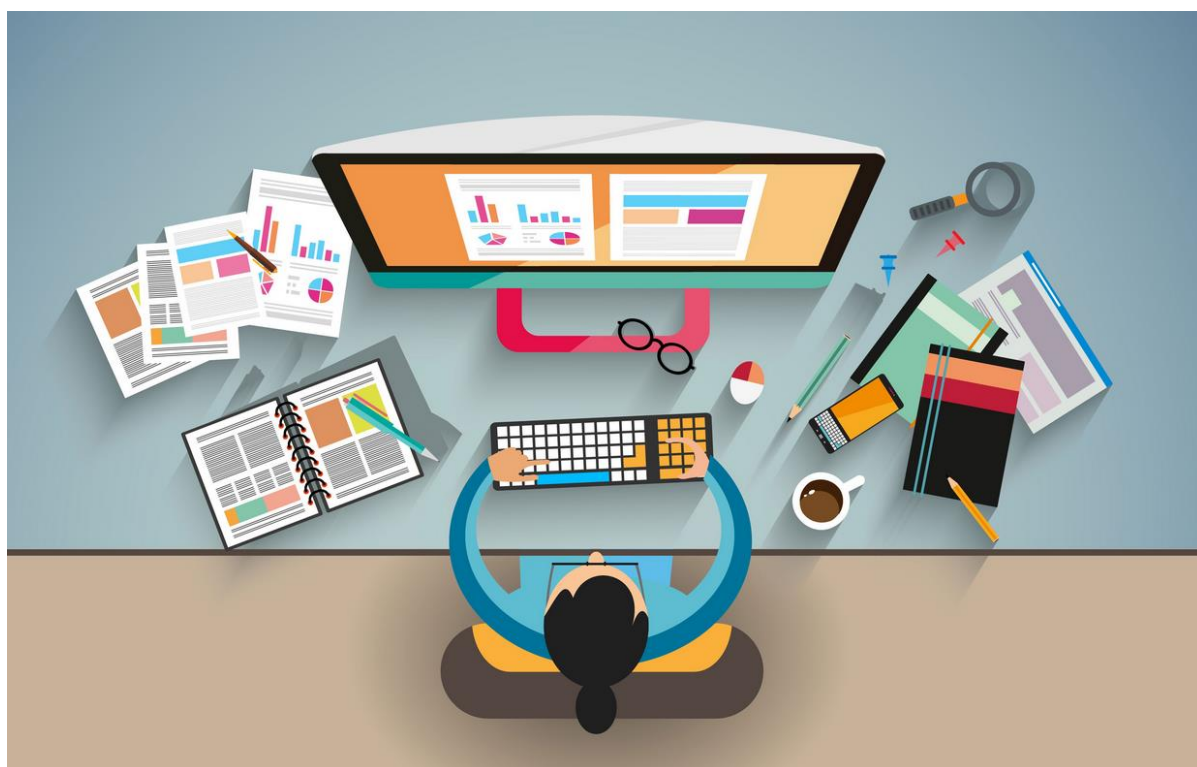


SUPPORT de COURS

# - FLEXBOX | HTML CSS -



Formation organisée par le GRETA d'Aix

Contenus pédagogiques et supports : Emmanuel ADINARAYANIN

# Flexbox

Présentation : Exo 02\_flexbox > 01 fonctionnement

## Le contexte flexbox

### Comportement flex-container au parent et comportement flex-item aux enfants

Le contexte s'applique au parent, les enfants deviennent automatiquement des flex-item

```
/*le bloc prend le contexte flex-container et se comporte comme un block*/  
.premier{ Display :flex }  
  
/*le bloc prend le contexte flex-container et se comporte comme un inline-block*/  
.deuxieme{Display :inline-flex }
```

## Possibilités du flex

- Distribution des éléments horizontale ou verticale
- Alignement et centrage verticaux et horizontaux, justifiés ou répartis
- Réorganisation des éléments indépendamment de l'ordre du flux
- Gestion de l'espace disponible (fluidité)

## Propriétés sur le parent

Exo 02\_flexbox > 02\_flex-base

### display (default :flex)

- **display :flex** (contexte **flex** pour les enfants mais fonctionne comme un **block** sur son environnement)
- **display :inline-flex** (contexte **flex** pour les enfants mais fonctionne comme un **inline-block** sur son environnement)

### flex-direction (default : row)

- row
- row-reverse
- column
- column-reverse

### flex-wrap (default : nowrap)

- nowrap
- wrap
- wrap-reverse

## Flex-flow (propriété raccourcie)

```
.container{  
    display :flex ;  
    flex-flow :nowrap column ;  
}
```

## Alignement (sur le parent)

### Fonctionnement

2 axes sont considérés

- primaire (defini par flex direction row ou column) propriété : **justify-content**
- secondaire (l'autre) propriété : **align-items**

#### justify-content (default : flex-start)

- flex-start
- flex-end
- center
- space-between
- space-around

#### align-items (default : stretch)

- stretch
- flex-start
- flex-end
- center

#### align-content (uniquement pour les container flex en wrap)

- flex-start
- flex-end
- center

## Alignement (sur les enfants)

Exo 02\_flexbox > 02\_flex-items

### Order (default : 0)

Change l'ordre des flex item.

Valeurs négative en premier, valeurs positives en dernier ;

Pour une même valeur les flex-item suivent l'ordre du flux.

### Align-self (default : auto / comme défini dans align-items)

Il est question ici de l'alignement d'un flex-item indépendamment de ses frères

Mêmes propriétés que align-items (axe secondaire)

### Flex (s'adapter à la taille du conteneur)

#### Flex-grow (default 0 / non autorisé à s'agrandir)

Force un flex-item à occuper l'espace disponible ; la valeur 0 signifie qu'on ne s'occupe pas de l'espace restant. Si tous les flex item ont le même chiffre 1, ils se répartissent l'espace restant uniformément.

Si l'un des flex-item a une valeur plus élevée, il prendra plus de place sur l'espace restant.

#### Flex-shrink (default : 1 / autorisé naturellement à se réduire)

Autorise un flex-item à se réduire, ou pas.

#### Flex-basis (default : auto / prend la taille de son contenu / unités %, px, etc)

- auto : taille du contenu
- valeur : la taille demandée au flex-item (propriété non prioritaire sur min-width, max-width)

## Ecriture courte

```
/* grow | shrink | flex-basis */  
.enfant{ Flex :1 1 3em ;}
```

### auto

L'élément est dimensionné selon ses propriétés `width` et `height` mais peut grandir pour absorber l'espace libre disponible dans le conteneur flexible ou rétrécir à sa taille minimale pour rentrer dans le conteneur. Cette valeur est équivalente à `"flex: 1 1 auto"`.

### initial

La valeur par défaut. L'élément est dimensionné selon ses propriétés `width` et `height`. Si besoin, l'élément rétrécit à sa taille minimale pour rentrer dans le conteneur mais il ne grandira pas s'il y a de l'espace disponible dans ce conteneur. Ce mot-clé est équivalent à `"flex: 0 1 auto"`.

### none

L'élément est dimensionné par rapport à ses propriétés `width` et `height`. Il n'est pas flexible : il ne peut ni rétrécir ni grandir selon l'espace du conteneur flexible. Ce mot-clé est équivalent à `"flex: 0 0 auto"`.

**Finir l'exercice 01\_mediaqueries > 02\_superhéros en mode Flex**

## Méthodes flexbox

### Occuper l'espace variation sur flex-grow, shrink et basis

Exo 02\_flexbox > 03\_interface-flex

### Centrer verticalement

Exo 02\_flexbox > 03\_centrer verticalement

En contexte float, les marges et le centrage des éléments fonctionnent aussi bien dans les 2 axes.

### Sur l'enfant :

```
.flex-item{margin :auto}
```

### Sur le parent :

```
.parent{align-items :center}
```

### Pousser un élément

Exo 02\_flexbox > 03\_pousser en bas

### Sur l'enfant :

```
.flex-item{margin-top :auto}
```

