

Umberto Emanuele

# HTML & CSS

## 2 Giorno

---

Novembre 2023



---

# CSS Positioning: floating e clear

Float:none

Testo testo testo testo  
testo testo  
testo testo testo

Altro testo testo testo  
testo testo testo  
testo testo testo

Float:  
left

Testo testo  
testo testo

testo testo  
testo testo testo

Altro testo testo testo  
testo testo testo  
testo testo testo

Float:  
right

Testo testo  
testo testo  
testo testo  
testo testo testo

Altro testo testo testo  
testo testo testo  
testo testo testo

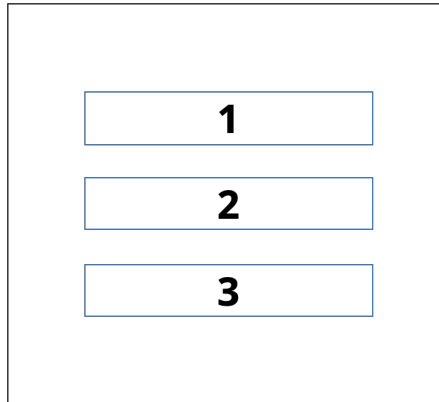
```
div p:last-child {  
    clear: left; /* left, right o both */  
}  
  
article#principale::after { /* clearfix */  
    content: "";  
    display: block;  
    clear: both;  
}
```

---

# CSS Positioning: absolute e relative

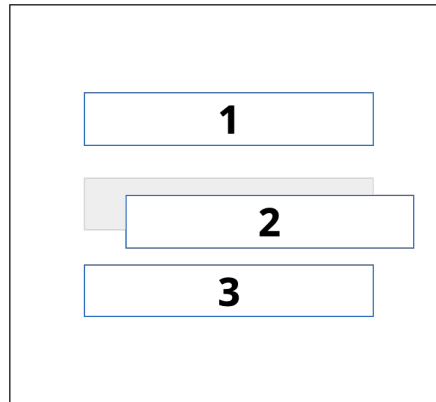
## Position: static

Posizionamento di default



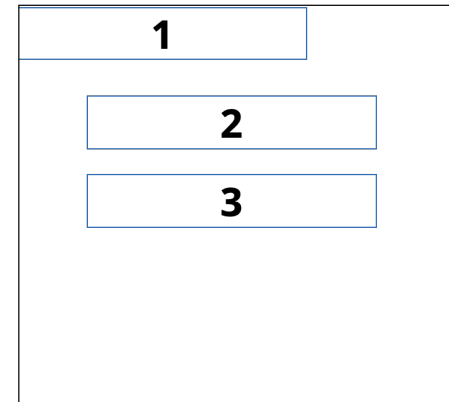
## Position: relative

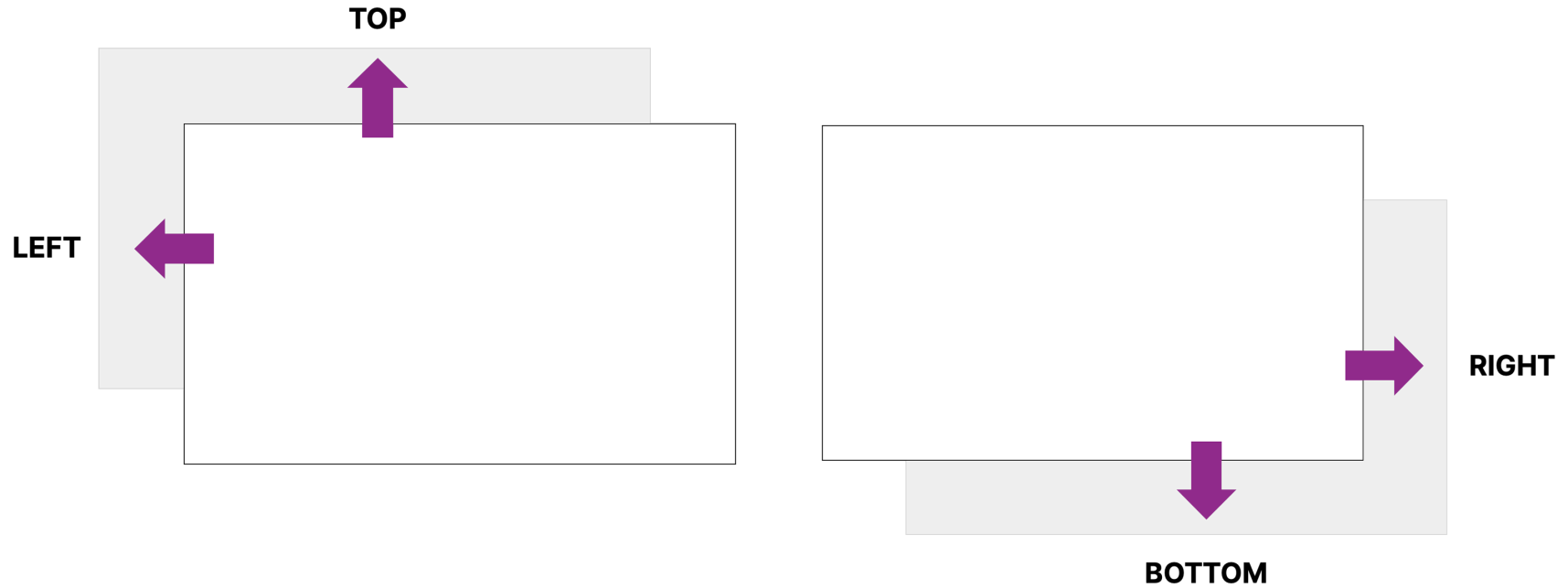
Relativo agli oggetti  
circostanti



## Position: absolute

Relativo alla finestra del  
browser

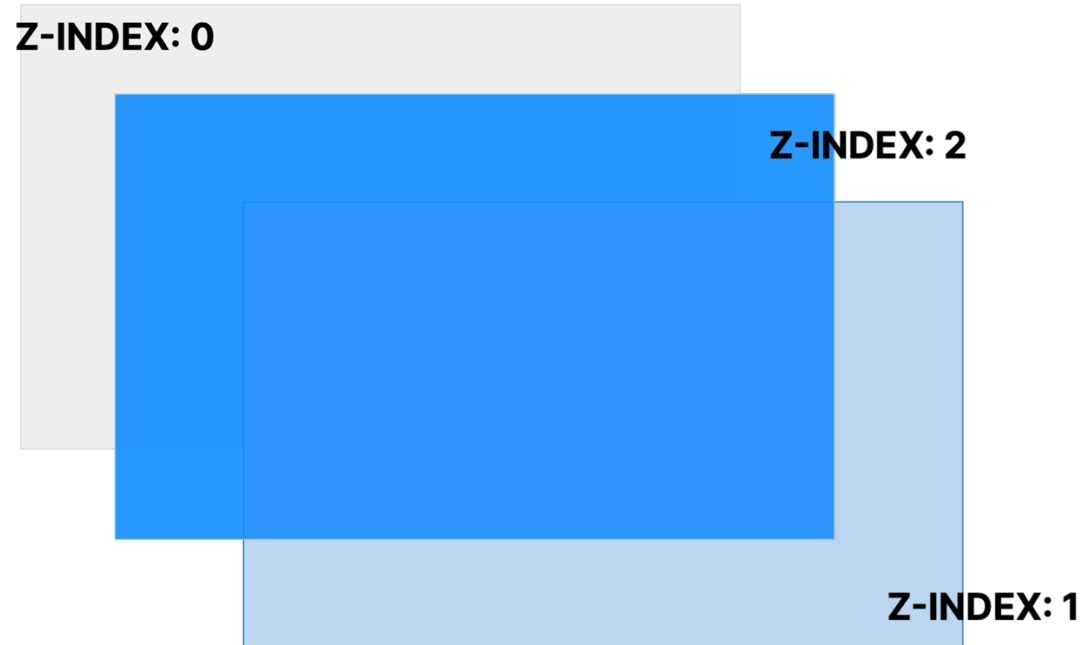


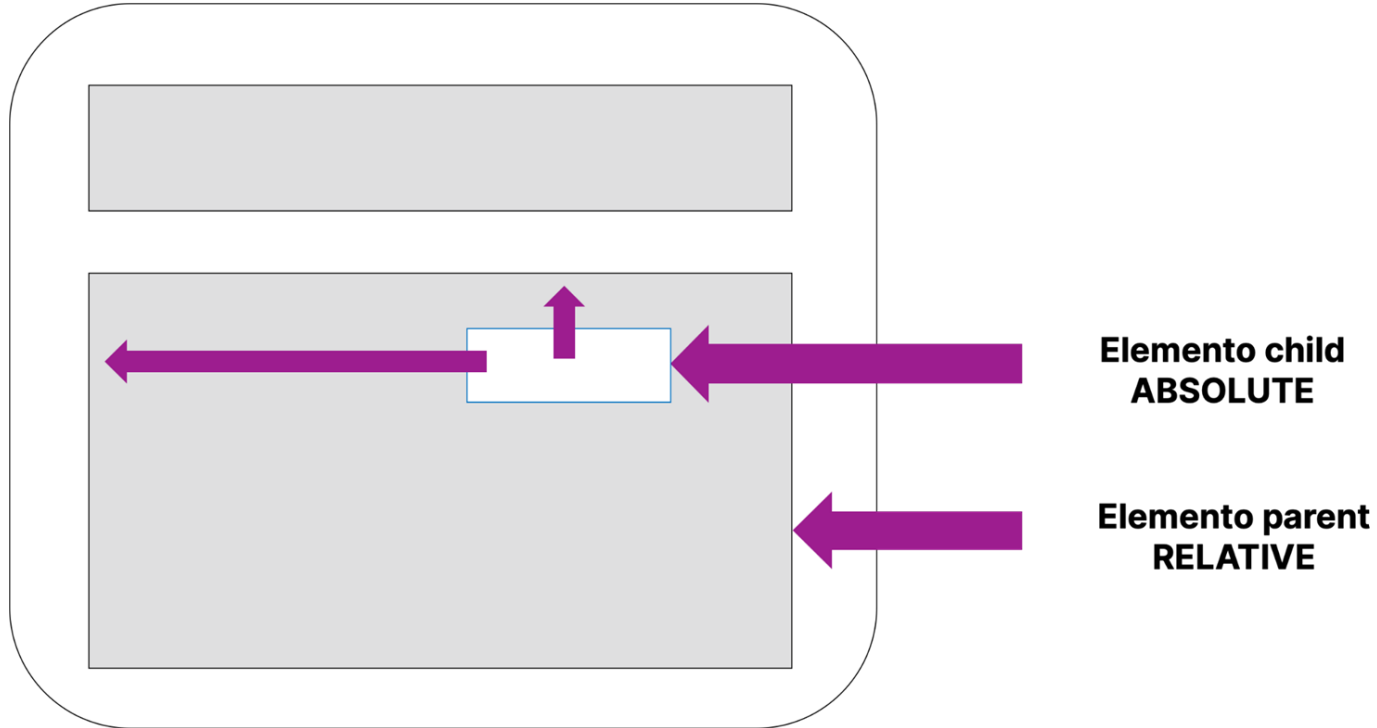


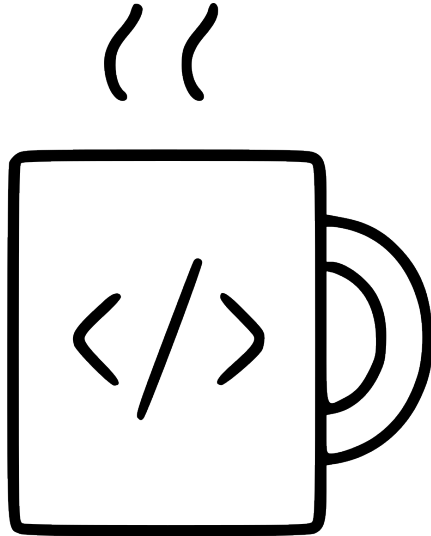
```
img#logo {  
    position: relative;  
    top: 10px;  
    right: 30px;  
}
```

```
.iconTop {  
    position: absolute;  
    top: -10px;  
    left: 3em;  
}
```









**PAUSA**

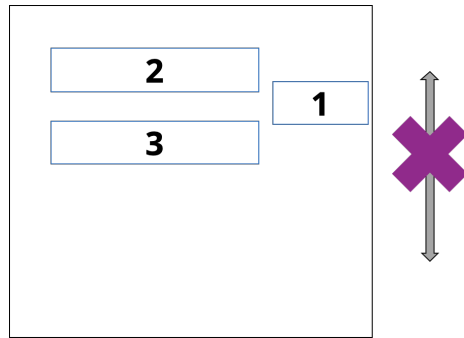
Ci vediamo alle ore 14.00

---

# CSS Positioning: fixed e sticky

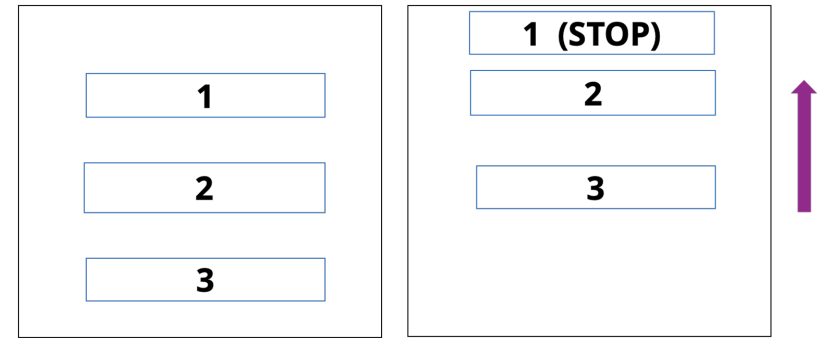
## Position: fixed

Relativo alla finestra del  
browser



## Position: sticky

Relativo alla finestra del browser e vincolato  
all'elemento parent



```
.iconToTop {  
  position: fixed;  
  bottom: 10vh;  
  right: 10px;  
}
```

```
nav {  
  position: sticky;  
  top: 0;  
}
```

---

# CSS Flexible Module: architettura ed elementi parent (containers)

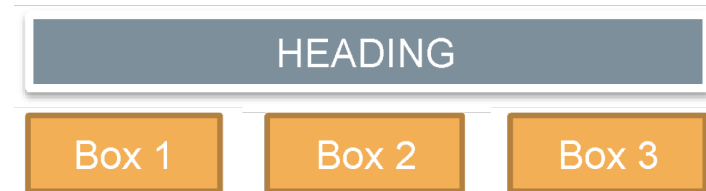
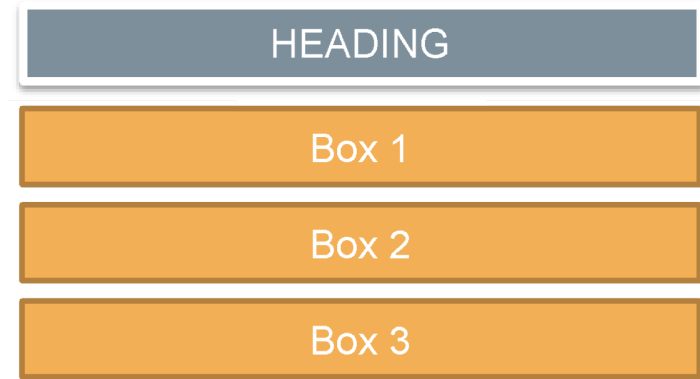
Dopo anni di trucchi, scorciatoie, usi impropri di tabelle o di proprietà di allineamento, i developers hanno a disposizione uno strumento per i layout monodimensionali.

## Prima

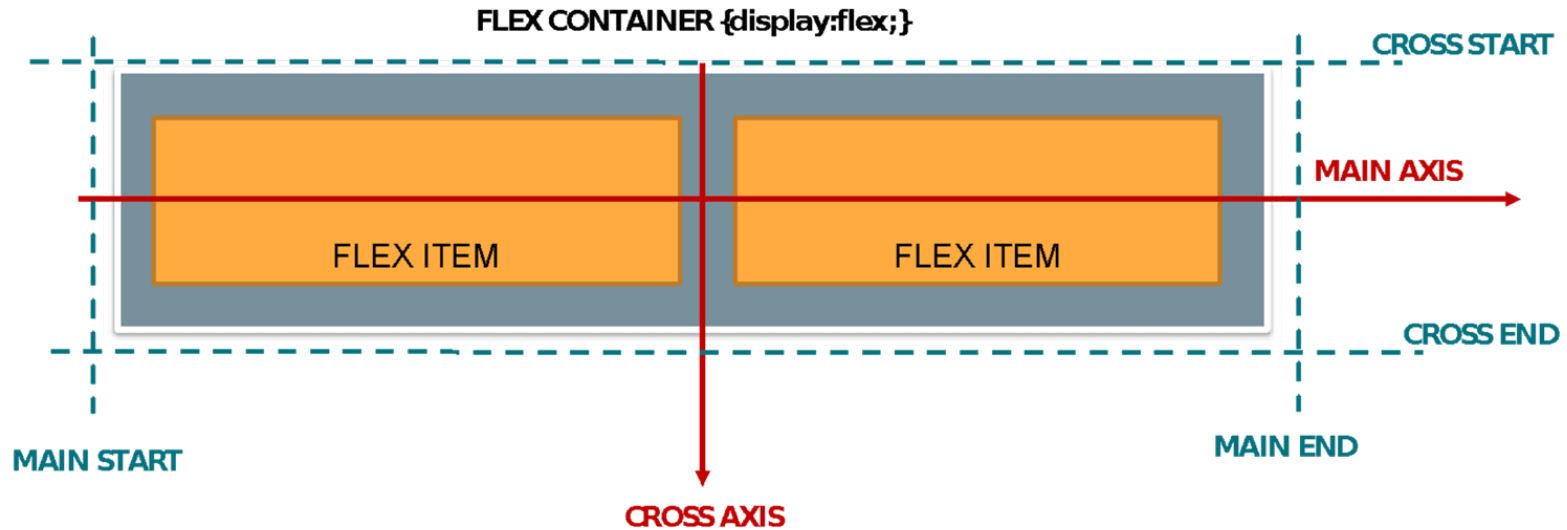
- Tabelle
- Float opposti
- Position: absolute

## Dopo

- Flexbox



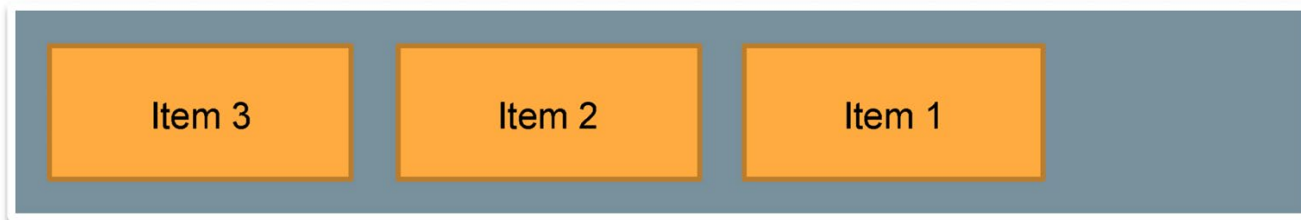




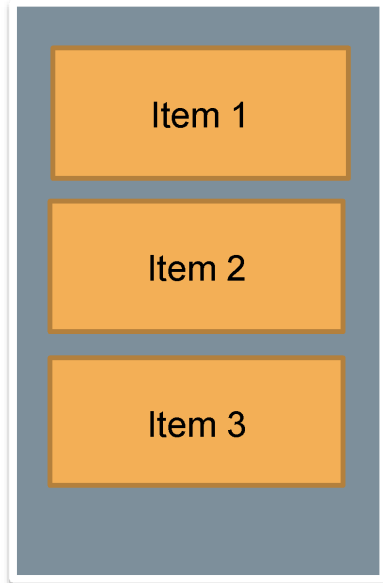
## FLEX-DIRECTION: ROW



## FLEX-DIRECTION: ROW-REVERSE



## FLEX-DIRECTION: COLUMN



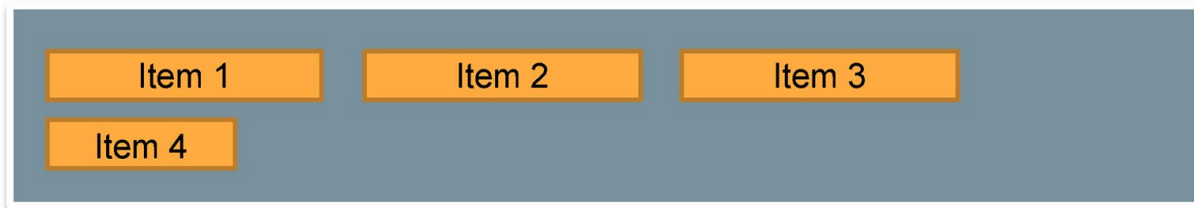
## FLEX-DIRECTION: COLUMN-REVERSE



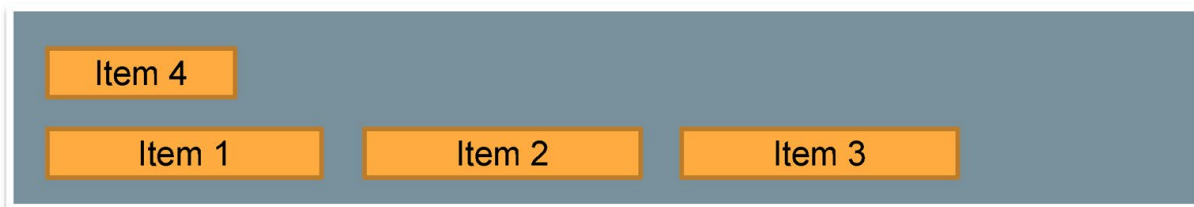
## FLEX-WRAP: NO-WRAP



## FLEX-WRAP: WRAP



## FLEX-WRAP: WRAP-REVERSE



## JUSTIFY-CONTENT: FLEX-START



## JUSTIFY-CONTENT: SPACE-BETWEEN



## JUSTIFY-CONTENT: FLEX-END



## JUSTIFY-CONTENT: SPACE-AROUND



## JUSTIFY-CONTENT: CENTER



## JUSTIFY-CONTENT: SPACE-EVENLY



ALIGN-ITEMS: FLEX-START



ALIGN-ITEMS: FLEX-STRETCH



ALIGN-ITEMS : FLEX-END



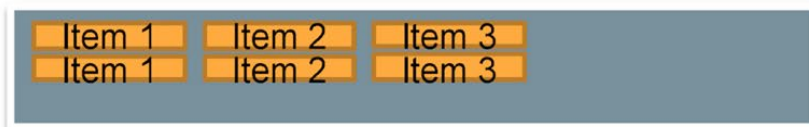
ALIGN-ITEMS: BASELINE



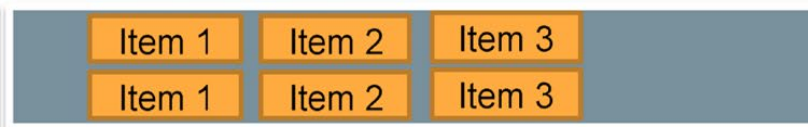
ALIGN-ITEMS : CENTER



## ALIGN-CONTENT: FLEX-START



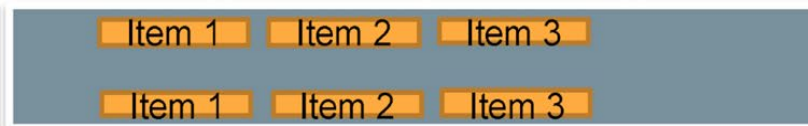
## ALIGN-CONTENT: STRETCH



## ALIGN-CONTENT: FLEX-END



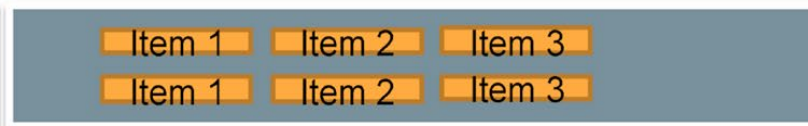
## ALIGN-CONTENT: SPACE-BETWEEN



## ALIGN-CONTENT: CENTER



## ALIGN-CONTENT: SPACE-AROUND

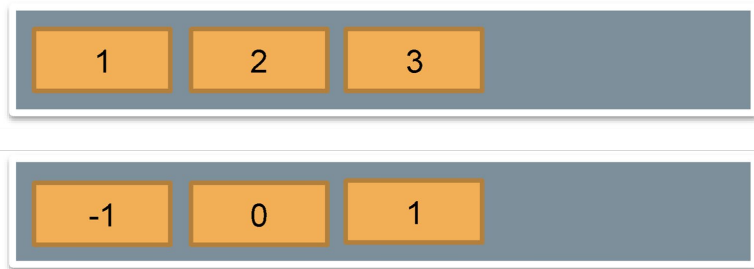


---

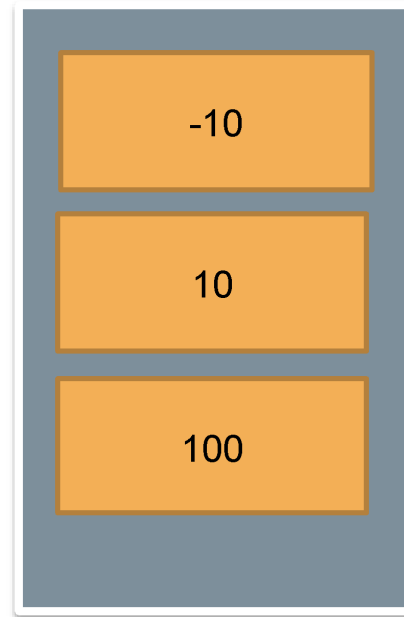
# CSS Flexible Module: elementi child (items)



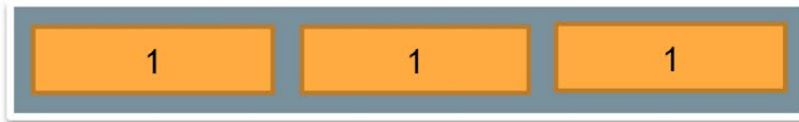
ORDER



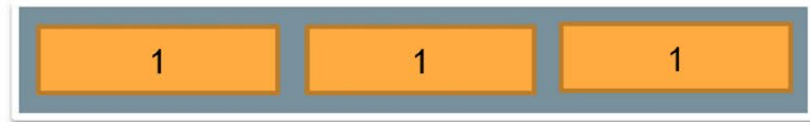
ORDER



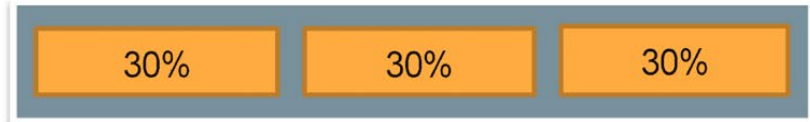
## FLEX-GROW



## FLEX-SHRINK

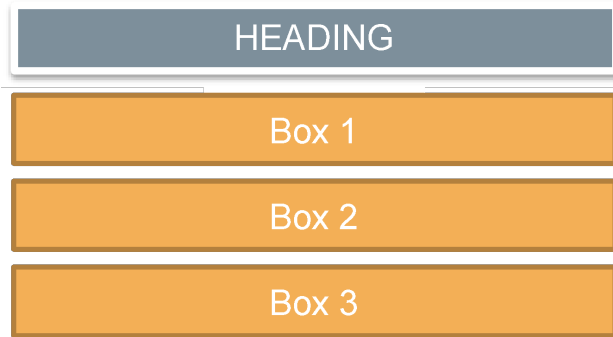


## FLEX-BASIS

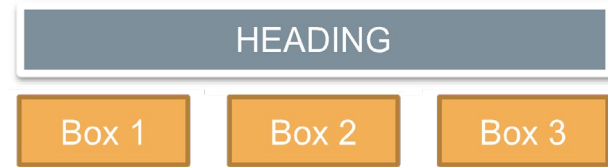


```
section div {  
  flex-grow: 0;  
  flex-shrink: 1;  
  flex-basis: auto;  
  flex: 0 1 auto; /* sintassi contratta */  
}
```

## SU MOBILE



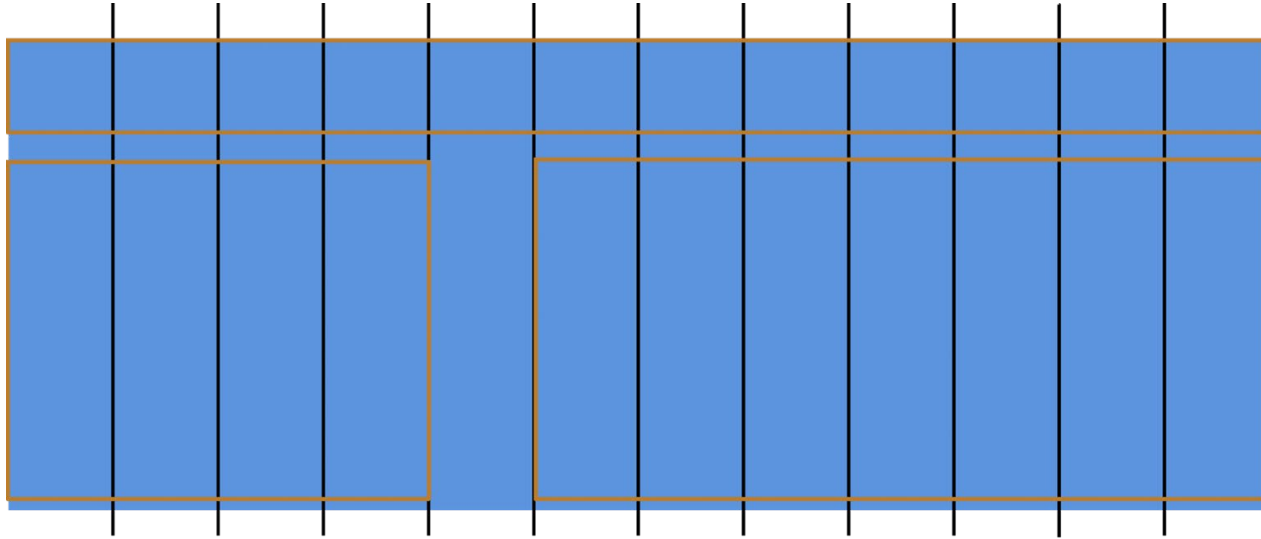
## MEDIA QUERY SU TABLET/DESKTOP



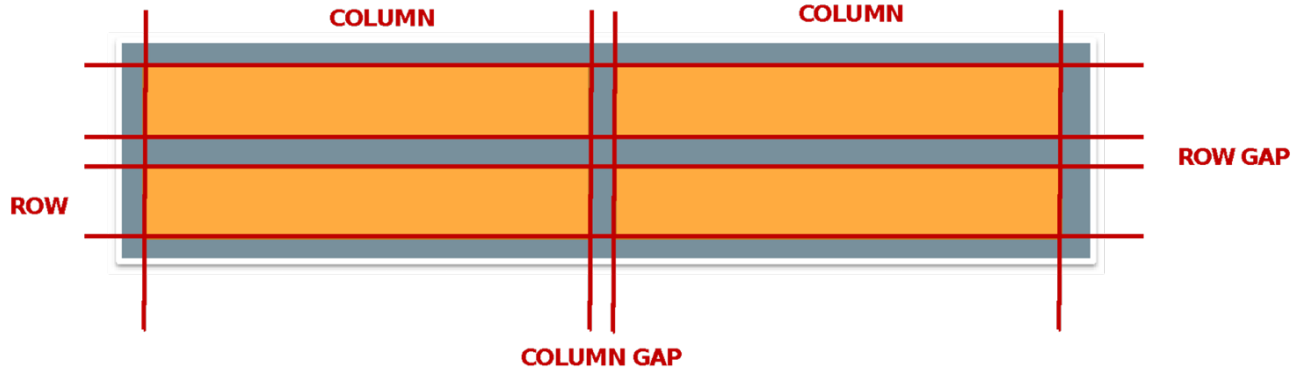
---

# CSS Grid: architettura ed elementi parent (containers)

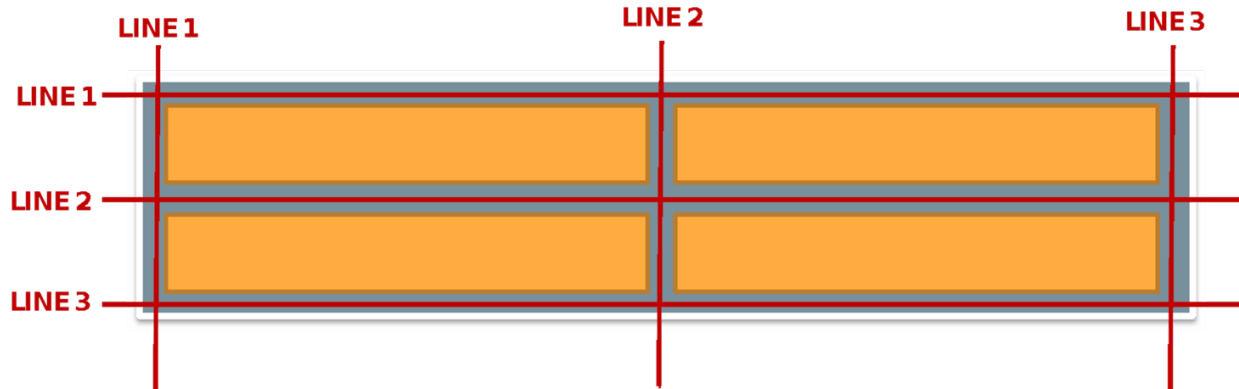
Divisione del viewport in colonne virtuali (solitamente 12)



GRID CONTAINER  
{display:grid;}

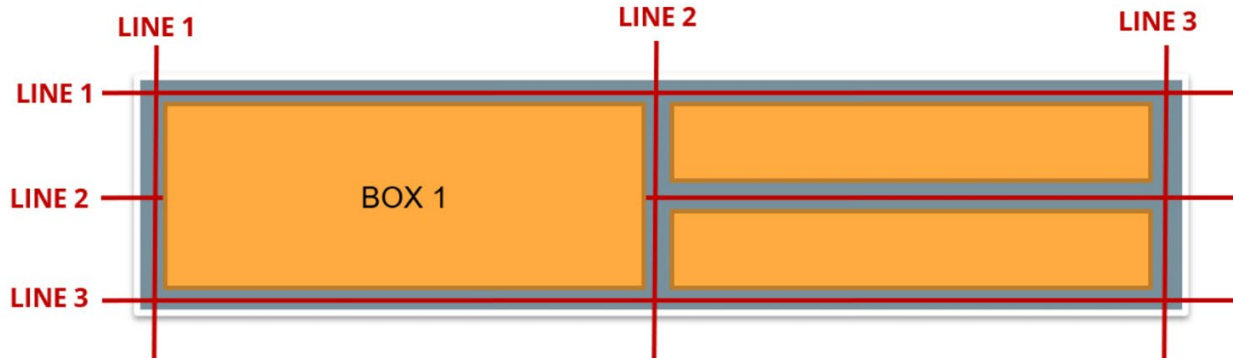


## GRID LINES



```
div.box1 {  
  grid-column-start: 1;  
  grid-column-end: 2;  
}
```

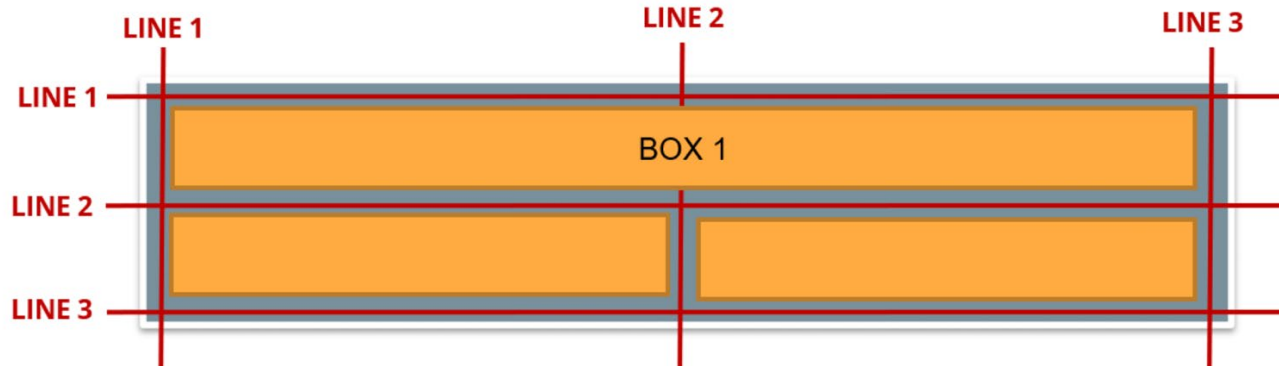
## GRID LINES



```
div.box1 {  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```



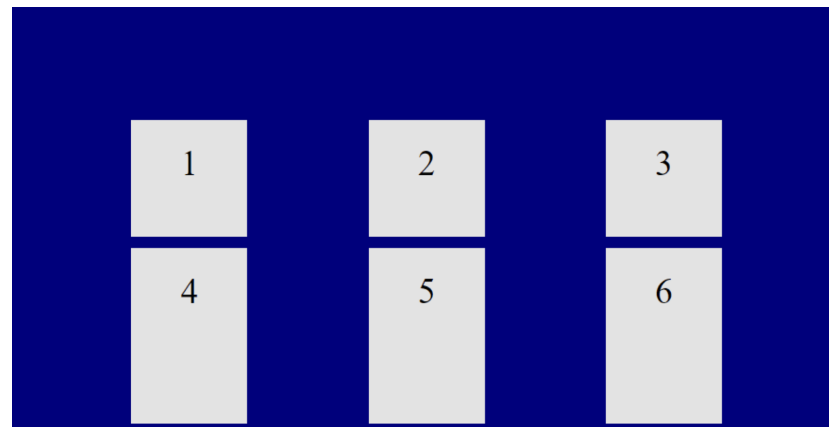
## GRID LINES



```
div.box1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```

<b>grid-template-columns</b>	Numero e larghezza di colonne (px, %, fr)
<b>grid-template-rows</b>	Numero e altezza di righe (px, %, fr)
<b>justify-content</b>	Start, End, Center, Space-evenly, Space-between, Space-around
<b>align-content</b>	Start, End, Center, Space-evenly, Space-between, Space-around

```
.grid {  
  display: grid;  
  grid-template-columns: 100px 100px 100px;  
  grid-template-rows: 100px 150px;  
  justify-content: space-evenly;  
  align-content: end;  
  grid-gap: 10px;  
  background-color: navy;  
  padding: 10px;  
  height: 350px;  
}
```



---

<b>Fr (fractional)</b>	1fr = 1 porzione di spazio rimanente. 1fr 2 fr = 33,3% 66,6%
<b>Minmax( <i>min,max</i> )</b>	Larghezza minima e massima di colonna. Es: <i>grid-template-columns: minmax(100px,40%), 1fr;</i>
<b>Repeat()</b>	Ripetizione di dichiarazione a scopo sintetico. Es: <i>grid-template-columns: repeat(4, 200px);</i>

---

## CSS Grid: elementi child (items)

<b>grid - column - start</b>	Inizio della colonna
<b>grid - column - end</b>	Fine della colonna
<b>grid - row - start</b>	Inizio della riga
<b>grid - row - end</b>	Fine della riga
<b>grid - column</b>	Sintassi abbreviata

```
div.box1 {  
  grid-column: 1 / 3;  
  grid-row: 1 / span 2;  
}
```

**grid -area**

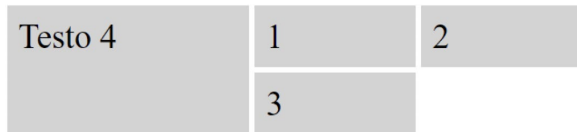
Sintassi abbreviata (row-start, column-start, row-end, column-end)

```
.grid {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 5px;  
}
```

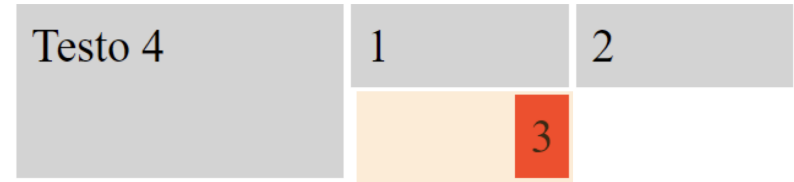
```
.grid > div {  
  background-color: lightgray;  
  padding: 10px;  
}
```

```
.box4 { grid-area: 1 / 1 / 3 / 2; }
```

```
<section class="grid">  
  <div class="box1">1</div>  
  <div class="box2">2</div>  
  <div class="box3">3</div>  
  <div class="box4">Testo 4</div>  
</section>
```

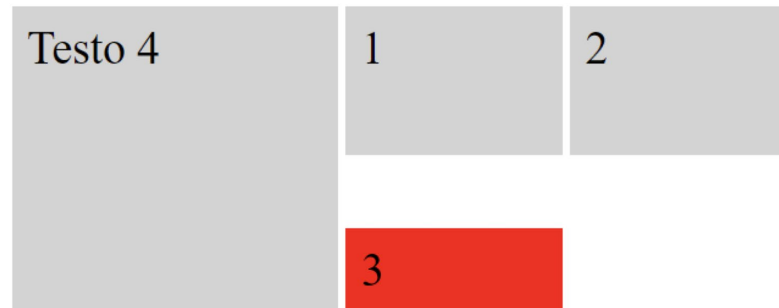


```
.grid .box3 {  
  background-color: ■ red;  
  justify-self: end;  
  /* start, center, end, stretch (default) */  
}
```





```
.grid {  
  display: grid;  
  height: 200px;  
}  
.grid .box3 {  
  background-color: red;  
  align-self: end;  
  /* start, center, end, stretch (default) */  
}
```



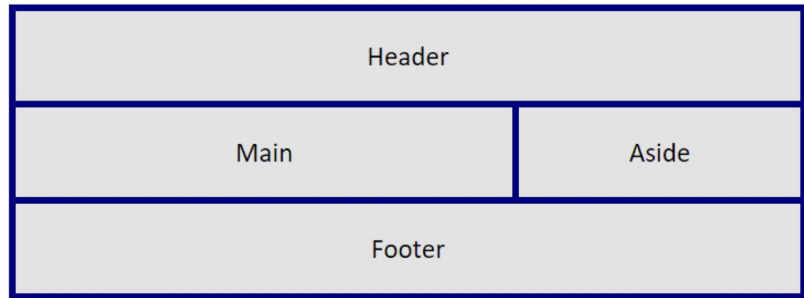
---

<b>grid -area</b>	Si assegna un nome
<b>grid -template -areas</b>	Nome della Grid-Area. Il «.» (punto) indica un item vuoto

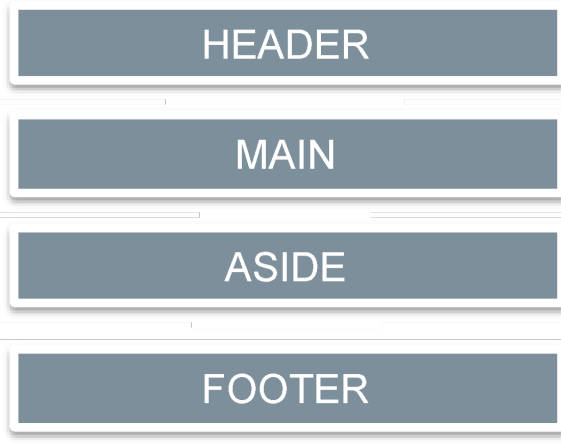
```
div.box1 {  
  grid-area: zona1;  
}  
.grid-container {  
  display: grid;  
  grid-template-areas: 'zona1 zona1 . . .';  
  /* 2 slot occupati + 3 senza nome */  
}
```

```
div.box1 { grid-area: header; }  
div.box2 { grid-area: aside; }  
div.box3 { grid-area: main; }  
div.box4 { grid-area: footer; }
```

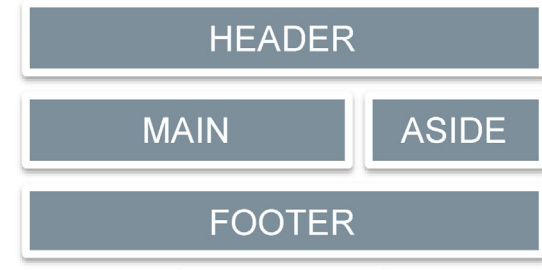
```
section.grid {  
  display: grid;  
  /* suddivisione in 3 colonne virtuali */  
  grid-template-areas:  
    'header header header'  
    'main main aside' /* il main occupa 2 colonne */  
    'footer footer footer';  
  grid-gap: 5px;  
  background-color: navy;  
  padding: 5px;  
}
```



## SU MOBILE



## MEDIA QUERY SU TABLET/DESKTOP





*shaping the skills of tomorrow*

challengenetwork.it

