

# Chapitre 10

## Le retour à la ligne

### 10.1. Une propriété CSS liée au retour à la ligne

Nous venons de voir qu'il est possible de définir la direction des contenus. Ici nous allons voir qu'il est également possible de définir un retour ou non à la ligne des différents contenus et cela grâce à une nouvelle propriété CSS liée à *flexbox*. La propriété qui va être utilisée pour réaliser cela est la propriété *flex-wrap*.

La valeur par défaut de cette propriété CSS est la valeur *nowrap* qui signifie que les contenus ne doivent pas revenir à la ligne. Par conséquent, en définissant un conteneur de type *flex*, alors par défaut tous ses contenus seront affichés les uns à la suite des autres sans jamais revenir à la ligne.

### 10.2. Empêcher le retour à la ligne des contenus

Nous avons vu qu'à partir du moment où nous avons défini un comportement *flex* pour un conteneur alors ses contenus sont affichés en ligne. Ce qu'il faut également savoir, si la largeur totale des contenus dépasse la largeur du conteneur, les contenus ne reviendront pas à la ligne et continueront à s'afficher les uns à côté des autres quitte à sortir du conteneur. Ceci démontre un comportement par défaut de *flexbox*, empêcher le retour à la ligne des contenus. Tout ceci est régi par la propriété CSS qui se nomme *flex-wrap*. Sa valeur par défaut est *nowrap*, qui signifie ne pas revenir à la ligne. Voici comment elle se déclare.

```
flex-wrap: nowrap;
```

Reprenons notre fichier *style.css* à qui nous allons appliquer la nouvelle propriété *flex-wrap*. Nous n'appliquerons pas la propriété *flex-direction* car nous souhaitons afficher nos contenus en ligne, les uns à côté des autres. Et c'est précisément la valeur par défaut de cette propriété, comme nous l'avons vu plus haut.

```
#conteneur {
    background-color:#CCC;
    display: flex;
    flex-wrap: nowrap;
}
.contenu1 {
    background-color:#FFC;
    padding: 20px;
}
```

```
.contenu2 {
    background-color:#ff3;
    padding: 20px;
}
.contenu3 {
    background-color:#FCF;
    padding: 20px;
}
```

Apportons une modification à notre fichier HTML en lui ajoutant de nouveaux contenus. Pour cela nous allons copier et coller les trois contenus, puis nous renommerons nos nouveaux contenus en **contenu 4**, **contenu 5** et **contenu 6**.

Voici notre fichier ***index.html***

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Flexbox</title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <div id="conteneur">
        <div class="contenu1">Contenu 1</div>
        <div class="contenu2">Contenu 2</div>
        <div class="contenu3">Contenu 3</div>
        <div class="contenu1"><b>Contenu 4</b></div>
        <div class="contenu2"><b>Contenu 5</b></div>
        <div class="contenu3"><b>Contenu 6</b></div>
    </div>
</body>
</html>
```

Affichons maintenant le résultat de nos fichiers, dans notre navigateur :

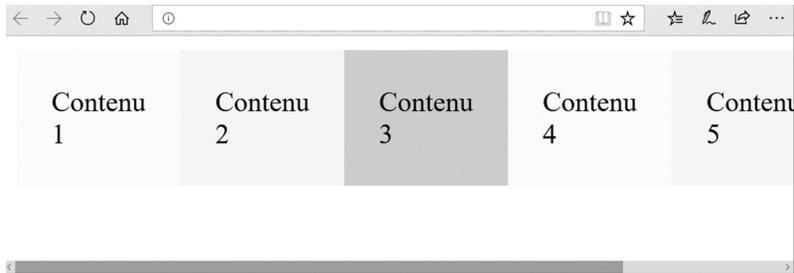


Figure 10-1 : Résultat de la propriété *flex-wrap* ayant pour valeur *nowrap*

Nous pouvons observer que les contenus ne reviennent pas à la ligne. Un scroll en bas du navigateur s'est affiché pour pouvoir faire défiler tous les contenus. Nous rappelons que ce comportement est un comportement par défaut. Donc que l'on écrive *flex-wrap:nowrap* dans le style CSS du conteneur ou non, le comportement des contenus sera celui-ci.

### 10.3. Autoriser le retour à la ligne des contenus

Il existe une valeur à donner à la propriété *flex-wrap* pour pouvoir autoriser un retour à la ligne des contenus lorsque ceux-ci n'ont plus de place pour s'afficher les uns à côté des autres. Cette valeur est *wrap*. Elle s'applique ainsi.

```
flex-wrap: wrap;
```

Nous allons appliquer cette nouvelle valeur à la propriété *flex-wrap* de notre conteneur.

```
#conteneur {  
    background-color:#CCC;  
    display: flex;  
    flex-wrap: wrap;  
}  
.contenu1 {  
    background-color:#FFC;  
    padding: 20px;  
}  
.contenu2 {  
    background-color:#ff3;  
    padding: 20px;  
}  
.contenu3 {
```

```
background-color:#FCF;
padding: 20px;
}
```

Nous ne touchons pas au fichier *index.html* que nous appelons à présent dans notre navigateur :

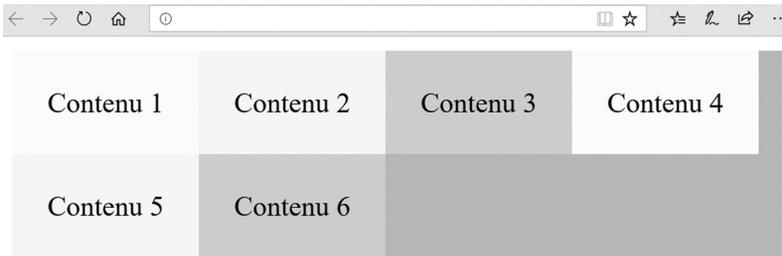


Figure 10-2 : Résultat de la propriété *flex-wrap* ayant pour valeur *wrap*

Nous pouvons observer que nos contenus reviennent automatiquement à la ligne dès qu'ils n'ont plus la place pour pouvoir continuer de s'afficher les uns à côté des autres. Et cela grâce à la valeur *wrap* donnée à la propriété *flex-wrap*.

#### 10.4. Autoriser le retour à la ligne inversé des contenus

Il existe une troisième valeur que l'on peut donner à la propriété *flex-wrap*, la valeur *wrap-reverse*. Cette valeur permet comme la valeur *wrap*, un retour à la ligne des contenus lorsque ceux-ci n'ont plus la place de pouvoir s'afficher les uns à côté des autres. La seule différence est qu'elle va les afficher de façon inversée. Voyons tout de suite cela en modifiant la valeur de la propriété *flex-wrap* du conteneur dans notre fichier *style.css*

```
#conteneur {
    background-color:#CCC;
    display: flex;
    flex-wrap: wrap-reverse;
}
.contenu1 {
    background-color:#FFC;
    padding: 20px;
}
.contenu2 {
    background-color:#ff3;
    padding: 20px;
}
```

```
}  
.contenu3 {  
  background-color:#FCF;  
  padding: 20px;  
}
```

Observons le résultat de cette modification dans notre navigateur.

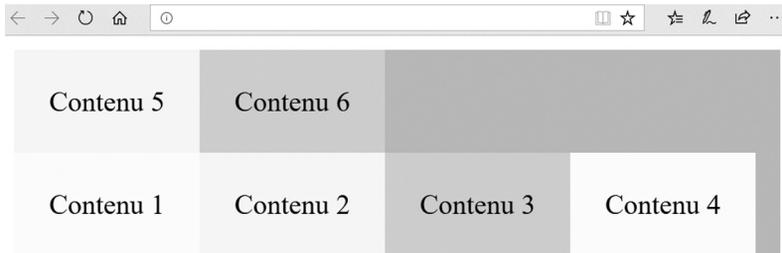


Figure 10-3 : Résultat de la propriété *flex-wrap* ayant pour valeur *wrap-reverse*

Nos contenus reviennent bien à la ligne mais cette fois en s'inversant.

## 10.5. Une propriété réunissant deux propriétés

Il est possible de pouvoir réunir les propriétés *flex-direction* et *flex-wrap* en une seule propriété. Cette propriété se nomme *flex-flow*. Elle prend pour valeur la valeur donnée à *flex-direction* puis celle donnée à *flex-wrap*. Par exemple, si nous voulons un alignement en ligne des contenus et autoriser un retour à la ligne, nous pouvons alors écrire ceci

```
flex-flow: row wrap;
```

## 10.6. Conclusion

Nous venons de découvrir une nouvelle propriété, la propriété *flex-wrap* qui permet le retour ou non à la ligne des contenus. Nous rappelons que par défaut, cette propriété n'autorise pas le retour à la ligne des contenus.



# Chapitre 11

## L'axe principal et l'axe secondaire

### 11.1. La notion d'axe

Nous avons vu précédemment que nous pouvons définir un alignement des contenus en ligne ou bien en colonne. Pour cela nous pouvons utiliser la propriété *flex-direction*. Si nous voulons un alignement des contenus en ligne alors nous donnons la valeur *row* et si nous voulons un alignement en colonne alors nous donnons la valeur *column*. Nous pouvons ajouter la valeur *reverse* dans le cas où nous souhaitons inverser l'ordre des contenus.

En clair, nous avons deux possibilités d'alignement, soit un alignement en ligne, c'est-à-dire suivant un axe horizontal, soit un alignement en colonne c'est-à-dire suivant un axe vertical. Cela va nous amener à une notion importante de *flexbox*, la notion d'axe principal et d'axe secondaire.

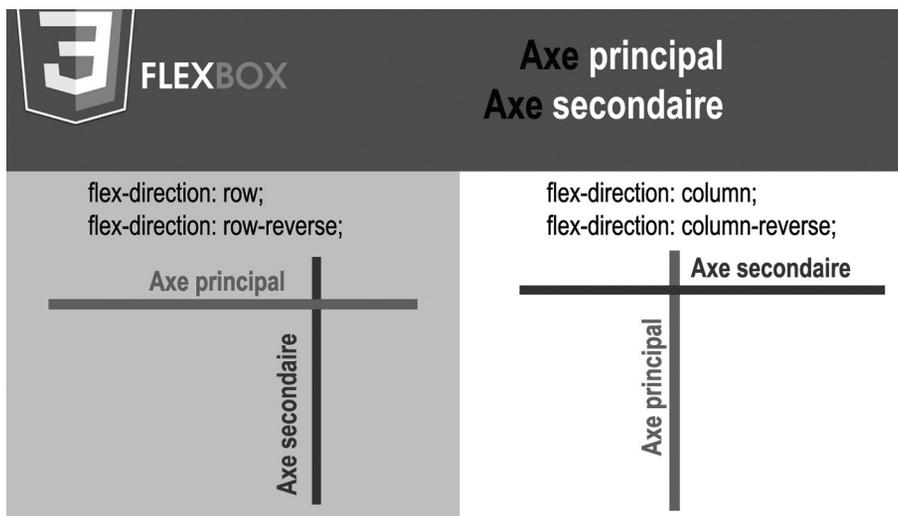


Figure 11-1 : Axe principal et axe secondaire

Lorsque nous définissons une direction en ligne (ou en ligne inversée), alors les contenus sont affichés en ligne. Cette direction est alors définie comme étant l'axe principal. De ce fait, tout ce qui sera affiché verticalement sera aligné suivant l'axe secondaire.

Dans le sens inverse, lorsque nous définissons une direction en colonne (ou en colonne inversée), alors les contenus sont affichés en colonne. Cette direction est alors définie comme étant l'axe principal. De ce fait, tout ce qui sera affiché horizontalement sera aligné suivant l'axe secondaire.

## 11.2. Alignement sur l'axe principal

Maintenant que nous avons vu ce que sont l'axe principal et l'axe secondaire, nous allons alors pouvoir aborder une nouvelle propriété *flex* qui va nous permettre de positionner nos contenus suivant ces axes.

Nous allons créer au niveau HTML, un conteneur qui va posséder trois boîtes *div*.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>FlexBox</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div id="conteneur">
  <div class="contenu1">Contenu 1</div>
  <div class="contenu2">Contenu 2</div>
  <div class="contenu3">Contenu 3</div>
</div>
</body>
</html>
```

Au niveau de notre fichier *style.css*, nous allons définir une direction des contenus en ligne au niveau du conteneur et nous allons donner une couleur de fond à chacun des contenus ainsi qu'à notre conteneur.

```
#conteneur {
  display: flex;
  flex-direction: row;
  background-color: #ccc;
}
.contenu1 {
  background-color: #f9f;
  padding: 20px;
}
.contenu2 {
  background-color: #ff3;
  padding: 20px;
}
```