

CSS

# I FOGLI DI STILE

- HTML serve informare il browser di quali sono le componenti necessarie a mostrare un documento e ad articolare il documento in blocchi semantici.
- I fogli di stile (**Cascading Style Sheets**) definiscono come i vari elementi che compongono un documento verranno resi su un media specifico (schermo, stampante, dispositivo mobile).

# INSERIMENTO

- Un foglio di stile può essere esterno e interno:
  - È esterno un foglio di stile definito in un file separato dal documento.
  - Un foglio di stile si dice interno quando il suo codice è compreso in quello del documento.
- Un foglio esterno si carica:
  - Utilizzando l'elemento <LINK>.
  - Usando @import.
- Un foglio interno può essere compilato
  - utilizzando l'elemento <style>
  - Utilizzando l'attributo style di un singolo elemento

# FOGLI COLLEGATI

- Uso dell'elemento <LINK>:

- La dichiarazione va sempre collocata all'interno della sezione <HEAD> del documento (X)HTML:

```
<html>
```

```
<head>
```

```
<title>Inserire i fogli di stile in un documento</title>
```

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

```
</head>
```

```
<body>
```

- L'elemento <link> presenta una serie di attributi di cui è importante spiegare significato e funzione:

Attributo	Descrizione
rel	descrive il tipo di relazione tra il documento e il file collegato. È <b>obbligatorio</b> . Per i CSS due sono i valori possibili: <b>stylesheet</b> e <b>alternate stylesheet</b> .
href	serve a definire l'URL assoluto o relativo del foglio di stile. È <b>obbligatorio</b>
type	identifica il tipo di dati da collegare. Per i CSS l'unico valore possibile è <b>text/css</b> . L'attributo è <b>obbligatorio</b>
media	con questo attributo si identifica il supporto (schermo, stampa, etc) cui applicare un particolare foglio di stile. Attributo <b>opzionale</b> .

# FOGLIO COLLEGATI

- Un altro modo per caricare CSS esterni è usare la direttiva @import .
- La direttiva @import può essere usata all'interno dell'elemento <style>:

```
<style>
```

```
    @import url(stile.css);
```

```
</style>
```

- Oppure all'interno di un foglio di stile:

```
    @import url(stile.css);
```

- In entrambi i casi le regole contenute nel foglio di stile saranno importate, aggiunte a quelle già caricate.

# FOGLIO INCORPORATI

- I fogli incorporati sono quelli inseriti direttamente nel documento (X)HTML tramite l'elemento `<style>`. Anche in questo caso la dichiarazione va posta all'interno della sezione `<head>`:

```
<html>
  <head>
    <title>Inserire i fogli di stile in un documento</title>
    <style type="text/css">
      body {
        background:#FFFCC;
      }
    </style>
  </head>
  <body>
    ...
```

- `<style>` può avere due attributi:
  - type (obbligatorio)
  - media (opzionale)

# STILE IN LINEA

- L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'attributo 'style'.
- Esso fa parte della collezione di attributi (X)HTML definita Common: si tratta di quegli attributi applicabili a tutti gli elementi.
- La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea. La sintassi generica è la seguente:

```
<elemento style="regole_di_stile">
```

# REGOLE

- Un foglio di stile è costituito da una serie di regole che stabiliscono come un elemento (identificato da un selettore) viene reso su un media.



- Esempio:

```
p{  
    font-family: Verdana, sans-serif;  
    font-size:16px;  
}
```



# SINTASSI ABBREVIATA

- Ogni elemento presenta sui suoi quattro lati un certo margine rispetto a quelli adiacenti.
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- La regola sarebbe questa:

```
div { margin-top: 10px;  
      margin-right: 5px;  
      margin-bottom: 10px;  
      margin-left: 5px;  
}
```
- Sintassi abbreviata:

```
div {margin: 10px 5px 10px 5px;}
```

# SELETTORI

# ELEMENTI

CSS  
1.0

È il più semplice dei selettori. È costituito da uno qualunque degli elementi di (X)HTML.

```
h1 {color: #000000;}
```

```
p {background: white; font: 12px Verdana, arial, sans-serif;}
```

```
table {width: 200px;}
```

CSS  
1.0

È possibile nei CSS raggruppare diversi elementi al fine di semplificare il codice.

```
h1 {background: white;}
```

```
h2 {background: white;}
```

```
h3 {background: white;}
```

```
h1, h2, h3 {background: white;}
```

```
* { color: black; }
```

CSS  
2.1

# ELEMENTI

CSS  
1.0

Elementi che nella struttura ad albero di un documento siano discendenti di un altro elemento specificato nella regola.

```
div p {color: black;}  
p strong {color: red;}
```

- Nel primo esempio verranno selezionati tutti i paragrafi (<p>) discendenti di elementi <div>. Nel secondo tutti gli elementi <strong> che si trovino all'interno di un paragrafo.

CSS  
2.1

Elementi che nella struttura ad albero di un documento siano i figli diretti di un elemento.

```
body > p {color: black;}  
  
<body>  
  <p>Primo paragrafo</p>  
  <div>  
    <p>Secondo paragrafo</p>  
  </div>  
</body>
```

IE7+

# ELEMENTI

CSS  
2.1

Elementi che nel codice del documento siano immediatamente vicini (adiacenti) ad un altro.

```
h1 + p {color: red;}
```

```
<h1>Titolo</h1>
```

```
<p>Primo paragrafo</p>
```

```
<p>Secondo paragrafo</p>
```

– In base a questa dichiarazione solo il primo dei due paragrafi avrà il testo rosso..

IE7+

# SELEZIONE DEGLI ELEMENTI IN BASE AI LORO ATTRIBUTI



CSS

2.1

## Attributo semplice

```
input [ id ] {background: red;}
```

- applicherà uno sfondo rosso a tutti gli elementi input per cui sia stato impostato un attributo id, a prescindere dal valore di id.

CSS

2.1

## Attributo con valore

```
input [ id = "text" ] { backgorund: red; }
```

- applicherà un sfondo rosso a tutti gli elementi input che abbiano come valore dell'attributo id "text".

CSS

3.0

## Attributo il cui valore contiene una stringa

```
img [ alt*= "foto" ] {margin: 10px;}
```

- La regola applicherà un margine di 10px a tutte le immagini in cui l'attributo alt contiene la stringa "foto".

CSS

3.0

## Attributo il cui valore inizia con una stringa

```
img [ alt |= "figura" ] {margin: 10px;}
```

- selezionerà tutte le immagini in cui l'attributo alt inizia con la stringa "figura".

# CLASSI E ID

- In questa pagina abbiamo assegnato al paragrafo l'attributo class="testorosso":  
`<p class="testorosso">....</p>`
- Possiamo ora creare una regola e assegnargli il nome testorosso:  

```
.testorosso {  
    font: 12px arial, Helvetica, sans-serif;  
    color: #FF0000;  
}
```
- In un documento potrò avere senza problemi questa situazione:  
`<p class="testorosso">....</p>`  
`<div class="testorosso">....</div>`  
`<table class="testorosso">...</table>`  
`<p class="testorosso">....</p>`
- E l'elemento seguirà la regola definita nella classe testorosso

# CLASSI E ID

CSS  
1.0

Per definire una classe si usa far precedere il nome da un semplice punto:

```
.testorosso {  
    font: 12px arial, Helvetica, sans-serif;  
    color: #FF0000;  
}
```

CSS  
1.0

Se scriviamo:

```
p.testorosso {color: red;}
```

- lo stile verrà applicato solo ai paragrafi che presentino l'attributo class="testorosso".

CSS  
1.0

Sono possibili dichiarazioni di classi multiple:

```
p.testorosso.grassetto {color:red; font-weight:bold;}
```

- Questa regola applicherà gli stili impostati a tutti gli elementi in cui siano presenti (in qualunque ordine) i nomi delle classi definiti nel selettore.

IE7+



# CLASSI E ID

CSS  
1.0

La sintassi di un selettore ID è semplicissima. Basta far precedere il nome dal simbolo di cancelletto #:

```
#titolo {  
  color: blue;  
}
```

- assegniamo il colore blue all'elemento che presenti questa definizione:

```
<h1 id="titolo">...</h1>
```

CSS  
1.0

Come per le classi è possibile usare una sintassi con elemento:

```
p#nome_id {  
  color: red;  
}
```

- Ma non ha senso perché l'id per sua natura dovrebbe essere unico.

# PSEUDO-CLASSI

- Una pseudo-classe non definisce un elemento ma un particolare stato di quest'ultimo. In buona sostanza imposta uno stile per un elemento al verificarsi di certe condizioni.
- A livello sintattico le pseudo-classi non possono essere mai dichiarate da sole, ma per la loro stessa natura devono sempre appoggiarsi ad un selettore.

```
a:link {color: blue;}
```

- La regola vuol dire: i collegamenti ipertestuali (<a>) che non siano stati visitati (:link) avranno il colore blue.

# PSEUDO-CLASSI

CSS  
2.1

:first-child

- Seleziona e formatta un elemento che si trovi ad essere il primo elemento figlio di un altro elemento.

IE  
7+

CSS  
1.0

:link

- Si applica solo all'elemento (X)HTML <a> che abbia anche l'attributo href. Definisce lo stile per questo elemento quando il collegamento punta ad un sito o ad una pagina non ancora visitati.

CSS  
2.1

:hover

- Definisce lo stile per questo elemento quando il puntatore è sopra all'elemento.

IE  
7+

CSS  
1.0

:visited

- Si applica solo all'elemento (X)HTML <a> che abbia anche l'attributo href. Definisce lo stile per questo elemento quando il collegamento punta ad un sito o ad una pagina già visitata.

# PSEUDO-CLASSI



:first-letter

- Imposta lo stile della prima lettera di un elemento contenente del testo.



:first-line

- Imposta lo stile della prima riga di un elemento contenente del testo.



:before

```
blockquote:before {content: "Nota"}
```



:after

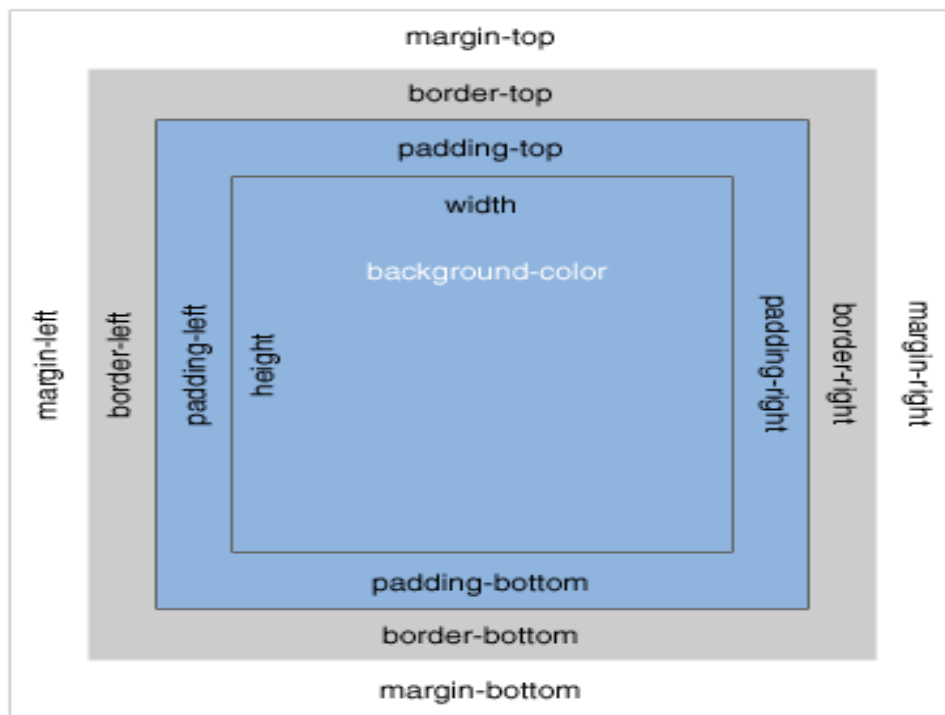
```
blockquote:after {content: "Nota"}
```



# DETTAGLI

- <http://kimblim.dk/css-tests/selectors/>

# PROPRIETÀ DEI BLOCCHI



# BACKGROUND

## ■ background-color

- Definisce il colore di sfondo di un elemento.

```
selettore {background-color: #FFF;}
```

```
selettore {background-color: transparent;}
```

## ■ background-image

- Definisce l'URL di un'immagine da usare come sfondo di un elemento.

```
selettore { background-image: url(valore); }
```

```
selettore { background-image: none }
```

## ■ background-repeat

- Consente di definire la direzione in cui l'immagine di sfondo viene ripetuta.

```
selettore {background-repeat: valore;}
```

- Valori: repeat, repeat-x, repeat-y, no-repeat

## ■ background-attachment

```
selettore {background-attachment: valore;}
```

- Valori: scroll, fixed.

# BACKGROUND

- **background-position**

- Definisce il punto in cui verrà piazzata un'immagine di sfondo.

```
selettore {background-position: valoreOriz | valoreVert;}
```

- Valori: valori in percentuale, valori espressi con unità di misura, parole chiave top, left, bottom, right, center.

- **background**

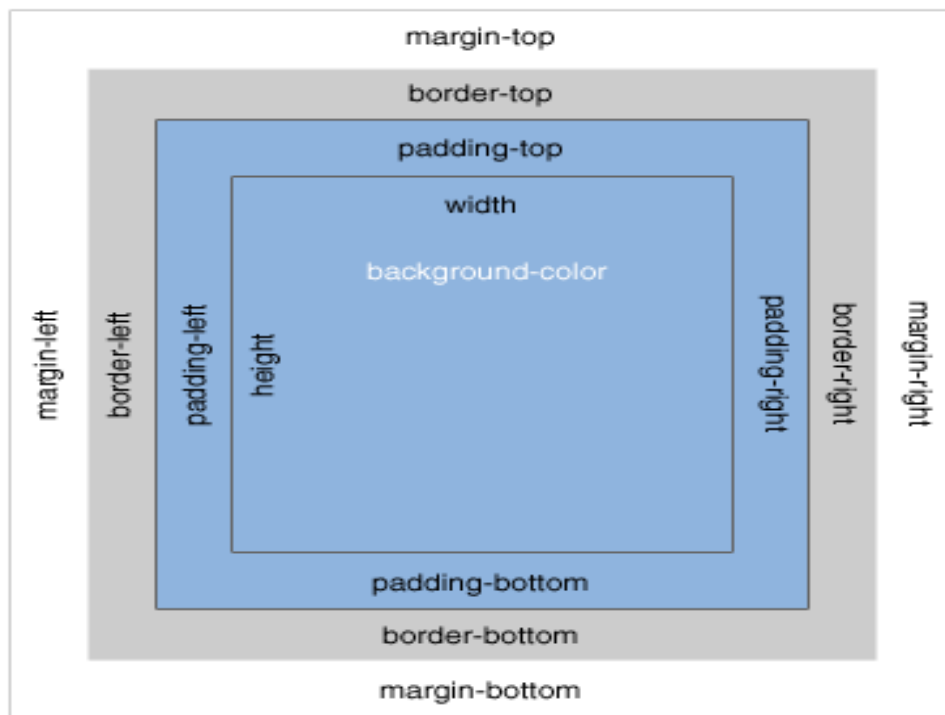
- Per essere valida, la dichiarazione non deve contenere necessariamente riferimenti a tutte le proprietà viste finora, ma deve contenere almeno la definizione del colore di sfondo.

```
selettore { background: background-color  
                background-image  
                background-repeat  
                background-attachment  
                background-position;  
}
```

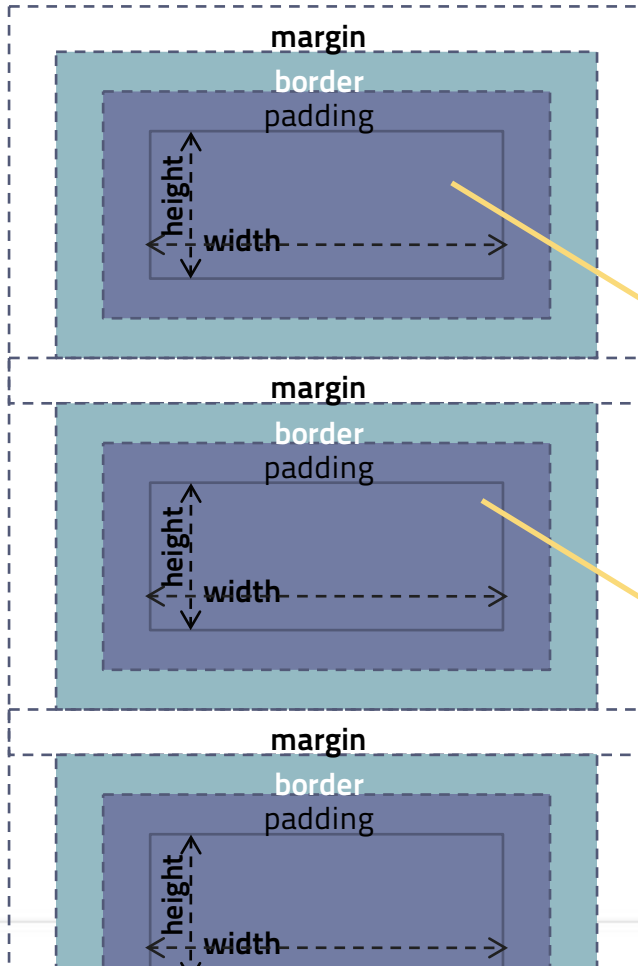


**FLOAT**

# PROPRIETÀ DEI BLOCCHI



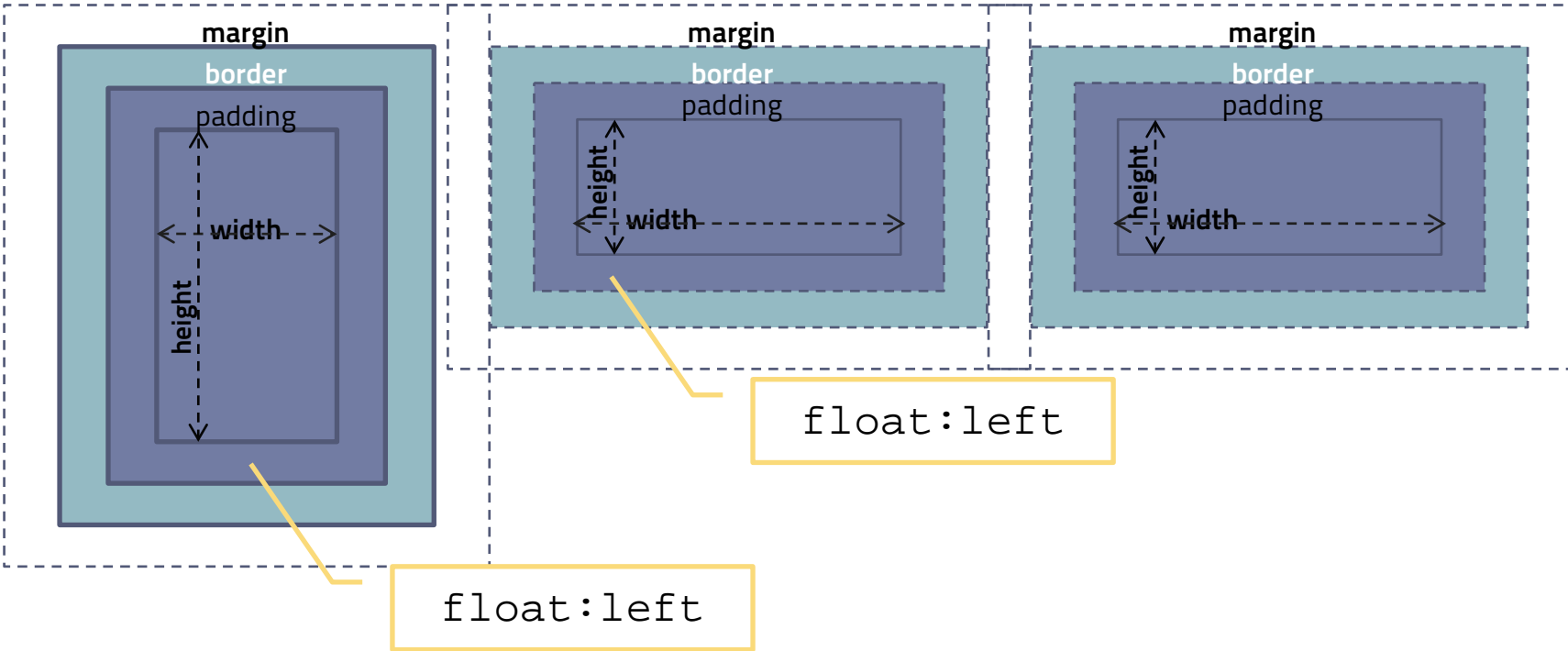
# FLOAT



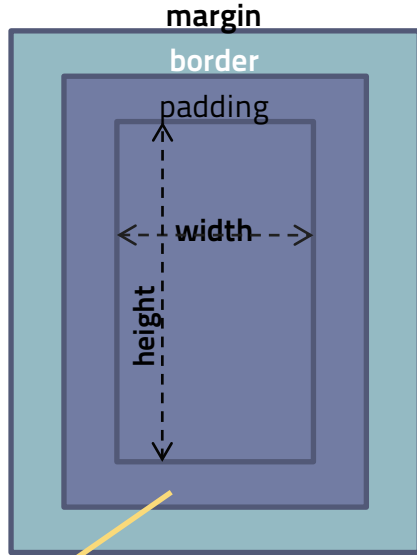
`float:none`

`float:none`

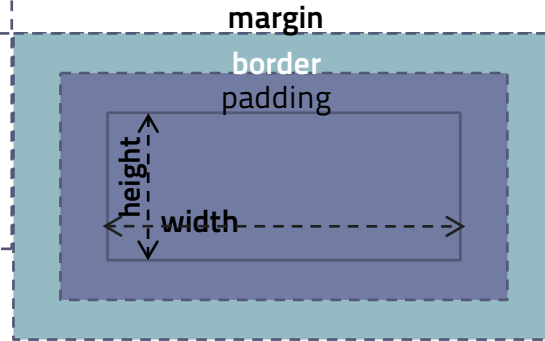
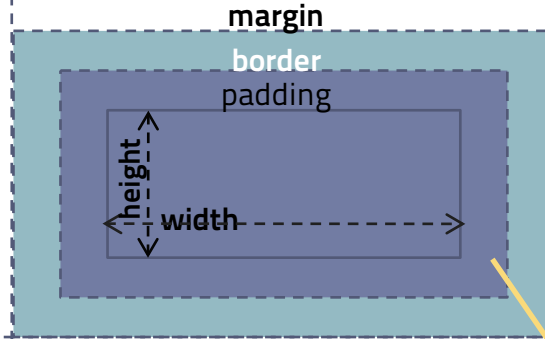
# FLOAT



# FLOAT

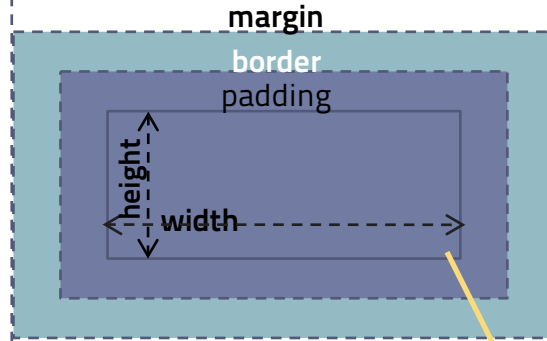
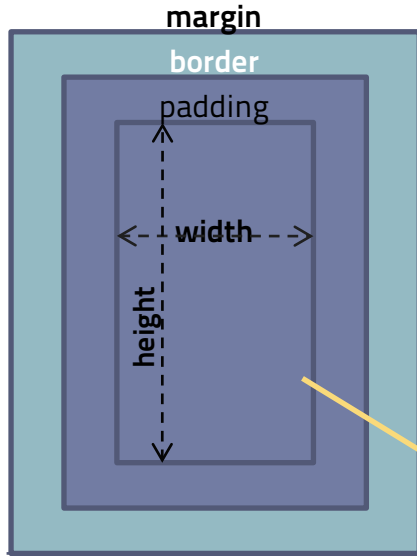


`float:left`



`float:none`

# FLOAT



```
float:none;  
clear:both;
```

```
float:left
```