

Cours 1 bis

Compléments CSS

HEI 2021 / 2022

Polices de caractère

Familles de polices

- Il existe 2 grand types de polices : celles avec empattements et celles sans empattements.

M



Empattement

M

« Web-safe » fonts

- Par défaut, les polices utilisables sur un site web sont celles présentes sur l'ordinateur de l'utilisateur.
 - Ces polices disponibles sont différentes suivant le système d'exploitation utilisé.
- Certaines polices sont considérées « web-safe », c'est-à-dire qu'elles sont disponibles sur la majorité des systèmes.

Exemples de polices « web-safe »

- Sans empattement :

Arial

Helvetica

Verdana

Comic Sans MS

Trebuchet

- Avec empattement :

Times New Roman

Georgia

Garamond

Palatino

Déclaration d'une police

- La propriété `font-family` permet de spécifier la police à utiliser pour un élément.
- En valeur, une liste de polices est donnée. Le navigateur va sélectionner la première police présente sur l'ordinateur.
- Cette liste de polices doit se terminer par une police générique :
 - `serif` pour une police avec empattement
 - `sans-serif` pour une police sans empattement
 - `monospace` pour une police à chasse fixe

Polices personnalisées

- Il est possible d'utiliser des polices personnalisées avec l'instruction `@font-face`.
- Le fichier de définition de la police doit être fourni par le site et va être téléchargé sur l'ordinateur de l'utilisateur.
- Les propriétés suivantes permettent de définir la police :
 - `font-family` : nom de la police
 - `src` : fichier(s) décrivant la police
 - `font-style` : caractère italic ou non de la police
 - `font-weight` : caractère gras ou non de la police

Exemple

```
@font-face {  
    font-family: 'OrkneyRegular';  
    src: url('fonts/orkney/Orkney Regular.woff2') format('woff2'),  
         url('fonts/orkney/Orkney Regular.woff') format('woff'),  
         url('fonts/orkney/Orkney Regular.ttf') format('truetype');  
    font-weight: normal;  
    font-style: normal;  
}  
  
body {  
    font-family: 'OrkneyRegular', Verdana, sans-serif;  
}
```


Compatibilité des navigateurs

- @Font-face a une plutôt bonne compatibilité navigateur mais les différents formats de police ne sont pas tous aussi bien supportés.
- <https://caniuse.com/#feat=fontface>
- <https://caniuse.com/#feat=woff2>
- <https://caniuse.com/#feat=woff>
- <https://caniuse.com/#feat=ttf>

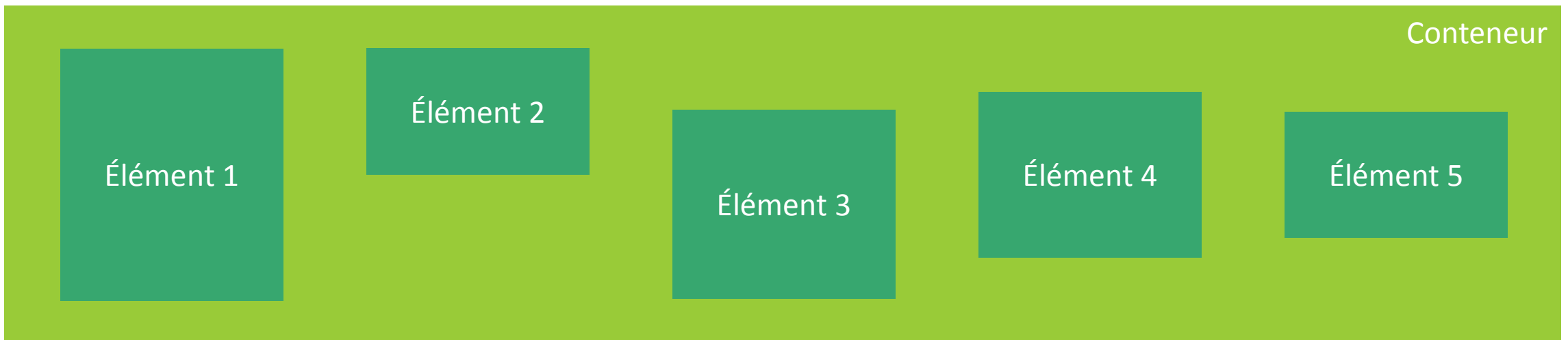
Flex box

Introduction

- Flex box (ou flex) est une autre manière de positionner des éléments dans une page web.
- Elle permet de remplacer localement le flux HTML par un « flux flex » donnant de nouvelles possibilités.

Conteneur et éléments

- Un positionnement flex fait intervenir 2 types de blocs :
 - Le conteneur qui est l'élément dans lequel va se faire le positionnement flex.
 - Les éléments qui sont positionnés au sein du conteneur.



Activer le positionnement flex

- Pour utiliser flex, il suffit de changer la propriété `display` de l'élément conteneur.

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

Exemple

- Sans Flex

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus!

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

- Avec Flex

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus!

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Pellentesque nihil, hendrerit, class facilisis voluptate, et proin impedit fermentum tincidunt omnis, delectus! Quo quaerat quaerat aliquet officiis exercitation? Ipsum atque, sollicitudin?

Orientation

- Un conteneur Flex possède 2 axes : un axe principal et un axe secondaire. Ces deux axes sont orientés.
- La propriété `flex-direction` permet de définir l'axe principal.

Valeur de flex-direction	Orientation de l'axe principal	
row (défaut)	Horizontal de gauche à droite	1 2 3 4
row-reverse	Horizontal de droite à gauche	4 3 2 1
column	Vertical de haut en bas	1 2
column-reverse	Vertical de bas en haut	2 1







Plusieurs lignes

- Par défaut, les éléments d'un positionnement flex se répartissent sur une ligne (ou colonne). La propriété `flex-wrap` du conteneur permet d'autoriser les éléments à passer à la ligne.

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






Alignement principal

- La propriété `justify-content` du conteneur permet de manipuler la façon dont se répartissent les éléments le long de l'axe principal.

Valeur de <code>justify-content</code>	Effet
flex-start (défaut)	
flex-end	
center	
space-between	
space-around	
space-evenly	

Alignement secondaire

- La propriété `align-items` du conteneur permet de manipuler l'alignement des éléments sur l'axe secondaire.

Valeur de align-items	Effet
flex-start	
flex-end	
center	
stretch (défaut)	

Can I Use ?

- Depuis la sortie de Edge en 2015, Flex box est maintenant supporté par tous les principaux navigateurs.
 - Attention à Internet Explorer néanmoins qui n'est compatible que partiellement et seulement dans ses dernières versions.
- <https://caniuse.com/#feat=flexbox>

Transitions et animations CSS

Transitions

- Les transitions CSS permettent de changer progressivement un élément CSS entre 2 styles.

```
a {  
    color:red;  
    text-decoration: underline;  
}
```

```
a:hover {  
    color: blue;  
    text-decoration: none;  
}
```

Ajouter une transition

- La propriété `transition` permet d'appliquer une règle de transformation à un élément.

```
a {  
    color:red;  
    text-decoration: underline;  
  
    transition: color 1s;  
}
```

Propriété sur laquelle
s'applique la transition

Temps de la
transition

Multiple transitions

- Il est possible d'avoir plusieurs transition sur un même élément.

```
a {  
    display: inline-block;  
    color:red; background-color: blue;  
    text-decoration: none;  
    transition: color 1s,background-color 1s,transform 2s;  
}  
  
a:hover {  
    color: blue; background-color: red;  
    transform: rotate(45deg);  
}
```

Animations

- Une animation est décrite avec l'instruction `@keyframes`. On peut y décrire les différentes étapes de l'animation.

```
@keyframes monAnimation {  
  0% {  
    background-color: red;  
  }  
  50% {  
    background-color: green;  
  }  
  100% {  
    background-color: red;  
  }  
}
```

Nom de l'animation

Appliquer une animation

- L'attribut `animation` permet d'appliquer une animation à un élément.

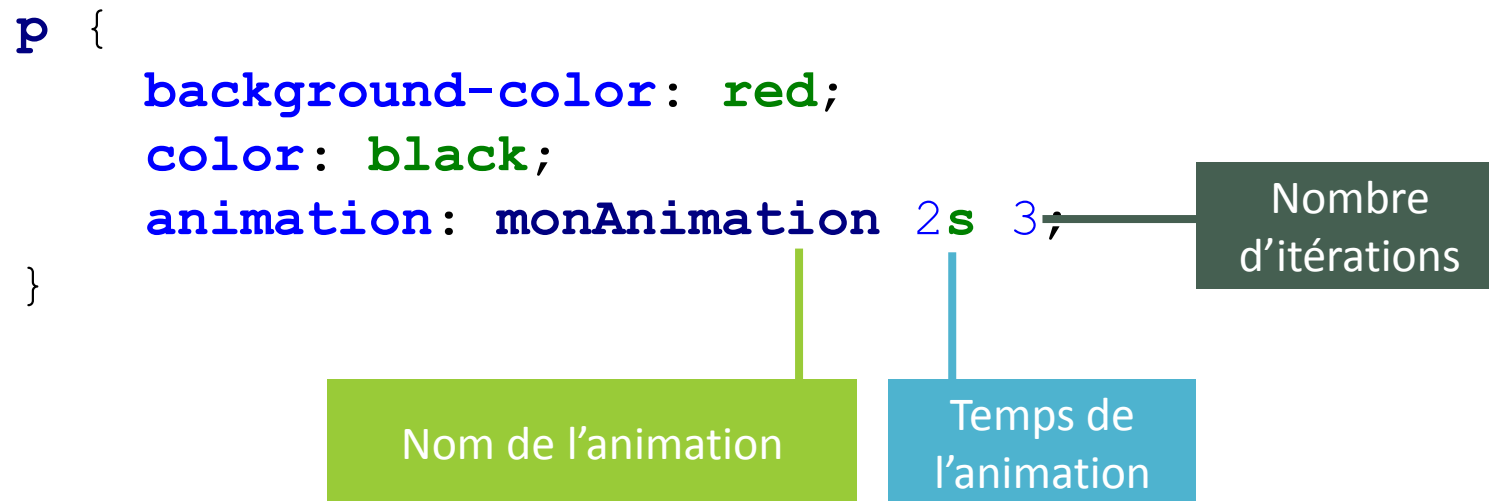
```
p {  
  background-color: red;  
  color: black;  
  animation: monAnimation 2s;  
}
```

Nom de l'animation

Temps de
l'animation

Itérations

- Par défaut, l'animation tourne une fois. Il est possible de spécifier un nombre d'itérations.
 - La valeur `infinite` permet de faire une animation permanente.



Can I Use ?

- Les transitions et animations sont bien supportées par les différents navigateurs.
- <https://caniuse.com/#feat=css-transitions>
- <https://caniuse.com/#feat=css-animation>

Sources

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- https://www.w3schools.com/css/css3_animations.asp
- https://www.w3schools.com/css/css3_transitions.asp