

Tecnologie Web - MFN0634

Layout

Docente

Giancarlo **Ruffo** [ruffo@di.unito.it - twitter: @giaruffo]

- ★ Selettori contestuali
- ★ Selettori di classe e di id
- ★ Definire lo stile delle sezioni
- ★ Introduzione ai layout
- ★ Proprietà float
- ★ Ridimensionamento e posizionamento

★ **Selettori contestuali**

- ★ Selettori di classe e di id
- ★ Definire lo stile delle sezioni
- ★ Introduzione ai layout
- ★ Proprietà float
- ★ Ridimensionamento e posizionamento

Combinare i selettori in base al contesto

- ★ http://www.w3schools.com/css/css_combinators.asp
- ★ Selettori di discendenti
- ★ Selettori di figli

Selettori di discendenti

★ Selettori di discendenti

- ★ Un selettore di discendente della forma "A B" seleziona un elemento B quando è un discendente arbitrario di un elemento A.

- ★ Per esempio, si considerino le seguenti regole:

```
H1 { color: red }
```

```
EM { color: red }
```

```
H1 EM { color: blue }
```

- ★ La terza regola selezionerà EM nel seguente frammento:

```
<H1>This <SPAN class="myclass">headline is <EM>very</EM>  
important</SPAN></H1>
```

- ★ La regola imposta il colore del testo sul blu ogni volta che un EM ricorre all'interno di un H1

Selettori di figli

★ Selettori di figli

- ★ Un selettore di discendente della forma "A > B" seleziona un elemento B quando è figlio diretto di un elemento A.
- ★ La seguente regola imposta lo stile di tutti gli elementi P che sono figli di BODY:

```
BODY > P { line-height: 1.3 }
```

- ★ Il seguente esempio combina i selettori di discendenti e i selettori di figli:

```
DIV OL>LI P
```

- ★ Seleziona un elemento P che è un discendente di un elemento LI; l'elemento LI deve essere il figlio di un elemento OL; l'elemento OL deve essere un discendente di un DIV.

- ★ Selettori contestuali
- ★ **Selettori di classe e di id**
- ★ Definire lo stile delle sezioni
- ★ Introduzione ai layout
- ★ Proprietà float
- ★ Ridimensionamento e posizionamento

Selettori di classe `.class`

- ★ http://www.w3schools.com/cssref/sel_class.asp
- ★ **notazione col punto (.):** la regola si applica su tutti gli elementi che specificano quel nome sull'attributo `class`.
 - ★ Es. possiamo assegnare informazioni di stile a tutti gli elementi con `class="pastoral"` come segue:
`.pastoral { color: green }`
- ★ Il seguente esempio assegna lo stile solo agli elementi `H1` con `class="pastoral"`:

`H1.pastoral { color: green }`

`<H1>Not green</H1> <H1 class="pastoral">Very green</H1>`

Selettori di ID #id

- ★ Un attributo `id` può essere usato per identificare univocamente il suo elemento.
- ★ Molto più specifico del selettore di classe
- ★ L'attributo `id` permette agli autori di assegnare un identificatore all'istanza di un elemento nell'albero del documento.
- ★ Un selettore di ID CSS contiene un `"#"` immediatamente seguito dal valore di ID.

Selettori di ID

- ★ Esempio:

```
#footer {  
    font-size: 11px;  
    margin-top: 25px  
}
```

```
<div id="footer"> testo </div>
```

- ★ **I selettori di ID hanno una precedenza più alta dei selettori di attributo.**
- ★ http://www.w3schools.com/cssref/sel_id.asp

- ★ Selettori contestuali
- ★ Selettori di classe e di id
- ★ **Definire lo stile delle sezioni**
- ★ Introduzione ai layout
- ★ Proprietà float
- ★ Ridimensionamento e posizionamento

DIV e SPAN

- ★ Permettono di strutturare le pagine HTML
- ★ `<div>` definisce un blocco generico (block level)
- ★ `` permette di specificare elementi in linea (inline)
- ★ Associando degli stili ai tag `<div>` e `` si possono ottenere suddivisioni stilistiche sofisticate.

DIV

- ★ http://www.w3schools.com/tags/tag_div.asp
- ★ Serve a delimitare aree della pagina html (tecnica del *wrapping*)
 - ★ Usare `id` per identificare quella sezione in particolare
 - ★ Buona regola farlo anche se al momento non si sa quale stile applicare

```
#main {color: #0000FF}
```

CSS

```
<div id="main">  
  <h3>This is a heading in a div element</h3>  
  <p>This is some text in a div element.</p>  
</div>
```

HTML

SPAN

- ★ http://www.w3schools.com/tags/tag_span.asp
- ★ Serve a delimitare stilisticamente delle piccole porzioni di testo *inline* quando manca un tag html che non ha la semantica desiderata
 - ★ Usare `class` per identificare la “nuova” tipologia di elemento

SPAN - esempio

```
.announcement {  
    background-color: blue;  
    color: white;  
}  
.firstword {  
    font-size: 20px;  
}
```

CSS

```
<h2>  
    <span class="announcement">Annual Members Meeting</  
span>  
</h2>  
<p>  
    <span class="firstword">The</span> Annual Members  
Meeting of the ultimate players association will take  
place on Sunday, January 28, ...  
</p>
```

HTML

Pseudo-classi

★ **pseudo-classe :first-child**

- ★ seleziona un elemento che è il primo figlio di un altro elemento.
- ★ Nell'esempio seguente, il selettore seleziona ogni elemento P che è il primo figlio di un elemento DIV. La regola sopprime l'indentazione per il primo paragrafo di un DIV:

```
DIV > P:first-child { text-indent: 0 }
```

- ★ Questo selettore selezionerà il P dentro il DIV del seguente frammento:

```
<P> The last P before the note.</P> <DIV  
class="note"> <P> The first P inside the note.</P>  
</DIV>
```


Pseudo-classi

- ★ La **pseudo-classe :link** si applica per i collegamenti che non sono ancora stati visitati.

```
A:link { color: blue }
```

- ★ La **pseudo-classe :visited** si applica una volta che il collegamento è stato visitato dall'utente.

```
A:visited { color: red }
```

Pseudo-classi

- ★ La **pseudo-classe :hover** si applica mentre l'utente indica un elemento (con qualche dispositivo di puntamento), ma non lo attiva.
- ★ La **pseudo-classe :active** si applica mentre un elemento viene attivato dall'utente.
 - ★ Per esempio, l'attimo che intercorre fra la pressione del bottone del mouse da parte dell'utente ed il suo rilascio.
- ★ La **pseudo-classe :focus** si applica mentre un elemento ha il focus (ossia accetta eventi da tastiera o altre forme di input testuale).

Altre pseudo-classi

- ★ Lo pseudo-elemento **:first-line** applica stili speciali alla prima riga formattata di un paragrafo.

```
P:first-line { text-transform: uppercase }
```

- ★ Lo pseudo-elemento **:first-letter** può essere usato per i "caratteri iniziali" e "capilettura iniziali", che sono comuni effetti tipografici.

```
P:first-letter {  
    font-size:200%; font-style: italic; font-  
    weight: bold; float: left  
}
```

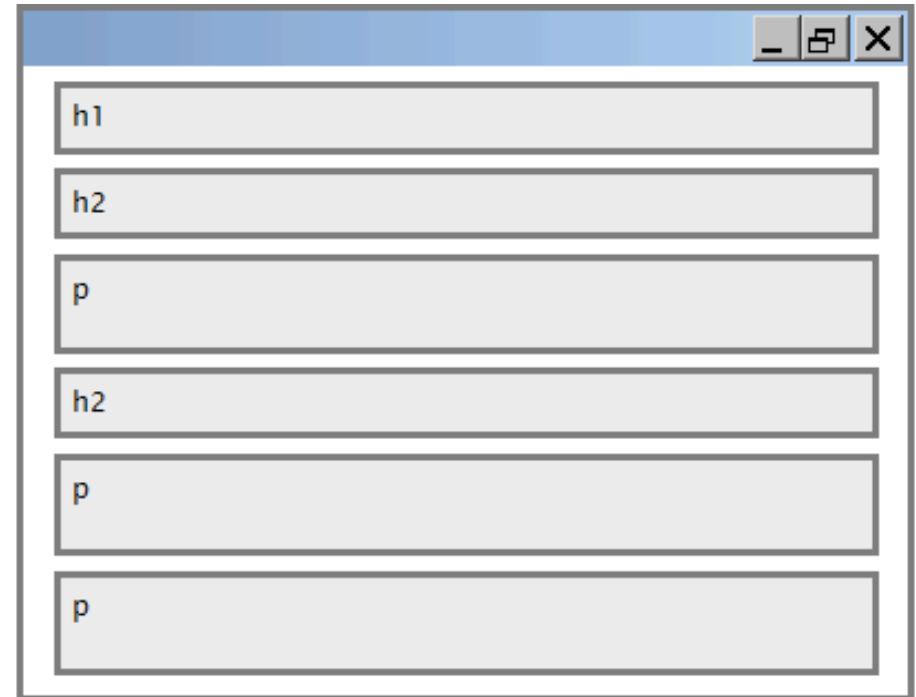
Riepilogo pseudo-classi

classe	descrizione
<code>:active</code>	Elemento attivato o selezionato
<code>:focus</code>	Elemento che ha il focus da tastiera
<code>:hover</code>	Elemento puntato dal cursore del mouse
<code>:link</code>	Link non ancora visitato
<code>:visited</code>	Link già visitato
<code>:first-letter</code>	La prima lettera di un testo dentro ad un elemento
<code>:first-line</code>	La prima riga di un testo dentro ad un elemento
<code>:first-child</code>	Elemento che appare come il primo incluso dentro ad un altro elemento
<code>:nth-child(N)</code>	L'elemento N-esimo interno ad un altro elemento "padre"

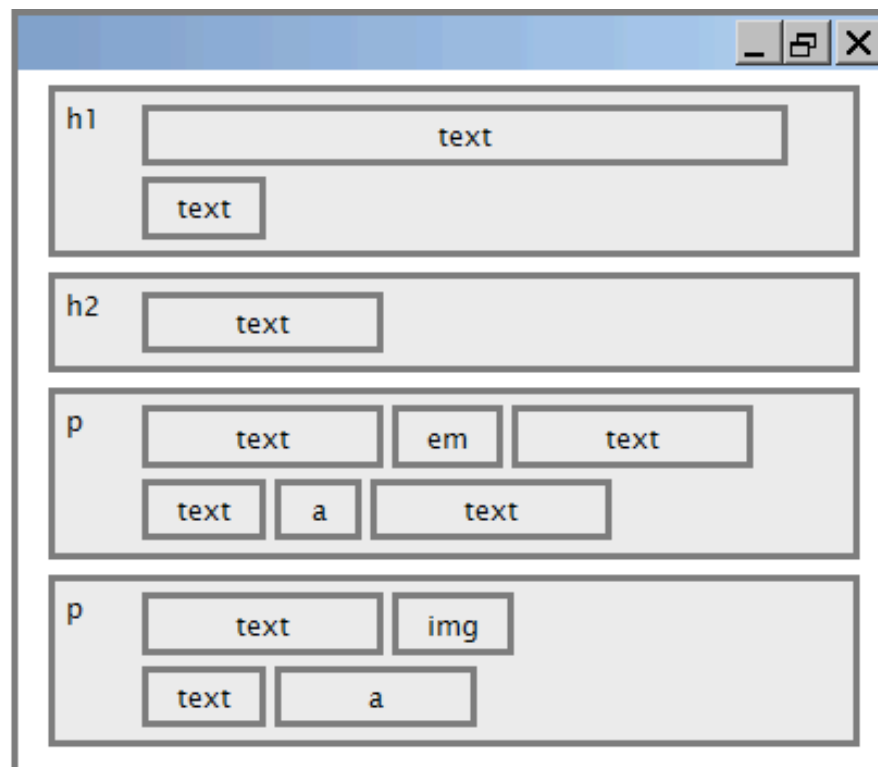
- ★ Selettori contestuali
- ★ Selettori di classe e di id
- ★ Definire lo stile delle sezioni
- ★ **Introduzione ai layout**
- ★ Proprietà float
- ★ Ridimensionamento e posizionamento

Flusso del documento

```
<body>  
  <h1>...</h1>  
  <h2>...</h2>  
  <p>...</p>  
  <h2>...</h2>  
  <p>...</p>  
  <p>...</p>  
</body>
```



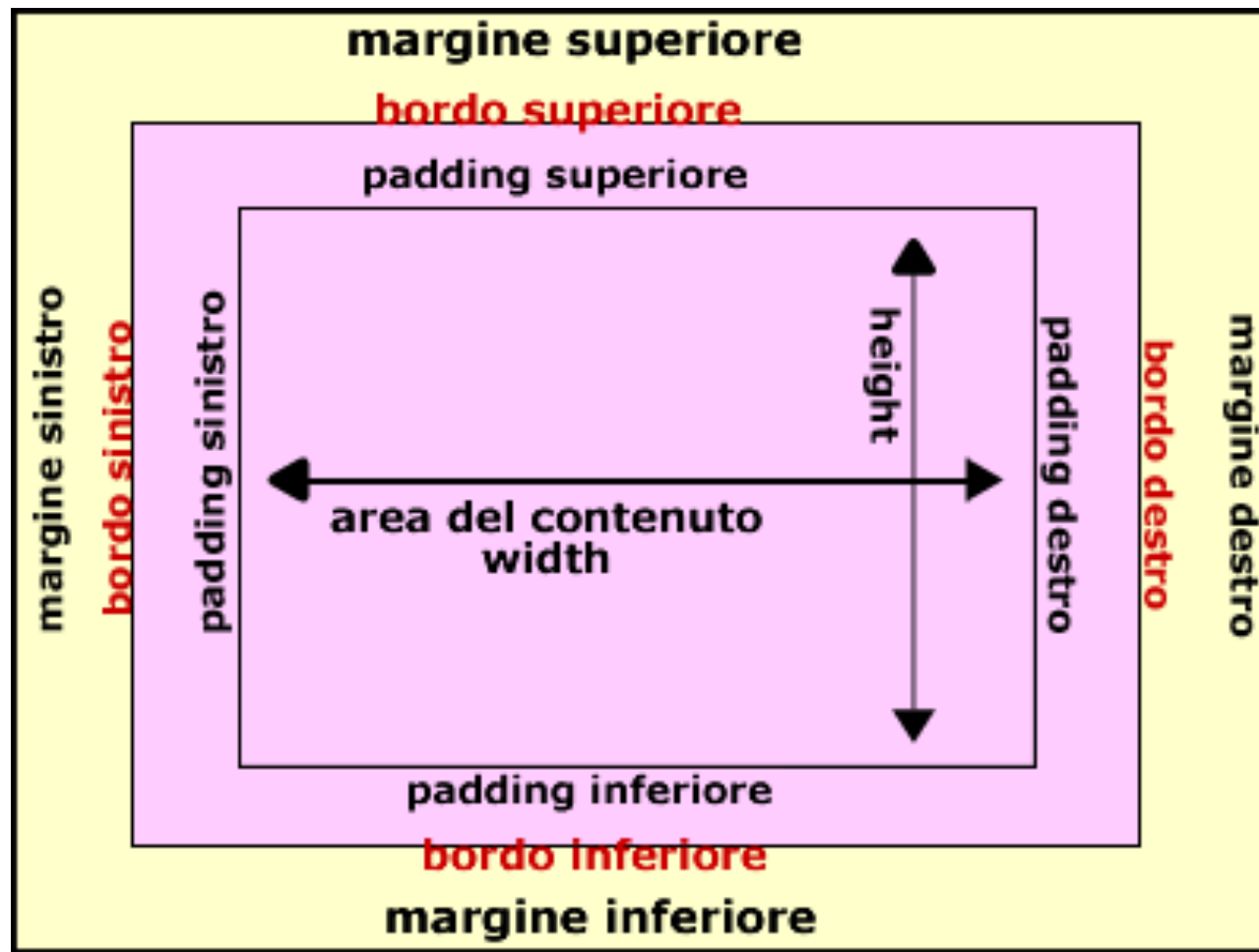
Flusso del documento completo



I box model

- ★ **Box model:** gli oggetti che si possono influenzare con i CSS sono scatole rettangolari che si possono comporre.
 - ★ Ogni elemento di HTML occupa una porzione rettangolare dello schermo, una scatola (box)
 - ★ Ogni scatola ha un'area per il contenuto (un testo, un'immagine), un padding, un bordo, un margine.
 - ★ Tutte queste sezioni rettangolari hanno proprietà che possono essere influenzate tramite i CSS
- ★ $\text{width} = \text{content width} + \text{L/R padding} + \text{L/R border} + \text{L/R margin}$
 $\text{height} = \text{content height} + \text{T/B padding} + \text{T/B border} + \text{T/B margin}$

Box Model



Gestione dei bordi

- ★ I bordi sono identificati da un <lato> che può assumere i seguenti valori:
 - ★ **top, right, left, bottom.**
 - ★ E.g., border-right specifica le proprietà del bordo destro dell'elemento.
- ★ È possibile specificare tre proprietà per ogni lato del bordo :
 - ★ **color**
 - ★ **style**
 - ★ none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
 - ★ **width**
 - ★ thin, medium, thick o una length specifica

Gestione dei bordi

★ Proprietà singole:

```
selettore {  
    border-<lato>-color: <valore>;  
    border-<lato>-style: <valore>;  
    border-<lato>-width: <valore>  
}
```

★ Esempio:

```
.div1 {  
    border-right-color: red;  
    border-right-style: dotted;  
    border-right-width: thin  
}
```

Gestione dei bordi

- ★ Sintassi abbreviata:

```
selettore {  
    border-<lato>: <valore width> <valore style> <valore  
    color>;  
}
```

- ★ Esempio:

```
.div1 {  
    border-right: thin, dotted, red;  
}
```

- ★ Esiste la possibilità di specificare le singole proprietà per l'intero bordo di un elemento:

- ★ border-color, border-style, border-width

Gestione dei bordi

- ★ Specificare tutti e quattro i lati

```
selettore {  
    border-width: <valori>;  
    border-style: <valori>;  
    border-color: <valori>  
}
```

- ★ Specificare tutto con una sola proprietà:

```
selettore {  
    border: <valore spessore>  
           <valore stile> <valore colore>  
}
```

Bordi – Esempi

```
h2 { border: 5px solid red; }
```

This is a heading.

output

```
h2 {  
  border-left: thick dotted #CC0088;  
  border-bottom-color: rgb(0, 128, 128);  
  border-bottom-style: double;  
}
```

This is a heading.

border

★ http://www.w3schools.com/css/css_border.asp

property	description
<u>border-color</u> , <u>border-width</u> , <u>border-style</u>	specific properties of border on all 4 sides
<u>border-bottom</u> , <u>border-left</u> , <u>border-right</u> , <u>border-top</u>	all properties of border on a particular side
<u>border-bottom-color</u> , <u>border-bottom-style</u> , <u>border-bottom-width</u> , <u>border-left-color</u> , <u>border-left-style</u> , <u>border-left-width</u> , <u>border-right-color</u> , <u>border-right-style</u> , <u>border-right-width</u> , <u>border-top-color</u> , <u>border-top-style</u> , <u>border-top-width</u>	properties of border on a particular side

★ Proprietà CSS3: border-radius
http://www.w3schools.com/cssref/css3_pr_border-radius.asp

Gestione del padding

- ★ **padding-bottom, padding-left, padding-top, padding-right**

- ★ valori

- ★ numerici

- ★ percentuale

- ★ non è ereditata

- ★ **padding**

- ★ proprietà a sintassi abbreviata

- ★ valgono le regole esposte per margin

Padding - esempi

```
p { padding: 20px; border: 3px solid black; }  
h2 { padding: 0px; background-color: yellow; }
```

This is the first paragraph

This is the second paragraph

This is a heading

```
p {  
  padding-left: 200px; padding-top: 30px;  
  background-color: fuchsia;  
}
```

This is the first paragraph

This is the second paragraph

Gestione dei margini

- ★ **margin-bottom, margin-left, margin-top, margin-right**

- ★ valori

- ★ numerico
- ★ percentuale
- ★ auto (di solito equivale a 0)

- ★ **margin**

- ★ proprietà a sintassi abbreviata
- ★ valori elencati in ordine orario a partire da top
- ★ `div {margin: 10px 15px 10px 20px;}`

Margini – esempi

```
p {  
  margin: 50px;  
  background-color: fuchsia;  
}
```

This is the first paragraph

This is the second paragraph

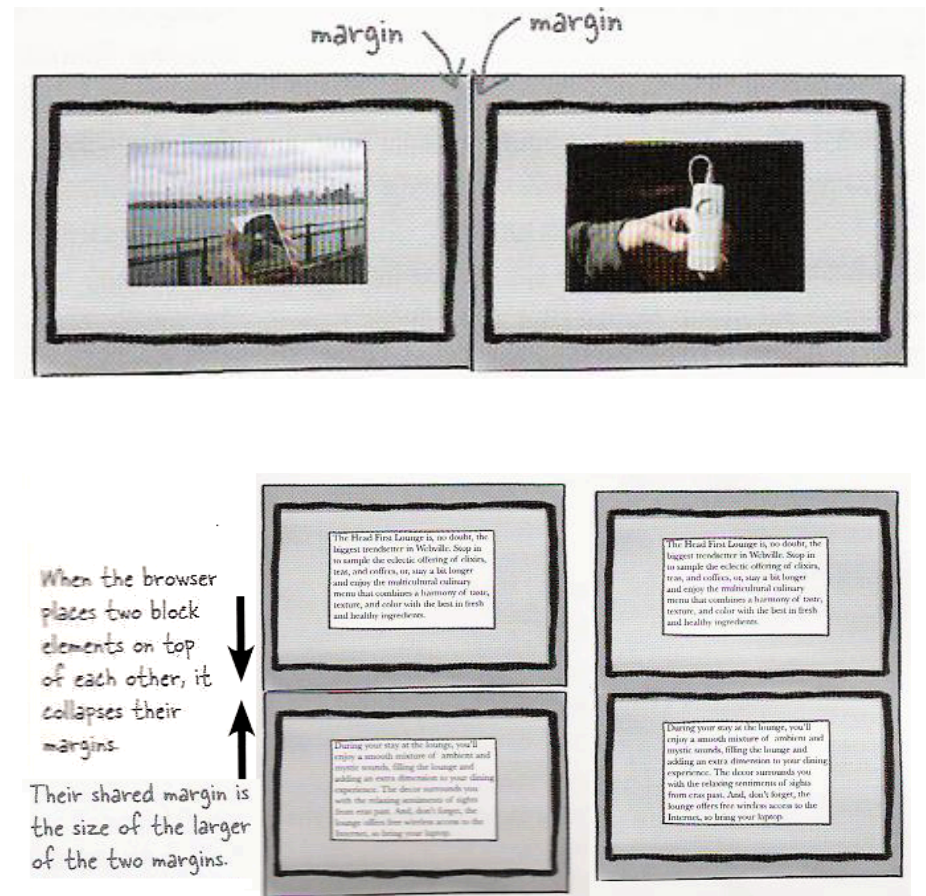
```
p {  
  margin-left: 8em;  
  background-color: fuchsia;  
}
```

This is the first paragraph

This is the second paragraph

Collassamento margini top/bottom

- ★ Quando due elementi blocco appaiono uno al top dell'altro, i loro margini vengono collassati
- ★ Il loro margine condiviso è il più grande tra i due margini individuali



Gestione dell'altezza

★ height

- ★ gestisce la distanza tra il bordo superiore e quello inferiore di un elemento
- ★ non è ereditata
- ★ sintassi
 - ★ selettore {height: <valore>;}
- ★ valori
 - ★ valore numerico
 - ★ valore in percentuale
 - ★ auto (valore di default)

★ overflow, min-height, max-height

Gestione della larghezza

★ width

- ★ specifica la larghezza dell'area del contenuto
- ★ non è ereditata
- ★ sintassi
 - ★ selettore {width: <valore>;}
- ★ valori
- ★ numerico
- ★ percentuale
- ★ auto (valore di default): se non si impostano margini, bordi e padding la larghezza dell'elemento sarà uguale all'area del contenuto dell'elemento contenitore.

★ min-width, max-width

Elemento allineato al centro (elementi block)

```
p {  
  margin-left: auto;  
  margin-right: auto;  
  width: 750px;  
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua.

output

- ★ Usatelo insieme al `width` (altrimenti occuperà l'intera larghezza della pagina)
- ★ Per centrare elementi *inline* dentro un elemento *block*, usate `text-align: center`

Domande riassuntive (1)

- ★ Se per un elemento specificate un width di 10em, un padding di 2em, un border-width di 1em, quale sarà lo spazio totale occupato dall'elemento?
 - ★ 10em
 - ★ 12em
 - ★ 14em
 - ★ 16em
 - ★ 18em
 - ★ 20em
- ★ Quale è la differenza tra margin e padding?

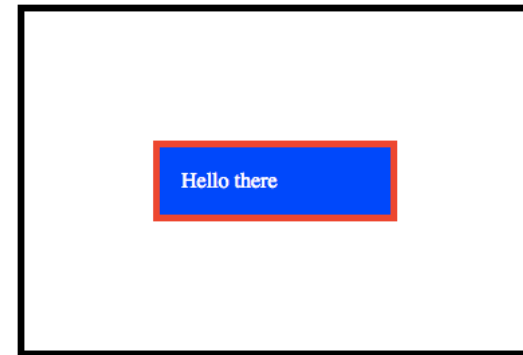
Domande riassuntive (2)

- ★ Considerate il codice seguente ed indicate quale sarà il risultato più probabile tra quelli indicati nelle figure

```
<div><p>Hello there</p></div>

div {border: 5px solid black;
      padding: 1em;
}
p {background-color: blue;
   border: 5px solid red;
   color: white;
   margin: 1em;
   padding: 4em;
}
```

a)



b)



c)



- ★ Selettori contestuali
- ★ Selettori di classe e di id
- ★ Definire lo stile delle sezioni
- ★ Introduzione ai layout
- ★ **Proprietà float**
- ★ Ridimensionamento e posizionamento

La proprietà float

```
img.headericon {  
  float: right; width: 130px;  
}
```

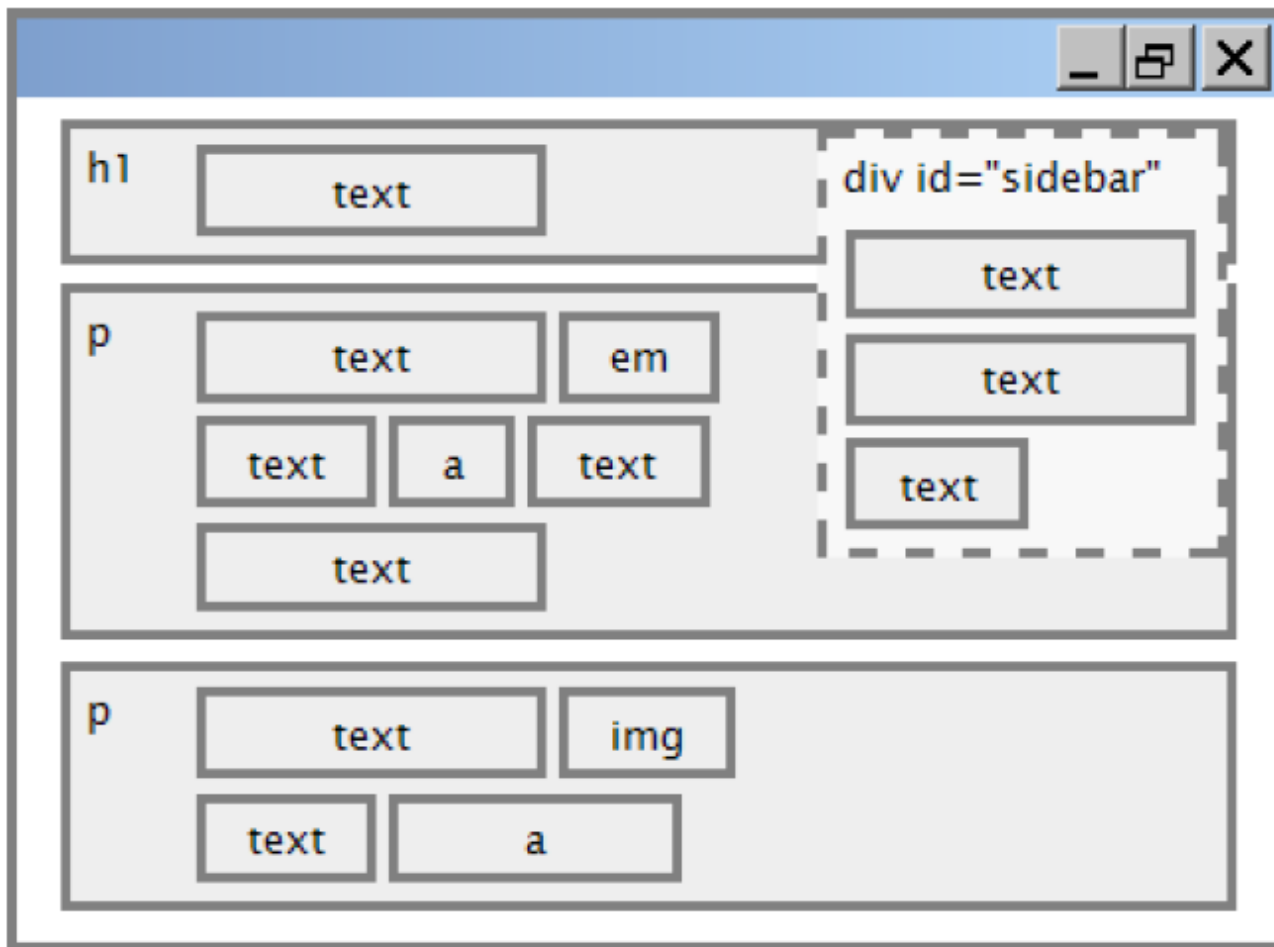
Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.



property	description
float	side to hover on; can be left, right, or none (default)

- ★ Un elemento fluttuante è svincolato dal normale flusso del documento; il testo adiacente si dispone intorno in modo automatico

Elementi fluttuanti



Errore comune con `float`: width mancante

I am not floating, no width set

I am floating right, no width set

I am floating right, no width set, but my text is very long so this paragraph doesn't really seem like it's floating at all, darn

I am not floating, 45% width

I am floating right, 45% width

- ★ Spesso i blocchi flottanti devono avere assegnare un valore all'attributo `width`
- ★ Provare su w3schools:
http://www.w3schools.com/css/tryit.asp?filename=trycss_float_elements

La proprietà `clear`

```
p { background-color: fuchsia; }  
h2 { clear: right; background-color: yellow; }
```

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation



Super Mario Fan Site!

output

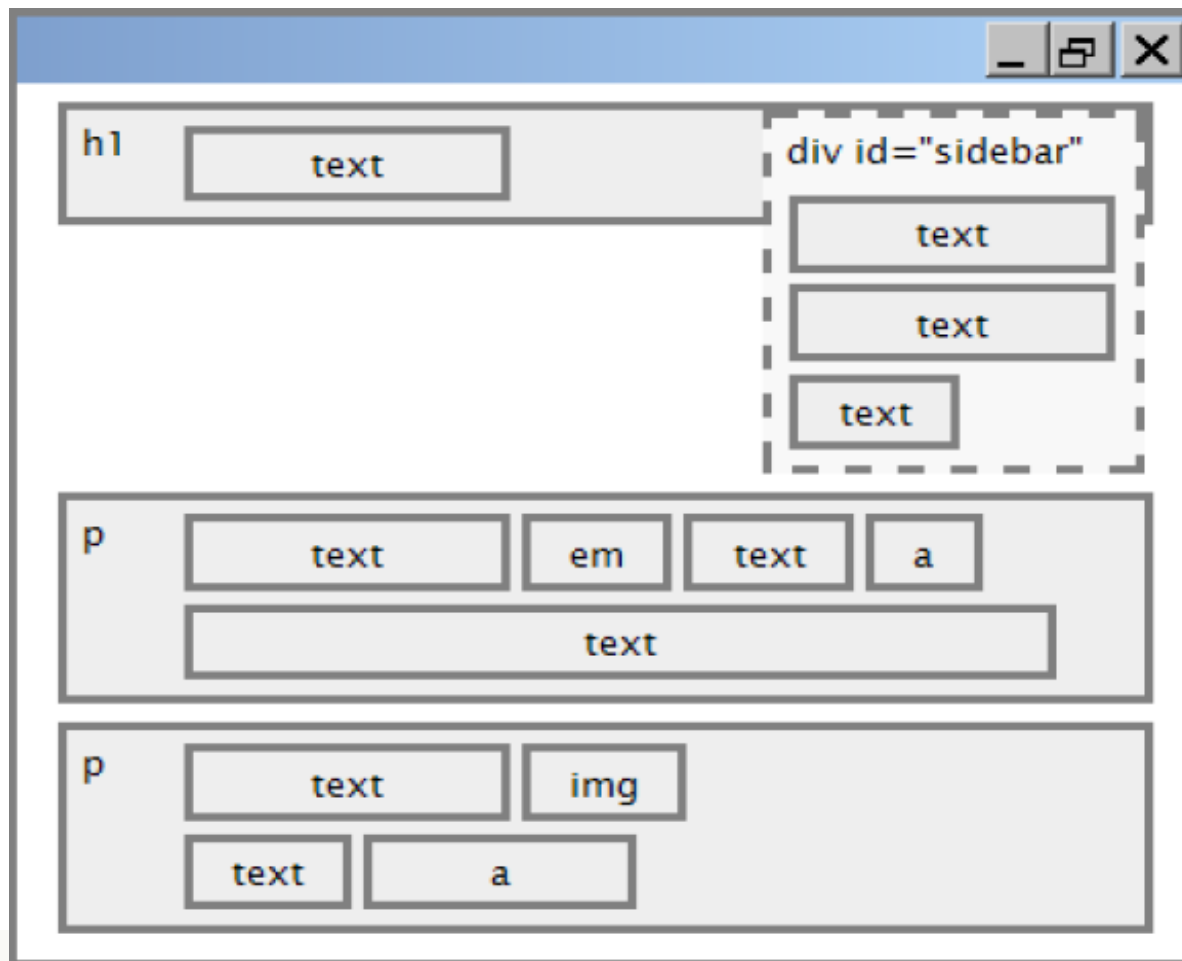
La proprietà `clear` (cont.)

proprietà	descrizione
<code>clear</code>	Impone agli elementi floating di non sovrapporsi più a questo elemento; può avere valori <code>left</code> , <code>right</code> , o <code>none</code> (default)

Clear diagram

```
div#sidebar { float: right; }  
p { clear: right; }
```

CSS



Errore comune: contenitore troppo corto

```
<p>
Mario is a fictional character in his video game
series.....</p>
```

HTML

```
p { border: 2px dashed black; }
img { float: right; }
```

CSS

Mario is a fictional character in his video game series.
Serving as Nintendo's mascot and the main protagonist
of the series, Mario has appeared in over 200 video
games since his creation.



output

La proprietà `overflow`

```
p { border: 2px dashed black;  
overflow: hidden; }
```

CSS

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



output

La proprietà `overflow` (cont.)

proprietà	descrizione
<code>overflow</code>	Specifica cosa fare se il contenuto di un elemento è troppo grande; può avere valori <code>auto</code> , <code>visible</code> , <code>hidden</code> , o <code>scroll</code>

Layout multi-colonne

```
<div>  
  <p>first paragraph</p>  
  <p>second paragraph</p>  
  <p>third paragraph</p>  
  Some other text that is important  
</div>
```

HTML

```
p { float: right; width: 25%; margin: 0.5em;  
border: 2px solid black; }  
div { border: 3px dotted green; overflow: hidden; }
```

CSS

Some other text that is important

third paragraph

second paragraph

first paragraph

output

- ★ Selettori contestuali
- ★ Selettori di classe e di id
- ★ Definire lo stile delle sezioni
- ★ Introduzione ai layout
- ★ Proprietà float
- ★ **Ridimensionamento e posizionamento**

La proprietà position

http://www.w3schools.com/css/css_positioning.asp

```
div#ad {  
  position: fixed;  
  right: 10%;  
  top: 45%;  
}
```

CSS

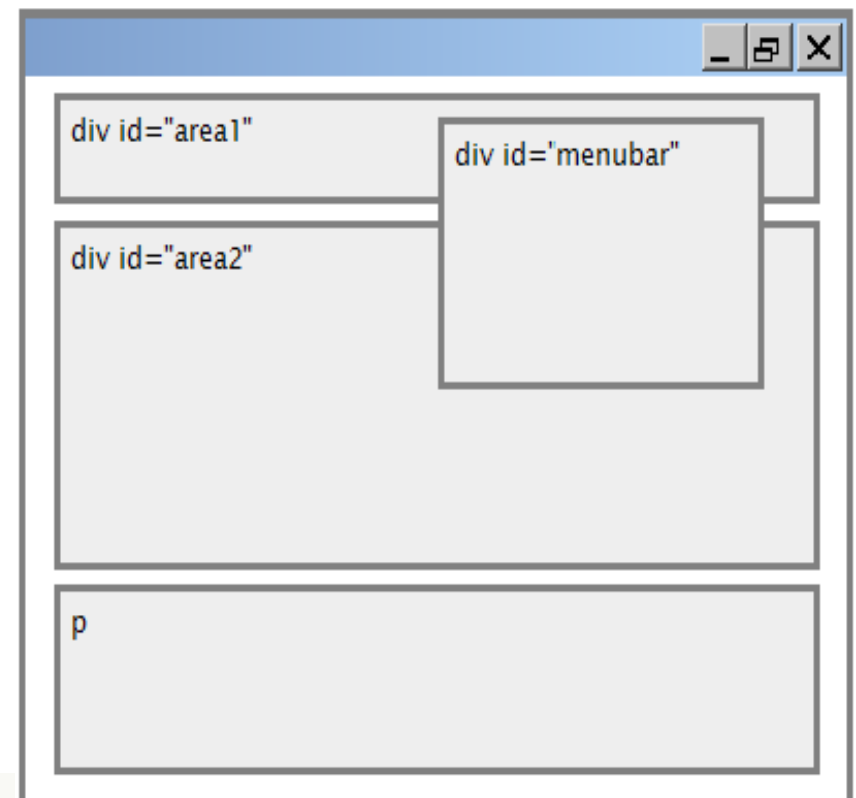
proprietà	valore	descrizione
position	static	Posizione di default
	relative	Definisce le distanze rispetto al suo posizionamento normale static
	absolute	Una posizione fissa interna all'elemento contenitore
	fixed	Una posizione fissa all'interno della finestra del browser
top, bottom, left, right	Posizioni degli angoli dell'elemento	

Posizionamento assoluto

```
#menubar {  
position: absolute;  
left: 400px;  
top: 50px;  
}
```

CSS

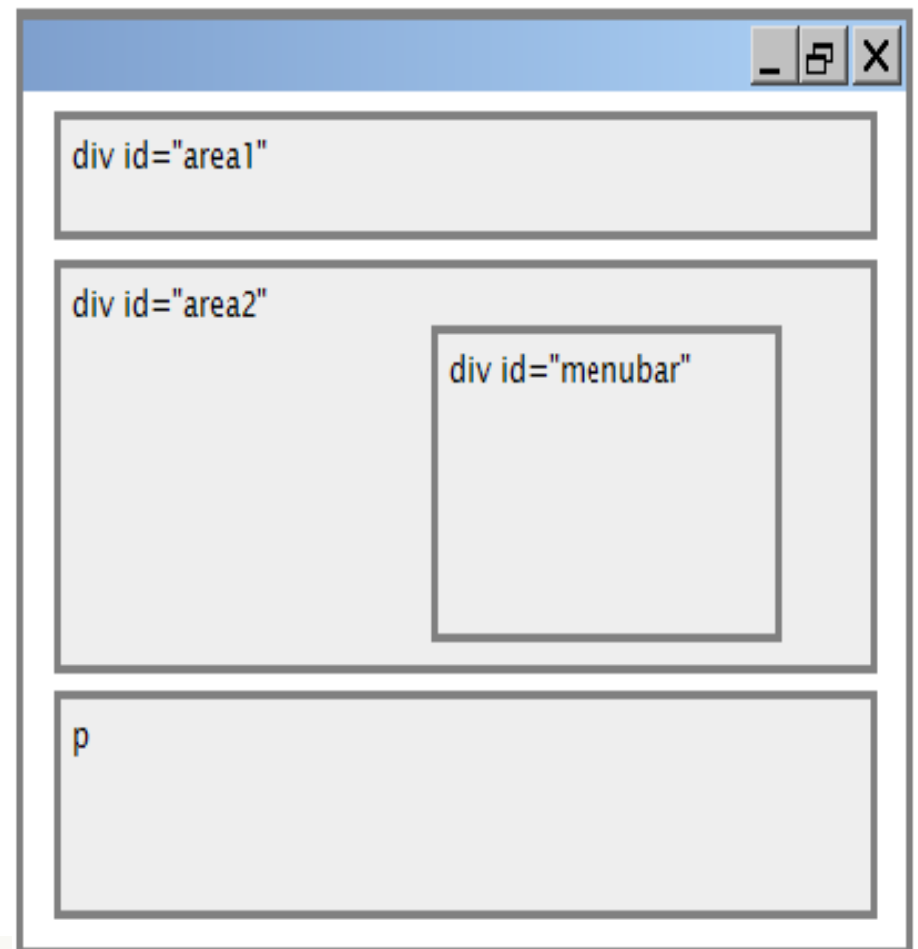
- ★ Rimuove dal flusso normale
- ★ Posizione relativa al blocco contentitore (che deve a sua volta avere `position` uguale a `relative` o `absolute`)
- ★ Posizione determinata dai valori per: `top`, `bottom`, `left`, `right`
- ★ Buona norma assegnare un valore anche a `width`



Posizionamento relativo

```
#area2 { position: relative; }  
CSS
```

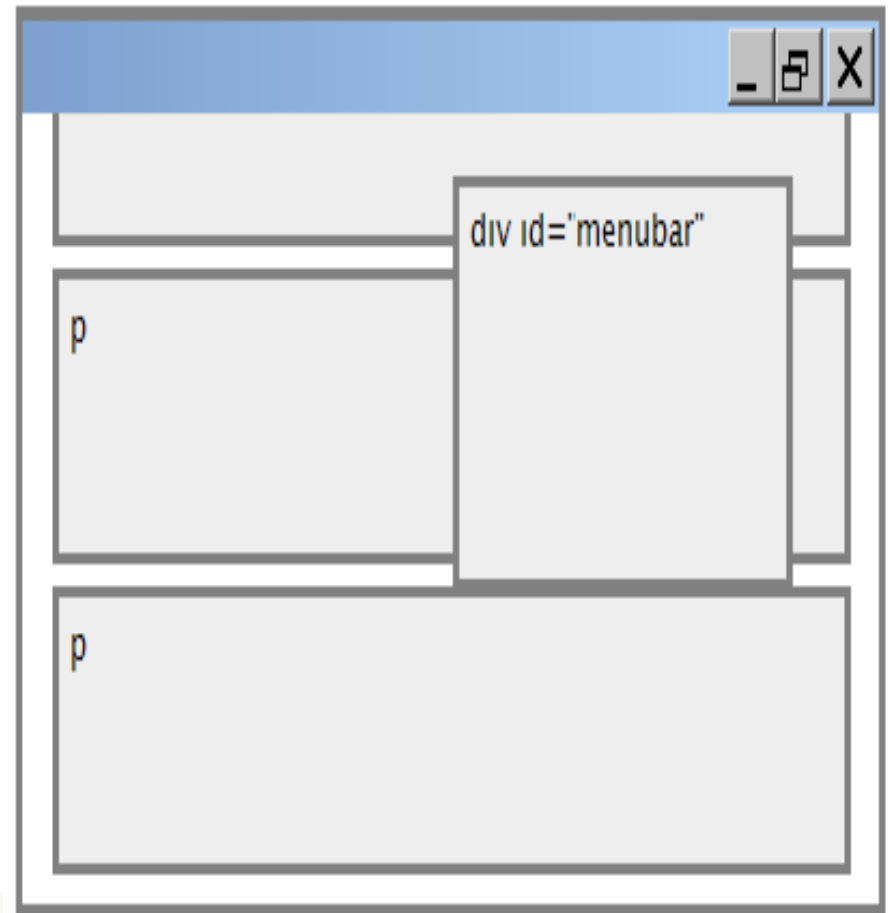
- ★ Elementi con posizionamento assoluto sono posizionati ad un determinato offset di distanza dall'angolo della pagina
- ★ Per posizionare l'elemento assoluto rispetto invece all'angolo del proprio contenitore, bisogna che il contenitore a sua volta abbia posizionamento relativo



Posizioni Fixed

```
#menubar { position: fixed; }  
CSS
```

- ★ Rimuove dal flusso normale
- ★ Posizione relative alla finestra del browser; anche quando l'utente scorre la finestra stessa, l'elemento rimane nello stesso posto



Align vs. float vs. position

1. Quando possibile, visualizza un elemento *allineando* il suo contenuto
 - ★ Allineamento orizzontale: text-align
 - ★ Applicato su un elemento block, allinea il contenuto interno (non l'elemento blocco stesso)
 - ★ Allineamento verticale: vertical-align
 - ★ Applicato su un elemento inline, lo allinea verticalmente rispetto al suo elemento contenitore
2. Se l'allineamento non restituisce l'effetto desiderato, provate a rendere l'elemento *fluttuante*
3. Se float non restituisce l'effetto desiderato, provate sull'elemento le tecniche di *posizionamento*
 - ★ Vedete il posizionamento absolute/fixed come l'ultima spiaggia: meglio non abusarne l'utilizzo

Dettagli sugli elementi inline

- ★ Le dimensioni (`width`, `height`, `min-width`, etc.) vengono ignorate per gli elementi inline
- ★ `margin-top` e `margin-bottom` sono ignorati
- ★ Ma `margin-left` e `margin-right` non sono ignorati

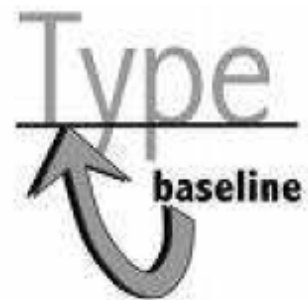
Dettagli sugli elementi inline (cont.)

- ★ La proprietà `text-align` dell'elemento block contenitore controlla la posizione orizzontale di un elemento inline al suo interno
 - ★ `text-align` non allinea elementi blocco nella pagina
- ★ Ogni proprietà `vertical-align` di un elemento inline, controlla il suo allineamento verticale rispetto al suo elemento block contenitore

La proprietà vertical-align

proprietà	descrizione
vertical-align	Specifica dove un elemento inline dovrebbe venire allineato verticalmente, rispetto al resto del contenuto presente sulla stessa linea all'interno del'elemento block

- Valori: `top`, `middle`, `bottom`, `baseline` (default), `sub`, `super`, `text-top`, `text-bottom`, or a length value or %
 - ▣ `baseline` significa che l'allineamento si riferisce all'estremità inferiore del testo, escluso le lettere fuori bordo



vertical-align esempio

```
<p style="background-color: yellow;">
  <span style="vertical-align: top; border: 1px solid
red;">
    Don't be sad! Turn that frown
     upside down!
    
    Smiling burns calories, you know.
     Anyway, look at this cute puppy;
isn't he adorable! So cheer up, and have a nice day. The
End.
  </span>
</p>
```

HTML

vertical-align example (cont.)

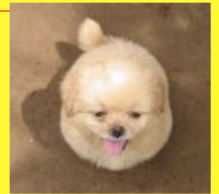
Don't be sad! Turn that frown



upside down!



Smiling burns calories, you know.



Anyway, look at this cute puppy; isn't he adorable! So cheer up, and have a nice day. The End.

Errore comune: spazio sotto l'immagine

```
<p style="background-color: red; padding: 0px; margin: 0px">  
  
</p>
```

HTML



- ❑ Nonostante padding e margin posti a 0, rimane lo spazio rosso in basso all'immagine
- ❑ Questo accade perché l'immagine è allineata verticalmente alla *baseline* del paragrafo (che è diverso rispetto al *bottom*)
- ❑ Fix: `vertical-align: bottom` (oppure `line-height: 0px`)

La proprietà `display`

```
h2 { display: inline; background-color: yellow; }
```

CSS

This is a heading **This is another heading**

output

proprietà	descrizione
display	Può variare il box model CSS di un elemento

- ❑ valori: `none`, `inline`, `block`, `run-in`, `compact`, ...
- ❑ Usatelo con cautela, perché può cambiare radicalmente il layout della pagina

La proprietà `display` (cont.)

```
<ul id="topmenu">  
  <li>Item 1</li>  
  <li>Item 2</li>  
  <li>Item 3</li>  
</ul>
```

HTML

```
#topmenu li {  
  display: inline;  
  border: 2px solid gray;  
  margin-right: 1em;  
}
```

CSS

Item 1 Item 2 Item 3

output

- ❑ Liste e altri elementi block possono essere visualizzati inline
- ❑ Scorrono da sinistra a destra sulla stessa riga
- ❑ width è determinata dal contenuto

La proprietà `visibility`

```
p.secret {  
  visibility: hidden;  
}
```

CSS

output

- ❑ Gli elementi nascosti occupano spazio, ma non vengono visualizzati
 - ▣ Se non volete che venga occupato spazio, usate `display: none`
- ❑ Sono usati per mostrare/nascondere contenuto HTML dinamico in risposta a determinati eventi (con JS, DOM, etc.)

Sezioni e layout con div

- ★ Il metodo “tradizionale”



- ★ Si usano `class` e `id` per assegnare semantica alle sezioni che contengono il contenuto in modo da definirne poi posizioni relative e dimensioni (più avanti)

HTML5

- ★ Con HTML5 si possono usare dei tag semantici più appropriati

