

# Introduction au Web et Interface Homme/Machine

## Cascading StyleSheets (CSS) – positionnement

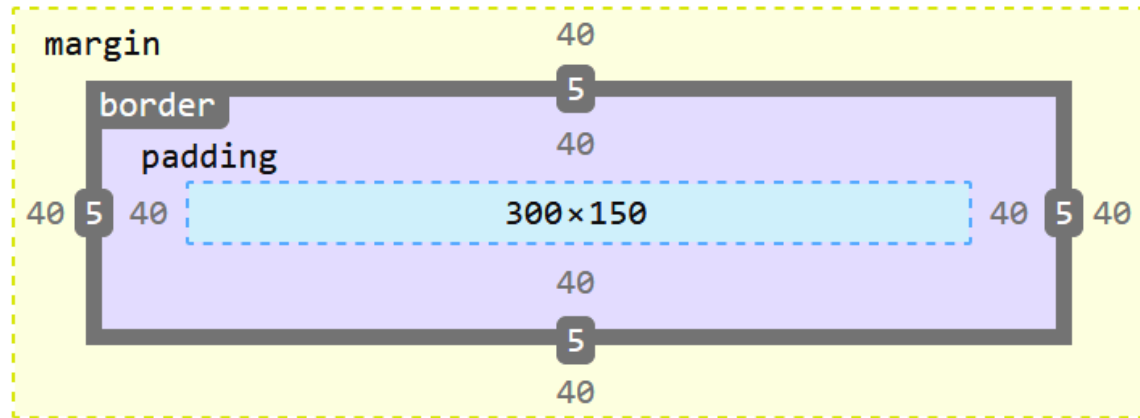
Semestre 2 – Romuald THION

# CSS : le positionnement

Comment le contenu HTML est-il positionné dans la page ?

# Le modèle de boîtes

## ▼ Box Model



390×240

static

## ▼ Box Model Properties

`box-sizing``content-box``line-height``28.8px``display``block``position``static``float``none``z-index``auto`

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/The\\_box\\_model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model)

# Margin versus padding

## Jeux vidéos

MOBA
DOTA 2
FPS
Borderlands 2, Team Fortress, Borderlands the pre-sequel
Strategy
Civilization V, Cities Skyline
RPG
Skyrim, Fallout 4, The Witcher 3

## Jeux vidéos

MOBA
DOTA 2
FPS
Borderlands 2, Team Fortress, Borderlands the pre-sequel
Strategy
Civilization V, Cities Skyline
RPG
Skyrim, Fallout 4, The Witcher 3

## Informatique

- |                                    |
|------------------------------------|
| • C, C++, Python                   |
| • HTML5, CSS3, JavaScript, Node.js |
| • SQL, pgPLSQL                     |

## Informatique

- |                                    |
|------------------------------------|
| • C, C++, Python                   |
| • HTML5, CSS3, JavaScript, Node.js |
| • SQL, pgPLSQL                     |

Qu'est ce qui a changé entre ces deux versions ?

# Margin versus padding

## Jeux vidéos

MOBA
DOTA 2
FPS
Borderlands 2, Team Fortress, Borderlands the pre-sequel
Strategy
Civilization V, Cities Skyline
RPG
Skyrim, Fallout 4, The Witcher 3

## Informatique

- C, C++, Python
- HTML5, CSS3, JavaScript, Node.js
- SQL, pgPLSQL

Comment « désindenter » les `<li>` ?

# Annuler ou homogénéiser la CSS

## Jeux vidéos

MOBA

DOTA 2

FPS

Borderlands 2, Team Fortress, Borderlands the pre-sequel

Strategy

Civilization V, Cities Skyline

RPG

Skyrim, Fallout 4, The Witcher 3

## Informatique

- C, C++, Python
- HTML5, CSS3, JavaScript, Node.js
- SQL, pgPLSQL

<https://necolas.github.io/normalize.css/>

Assure une CSS **homogène** sur les différents navigateurs (MDN *tolère* la pratique)

## Jeux vidéos

MOBA

DOTA 2

FPS

Borderlands 2, Team Fortress, Borderlands the pre-sequel

Strategy

Civilization V, Cities Skyline

RPG

Skyrim, Fallout 4, The Witcher 3

Informatique

C, C++, Python

HTML5, CSS3, JavaScript, Node.js

SQL, pgPLSQL

<https://meyerweb.com/eric/tools/css/reset/>

Annule **toute mise en forme** par défaut des navigateurs (MDN *ne recommande pas*)

# Les éléments blocks et inline

- Élément *inline* : s'étend *horizontalement*
  - pas de retour à la ligne
  - ne prend que la place horizontale **nécessaire**
  - `<span>` (la balise neutre), `<a>`, `<img>`, etc.
  - Voir sur MDN
- Élément *block* : s'étend *verticalement*
  - retour à la ligne
  - prend **autant** de place horizontale que **possible**
  - `<div>` (la balise neutre), `<section>`, `<p>`, etc.
  - Voir sur MDN

# Le *flow* normal

## Basic document flow

I am a basic block level element. My adjacent block level elements sit on new lines below me.

By default we span 100% of the width of our parent element, and we are as tall as our child content. Our total width and height is our content + padding + border width/height.

We are separated by our margins. Because of margin collapsing, we are separated by the width of one of our margins, not both.

inline elements like this one and this one sit on the same line along with adjacent text nodes, if there is space on the same line. Overflowing inline elements will wrap onto a new line if possible (like this one containing text), or just go on to a new line if not, much like this image will do:



Source [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Normal\\_Flow](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Normal_Flow)



# La propriété CSS *display*

- `display`: valeur spécifie le flux :
  - **inline** et **block** : les deux modes principaux
  - **inline-block** : inline mais avec un enfants block
  - **table** : pour se comporter comme un `<table>` (et idem pour `table-cell`, `table-row` etc.)
  - **flex** : block conteneur flexible
  - **grid** : grille 2D contrôlée en CSS
  - **none** : n'affiche pas, ne réserve pas de place

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Introduction](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction)  
<https://css-tricks.com/almanac/properties/d/display/>

# Les éléments sémantiquement neutres

- Sémantiquement neutre :
  - `<div>` : la balise *block* neutre
  - `<span>` : la balise *inline* neutre
- N'ont pas de *sens* particulier **mais** font partie de l'arbre HTML

Dénuées de sens, ces balises **permettent de contrôler la mise en forme**, par exemple :

- pour grouper des éléments (`<div>`)
- altérer des éléments textuels (`<span>`).

When people ask me: « So how does it feel to be a web developer? »



<https://twitter.com/darkgaro/status/1433893426139959315>

# When people ask me: « So how does it feel to be a web developer? »

The image shows a browser window with a tweet from Faruk Brbovic (@darkgaro) that reads: "When people ask me: 'So how does it feel to be a web developer?'". Below the tweet is a blue banner with the text "WEB DEVELOPER". The browser's developer tools are open, displaying the HTML source code. The selected element is an image with the following attributes: ``. The rest of the HTML code is visible in the background, showing a complex structure of nested `<div>` tags.

# La propriété CSS *position*

- `position`: `val` spécifie le positionnement
  - **static** : la valeur par défaut
  - **relative** : par rapport à la position *normale* : celle que le navigateur utilisera si on ne précise rien
  - **fixed** : fixe par rapport à la fenêtre du navigateur
  - **absolute** : relatif au dernier ancêtre positionné
  - **sticky** : est relatif *puis* fixé

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Positioning](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Positioning)  
<https://css-tricks.com/almanac/properties/p/position/>

# Exemple : le positionnement de nav

```

<body>
  <header>
    <nav>
      <ul class="navigation">
        <li><a href="#jeux">Jeux</a></li>
        <li><a href="#info">Info</a></li>
        <li><a href="#other">Other</a></li>
        <li><a href="#misc">Misc</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <h1>Un exemple de menu</h1>
  </main>

```

- [Jeux](#)
- [Info](#)
- [Other](#)
- [Misc](#)

## Un exemple de menu

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>

*Comment faire un menu horizontal ?*



## Un exemple de menu

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>

# Le modèle Flexbox

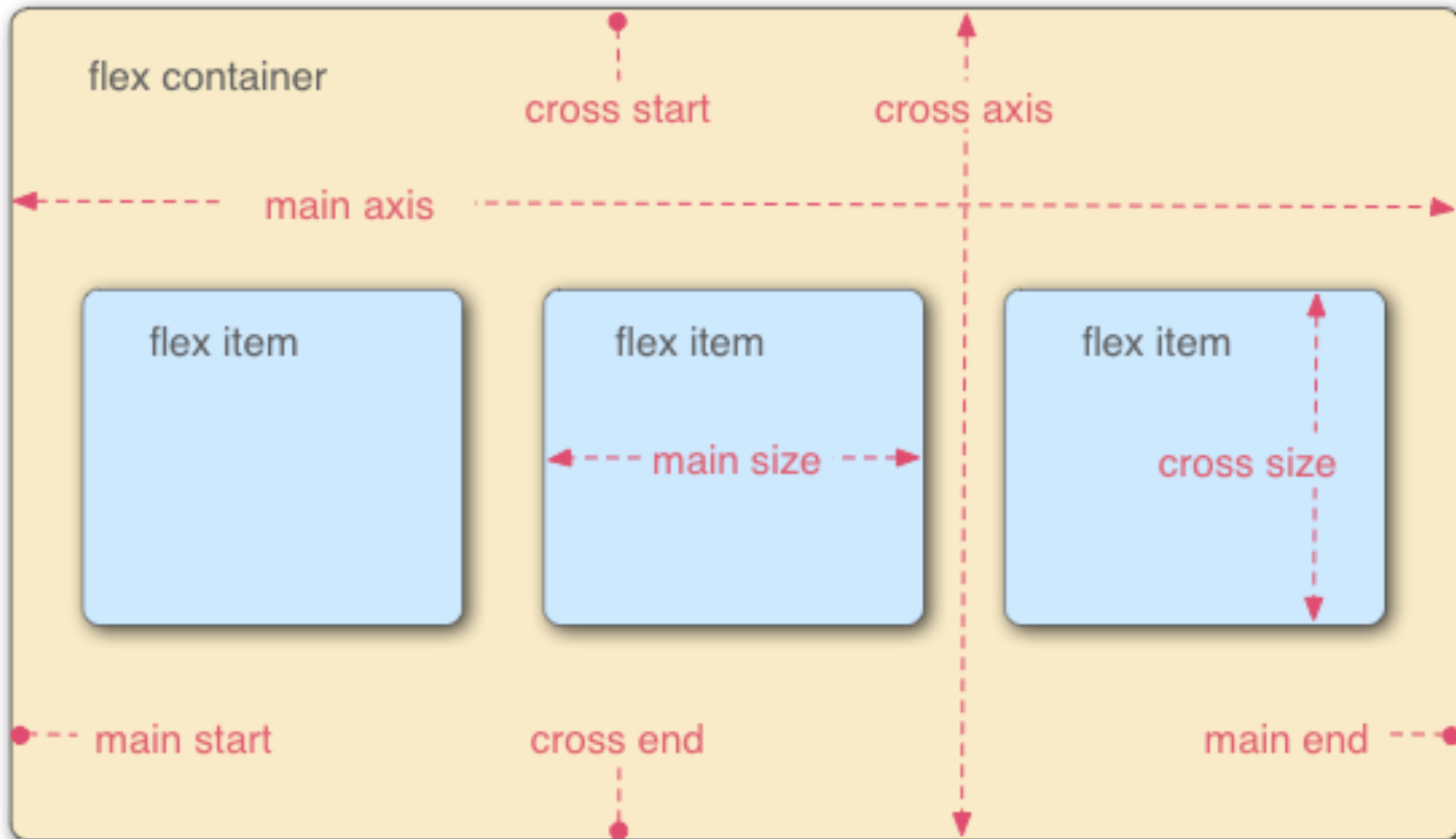
*For a long time, the only reliable cross-browser compatible tools available for creating CSS layouts were features like floats and positioning. These work, but in some ways they're also limiting and frustrating.*

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)

- Les chose (très) pénibles voire *impossibles* (!)
  - Centrer verticalement
  - Avoir des enfants de même largeur
  - Avoir des colonnes de même hauteur

Cas d'utilisation (MDN)

# Le modèle Flexbox

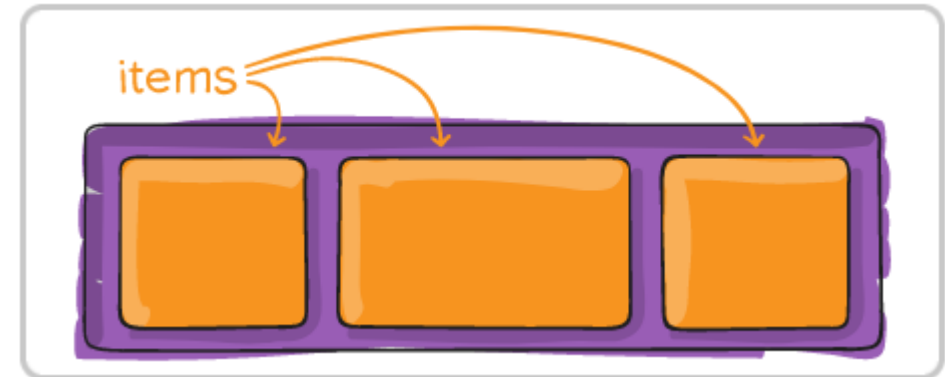
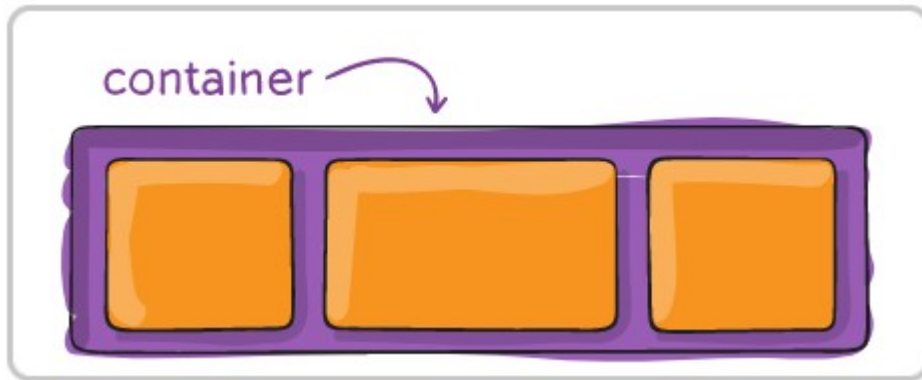


On donne `display: flex` **au conteneur** pour que ses enfants deviennent des *flex items* dont le conteneur contrôlera les positions et les dimensions.



# Le modèle Flexbox

On donne `display: flex` **au conteneur** pour que ses enfants deviennent des **enfants** appelés *flex items* dont le conteneur contrôlera les positions et les dimensions.

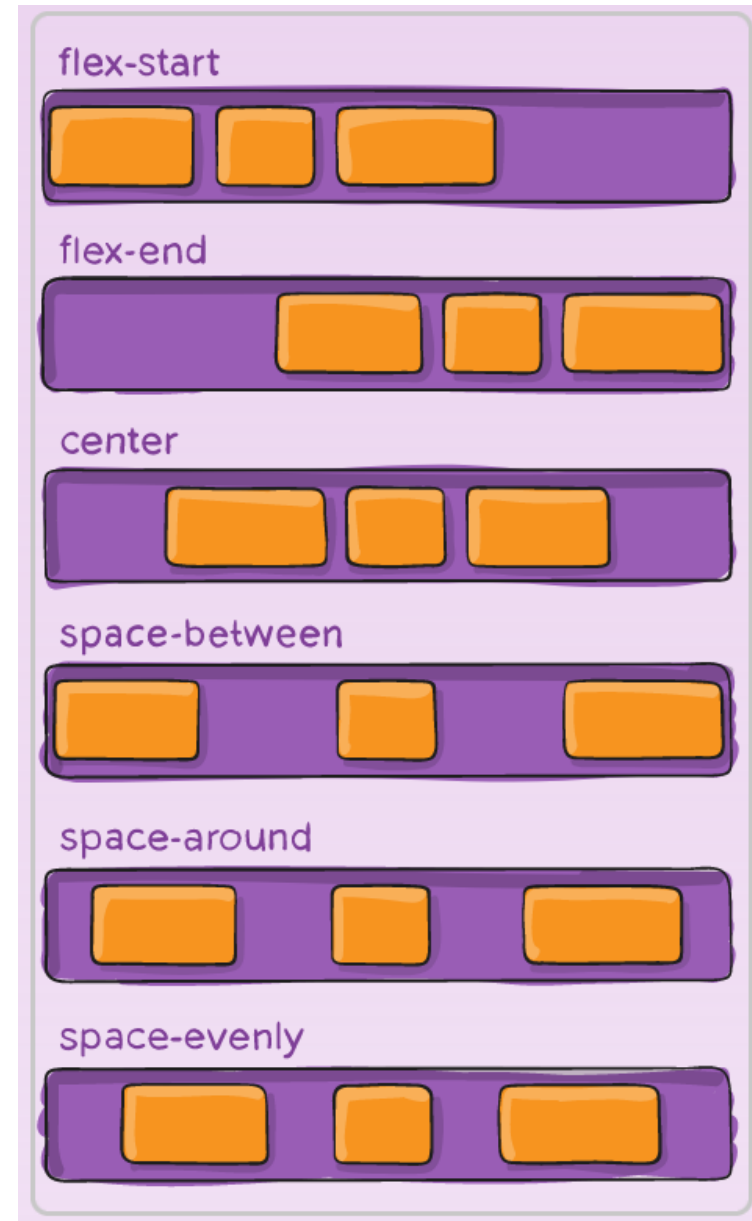


<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://bennettfeely.com/flexplorer/> (excellent)

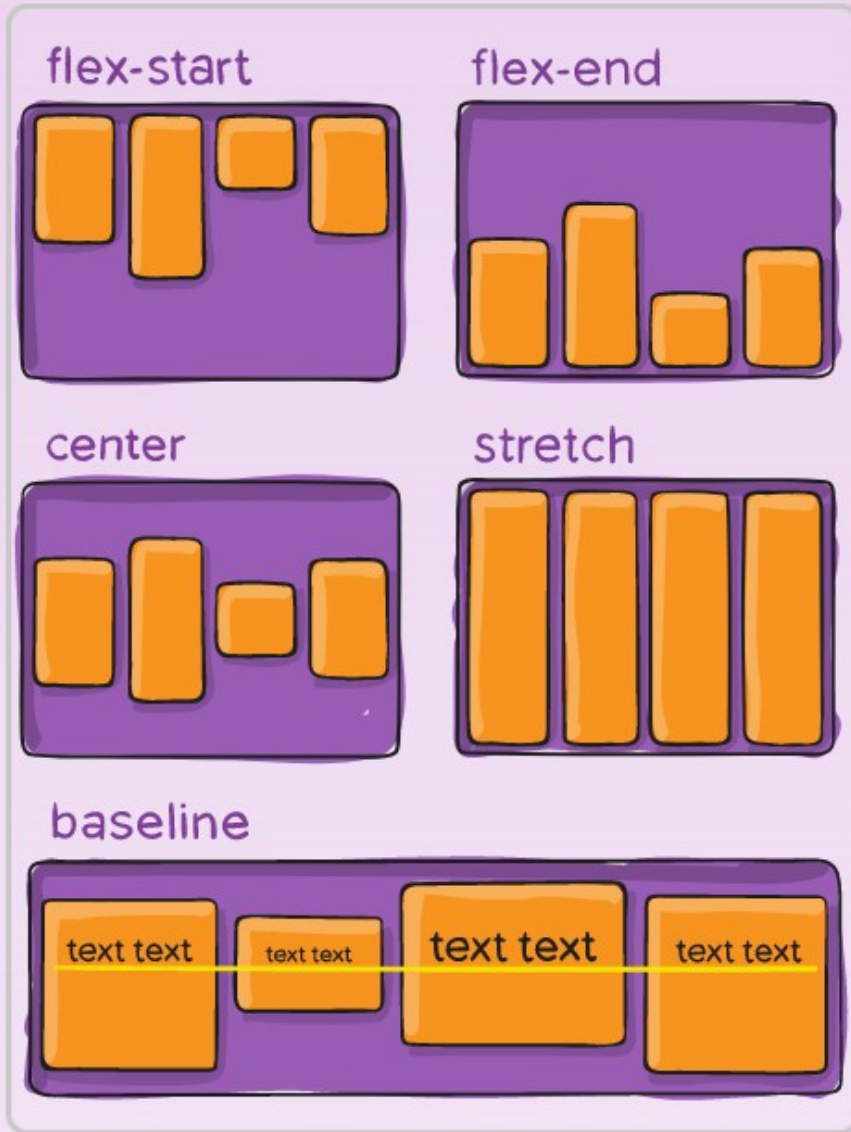
# Les propriétés Flexbox des conteneurs

- `flex-direction`: la direction de l'axe primaire flex (e.g., horizontal)
- `flex-wrap` : le retour à la ligne lors des débordements sur l'axe primaire
- `justify-content`: la position des items (entres eux) sur l'axe primaire
- `align-items`: la position des items dans la direction de l'axe secondaire
- `align-content`: les positions des axes secondaires (entre eux)

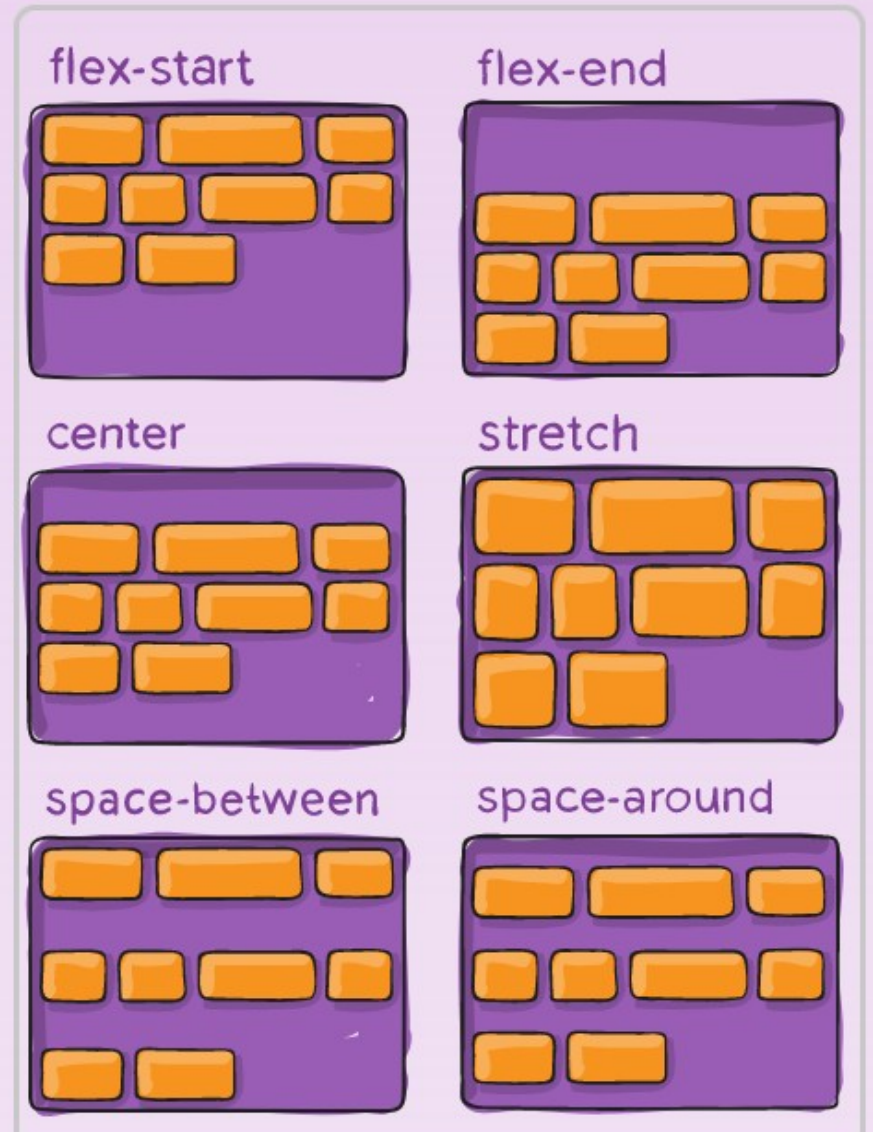


# Les propriétés Flexbox des conteneurs

## align-items

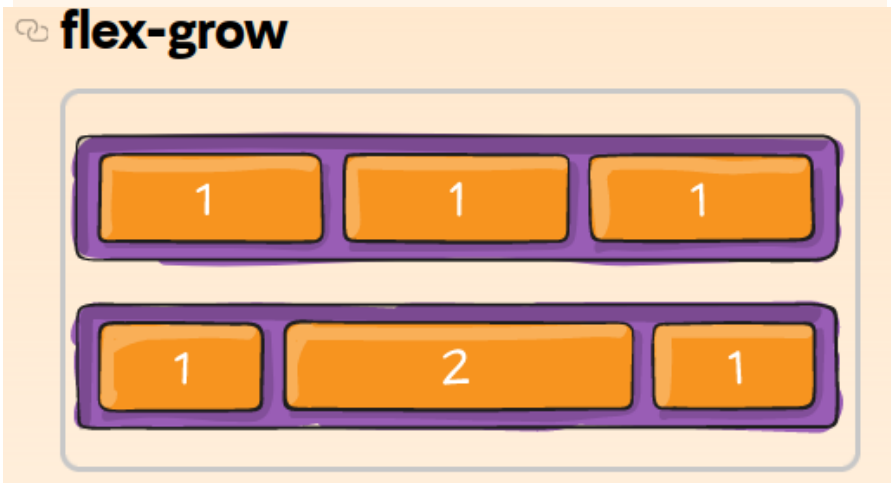
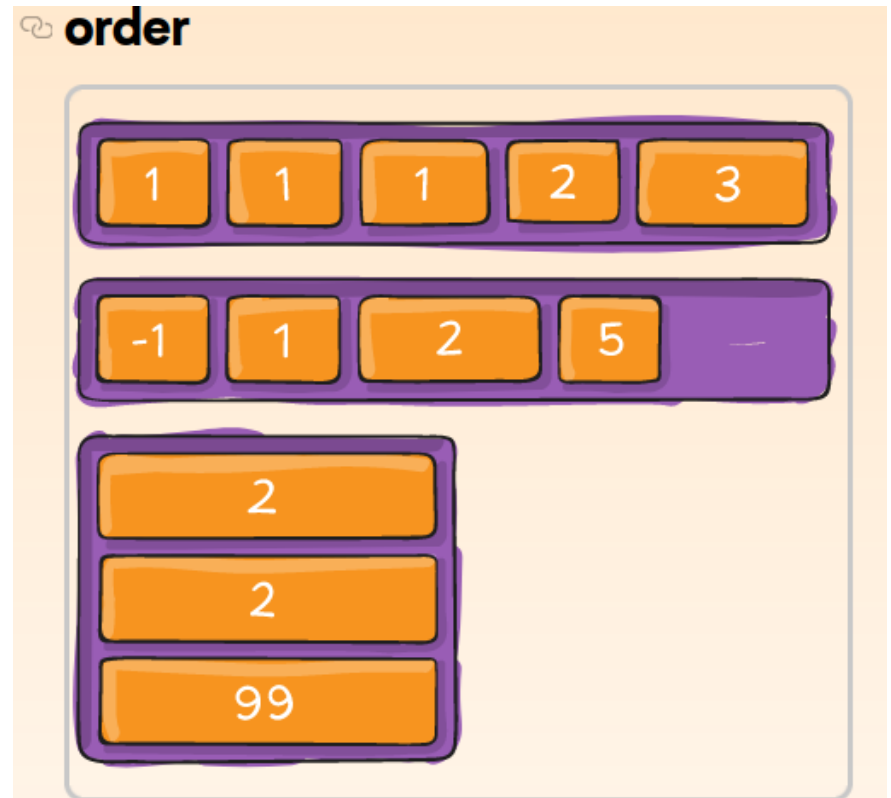


## align-content



# Les propriétés Flexbox des items

- `order`: permet de contrôler la position d'un fils
- `flex-grow` : autorise l'élargissement de l'item
- `flex-shrink` : autorise le rétrécissement de l'item
- `flex-basis` : la taille de départ de l'item
- `align-self` : force la position de l'item en écrasant `align-items`

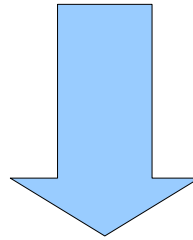


# Démonstration sur l'élément <nav>

- [Jeux](#)
- [Info](#)
- [Other](#)
- [Misc](#)

## Un exemple de menu

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>



## Un exemple de menu

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>

# S'adapter au terminal : les *media queries*

@media	screen	(min-width: 320px)	and	(max-width: 768px)
AT-RULE	MEDIA TYPE	MEDIA FEATURE	OPERATOR	MEDIA FEATURE

<https://css-tricks.com/a-complete-guide-to-css-media-queries/>  
Using media queries (MDN)

- `width` : the widths of the viewport. A specific number (e.g. 400px) or a range (using `min-width` and `max-width`).
- `height` : the height of the viewport. A a specific number (e.g. 400px) or a range (using `min-height` and `max-height`).
- `aspect-ratio` : the width-to-height aspect ratio of the viewport
- `orientation` : tall (portrait) or wide (landscape) based on how the device is rotated

# S'adapter au terminal : les *media queries*



## Un exemple de menu

Large

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>



## Un exemple de menu

Moyen

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>



## Un exemple de menu

Petit

Exemple pour <https://romulusfr.github.io/unc-s2-web-ihm/>

# Conclusion

- Utilisez les outils du développeur de Firefox ou Chrome
- Faites vous aider par votre IDE
  - Live server, completion, doc, linting etc.
  - Valable pour la programmation en général !
- Utilisez de bonnes documentations :
  - <https://developer.mozilla.org/en-US/docs/Web/CSS>
  - <https://css-tricks.com/>
  - <https://www.pierre-giraud.com/> (en français, didactique)
- ***Use the platform***
  - Les navigateurs proposent toujours plus de *fonctionnalités natives* : **utilisez les** !