



# Informatica Umanistica

## A.A. 2008/2009

---

### LEZIONE 3

HTML + CSS

Il contenuto e la sua visualizzazione:  
separati finalmente!

## Modifica al calendario

---

- 22/04 mercoledì ore 16-18
- 23/04 giovedì ore 16-18
- 29/04 mercoledì ore 16-18
- ~~30/04 giovedì ore 16-18~~
- 06/05 mercoledì ore 16-18
- 07/05 giovedì ore 16-18
- 13/05 mercoledì ore 16-18
- 14/05 giovedì ore 16-18
- **19/05 martedì ore 16-18**
- 20/05 mercoledì ore 16-18
- 21/05 giovedì ore 16-18



# Lezione 3

## SOMMARIO

---

1. Ripasso html
2. Tabelle
3. Progettazione universale e accessibilità
4. Contenuto e presentazione: separati!
5. CSS

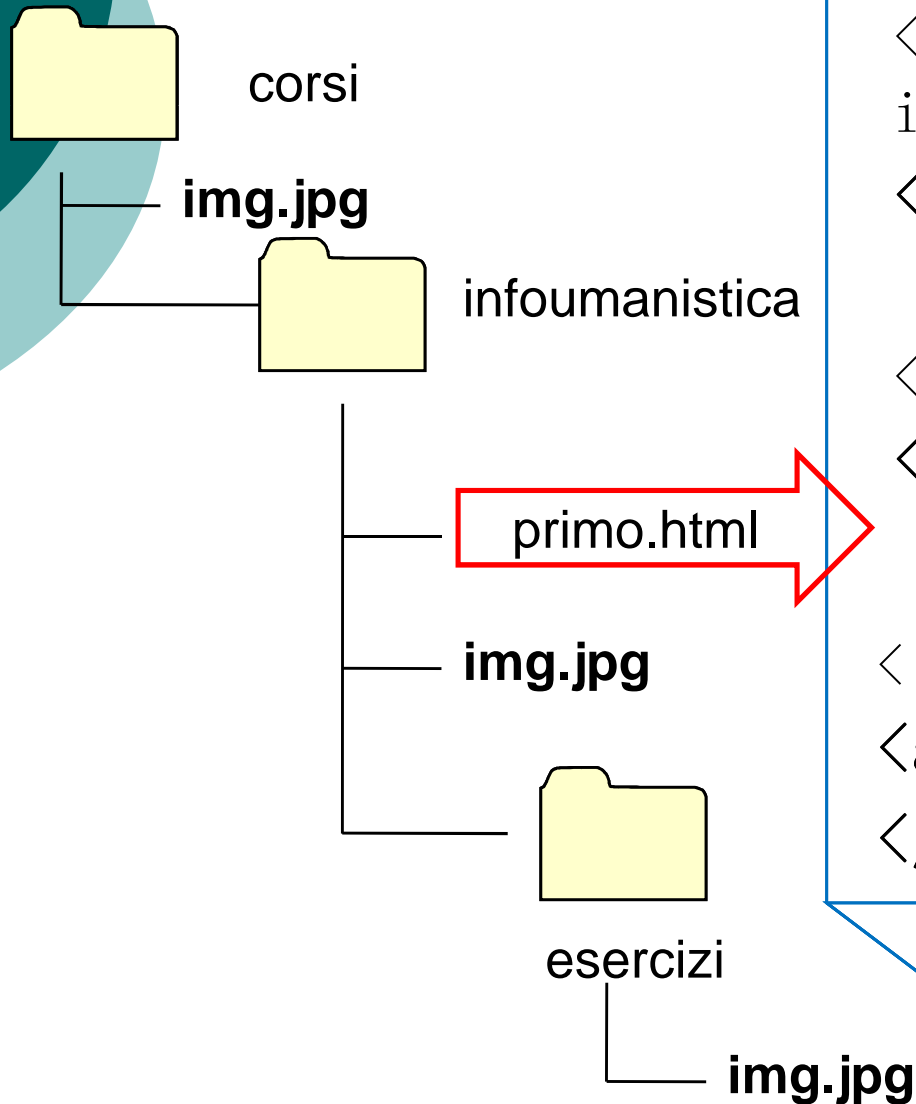
# 1. Link – Collegamenti Iperestuali il tag <a>

---

<a href="URI"> risorsa linkata </a>

<a href="mailto:indirizzo@posta.it"> scrivimi </a>

# 1. Link – Collegamenti Iper testuali URI relativi



```
<!-- l'immagine si trova in  
infoumanistica -->
```

```
<a href="img.jpg"> img </a>
```

```
<!-- l'immagine si trova in corsi -->
```

```
<a href="../img.jpg"> img </a>
```

```
<!-- l'immagine si trova in esercizi-->
```

```
<a href="esercizi/img.jpg"> img  
</a>
```

## 2. Tabelle

---

- Le tabelle costituiscono un modo per organizzare "dati" in righe e colonne;
- Le "celle" delle tabelle possono contenere:
  - Testo
  - Immagini
  - Link
  - Altre tabelle
  - ....
- Le tabelle permettono anche di "strutturare" le pagine, di raggruppare celle, contenere titoli e didascalie;

# 2. Tabelle

## Tag per le tabelle

---

- Il TAG fondamentale: **<table>...</table>**
- Didascalia: **<caption> </caption>**  
Indica una didascalia della tabella. Se si vuole inserire, va messo subito dopo <table>
- Gli elementi di <table> sono
  - **Riga <TR></TR>** (Table Row)
  - **Cella <TD></TD>** (Table Data) deve essere all'interno di una <TR>

Esempio:

```
<table>
  <caption> Tabella con una cella</caption>
  <tr> <td> </td> </tr>
</table>
```

## 2. Tabelle

### Attributi di <table>

---

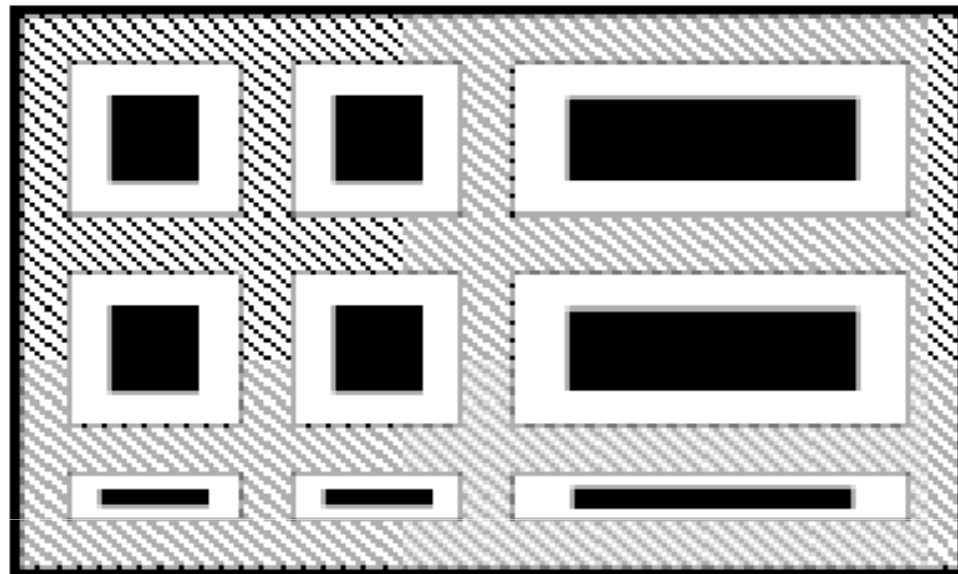
- **Larghezza** <TABLE WIDTH="?"> (in pixel o percentuale %)
- **Allineamento**  
<TR ALIGN=LEFT|RIGHT|CENTER|MIDDLE|BOTTOM>
- **Allineamento** <TD ALIGN=LEFT|RIGHT|CENTER|MIDDLE|BOTTOM>
- **Colonne da occupare** <TD COLSPAN=?>
- **Righe da occupare** <TD ROWSPAN=?>
- **Larghezza desiderata** <TD WIDTH=?> (in pixel)
- **Larghezza desiderata in percentuale** <TD WIDTH="%">
- **Colore di sfondo della cella** <TD BGCOLOR="#\$\$\$\$\$\$">


# 2. Tabelle

## Bordi e Margini

---

Table border 



Cellspacing 

Cellpadding 

Cell content 



## 2. Tabelle

### Bordi e Margini – in pixel

---

- o **Bordo:**

`<table border="?" bordercolor="?">`

- o **Spazio tra celle:**

`<TABLE CELSPACING="?">`

- o **Spazio all'interno:**

`<TABLE CELLPADDING="?">`

## 2. Tabelle

### Esempio (3 righe – 2 colonne)

```
<table><caption> eventuale didascalia </caption>
```

1° riga

```
<tr>
```

```
<td><strong>Cella 1, Riga 1</strong></td>
```

```
<td><strong>Cella 2, Riga 1</strong></td>
```

```
</tr>
```

2° riga

```
<tr>
```

```
<td><em>Cella 1, Riga 2</em></td>
```

```
<td><em>Cella 2, Riga 2</em></td>
```

```
</tr>
```

3° riga

```
<tr>
```

```
<td><em>Cella 1, Riga 3</em></td>
```

```
<td><em>Cella 2, Riga 3</em></td>
```

```
</tr>
```

```
</table>
```

## 2. Tabelle

### Riassunto Elementi Tabelle

---

```
<table width="" border="" cellspacing=""  
  cellpadding="" bgcolor="" bordercolor="">  
  <caption align="" > didascalia </caption>  
  <tr bgcolor="" align="" >  
    <td bgcolor="" rowspan="" colspan="" width  
    ="" height="" align="" >  
  [...]  
</table>
```

## 2. Tabelle

---

- Se non viene specificata una dimensione per la tabella le celle si adattano al contenuto.
- Restrungendo la finestra del browser la tabella manderà a capo il contenuto delle celle, poiché' la tabella si adegua alle dimensioni della finestra.
- Quindi fissare le dimensioni è importante!
- E' consigliabile l'utilizzo delle dimensioni in percentuale per creare layout di pagina che si adeguino a qualsiasi risoluzione di schermo.
- Per grandezze fissate e per non perdersi nel calcolare la percentuale, si lavora in pixels.


## 2. Tabelle

### Esempio Tabella a grandezza fissa

---

```
<table width="75%" height="20%" border="3">
  <tr>
    <td><strong>Cognome</strong></td>
    <td><strong>Nome</strong></td>
    <td><strong>Indirizzo</strong></td>
  </tr>
  <tr>
    <td><em> Ferron</em></td>
    <td><em> Michela</em></td>
    <td><em>Via .... </em></td>
  </tr>
  <tr>
    <td><em>Massa</em></td>
    <td><em>Paolo</em></td>
    <td><em>Via ....</em></td>
  </tr>
</table>
```

NB: Il tag `<strong>` sembra identico al tag `<b>`:  
In realtà il testo marcato con `strong` verrà letto con un tono di voce più alto dai browser vocali, cosa che non avviene per `<b>`



## 2. Tabelle

### Larghezza celle e bordo

---

- La larghezza va espressa solo per le celle della prima riga, in quanto le righe seguenti seguiranno le impostazioni della prima.
- Nota: impostando `border="0"` i bordi della tabella non sono più visibili.

## 2. Tabelle

### Link nelle tabelle

---

```
<table width="50%" border="0">
<tr>
  <td width="30%" align="center"><strong>
Link</strong></td>
  <td width="70%"
align="center"><strong>Categoria</strong></td>
</tr>
<tr>
  <td align="center" ><a
href="http://www.google.it">Google.it</a></td>
  <td align="center" ><em>Motore di ricerca </em></td>
</tr>
<tr>
  <td align="center" ><a
href="http://www.yahoo.it">Yahoo.it</a></td>
  <td align="center" ><em>Motore di ricerca</em></td>
</tr>
</table>
```

## 2. Tabelle

### Ultimo esempio

---

```
<table width="75%" border="3" bordercolor="#220066"
  bgcolor="#ccbfff" cellpadding="5" cellspacing="10">
<tr>
  <td width="30%" bgcolor="#7755ff"> <strong> Link
  </strong></td>
  <td width="70%" bgcolor="#7755ff"><strong> Categoria
  </strong></td>
</tr>
<tr>
  <td><a href="http:\\www.google.it">Google.it</a></td>
  <td><em>Motore di ricerca</em></td>
</tr>
<tr>
  <td><a href="http:\\www.yahoo.it">Yahoo.it</a></td>
  <td><em>Motore di ricerca </em></td>
</tr>
</table>
```

# Esercizi

---

- <http://www.antoniobucchiarone.it/Esercizi/Esercizi-Lezione2.pdf> (quiz html e tabelle)
- [http://virtualltips.files.wordpress.com/2009/04/quiz\\_html\\_tag.pdf](http://virtualltips.files.wordpress.com/2009/04/quiz_html_tag.pdf) (quiz html e tag)
- <http://www.antoniobucchiarone.it/Esercizi/Esercizio3.html> (esercizi tag, liste e tabelle)



# Informatica Umanistica

## A.A. 2008/2009

---

### **LEZIONE 3**

HTML + CSS

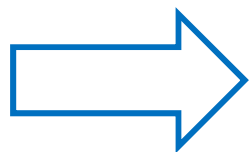
Il contenuto e la sua visualizzazione:  
separati finalmente!

# 3. Progettazione universale (design for all)

---

- Prodotti concepiti per essere utilizzati dal **maggior numero di persone possibili**

Esempi: ridotta capacità visiva, uditiva, cognitiva, motoria, software non aggiornato, browser testuale, sintetizzatore vocale, telefono cellulare, ambiente di lavoro poco confortevole, rumoroso o poco illuminato.



Prodotti più usabili per tutti

# 3. Importanza dell'accessibilità: Direttive europee e ministeriali

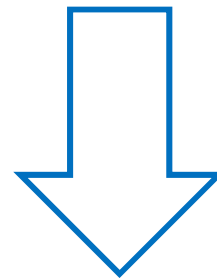
---

- Direttive dell'Unione Europea
- Direttive ministeriali italiane (es. legge Stanca del 2004, che obbliga le amministrazioni pubbliche al rispetto dei requisiti di accessibilità dei sistemi informatici)
- **Rendere accessibile un sito web non significa rinunciare a qualcosa, ma ristrutturarlo e arricchirlo** (es. uso dell'attributo "alt" di <img>)

# 3. Accessibilità: la Web Accessibility Initiative del W3C

---

- Il World Wide Web Consortium definisce le linee guida per progettare siti web accessibili (<http://www.w3.org/TR/WCAG20/>)



**Trasformazione elegante delle pagine**

### 3. Trasformazione elegante delle pagine: che significa?

---

- **I siti web si trasformano a seconda dell'utente** che li visita. Ma come?
- **Separando contenuto e presentazione!** In modo che tutto ciò che è testo e informazione possa essere riprodotto in base alle caratteristiche dei vari strumenti (progettazione universale)

## 4. Contenuti vs presentazione

---

- Una [pagina web](#) è formata fondamentalmente da due elementi: i **contenuti** veri e propri (con la loro **struttura logica**) e la **presentazione** o **formattazione** ovvero l'aspetto con cui i contenuti sono mostrati all'utente.
- HTML ha lo scopo di **gestire i contenuti**, specificandone la struttura attraverso *tag* diversi.

**Ogni tag** (ad esempio `<h1>` o `<p>`) **specifica un diverso ruolo dei contenuti** che contrassegna (quindi il tag `<h1>` definirà un'importanza maggiore del tag `<p>`).

# 4. Contenuti vs presentazione

---

Definizioni del World Wide Web Consortium, 1999:

- **Contenuto:** ciò che si comunica (es. linguaggio naturale)
- **Struttura logica:** organizzazione logica del documento (capitoli, indici, titoli, ...)
- **Presentazione:** come si presenta il documento (per la stampa, su cellulare, text-only)



## 4. Esempio: non usiamo male le tabelle

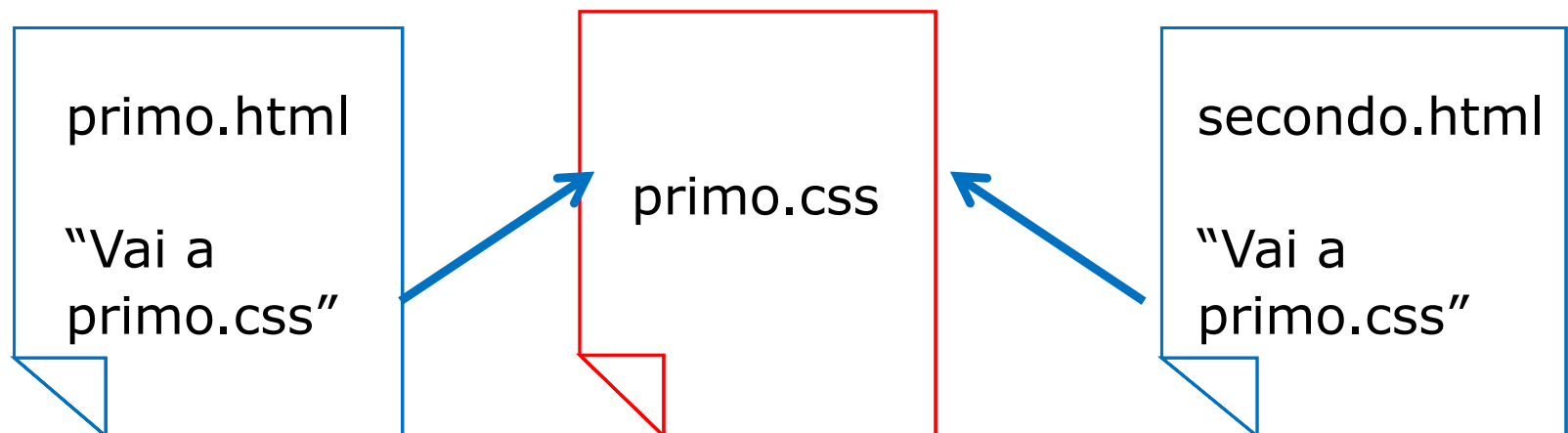
---

- Una volta le tabelle servivano per **organizzare la struttura del contenuto**
- Poi si è cominciato ad usarle per presentare graficamente il contenuto (immagini, annidamento di altre tabelle, posizionamento preciso del testo)
  - **Pagine web più pesanti**
  - **Difficoltà di visualizzazione** da parte di altri dispositivi (es. larghezza fissa)

# 5. Fogli di stile a cascata: CSS - Cascading Style Sheet)

- O **fogli di stile**, definiscono la presentazione grafica di documenti HTML, XHTML e XML.
- "A cascata": nel mio documento HTML posso incorporare più fogli di stile, ognuno dei quali è in grado di prevalere sull'altro grazie a delle regole gerarchiche

**Quando accedo alla pagina web, il browser viene indirizzato al file CSS per recuperare le informazioni sulle modalità di visualizzazione del contenuto**



# 5. Separare contenuto (html) e presentazione (css): vantaggi

---

## Per l'utente

- **siti web leggeri e tempi di caricamento brevi:** un solo file CSS può controllare la presentazione grafica di tutte le pagine
- l'utente può **personalizzare il layout** (dimensioni del testo, colori)
- **trasformazione elegante delle pagine** a seconda dello strumento (sintetizzatore vocale, display Braille o testuale, cellulare)

## 5. Separare contenuto (html) e presentazione (css): vantaggi

---

**Per il webmaster**

**aggiornamento più veloce delle pagine** perché:

- il documento HTML è pulito e ordinato
- lo stile di tutte le pagine si può cambiare modificando un unico file CSS

○ **Esempio:**

<http://www.csszengarden.com/tr/italiano/>

# 5. Come specifico il CSS per il mio HTML?

1) Inserendo nel tag <head> della pagina un collegamento ad un foglio di stile esterno con estensione .css

```
<head> <title>Esempio</title>
```

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

```
</head>
```

2) Inserendo tra gli specifici tag <style> e </style> le dichiarazioni CSS.

```
<head> <title>Esempio</title>
```

```
<style type="text/css">
```

```
codice css
```

**NOI USIAMO LA 1)**

# 5. Come specifico il CSS per il mio HTML? (2)

---

1) Inserendo nel tag <head> della pagina un collegamento ad un foglio di stile esterno con estensione .css

**POSSIBILE SPECIFICARE PIU' DI UN CSS!!!**

Ad esempio un CSS per visualizzazione su monitor e uno per stampa (senza barra laterale, senza menu, font piu' piccolo, ...)

**Noi usiamo solo un CSS**

```
<head> <title>Esempio</title>
```

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

```
<style type="text/css" media="screen">...</style>
```

## 5. Esercizio: creare il foglio di stile **stile1.css** per il documento **primo.html**

---

- Aprite **primo.html**
- Create **stile1.css** (per il momento è vuoto)
- Inserite in primo.html il **collegamento a stile1.css**:

```
<head> <title>Esempio</title>
```

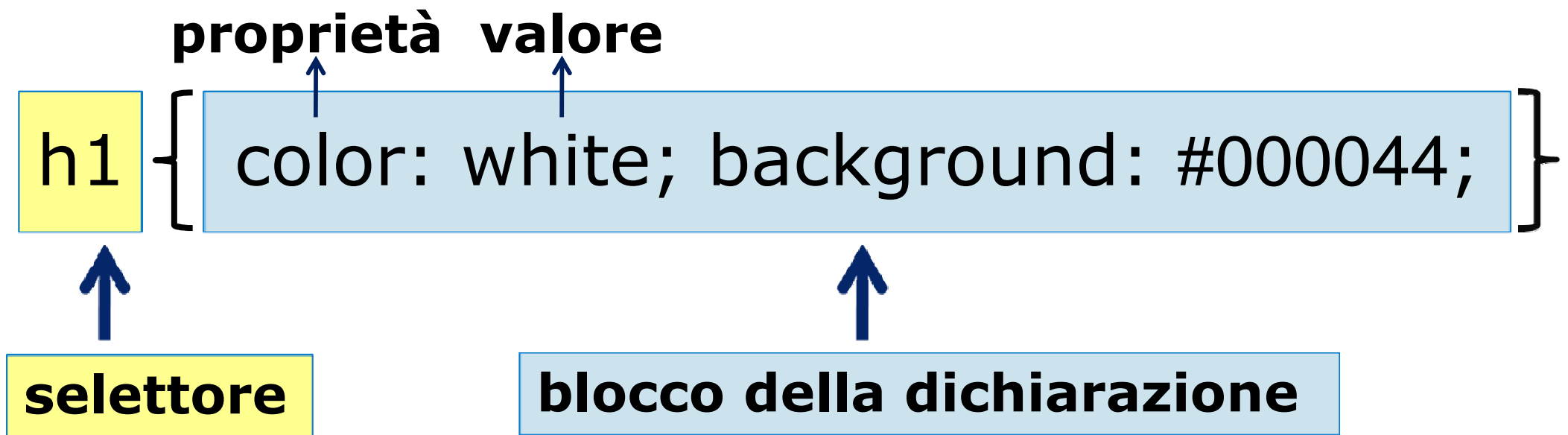
```
<link rel="stylesheet" type="text/css" href="foglio_di_stile.css">
```

```
</head>
```

# 5. Com'è fatto un foglio di stile CSS

Un foglio di stile è composto da due tipi di **dichiarazioni: regole** e **commenti** (i commenti non vengono "letti" dal browser: `/* commento */` )

## Struttura di una regola



# 5. Cosa posso specificare con un CSS

---

Proprietà CSS sono molte, circa 60.

- **background**: definisce lo sfondo di un elemento;
- **border**: definisce il bordo di un elemento;
- **color**: definisce il colore del testo di un elemento;
- **float**: definisce un blocco flottante, ovvero che permette la disposizione di altri elementi ai suoi lati;
- **font**: definisce le proprietà del carattere;
- **margin** e **padding**: definiscono lo spazio circostante gli elementi (esterno, interno);
- **text-align**: definisce l'allineamento del testo.

# 5. Cosa posso specificare con un CSS: i selettori

---

- L'aspetto grafico (formattazione) di qualcosa in una pagina HTML.
- Di cosa?
  - 1) Elementi di un certo **tipo** (tag), es: "h1"
  - 2) Elementi di una certa **classe**
  - 3) Elementi identificati con un **id**

**Li specifico tramite un selettore:** una dichiarazione che **seleziona** la parte o le parti di un documento soggette ad una specifica regola

## 5. Selettori di tipo

---

- sono i selettori più generici, e indicano che la regola deve essere applicata a tutti gli elementi del tipo indicato
  - **h1** {...} → Applica la regola a tutto ciò che nell'html è marcato con il tag <h1>
  - **ul** {...}
  - **p** {...}
  - **a** {...}
  - **div** {...}
  - **\*** {...} → **\*** {...} è un selettore **particolare**: seleziona tutti gli elementi del documento

# 5. Selettori di tipo: esempi

---

- **h1** { color: #0077cc; }
- **ul** {list-style-type: square; }
  
- **p** {  
font-family: arial, Verdana, sans-serif;  
font-size: 12px;  
text-decoration: underline;  
background-color: yellow;  
color: blue;  
border : 1px solid red;  
}

## 5. Selettori di tipo: esempi

---

- **a:link** { background: #fff; color: #f00; text-decoration: underline; }
- **a:visited** { background: #fff; color: #f77; text-decoration: underline; }
- **a:hover, a:active, a:visited** { background: #f00; color: #fff; text-decoration: none; }

### **Pseudo- classi:**

Applicano uno stile per un elemento al verificarsi di certe condizioni

## 5. Selettori di tipo: esempi

---

HTML

```
<div>Vuolsi cosi' cola'</div>
```

```
<div>dove si puote</div>
```

```
<div>cio' che si vuole</div>
```

CSS

```
div {
```

```
border: 3px solid #ff0000;
```

```
}
```

**Selettore di elementi (type selector)**

# 5. Selettori di tipo: esempi

---

HTML

```
<div>Vuolsi cosi' cola'</div>
```

```
<div>dove si puote</div>
```

```
<div>cio' che si vuole</div>
```

CSS

```
div {  
  border: 3px dotted green;  
  margin : 5px;  
}
```

# 5. Selettori di tipo: esempi

---

HTML

```
<div>Vuolsi cosi' cola'</div>
```

```
<div>dove si puote</div>
```

```
<div>cio' che si vuole</div>
```

CSS

```
div {  
  border: 10px solid #ff0000;  
  margin : 5%;  
  padding: 10px;  
}
```

# 5. Selettore del discendente

---

Seleziona il discendente di un elemento.

Un elemento è discendente di un altro se è contenuto al suo interno, a qualsiasi livello.

Es.: liste

- `ul li { color: #000000; }`
- `ul li a {text-decoration: underline;}`

Seleziona tutti i link `<a>` contenuti nei list item `<li>`, contenuti a loro volta in liste non ordinate `<ul>` (unordered list)

# 5. Selettore di classi (class)

---

- Le classi definiscono un insieme di oggetti omogenei. Per associare un elemento a una è sufficiente specificarne il nome nel file HTML attraverso l'attributo **class**.

HTML:

- `<p class="testorosso">....</p>`  
`<div class="testorosso">....</div>`  
`<table class="testorosso">...</table>`  
`<p class="testorosso">....</p>`

CSS (qui la classe si indica con il punto " . "):

- `.testorosso { color: red; }`

Seleziona tutto ciò che nell'html è indicato con la classe "testorosso"

# Selettori di identificatori (id)

---

- Simili alle classi, ma **identificano in modo univoco un elemento**. Se assegno ad un paragrafo l'id "testorosso", non potrò più usare questo valore nel resto della pagina.

HTML:

- `<p id="testorosso">....</p>`  
`<div id="intestazione">....</div>`

CSS (l'identificatore si indica con il cancelletto "#"):

- **#testorosso** { color: red; }
- #intestazione { font-size: 16px; }

Seleziona in modo univoco l'unico elemento identificato nell'html con l'id "testorosso"