COURS HTML/CSS J5

LES FLEXBOX CSS

Au débuts d'internet les ensembles d'éléments importants constituant une pages se succédaient en sautant à la ligne (Comportement par défaut des blocs).

Il y eu plusieurs solutions pour pouvoir pouvoir mettre des blocs sur une mème ligne.

On utilisait dans un premier temps des tableaux html , les cellules contenant des ensemble d'élément.

L'avènement du CSS a donné la possibilité d'aligner plusieurs bloc avec une nouvelle propriété : float. Mais cette propriété est loin d'être satisfaisante.

L'invention des flexbox donna enfin la possibilité de faire des mises en page complexes.

Disposition des balises bloc : en ligne ou à la ligne

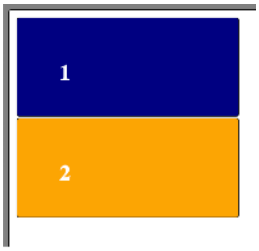
Par défaut, les balises de type bloc se succèdent à la ligne.

C'est le cas pour les balises de texte :

<hn> <p>

Et les balises structurantes :

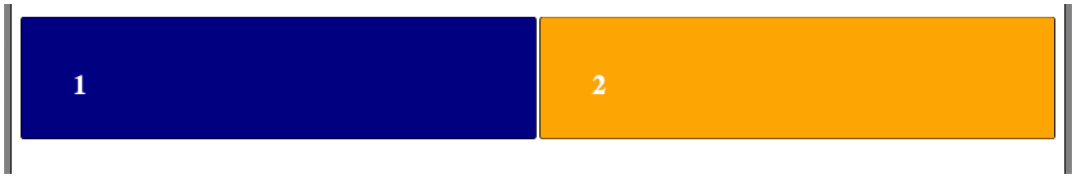
<header> <nav> <main> <section> <articles> <aside> <footer>



Il est possible de changer cette disposition par défaut grâce aux flexbox.

Pour aligner plusieurs blocs sur une même ligne on utilise la propriété ***display*:** suivi de la valeur ***flex*;**

#selector { ***display***: **flex**; }



Cette propriété est appliqué à l'élément parent pour que ces enfant se mettent sur une même ligne.

Les flex box combines un ensemble de propriétés qui nous permettent de modifier radicalement la mise en page.

*Exercice : Créez un conteneur #container comprenant quatre blocs de couleurs différentes ayant une classe .elements nommés **1, 2, 3** & **4** avec une hauteur et une largeur minimum de 100 pixels puis testez les propriétés suivantes. (vous pouvez créer 4 classes de couleurs red, green, blue, orange avec la propriété background-color)*

La direction :

On peut modifier la direction de l'alignement avec la propriété ***flex-direction*:**
Cette propriété a 4 valeurs possibles :

- ***row*** : Sur la même ligne de gauche à droite (valeur par défaut).
- ***row-reverse*** : Sur la même ligne de droite à gauche.
- ***column*** : Sur une colonne (rétablit le comportement initial du saut de ligne).
- ***column-reverse*** : Sur une colonne mais en sens inverse.

Retour à la ligne:

Dans le cas où il y aurait trop de blocs pour un même ligne, on peut utiliser la propriété ***flex-wrap:*** qui permet de mettre les blocs excédentaires à la ligne suivante.

Cette propriété possède trois valeurs :

- ***wrap:*** Les blocs excédentaires passent à la ligne suivante
- ***no-wrap:*** On force tous les blocs à rester sur un même ligne (valeur par défaut)
- ***wrap-reverse:*** Le sens des blocs est inversé (de droite à gauche) et les blocs excédentaires passent à la ligne suivante

Modifier l'ordre des éléments.

Il est possible d'assigner un ordre à chacun des éléments avec la propriété ***order***

```
.element:nth-child(1) { order: 3; }
```

Déterminer une taille pour chaque élément du conteneur

Nous pouvons déterminer une taille pour chacun des éléments du conteneur.

Cette taille sera déterminée en fonction des autres éléments. Si par exemple nous déterminons un conteneur de 12 colonnes, on pourra ainsi avoir un premier élément qui occupe 4 colonnes et le deuxième élément qui occupe 8 colonnes.

On utilise la propriété flex pour chacun des éléments.

Ex:

```
.element:nth-child(1)
{
    flex: 4;
}
.element:nth-child(2)
{
    flex: 8;
}
```

L'alignement avec la propriété *justify-content* et cinq valeurs possibles :

- ***flex-start*** : Un conteneur de plusieurs éléments est aligné au début de la ligne(par défaut)
- ***flex-start*** : Le conteneur est aligné à la fin de la ligne.
- ***center*** : Le conteneur est centré sur la ligne.
- ***space-between*** : Les éléments du conteneurs sont étirés sur la ligne.
- ***space-around*** : Les éléments du conteneurs sont dispatchés sur la ligne.

L'alignement s'applique également avec la propriété ***flex-direction*****:** ***column*****;**

NB: La valeur space-around appliquée au conteneur est similaire à la propriété margin: auto; appliquée à tous les selecteurs.

L'alignement sur l'axe secondaire.

Dans le cas où les éléments sont placé sur une ligne, donc horizontalement, l'axe secondaire sera vertical. Si ils sont placés sur colonne, donc verticalement, l'axe secondaire sera horizontal.

La propriété ***align-items:*** nous permet de modifier l'emplacement du conteneur par rapport à l'axe secondaire. Cette propriété est déclinée en 5 valeurs possibles :

- ***flex-start*** : Le conteneur est aligné au début de l'axe secondaire.
- ***flex-end*** : Le conteneur est aligné à la fin de l'axe secondaire.
- ***center*** : Le conteneur est centré sur l'axe secondaire.
- ***stretch*** : Étiré sur tout l'axe secondaire.
- ***baseline*** : Aligné sur la ligne de base (similaire à ***flex-start***).

Exercice : Testez ces nouvelles propriétés avec le conteneur que vous avez créé dans le premier exercice. (Il faudra assigner une hauteur minimum au conteneur sans quoi vous ne pourrez vérifier la propriété aligns-items)

NB: Le centrage vertical et horizontal du conteneur peut être simplement obtenu aux éléments du conteneur en leurs appliquant la propriété & la valeur ***margin*: *auto*;**

Il existe donc deux possibilités pour obtenir ce centrage vertical & horizontal :

```
#container
{
display: flex;
justify-content: center;
align-items: center;
}
```

```
#container
{
display: flex;
}
.elements
{
margin: auto;
}
```

Aligner sur l'axe secondaire un seul élément parmi les autres avec le sélecteur avancé :nth-child()

On peut aligner un seul élément à l'exclusion des autres en le ciblant sur l'un des enfants et en utilisant la propriété ***align-self*** déclinée principalement avec les quatre valeurs : ***start, end, center*** & ***stretch***.

(Le sélecteur :nth-child() en indiquant un chiffre à l'intérieur des parenthèses permet de cibler l'un des enfant de l'élément parent)

Entrainez-vous avec Flexbox Froggy. Un jeu pour s'initier aux flex box :

https://flexboxfroggy.com/#fr