LEZIONI 3 e 4

SELETTORI

SELETTORI

- I selettori sono pattern (criteri di selezione) usati per individuare gli elementi (o l'elemento) a cui si desidera attribuire lo stile.
- Seguono una sintassi precisa e i criteri di ricerca possono essere combinati fra loro.
- La forma di combinazione più semplice e un elenco di selettori separati da virgole che ci dice che le regole che seguano vanno applicate a tutti i singoli selettori compresi nella lista.

ELEMENTO

- Inserendo un selettore senza prefisso si identifica un **elemento** HTML.
- Se si inseriscono più selettori separati da virgola il gruppo di regole viene attribuito ad ogni selettore

Prova il codice

ELEMENTO

```
body, html {
    height: 100%;
}

body {
    font-family: 'Titillium Web', sans-serif;
    font-size: 16px;
    font-weight: 300;
    color: #333333;
    background-color: #ffffff;
}

h1, h2, h3, h4, h5, h6 {
    font-weight:700;
}
```

CLASSE

- Un selettore preceduto da un punto identifica una **classe**.
- Una classe è un nome che identifica un gruppo di regole di stile.
- Gli elementi della pagina che contengono quel nome nell'attributo class seguono le regole di stile che la classe identifica

Prova il codice

CLASSE

```
.rosso-grassetto {
   color: #dd0000;
   font-weight: 900;
}

.text-center {
   text-align:center;
}

.text-right {
   text-align:right;
}
```

CLASSE

- Se scriviamo:

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

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

- 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.

ID

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

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

assegniamo il colore blue all'elemento il cui attributo id è 'titolo'

- Come per le classi è possibile usare la sintassi di specifica:

```
h1#titolo {
    color: : blue;
}
```

ma non ha senso perché l'id per sua natura è un identificatore unico.

ATTRIBUTO

- Posso selezionare gli elementi in base a un qualsiasi attributo. Il selettore di attributi è compreso fra parentesi quadre.

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

indica tutti gli elementi che hanno l'attributo id.

- Posso usarlo per specificare meglio il selettore con cui vene unito:

```
input[type="text"] {
    background: red;
}
```

SELEZIONE DEGLI ELEMENTI IN BASE AI LORO ATTRIBUTI

- Attributo semplice

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

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

- Attributo con valore

```
input[type="text"] { background: red; }
```

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

- 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".

- Attributo con lista di valori separati da trattini

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

selezionerà tutte le immagini in cui l'attributo **title** contiene una lista di valori separati da trattini tra cui figura".

SELEZIONE DEGLI ELEMENTI IN BASE AI LORO ATTRIBUTI

- Attributo con valore in una lista di parole (stringhe separate da spazi)

```
input[title~="dolor"] { background: red; }
```

applicherà un sfondo rosso a tutti gli elementi input in cui l'attributo **title** contenga la parola "dolor" (sono considerate parole stringe separate da spazi).

- Attributo il cui valore inizia con una stringa

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

La regola applicherà un margine di 10px a tutte le immagini in cui l'attributo alt inizia con "foto".

- Attributo il cui valore termina con una stringa

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

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

ESEMPI

```
/* Seleziona tutti gli elementi che
  hanno l'attributo title*/
[title]{
  background-color:#ffffdd;
}
/* Seleziona tutti gli elementi a
in cui l'attributo href inizia con #*/
a[href^="#"]{
  text-decoration: underline;
  color: darkmagenta;
}
```

PSEUDO SELETTORI

PSEUDO CLASSI

- La pseudo classe è un specifica del selettore. La sintassi è:

```
selettore:pseudo-classe
```

- Usando le pseudo-classi posso attribuire un particolare aspetto ad un elemento quando si trova in un determinato stato. Esempi:
  - **:hover** indica lo stato di un elemento quando il puntatore del mouse passa sopra all'elemento stesso
  - **:focus** indica lo stato di un elemento quando ha il «il focus»
  - **:nth-child(n)** indica un elemento quando è l'ennesimo figlio del suo `parent`.

Prova il codice

http://www.sisteminterattivi.org

PSEUDO CLASSI

```
/* Seleziona ogni elemento a contenuto in un elemento li  a sua volta cont
enuto in un elemento ul che sta dentro
un nav con classe indice quando il cursore del mouse e  sopra l'elemento*/
nav.indice ul li a:hover {
  color: #dd0000;
}
/* Vengono selezionati il secondo e il terzo figlio  ci ogni elemento con
classe header-col ccontenuto in un elemento con classe header-main*/
.header-main .header-col:nth-child(2),
.header-main .header-col:nth-child(3) {
  height:40%;
  text-align: right;
}
```

PSEUDO ELEMENTI

- Lo pseudo ELEMENTO è un specifica del selettore. La sintassi è:

`selettore::pseudo-elemento`

- Usando gli pseudo-elementi posso attribuire un particolare aspetto ad un parte dell'elemento senza che la struttura HTML la differenzi. Esempi:
  - **`::after`** e **`::before`** consentono di inserire contenuto dopo o prima l'elemento identificato dal selettore
  - **`::first-line`** consente di modificare l'aspetto della prima riga di testo di un elemento
  - **`::first-letter`** consente di modificare l'aspetto della prima lettera di un elemento
  - **`::selection`** determina l'aspetto della porzione di testo selezionata dell'utente

Prova il codice

PSEUDO ELEMENTI

Nota bene

Tutti i maggiori browser, per il momento, supportano si la sintassi con doppi due punti

```
selettore::pseudo-elemento
```

che la sintassi con i due punti singoli

```
selettore:pseudo-elemento
```

mentre Microsoft Internet Explorer 8 non supporta la sintassi con i doppi due punti

SELETTORI COMPOSTI

UNIONE DI SELETTORI

- Quando un selettore viene seguito senza separatori da un selettore di classe, di id, di attributo o da uno pseudo-selettore, vengono selezionati solo gli elementi che soddisfano ad entrambi i criteri. Esempi:
  - **nav.indice**: gli elementi *nav* il cui attributo class contiene *indice*
  - **div#main**: la div con id uguale a main
  - **input[type='text']**: tutti gli elementi *input* con l'attributo type uguale a *text*

UNIONE DI SELETTORI

```
nav.indice {
  width:100%;
  position:static;
  text-align: left;
  transform: translate(0,0);
  margin: 40px auto 10px;
}
```

SELEZIONARE UN ELEMENTO CHE È DENTRO UN ALTRO ELEMENTO

```
nav.indice ol li a {
  color: #000;
}
```

Seleziona tutti gli elementi **<a>** che sono dentro un elemento **<li>** che a suo volta è dentro un elemento **<ol>** che è dentro un elemento **<nav>** con classe **indice**.

accademia di belle arti urbino

```
nav.indice ol li a { color: #000; }
```

SELEZIONARE UN ELEMENTO CHE È FIGLIO DI UN ALTRO ELEMENTO

```
nav.indice > ul > li a {
  color: #000;
}
```

Seleziona tutti gli elementi **<a>** che sono dentro un elemento **<li>** che è **figlio** di un elemento **<ul>** che è **figlio** di un elemento **<nav>** con classe **indice**.

In questo modo viene stilizzata solo la lista di primo livello, le sottoliste annidate dentro gli elementi **<li>** sono escluse.

SELETTORI

- Il link qui sotto vi rimanda alla w3school.
- È una pagina interattiva in cui potete provare tutti i selettori.

Prova il codice

PROPRIETÀ CSS

PROPRIETÀ CSS

FONT E TESTO

font-family

La famiglia di font di un testo viene impostato con la proprietà **font-family**

La proprietà **font-family** può contenere diversi nomi dei font. Se il browser non supporta la prima font, cercherà la successiva, e così via.
Il tipo di carattere preferito sarà il promo della lista, e la famiglia generica, l'ultima e verrà utilizzata se nessun altra font è disponibile.
Nota : Se il nome di una famiglia di font è più di una parola, deve essere tra virgolette, come: "Times New Roman".
L'elenco delle font è separato da virgole:

```
body {
    font-family: "Times New Roman", Times, serif;
}
```

font-style

La proprietà **font-style** è principalmente utilizzato per specificare il testo corsivo.

Tre valori possibili:

- **normal** - Il testo viene visualizzato normalmente
- **italic** - Il testo viene mostrato in corsivo
- **oblique** - Simile al corsivo, ma meno supportato

```
p.normal {
    font-style: normal;
}
p.italic {
    font-style: italic;
}
p.oblique {
    font-style: oblique;
}
```

font-size

La proprietà **`font-size`** è determina il corpo del font.

Posso assegnare:

- **una misura** – **`16px`**, **`1.2em`**
- **una percentuale** – **`100%`**
- **una parola riservata** – **`small`**, **`large`**

Le misure più utilizzate sono i pixel (suffisso px) e em (suffisso em). Nel primo caso viene indicata una misura assoluta (l'altezza che ha, in pixel, il carattere sullo schermo). L'unità di misura em è relativa alla dimensione di default che è stata assegnata al carattere.

```
p {
    font-size: 18px;
}
h1 {
    font-size : 2.4em; /* Equivalente a (2.4 * 18) 43px */
}
```

font-weight

La proprietà **font-weight** determina il *peso* del font.

Posso assegnare:

- **una misura** – **100** - **900**
- **una parola riservata** – **bold**, **normal**, **lighter**, **bolder**

```
p {
    font-weight: normal;
}
strong {
    font-weight: 900;
}
```

font-variant

La proprietà **font-variant** è determina se il font è reso in maiuscoletto.

Posso assegnare:

- **normal**
- **small-caps**

```
p {
    font-variant: normal;
}
h1 {
    font-variant : small-caps;
}
```

color

La proprietà **color** è determina il colore del testo.

Posso assegnare:

- Il nome di un colore come **red**
- Un valore esadecimale (usando il prefisso #) come **#ff0000**
- Le funzioni rgb e rgba come **rgb(255,0,0)**

```
body {
    color: #333333;
}
h1 {
    color : darkblue;
}
```

text-align

La proprietà **text-align** è determina l'allineamento del testo.

Posso assegnare i valori **left**, **right**, **center** e **justified**.

```
h1 {
    text-align: center;
}
h2 {
    text-align: left;
}
```

text-decoration

La proprietà **text-decoration** aggiunge o elimina la sottolineatura e altre decorazioni.

Posso assegnare i valori **none**, **underline**, **overline** e **line-through**.

```
.errore {
    text-decoration: line-through;
}
a {
    text-decoration: none;
}
```

text-transform

La proprietà **text-transform** determina se il testo viene reso in maiuscolo, minuscolo o normale.

Posso assegnare i valori **none**, **lowercase**, **uppercase** e **capitalize**.

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

text-indentation

La proprietà **text-indentation** determina l'ammontare del rientro della prima riga del testo.

Posso assegnare una misura (px, em, % ecc.).

```
p{
    text-indentation: 30px;
}
```

letter-spacing

La proprietà **letter-spacing** determina l'ammontare dello spazio tra i caratteri.

Posso assegnare una misura (normalmente pixel) . **0** indica la distanza normale, una misura positiva aumenta, una negativa diminuisce.

```
h1{
    letter-spacing: 3px;
}
```

line-height

La proprietà **line-height** determina l'interlinea utilizzata per il testo

Posso assegnare:

- una misura assoluta (es. **20px**)
- una misura relativa al corpo del carattere (es: **1.4**)

```
h1{
    line-height: 1.1;
}
```

text-direction

La proprietà **text-direction** determina la direzione di scrittura

Posso assegnare **rtl** (**r**ight **t**o **l**eft) o **ltr** (**l**eft **t**o **r**ight):

```
body{
    text-direction: rtl;
}
```

word-spacing

La proprietà **word-spacing** determina l'ammontare dello spazio tra le parole.

Posso assegnare una misura (normalmente pixel) . **0** indica la distanza normale, una misura positiva aumenta, una negativa diminuisce.

```
h1{
    word-spacing: -2px;
}
```

text-shadow

La proprietà **text-shadow** aggiunge un ombreggiatura al testo.

Posso assegnare tre misure e un colore:

```
h1{
    text-shadow: 2px 2px 5px red;
}
```

crea un ombra spostata a destra di **2px**, in basso di **2px**, sfumata per **5px** di colore **red**.

LISTE

REGOLE DI STILE PER LE LISTE

- In HTML, ci sono due tipi principali di liste:
  - liste non ordinate (`<ul>`) (gli elementi della lista sono contrassegnati con bullets)
  - liste ordinate (`<ol>`) (gli elementi della lista sono contrassegnati da numeri o lettere)
- Con i fogli di stile però posso modificarne completamente l'apparenza. In particolare:
  - Personalizzare il modo in cui vengono presentati gli elenchi ordinati
  - Personalizzare i bullets delle liste non ordinate
  - Sostituire gli indicatori con un immagine personalizzata

list-style-type

La proprietà **list-style-type** definisce l'aspetto del marcatore di lista .

```
ul.a {
  list-style-type: circle;
}
ul.b {
  list-style-type: square;
}
ol.c {
  list-style-type: upper-roman;
}
ol.d {
  list-style-type: lower-alpha;
}
```

La scelta dei valori possibili è lunghissima. La tabelle che segue illustra i principali

Prova il codice

Valori per list-style-type

Valore	Descrizione
`disc`	un cerchietto pieno colorato; il colore può essere modificato per tutti i tipi con la proprietà color
`circle`	un cerchietto vuoto
`square`	un quadratino
`decimal`	sistema di conteggio decimale 1, 2, 3, ...
`decimal-leading-zero`	sistema di conteggio decimale ma con la prima cifra preceduta da 0 (01, 02, ...)
`lower-roman`	cifre romane in minuscolo (i, ii, iii, iv, ...)
`upper-roman`	cifre romane in maiuscolo (I, II, III, IV, ...)
`lower-alpha`	lettere ASCII minuscole (a, b, c, ...)
`upper-alpha`	lettere ASCII maiuscole (A, B, C, ...)
`lower-greek`	Lettere minuscole dell'alfabeto greco

list-style-type

Impostando la proprietà **list-style-type** a none vene eliminata l'impostazione di default.

```
ul.unstyled {
 list-style-type: none;
 padding: none;
 margin: 0;
}
```

Prova il codice

list-style-image

La proprietà **list-style-image** consente di utilizzare un'immagine personalizzata al posto del marcatore.

```
ul {
 list-style-image: url(square.gif);
}
```

Prova il codice

DISPLAY

display

La proprietà **display** determina come viene visualizzato un elemento nel flusso di elementi che compongono la pagina HTML.

```
nav ul li {
    display: inline-block;
}
.hidden {
    display: none;
}
```

Prova il codice

Valori per display

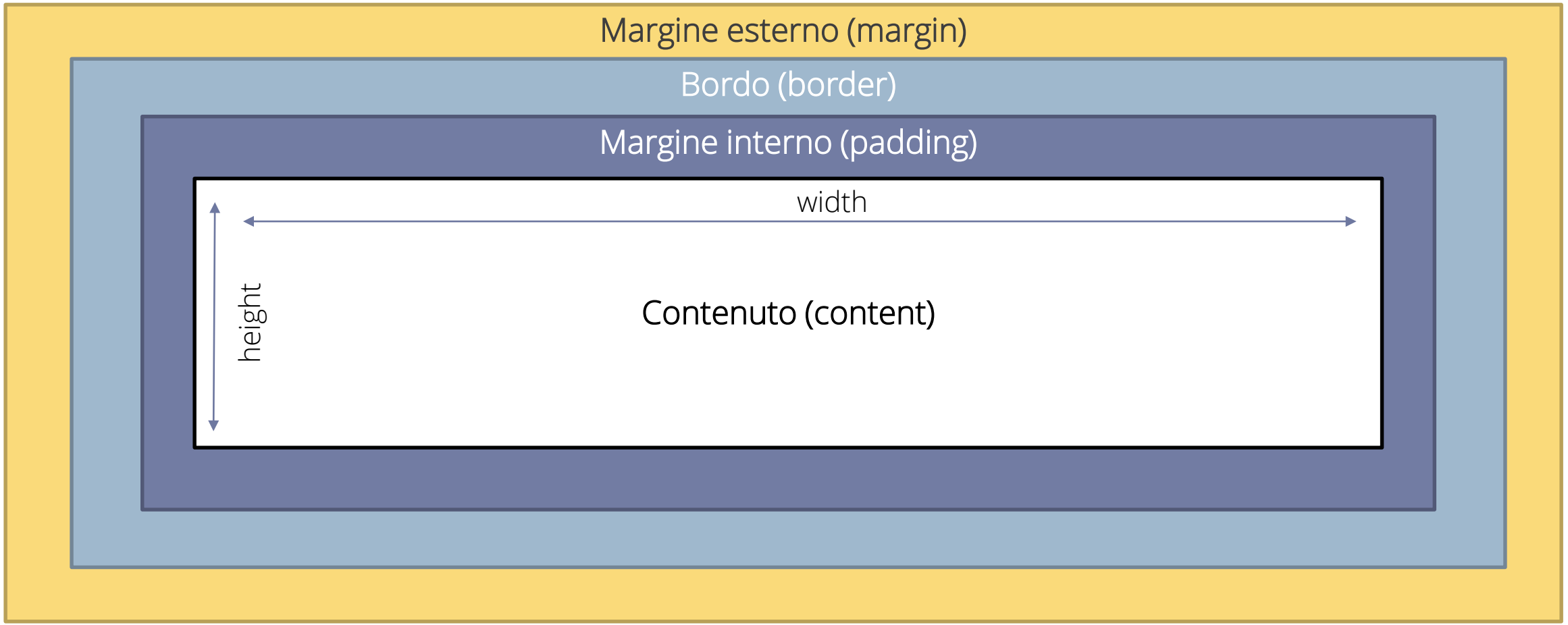
Valore	Descrizione
`block`	l'elemento viene reso come un elemento blocco
`inline`	l'elemento a cui viene applicata assume le caratteristiche degli elementi inline
`inline-block`	l'elemento può assumere, come gli elementi blocco, dimensioni esplicite (larghezza e altezza), margini e padding, ma come tutti gli elementi inline, si disporrà orizzontalmente e non verticalmente, potendo essere circondato dal testo ed essendo sensibile all'allineamento verticale
`none`	l'elemento non viene mostrato; o meglio: è come se non fosse nemmeno presente nel documento, in quanto non genera alcun box; l'uso del valore none è uno dei mezzi con cui, nei CSS, si può nascondere un elemento
`list-item`	consente di impostare per un elemento una presentazione equivalente a quella degli item di una lista
`table | inline-table | table-cell | table-row | table-row-group | table-column | table-column-group | table-header-group | table-footer-group | table-caption`	display degli elementi come elementi di una tabella.

BOX MODEL

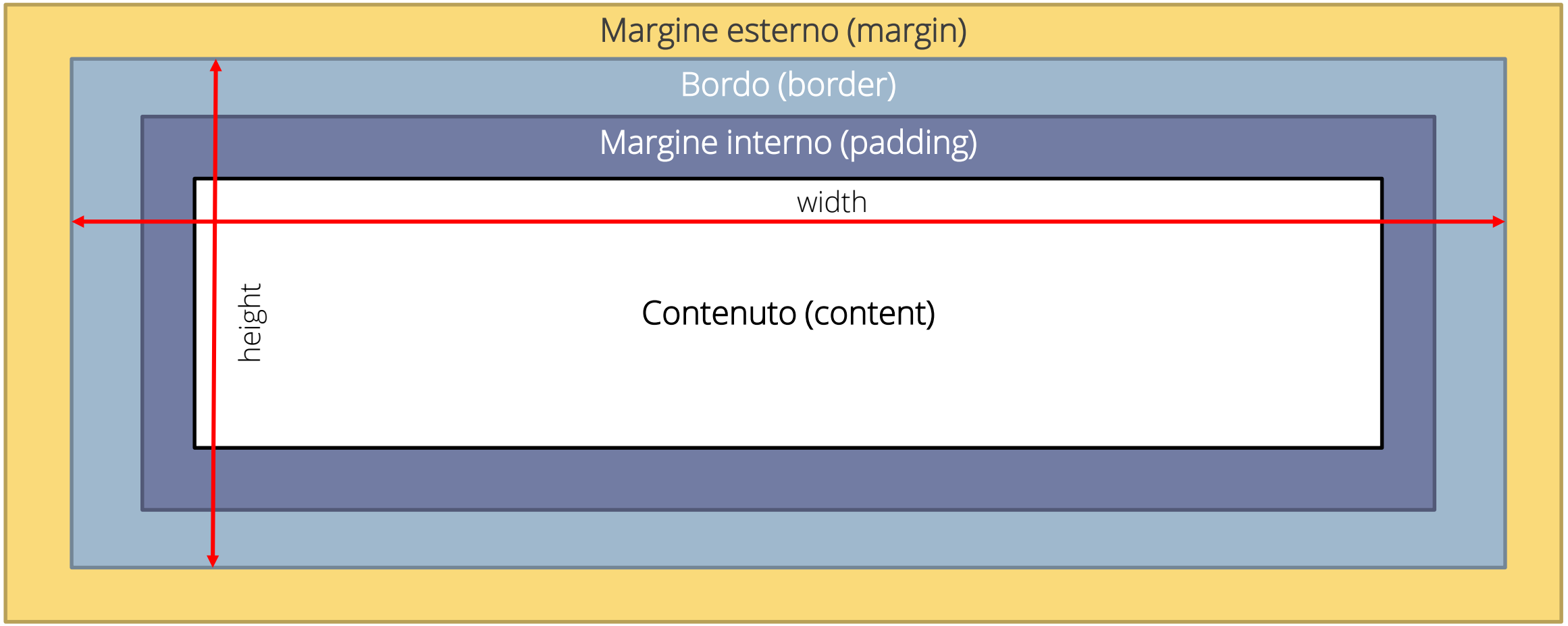
BOX MODEL

- Tutti gli elementi HTML possono essere considerati come scatole, riquadri che occupano un certo spazio sulla schermo. In CSS, il termine "box model" viene usato quando si parla di design e il layout.
- Il box model si compone di: margini, bordi, margini interni (padding), e contenuto effettivo.
- Content - Il contenuto dell'elemento, dove compaiono testo e immagini
- Padding – Margine interno attorno al contenuto.
- Border – Il bordo
- Margine – Distanza tra l'elemento e gli elementi che lo circondano

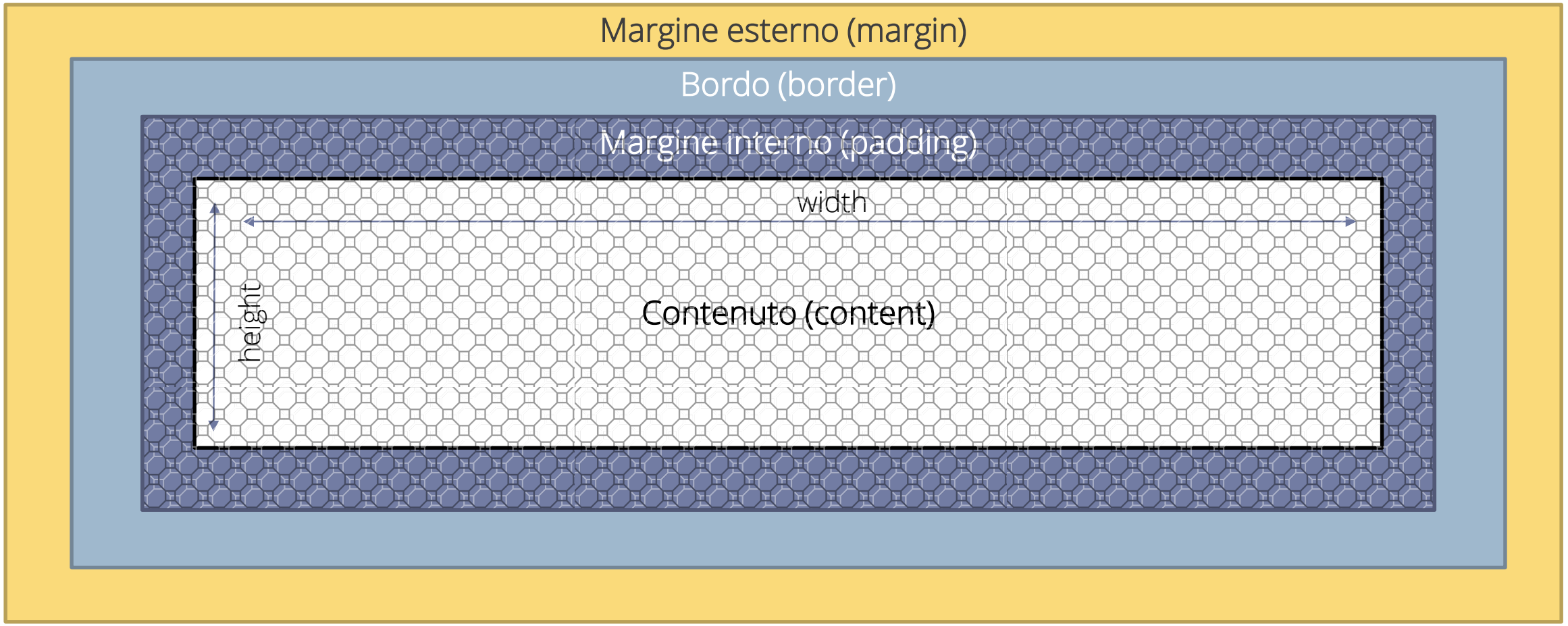
BOX MODEL (content-box) default



BOX MODEL (border-box)



BOX MODEL (background)



BOX-SIZING

- Le proprietà **box-sizing** viene utilizzato per indicare al browser quali componenti del box model devono essere inclusi nel calcolo di larghezza e altezza.
- Il valore di **box-sizing** deve essere:
  - **content-box** (default): il calcolo comprende solo il contenuto.
  - **border-box**: il calcolo comprende anche margine interno (padding) e border ma non margin .

```
* {
    box-sizing : border-box;
}
```

BACKGROUND

- background-color
  - Definisce il colore di sfondo di un elemento.

```
body {
    background-color: #FFFFFF;
}
div.main {
    background-color: transparent;
}
```

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

```
body {
    background-image: url(sfondo-scuro.jpg);
}
div.main {
    background-image: none;
}
```

BACKGROUND

- background-repeat
  - Consente di definire la direzione in cui l'immagine di sfondo viene ripetuta.
  - Valori: repeat, repeat-x, repeat-y, no-repeat

```
body {
    background-repeat: repeat-x;
}
```

- background-attachment
  - Valori: scroll, fixed.

```
div.main {
    background-attachment: fixed;
}
```

BACKGROUND

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

```
body {
    background-position: top center;
}
```

  - 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.

```
body {
    background: url(pattern.png) repeat-x scroll;
}
```

BACKGROUND

- background-clip
  - Definisce l'are dello sfondo.
  - Valori: border-box|padding-box|content-box|initial|inherit

```
div.main {
   background-clip: padding-box;
}
```

- background-origin
  - A cosa è relativa la posizione dell'immagine.
  - Valore: border-box|padding-box|content-box|initial|inherit

```
div.main {
   background-origin: border-box;
}
```

BACKGROUND

- background-size
  - Consente di definire come l'immagine di sfondo riempirà il contenitore.
  - auto | larghezza altezza | cover | contain | initial| inherit

```
div.main {
   background-size: cover;
}
```

VALORI

Valore	Esempio
parola riservata	display: none; margin-left: auto;
numero	line-height: 1.3; font-weight: 300;
misura	margin-top: 20px; line-height: 24px;
percentuale	width: 80%; margin-top: 2%;
url	background-image: url(img/sfondo.jpg);
colore	nome-colore, #RRGGBB, #RRGGBBAA, rgb(rrr,ggg,bbb), rgba(rrr,ggg,bbb,aaa), hsl(hhh,sss,lll), hsla(hhh,sss,lll,aaa)
più valori	background: url(img/sfondo.jpg) no-repeat center; margin: o auto;
attr()	Restituisce il valore dell'attributo inserita tra parentesi
calc()	Consente di inserire una misura come risultato di un calcolo

BORDER

- Stile del bordo
  - La proprietà **`border-style`** specifica quale tipo di bordo visualizzare.
  - Valori sono consentiti:
    - dotted - Definisce un bordo punteggiato
    - dashed - Definisce un bordo tratteggiato
    - solid - Definisce un bordo solido
    - double - Definisce un doppio bordo
    - groove- Definisce un bordo 3D scanalato. L'effetto dipende dal valore border-color
    - ridge- Definisce un bordo 3D increspato. L'effetto dipende dal valore border-color
    - inset- Definisce un bordo 3D inserto. L'effetto dipende dal valore border-color
    - outset- Definisce un bordo in 3D. L'effetto dipende dal valore border-color
    - none - Definisce nessun bordo
    - hidden - Definisce un bordo nascosto
  - La proprietà **`border-style`** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

BORDER-STYLE

```
p.dotted  { border-style:  dotted; }
p.dashed  { border-style:  dashed; }
p.solid  { border-style:  solid; }
p.double  { border-style:  double; }
p.groove  { border-style:  groove; }
p.ridge  { border-style:  ridge; }
p.inset  { border-style:  inset; }
p.outset  { border-style:  outset; }
p.none  { border-style:  none; }
p.hidden  { border-style:  hidden; }
p.mix  { border-style:  dotted dashed solid double; }
```

LARGHEZZA DEL BORDO

Larghezza del bordo

- La proprietà **border-width** specifica la larghezza dei quattro bordi.
- La larghezza può essere impostata come una dimensione specifica (in px, pt, cm, em, ecc.) o utilizzando uno dei tre valori predefiniti: **thin**, **medium**, o **thick**.
- La proprietà **border-width** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

```
p.two {
  border-style: solid;
  border-width: medium; }
p.three {
  border-style: solid;
  border-width: 2px 10px 4px 20px;
}
```

COLORE DEL BORDO

Larghezza del bordo

- La proprietà **border-color** specifica la larghezza dei quattro bordi.
- Il colore deve essere specificato con uno dei valori legali
- La proprietà **border-color** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

```
p.one {
  border-style: solid;
  border-color: red;
}
p.two {
  border-style: solid;
  border-color: red green #0000FF rgb(255,255,0);
}
```

SINGOLI LATI

- Per ognuna delle proprietà elencate esiste la versione che consente di stilizzare un singolo lato;

```
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

VERSIONE COMPATTA

- Con la proprietà **border** è possibile settare tutte le proprietà di tutti i quattro bordi con un'unica regola;

```
p {
  border: 5px solid blue;
}
```

- Con le proprietà **border-top**, **border-right**, **border-bottom**, **border-left** settare tutte le proprietà dei singoli bordi.

BORDER RADIUS

- Con la proprietà **border-radius** è possibile impostare dei bordi arrotondati;

```
p {
  border: 2px solid darkred;
  border-radius: 5px;
}
```

OUTLINE

- L'outline è un bordo esterno che serve ad evidenziare un elemento e non influisce in alcun modo sulle dimensioni dell'elemento. Per default un outline tratteggiato circonda gli elementi che hanno il focus. La proprietà è la stesse caratteristiche di border
  - La proprietà **`outline-style`** specifica lo stile.
  - Valori sono consentiti:
    - dotted - Definisce un bordo punteggiato
    - dashed - Definisce un bordo tratteggiato
    - solid - Definisce un bordo solido
    - double - Definisce un doppio bordo
    - groove- Definisce un bordo 3D scanalato. L'effetto dipende dal valore border-color
    - ridge- Definisce un bordo 3D increspato. L'effetto dipende dal valore border-color
    - inset- Definisce un bordo 3D inserto. L'effetto dipende dal valore border-color
    - outset- Definisce un bordo in 3D. L'effetto dipende dal valore border-color
    - none - Definisce nessun bordo
    - hidden - Definisce un bordo nascosto
  - La proprietà **`outline-style`** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

OUTLINE-STYLE

```
p.dotted  {outline-style :  dotted; }
p.dashed  {outline-style :  dashed; }
p.solid  {outline-style :  solid; }
p.double  {outline-style :  double; }
p.groove  {outline-style :  groove; }
p.ridge  {outline-style :  ridge; }
p.inset  {outline-style :  inset; }
p.outset  {outline-style :  outset; }
p.none  {outline-style :  none; }
p.hidden  {outline-style :  hidden; }
p.mix  {outline-style :  dotted dashed solid double; }
```

LARGHEZZA DEL OUTLINE

Larghezza del bordo

- La proprietà **outline-width** specifica la larghezza dei quattro bordi esterni.
- La larghezza può essere impostata come una dimensione specifica (in px, pt, cm, em, ecc.) o utilizzando uno dei tre valori predefiniti: **thin**, **medium**, o **thick**.
- La proprietà **outline-width** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

```
p.two {
    outline-width : solid;
    outline-width : medium; }
p.three {
    outline-width : solid;
    outline-width : 2px 10px 4px 20px;
}
```

COLORE DEL OUTLINE

Larghezza del bordo

- La proprietà **outline-color** specifica la larghezza dei quattro bordi esterni.
- Il colore deve essere specificato con uno dei valori legali
- La proprietà **outline-color** può avere da uno a quattro valori (superiore, destro, inferiore, sinistro).

```
p.one {
    outline-color : solid;
    outline-color : red;
}
p.two {
    outline-color : solid;
    outline-color : red green #0000FF rgb(255,255,0);
}
```

VERSIONE COMPATTA

- Con la proprietà **outline** è possibile settare tutte le proprietà di tutti i quattro bordi con un'unica regola;

```
p {
   outline : 5px solid blue;
}
```

- Con le proprietà **outline-top**, **outline-right**, **outline-bottom**, **outline-left** settare tutte le proprietà dei singoli bordi.

MARGINI

CSS ha proprietà per specificare il margine per ogni lato di un elemento:

- margin-top
- margin-right
- margin-bottom
- margin-left

Tutte le proprietà dei margini possono avere i seguenti valori: auto, misura (px, em, ecc.), percentuale Sono consentiti valori negativi.

Esempi

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
p {
    margin: 100px 150px 100px 80px;
}
```

MARGINE INTERNO

CSS ha proprietà per specificare il margine interno (padding) per ogni lato di un elemento:

- padding-top
- padding-right
- padding-bottom
- padding-left

Tutte le proprietà possono avere i seguenti valori: auto, misura (px, em, ecc.), percentuale

Esempi

```
p {
    padding-top: 100px;
    padding-bottom: 100px;
    padding-right: 150px;
    padding-left: 80px;
}
p {
     padding: 100px 150px 100px 80px;
}
```

ALTEZZA E LARGHEZZA

- Le proprietà **height** e **width** sono utilizzati per impostare l'altezza e la larghezza di un elemento.
- **height** e **width** possono essere impostate su auto (impostazione predefinita), e sarà il browser a calcolare l'altezza e la larghezza), o essere specificato con una misura, come px, cm, em, ecc., o in percentuale (%) del blocco che contiene l'elemento.

```
div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
}
```

MAX-HEIGHT E MAX-WIDTH

- Le proprietà **max-height** e **max-width** sono utilizzati per impostare l'altezza e la larghezza massima di un elemento.
- **max-height** e **max-width** possono essere una misura, come px, cm, em, ecc., una percentuale (%) del blocco che contiene l'elemento o **none**.

```
div {
    max-height: 200px;
    max-width: 50%;
    background-color: powderblue;
}
```

MIN-HEIGHT E MAX-WIDTH

- Le proprietà **min-height** e **min-width** sono utilizzati per impostare l'altezza e la larghezza minima di un elemento.
- **min-height** e **min-width** possono essere una misura, come px, cm, em, ecc., una percentuale (%) del blocco che contiene l'elemento. Per eliminare il limite vanno impostate su 0 .

```
div {
    min-height : 200px;
    min-width : 50%;
    background-color: powderblue;
}
```

FLOAT

- Un elemento di tipo blocco (**display:block**) interrompe il flusso naturale della pagina anche se non ne occupa tutta la larghezza. Le proprietà **float** consente agli elementi che seguono l'elemento blocco di affiancarsi all'elemento stesso, a destra o a sinistra se lo spazio rimanente nella pagina lo consente.
- Il valore di **float** può essere:
  - **left**: L'elemento si colloca a sinistra nella pagina e gli elementi che seguono lo affiancano a destra.
  - **right**: L'elemento si colloca a destra nella pagina e gli elementi che seguono lo affiancano a sinistra.
  - **none**: (valore di default) l'elemento non consente il float.

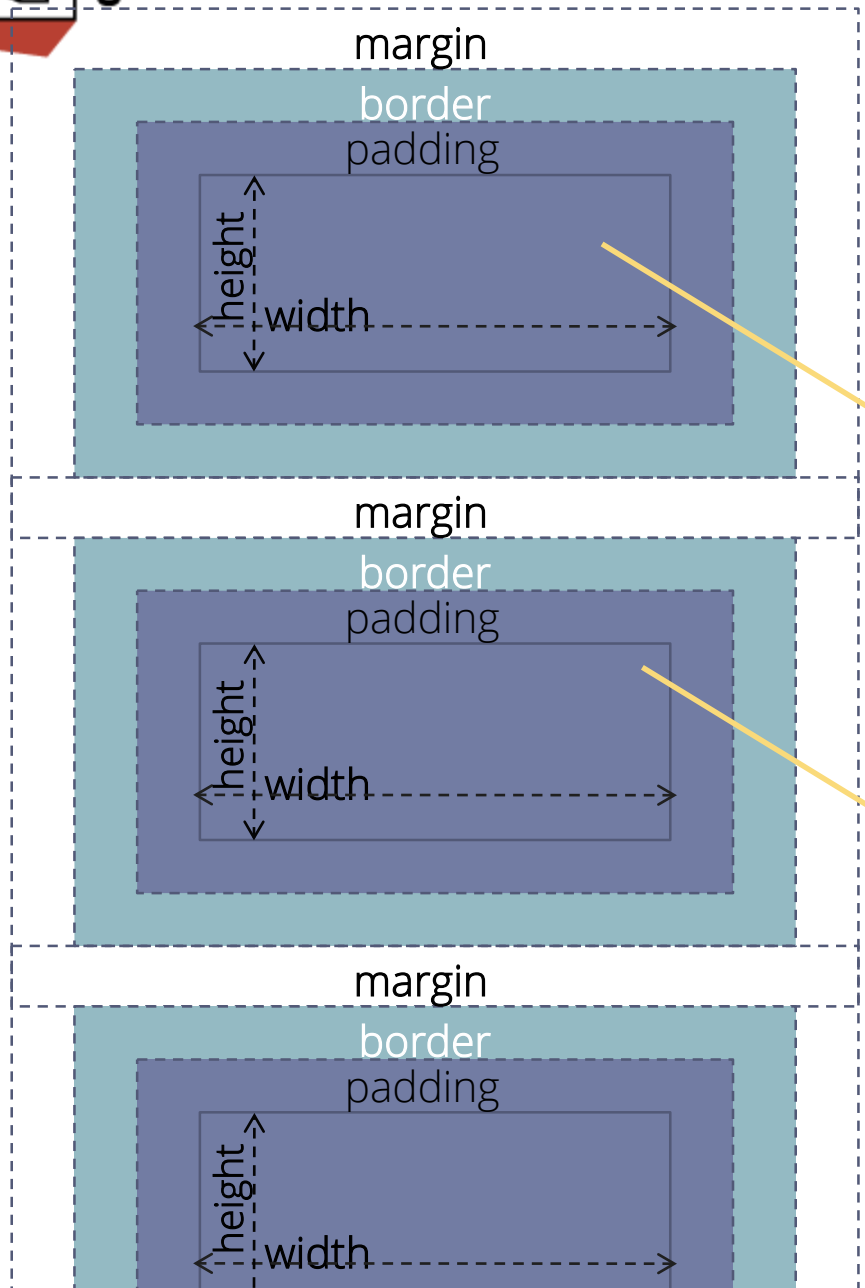
```
img {
    float : right;
     margin: 0 0 15px 15px;
}
```

CLEAR

- Le proprietà **clear** consente di interrompere il flusso float
- Il valore di **clear** può essere:
  - **left**: Interrompe il float a sinistra.
  - **right**: Interrompe il float a destra.
  - **both**: Interrompe il float sia sinistra che a destra .

```
div.clear {
    float : none;
    clear: both;
}
```

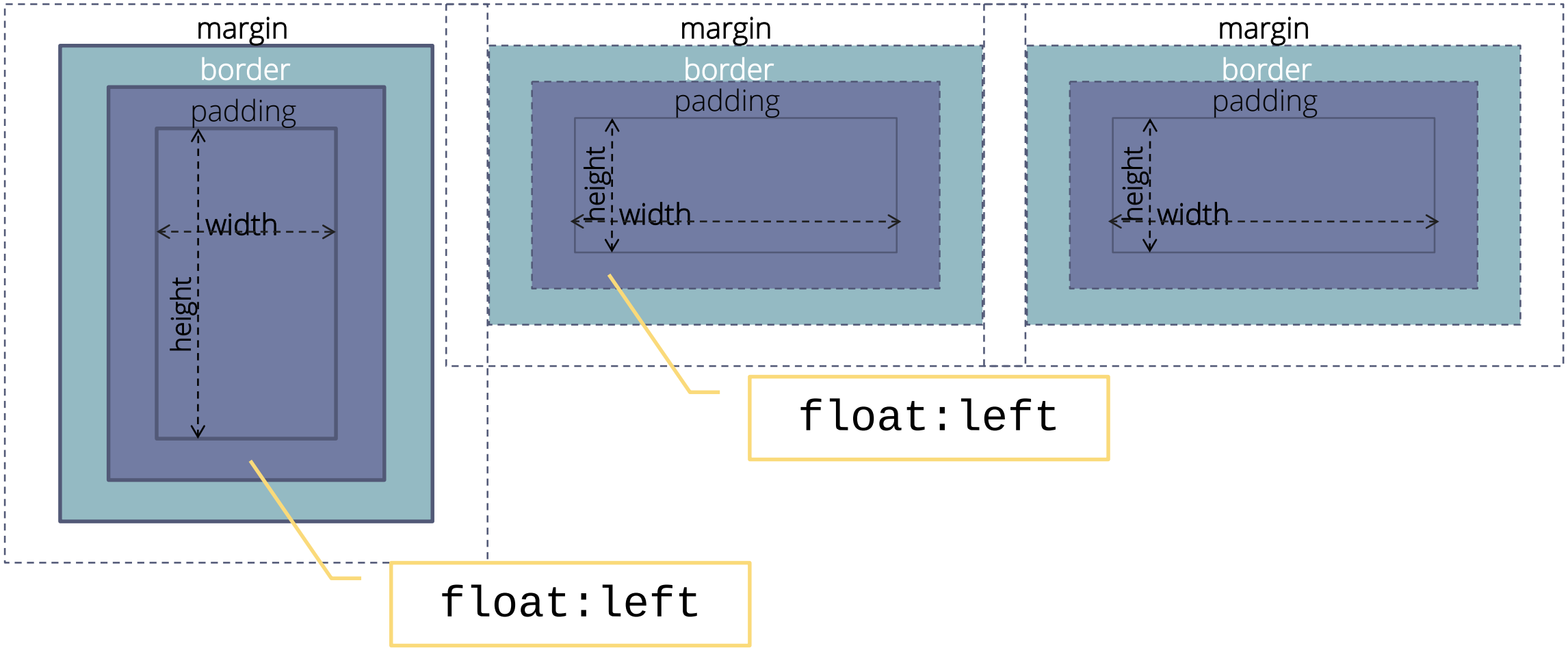
FLOAT



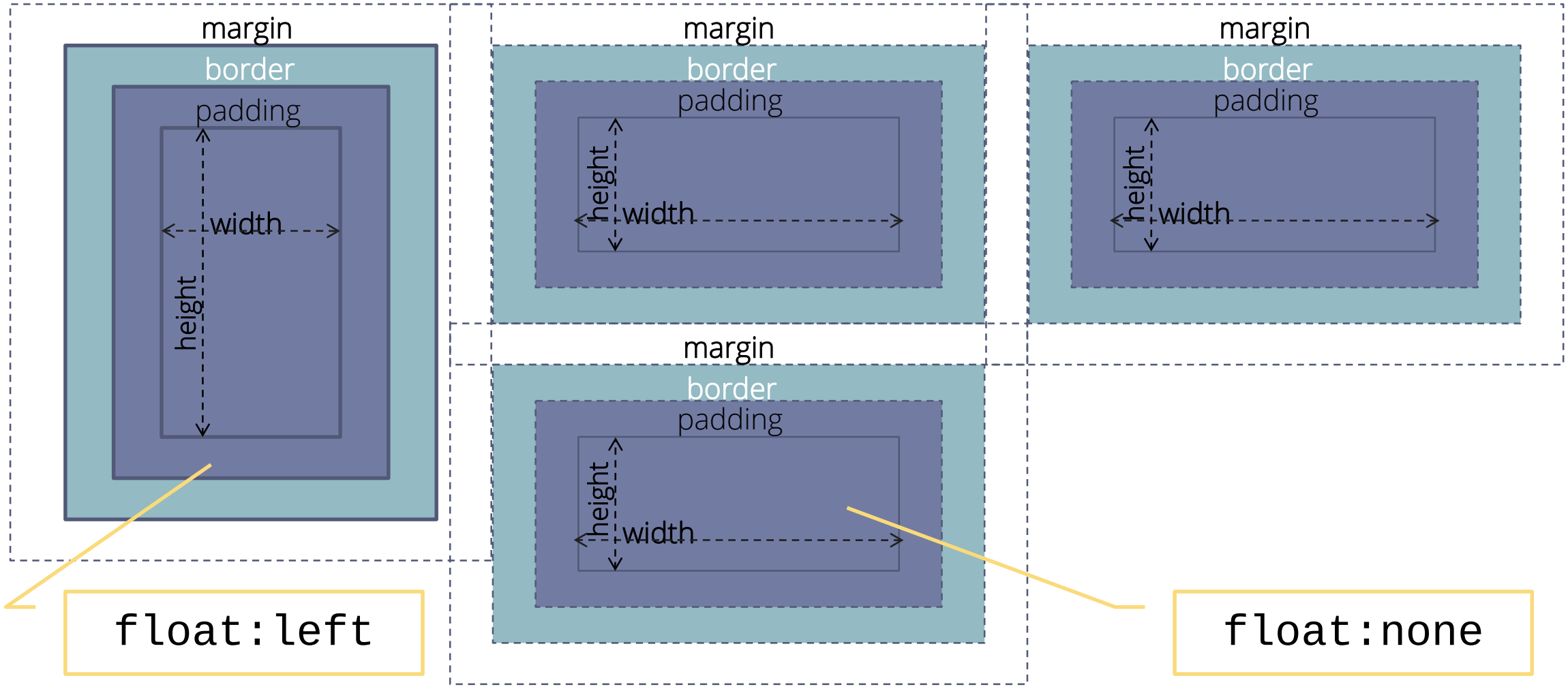
`float:none`

`float:none`

FLOAT


margin
border
padding
width
height
margin
border
padding
height
width
margin
border
padding
height
width
float:left
float:left

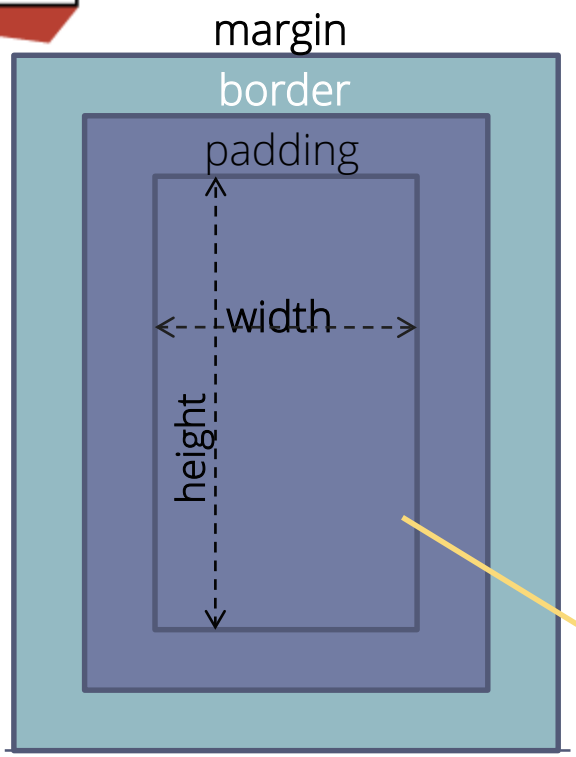
FLOAT

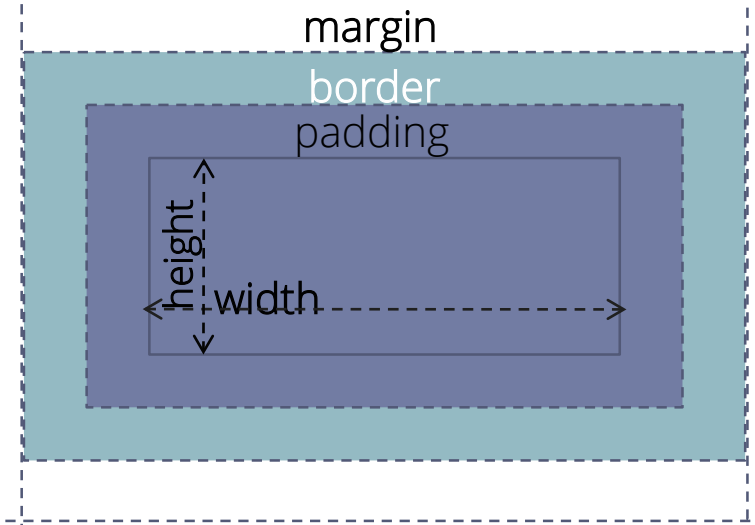

margin
border
padding
width
height
margin
border
padding
height
width
margin
border
padding
height
width
margin
border
padding
height
width
float:left
float:none

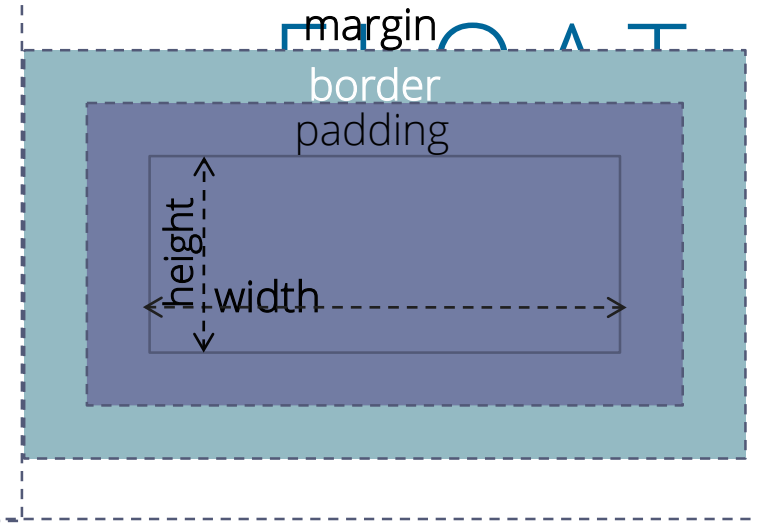
CLEARFIX

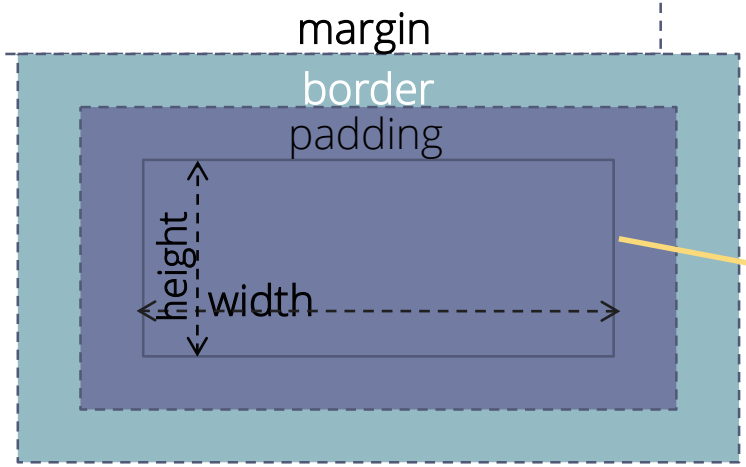
- Spesso è un problema interrompere correttamente il flusso float se gli elementi non sono della stessa altezza. In rete si trova questo trucco. Assegnando la classe **clearfix** ad un elemento che contiene elementi float il flusso si chiude correttamente.

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```










```
<div class="clearfix">
```

```
float:left
```

```
float:none;
```

POSITION

POSITION

- La proprietà **position** specifica il metodo di posizionamento utilizzato per un elemento.
- **position** può assumere quattro diversi valori:
  - **static**
  - **relative**
  - **fixed**
  - **absolute**
- La posizione degli elementi è determinata dalle proprietà **top**, **right**, **bottom** e **left**.
- Modificare i valori di queste proprietà ha effetto solo se **position** è diversa da **static**.
- L'effetto sulla posizione cambia a secondo dell'impostazione di **position**.

STATIC

- Gli elementi HTML sono posizionati così per impostazione predefinita.
- Un elemento con **position:static** non è posizionato in modo speciale, ma secondo il normale flusso della pagina
- La posizione non è influenzata dalle proprietà **top**, **right**, **bottom** e **left**.

```
div.normale  {
    position: static;
    border: 3px solid #73AD21;
}
```

RELATIVE

- La posizione di un elemento con **position:realtive** è calcolata in maniera relativa rispetto alla sua posizione naturale secondo il normale flusso della pagina
- Le proprietà **top**, **right**, **bottom** e **left** causeranno uno scostamento dell'elemento rispetto alla sua posizione naturale.

```
div.relativa  {
     position: relative;
     left: 50px;
}
```

FIXED

- La posizione di un elemento con **position:fixed** è calcolata in relazione alla finestra del browser il che significa che rimane sempre nello stesso posto, anche se la pagina viene fatto scorrere.
- Le proprietà **top**, **right**, **bottom** e **left** determinano la posizione dell'elemento.

```
div.fissa  {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
}
```

ABSOLUTE

- La posizione di un elemento con **`position:absolute`** è calcolata in relazione all'elemento che lo contiene se questo non è **`position:static`**, altrimenti in relazione al documento.
- Le proprietà **`top`**, **`right`**, **`bottom`** e **`left`** determinano la posizione dell'elemento.
- L'elemento **`position:absolute`** quando la pagina scrolla, scrolla insieme all'elemento relativamente al quale è posizionato.

ABSOLUTE

```
div.relativa {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.assoluta {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
```

Z-INDEX

- Quando un elemento è posizionato (cioè non è **position:static**) può sovrapporsi ad altri elementi-
- La proprietà **z-index** determina l'ordine di questa sovrapposizione è può avere sia un valore positivo che uno negativo.

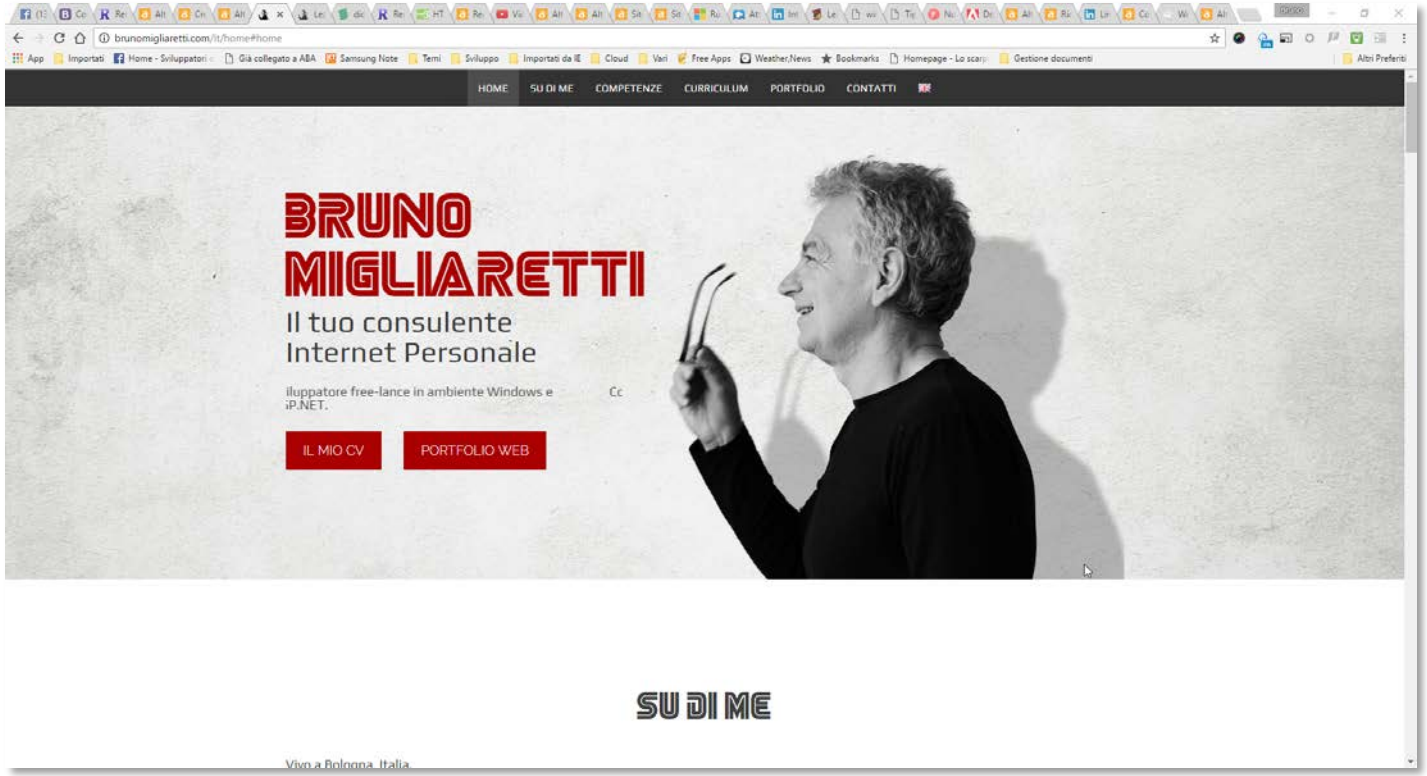
```
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
```

CSS RESPOSIVE

≥1200px

1920 PX

1080 PX



≥992px < 1200px

1024 PX

768 PX

≥768px < 992px


768 PX
1024 PX
HOME
SU DI ME
COMPETENZE
CURRICULUM
PORTFOLIO
CONTATTI
BRUNO MIGLIARETTI
Il tuo consulente Internet Personale
Consulente SEO e Social Media.
IL MIO CV
PORTFOLIO WEB

<768px

320 PX

568 PX

VIEWPORT

- La **`viewport`** è l'area visibile da parte dell'utente di una pagina web.
- La **`viewport`** varia con il dispositivo.
- Prima della diffusione di tablet e telefoni cellulari, le pagine web erano progettate solo per gli schermi di computer:
  - progettazione statica
  - dimensione fissa.
- Con la diffusione di tablet e telefoni cellulari, le pagine web erano troppo grandi per adattarsi alla finestra. Per risolvere questo problema, i browser su tali dispositivi scala verso il basso l'intera pagina web per adattarsi allo schermo.

IMPOSTAZIONE DELLA VIEWPORT

HTML5 ha introdotto un metodo per consentire web designer di controllare la viewport, attraverso il tag **<meta>.**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- L'elemento **<meta>** fornisce le istruzioni al browser su come controllare il ridimensionamento della pagina.
- **width=device-width** impone che la larghezza della pagina coincida con quella del dispositivo.
- **initial-scale=1.0** imposta il livello di zoom iniziale

IMPOSTAZIONE VEWPORT

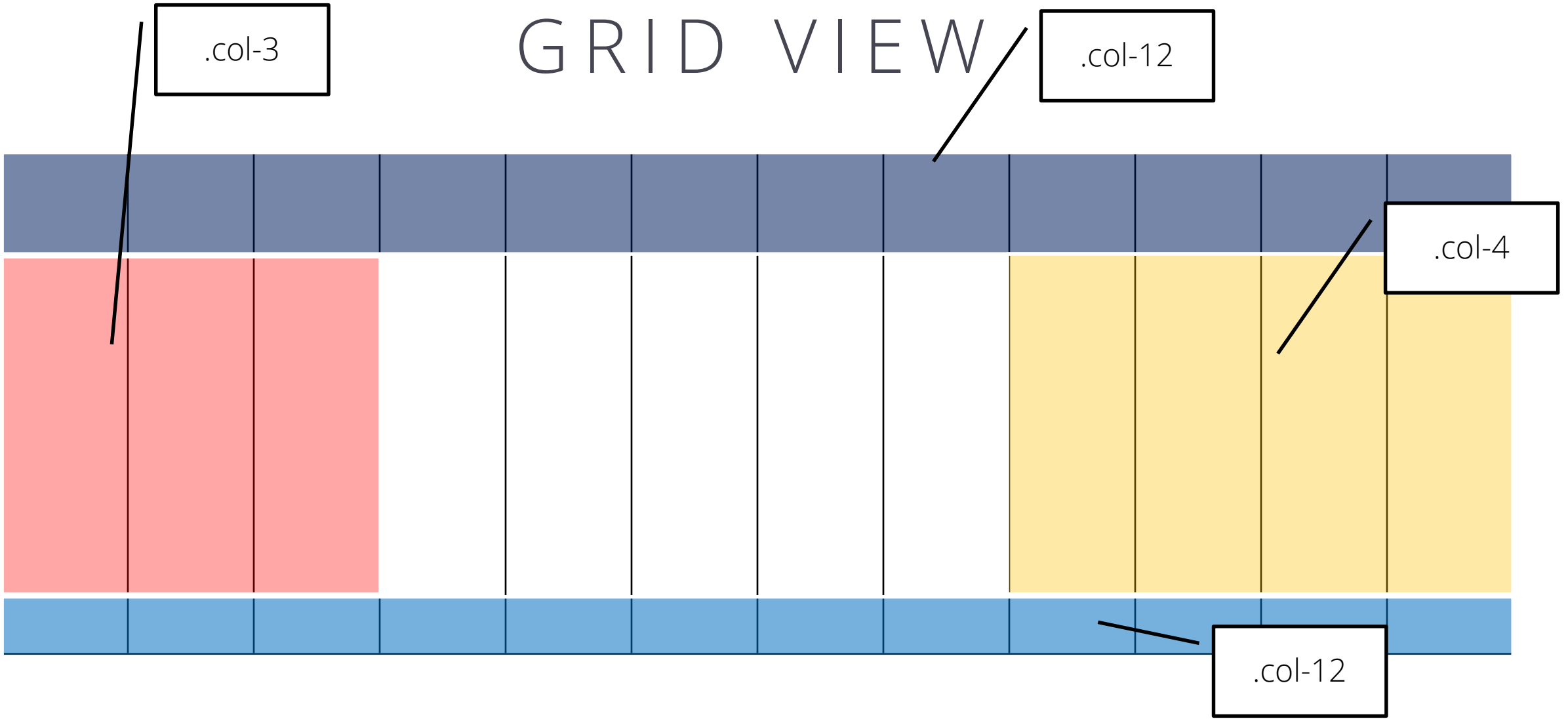
Viewport di default

Viewport correttamente impostata

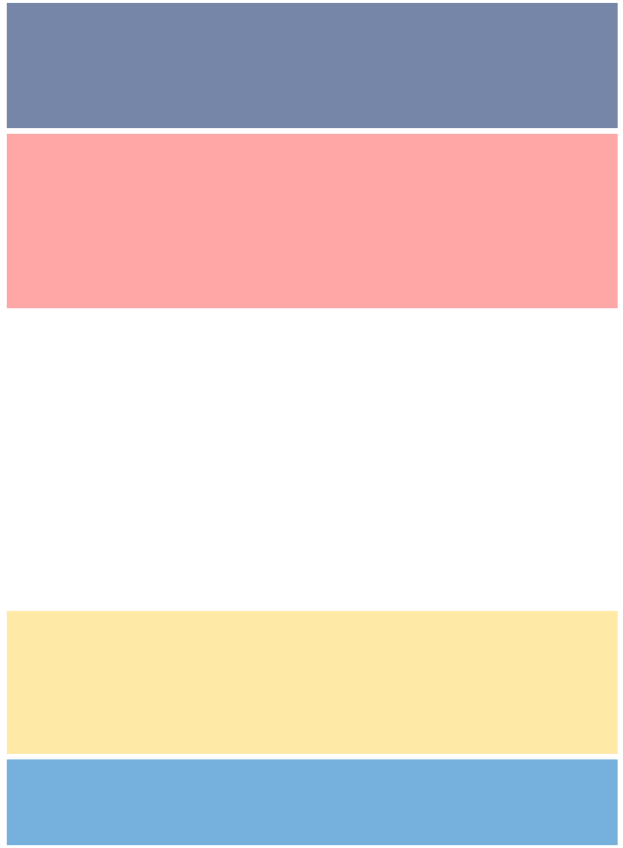
PRECAUZIONI

- Non bisogna costringere l'utente a scorrere in orizzontale, o diminuire lo zoom, per vedere l'intera pagina: **Gli utenti sono abituati a scorrere i siti web in verticale**.
- **Non utilizzare grandi elementi larghezza fissa.** Le immagini, ad esempio, devono sempre adattarsi al contenitore in modo da non costringere ma l'utente a scrollare orizzontalmente o a diminuire lo zoom.
- **Creare contenuti che possano essere resi correttamente indipendentemente dalla larghezza.**
- **Utilizzare le regole media query per applicare stili diversi per schermi diversi**.
- **Se la larghezza di un elemento non è controllata dalle media query usare dimensioni percentuali.**
- **Fare attenzione nell'utilizzo del posizionamento assoluto.**

GRID VIEW



GRID VIEW



CSS

```
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
@media only screen and (max-width: 768px) {
    /* For mobile phones: */
    [class*="col-"] {
        width: 100%;
    }
}
```

MEDIA QUERY

```
@media only screen and (max-width:1024px){
    .container {
        width: 960px;
    }
}
```

ANIMAZIONI

TRANSITION

- **`trasition`** consente di ottenere un animazione di una certa durata da un cambio di valori delle proprietà di un elemento.
- Per creare un effetto di transizione, è necessario specificare due cose:
  - la proprietà CSS su cui si desidera impostare l'effetto
  - la durata dell'effetto

TRANSITION

- L'esempio seguente mostra un elemento ** ** di 100px * 100px rosso L'effetto di transizione opera sulla proprietà **width**, e ha una durata di 2 secondi:

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s; /* Safari */
    transition: width 2s;
}
```

- L'effetto di transizione inizierà quando la proprietà CSS specificata (**width**) cambia il valore.
- Ora definiamo un nuovo valore per la proprietà **width** quando il mouse passa sopra l'elemento:

```
div:hover {
    width: 300px;
}
```

TRANSFORM

- transform permette di spostare, ruotare, scalare e inclinare gli elementi HTML.
- Una trasformazione è un effetto che permette un cambiamento di forma, di dimensione e/o di posizione dell'elemento
- CSS3 supporta le trasformazioni 2D e 3D.
- Le trasformazioni si realizzano applicando le seguenti funzioni:
  - translate()
  - rotate()
  - scale()
  - skewX()
  - skewY()

translate(x,y)

Il metodo translate() sposta un elemento dalla posizione correte secondo i parametri indicati per l'asse X e l'asse Y.
L'esempio seguente sposta l'elemento 50 pixel a destra, e 100 pixel verso il basso dalla sua posizione attuale:

```
div {
    -ms-transform: translate(50px, 100px); /* IE 9 */
    -webkit-transform: translate(50px, 100px); /*
Safari */
    transform: translate(50px, 100px);
}
```

rotate(angolo)

Il metodo rotate() ruota un elemento in senso orario o antiorario secondo un determinato angolo.
L'esempio seguente ruota l'elemento in senso orario di 20 gradi:

```
div {
    -ms-transform: rotate(20deg); /* IE 9 */
    -webkit-transform: rotate(20deg); /* Safari */
    transform: rotate(20deg);
}
```

scale(x,y)

Il metodo scale() aumenta o diminuisce la dimensione di un elemento.
L'esempio seguente ingrandisce l'elemento di due volte del rispetto alla larghezza originale, e tre volte rispetto alla sua altezza originale:

```
div {
    -ms-transform: scale(2, 3); /* IE 9 */
    -webkit-transform: scale(2, 3); /* Safari */
    transform: scale(2, 3);
}
```

skewX(angolo)

Il metodo skewX() inclina un elemento lungo l'asse X per l'angolo determinato. L'esempio seguente distorce l'elemento 20 gradi lungo l'asse X:

```
div {
    -ms-transform: skewX(20deg); /* IE 9 */
    -webkit-transform: skewX(20deg); /* Safari */
    transform: skewX(20deg);
}
```

skewY(angolo)

Il metodo skewY() inclina un elemento lungo l'asse Y per l'angolo determinato.
L'esempio seguente distorce l'elemento 20 gradi lungo l'asse Y:

skewX

```
div {
    -ms-transform: skewY(20deg); /* IE 9 */
    -webkit-transform: skewY(20deg); /* Safari */
    transform: skewY(20deg);
}
```