

CSS

cascading style sheet

Sintassi CSS

```
1 selettore {  
2     proprieta: valore;  
3     proprieta: valore;  
4 }  
5  
6  
7 /* Questo è un commento */
```

Selezione con *

Il selettore universale serve a dare delle regole generali a tutti gli elementi.

Queste regole potranno poi essere sovrascritte da altri selettori inseriti successivamente.

HTML

```
1 <h1>
2     Questo testo sarà rosso
3 </h1>
4
5 <p>
6     e anche questo
7 </p>
```

CSS

```
1 /* I testi di tutti gli elementi saranno rossi */
2
3 * {
4     color: red;
5 }
```

Selezione con il tag

Tutti i tag di quel tipo avranno le proprietà definite nel css.

In questo caso, tutti i paragrafi saranno di colore rosso.

HTML

```
1 <h1>Questo titolo non sarà rosso</h1>
2
3 <p>Questo testo sarà rosso e così ogni tag p</p>
4
5 <p>Anche questo testo sarà rosso</p>
6
```

CSS

```
1 /* tutti i tag p saranno rossi */
2 p {
3     color: red;
4 }
```

Selezione con classe

Tutti i tag a cui è stata assegnata la stessa classe avranno le proprietà definite nel css.

In questo esempio, tutti gli elementi con classe "bot" saranno di colore rosso.

index.html

```
1 <h1 class="bot">
2     Questo testo sarà rosso e così ogni tag con
   classe bot
3 </h1>
4
5 <p class="bot">
6     Anche questo paragrafo sarà rosso
7 </p>
8
9 <p>
10    questo paragrafo, invece, NON sarà rosso
11 </p>
12
```

style.css

```
1 /* tutti i tag con classe bot saranno rossi */
2 .bot {
3     color: red;
4 }
```

Dove si scrive il codice CSS?

Separato dall'HTML, ma pur sempre collegato!

In un file dedicato con estensione `.css`, il quale viene collegato alla pagina nel tag head

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <!-- ... -->
5   <link rel="stylesheet" href="style.css">
6 </head>
7 <body>
8
9   <h1>Questo titolo sarà blu</h1>
10
11   <p class="bot">Questo testo sarà rosso</p>
12
13 </body>
14 </html>
```

style.css

```
1 h1 {
2   color: blue;
3 }
4
5 .bot {
6   color: red;
7 }
```

Il Testo

La presentazione del testo può essere modificata in vari modi.

Possiamo ad esempio rendere il testo sottolineato, maiuscolo oppure modificarne l'allineamento.

Allineamento del testo



```
1 h1 {  
2   /* grafia del testo */  
3   text-transform: uppercase;  
4  
5   /* allineamento del testo */  
6   text-align: center;  
7  
8 }
```

Font

Il font è lo "stile" delle lettere utilizzate per il testo. Lo stesso termine viene utilizzato anche in editoria.

Con il CSS possiamo andare a modificare l'aspetto del testo in tanti modi, impostando ad esempio, la tipologia di font, la dimensione o lo spessore.

```
1 p {  
2  
3   /* dimensione del font */  
4   font-size: 12px;  
5  
6   /* spessore del font */  
7   font-weight: bold;  
8  
9   /* nome o tipologia del font */  
10  font-family: Arial, sans-serif;  
11  
12 }
```

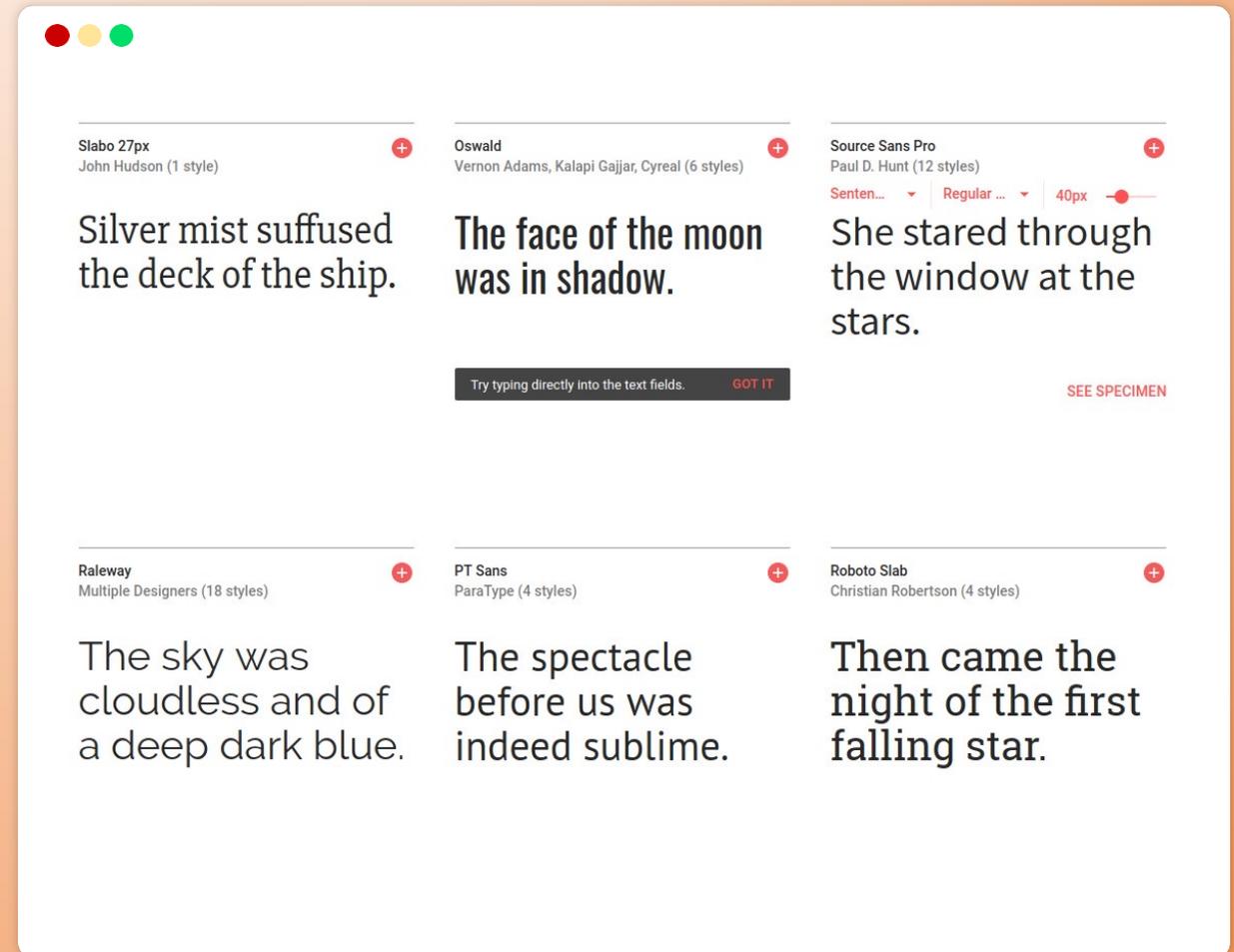
Font Family

Web Safe Fonts

Font che sono (molto probabilmente) installati sul pc dell'utente e quindi sono "sicuri" da utilizzare (Arial, Verdana, ...) ([web safe fonts](#))

Google Fonts

Font che non sono installati sul pc dell'utente ma ogni utente dovrà scaricarne una copia da Google Fonts ([fonts.google.com](#))



Color

Il colore del testo

La proprietà **color** permette di impostare il colore del testo.

Il valore viene indicato utilizzando una delle notazioni valide per i colori.

```
1 p {  
2   /* notazione esadecimale */  
3   color: #A2CC85;  
4  
5   /* notazione rgb */  
6   color: rgb(162, 204, 133);  
7 }  
8  
9 /* Nota: ci sono anche altre proprietà che usano il colore ad esempio border e background-color */
```

Width e Height

Per impostare la larghezza e l'altezza del contenuto dell'elemento

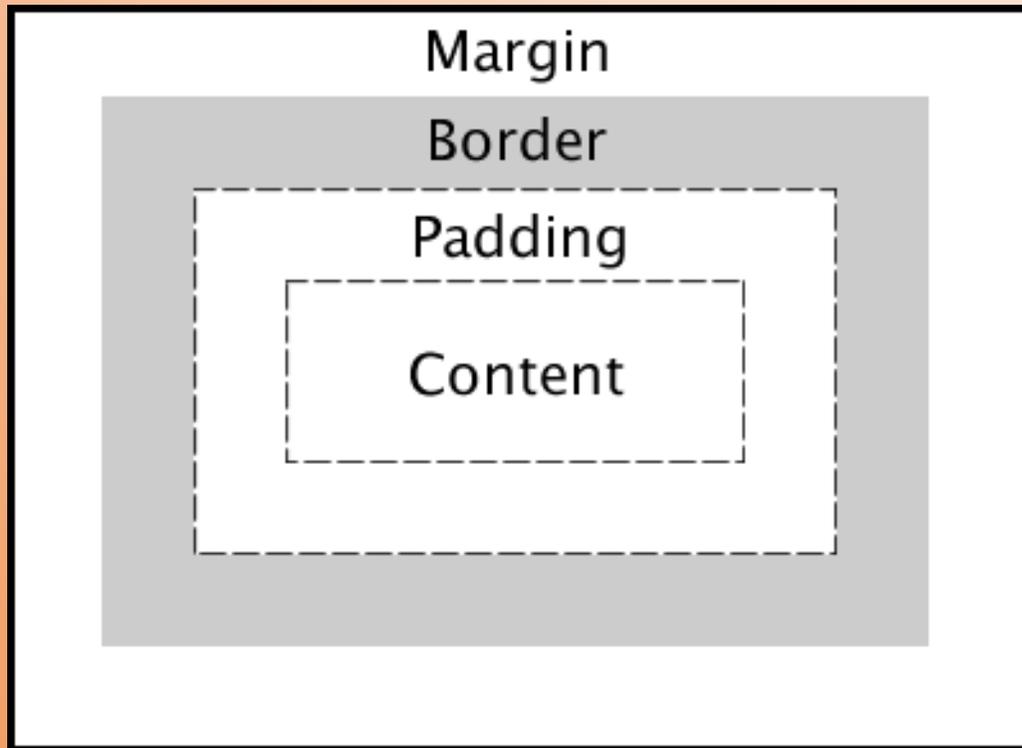
Ci sono numerose unità di misura utilizzabili, ma le unità più comunemente usate sono:

- **px** (pixel)
- **%** (percentuale rispetto al tag contenitore)

```
1 div {  
2     width: 80%;  
3     height: 80px;  
4 }
```

Bordi, Margini e Padding, cosa sono?

Surprise surprise, non è così banale



Margin

Spazio esterno al bordo

Specifica la distanza tra il bordo di un elemento ed il bordo degli elementi adiacenti.

Può essere indicato in uno dei seguenti modi:

Con un'unità di misura, preferibilmente in pixel

Come percentuale delle dimensioni del "contenitore"

Se necessario, è possibile definire i margini dei quattro lati singolarmente, per poterli stilare in modo differente e indipendente.

```
1 div {
2   /* Specifichiamo tutti i lati */
3   margin: 10px;
4
5   /* Specifichiamo sopra/sotto e destra/sinistra */
6   margin: 10px 5px;
7
8   /* Specifichiamo ogni lato, in senso orario */
9   margin: 10px 5px 20px 15px;
10
11  /* Specificiamo solo un lato */
12  margin-right: 20px;
13 }
```

Padding

Spazio interno al bordo

Il **padding** imposta la distanza tra il contenuto ed il bordo del blocco.

Può essere indicato in uno dei seguenti modi:

- Con un'unità di misura, preferibilmente in pixel
- Come percentuale delle dimensioni del "contenitore"

Valgono le stesse regole già dette per **margin** per quanto riguarda impostare singolarmente ogni lato e sulle forme abbreviate.

```
1 div {
2   /* Specifichiamo tutti i lati */
3   padding: 10px;
4
5   /* Specifichiamo sopra/sotto e destra/sinistra */
6   padding: 10px 5px;
7
8   /* Specifichiamo ogni lato, in senso orario */
9   padding: 10px 5px 20px 15px;
10
11  /* Specificiamo solo un lato */
12  padding-right: 20px;
13 }
```



Di default le dimensioni del padding si sommano alle dimensioni di width e height del blocco, quindi aumentano le dimensioni totali del blocco.

Box Sizing

La rivoluzione nel calcolo del dimensionamento dei blocchi

Prima dell'introduzione del concetto di box-sizing, il dimensionamento dei blocchi avveniva solo in una modalità:

width ed height determinano la dimensione del contenuto, e padding e border-width vengono ad esso aggiunti per stabilire lo spazio realmente occupato dal blocco.

Adesso è possibile utilizzare due metodi diversi di **box-sizing**:

- **content-box**
la "vecchia" modalità di default, in cui **padding e border si sommano** a width e height.
- **border-box**
padding e border vengono assorbiti da width e height.

CSS Reset

La prima regola CSS da scrivere

E' buona norma iniziare ogni CSS con una regola che "azzerà" margin, padding e imposta il box-sizing a border-box.

In questo modo, non subiremo le scelte "preconfezionate" dei browser e decideremo noi margin e padding di ogni elemento.

Ovviamente, non è obbligatorio ma è un **ottimo consiglio**

```
1 * {  
2   margin: 0;  
3   padding: 0;  
4   box-sizing: border-box;  
5 }  
6  
7 /* Le regole successive potranno  
8   sovrascriverle quando serve! */
```

CSS Flexbox

Che cosa fa Flexbox?

Il CSS3 **FlexBox** è una modalità di layout che prevede la disposizione degli elementi in una pagina in modo «flessibile», con la possibilità di adattare il contenuto più facilmente a schermi e dispositivi diversi.

Gli elementi che vogliamo siano disposti in modo flessibile devono essere contenuti in un **genitore** «Flex», detto «**Flex-Container**».

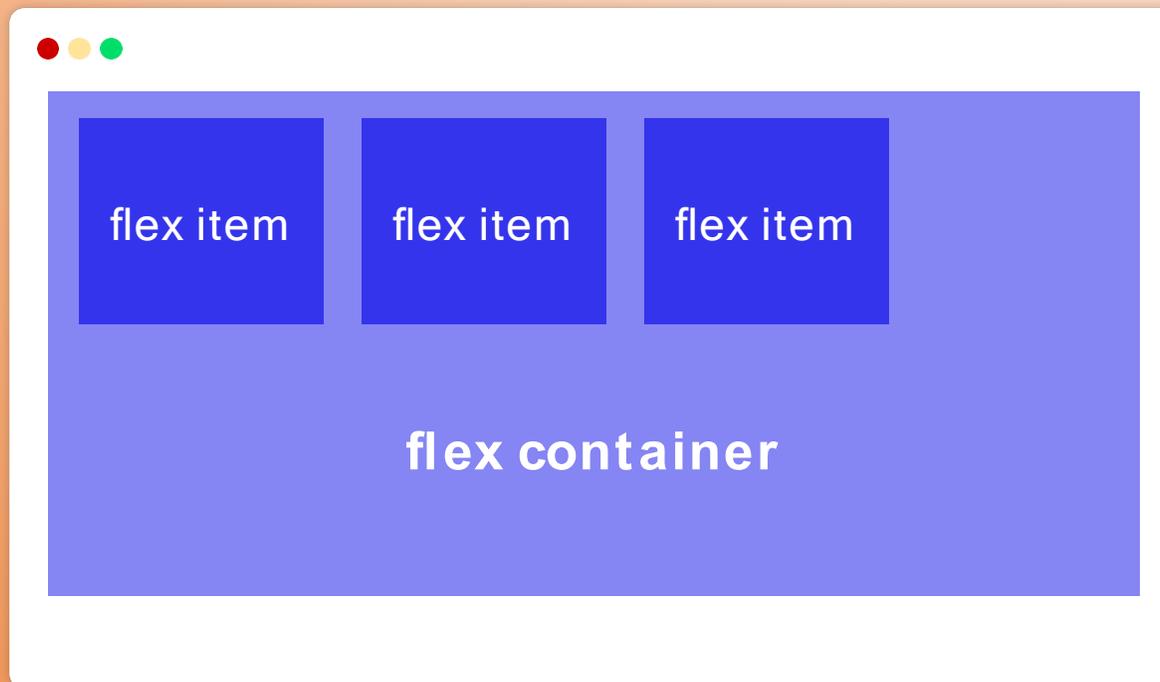
Gli elementi “**figli**” in un contenitore «Flex» sono detti «**Flex-Item**».

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

CSS Flexbox

Display: flex è una proprietà del genitore!

La proprietà **display: flex** va data al tag genitore (**flex-container**).
I suoi figli diventeranno automaticamente **flex-item**.



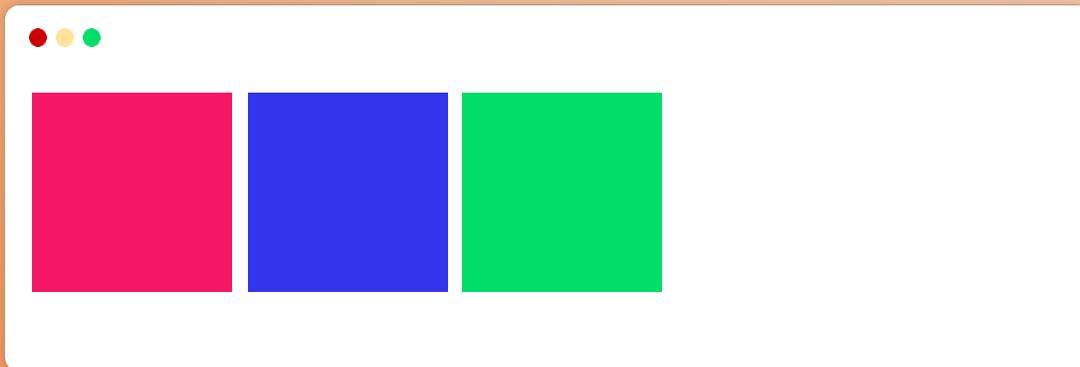
```
1 <div class="container-flex">
2   <div class="item"></div>
3   <div class="item"></div>
4   <div class="item"></div>
5 </div>
```

```
1 .container-flex {
2   display: flex;
3 }
```

CSS Flexbox

flex-direction - Flex-container rule

Di default, la direzione in cui vengono allineati i flex-item è row: **qualunque tipo di elemento verrà sistemato sulla stessa riga.**

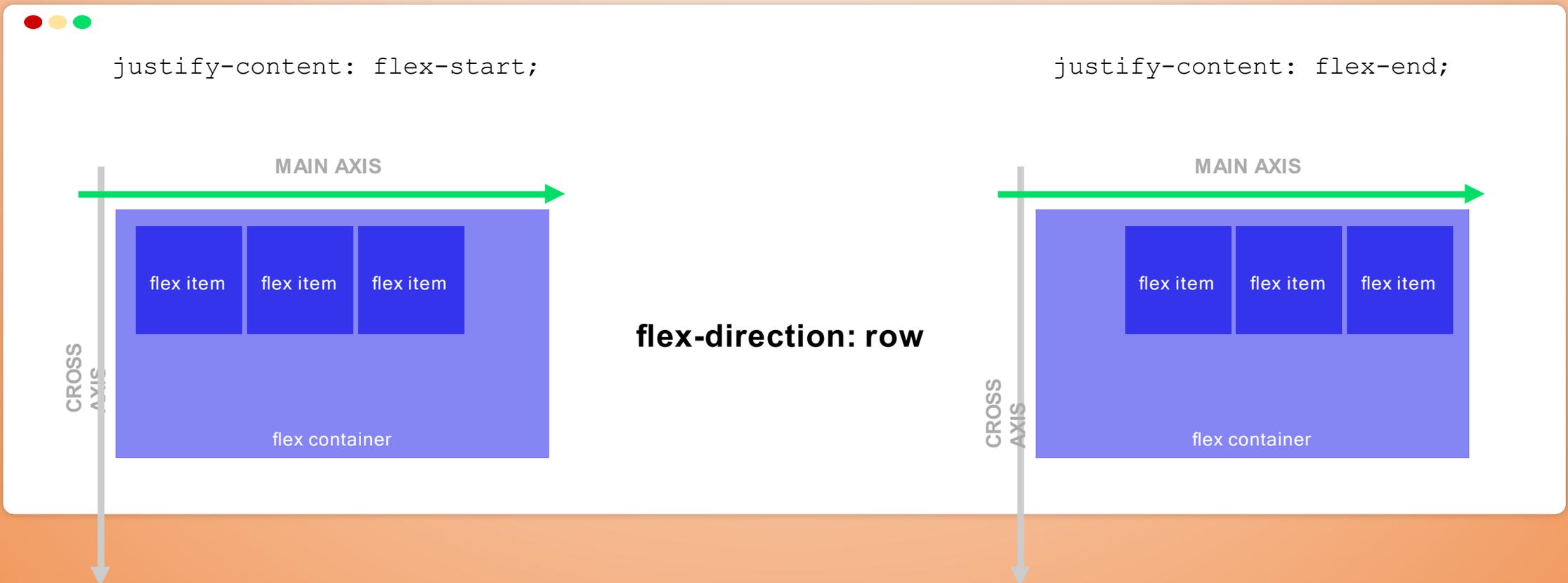


```
1 <div class="container-flex">
2   <div class="pink"></div>
3   <div class="blue"></div>
4   <div class="green"></div>
5 </div>
```

```
1 .container-flex {
2   display: flex;
3 }
```

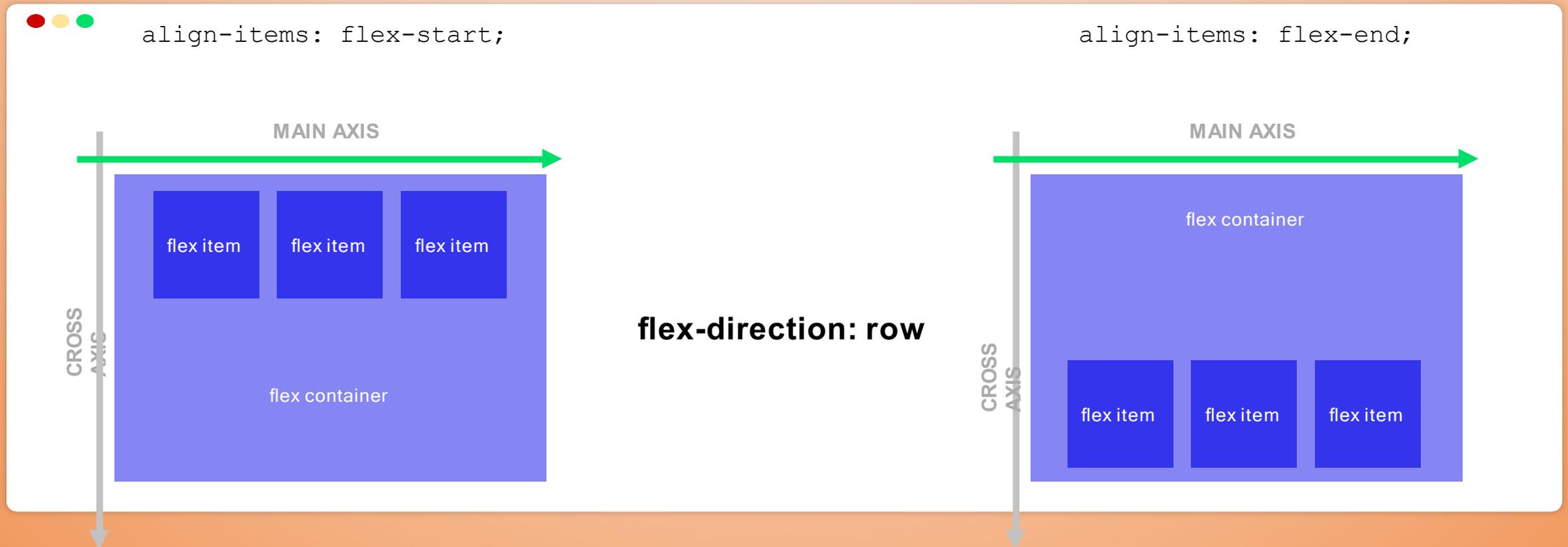
Possiamo allineare gli item sul Main-Axis

`justify-content` - Flex-container rule



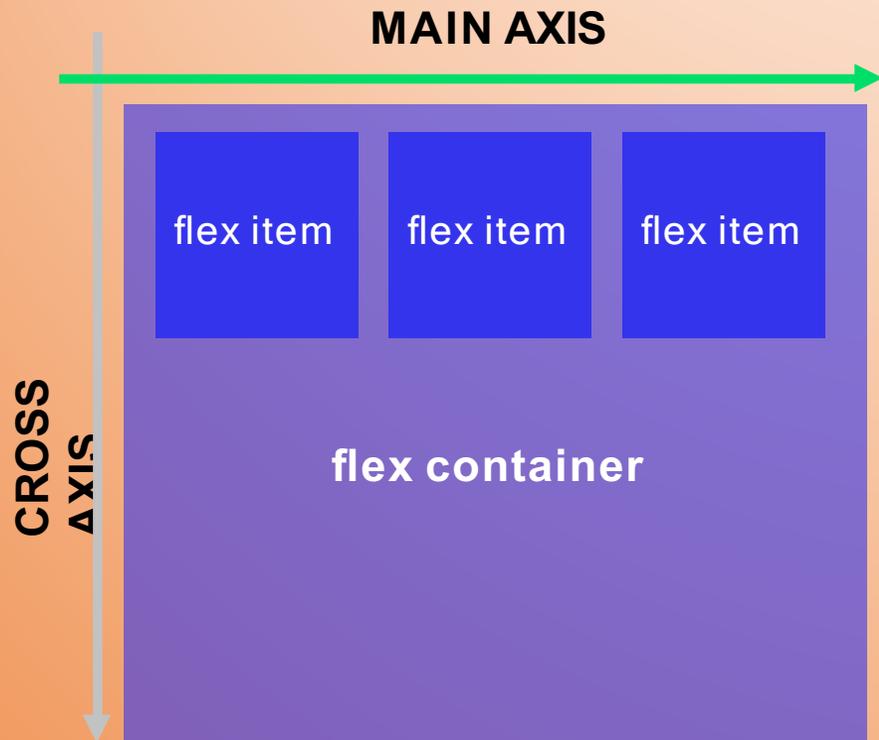
Possiamo allineare gli item sul Cross-Axis

align-items - Flex-container rule

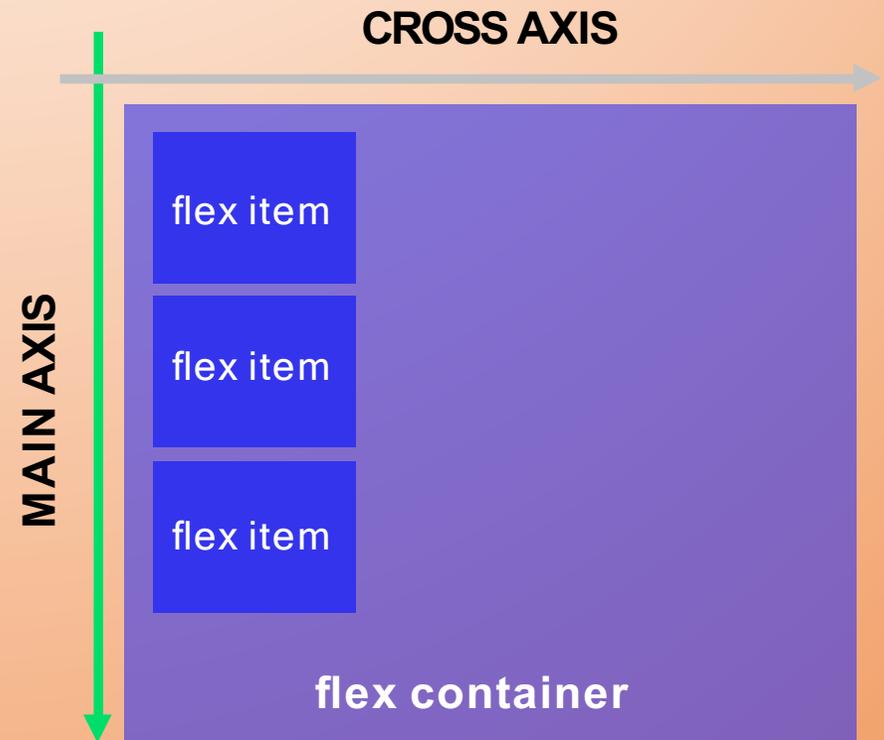


CSS Flexbox

`flex-direction: row`



`flex-direction: column`



Se cambiamo la direction tutto si sposta!

`justify-content` - Flex-container rule

```
justify-content: flex-start;
```

```
justify-content: flex-end;
```

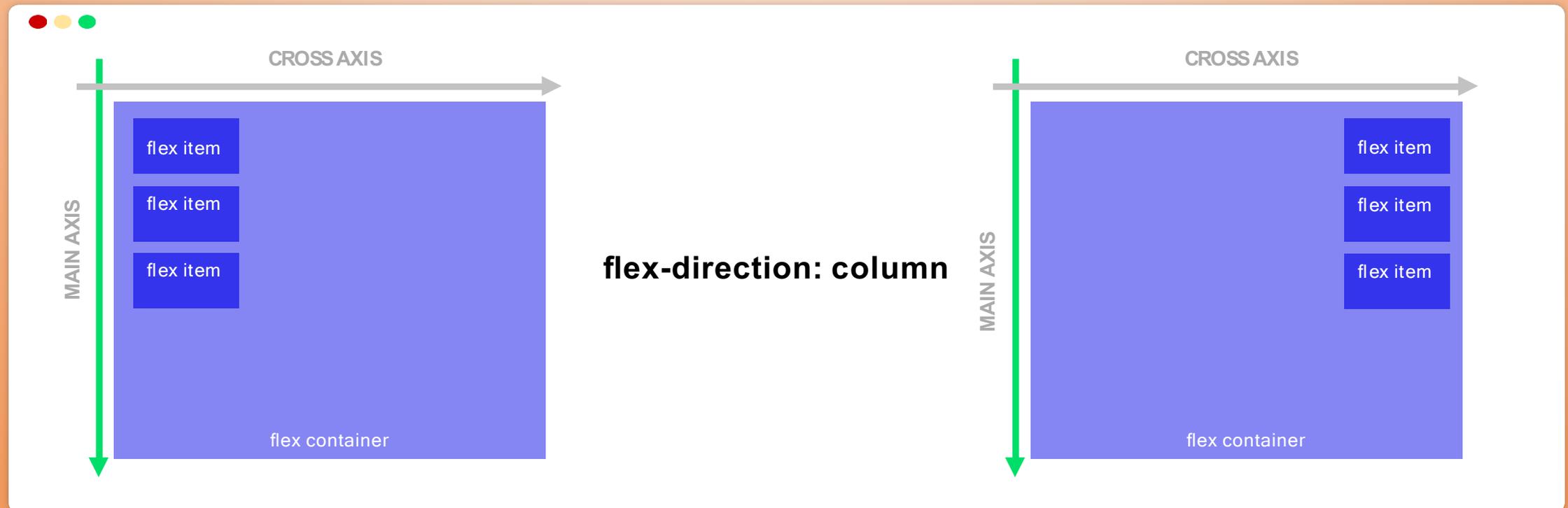


Se cambiamo la direction tutto si sposta!

align-items - Flex-container rule

```
align-items: flex-start;
```

```
align-items: flex-end;
```



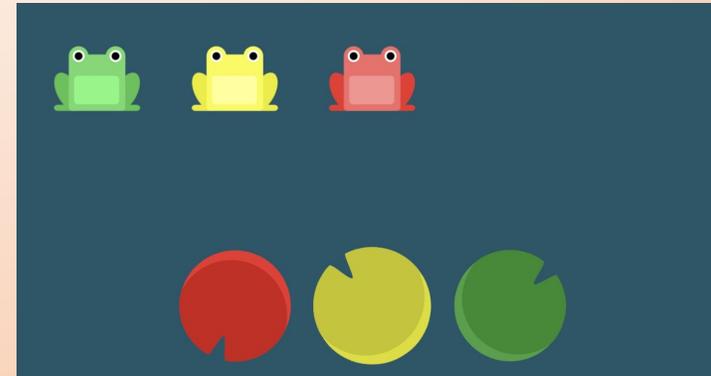
Giochiamo con flexbox?

Prova questi mini-giochi!

Flexfroggy

Aiuta Froggy e i suoi amici scrivendo codice CSS!

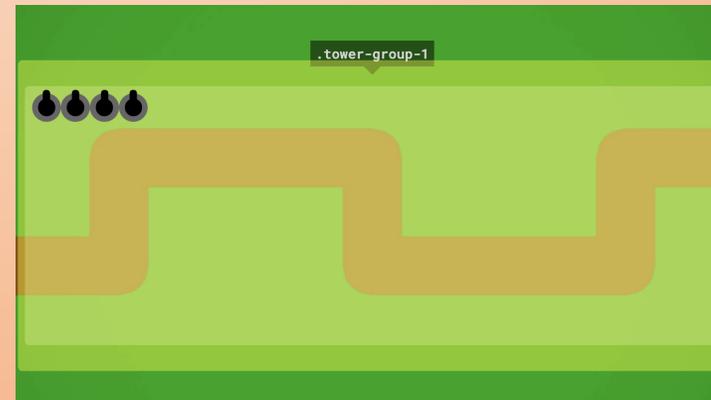
<https://flexboxfroggy.com/#it>



Flexbox Defence

Tower defence game basato su flexbox.

<http://www.flexboxdefense.com/>



Alcune risorse utili

Personalizza l'interfaccia

Crea un tema personalizzato e cambia sfondo, font, colori o immagini.

Unsplash

DB di immagini gratuite ad alta risoluzione.

<https://unsplash.com/it>

Coolors

Palette generator per la scelta dei colori.

<https://coolors.co/>

Google Fonts

Web Fonts supportati dai browser

<https://fonts.google.com/>

SVG Repo

Raccolta di icone gratuite anche in svg.

<https://www.svgrepo.com/>