

DÉVELOPPEMENT D'APPLICATIONS WEB

COURS 4: MISE EN PAGE CSS AVEC FLEXBOX

Enseignante: NGUYEN Thi Minh Tuyen

Plan du cours

1. Terminologie Flexbox
2. Conteneurs Flexbox
3. Flow: direction du flux et retour à la ligne
4. Alignement sur les axes principal et transversal
5. Spécifier les éléments d'une flexbox "flex"
6. Changer l'ordre des éléments flexibles

Flexbox

- Flexbox est un mode d'affichage qui positionne les éléments le long d'un axe (horizontal ou vertical).
- Utile pour les options de menu, les galeries, les listes de produits, etc.
- Les éléments d'une boîte flexible peuvent étendre, se réduire et / ou se dérouler sur plusieurs lignes, ce qui en fait un excellent outil pour les présentations dynamiques.
- Les éléments peuvent être réorganisés, ils ne sont donc pas liés à l'ordre source.
- Flexbox peut être utilisé pour des composants individuels sur une page ou pour la mise en page complète.

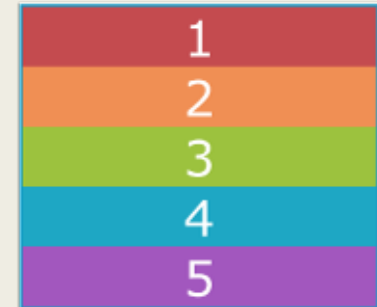
Conteneur Flexbox [1]

`display: flex`

- Pour activer le mode Flexbox, réglez l’affichage de l’élément sur flex.
- Cela fait de l’élément un conteneur flexbox.
- Tous ses enfants directs deviennent des éléments flexibles dans ce conteneur.
- Par défaut, les éléments s'alignent dans le sens d'écriture du document (lignes de gauche à droite dans les langues de lecture de gauche à droite).

Conteneur Flexbox [2]

By default, the `div`s display as block elements, stacking up vertically. Turning on flexbox mode makes them line up in a row.



block layout mode

`display: flex;`



flexbox layout mode



Lignes et colonnes (Direction)

flex-direction

- Valeurs: row, column, row-reverse, column-reverse
- La valeur par défaut: row,
- Vous pouvez modifier le sens pour que les éléments soient affichés en colonnes ou dans l'ordre inverse:

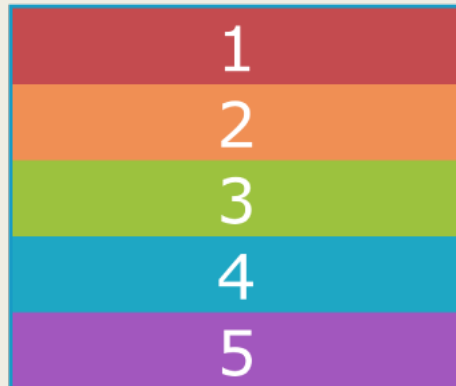
flex-direction: row; (default)



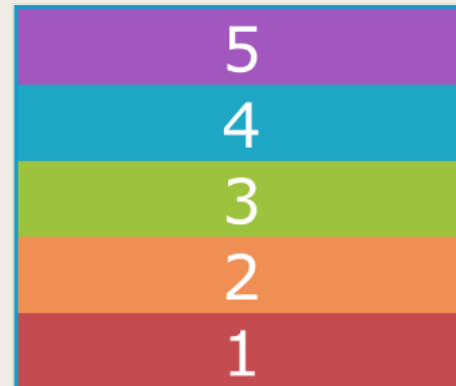
flex-direction: row-reverse;



flex-direction: column;



flex-direction: column-reverse;



Couverture des lignes flexibles

flex-wrap

- Valeurs: wrap, nowrap, wrap-reverse
- Les éléments flexibles s'alignent sur un axe, mais vous pouvez autoriser cet axe à se placer sur plusieurs lignes avec la propriété flex-wrap:

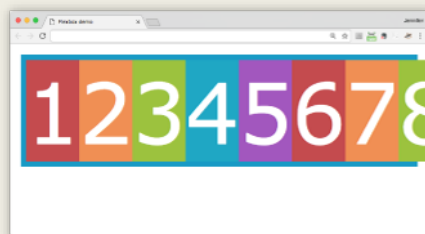
flex-wrap: wrap;



flex-wrap: wrap-reverse;



flex-wrap: nowrap; (default)



When wrapping is disabled, flex items squish if there is not enough room, and if they can't squish any further, may get cut off if there is not enough room in the viewport.

Flux flexible (Direction +couverture)

flex-flow

- Valeurs: flex-direction flex-flow
- C'est la propriété raccourcie pour les propriétés flex-direction et flex-wrap.
- Exemple:

```
#container {  
  display: flex;  
  height: 350px;  
  flex-flow: column wrap;  
}
```

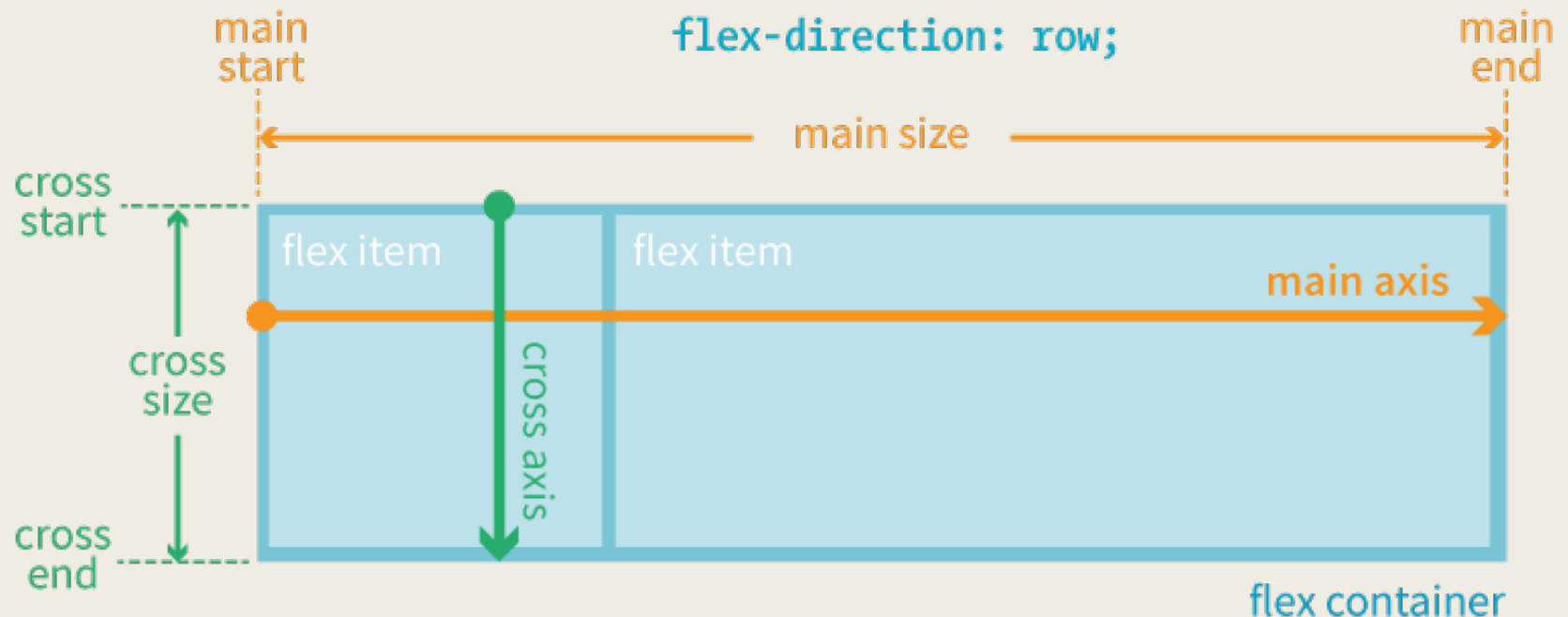

Terminologie d'alignement de Flexbox

- Flexbox est «direction-agnostique» nous parlons donc d'axe principal et d'axe transversal au lieu de lignes et de colonnes.
- **L'axe principal** s'étend dans la direction dans laquelle le flux a été défini.
- **L'axe transversal** est perpendiculaire à l'axe principal.
- Les deux axes ont un début, une fin et une taille.

row: Axes principaux et transversals

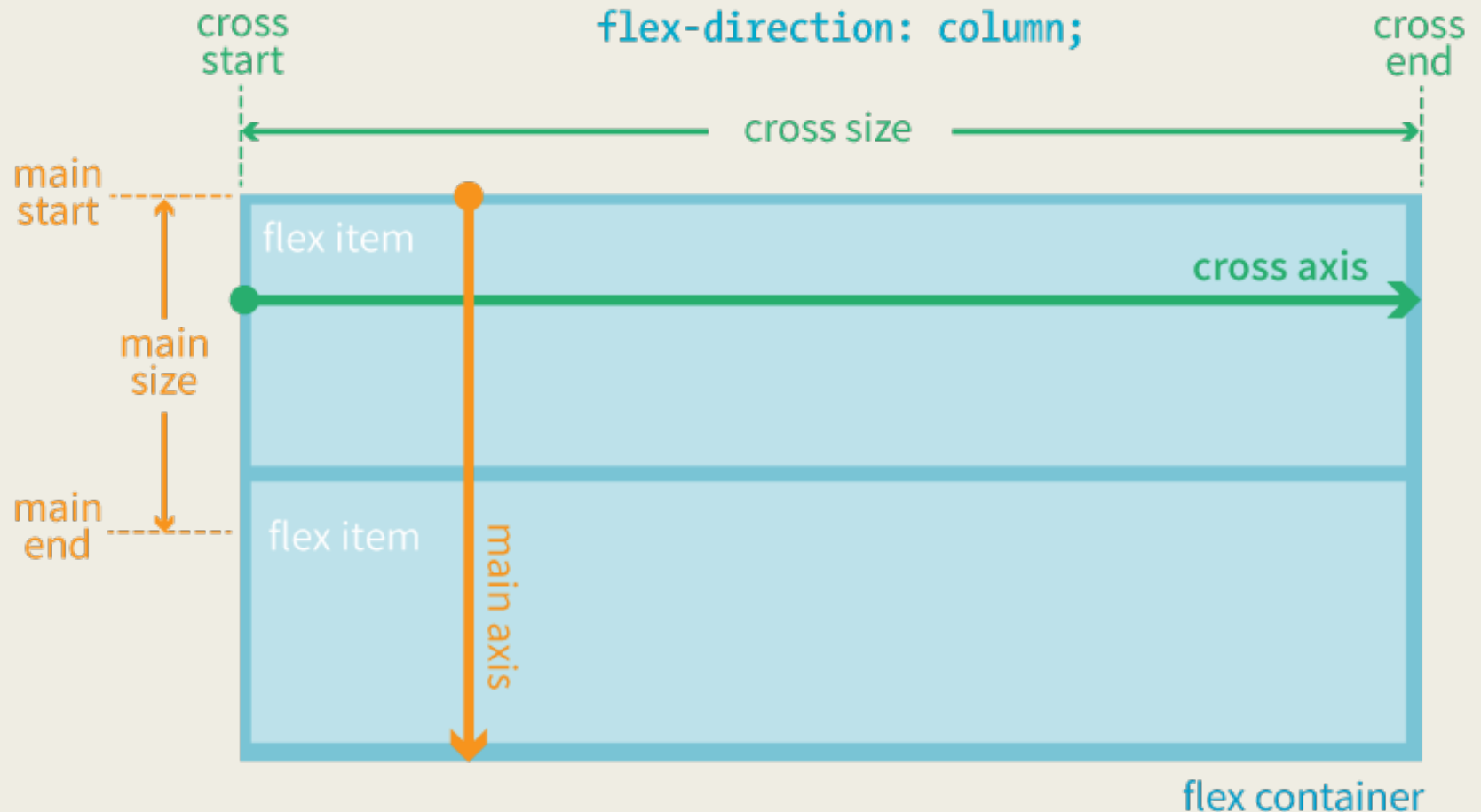
FOR LANGUAGES THAT READ HORIZONTALLY FROM LEFT TO RIGHT:

When `flex-direction` is set to `row`, the main axis is horizontal and the cross axis is vertical.



column: Axes principaux et transversals

When `flex-direction` is set to `column`, the main axis is vertical and the cross axis is horizontal.



Alignement sur l'axe principal [1]

`justify-content`

- Valeurs: `flex-start`, `flex-end`, `center`, `space-between`, `space-around`
- Spécifier l'alignement des éléments (notez que nous disons début et fin au lieu de gauche / droite ou haut / bas).
- S'appliquer au conteneur flex.

```
#container {  
  display: flex;  
  justify-content: flex-start;  
}
```

Alignement sur l'axe principal [2]

- Lorsque la direction est row : l'axe principal est horizontal

`justify-content: flex-start;` (default)



`justify-content: flex-end;`



`justify-content: center;`



`justify-content: space-between;`



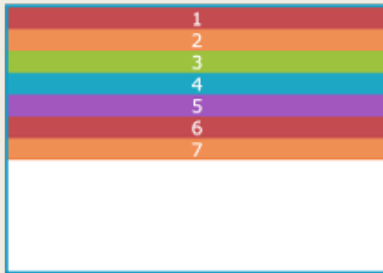
`justify-content: space-around;`



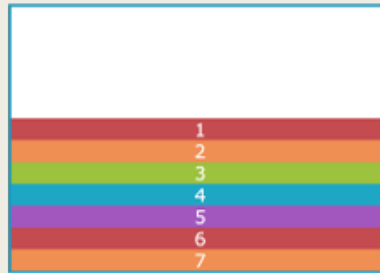
Alignement sur l'axe principal [3]

- Lorsque la direction est `column`: l'axe principal est vertical

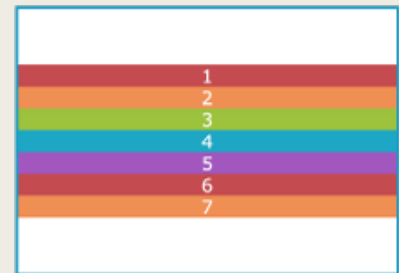
`justify-content: flex-start;` (default)



`justify-content: flex-end;`



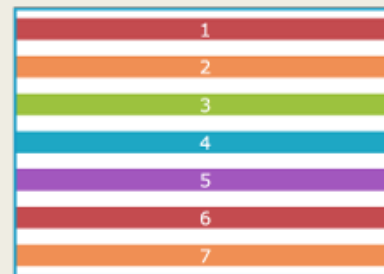
`justify-content: center;`



`justify-content: space-between;`



`justify-content: space-around;`



Alignement sur l'axe transversal [1]

`align-items`

- Valeurs: `flex-start`, `flex-end`, `center`, `baseline`, `stretch`
- Spécifie l'alignement des éléments.
- S'applique au conteneur flex.
- Exemple:

```
align-items#container {  
  display: flex;  
  flex-direction: row;  
  height: 200px;  
  align-items: flex-start;  
}
```

Alignement sur l'axe transversal

[2]

- Lorsque la direction est row, l'axe principal est horizontal et l'axe transversal est vertical.

`align-items: flex-start;`



`align-items: flex-end;`



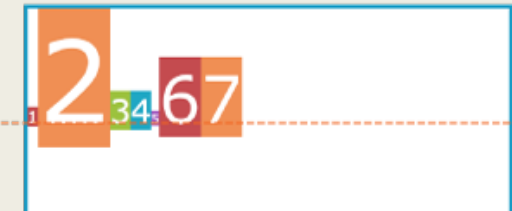
`align-items: center;`



`align-items: stretch;` (default)



`align-items: baseline;`



Items are aligned so that the baselines of the first text lines align.

Alignement sur l'axe transversal

[3]

`align-self`

- Valeurs: `flex-start`, `flex-end`, `center`, `baseline`, `stretch`
- Aligne un élément individuel sur l'axe transversal.
- Utile si un ou plusieurs éléments doivent remplacer le paramètre `align-items` pour le conteneur.
- S'applique à l'élément flex.
- Exemple:

```
.box4 {  
  align-self: flex-end;  
}
```



Alignement sur l'axe transversal [4]

align-content

- Valeurs: flex-start, flex-end, center, space-around, space-between, stretch
- Aligne les lignes de contenu.
- S'applique au conteneur flex.

align-content: flex-start;

1	2	3	4
5	6	7	8
9	10		

align-content: flex-end;

1	2	3	4
5	6	7	8
9	10		

align-content: center;

1	2	3	4
5	6	7	8
9	10		

align-content: space-between;

1	2	3	4
5	6	7	8
9	10		

align-content: space-around;

1	2	3	4
5	6	7	8
9	10		

align-content: stretch; (default)

1	2	3	4
5	6	7	8
9	10		

Aligner sur les marges

- Utilise une marge (définie sur `auto`) pour mettre un espace supplémentaire sur le côté d'éléments flexibles particuliers.
- Exemple: Ajouter une marge automatique à la droite du premier élément flexible (le `li` avec le logo) pousse le `li` restant vers la droite:

```
ul {  
  display: flex;  
  align-items: center;  
  ...  
}  
li.logo {  
  margin-right: auto;  
}
```

flex

flex

- Valeurs: none, 'flex-grow flex-shrink flex-basis'
- Les éléments peuvent redimensionner (flex) pour remplir l'espace disponible sur l'axe principal du conteneur.
- La propriété flex identifie combien un article peut grossir et se rétrécir et identifie une taille de départ
- Il distribue de l'espace supplémentaire dans le conteneur à l'intérieur des éléments (par rapport à un contenu qui justifie la répartition de l'espace entre et autour des éléments).

Exemple: flex

- flex est un raccourci pour des propriétés **flex-grow**, **flex-shrink**, et **flex-basis**.
- Les valeurs 1 et 0 fonctionnent comme des interrupteurs marche / arrêt.

```
li {  
    flex: 1 0 200px;  
}
```

Les éléments de liste du conteneur flex commencent à une largeur de 200 pixels, sont autorisés à se développer plus largement (**flex-grow: 1**) et ne sont pas autorisés à se réduire (**flex-shrink: 0**).

Expansion d'un élément flexible [1]

flex-grow

- Valeurs: nombre
- Spécifie si et dans quelle proportion un élément peut s'étirer plus grand. 1 permet l'expansion; 0 l'empêche.
- flex-grow est appliqué à l'élément flex item.

flex: 0 1 auto; (prevents expansion)



flex: 1 1 auto; (allows expansion)



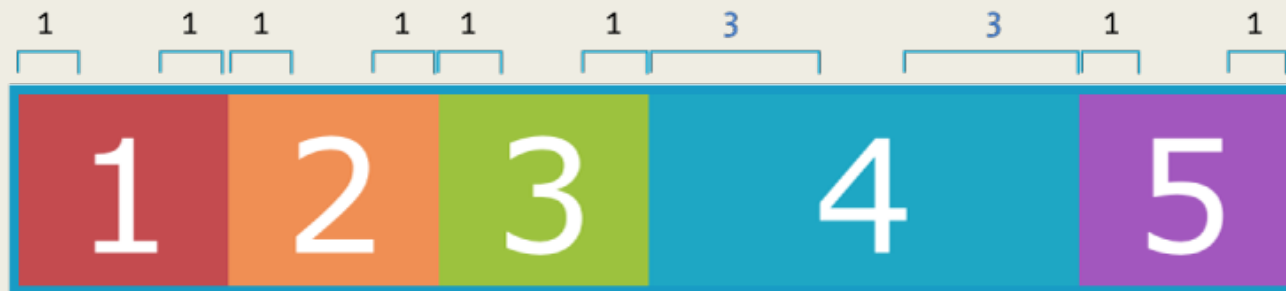
Expansion d'un élément flexible [2]

Flex relatif

- Lorsque **flex-basis** a une valeur autre que 0, les valeurs entières plus élevées agissent comme un ratio qui applique plus d'espace dans cet élément.
- Exemple: une valeur de 3 attribue trois fois plus d'espace à box4 qu'à des éléments dont la valeur de `flex-grow` est de 1.

(Notez que ce n'est pas nécessairement 3 fois plus large que les autres éléments.)

```
.box4 { flex: 3 1 auto; }
```



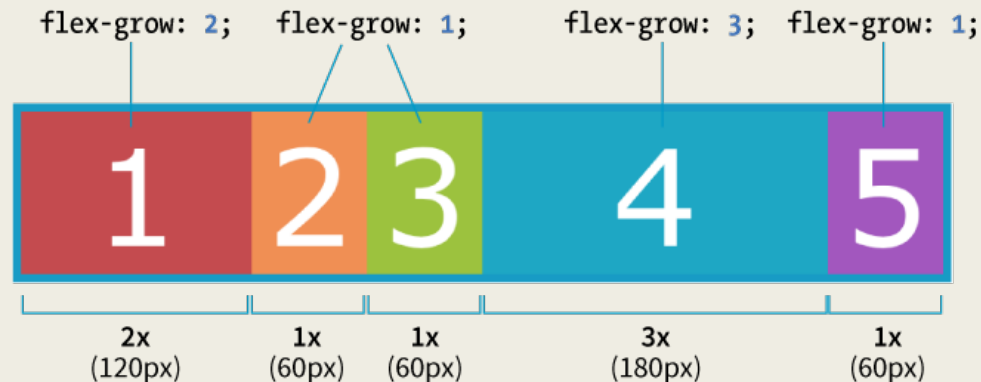
`flex: 3 1 auto;`

Expansion d'un élément flexible [3]

Flex absolut

- Lorsque `flex-basis` est 0, la taille des éléments est proportionnelle au rapport de flexion.
- Exemple: une valeur de 3 rend «box4» 3 fois plus large que les autres lorsque `flex-basis: 0`.

```
.box4 { flex: 3 1 0%; }
```



Valeurs raccourcis flex

- **flex: initial** (identique à `flex: 0 1 auto;`) Empêche la croissance de l'élément, mais lui permet de se contracter pour s'adapter au conteneur
- **flex: auto** (identique à `flex: 1 1 auto;`) Permet aux éléments d'être totalement flexibles au besoin. La taille est basée sur les propriétés `width / height`.
- **flex: none** (identique à `flex: 0 0 auto;`) Crée un élément totalement inflexible tout en le dimensionnant aux propriétés `width / height`.
- **flex: integer** (identique à `flex: integer 1 0px;`) Crée un élément flexible avec un flex absolu (les valeurs de `flex-grow` sont donc appliquées proportionnellement)

Changer l'ordre de l'élément [1]

order

- Valeurs: nombre
- Spécifie l'ordre dans lequel un élément particulier doit apparaître dans le flux (indépendamment de l'ordre des sources HTML):
- `order` est appliqué à d'élément flex.
- La valeur par défaut est 0. Les éléments avec la même valeur d'ordre sont placés en fonction de leur ordre dans la source.
- Les éléments avec des valeurs d'ordre différentes sont classés du plus bas au plus élevé.
- La valeur numérique spécifique n'a pas d'importance; seul le lien avec d'autres valeurs (comme z-index) est important.

Changer l'ordre de l'élément [2]

```
.box3 {  
  order: 1;  
}
```



order: 0
(default)

order: 0
(default)

order: 0
(default)

order: 0
(default)

order: 1

Changer l'ordre de l'élément [3]

- Groupes ordinaux
- Les articles qui partagent la même valeur d'ordre sont appelés un groupe ordinal.
- Les groupes ordinaux s'unissent et sont classés du plus bas au plus élevé:

```
.box2, .box3 {  
  order: 1;  
}
```



Navigateurs supportant Flexbox

- La spécification Flexbox a changé au fil des ans et a été implémentée par les navigateurs:

- Version actuelle (2012): `display: flex;`

Supporté par toutes les versions actuelles de bureau et de navigateur mobile

- Version «Weener» (2011): `display: flexbox;`

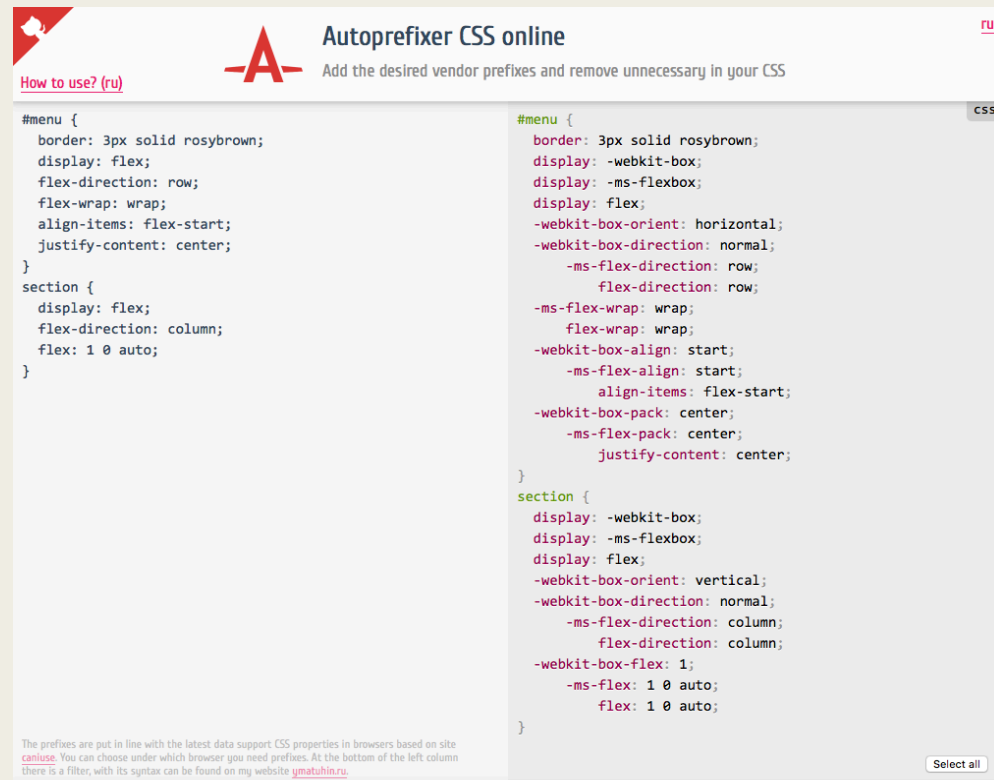
Supporté par IE10 uniquement

- Ancienne version (2009): `display: box;`

Supporté par Chrome <21, Safari 3.1–6, Firefox 2–21; iOS 3.2 - 6.1, Android 2.1 - 4.3

Navigateurs supportant Flexbox

- Pour vous assurer que Flexbox fonctionne avec tous les navigateurs pris en charge, vous avez besoin de nombreux préfixes de fournisseur et déclarations redondantes.
- Utilisez un outil tel que Autoprefixer pour générer tout ce code pour vous (autoprefixer.github.io).



Revue de propriété Flexbox

Propriétés du conteneur Flex

display

flex-flow

flex-direction

flex-wrap

justify-content

align-items

align-content

Propriétés de l'élément Flex

align-self

flex

flex-grow

flex-shrink

flex-basis

order

Question ?