

TD4

Positionnement 2

Flexbox et Grid-Layout

Table des matières

Flexbox.....	2
Propriétés sur le parent.....	3
display.....	3
flex-direction.....	4
justify-content.....	4
flex-wrap.....	4
align-items.....	4
align-content.....	5
Propriétés sur les enfants.....	5
order.....	5
align-self.....	5
flex.....	6
Application : mise en page avec <i>Flexbox</i>	7
Responsive design.....	7
Grid Layout.....	8
Propriétés sur le parent.....	9
display.....	9
grid-template-rows.....	9
grid-template-column.....	9
grid-template-areas.....	9
grid-auto-rows.....	9
grid-auto-columns.....	9
grid-auto-flow.....	9
row-gap.....	9
column-gap.....	9
justify-content.....	9
justify-items.....	9
align-content.....	9
align-items.....	9
Propriétés sur les enfants.....	9
grid-row.....	9
grid-column.....	9
justify-self.....	9
align-self.....	9
order.....	9
Application : mise en page avec <i>Grid Layout</i>	10

Pré-requis : cours INFO102, TD 1, 2 et 3

A la fin de ce TD vous devrez **être capable de** :

- faire une mise en page simple en utilisant *Flexbox*
- faire une mise en page simple en utilisant *Grid-Layout*

Flexbox

Flexbox peut être considéré comme un modèle de *boite flexible* avec :

- un **flex-container** créant le contexte général d'affichage pour ses enfants ;
- des **flex-item** qui sont les enfants directs du conteneur.

Le **flex-container** peut être n'importe quel élément HTML doté de la déclaration `display: flex` ou `display: inline-flex`

Ses enfants directs deviennent alors automatiquement des éléments de type **flex-item**.

■ Créez une nouvelle page web *TD4.html* et copiez-collez le code suivant :

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>TD4</title>
  <link rel="stylesheet" href="style_TD4.css">
  <script src="script.js"></script>
</head>
<body>

<header>
<h1>TD4<br>positionnement 2<br>Flexbox et Grid Layout</h1>
</header>

<section id="flex">
<h2>Flexbox</h2>
<div>Element 1</div>
<div>Element 2</div>
<div>Element 3</div>
<div>Element 4</div>
<div>Element 5</div>
<div>Element 6</div>
</section>

</body>
</html>
```

■ Créez une nouvelle nouvelle feuille de style *style_TD4.css* et copiez-collez le code suivant :

```
h2 {
  background-color: #203961;
}
```

```
div:nth-child(even) {
    background-color: #e1755e;
}

div:nth-child(odd) {
    background-color: #5d554b;
}

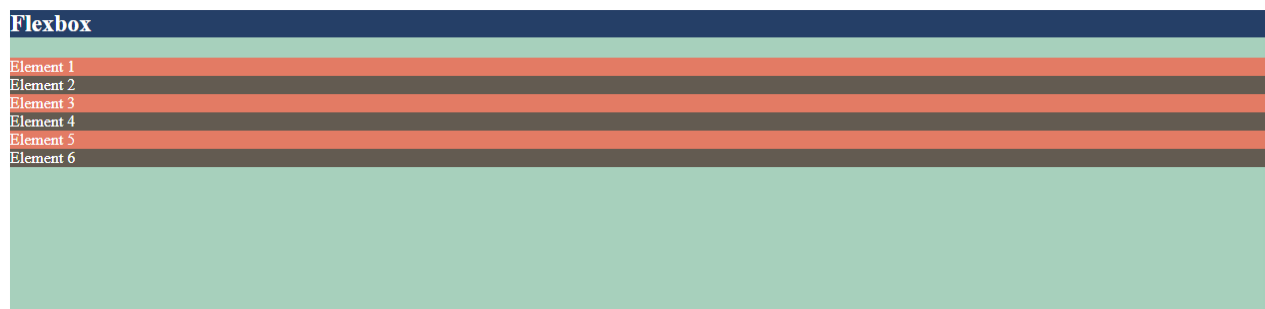
#flex {
    height: 300px;
    background-color: #a2cdb8;
    color: white;
}
```

Sans surprise, les éléments *div* occupent toute la largeur de leur parent et s'affichent les uns en dessous des autres (comportement de *type block*).

TD4

positionnement 2

Flexbox et Grid Layout



Propriétés sur le parent

display

- Affectez la déclaration `display: flex;` à la section *flex*.

Le *h2* et les *div* sont transformés en **flex-items** (enfants) par le contexte d'affichage **flex** de leur **flex-container** (parent).

Par défaut, la largeur des *flex-items* est égale à la largeur de leur contenu.

- Rajoutez du texte à un des *div*.

TD4

positionnement 2

Flexbox et Grid Layout



flex-direction

Par défaut, les *flex-items* sont distribués horizontalement `flex-direction: row;`

- Ajoutez la déclaration `flex-direction: column;` à la section *flex*.
- Essayez les autres valeurs de la propriété *flex-direction*: `column-reverse`, `row`, `row-reverse`
- Revenez à une distribution horizontale.

justify-content

Par défaut, les *flex-items* s'alignent à gauche (distribution horizontale) ou en haut (distribution verticale) `justify-content: flex-start;`

- Essayez les autres valeurs de la propriété *justify-content*: `flex-end`, `center`, `space-between`, `space-around`

flex-wrap

Par défaut, les *flex-items* ne passent pas à la ligne (par exemple si vous réduisez la largeur de la fenêtre du navigateur) `flex-wrap: nowrap;`

- Essayez les autres valeurs de la propriété *flex-wrap*: `wrap` et `wrap-reverse`

Les propriétés *flex-wrap* et *justify-content* concernent l'**axe principale** de distribution défini par *flex-direction*.

Voyons maintenant les propriétés qui gèrent l'**axe secondaire** (vertical si *flex-direction: row;* horizontal si *flex-direction: column;*).

align-items

Par défaut, les *flex-items* (marges externes comprises) s'étendent sur toute la hauteur du *flex-container* `align-items: stretch;`

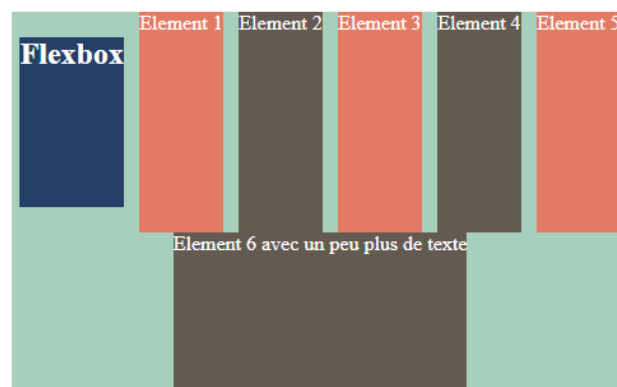
- Essayez les autres valeurs de la propriété *align-items*: `flex-start`, `flex-end`, `center`, `baseline`.

align-content

La propriété *align-content* s'applique uniquement sur un *flex-container* autorisant le passage à la ligne. Elle définit l'alignement des rangées ou des colonnes.

- Autorisez le passage à la ligne des *flex-items* et essayez les valeurs de la propriété *align-content* : *stretch* (valeur par défaut), *flex-start*, *flex-end*, *center*, *space-between*, *space-around*
- Essayez de deviner la valeur des propriétés précédentes qui produisent l'affichage suivant :

TD4 positionnement 2 Flexbox et Grid Layout



Propriétés sur les enfants

order

La propriété *order* permet de modifier l'ordre d'affichage des flex-items.

- Ajoutez la déclaration `order: 1;` à `h2`.

align-self

La propriété *align-self* permet d'appliquer à un *flex-item* un alignement différent de celui de ses frères.

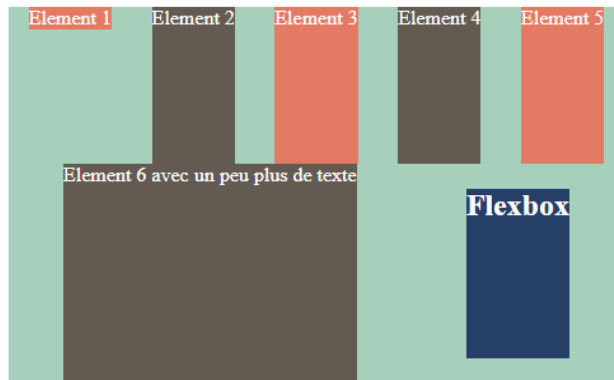
Les valeurs de la propriété *align-self* sont les mêmes que celles de la propriété *align-items*.

- Essayez les différentes valeurs de la propriété *align-self* sur le premier *div*.

TD4

positionnement 2

Flexbox et Grid Layout



flex

La propriété *flex* est un raccourci de 3 propriétés *flex-grow*, *flex-shrink* et *flex-basis*.

Les spécifications du W3C encouragent fortement à utiliser la propriété *flex*.

Par défaut les valeurs de flex sont 0 1 et auto :

- un *flex-item* n'est pas autorisé à s'élargir
- un *flex-item* peut se rétrécir
- la taille d'un *flex-item* dépend de son contenu

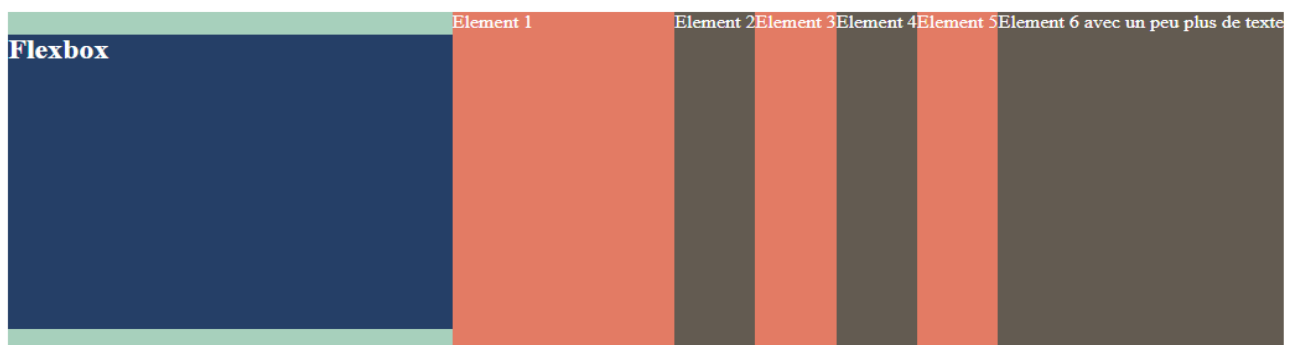
■ Ajoutez la déclaration `flex: 1;` au premier *div*.

■ Ajoutez la déclaration `flex: 2;` à *h2*.

TD4

positionnement 2

Flexbox et Grid Layout



Pour aller plus loin :

https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Flexbox

<https://developer.mozilla.org/fr/docs/Web/CSS/flex>

https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

Application : mise en page avec *Flexbox*

- Créez une feuille de style pour le fichier *maquette_flex.html* pour obtenir la mise en page suivante :



Responsive design

- Rajoutez dans la feuille de style de *maquette_flex.html* :

```
@media (max-width: 800px){
    main{
        flex-direction: column;
    }
}
```

- Commentez...

Grid Layout

En reprenant la même démarche que pour *flexbox* et en vous aidant de la documentation mozilla https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout, essayez de comprendre comment utiliser *Grid Layout*.

Propriétés sur le parent

display

grid-template-rows

grid-template-column

grid-template-areas

grid-auto-rows

grid-auto-columns

grid-auto-flow

row-gap

column-gap

justify-content

justify-items

align-content

align-items

Propriétés sur les enfants

grid-row

grid-column


justify-self

align-self

order

Application : mise en page avec *Grid Layout*

■ Créez une feuille de style pour le fichier *maquette_grid.html* pour obtenir la mise en page suivante :



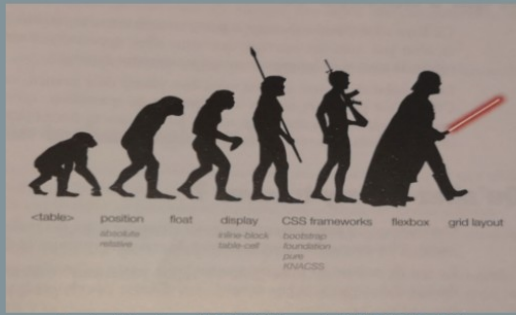
Grid Layout

Propriétés sur le parent

- display
- grid-template-rows
- grid-template-columns
- grid-template-areas
- grid-auto-rows
- grid-auto-columns
- grid-auto-flow
- row-gap
- column-gap
- justify-content
- justify-items
- align-content
- align-items

Propriétés sur les enfants

- grid-row
- grid-column
- justify-self
- align-self
- order



Raphaël Goetter - CSS3 Grid Layout - EYROLLES 2019

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce posuere libero ac ex sodales, quis eleifend enim ultricies. Donec vitae felis eget lectus sagittis tempor et nec sapien. Phasellus in nibh ut tellus cursus condimentum sit amet ut sapien. Mauns sed sem id risus elementum dictum. Aliquam id maximus massa. Nullam a nunc et dui consequat scelerisque. Vestibulum egestas sed nulla sed tempor. Morbi lobortis neque nec mattis semper. Vestibulum luctus facilisis dictum. Fusce at ligula in dolor posuere pulvinar. Sed sed vulputate nisl, eu eleifend nunc. Maecenas ut augue id sem viverra semper. Nam at sem nec felis feugiat egestas ut in purus. Morbi auctor aliquam lacus, ut auctor sem maximus eget. In porta, turpis at rutrum pharetra, ligula turpis placerat felis, in finibus felis lectus eget sem. In ac magna ultrices, eleifend risus vitae, condimentum lacus. Nullam hendrerit, massa finibus volutpat consequat, felis velit fermentum mauris, quis laoreet nibh justo ut tellus. Curabitur lacus dolor, volutpat in ornare ac, efficitur ut lacus. Fusce sit amet rutrum purus, vel molestie dui. Proin vel ligula quis nisl eleifend vulputate sit amet quis lectus. Quisque fermentum metus felis, finibus mollis arcu dictum quis. Aliquam sollicitudin, tellus vel convallis hendrerit, ligula lacus viverra ipsum, at fermentum massa purus eu lacus. Praesent bibendum posuere sapien sed maximus. Praesent commodo mi in ultrices lobortis. Mauris eget sapien leo. Vivamus et lobortis sem. Praesent suscipit libero at pretium condimentum. Sed consequat neque ut mi cursus, eu scelerisque elit consequat. Nunc fringilla, massa at ultrices vehicula, orci lectus cursus nisl, vel gravida velit mauris at sem. Morbi consectetur urna leo, imperdiet sodales diam lobortis a. Ut tristique fringilla risus eget ullamcorper. Nam nisl arcu, rhoncus sed orci eu, malesuada congue dolor. Sed nec lorem justo. Vivamus nisl libero, finibus id volutpat eu, dapibus et lectus. Nullam tempus a mi id tempor. Ut diam nulla, vestibulum ac elit quis, sagittis porta enim. Donec nec ornare sapien, in pharetra libero. Nam sed nisi sed nibh ultricies congue. Ut elit sem, pharetra in erat in, tempus porta eros. Mauris ac congue enim, in pretium metus. Nam commodo ultricies metus eu feugiat. Suspendisse sed mauris diam. Mauris iaculis, turpis in ullamcorper mattis, dui tellus fermentum lorem, eget maximus tortor orci eget mauris. Vivamus vel mauris eu tortor condimentum posuere sed sit amet mauris. Nam facilisis ligula at nisi hendrerit, eu imperdiet tellus luctus. Nullam lectus nunc, porta non urna at, mattis pulvinar felis. Nunc fringilla arcu lacus, lacinia commodo ante fermentum ut. Quisque dignissim, mi eget dapibus lobortis, justo dui mollis dolor, quis feugiat augue libero non justo. Mauris sed ultricies lacus. Vestibulum faucibus dictum felis at fringilla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam facilisis ac lacus ut ultrices. Suspendisse vestibulum maximus semper. Quisque tincidunt enim eget leo pretium, a mattis nibh pharetra. Sed at dapibus eros. Quisque scelerisque augue varius, viverra ex sed, porttitor erat. Ut finibus orci id urna lobortis, id fringilla justo sollicitudin. In eget ex ac enim consectetur aliquet.

Le pied de page