

# Tecnologie Web - MFN0634

## *Layout*

Docente

Marco **Botta**[ [marco.botta@unito.it](mailto:marco.botta@unito.it) ]

- \* Introduzione ai layout
- \* Proprietà float
- \* Ridimensionamento e posizionamento

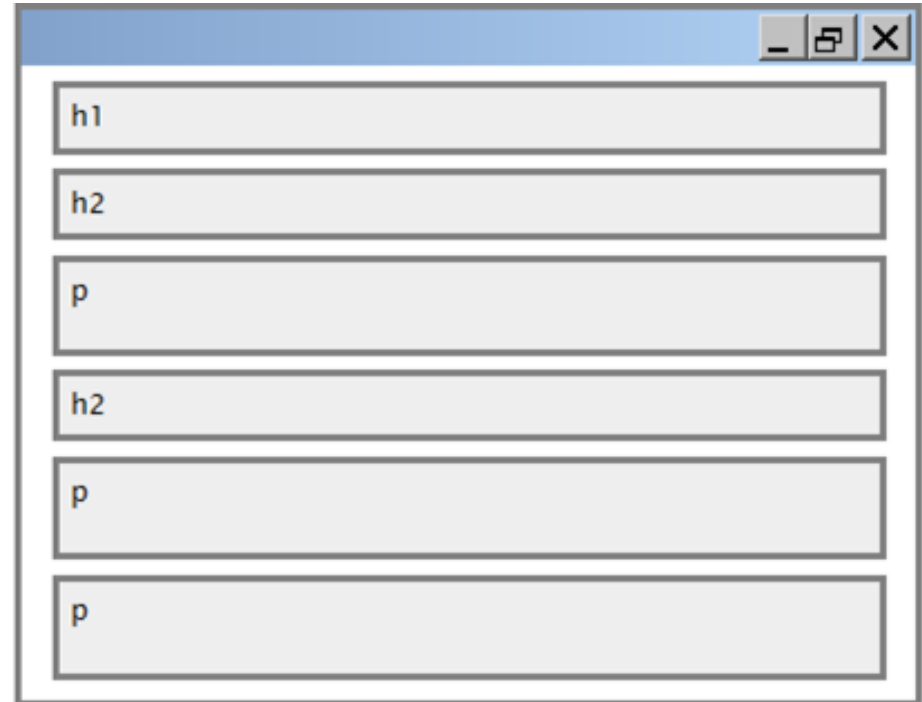
- ✧ **Introduzione ai layout**

- ✧ Proprietà float

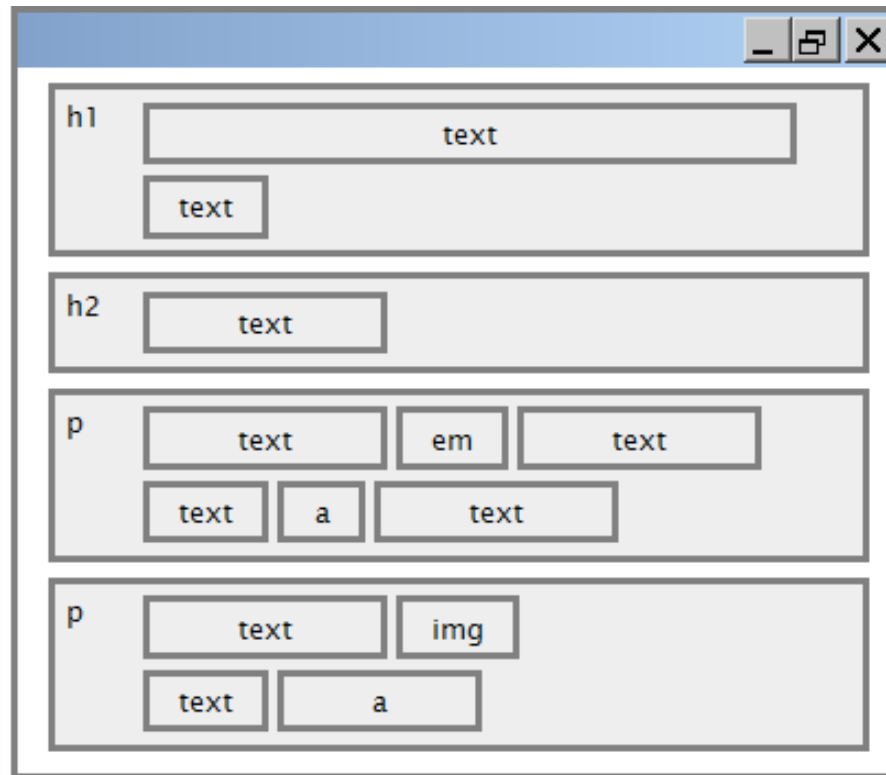
- ✧ Ridimensionamento e posizionamento

# Flusso del documento

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



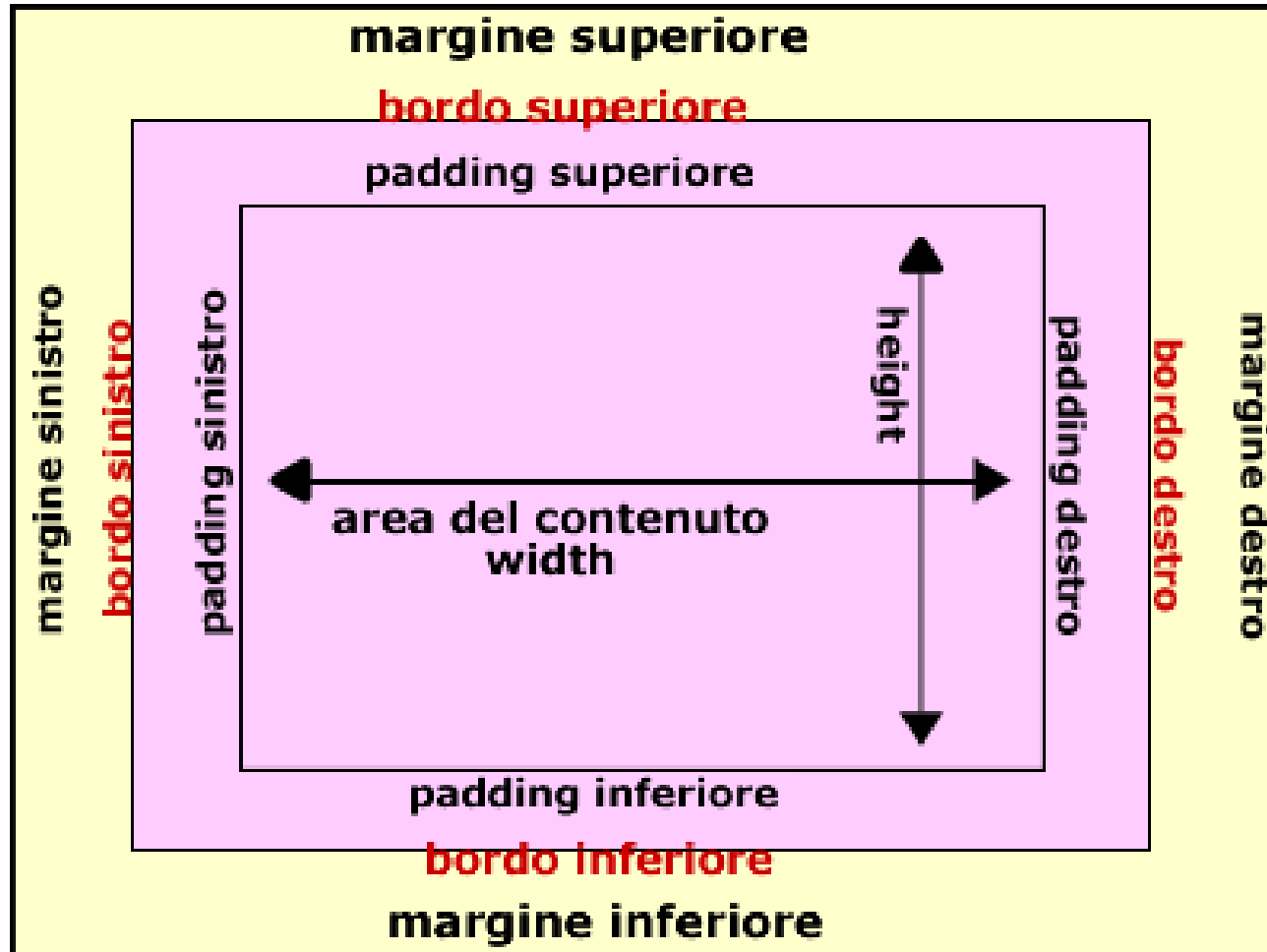
# Flusso del documento completo



# 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](http://www.w3schools.com/css/css_border.asp)

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

[http://www.w3schools.com/cssref/css3\\_pr\\_border-radius.asp](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

output

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

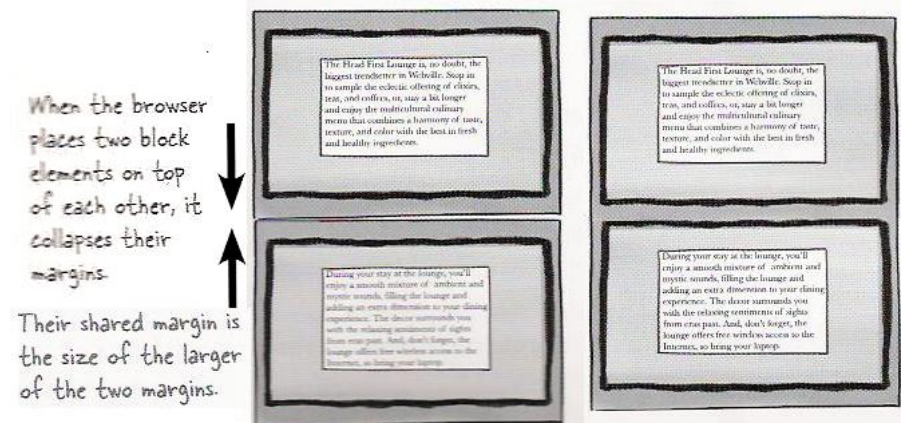
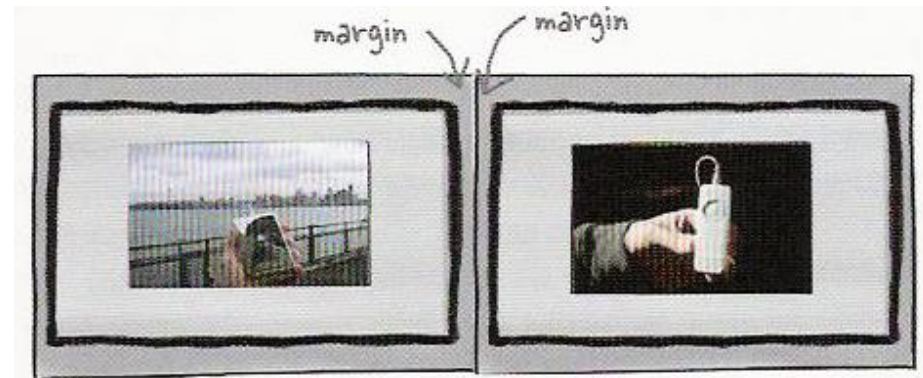
This is the first paragraph

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`

- \* Introduzione ai layout
- \* **Proprietà float**
- \* Ridimensionamento e posizionamento

# La proprietà float

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

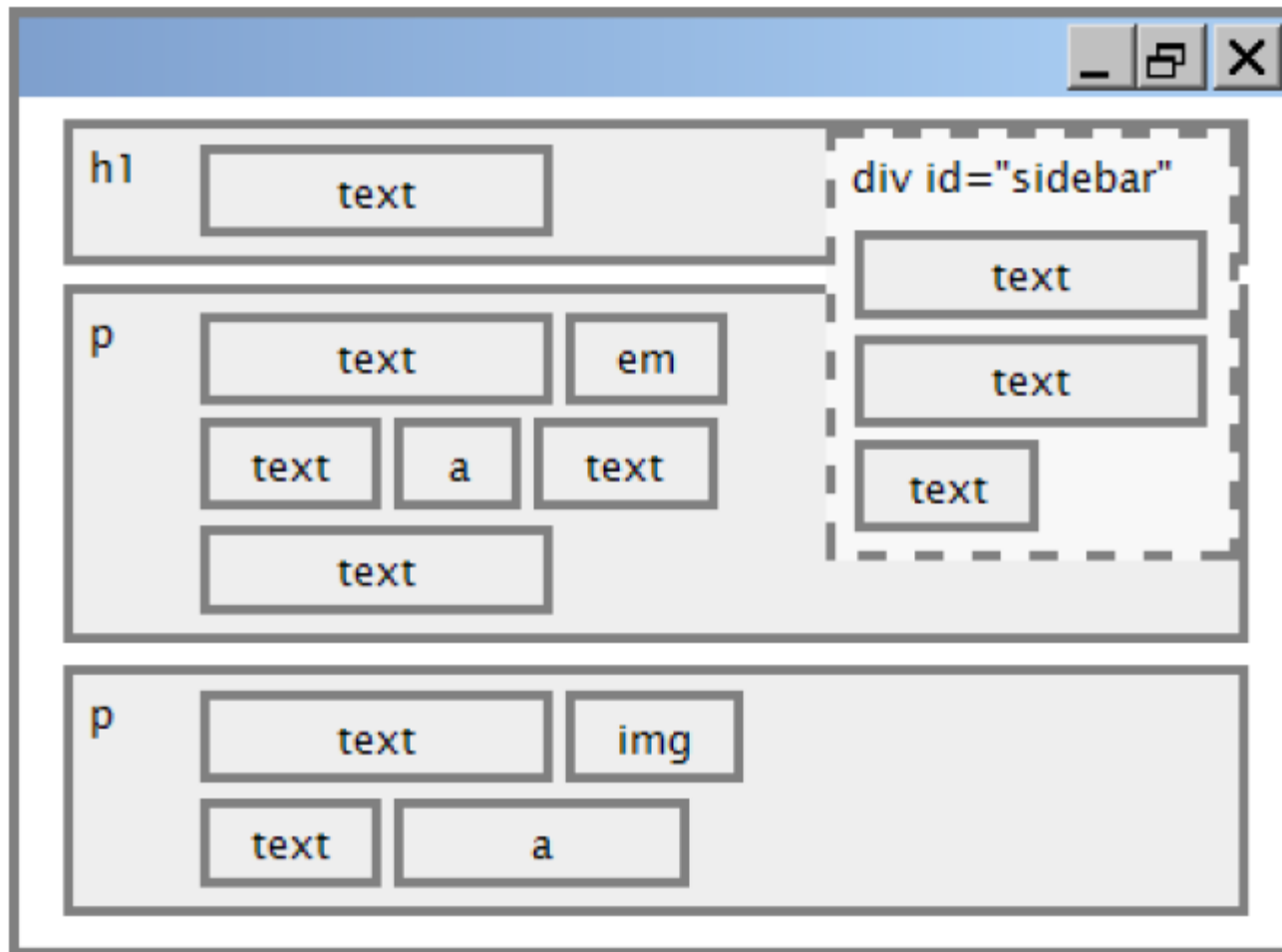
Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.



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](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*

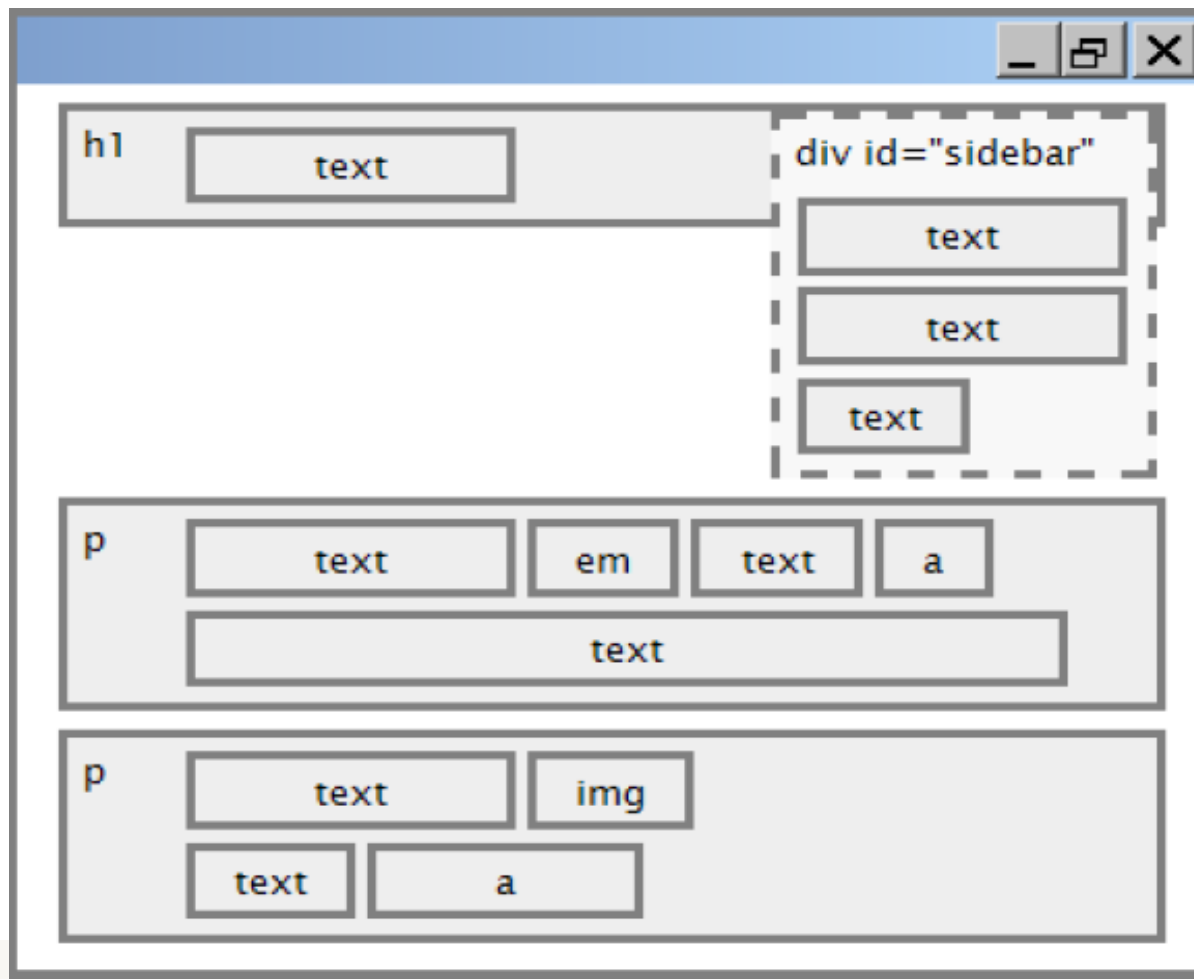
# La proprietà `clear` (cont.)

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.)

<b>proprietà</b>	<b>descrizione</b>
overflow	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>  
  Some other text that is important  
  <p>first paragraph</p>  
  <p>second paragraph</p>  
  <p>third paragraph</p>  
</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*



- \* Introduzione ai layout
- \* Proprietà float
- \* **Ridimensionamento e posizionamento**

# La proprietà position

[http://www.w3schools.com/css/css\\_positioning.asp](http://www.w3schools.com/css/css_positioning.asp)

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

CSS

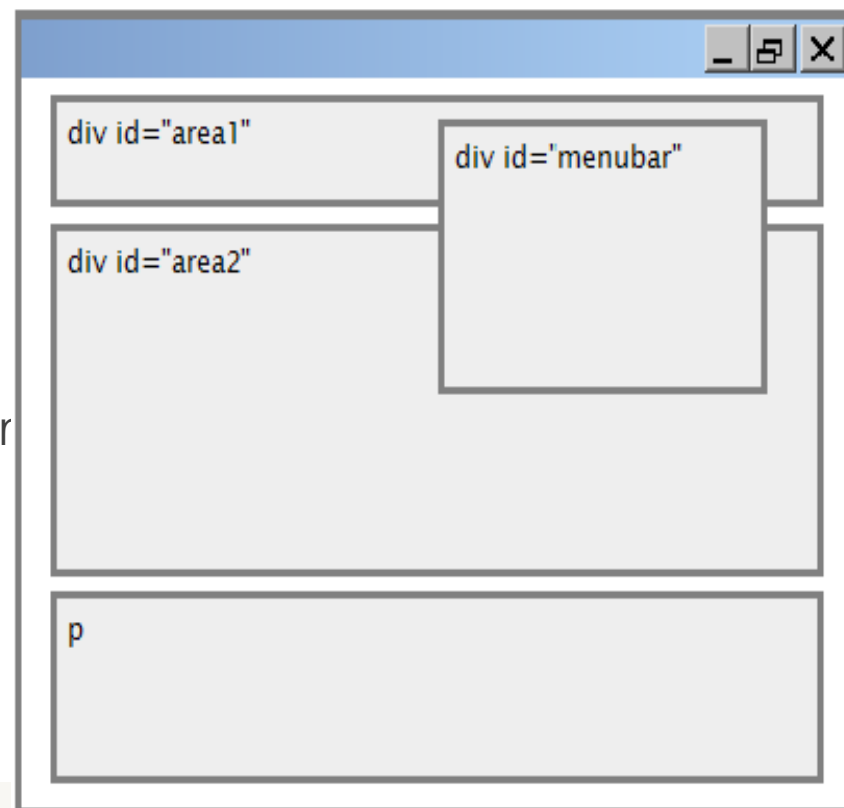
proprietà	valore	descrizione
position	static	Posizione di default
	relative	Definisce le distanze rispetto al suo posizionamento normale static
	absolute	Una posizione fissa interna all'elemento contenitore
	fixed	Una posizione fissa all'interno della finestra del browser
top, bottom, left, right	Posizioni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

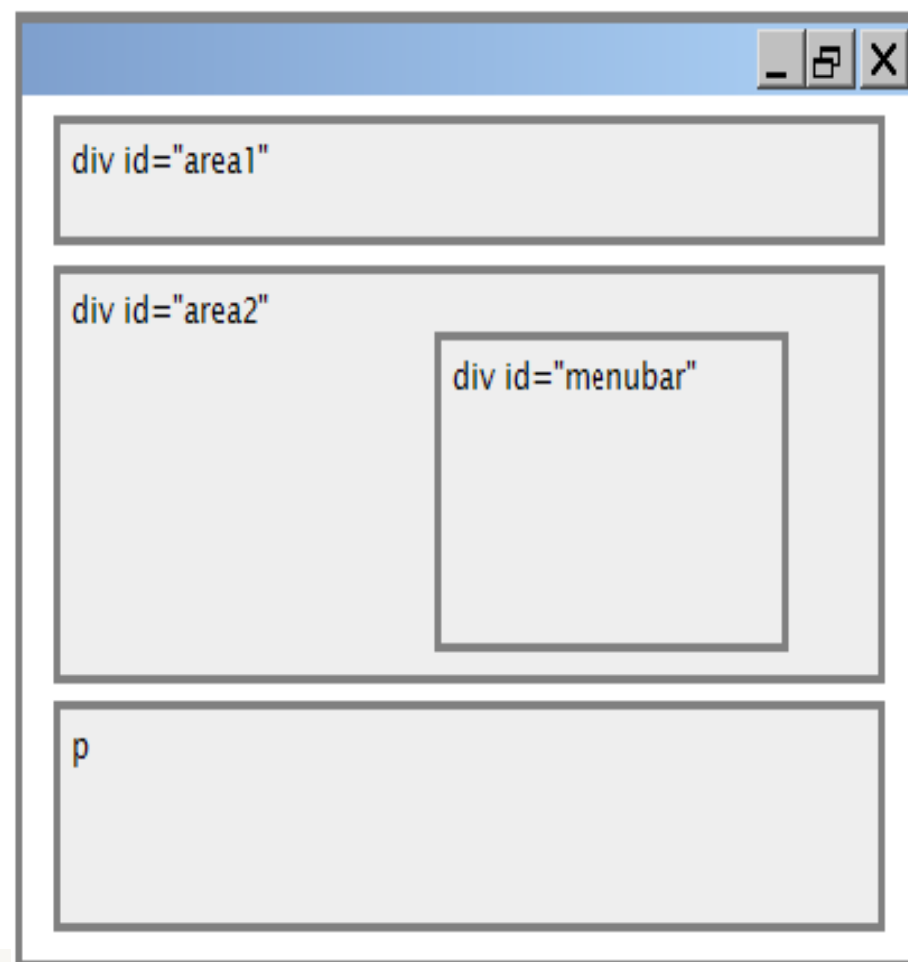


# Posizionamento relativo

```
#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

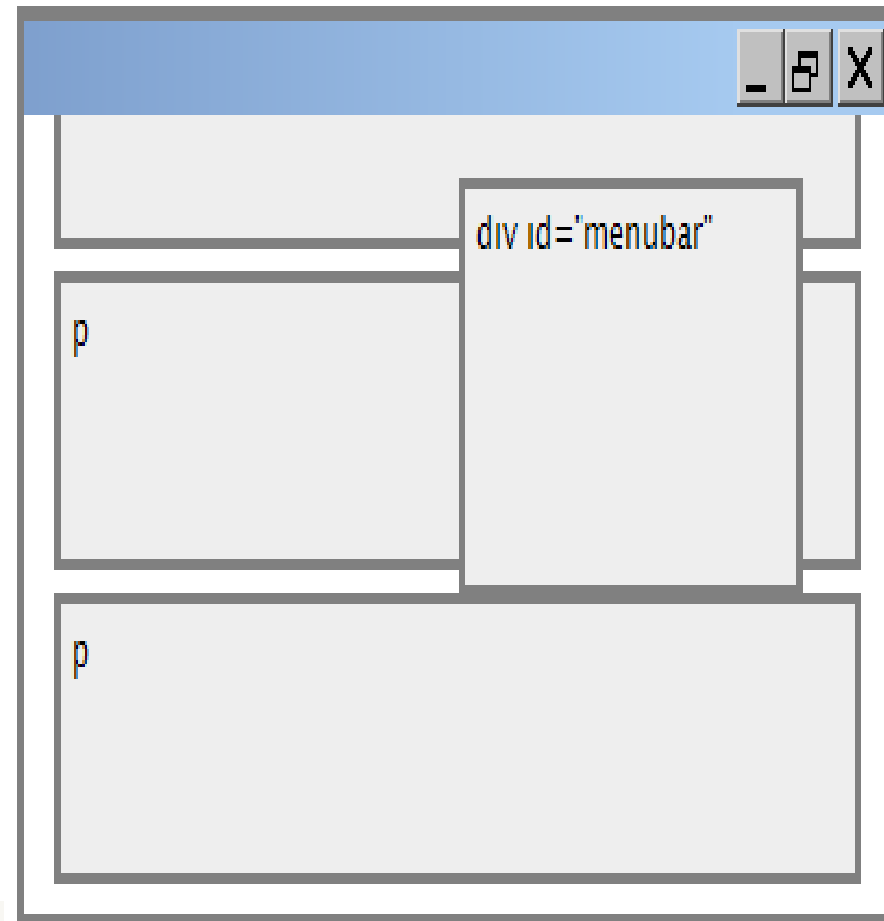


# Posizioni Fixed

```
#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.)

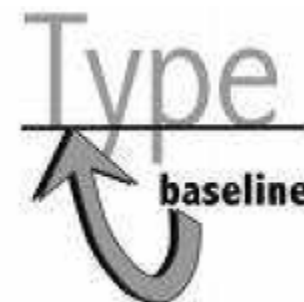
- \* La proprietà `text-align` dell'elemento block contenitore controlla la posizione orizzontale di un elemento inline al suo interno
  - \* `text-align` non allinea elementi blocco nella pagina
- \* Ogni proprietà `vertical-align` di un elemento inline, controlla il suo allineamento verticale rispetto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

