

# Laboratorio di XHTML e CSS

Lezione 2: Formattazione del testo

# Ripasso

- ▶ (X)HTML è una versione di HTML basata sul metalinguaggio XML e ha una sintassi rigorosa. Serve per definire la struttura logica dei documenti.
- ▶ I fogli di stile CSS (Cascading Style Sheet) permettono di definire gli aspetti di presentazione dei documenti.

Quindi:

- (X)HTML: struttura
- CSS: presentazione

# Applicare i CSS ad XHTML

# CSS e XHTML

Ci sono tre tipi di collegamenti tra CSS e XHTML:

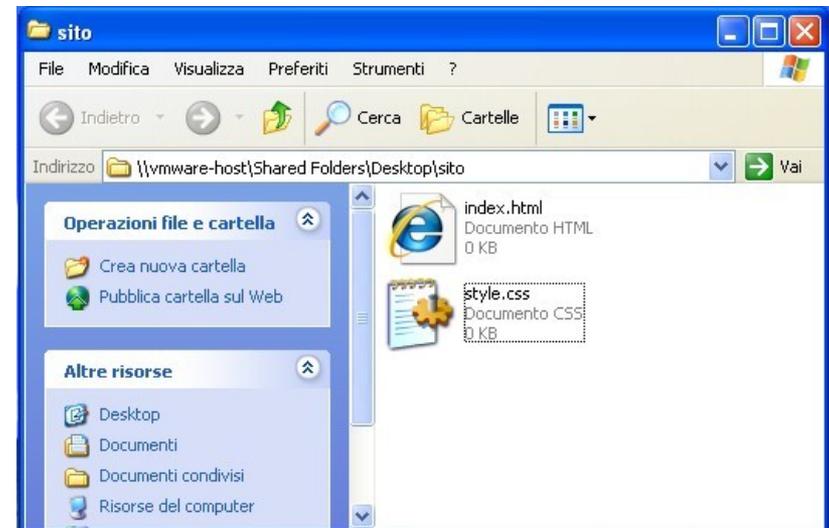
1. CSS esterni
2. CSS embedded
3. CSS inline

# CSS esterni

- ▶ Le regole CSS si trovano in un file esterno, con estensione .CSS
- ▶ Struttura (html) e presentazione(css) sono completamente separate!
- ▶ Il file CSS è richiamato all'interno del file (X)HTML attraverso un apposito elemento “link” all'interno dell'elemento “head”

«link rel=“stylesheet” type=“text/css” href=“nomeFile.css”/»

```
<html>  
<head>  
<title>Titolo</title>  
<link rel=“stylesheet”  
  type=“text/css”  
  href=“style.css”/»  
</head>
```



# CSS embedded

- ▶ Le regole CSS sono contenute in un elemento “**style**” all’interno dell’elemento “**head**” del documento (X)HTML.
- ▶ Struttura e presentazione non sono completamente separate: si usa solo se ci sono regole specifiche per una sola pagina.

```
<html>
<head>
<title>Titolo</title>
<style type="text/css">
p{
  font-size:2em;
}
</style>
</head>
```

# CSS inline

- ▶ Le regole CSS sono contenute nell'attributo “style” dell'elemento che si vuole modificare (nel body).
- ▶ Struttura e presentazione non sono separate: si usa solo per test e prove rapide!

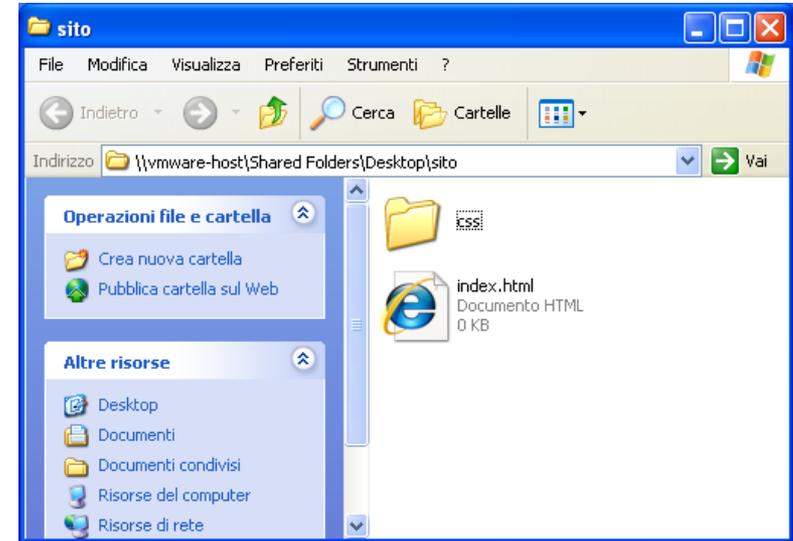
```
<html>
<head>
<title>Titolo</title>
</head>
<body>
<p style="font-size:2em;">
Il laboratoriodiXHTML
  verra' ...
</p>
```

# Ordine di applicazione

- ▶ Nel caso in cui ci siano definizioni di stile ripetute (con lo stesso selettore), è sempre l'ultima a prevalere. Nel caso di dichiarazioni in fogli di stile diversi, l'ordine di applicazione è il seguente (prevale l'ultimo foglio applicato):
  - CSS di default del browser
  - CSS esterno
  - CSS embedded
  - CSS inline

# Al lavoro! – esercizio 1

- ▶ Create una cartella di nome sito e dentro questa create una sottocartella (es. css) che conterrà i vostri file css.
- ▶ Aprite il vostro editor:
  - Create un nuovo documento html (con tutti i tag “head”, “title”, “body”) e salvatelo come file index.html in sito
  - Inserite in index.html un paragrafo:



“Alla domanda da dove inizi veramente la poesia italiana del Novecento, molti critici letterari hanno dato differenti e contrastanti risposte. I più grandi esponenti di questo periodo sono:”

- Create un altro nuovo documento e salvatelo come foglio stile dentro la sottocartella css:

Nome file: style.css e Salva come: All types (\*.\*)

- ▶ Aprire index.html con il browser

# Al lavoro! (2) – esercizio 1

- ▶ All'interno di style.css create la vostra prima regola css, che modifica il colore del font dei paragrafi:

```
p {  
    color: #600;  
}
```

- ▶ Aprite index.html e inserite l'elemento "link" nell'"head" per linkare il file CSS:

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

- ▶ Aprite il file xhtml con il browser: cosa succede?

# CSS: sintassi

# Nomenclatura

- ▶ I CSS permettono di definire gli aspetti di presentazione di un documento (X)HTML.

Esempio:



# Sintassi: introduzione

- ▶ All'interno di un file CSS (o all'interno della dichiarazione embedded) possiamo trovare due tipologie di regole:
  - quelle composte da selettori HTML
  - quelle che fanno riferimento ad attributi comuni:
    - id
    - class

# Selettori HTML

- ▶ Selettori HTML: utilizzano i nomi dei tag HTML (h1, p, a ...). Le dichiarazioni saranno applicate a tutti gli elementi caratterizzati dal tag usato.

Esempio:

```
h1 {  
    font-size: 2em;  
}
```

La dichiarazione viene applicata a tutti gli elementi di tipo h1

# Attributi comuni – 1

- ▶ Classi (“class”) e identificativi (“id”) consentono di assegnare un’etichetta (un “nome”) ad un qualunque elemento nel file html.
- ▶ Gli `id` identificano in modo univoco un singolo elemento all’interno di un documento HTML;

```
<p id = “unito”> Testo</p>
```

- ▶ Le classi possono essere associate ad un numero indefinito di elementi.

```
<p class = “matricola”> Testo</p>
```

# Attributi comuni -2

Esempio “id” e “class”:

```
<body>  
<p id =“unito”>Universita’ degli Studi di Torino</p>  
<p class=“matricola”>894830</p>  
<p class=“matricola”>894809</p>  
<p id =“unibo”>Universita’ degli Studi di Bologna</p>  
<p class=“matricola”>894830</p>  
</body>
```

# Attributi comuni -3

- ▶ Più classi possono essere associate allo stesso elemento, con la seguente sintassi:

```
<p class="classeUno classeDue"> Testo</p>
```

# Selettori id

- ▶ Selettori id: si basano sul valore dell'attributo "id". Le dichiarazioni saranno applicate a tutti gli elementi, il cui "id" corrisponde a quello indicato (uno solo per documento!).

Esempio:

```
#unito{  
  }  
}
```

font-size: 2em;

Nelle regole CSS,  
i selettori id sono  
introdotti dal simbolo "#"

- ▶ La dichiarazione viene applicata all'elemento il cui id è "unito"

# Selettori di classe

- ▶ Selettori di classe: si basano sul valore dell'attributo "class". Le dichiarazioni saranno applicate a tutti gli elementi la cui classe corrisponde a quella indicata.

Esempio:

```
.argomento{  
    font-size: 2em;  
}
```



Nelle regole CSS,  
i selettori class sono  
introdotti dal simbolo “.”

- ▶ La dichiarazione viene applicata agli elementi della classe “argomento”.
- ▶ Se ad un elemento sono collegate più classi, gli verranno applicate tutte le regole corrispondenti.

# Selettori di classe – 2

- ▶ Si può restringere il campo di applicazione di un selettore di classe ad un solo tipo di tag accodandolo al selettore HTML corrispondente.

Esempio:

```
h1.argomento {  
    font-size: 2em;  
}
```

- ▶ La dichiarazione viene applicata a tutti gli elementi di tipo h1 il cui attributo “class” ha valore “argomento”.

# Selettori raggruppati

- ▶ Si possono associare le stesse dichiarazioni a più selettori separando i nomi degli stessi con una virgola.
- ▶ Non è necessario definire regole separate.

Esempio:

```
#mittente, .argomento, h2 {  
    font-size: 2em;  
}
```

# Selettori annidati

- ▶ Si possono applicare delle regole CSS a degli elementi annidati indicando, prima del nome dell'elemento prescelto, una sequenza di elementi genitore, separati da spazi. Il selettore diventa più specifico!

Esempio:

```
#unitop a {  
    font-size: 2em;  
}
```

- ▶ La dichiarazione viene applicata agli elementi di tipo “a” contenuti in un elemento “p” contenuto a sua volta in un elemento con identificativo “unito”.
- ▶ Non è necessario indicare tutti gli elementi genitore!

# Ordine di applicazione delle regole

- ▶ Nel caso di più regole con lo stesso selettore, prevarrà quella che compare per ultima nel foglio stile
- ▶ Nel caso di più dichiarazioni (con selettori diversi) applicate allo stesso elemento, prevarrà quella con il selettore più specifico (ad es. l'id prevale sulla classe).

# Al lavoro! – esercizio 2

- ▶ Inserite nel file index.html un'intestazione di primo livello (`<h1>`) **“Poesia italiana del Novecento”** e nel file css una regola contenente le seguenti proprietà all'elemento h1:

```
color:#033;  
font-size: 3em;
```

- ▶ Dopo il paragrafo introduttivo inserite quattro paragrafi

```
<p id="ungaretti">Ungaretti Giuseppe</p>  
<p class="poesia">Soldati</p>  
<p class="poesia">Mattina</p>  
<p id="pascoli">Giovanni Pascoli</p>  
<p class="poesia">Carmina</p>
```

Apriete index.html con il browser.

# Al lavoro! (2) – esercizio 2

- ▶ Nel foglio stile impostate proprietà differenti per le classi e per gli id:

- Per le classi: `font-size: 1em;`  
`color:#333;`

- Per gli id: `font-size: 1.5em;`  
  
`color:#039; (per ungaretti)`  
`o color:#963; (per pascoli)`

- ▶ Aprite index.html con il browser.

# Dare stile al testo con i CSS

# Dare stile al testo coi CSS

Significa impostare con i CSS:

- ▶ Il tipo di font
- ▶ Il colore del font
- ▶ La dimensione del font
- ▶ Grassetto e corsivo del font
- ▶ Allineamento
- ▶ Sottolineatura
- ▶ Spaziature, bordi, margini(box)

# Tipo di font – 1

- ▶ Il tipo di font si può scegliere attraverso la proprietà CSS “font-family”. Il tipo di font viene generalmente definito per l’elemento body e sarà così utilizzato per tutto il documento.

Esempio:

```
body {  
    font-family: “Times New Roman”;  
}
```

- ▶ I nomi di font composti da più parole vanno inseriti tra virgolette.
- ▶ Possono essere visualizzati solo i font presenti sul computer dell’utente. I font più comunemente utilizzati sono: “Times New Roman”, arial, helvetica(per Mac), Century Gothic, AvantGarde (per Mac), verdana.

# Tipo di font – 2

- ▶ Se si indicano come valore di “font-family” più nomi di font, separati da virgola, il browser visualizzerà il primo disponibile. In questo modo, si può controllare la presentazione su diversi sistemi operativi o quando la prima scelta non è disponibile.
- ▶ E’ bene indicare, come ultima alternativa, un tipo di font generico, come “serif”, “sans-serif” o “monospace”.

Esempio: (inserirlo nel vostro style.css)

```
body {  
    font-family: arial, helvetica, sans-serif;  
}
```

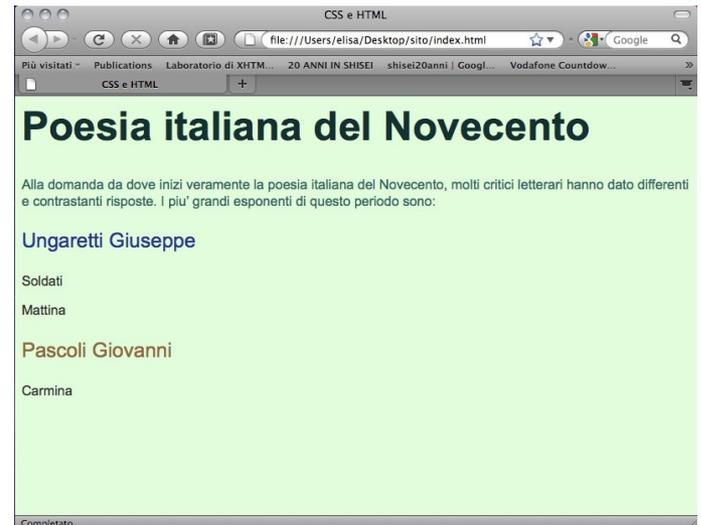
# Colore

- ▶ La proprietà “color” permette di controllare il colore del testo.
- ▶ La proprietà “background-color” (applicabile anche ad elementi non testuali) permette di definire il colore di sfondo.

Esempio: (aggiungetele proprieta' mancanti al vostro body)

```
body {  
    color: #666;  
    background-color: #DBFFDB;  
}
```

- ▶ E' pratica comune definire dei colori “di base” per l'elemento “body”, che saranno applicati a tutto il documento. Si possono poi definire colori diversi per elementi specifici.



# Valori: colore -1

- ▶ Valori RGB: sono costituiti da tre valori (numeri da 0 a 255 percentuali) che indicano i livelli di rosso, verde e blu. (red, green, blu)

```
p {  
    color: rgb(0, 0, 255);  
}
```

- ▶ Valori esadecimali: si utilizza il sistema di codifica esadecimale anziché quello decimale. Si basa su un sistema di numerazione in base 16 che comprende le cifre da 0 a 9 e da A a F. I valori Esadecimali sono costituiti dal simbolo “#” seguito da 3 o 6 caratteri esadecimali (le cifre vanno da 0 a 9 e poi da a ad f).

```
p {  
    color: #0000ff;  
}
```

# Valori: colore -2

## ► Nomi di colori:

esistono 17 valori standard, oltre a transparent: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

	Black		Yellow		Red		Maroon
	Gray		Lime		Green		Olive
	Silver		Aqua		Blue		Navy
	White		Fuchsia		Purple		Teal

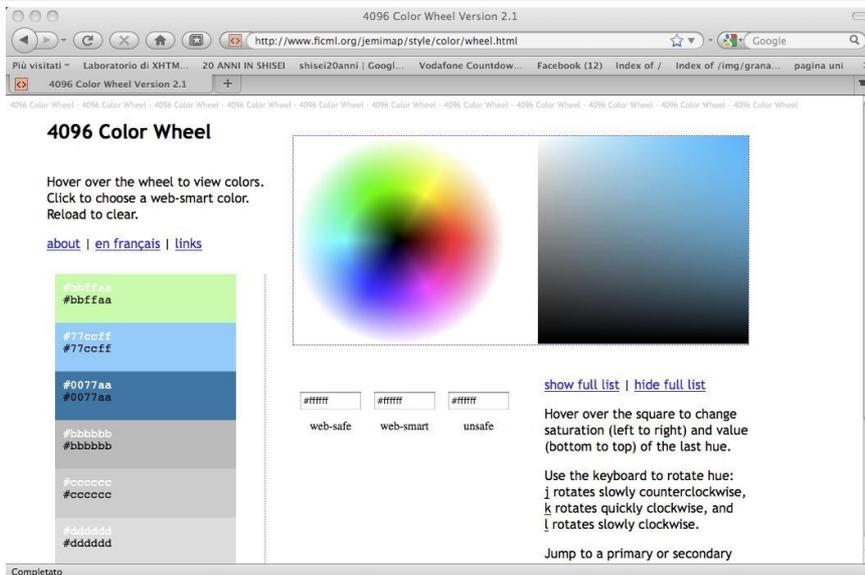
# Colori “sicuri”

- ▶ Web Safe: sono 216 colori che si vedono nello stesso modo, indipendentemente dal browser in cui appaiono. Sono composti (in notazione esadecimale) soltanto da combinazioni delle seguenti coppie: 00, 33, 66, 99, CC e FF.  
Esempio: “#990066” è web safe, “#99006F” no
- ▶ Web Smart: sono 4096 colori che si vedono nello stesso modo sui monitor con 56000 o 16 milioni di colori. Sono composti da combinazioni di tutte le possibili coppie di cifre (da 0 a F).  
Esempio: “#5522EE” è web smart

# Link utili

- ▶ Alcune risorse online per trovare i codici RGB e esadecimali dei colori:

- <http://www.ficml.org/jemimap/style/color/wheel.html>



- <http://cloford.com/resources/colours/webSMART1.htm>
- <http://colorblender.com/>
- <http://www.colorschemer.com/online.html>

# Valori: unità di misura

- ▶ Le unità di misura possono si distinguono in assolute e relative.
  1. Assolute: punti (pt), pixel (px)
  2. Relative: percentuali(%), em

Esempio:

```
h2 {  
    font-size: 2em;  
}
```

Em è un'unità di misura relativa che si basa sulla dimensione del carattere M maiuscolo del font dell'elemento contenitore

- ▶ Il suffisso che indica l'unità di misura si scrive subito dopo il numero, senza spazi.

# Dimensione

- ▶ La dimensione dei font viene controllata attraverso la proprietà “font-size”.
- ▶ Le unità di misura più utilizzate sono i pixel (px), gli em e le percentuali (%).
- ▶ In genere, è meglio usare unità di misura relative (em , %) per consentire all’utente di ridimensionare il testo. Si consiglia di definire il valore di “font-size” in % per il “body” e in em per gli altri elementi.

```
body {  
    font-size: 80%;  
}  
  
h1 {  
    font-size: 3em;  
}
```

# Grassetto & corsivo

- ▶ La proprietà CSS “**font-weight**” definisce il “peso” (lo spessore...) di un font. I valori più comuni sono “bold” (grassetto) e “normal”. Esempio: (aggiungete il bold a Pascoli e Ungaretti)

```
#pascoli{  
    font-weight:bold;  
}
```

- ▶ La proprietà “**font-style**” definisce lo “stile” (l’inclinazione...) del testo. I valori più comuni sono “italic” (corsivo) e “normal”.

Esempio: (aggiungete il corsivo ai titoli delle poesie)

```
.poesia{  
    font-style:italic;  
}
```



# Allineamento orizzontale

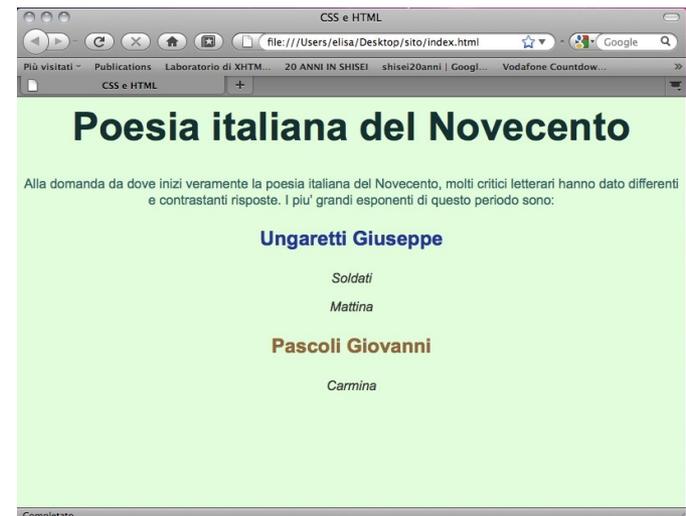
- ▶ Con la proprietà “**text-align**” si gestisce l’allineamento del testo all’interno di un elemento di blocco (ad esempio un paragrafo).
  - ❑ “left” (allineato a sinistra),
  - ❑ “right” (allineato a destra),
  - ❑ “center” (centrato),
  - ❑ “justify” (giustificato).

Esempio: (aggiungete l’allineamento centrale al vostro body)

```
body{  
    text-align:center;
```



- ▶ Sul Web, il testo allineato a sinistra risulta più leggibile del testo giustificato.



# Maiuscole & minuscole

- ▶ La proprietà CSS “**font-variant**” valorizzata con (= con valore) “`smallcaps`” converte il testo in maiuscoletto.
- ▶ La proprietà CSS “**text-transform**” può avere tre valori:
  - “`capitalize`” (rende tutte le iniziali maiuscole),
  - “`uppercase`” (tutto il testo maiuscolo)
  - “`lowercase`” (tutto il testo minuscolo)

Esempio: (aggiungetela proprietà `uppercase` a pascolie ungaretti)

```
h1{  
    text-transform:uppercase;  
}
```



# Sottolineature

- ▶ La proprietà “**text-decoration**” può avere 4 valori: “underline” (sottolineato), “overline” (riga sopra al testo), “line-through” (testo barrato), “none” (nessuna sottolineatura).

Esempio:

```
a{  
    text-decoration:none;  
}
```

- ▶  Nei doc ipertestuali gli elementi sottolineati sono normalmente riconosciuti come collegamenti (link): meglio usare le sottolineature con cautela per non confondere l'utente!

# Commenti

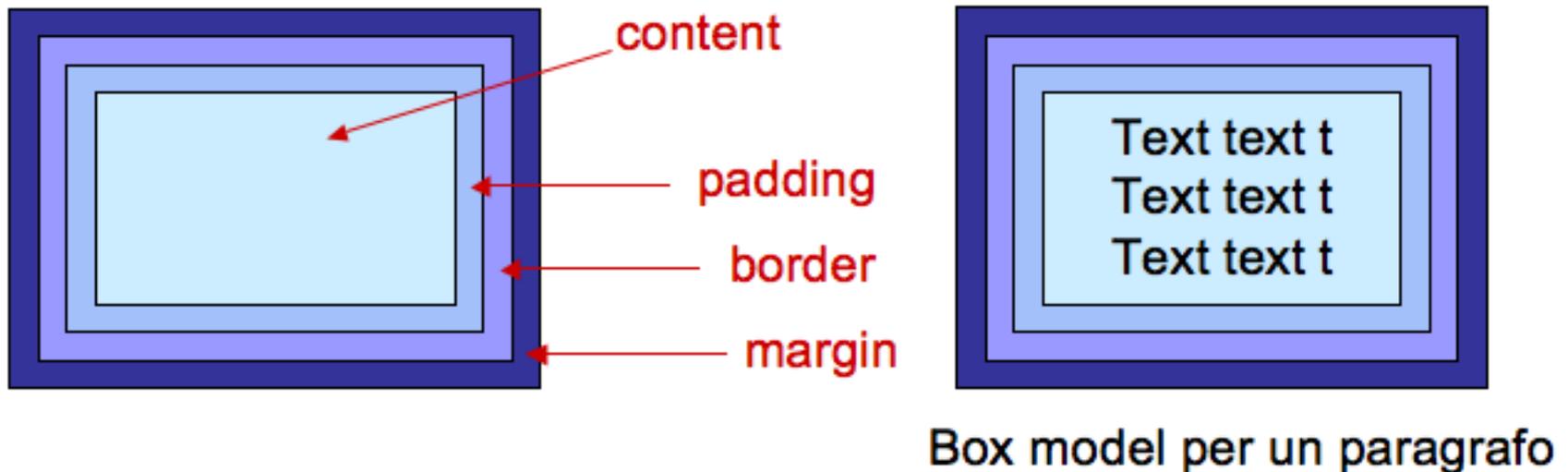
- ▶ Si può inserire un commento in qualunque punto del codice CSS con la seguente sintassi:

```
/*testode1 commento*/
```

# Il box model

# Il box model: introduzione

- ▶ Ciascun elemento di un documento ipertestuale (ad es, un paragrafo o un'intestazione ...) è inserito in un contenitore (box) a strati, le cui caratteristiche possono essere manipolate tramite apposite proprietà CSS.



# Padding

- ▶ Le dimensioni di ogni singolo lato dell'area di padding possono essere definite attraverso le proprietà “**padding-top**”, “padding-right”, “padding-bottom”, “padding-left”. Attraverso la proprietà “padding” si può definire il padding per tutti i lati contemporaneamente.
- ▶ I valori per il padding sono unità di misura, ad esempio in em o in pixel.
- ▶ Esempio: 

```
h1{  
    padding: 2em;  
}
```
- ▶ Quando si definisce uno sfondo per un elemento, con il semplice colore o con un'immagine (vedremo come ...) questo occupa l'area definita da contenuto e padding.

# Border

- ▶ La proprietà **“border-width”** (e **border-top-width**, **border-right-width**, **border-bottom-width**, **border-left-width**) permette di definire le dimensioni del bordo, come per il padding.
- ▶ La proprietà **“border-color”** (e **border-top-color**, **border-right-color**, **border-bottom-color**, **border-left-color**) permette di definire il colore del bordo.
- ▶ La proprietà **“border-style”** (e **border-top-style**, **border-right-style**, **border-bottom-style**, **border-left-style**) permette di definire lo “stile” del bordo. I valori più comuni sono: solid (continuo), dashed (tratteggiato) e dotted (puntinato).

# Border – 2

```
p {  
    border-width: 1px;  
    border-color: silver;  
    border-style: dashed;  
}
```

- ▶ E' obbligatorio definire lo stile del bordo perché le dichiarazioni relative a colore e dimensione abbiano un effetto visibile.
- ▶ Le caratteristiche dei bordi possono essere gestite in modo più sintetico attraverso la proprietà "border" (e border-top, border-right, border-bottom, border-left), in cui il primo valore corrisponde alla dimensione, il secondo allo stile e il terzo al colore.

```
p {  
    border: 1px dashed silver;  
    border-left: 2px solid maroon;  
}
```

# Margin

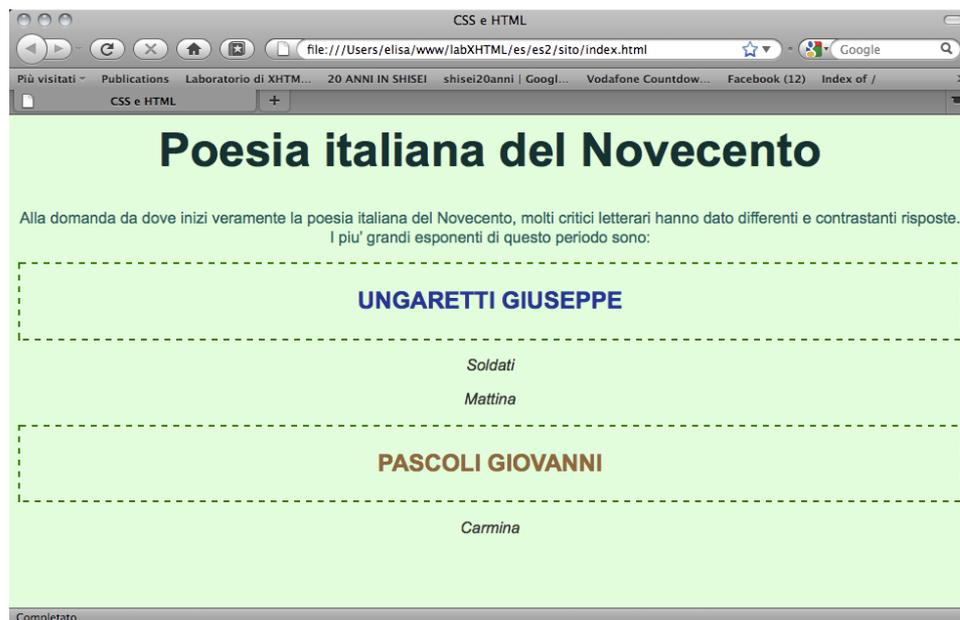
- ▶ Il “**margin**”<sup>99</sup> rappresenta l’involucro esterno del box. Come per il padding, se ne possono definire le dimensioni tramite le proprietà “margin-top”, “margin-right”, “margin-bottom”, “margin-left” e “margin”.

Esempio:

```
p {  
    border-bottom: 15px;  
}
```

# Al lavoro! – esercizio 3

- ▶ Definite una regola css per impostare le caratteristiche del box model dei paragrafi dei poeti italiani come segue:
  - Dimensione di padding 1 px
  - Bordo con dimensione 2px, di colore verde e tratteggiato;
  - Un margine di 1 px



# Link utili

- ▶ Generatore di testo: <http://www.lipsum.com/>

“Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sed tortor nunc. Nulla nibh nulla, dictum a dapibus sit amet, scelerisque eu velit. Pellentesque aliquam pulvinar blandit. Ut magna nisi, laoreet id ornare nec, ultricies a purus. Morbi sed nunc ut leo porttitor auctor sit amet nec enim. Mauris dictum urna ac est euismod a viverra leo placerat. Nullam mi erat, tincidunt et bibendum eget, bibendum ut turpis. Proin ultricies ligula quis diam aliquet ut ultrices elit dapibus. Vestibulum ultricies molestie accumsan.

Nullam sit amet tellus consectetur erat varius aliquam a at nisl. Integer in nunc ac leo euismod sagittis. Quisque lorem mauris, sagittis sit amet pellentesque vel, accumsan tempor dui. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vitae libero leo. Aliquam posuere, nulla vel imperdiet tempus, est mi volutpat sapien, quis tempus purus odio nec mauris. Ut vulputate vulputate lectus ac lobortis. Maecenas lacinia nisi id augue scelerisque sagittis. Mauris eget augue et justo faucibus facilisis vitae sed urna. Duis in nunc quam, quis fringilla diam. “