

## Fogli di stile

- I fogli di stile permettono la separazione della forma dal contenuto.
- Il concetto di foglio di stile è quello di creare un documento separato che contenga tutte le informazioni necessarie per la formattazione o la trasformazione di qualsiasi tipo di dati a cui si decida di applicarlo.

Andres Baravalle – Vitaveska Lanfranchi Dipartimento di Informatica

## Fogli di stile (2)

- Vantaggi:
  - Risparmio di tempo e maggior praticità di gestione del sito (se si desidera infatti modificare l'aspetto di un sito *web* non è necessario effettuare le correzioni in ogni pagina ma è sufficiente modificare il foglio di stile per rendere effettivi i cambiamenti).
  - Personalizzazione grafica, in quanto è l'aspetto degli elementi è modificabile.
  - Utilizzo di un unico foglio di stile per formattare più documenti.
  - Possibilità di adattare la formattazione al tipo di utente e *device* che ha accesso ai dati.

Andres Baravalle – Vitaveska Lanfranchi Dipartimento di Informatica

## Fogli di stile (3)

- I fogli di stile oggi utilizzati sono essenzialmente di tre tipi:
  - **CSS** (*Cascading Style Sheet*), utilizzati soprattutto per HTML.
  - **XSL** (*Extensible Stylesheet Language*), utilizzati per formattare un documento XML destinato alla stampa.
  - **XSLT** (*Extensible Stylesheet Language Transformation*), utilizzati per la trasformazione di un documento XML.

Andres Baravalle – Vitaveska Lanfranchi Dipartimento di Informatica

## Introduzione ai CSS

- I CSS (*Cascading Style Sheets*) sono i fogli di stile più comunemente utilizzati per formattare pagine HTML o XHTML.
- Sono stati inventati da Bert Bos e Hakon Lie del W3C.
- La prima versione, rilasciata nel 1996 dal W3C conteneva specifiche relative all'apparenza dei font, ai colori, al *background* di un documento.
- La seconda versione (CSS2), risalente al 1998, incorpora funzioni più avanzate per controllare il *layout* della pagina, la stampa, il supporto per lo scaricamento dei *font*.

Andres Baravalle – Vitaveska Lanfranchi Dipartimento di Informatica

## Regole

- I fogli di stile CSS definiscono **regole di visualizzazione**.
- Ogni regola è costituita da un elenco di proprietà, ciascuna formata da due parti proprietà:valore
- Le regole vengono associate ai tag

```
tag { proprietà1:valore1;
    proprietà2:valore2; ... }
```

Andres Baravalle – Vitaveska Lanfranchi Dipartimento di Informatica

## Differenze con HTML

- Gli attributi sono inseriti tra parentesi graffe { }
- Al posto del segno = 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:
  - p {font-family: "sans serif"}
  - P {font-family: "san serif", verdana}

## Sili interno o esterni?

- I fogli di stile possono essere posizionati:
  - all'interno del documento:

```
<head>
  <style type="text/css">
    H1 {font-size:medium; color:#FF00FF}
  </style>
</head>
```
  - all'interno di un singolo tag:

```
<h1 style="color:red; text-
transform:capitalize;">
  Prova CSS per H1
</h1>
```
  - in un file esterno con estensione .css.

## Stili esterni

- 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. Nel nostro caso: rel="stylesheet"
  - type, che indica il tipo di foglio di stile. Nel nostro caso type="text/css"
  - href, che indica l'URL del file .css.
- Esempio:

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

## File CSS

- Un file CSS avrà la sintassi:

```
body {background-color:white; color: black; font-
family: Verdana, Arial, Helvetica; font-size: x-
small; font-style: normal; text-align: left}
a:link {background-color:white; color:#000066;
font-family: Verdana, Arial, Helvetica; font-
size: small; font-style: normal; text-
decoration: none; font-weight: bold}
h1 {font-family:Verdana, Arial, Helvetica; text-
align: center; font-size:large; color:#000000;;
font-style: normal}
h2 {font-family: "Dungeon", Arial; color:#000000;
text-align: left; font-size:medium; }
ul {list-style-image: url(Media/Ereccia.gif)}
```

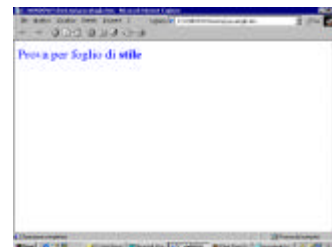
## Ereditarietà

- Il principio base di CSS è l'ereditarietà delle proprietà.
- Ogni elemento innestato in un altro eredita le proprietà dell'elemento che gli è padre.
- Esempio:
  - se un paragrafo è scritto in carattere "Times New Roman", colore blu, una parola in grassetto al suo interno resterà comunque sempre blu e in "Times New Roman", oltre ad essere in grassetto.
  - Nel codice CSS:

```
P {color:blue, font-family:Times New Roman; font-
size:200%}
```
  - Nel codice HTML

```
<p>Prova per foglio di
<b>stile </b>
</p>
```

## Esempio di ereditarietà



## Ordine di ereditarietà

- 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 è:
  1. Browser default
  2. External Style Sheet
  3. Internal Style Sheet (all'interno del tag <head>)
  4. Inline Style (all'interno dell'elemento HTML)

## Definizione di regola

- Per definire una regola la si associa a uno o più marcatori presenti all'interno del codice HTML.
- `p { text-align: center; color: black; font-family: arial }`
- Nel caso i tag a cui si applica la regola siano più di uno devono essere separati da una virgola.
- `h1,h2,h3,h4,h5,h6 { color: green }`

## Classi

- Le classi servono a definire degli stili astratti che possono:
  - essere associati a diversi marcatori all'interno di un documento
  - definire diversi stili per un unico marcatore.
- Quando si crea una classe si usa la sintassi
  - `.nomeclasse`
  - `.testocentrato {text-align: center}`
  - `Nomemarcatore.nomeclasse`
  - `p.testodestra {text-align: right}`
  - `p.testosinistra {text-align: left}`

## Utilizzare una classe

- Quando una classe è stata definita in un foglio di stile può essere richiamata utilizzando l'attributo `class` all'interno di un marcatore, avente come valore il nome della classe che si desidera utilizzare.  
`<p class="testodestra"> allineato a destra. </p>`  
`<p class="testocentrato"> allineato al centro. </p>`

## Background

- Per impostare un colore di background si utilizza il comando `background-color` che può avere come valori:
  - Un codice esadecimale
  - Un nome di colore
- Per impostare un'immagine di sfondo si utilizza l'elemento `background-image: url(stars.gif)` che ha come valore il pathname di un'immagine.

## Background (2)

- `background-position` permette di definire la posizione di un'immagine di background.
- Può avere come valori:
  - top left
  - top center
  - top right
  - center left
  - center center
  - center right
  - bottom left
  - bottom center
  - bottom right
  - x-% y-%

### Background (3)

- **background-repeat** permette di impostare un'eventuale ripetizione di un'immagine di sfondo.
- Ha come valori:
  - repeat: ripetizione
  - repeat-x: ripetizione orizzontale
  - repeat-y: ripetizione verticale
  - no-repeat: nessuna ripetizione

### Background (4)

- **background-attachment** indica se il background (un'immagine) si sposti o meno in relazione allo scrolling della pagina.
- Ha come valori:
  - scroll
  - fixed

### Testo

- **Color** permette di settare il colore di un testo.
- Ha come valori:
  - Un codice esadecimale
  - Un nome di colore
- **text-align** permette di definire l'allineamento di un testo.
- Ha come valori:
  - left: sinistra
  - right: destra
  - center: centrato
  - justify: giustificato (è l'unico modo standard di creare un testo giustificato!)

### Testo (2)

- **text-decoration** permette di stabilire un'eventuale decorazione per un testo.
- Ha come valori:
  - None: nessuna
  - Underline: sottolineato
  - Overline: linea sopra
  - line-through: linea sopra il testo
  - Blink: lampeggiante
- **text-indent** permette di fissare un indentamento per un testo.
- Ha come valore una lunghezza in pixel o percentuale (anche negativa).

### Testo (3)

- **text-transform** permette di trasformare il testo.
- Ha come valori:
  - none
  - capitalize: parola con prima lettera maiuscola
  - uppercase: lettere tutte maiuscole
  - lowercase: lettere tutte minuscole
- **word-spacing** permette di settare uno spazio normale o esteso tra le lettere.
- Ha come valori:
  - normal
  - length: lunghezza in pixel o percentuale

### Caratteri

- **font-family** permette di stabilire quale famiglia di font si vuole utilizzare.
- Generalmente ha come valore una lista di tre famiglie:  
P{font-family: arial, "lucida console", sans-serif}

## Caratteri (2)

- **font-size** permette di definire la grandezza del carattere:
- Ha come valori:
  - xx-small
  - x-small
  - small
  - medium
  - large
  - x-large
  - xx-large
  - smaller
  - larger
  - *length* (grandezza fissa in punti)
  - % (grandezza in percentuale)

## Caratteri (3)

- **font-style** permette di definire lo stile di un carattere.
- Ha come valori:
  - normal
  - italic
  - oblique
- **font-weight** permette di definire le dimensioni di un carattere.
- Ha come valori:
  - normal
  - bold
  - bolder
  - lighter
  - 100-800

## Liste

- **list-style-image** permette di definire un marcatore (immagine) per una lista.
- Ha come valore il pathname dell'immagine.  
list-style-image: url(blueball.gif);
- **list-style-position** permette di definire la posizione degli elementi della lista.
- Ha come valori:
  - inside: lista indentata
  - outside: lista non indentata

## Liste (2)

- **list-style-type** permette di definire il tipo di marcatore che si vuole utilizzare:
- Ha come valori, tra gli altri:
  - disc
  - circle
  - square
  - decimal
  - decimal-leading-zero
  - lower-roman
  - upper-roman
  - lower-alpha
  - upper-alpha
  - lower-greek
  - lower-latin
  - upper-latin

## Bordi

- **border-color** permette di definire un colore per il bordo.
- Ha come valori:
  - Un codice esadecimale
  - Un nome di colore.
- **border-style** permette di definire lo stile di un bordo.
- Ha come valori:
  - hidden
  - dotted
  - dashed
  - solid
  - double
  - groove
  - ridge
  - inset
  - outset

## Bordi (2)

- **border-width** permette di definire la grandezza del bordo.
- Ha come valori:
  - thin
  - medium
  - thick
  - *length*

## Margini

- `margin-bottom`, `margin-top`, `margin-left`, `margin-right` permettono di definire le dimensioni dei margini.
- Hanno come valore:
  - auto (dimensione automatica fissata dal browser)
  - `length`
  - %

## Padding

- `padding-bottom`, `padding-left`, `padding-right`, `padding-top` permettono di definire lo spazio tra il bordo e il contenuto di un elemento.
- Hanno come valori:
  - `length`
  - %

## Dimensioni

- `Height` permette di definire l'altezza di un elemento.
- Ha come valori:
  - auto
  - `length`
  - %
- `width` permette di definire la larghezza di un elemento.
- Ha come valori:
  - auto
  - `length`
  - %

## Visualizzazione

- `Cursor` specifica il tipo di cursore da utilizzare.
- Ha come valori:
  - `url`
  - Auto
  - Crosshair
  - Default
  - Pointer
  - Move
  - e-resize
  - ne-resize
  - nw-resize
  - n-resize
  - se-resize
  - sw-resize
  - s-resize
  - w-resize

## Visualizzazione (2)

- `Display` definisce come un elemento verrà visualizzato.
- Ha come valori, tra gli altri:
  - none
  - inline
  - block
  - list-item
  - run-in
  - compact
  - marker
  - table

## Visualizzazione (3)

- `Position` fissa la posizione di un elemento.
- Ha come valori:
  - static
  - relative
  - absolute
  - fixed

## Allineamento

- `vertical-align` permette di definire l'allineamento verticale di un elemento.
- Ha come valori:
  - `baseline`
  - `sub`
  - `super`
  - `top`
  - `text-top`
  - `middle`
  - `bottom`
  - `text-bottom`
  - `length`
  - `%`

## Link

- Sono presenti alcune pseudoclassi per definire particolari visualizzazioni dei link.
- Link non visitati: `a:link {color: #FF0000}`
- Link visitati: `a:visited {color: #00FF00}`
- Mouse sopra un link: `a:hover {color: #FF00FF}`
- Link selezionato: `a:active {color: #0000FF}`
- L'ordine delle pseudoclassi deve essere rispettato perché le opzioni abbiano effetto.

## Altre pseudoclassi

- `:first-child`, applica un particolare stile al primo elemento figlio di un altro.
- `a:first-child { text-decoration:none }`

```
<p> Visit
  <a href="http://www.w3schools.com">W3Schools</a> and learn
  CSS! Visit
  <a href="http://www.w3schools.com">W3Schools</a>
  and learn HTML!
</p>
```

## Altre pseudoclassi (2)

- `:first-line` applica una particolare formattazione alla prima riga di un testo.
- `:first-letter` applica una particolare formattazione alla prima lettera di una parola

## Livelli

- Usare i fogli di stile permette di definire livelli nelle pagine.
- I livelli sono introdotti in HTML dal tag `<div>`
- L'utilizzo dei livelli permette di:
  - porre blocchi di informazioni in un determinato punto della pagina
  - Porre blocchi di informazioni davanti o dietro ad altri

## Livelli (2)

- I principali attributi utilizzabili per definire i livelli:
  - `Width`: larghezza
  - `Height`: altezza
  - `Left`: distanza da sinistra
  - `Right`: distanza da destra
  - `Bottom`: distanza dal margine inferiore
  - `Top`: distanza dal margine superiore
  - `Position` (relative | absolute): se il posizionamento dipende dalla posizione in cui è inserito il codice o dall'origine della pagina.
  - `Z-index`: posizione del livello rispetto agli altri

## Livelli (3)

### ◀ Esempi di livelli:

- ◀ `<div id="right" style="position: absolute; left: 500px; top: 95px; width: 150px; height: 450px; z-index: 2"></div>`
- ◀ `<div id="right" style="position: relative; top: -12px; color: red"></div>`