

# Cascading Style Sheets (CSS)

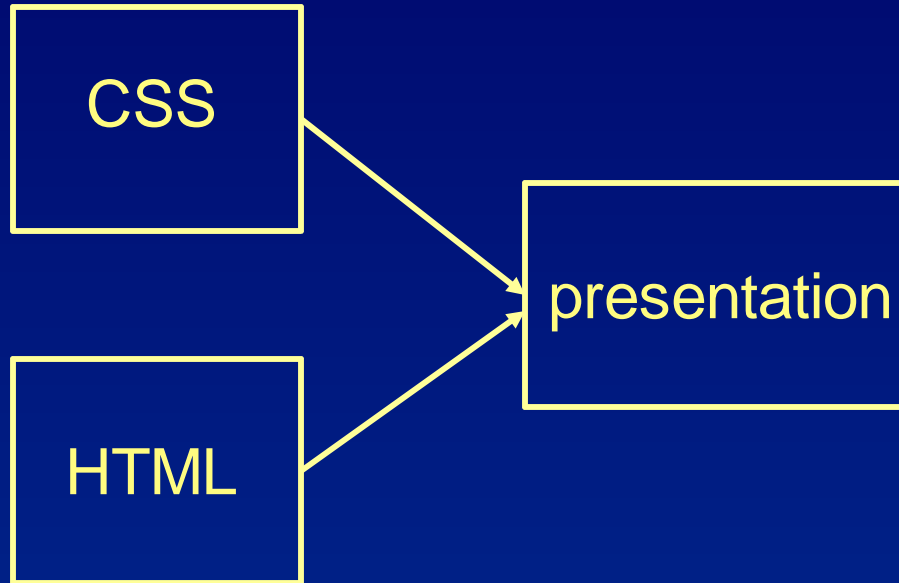
Autore slide: Stefano Marchetti

# Introduzione

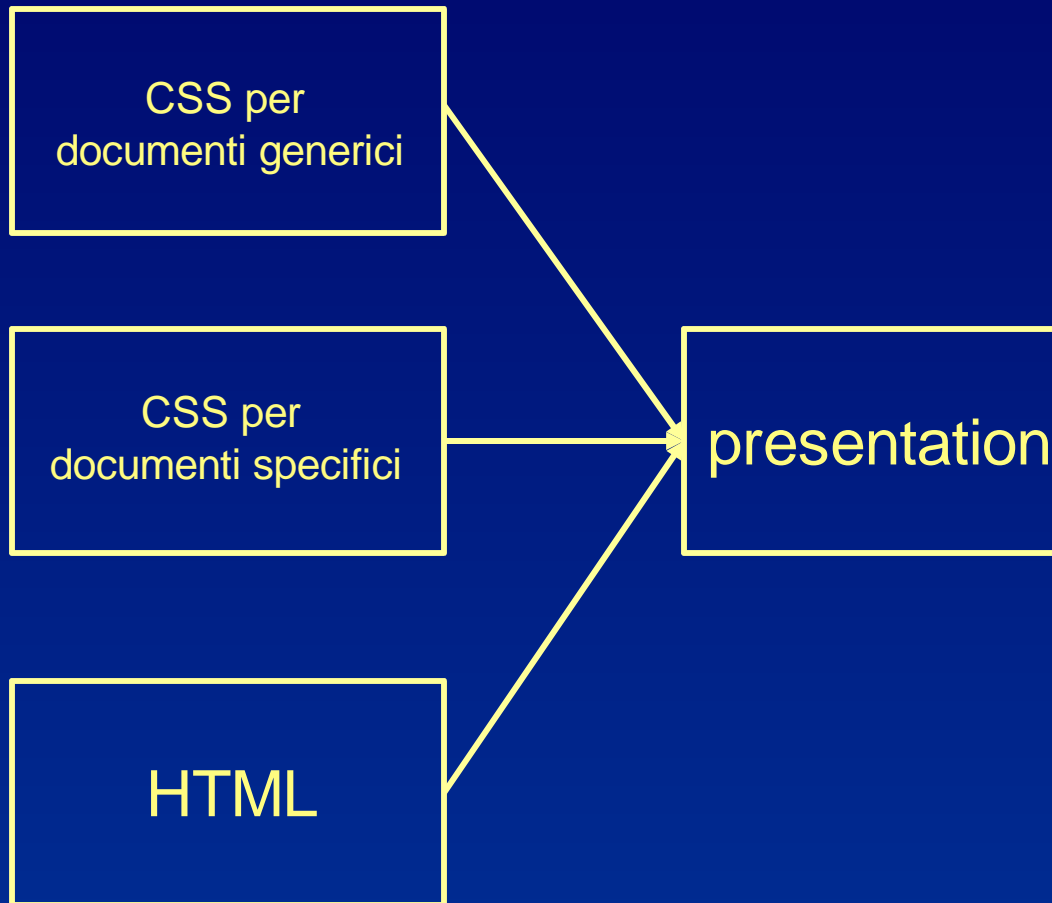
Oggi esaminiamo in breve:

- Principi di CSS
- Meccanismi per definire CSS

# HTML + CSS = Presentazione



# HTML + CSS = Presentazione



# CSS nelle pagine HTML

Esistono 3 meccanismi per utilizzare i fogli di stile in una pagina HTML:

- Usare l'elemento STYLE; non esiste una separazione fisica tra contenuto del documento e presentazione

```
<STYLE TYPE="text/css">
```

```
  H1 { color: green }
```

```
</STYLE>
```

- Usare l'elemento LINK; il documento contiene solo il riferimento al foglio di stile. Il foglio di stile può essere usato per più documenti

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="...">
```

- E' possibile combinare entrambi i metodi.

# Cascading Style Sheets level 1 (CSS1)

Un foglio di stile è composto da una serie di *regole*.  
Ogni regola ha la forma:

*selettore dichiarazione*

Dove *dichiarazione* ha la forma:

*{ proprietà:valore }*

Esempio:

P {color:green}

[Si veda 1.html](#)

# Ereditarietà

Un documento HTML può essere visto come un albero:

Si veda schema a pag. 273

# Selettori (1)

Il selettore è utilizzato per limitare lo scope di una dichiarazione (che contiene informazioni su come formattare un elemento)

# Selettori – type selector (2)

## *Type selectors*

Un type selector seleziona un elemento in accordo con il suo tipo

P {color:green}

Il selettore P indica che la dichiarazione deve essere applicata ad ogni istanza di P

H1 { color: blue }

H2 { color: blue }

H3 { color: blue }

H1,H2,H3 { color: blue }

[Si veda 2.html](#)

# Selettori – attributo CLASS (3)

## CLASS

L'attributo CLASS può essere usato per assegnare una “classe” ad un elemento (possono essere specificate più classi per un elemento).

```
<P CLASS="IMPORTANTE">Contenuto del paragrafo ...
```

HTML non assegna nessuna semantica alla classe. Se ci sono molti documenti è meglio scegliere le classi da usare.

```
.IMPORTANTE { color : red }
```

Notare le differenze rispetto ai selettori *type selector*.

[Si veda 3.html](#)

# Selettori – attributo ID (4)

## *ID*

L'attributo ID può essere usato per assegnare un identificatore unico ad un elemento.

```
<P ID="IMPORTANTE">Contenuto del paragrafo ...
```

L'elemento ID può essere usato come destinazione di un link p.e.  

```
<A HREF="#IMPORTANTE"> ... </A>.
```

```
#IMPORTANTE { color : red }
```

Notare le differenze rispetto ai selettori *type selector*.

# Selettori – attributo STYLE (5)

## *STYLE*

Non è un selettore quindi non dovrebbe essere nella spiegazione dei selettori. E' qui perché è un meccanismo per specificare lo stile di un elemento. A differenza di CLASS e ID l'attributo STYLE è usato per dichiarare direttamente la formattazione di un elemento.

`<P STYLE="font-style:italic">Contenuto del paragrafo ...`

Con questo approccio non c'è una separazione netta tra contenuto e presentazione. E' meglio l'elemento STYLE (non l'attributo) all'inizio del documento (quindi in un unico punto) o un foglio di stile esterno.

# Selettori – selettori contestuali (6)

Quello visto fino ad adesso sono *selettori semplici*: si specifica un elemento per i type selector o il valore di un attributo.

Un meccanismo più sofisticato è dato dai *selettori contestuali*.

I selettori contestuali usano selettori semplici per specificare relazioni di contesto.

Esempi di selettore contestuale:

LI P { color : red } uso 2 type-selector

TABLE .IMPORTANTE P { color : red } uso di type selector e attribute selector

[Si veda 4.html](#)

# Selettori – selettori esterni (7)

In alcuni casi la selezione del contenuto deve essere basata su informazioni esterne cioè su informazioni che non fanno parte della pagina HTML.

CSS1 definisce 2 tipi di informazioni esterne:

- *Pseudo-classes*: è una proprietà di un elemento che è determinato da informazioni esterne per esempio lo stato di un link.

CSS1 definisce 3 stati per l'elemento A.

- *link*: link non ancora visitato
- *visited*: link già visitato
- *active*: link attivo, in corso di attraversamento

Esempio:

```
A:LINK { color : green }
```

```
A:ACTIVE { color : red }
```

```
A:VISITED { color : blue }
```

- *Pseudo-elements*: CONTINUA -->

# Selettori – selettori esterni (8)

*Pseudo-elements*: è contenuto che è identificato da informazioni esterne e che non è esplicitamente definito da un elemento HTML come per esempio il primo carattere di un paragrafo. Per esempio la prima riga di un paragrafo cambia quando la larghezza della finestra del browser cambia.

CSS1 definisce 2 pseudo-elements, entrambi applicabili all'elemento P.

first-letter

first-line

Esempio

```
P.IMPORTANTE:FIRST-LETTER { font-size: 200%; float:left }
```

```
P.IMPORTANTE:FIRST-LETTER { text-trasformation:uppercase }
```

Type selector + Attribute selector + Pseudo element

[Si veda 5.html](#)

# Selettori – nuovi elementi (9)

L'elemento CLASS viene usato per definire la formattazione di elementi che hanno già un certo tipo di semantica per la formattazione.

Per evitare questo HTML definisce 2 elementi: DIV e SPAN

## DIV

È un element block-level usato per evidenziare un blocco di testo. Usato senza fogli di stile non cambia la formattazione del contenuto. Usato con CLASS o ID consente di specificare la formattazione di un blocco di contenuto.

## SPAN

E' un elemento inline e come per l'elemento DIV se usato senza stili non cambia la formattazione del testo.

# Selettori – combinare diversi tipi di selettori (10)

E' possibile combinare diversi tipi di selettori.

```
TABLE P.INITIAL:FIRST-LETTER { font-weight: bold }
```

# Unità - colori

- E' possibile specificare il nome scegliendo tra 16 nomi (come in HTML)
- E' possibile specificare una tripla RGB p.e. #FF0000
- E' possibile specificare il valore in forma percentuale p.e. rgb(100%,0%,0%)
- E' possibile specificare il valore con un numero decimale p.e. rgb(255,0,0)

# Unità - lunghezze

Esistono 2 tipi di lunghezze:

- Lunghezze misurate in relazione a qualcosa d'altro (per esempio rispetto al font o al display)
  - ★ em: specifica per il font, larghezza della lettera 'M'
  - ★ ex: specifica per il font, altezza della lettera 'x'
  - ★ px: specifica del display, dimensione di un pixel
- Lunghezze assolute; non dipendono dalla presentazione
  - ★ in – inches (2,54cm)
  - ★ cm – centimetri
  - ★ mm – millimetri
  - ★ pt – points (1pt = 1/72 in)
  - ★ pc – picas (1pc = 12 pt)

Per mantenere la scalabilità del documento è preferibile usare lunghezze relative.

E' possibile usare anche valori in percentuale p.e. font-size:200%;

# Dichiarazioni

L'uso dei selettori è solo il prerequisito per il vero lavoro da fare e cioè la definizione delle istruzioni per la formattazione.

# Dichiarazioni - font

`font-family`

si può indicare il nome specifico della famiglia del font p.e. Times o usare il nome generico della famiglia del font p.e. serif, sans-serif, monospace, cursive e fantasy. Si usa una lista e viene usato il primo font disponibile sul browser. E' buona norma indicare più valori.

`font-style`

Per alcune famiglie di font esistono diverse stili del font. I valori possibili sono normal, oblique (o italic)

`font-variant`

Valori possibili normal o small-caps

small-caps: simile a font-style assegna tutte maiuscole grandi per le lettere maiuscole e maiuscole piccole per le lettere minuscole. normal per rimuovere il tutto.

`font-weight`

Accetta valori compresi tra 100 e 900 a passi di 100. E' possibile usare normal e bold come valori; essi corrispondono rispettivamente a 400 e 700. E' possibile anche usare lighter e bolder.

# Dichiarazioni - font

font-size

- **dimensione assoluta:** xx-small, x-small, small, medium, large, x-large, xx-large
- **dimensione relativa:** smaller e larger
- **lunghezza:** in genere si usa il punto (pt)
- **percentuale:** è una percentuale riferita alla dimensione del genitore dell'elemento che si sta considerando

text-decoration

**Valori possibili:** none, underline, overline, line-through, blink

text-transform

**Valori possibili:** capitalize (solo la prima lettera di ogni parola), uppercase, lowercase, none

# Good usage of CSS's font properties

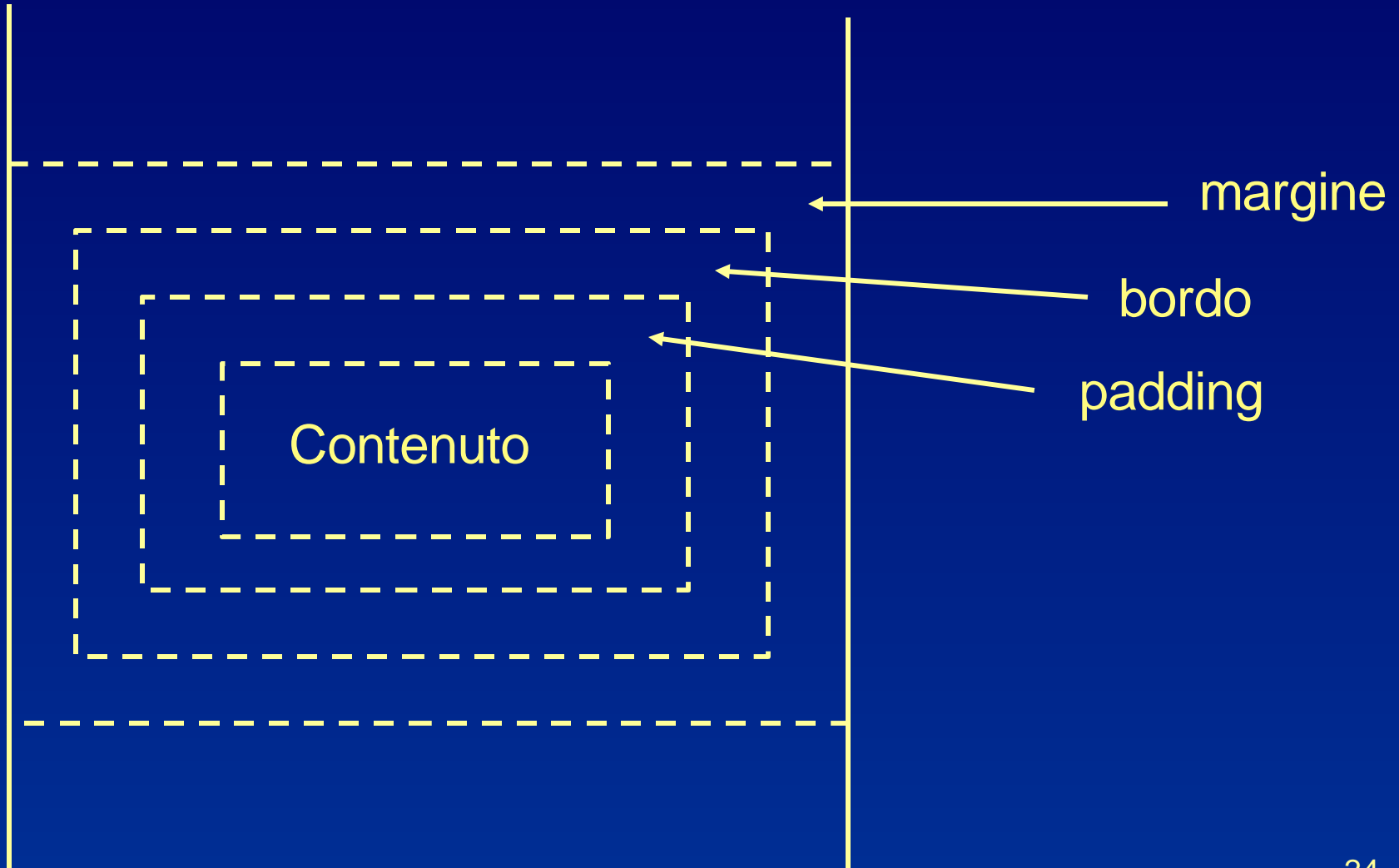
...

\* Do not specify the font-size in pt, or other absolute length units. They render inconsistently across platforms and can't be resized by the User Agent (e.g browser).

\* Use relative length units such as **percent** or (better) **em**, or, even better, set a base font-size for the document and **use absolute size** ([ xx-small | x-small | small | medium | large | x-large | xx-large ]) or **relative size** ([ larger | smaller ]) when defining the font size for a particular element within the document.

...

# Dichiarazioni – spazi (1)



# Dichiarazioni – spazi (2)

`margin-left`

`margin-right`

`margin-top`

`margin-bottom`

`margin`

1 valore: si applica a tutti e 4 i margini

2 valori: top & bottom, left & right

3 valori: top, left & right, bottom,

4 valori: top, right, bottom, left

Esempio

`margin: 10px 5px 10px 5px`

[Si veda 7.html](#)

# Dichiarazioni – spazi (3)

`padding-left`

`padding-right`

`padding-top`

`padding-bottom`

`padding`

1 valore: si applica a tutti e 4 i padding

2 valori: top & bottom, left & right

3 valori: top, left & right, bottom,

4 valori: top, right, bottom, left

# Dichiarazioni – spazi (4)

`border-style`

Valori possibili

- `none`
- `dotted`, `dashed`, `solid`, `double` per definire lo stile della linea
- `groove`, `ridge`, `inset`, `outset` per definire lo stile 3D

E' possibile specificare le caratteristiche dei 4 bordi

1 valore: si applica a tutti e 4 i bordi

2 valori: `top & bottom`, `left & right`

3 valori: `top`, `left & right`, `bottom`,

4 valori: `top`, `right`, `bottom`, `left`

# Dichiarazioni – spazi (5)

`border-width`

Valori possibili

- none
- thin, medium, thick per definire lo spessore
- una lunghezza

E' possibile specificare lo spessore dei 4 bordi

1 valore: si applica a tutti e 4 i bordi

2 valori: top & bottom, left & right

3 valori: top, left & right, bottom,

4 valori: top, right, bottom, left

# Dichiarazioni – spazi (6)

`border-color`

E' possibile specificare il colore dei 4 bordi

1 valore: si applica a tutti e 4 i bordi

2 valori: top & bottom, left & right

3 valori: top, left & right, bottom,

4 valori: top, right, bottom, left

# Dichiarazioni – spazi (7)

`border`

Se tutti e 4 i bordi sono uguali è possibile specificare `style`, `width` e `color`

E' possibile specificare il colore dei 4 bordi

- 1 valore: si applica a tutti e 4 i bordi

- 2 valori: `top & bottom`, `left & right`

- 3 valori: `top`, `left & right`, `bottom`,

- 4 valori: `top`, `right`, `bottom`, `left`

# Dichiarazioni – spazi (8)

`text-align`

Valori possibili:

`left`

`right`

`centered`

`justified`

# Dichiarazioni – spazi (9)

`text-indent`

Per impostare l'indentazione della prima riga

E' possibile usare una lunghezza o una percentuale

`line-height`

Specifica la distanza tra le righe del testo

E' possibile usare una lunghezza, una percentuale o numero

# Dichiarazioni – spazi (10)

`word-spacing`

Per impostare la distanza tra le parole

E' possibile usare `normal` o una lunghezza (la lunghezza viene aggiunta allo spazio normale tra le parole). E' possibile specificare valori negativi.

`letter-spacing`

Specifica la distanza tra le lettere

E' possibile usare i valori come con `word-spacing`

[Si veda 8.html](#)

# Dichiarazioni – spazi (11)

`vertical-align`

Per alzare o abbassare lettere o immagini sopra o sotto la baseline

Valori possibili:

`baseline`

`sub`

`super`

`top`

`text-top`

`middle`

`text-bottom`

`bottom`

Valore percentuale

# Dichiarazioni - immagini

Le proprietà legate alle immagini valgono anche per gli elementi INPUT, SELECT, TEXTAREA, OBJECT.

Principali proprietà:

`width` può avere valori di tipo lunghezza o percentuale

`height` può avere valori di tipo lunghezza

`width` e `height` valgono per gli elementi di tipo “block”

Si veda [9.html](#)

# Nota proprietà “float” dei Box

'float'

Value: left | right | none

Initial: none

Applies to: all elements

Inherited: no

Percentage values: N/A

With the value 'none', the element will be displayed where it appears in the text. With a value of 'left' ('right') the element will be moved to the left (right) and the text will wrap on the right (left) side of the element. With a value of 'left' or 'right', the element is treated as block-level (i.e. the 'display' property is ignored). See section 4.1.4 for a full specification.

```
IMG.icon {  
  float: left;  
  margin-left: 0;  
}
```

The above example will place all IMG elements with 'CLASS=icon' along the left side of the parent element.

This property is most often used with inline images, but also applies to text elements.

# Dichiarazioni – colori e sfondi

## Proprietà:

`color`: per impostare il foreground

`background-color`: per impostare il colore di background

`background-image`: per impostare l'immagine di sfondo

`background-repeat` valori possibili:

`repeat`

`repeat-x`

`repeat-y`

`no-repeat`

`background-attachment`: per gestire lo scrolling (`scroll`, `fixed`)

`background-position`: percentuale, lunghezza, `top`, `center`, `bottom`,  
`left`, `right`, `center`

`background`: per impostare tutti i valori per il background

[Si veda 10.html](#)

# Dichiarazioni – proprietà di classificazione (1)

E' possibile cambiare la classificazione di un elemento e cioè farlo passare da inline a blocco e viceversa.

`display`: valori possibili `block`, `inline`, `list-item`, `none`

Se si specifica un elemento di tipo `list-item` è possibile indicarne il tipo con la proprietà

`list-style-type`

Valori possibili:

`disc` (cerchio pieno)

`circle` (cerchio vuoto)

`square`

`decimal` (p.e. 1,2,3, ecc.)

`lower-roman` e `upper-roman` (i,ii, iii e I, II, III ecc.)

`lower-alpha` e `upper-alpha` (a,b,c e A,B,C ecc.)

`none`

# Dichiarazioni – proprietà di classificazione (1)

- Oltre al tipo di simbolo che è possibile utilizzare si può usare la proprietà `list-style-image` (per un `list-item`) per specificare un URL di una immagine.
- `list-style-position`: proprietà usata per impostare l'etichetta dell'oggetto della lista: valori possibili `inside` e `outside`
- `list-style`: per impostare `list-style-type`, `list-style-image`
- `white-space` per gestire gli spazi, tab e newline
  - `normal`: è il default. Il browser ignora e compatta spazi, tab e newline
  - `pre`: il browser mantiene gli spazi e i tab e trasforma i newline in ritorni a capo
  - `nowrap`: come `normal` ma il browser va a capo solo in presenza di `<BR>`.

# Usare più fogli di stile

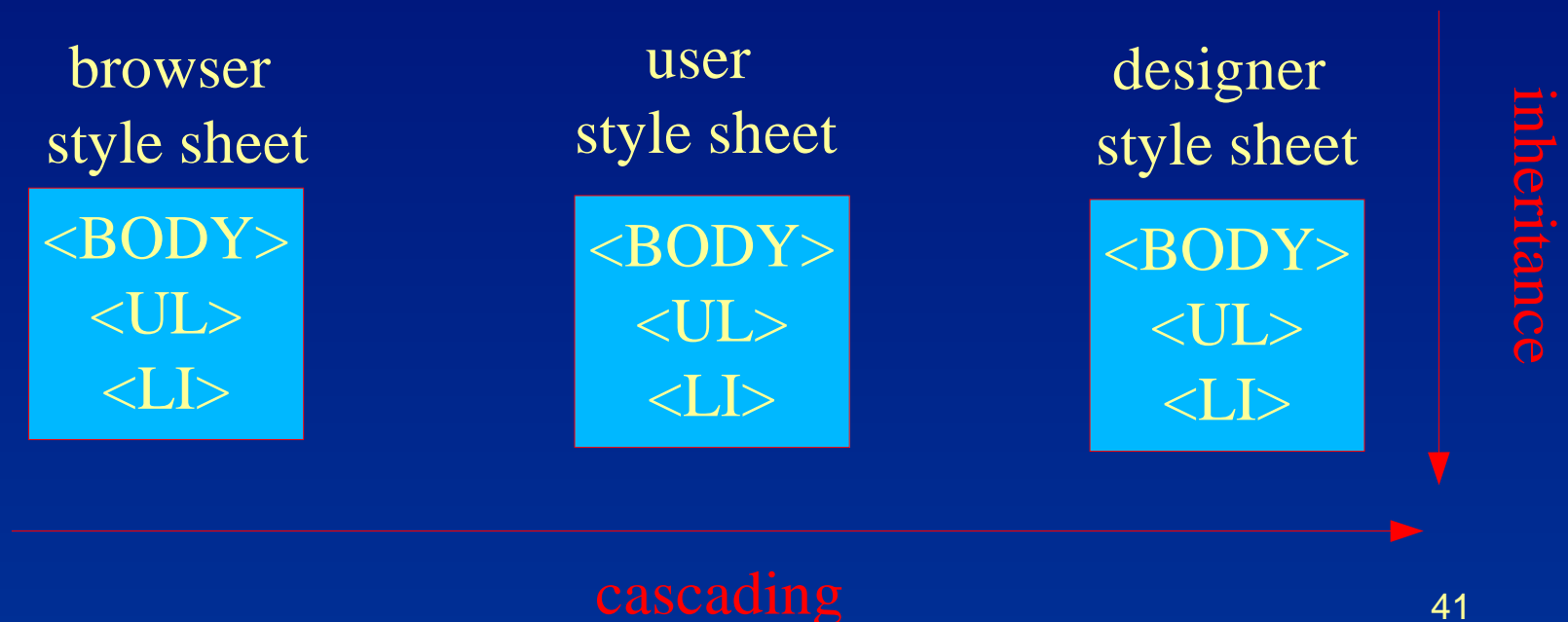
L'idea base di CSS è quella di combinare più fogli di stile (cascading).

Anche quando sembra ci sia un solo foglio di stile ce ne sono almeno 2 in quanto il browser usa regole di stile predefinite e interne e quindi è come se usasse un foglio di stile.

# Usare più fogli di stile – conflitti (1)

Poiché diversi fogli di stile possono contenere regole per la stessa istanza di un elemento possono sorgere dei *conflitti*.

- Conflitti in seguito alla definizione all'interno di uno stesso foglio di stile. Per esempio un type selector per un paragrafo e un selettore contestuale che seleziona solo paragrafi all'interno di una tabella
- Conflitti che nascono dall'uso di più fogli di stile



# Usare più fogli di stile – conflitti (2)

Il processo di risoluzione dei conflitti individua le regole che creano conflitti e si applica fino a quando terminano i conflitti.

Il processo di risoluzione dei conflitti è basato su 5 passi:

1. Trovare tutte le regole che è possibile applicare all'elemento in questione. Per fare questo si usano i selettori che sono stati definiti.
  - Si cercano eventuali regole segnate come `!important`
    - Es. `H1 { color: black !important; background: white !important }`
  - Ordine by origin. PAG. 311 e 316. Si veda schema slide precedente
  - Ordinamento per specificità p.e. una regola definita per `P` è più generale che una regola definita per `LI P`. Vedi SLIDE successiva.
  - Si ordinano le regole in base all'ordine in cui sono specificate. Le regole specificate più avanti hanno una precedenza maggiore.

# Usare più fogli di stile – conflitti (3)

## NOTA PER LA REGOLA 4

Sort by specificity of selector: more specific selectors will override more general ones. To find the specificity,

count the number of ID attributes in the selector (a),

the number of CLASS attributes in the selector (b),

and the number of tag names in the selector (c).

Concatenating the three numbers (in a number system with a large base) gives the specificity.

Some examples:

LI	{...} /* a=0 b=0 c=1 -> specificity = 1 */
UL LI	{...} /* a=0 b=0 c=2 -> specificity = 2 */
UL OL LI	{...} /* a=0 b=0 c=3 -> specificity = 3 */
LI.red	{...} /* a=0 b=1 c=1 -> specificity = 11 */
UL OL LI.red	{...} /* a=0 b=1 c=3 -> specificity = 13 */
#x34y	{...} /* a=1 b=0 c=0 -> specificity = 100 */

# Cascading Style Sheets level 2 (CSS2)

- In CSS1 non c'è il concetto di media. In CSS2 c'è. (per esempio il volume per il suono).
- CSS2 ha il concetto di paged-media
- CSS1 non ha proprietà per formattare tabelle ma solo testo e immagini dentro le tabelle.
- CSS2 introduce il concetto di posizionamento assoluto.
- CSS2 aggiunge altri selettori p.e. `child selector` e `attribute selector`
- CSS2 introduce la numerazione automatica p.e. per capitoli, intestazioni ecc.
- CSS2 definisce nuove pseudo classi p.e. per attivazione di un elemento alla pressione del mouse ecc.

# Conclusioni

Abbiamo parlato di

- Principi di CSS
- Meccanismi per definire CSS

# Riferimenti

## *Wilde's WWW, capitolo 6*

### Altri testi:

- *Cascading Style Sheets, level 1*

*<http://www.w3.org/TR/REC-CSS1>*

- *Cascading Style Sheets, level 2*

*CSS2 Specification <http://www.w3.org/TR/REC-CSS2>*