

RÉSURGENCES INSERTION



ACCATONE



# HTML/CSS

## CSS – Flexbox

# Rappels sur le cours précédent

## Les dimensions ( en px ou % )

width: largeur

height: hauteur

min-height : hauteur minimale

min-width: largeur minimale

max-height: hauteur maximale

max-width: largeur maximale

## Les marges:

Deux types de marges : interne et externe

Deux propriétés: padding et margin

**margin** : marge externe

**padding** : marge internet

Possibilité de mettre des marges uniquement à gauche, droite, haut ou bas

**prefixes** : **-left**, **-right**, **-top**, **-bottom** ( comme border )

# Sommaire du cours

- ❖ HTML : les flex-box
- ❖ Un peu de pratique!

The slide features decorative geometric shapes in the corners. In the top-left corner, there is a dark red polygon with a yellow rectangular section. In the bottom-right corner, there is a yellow pentagon and a dark red polygon.

# Les flex-box

# Les conteneurs

## Les conteneurs (Container)

*Un conteneur : balise de type block qui contient d'autres balises*

*Exemple : une boîte qui contient d'autres boîtes, qui elles mêmes contiennent d'autres boîtes.*

```
<section class="container">  
  <div class="element-1"></div>  
  <div class="element-2"></div>  
  <div class="element-3"></div>  
</section>
```

**Element 1**

**Element 2**

**Element 3**

# Les conteneurs

## Propriété display : flex;

### Propriété CSS

**display : flex;**

*Définition : Conteneur flexible !*

```
<section class="container">
  <div class="element-1"></div>
  <div class="element-2"></div>
  <div class="element-3"></div>
</section>
```

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

Element 1

Element 2

Element 3

# Les conteneurs flex-direction

## Propriété CSS

### flex-direction

*W3C : spécifie la direction des éléments flexibles*

Possibilités de valeurs: **column**, **row**,  
**column- reverse**, **row-reverse**

```
<section class="container">
  <div class="element-1"></div>
  <div class="element-2"></div>
  <div class="element-3"></div>
</section>
```

```
.container {
  display: flex;
  flex-direction: column;
}
```



# Les conteneurs flex-direction

Propriété CSS

**flex-direction**

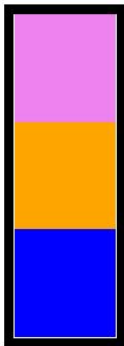
*W3C : spécifie la direction des éléments flexibles*

Possibilités de valeurs: **column**, **row**,  
**column-reverse**, **row-reverse**

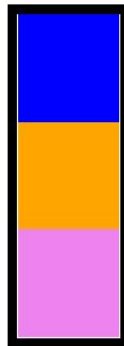
**row ( par défaut )**



**column**



**Column-reverse**



**row-reverse**





# Les conteneurs flex-direction

## Propriété CSS

### flex-direction

*W3C : spécifie la direction des éléments flexibles*

*Axe principal et axe secondaire*

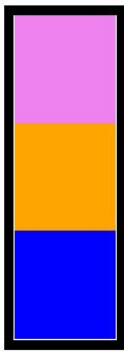
#### row ( par défaut )

Axe principal : **horizontal**  
Axe secondaire : **vertical**



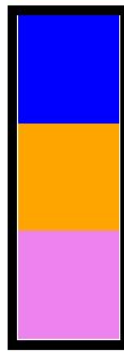
#### column

Axe principal : **vertical**  
Axe secondaire : **horizontal**



#### Column-reverse

Axe principal : **horizontal**  
Axe secondaire : **vertical**



#### row-reverse

Axe principal : **vertical**  
Axe secondaire : **horizontal**



# Alignement sur l'axe principal

## Propriété CSS

### **justify-content**

*Aligne les éléments sur l'axe principal*

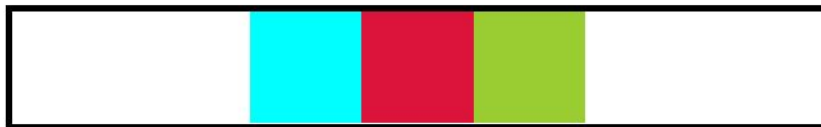
*Ici : flex-direction: row*



**flex-start**



**flex-end**



**center**



**space-between**



**space-around**

**Axe secondaire**

**Axe principal**

# Alignement sur l'axe principal

## Propriété CSS

### justify-content

Aligne les éléments sur l'axe principal

Ici : *flex-direction: column*

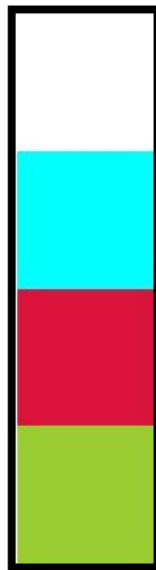
```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

Axe principal

Axe secondaire



flex-start



flex-end



center



space-between



space-around

# Alignement sur l'axe secondaire

## Propriété CSS

### align-items

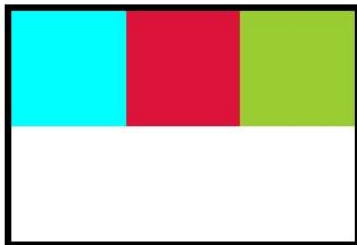
*Aligne les éléments sur l'axe secondaire*

*Ici : flex-direction: row*

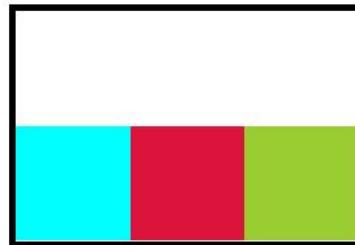
```
.container {  
  display: flex;  
  flex-direction: row; align-items:  
  flex-end;  
}
```

Axe secondaire

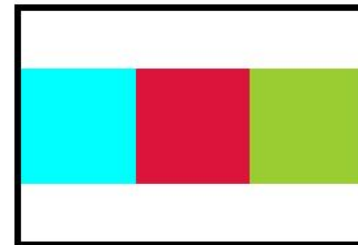
Axe principal



flex-start



flex-end



center

# Alignement sur l'axe secondaire

## Propriété CSS

### align-items

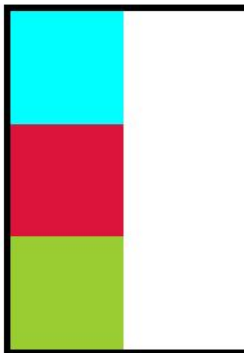
*Aligne les éléments sur l'axe secondaire*

*Ici : flex-direction: column*

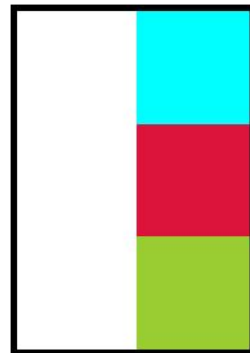
```
.container { display: flex;
flex-direction: column;
align-items: flex-end;
}
```

Axe principal

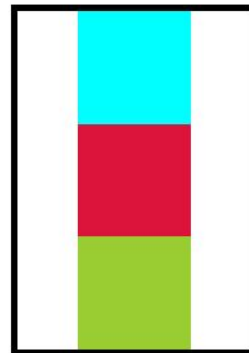
Axe secondaire



flex-start



flex-end



center

# Alignement sur l'axe secondaire

## Propriété CSS

### align-self

Aligne un élément sur l'axe secondaire

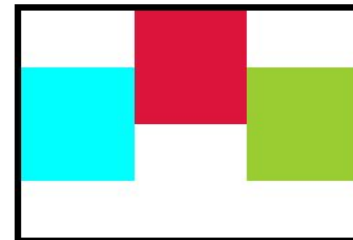
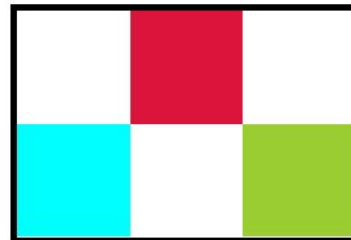
Ici : *flex-direction: row*

```
.container {  
  display: flex;  
  flex-direction: row; align-items:  
  flex-end;  
}
```

```
.element.rouge {  
  align-self: flex-start;  
}
```

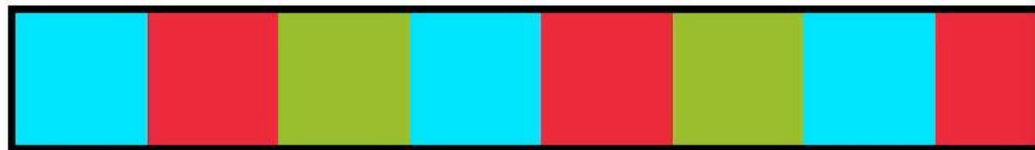
Axe secondaire

Axe principal



# Les conteneurs

Aller à la ligne



## Propriété CSS

### flex-wrap

*Autorise les éléments à aller automatiquement à la ligne*

```
1
2
3 .flex-container {
4     display: flex;
5     flex-direction: row;
6     border: 3px solid black;
7     margin: 10px;
8
9
10 }
11
12 .bleu {
13     background-color: cyan;
14 }
15
```

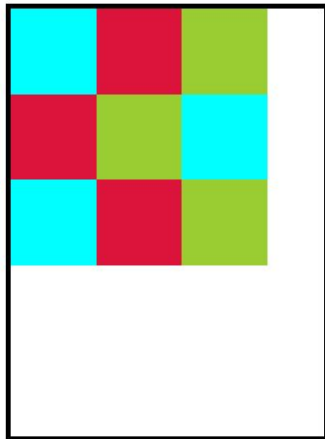
# Les conteneurs

## Aligner les lignes

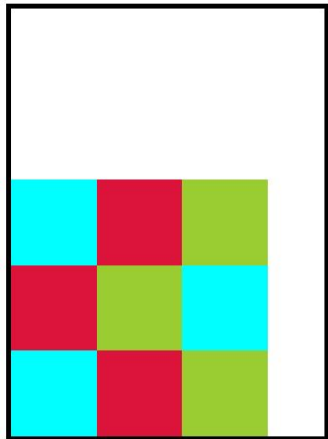
### Propriété CSS

### align-content

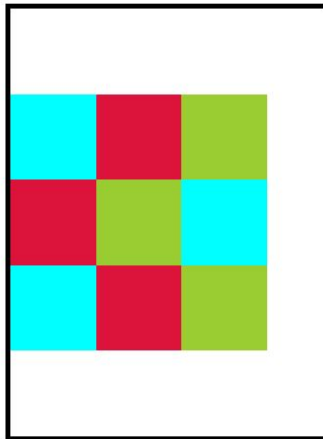
*W3C: modifie le comportement de la propriété flex-wrap*



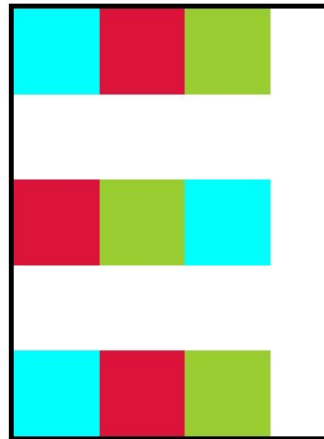
flex-start



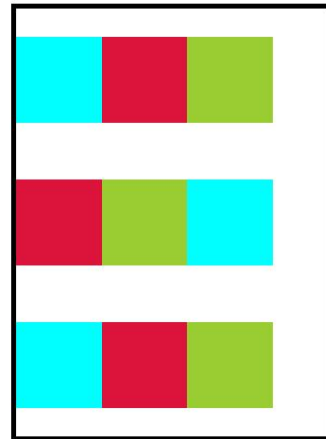
flex-end



center



space-between



space-around



# Les conteneurs

## Aligner les lignes

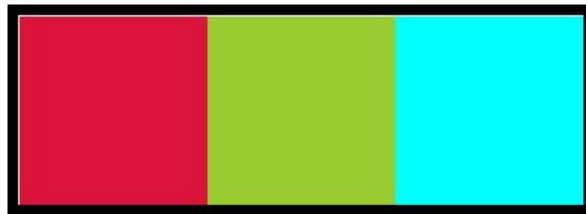
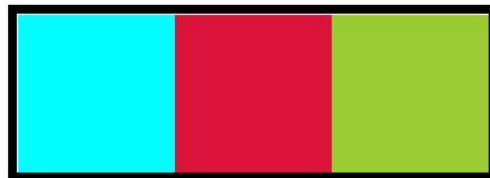
```
.element.bleu {  
    order: 3;  
}  
  
.element.rouge {  
    order: 1;  
}  
  
.element.jaune { order:  
    2;  
}
```

### Propriété CSS

#### order

*W3C: changer l'ordre des éléments*

Ordre par défaut



The slide features a light gray background. In the top-left corner, there is a dark red L-shaped polygon with a small yellow square attached to its top-left side. In the bottom-right corner, there is a yellow pentagon partially overlapping a dark red polygon.

Des questions?

## Exercise

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

<https://flukeout.github.io/>