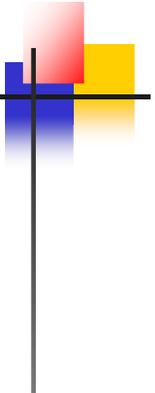


Proprietà	Significato	Valori	Ere- di- tata	Commento
-----------	-------------	--------	---------------------	----------

In sintesi
Regole per il
posizionamento:

<code>width:</code>	Larghezza di un elemento	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a elementi di blocco e rimpiazzati; il valore per difetto è <i>auto</i> ; il valore in percentuale è calcolato in base all'elemento contenitore.
<code>height:</code>	Altezza di un elemento	<i>auto</i> , valore espresso con una delle unità di misura consentite	No	Si applica a elementi di blocco e rimpiazzati; il valore per difetto è <i>auto</i> ;
<code>visibility:</code>	Visibilità di un elemento	<i>visible</i> , <i>hidden</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>visible</i> ; un elemento nascosto risulta comunque presente (cioè influenza comunque il comportamento degli elementi vicini).

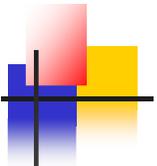
CSS Rules



Regole per il
posizionamento:

Proprietà	Significato	Valori	Ere- di- tata	Commento
<code>border-color:</code>	Colore dei bordi di un riquadro	<i>transparent</i> , codice colore in esadecimale o mnemonico	No	Si applica a qualsiasi elemento; il valore per difetto è il colore dell'elemento contenuto; se si inserisce un solo colore esso vale per tutti e quattro i bordi, altrimenti si possono inserire quattro colori separati da spazi relativi nell'ordine a: bordo alto, bordo destro, bordo basso, bordo sinistro (oppure si possono usare le proprietà specifiche <code>'border-top-color'</code> , <code>'border-right-color'</code> , <code>'border-bottom-color'</code> , <code>'border-left-color'</code>).
<code>border-style:</code>	Stile dei bordi di un riquadro	<i>none</i> , <i>dotted</i> , <i>dashed</i> , <i>solid</i> , <i>double</i> , <i>groove</i> , <i>ridge</i> , <i>inset</i> , <i>outset</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> ; se si inserisce un solo stile esso vale per tutti e quattro i bordi, altrimenti si possono inserire quattro stili separati da spazi relativi nell'ordine a: bordo alto, bordo destro, bordo basso, bordo sinistro (oppure si possono usare le proprietà specifiche <code>'border-top-style'</code> , <code>'border-right-style'</code> , <code>'border-bottom-style'</code> , <code>'border-left-style'</code>).

CSS Rules

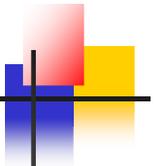


Regole per il posizionamento:

Proprietà	Significato	Valori	Ere-dita	Commento
<code>border-width:</code>	Spessore dei bordi di un riquadro	<i>thin</i> , <i>medium</i> , <i>thick</i> , valore espresso con una delle unità di misura consentite	No	Si applica a qualsiasi elemento; il valore per difetto è <i>medium</i> ; se si inserisce un solo valore o codice esso vale per tutti e quattro i bordi, altrimenti si possono inserire quattro valori o codici separati da spazi relativi nell'ordine a: bordo alto, bordo destro, bordo basso, bordo sinistro (oppure si possono usare le proprietà specifiche <code>'border-top-width'</code> , <code>'border-right-width'</code> , <code>'border-bottom-width'</code> , <code>'border-left-width'</code>).

63

CSS Rules

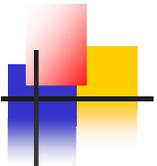


Regole per il posizionamento:

Proprietà	Significato	Valori	Ere-dita	Commento
<code>margin:</code>	Ampiezza dei margini di un riquadro	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento; il valore per difetto è <i>0</i> ; valori in percentuale vengono calcolati in base alla larghezza dell'elemento genitore; se si inserisce un solo valore esso vale per tutti e quattro i margini, altrimenti si possono inserire quattro valori separati da spazi relativi nell'ordine a: margine alto, destro, basso, sinistro (oppure si possono usare le proprietà specifiche <code>'margin-top'</code> , <code>'margin-right'</code> , <code>'margin-bottom'</code> , <code>'margin-left'</code>).

64

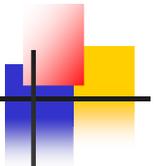
CSS Rules



Regole per il posizionamento:

Proprietà	Significato	Valori	Ere-dita	Commento
padding:	Ampiezza del <i>padding</i> di un riquadro	valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento; il valore per difetto è <i>0</i> ; valori in percentuale vengono calcolati in base alla larghezza dell'elemento genitore; se si inserisce un solo valore esso vale per tutti e quattro i lati, altrimenti si possono inserire quattro valori separati da spazi relativi nell'ordine a: <i>padding</i> alto, destro, basso, sinistro (oppure si possono usare le proprietà specifiche <code>'padding-top'</code> , <code>'padding-right'</code> , <code>'padding-bottom'</code> , <code>'padding-left'</code>).
position:	posizionamento di un riquadro	<i>static</i> , <i>relative</i> , <i>absolute</i> , <i>fixed</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>static</i> .
bottom:	Scostamento in basso di un riquadro posizionato	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento posizionato (con valore di <code>'position'</code> diverso da <i>static</i>); il valore per difetto è <i>auto</i> .
left:	Scostamento a sinistra di un riquadro posizionato	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento posizionato (con valore di <code>'position'</code> diverso da <i>static</i>); il valore per difetto è <i>auto</i> .

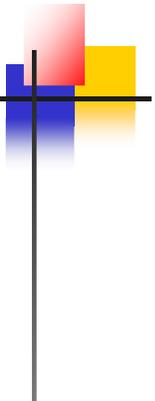
CSS Rules



Regole per il posizionamento:

Proprietà	Significato	Valori	Ere-dita	Commento
right	Scostamento a destra di un riquadro posizionato	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento posizionato (con valore di <code>'position'</code> diverso da <i>static</i>); il valore per difetto è <i>auto</i> .
top:	Scostamento in alto di un riquadro posizionato	<i>auto</i> , valore espresso con una delle unità di misura consentite o in percentuale	No	Si applica a qualsiasi elemento posizionato (con valore di <code>'position'</code> diverso da <i>static</i>); il valore per difetto è <i>auto</i> .
z-index:	Posizionamento sull'asse perpendicolare all'area di visualizzazione	<i>auto</i> , numero intero	No	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> .
float:	Direzione di spostamento di un elemento flottante	<i>none</i> , <i>left</i> , <i>right</i>	No	Si applica a qualsiasi elemento; il valore per difetto è <i>none</i> .
clear:	Controllo scorrimento elementi flottanti	<i>none</i> , <i>left</i> , <i>right</i> , <i>both</i>	No	Si applica a elementi di blocco; il valore per difetto è <i>none</i> ; si usa per impedire che elementi flottanti compiano a sinistra, a destra o da entrambi i lati dell'elemento cui si assegna questa proprietà (taller elemento si posizionerà quindi dopo eventuali elementi flottanti e non accanto ad essi).

CSS Rules



Regole per il posizionamento:

Proprietà	Significato	Valori	Ere- di- tata	Commento
overflow:	Modalità di gestione del traboccamento del testo	<i>visible</i> , <i>hidden</i> , <i>scroll</i> , <i>auto</i>	No	Si applica a elementi di blocco e rimpiazzati; il valore per difetto è <i>visible</i> ; con <i>visible</i> il contenuto che eccede le dimensioni del contenitore viene reso comunque visibile adattando queste ultime, con <i>hidden</i> invece è nascosto, con <i>scroll</i> e <i>auto</i> vengono aggiunte delle barre di scorrimento sul riquadro, nel primo caso sempre, nell'altro solo se necessario.
clip:	Definizione di un'area di taglio	<i>auto</i> , <i>rect</i> (valore_su, valore_dx, valore_giu, valore_sx)	No	Si applica a elementi posizionati in modo assoluto; il valore per difetto è <i>auto</i> ; si usa per definire un'area rettangolare in cui è reso visibile il contenuto del riquadro; il contenuto eccedente viene trattato in base a quanto stabilito con la proprietà <i>'overflow'</i> .

67

CSS - Rules



- Elenchi completi delle regole di formattazione, insieme ad esempi esplicativi, sono presenti numerosi in rete.
- Qualche link di elenchi di regole con i possibili valori:
 - <http://www.w3.org/TR/REC-CSS2/cover.html#minitoc>
 - specifica ufficiale (completa e precisa ma con pochi esempi)
 - <http://www.morpheusweb.it/html/manuali/css.asp>
 - con esempi, in italiano
 - <http://www.ialfrezi.com>
 - datato ma chiaro e conciso (in inglese)
 - http://www.w3schools.com/css/css_examples.asp
 - lista delle possibili regole, ognuna corredata da esempi
 - Link di esempi
 - <http://www.csszengarden.com/>
 - esempi complessi ma graficamente accattivanti di diversi CSS
 - <http://www.dynamicdrive.com/style/>
 - molti effetti già 'pronti da copiare ed utilizzare
 - <http://pmob.co.uk/>
 - molti effetti già 'pronti da copiare ed utilizzare

68

CSS - Link a fogli multipli

Fogli di stile alternativi

- Ad un documento HTML si possono collegare:
- un foglio di stile chiamato «**persistente**», o di base, che viene applicato indipendentemente dalle preferenze degli utenti; tale foglio di stile viene collegato impostando il valore "stylesheet" nell'attributo 'rel' e senza attributo 'title';
- un foglio di stile chiamato «**automatico**» che viene applicato se l'utente non fa alcuna scelta; tale foglio di stile viene collegato impostando il valore "stylesheet" nell'attributo 'rel' e un nome nell'attributo 'title';
- uno o più fogli di stile **alternativi** che vengono collegati impostando per ognuno il valore "alternate stylesheet" nell'attributo 'rel' e un nome univoco nell'attributo 'title'; l'utente può scegliere lo stile agendo sugli appositi menu disponibili nei browser; lo stile persistente si somma sempre a qualsiasi altro

69

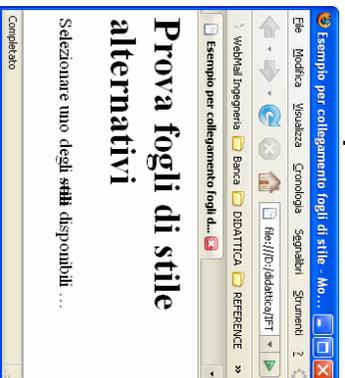
CSS - Link a fogli multipli

Esempio:

```
<html>
<head><title>Esempio per collegamento fogli di stile</title>
<link rel="stylesheet" href="foglio_pers.css" type="text/css"/>
<link rel="stylesheet" href="foglio_auto.css" type="text/css" title="stile1"/>
<link rel="alternate stylesheet" href="foglio1.css" type="text/css"
title="stile2"/>
<link rel="alternate stylesheet" href="foglio2.css" type="text/css"
title="stile3"/>
</head>
<body>
<h1>Prova fogli di stile alternativi</h1>
<p>Selezionare uno degli <b>stili</b> disponibili ...</p></body></html>
```

CSS - Link a fogli multipli

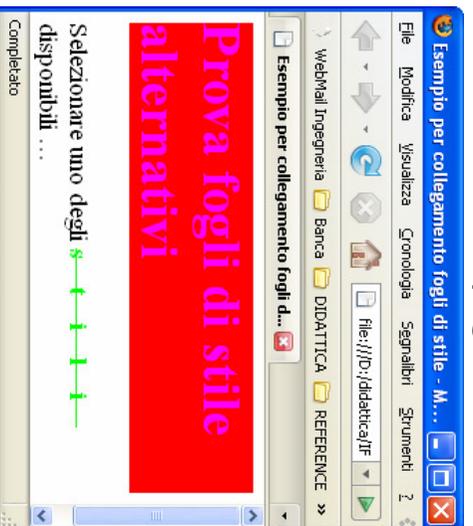
- Visualizzazione pagina senza stile, con il solo persistente, con persistente + default



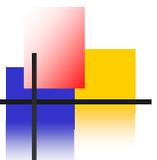
71

CSS - Link a fogli multipli

- Visualizzazione pagina con stile2 o stile3



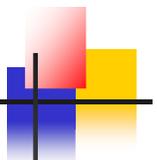
72



CSS - Tabelle

- Alcune proprietà dei CSS sono destinate alla modifica stilistica delle tabelle. Il supporto, chiariamolo subito, non è uniforme per tutti i browser. Mozilla brilla anche qui per la sua fedeltà agli standard, mentre Explorer Windows cade su parecchi aspetti. Nell'analisi delle proprietà ci limiteremo all'enumerazione dei concetti di base, tralasciando la miriade di regole particolari e spesso complesse che ne governano il comportamento. Del resto, l'uso di queste proprietà è davvero molto ridotto, dal momento che gli stessi risultati possono essere ottenuti in diversi modi, anche con i tradizionali sistemi incorporati in (X)HTML.
- **table-layout** imposta il metodo di layout di una tabella. Non è ereditata. Si applica solo alle tabelle. Valori: auto (Il layout della tabella viene definito automaticamente dal browser) o fixed (le regole di presentazione sono quelle impostate dall'autore nel CSS)

73



CSS - Tabelle

- **border-collapse** stabilisce in che modo trattare i bordi e gli spazi tra le celle di una tabella. Si applica solo alle tabelle ed è ereditata. Valori: collapse (se viene impostato un bordo, le celle della tabella lo condividono), o separate (ogni cella ha il suo, separato dalle altre). Lo spazio tra le celle e tra i bordi si imposta con la proprietà border-spacing.
- **border-spacing** imposta lo spazio tra le celle di una tabella (valore numerico con unità di misura). Va usata solo se border-collapse è settato su separate. Proprietà ereditata e applicabile solo alle tabelle.
- **empty-cells** gestisce il trattamento delle celle di tabella senza contenuto. Agisce solo su quelle che non presentino al loro interno alcun tipo di markup, nemmeno il classico inserito in genere proprio per simulare la presenza di contenuto. Valori: show (mostra i bordi della cella) o hide (i bordi non vengono mostrati e apparirà solo uno spazio vuoto)
- **caption-side** definisce il lato su cui vogliamo far comparire la didascalia. È ereditata. La supporta solo Mozilla. Valori: top, bottom, right, left

74

CSS - Cursori

- cursor Definisce l'aspetto del puntatore del mouse quando passa su un elemento. Si applica a tutti gli elementi.
- Sintassi
`<selettore> {cursor: <valore>;}`
- Valori I valori possibili sono molti ed è possibile praticamente usare tutti i tipi di cursore abituali dell'interfaccia utente: auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help

75

CSS - Media

- Dove prima c'era unicamente un browser, domani potranno sempre più esserci palmari, cellulari e altri dispositivi. Per non parlare dei software usati da disabili come i browser vocali. Ciascuno di questi supporti presenta caratteristiche diverse in termini di memoria, ampiezza dello schermo e funzionalità. Riuscire ad adattare uno stile unico a tutti è praticamente impossibile, oltre che controproducente. La soluzione ideale sta quindi nella creazione di fogli di stile ad hoc selezionati tramite l'attributo media:
- Sintassi: `media="<valore>"`
- L'attributo media può accompagnare sia l'elemento <LINK> che l'elemento <STYLE>

76

CSS - Media

- Per sfruttare a fondo questa opzione è fondamentale conoscere i diversi valori possibili per l'attributo:
- **all**. Il CSS si applica a tutti i dispositivi di visualizzazione.
- **screen**. Valore usato per la resa sui normali browser web.
- **print**. Il CSS viene applicato in fase di stampa del documento.
- **projection**. Usato per presentazioni e proiezioni a tutto schermo.
- **aural**. Da usare per dispositivi come browser a sintesi vocale.
- **braille**. Il CSS viene usato per supporti basati sull'uso del braille.
- **embossed**. Per stampanti braille.
- **handheld**. Palmari e simili.
- **tty**. Dispositivi a carattere fisso.
- **tv**. Web-tv.
- Il valore di default è **all**, usato automaticamente in mancanza di una dichiarazione esplicita (ricordiamo infatti che **media** è un attributo opzionale).

77

CSS - Media

- E' possibile usare più di un valore, ma i nomi della lista vanno separati da una virgola:


```
<link rel="stylesheet" type="text/css" media="print, tv, aural" href="print.css"/>
```
- L'uso più tipico di questa funzionalità consiste nel collegare al documento vari fogli di stile adatti a ciascun supporto. Lo user agent che visualizzerà la pagina caricherà quello giusto:


```
<link rel="stylesheet" type="text/css" media="screen" href="screen.css" />
<link rel="stylesheet" type="text/css" media="print, tv, aural" href="print.css"/>
```
- Un'applicazione immediatamente fruibile, a patto di non usare browser di quarta generazione, è quella che prevede l'uso di un foglio di stile per la stampa. Le motivazioni per una simile soluzione possono essere molteplici. L'uso di caratteri o colori particolari, la presenza nella pagina di elementi che vogliamo non far comparire sulla carta, può portare alla necessità di adottare un CSS speciale.

78

CSS - Stampa

- Nella specifica CSS2 sono indicati gli stili destinati alla formattazione di pagine stampate. L'implementazione di queste proprietà è però quasi nulla su tutti i browser.
- Sarà quindi in futuro possibile definire nei dettagli le modalità di formattazione della pagina in vista della stampa, a partire dalle dimensioni per finire all'orientamento della stessa: proprio come si fa in un comune word-processor. Con la direttiva **@page**, ad esempio, si può impostare la regola:
 - `@page {size: 210mm 297mm; margin: 30mm;}`
- Il layout della pagina in fase di stampa sarà quello di un normale foglio A4 con margini di 3cm su tutti i lati.

79

CSS - Stampa

- La parte preponderante della sezione è dedicata alla gestione delle interruzioni di pagina in fase di stampa. Capita infatti spesso che provando a stampare una pagina web una tabella si divida tra due pagine. Per risolvere il problema sono state implementate proprietà ad hoc, il cui obiettivo è promuovere (W3C) comportamenti corretti:
 - non interrompere una pagina in corrispondenza di blocchi con bordi
 - non interrompere una pagina nel mezzo di una tabella
 - non interrompere una pagina in corrispondenza di elementi float
- Per questi aspetti esistono le proprietà: **page-break-after**, **page-break-before**, **page-break-inside**
- Grazie ad esse è possibile con esattezza e per ciascun elemento quando e dove inserire un'interruzione di pagina. Ad esempio, per evitare interruzioni nel corpo della tabella si scrive:


```
table {page-break-inside: avoid;}
```
- Se invece vi sono una serie di sezioni di sommario alla fine delle quali si desidera sempre un'interruzione si scrive:


```
div.sommario {page-break-after: always;}
```
- **page-break-after** e **page-break-before** sono le uniche due proprietà supportate decentemente da tutti i browser. tra questi quello più performante su questo aspetto è Opera.

80

CSS - Altri media

- Al W3C sono attenti ai disabili e all'accessibilità dei siti Web e pertanto l'ultima sezione della specifica CSS2 è dedicata agli stili audio. I dispositivi interessati sono innanzitutto i browser vocali usati dai non vedenti. Nessuno dei tradizionali browser web supporta queste proprietà.
- Il loro uso è destinato a modificare le caratteristiche audio delle sezioni ed elementi di una pagina. Definiscono quindi l'intensità, il volume, la durata delle pause tra parole o sezioni del documento, la velocità del parlato, etc.
- Il loro uso è in tutto simile a quello delle normali proprietà visuali, ad esempio con la regola seguente si alza il tono del volume in coincidenza di un titolo H1:
h1 {volume: loud;}
- Oppure si può impostare una pausa al termine di ogni paragrafo:
p {pause-after: 300ms;}

81

CSS - Compatibilità

- I problemi di compatibilità dei CSS sui diversi browser sono ancora oggi significativi. Il problema è che la piena coscienza da parte di produttori di browser e autori dell'importanza di adottare linguaggi standard ha fatto molta fatica ad affermarsi. Netscape introduceva estensioni proprietarie, Explorer rispondeva. E i poveri web-designer a cercare soluzioni in grado di conciliare gli opposti.
- Una prima svolta, sul versante dei browser di massa, si ebbe con Explorer 5, il primo ad offrire un supporto adeguato del linguaggio. Oggi possiamo contare su strumenti di navigazione che supportano la quasi totalità della specifica CSS2. Con un browser, però, che surclassa tutti gli altri in questo ambito: Mozilla. Microsoft ha continuato la sua opera di adeguamento, deludendo un pò con Explorer 6
- Nonostante tutto la via dei CSS è ancora lastricata di tanti problemi per gli sviluppatori. Due le maggiori fonti di difficoltà:
 - la compatibilità con i vecchi browser; problema che, comunque, richiede di costruire fogli di stile distinti (la rilevazione del tipo di browser si può fare con JavaScript)
 - le diverse modalità di rendering di certe proprietà (quindi occorre testare su tutti i browser)

82

CSS - Software e validatori

- I principali vantaggi che strumenti software dedicati possono offrire sono tre:
 - **facilitazione nella scrittura del codice.** Può avvenire con la colorazione differenziata delle varie parti o con funzionalità di autocompletamento avanzate.
 - **anteprima visuale.** Avere, cioè, la possibilità, di visualizzare in una finestra di preview gli effetti di una regola.
 - **wizard e templates**
- Fra i software:
 - **TopStyle CSS & XHTML Editor** (<http://www.bradsoft.com/topstyle/>)
 - **StyleMaster PRO** (http://www.westciv.com/style_master/index.html)
 - **Dreamweaver** (<http://www.macromedia.com>)
- Come avviene già da tempo per (X)HTML, il W3C ha predisposto sul suo sito un validatore di fogli di stile CSS:
 - <http://jigsaw.w3.org/css-validator/>

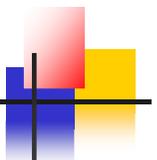
83

CSS - @rules

- Le cosiddette @-rules sono tipi particolari di costrutti che hanno una caratteristica comune: sono tutti introdotti dal simbolo della chiocciola. La pronuncia è identica a quella usata per gli indirizzi e-mail, ovvero at-import, at-media
- **@import** viene usata innanzitutto per collegare un foglio di stile esterno al documento
- Lo stesso risultato (un CSS per ogni dispositivo) si può ottenere con la direttiva **@media**. @media va seguito dal nome di uno dei supporti scelti e dalle regole di stile, ad esempio


```
@media screen
{
h1 {color: black;} }
```
- La direttiva **@charset** serve a specificare l'impostazione relativa alla codifica dei caratteri di un documento. Svolge in pratica la stessa funzione che in (X)HTML è possibile ottenere con l'uso dei meta-tag

84



CSS - Link

Numerosi i siti che trattano CSS, proponendo anche layout. Alcuni indirizzi:

- http://www.templateworld.com/free_templates.html
- diversi template di siti, scaricabili e modificabili
- <http://www.opencube.com/download.asp>
- diversi software per la creazione di menu CSS, con o senza Javascript
- una raccolta di effetti web
- <http://www.cssbeauty.com/>
- galleria di siti realizzati con fogli di stile e javascript
- <http://meyerweb.com/eric/css/edge/>
- ricco di esempi e di link
- <http://www.richinstyle.com/free/>
- qualche link
- <http://www.websitetips.com/css/index.shtml>
- guida per lo sviluppo di siti
- <http://alistapart.com/>
- rivista che approfondisce temi legati allo sviluppo dei siti web