Un contenitore flessibile: flexbox

Maria Chiara Pievatolo

Università di Pisa

pievatolo@dsp.unipi.it

26 novembre 2020

Sommario

Gli elementi a blocco e inline vanno in una sola direzione

"Float:left" induce i tre paragrafi colorati a allinearsi

Gli elementi a blocco e inline vanno in una sola direzione

Per impilarli devo usare condizionatamente “display:block”

Per cambiare direzione devo cambiare natura

Il foglio stile deve specificare la natura di ciascun elemento

```
/*Schermo grande*/
p.unof{width:30%;float:left;height:4em;}
p.duef {width:30%;float:left;height:4em;}
p.tref {width:30%;float:left;height:4em;}
/*Schermo piccolo*/
@media screen and (max-width: 980px) {
p.unof{width:90%;display:block;}
p.duef {width:90%;display:block;}
p.tref {width:90%;display:block;}}
```

Flexbox: mi fletto come vuoi

Bastano alcune indicazioni generali

```
/*Schermo grande*/
div.container {display:flex; flex-direction:row;
↪  justify-content:space-around;flex-wrap:wrap;}
div.container p {flex-basis:30%;margin:0px;
↪  padding:1em; color:#ffffff;height:4em; }
/*Schermo piccolo*/
@media screen and (max-width: 980px) { div.container
↪  {flex-direction:column; align-items:stretch;}
div.container p {flex-basis:90%;}}
```

Un contenitore con i suoi tre figli

Contenitore: qualsiasi elemento con dei discendenti

```
<div class="container">
<p class="unof"> Rosso </p>
<p class="duef"> Verde </p>
<p class="tref"><span id="grande">Con uno schermo
↪  grande mi allineo</span><span id="piccolo">Con
↪  uno schermo piccolo mi impilo</span></p>
</div>
```

display: flex ; flex-direction:

Il contenitore è una scatola flessibile...

```
div.container {display:flex; }
```

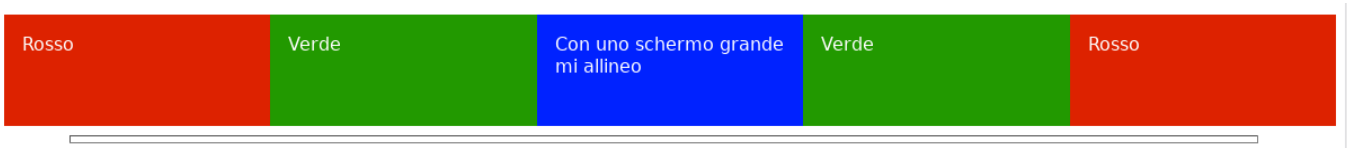
che allinea o incolonna quanto contiene: row ; column ; row-reverse ; column-reverse

```
div.container {display:flex;
flex-direction:row; }
```

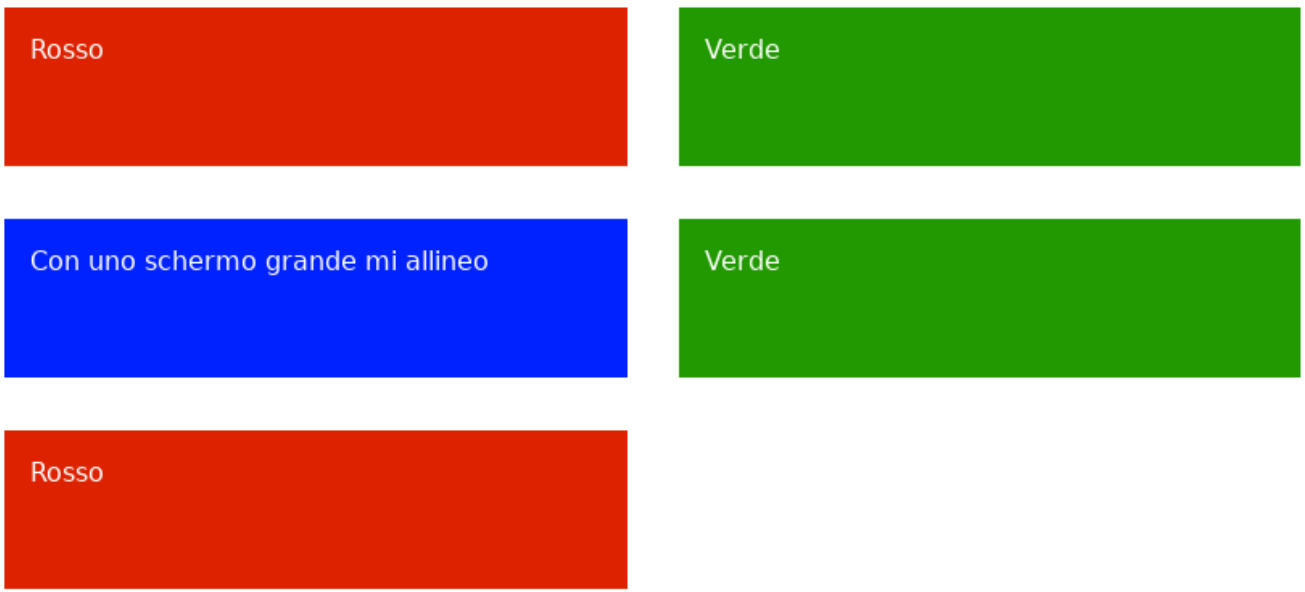
flex-wrap: nowrap wrap wrap-reverse

il cui contenuto si dispone su una riga, o si svolge su più righe dall'alto o dal basso

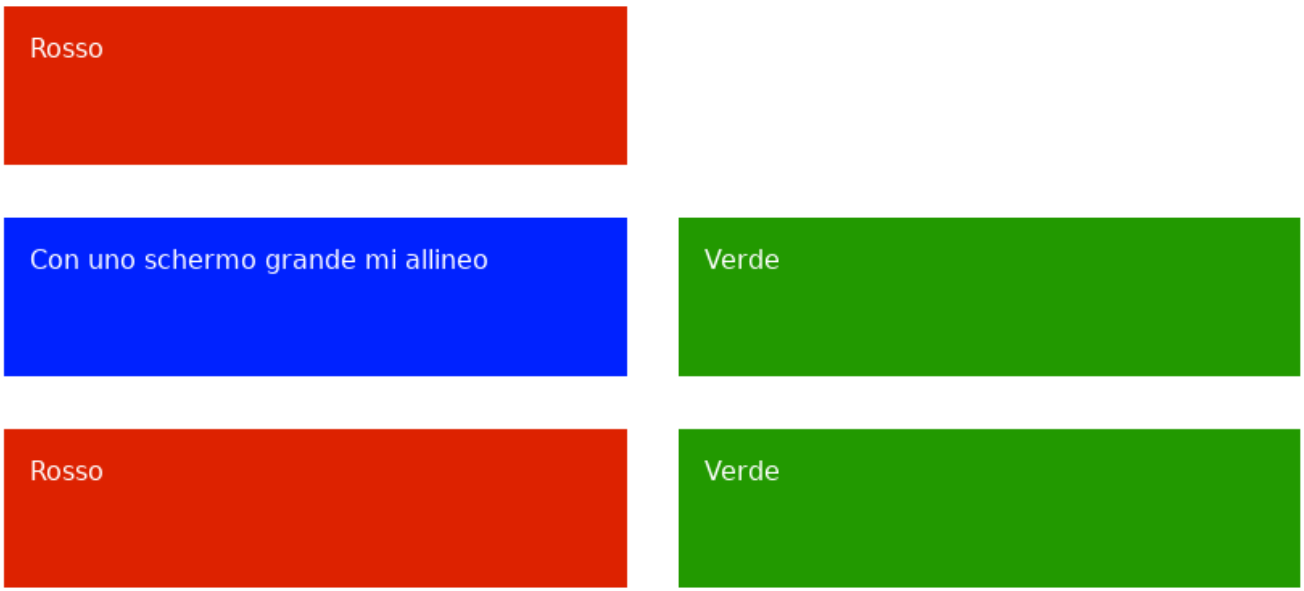
```
div.container {display:flex; flex-direction:row;
↪  flex-wrap:wrap;}
```



flex-wrap: nowrap



flex-wrap: wrap



flex-wrap: wrap-reverse

Stenograficamente:

Default: row, nowrap

```
/*flex-flow: <‘valore di flex-direction’> || <‘valore
↪  di flex-wrap’>*/
```

Per esempio

```
div.container {display:flex; flex-flow: row wrap; }
```

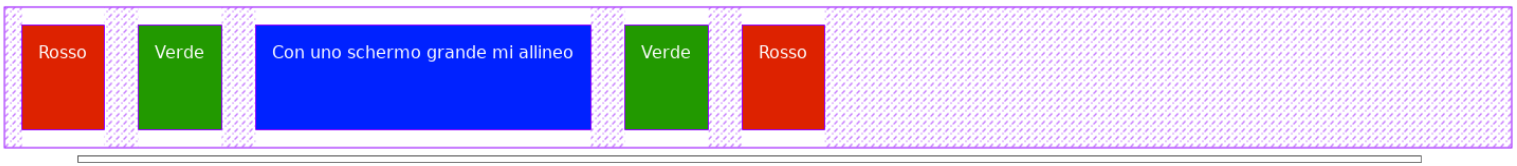
justify-content: flex-start flex-end center space-between space-around space-evenly stretch

```
div.container {display:flex; flex-flow: row wrap;
↪  justify-content:space-around;}
```

justify-content: flex-start (valore predefinito)

Gli elementi discendenti si dispongono all'inizio della riga

```
div.container {display:flex; flex-direction:row;
↪  justify-content:flex-start;}
```

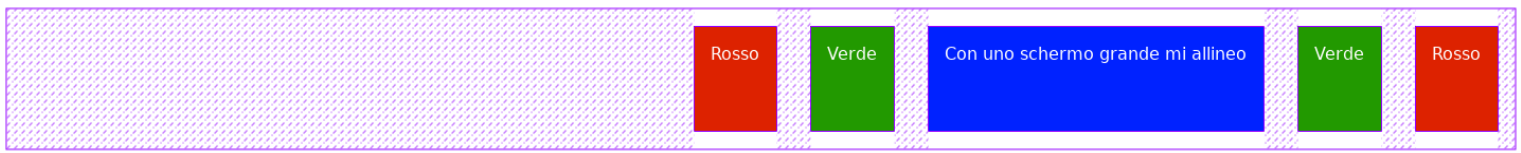


flex-start

justify-content: flex-end

Gli elementi discendenti si dispongono alla fine della riga

```
div.container {display:flex; flex-direction:row;
↪  justify-content:flex-end;}
```

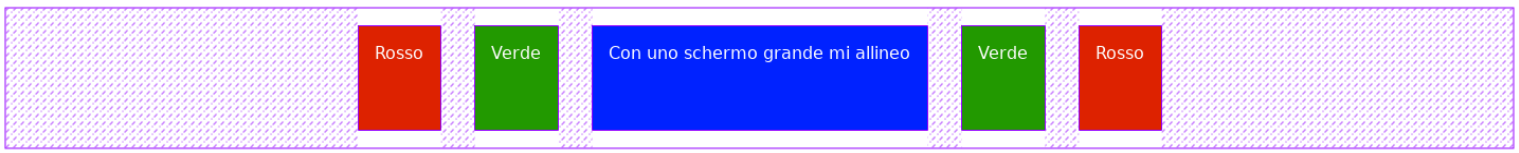


flex-end

justify-content: center

Gli elementi discendenti si dispongono al centro della riga

```
div.container {display:flex; flex-direction:row;
↪  justify-content:center;}
```

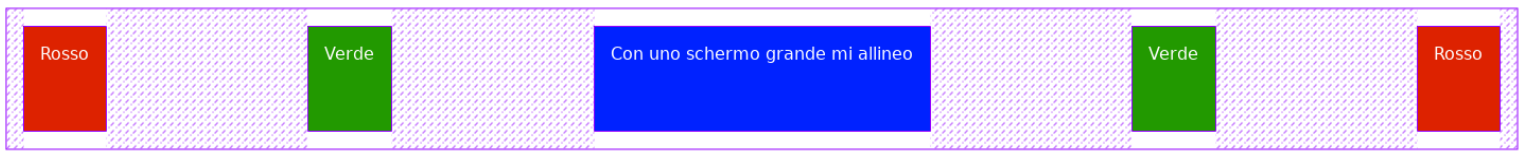


flex-end

justify-content: space-between

Gli elementi discendenti si dispongono su tutta la riga, a partire dalle estremità

```
div.container {display:flex; flex-direction:row;
↪  justify-content:space-between;}
```

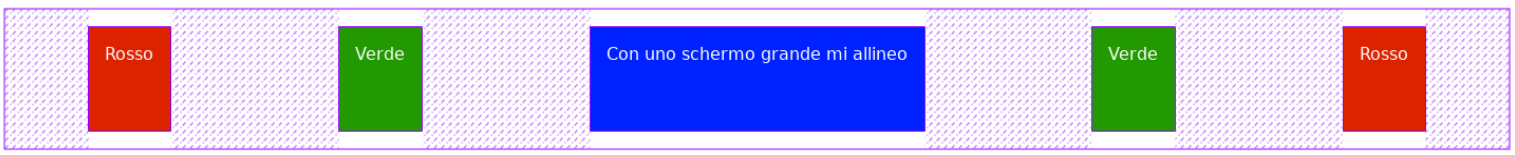


space-between

justify-content: space-around

Gli elementi discendenti si dispongono su tutta la riga, ma con uno spazio uguale *attorno* a ciascuno

```
div.container {display:flex; flex-direction:row;
↪  justify-content:space-around;}
```

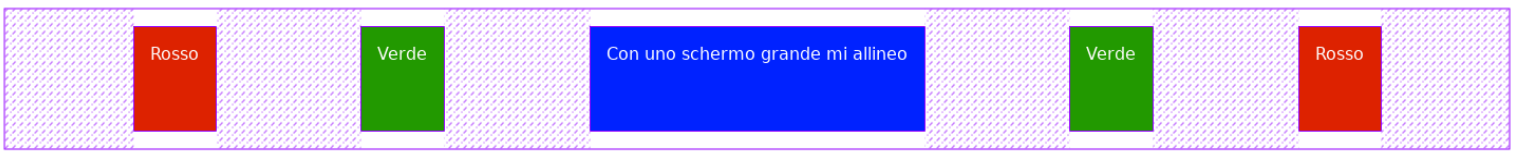


space-around

justify-content: space-evenly

Gli elementi discendenti si dispongono su tutta la riga, ma in modo che lo spazio che li separa reciprocamente sia uguale

```
div.container {display:flex; flex-direction:row;
↪  justify-content:space-evenly;}
```



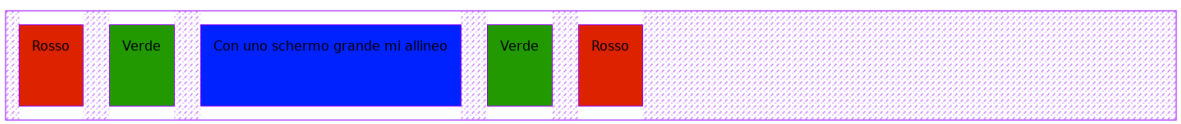
space-evenly

justify-content: stretch

Gli elementi discendenti cercano di riempire la riga

```
div.container {display:flex; flex-direction:row;
↪  justify-content:stretch;}
```

stretch con flex-basis predeterminata per i discendenti



stretch senza flex-basis predeterminata per i discendenti

justify-content: safe o unsafe ...

Voglio o no proteggermi dalle tracimazioni?

```
/*Certo!*/
div.container {display:flex; flex-direction:row;
↪  justify-content: safe center;}
/*No!*/
div.container {display:flex; flex-direction:row;
↪  justify-content: unsafe center;}
```

align-items: stretch flex-start flex-end center baseline

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: stretch;}
```

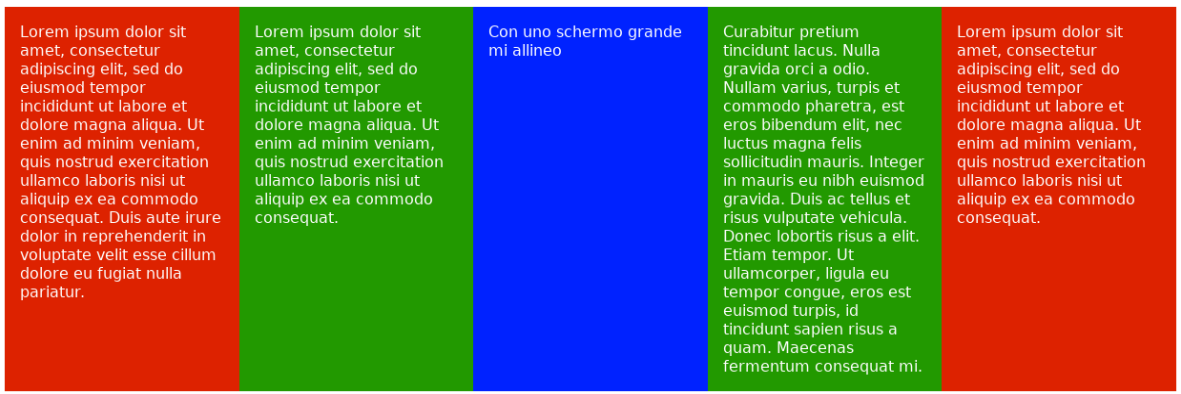
Align-items è diverso da align-content!

"align-items" governa distributivamente l'allineamento verticale riga per riga, non collettivamente tutto il contenuto!

align-items: stretch (valore predefinito)

Gli elementi discendenti riempiono verticalmente ogni riga

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: stretch;}
```

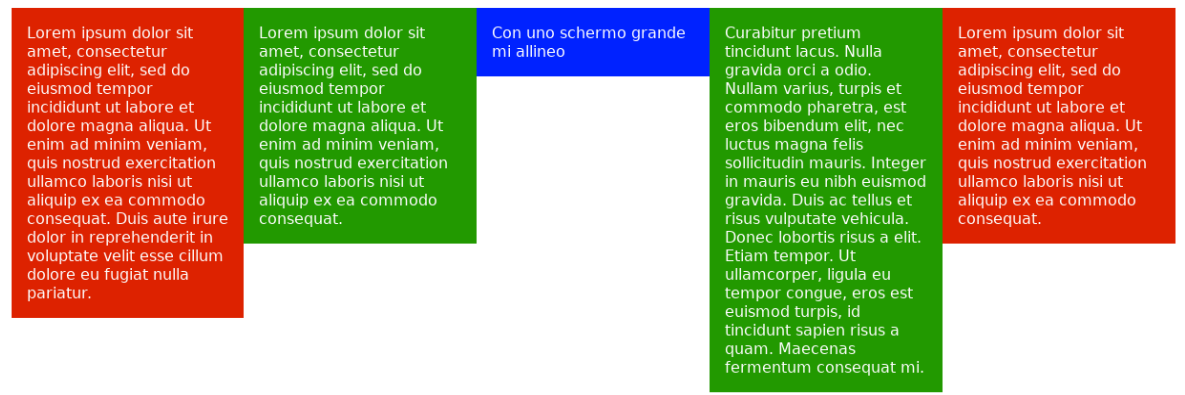


align-items: stretch;

align-items: flex-start

Gli elementi discendenti si allineano in alto

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: flex-start;}
```

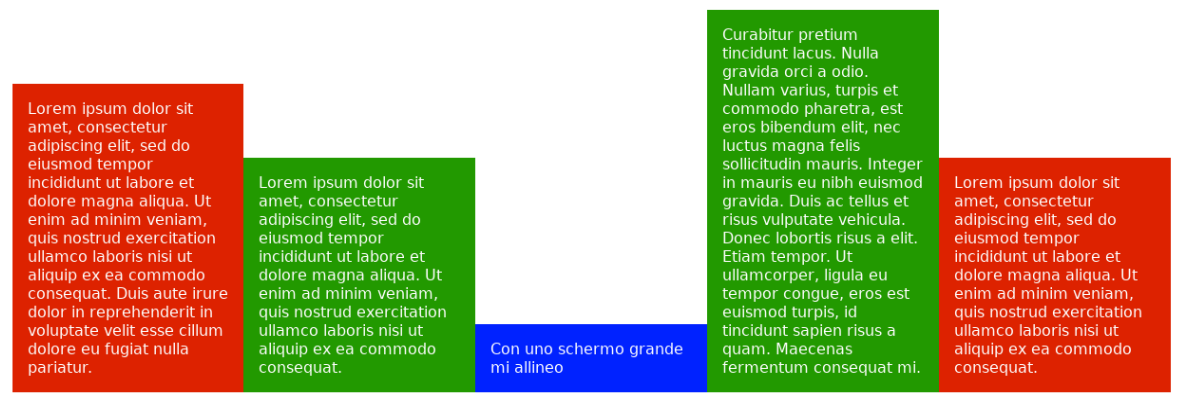


align-items: flex-start;

align-items: flex-end

Gli elementi discendenti si allineano in basso

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: flex-end;}
```

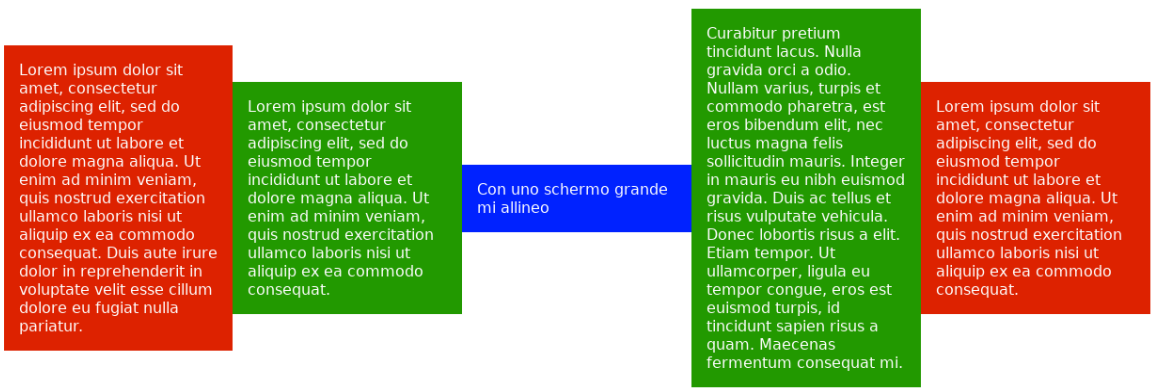


align-items: flex-end;

align-items: center

Gli elementi discendenti si allineano al centro

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: center;}
```

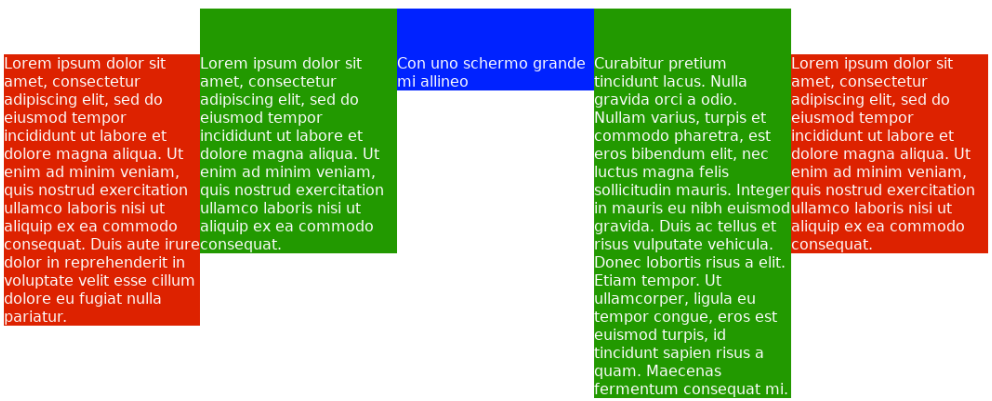


align-items: center;

align-items: baseline

Gli elementi discendenti si allineano in modo che l'inizio del testo che contengono si allinei alla medesima altezza

```
div.container {display:flex; flex-flow: row wrap;
↪  align-items: baseline;}
```



align-items: baseline;

Allineamento verticale

align-content: flex-start flex-end center space-between space-around space-evenly stretch ...

```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content: stretch;}
```

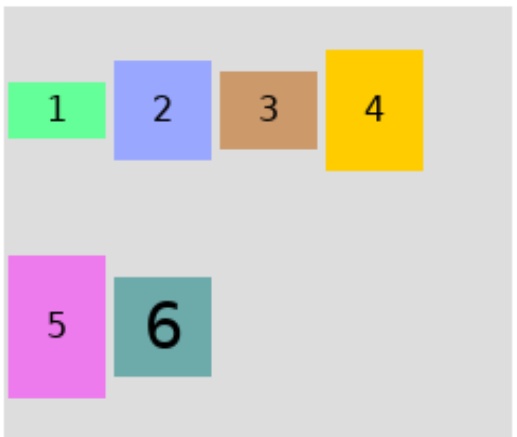
N.B.

Ha senso usare align-content solo se i discendenti si dispongono su più di una riga.

Stretch

align-content: stretch: si estende verticalmente per riempire lo spazio

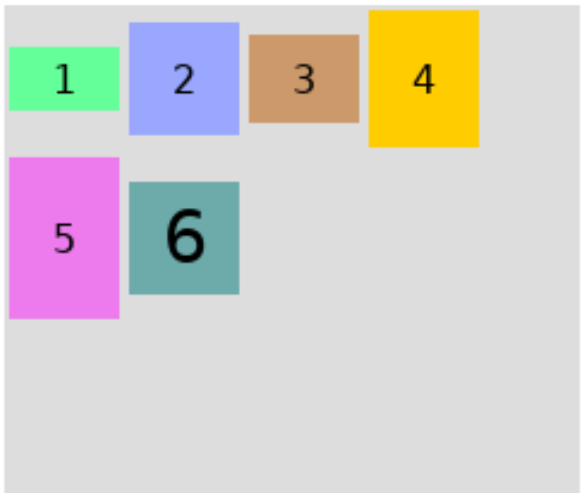
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content: stretch;}
```



Flex-start

align-content: flex-start: si raggruppa verso l'alto

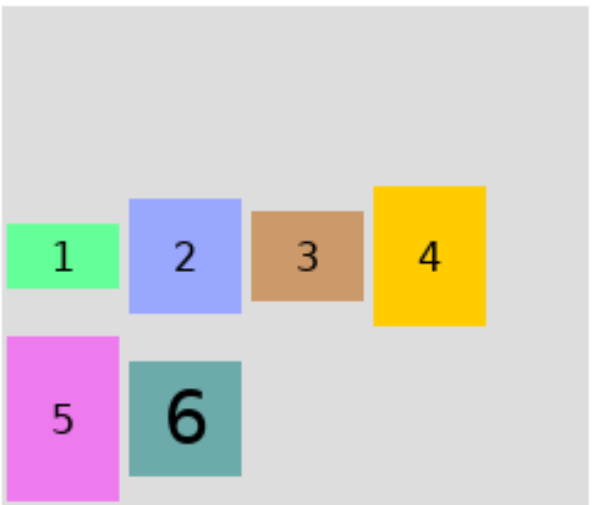
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content: flex-start;}
```



Flex-end

align-content: flex-end: si raggruppa verso il basso

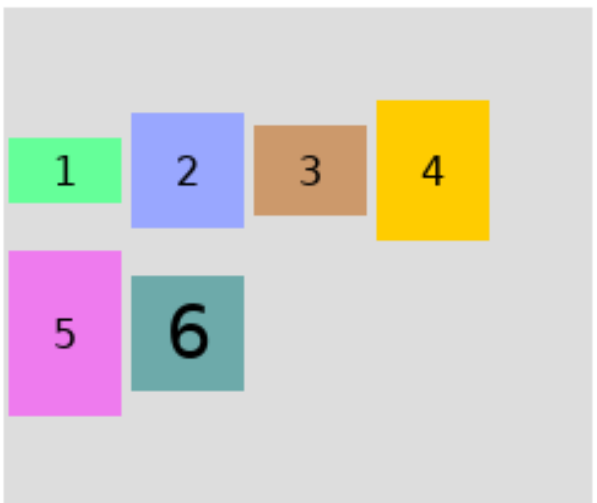
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content: flex-end;}
```



Center

align-content: center: si raggruppa al centro

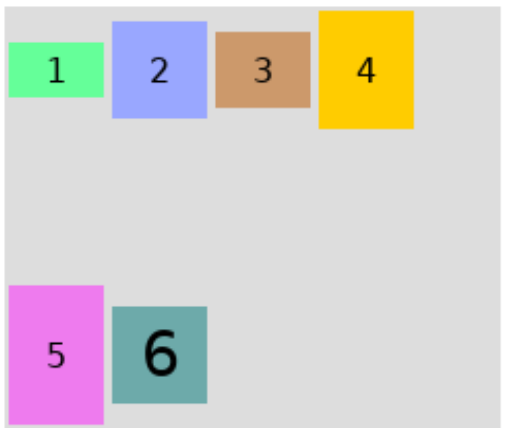
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content: center;}
```



Space-between

align-content: space-between: i contenuti si distribuiscono a intervalli regolari, ma iniziando dal bordo superiore e finendo a quello inferiore

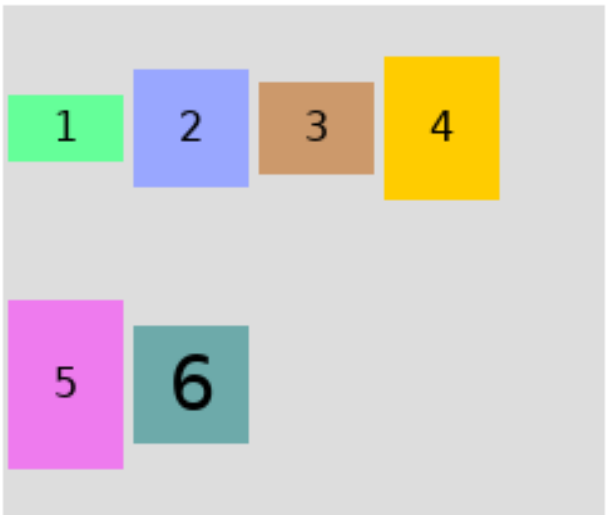
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content:
↪  space-between;}
```



Space-around

align-content: space-around: i contenuti si distribuiscono a intervalli regolari, con spazi uguali fra le righe

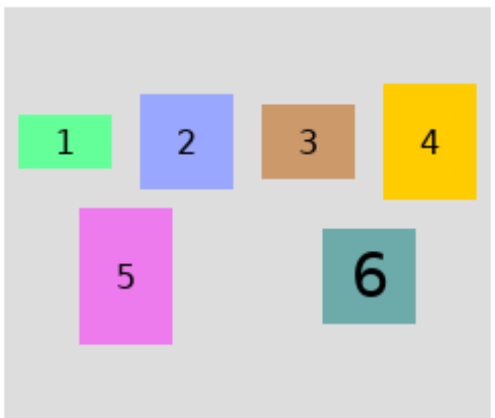
```
#contenitore {height:200px;width: 240px;display:
↪  flex; flex-wrap: wrap;align-content:
↪  space-around;}
```



Un centro perfetto

Margin-auto in flex assorbe tutto lo spazio in più, in questo caso ugualmente da una parte e dall'altra

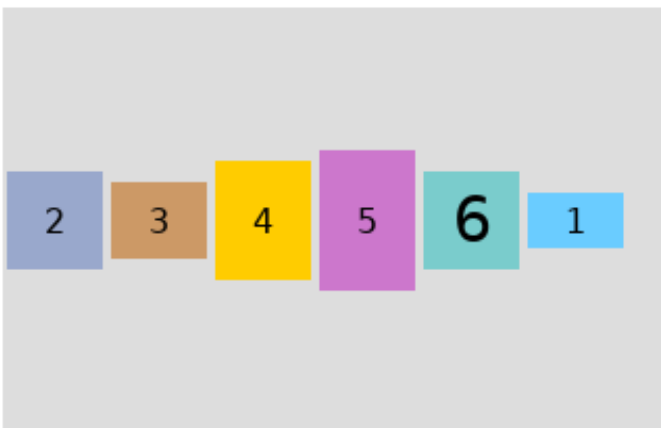
```
div > div {box-sizing: border-box; border: 2px solid
↪  #dddddd; width: 50px; display: flex; align-items:
↪  center; justify-content: center;  margin:auto;}
```



Order!

Per permettere a uno o più discendenti di trasgredire la sequenza presente nel documento HTML

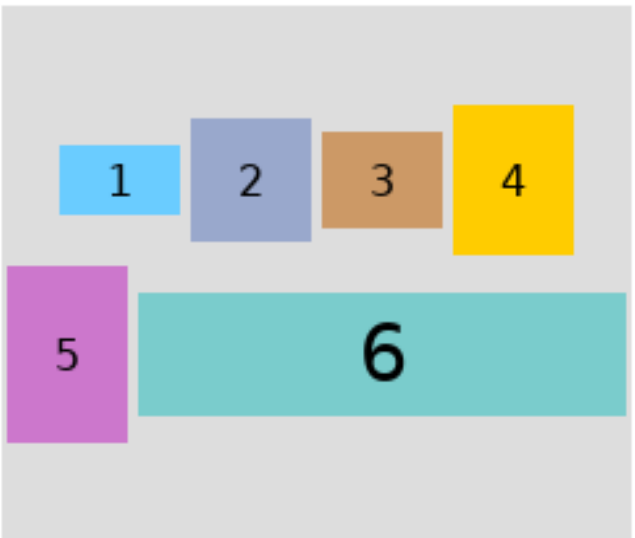
```
#figlio1 {background-color: #6accff; min-height:
↪  30px; order:6;}
```



Flex-grow

Per permettere a uno o più discendenti di occupare più spazio

```
#figlio6 {background-color: #7acccc; min-height:
↪  50px; font-size: 30px; flex-grow:2;}
```

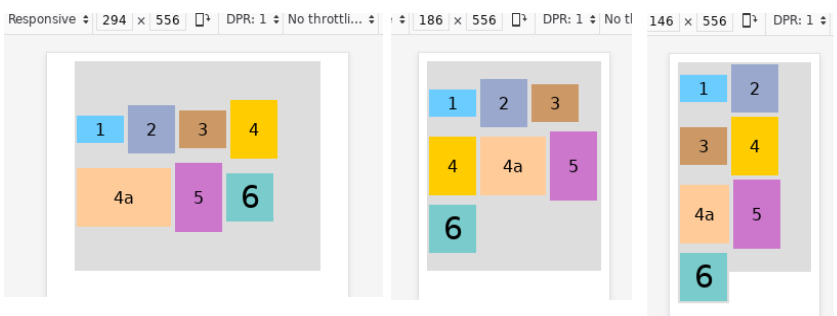


6 prende il doppio dello spazio rimanente

Flex-shrink

Per indicare a uno o più discendenti - *se flessibili* - di strizzarsi, quando non c'è abbastanza spazio nella riga

```
#contenitore {height:200px;width: 240px;align-items:
↪  center; background-color: #dddddd; display: flex;
↪  flex-wrap: wrap; align-content: safe center;}
#figlio4a {background-color: #ffcc99; min-height:
↪  60px; flex-basis:40%; flex-shrink:2;}
```



4a si strizza quando lo schermo si stringe

Flex-basis: dimensione di un figlio prima della distribuzione dello spazio ulteriore

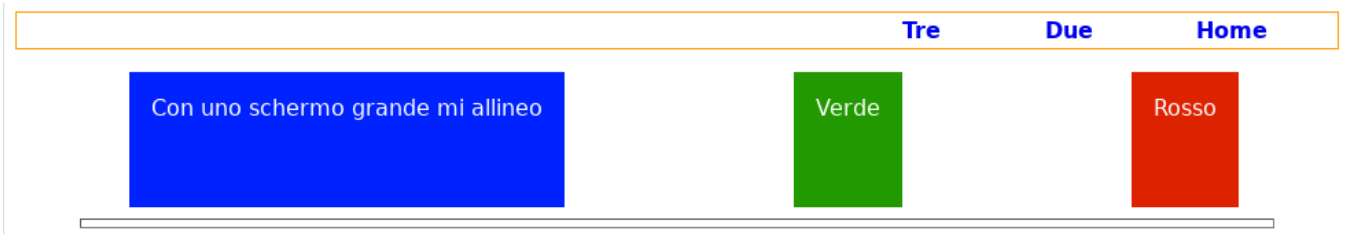
flex-basis: numero, auto o content

```
div.container {display:flex; flex-direction:row;
 ↪  justify-content:space-around;flex-wrap:wrap;}
div.container p {flex-basis:30%;}
@media screen and (max-width: 980px) { div.container
 ↪  {flex-direction:column;
 ↪  align-items:stretch;}div.container p
 ↪  {flex-basis:90%;}}
```

30 per cento e 90 per cento

Flex-basis: 0 e auto

Flex-basis:0 minimizza la larghezza degli elementi e distribuisce tutto lo spazio ulteriore



Flex-basis:auto considera la larghezza degli elementi e distribuisce tutto lo spazio ulteriore tenendo conto del valore di flex-grow

Stenografia: flex per flex-grow + (flex-shrink e flex-basis)

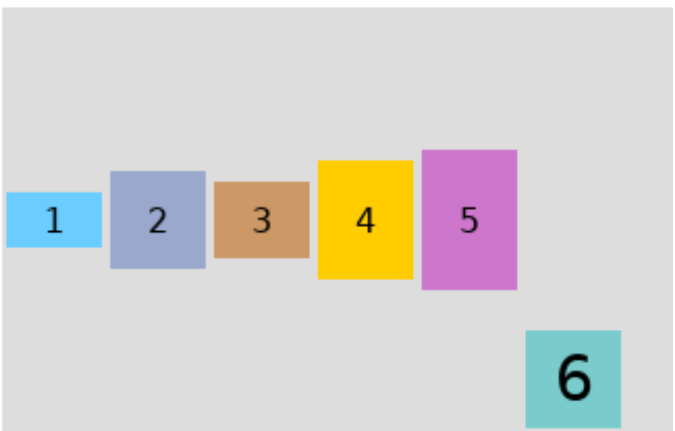
Indicando il valore di flex-grow, gli altri due vengono dedotti

- flex:1 o altro numero intero = flex-grow: 1 o altro numero intero , flex-shrink: 1 , flex-basis: 0 (figli flessibili senza spazio vitale)
- flex: auto = flex-grow: 1, flex-shrink: 1, flex-basis: auto (figli flessibili con spazio vitale)
- flex:none = flex-grow: 0, flex-shrink: 0, flex-basis: auto (figli inflessibili con spazio vitale)
- flex: initial =flex-grow: 0, flex-shrink: 1, flex-basis: auto (figli "altruisti" con spazio vitale)

Figli anticonformisti: align-self

Per permettere a un discendente di allinearsi in modo diverso dai suoi fratelli

```
#figlio6 {background-color: #7acccc; min-height:
↪  50px; font-size: 30px; align-self:flex-end;}
```



Io non mi allineo al centro!

vw = viewport width

Percentuale delle dimensioni attuali dell'area di visualizzazione

1vw =1 per cento dell'area di visualizzazione
2vw = 2 per cento dell'area di visualizzazione
. . . e così via.

Esercizio: da *frangar non flectar* a *flectar non frangar*

- Prendo il documento rigido all'url `https://elearning.sp.unipi.it/mod/resource/view.php?id=6280`
- Cerco di renderlo flessibile con la flexbox
- Confronto la mia soluzione con quelle proposte, basate sul vecchio metodo

Micro-bibliografia

CSS Tricks
A Complete Guide to Flexbox
https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Greg Smith (2012)
Dive into Flexbox
https://bocoup.com/blog/dive-into-flexbox

Continua?