

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a stylized tree structure, set against a blue gradient background.

MODÈLE DE BOÎTES FLEXIBLE - FLEXBOX

INTRODUCTION AUX FLEXBOX

- les éléments HTML pouvaient être affichés de différentes manières et que le type d'affichage conditionnait la disposition par défaut d'un élément :
 - Affichage en bloc avec `display:bloc`
 - Affichage en ligne avec `display:inline`
 - Affichage sous forme de tableau avec `display:table`
 - ETC
- Il existe un mode d'affichage et de disposition très puissant : les flexbox

DÉFINITION DU FLEXBOX

- Le FLEXBOX va nous permettre de contrôler facilement et avec précision l'alignement, la direction, l'ordre et la taille de nos éléments (ou plus précisément de nos boîtes). → conteneurs flexibles
- Flexibles : les enfants directs vont être des éléments flexibles qui vont pouvoir se réarranger (se redimensionner, se réaligner, etc.) automatiquement dans leur conteneur lorsque celui-ci change de dimension.
- Alternative à la propriété float ! (cf : [flexbox1.html](#))

CONTENEUR & ÉLÉMENTS FLEXIBLES (FLEX-ITEMS)

- Définir un conteneur flexible avec `display:flex`

→ Tous les éléments directement contenus dans ce conteneur (c'est-à-dire tous les enfants directs) vont alors automatiquement devenir des éléments flexibles.

On peut créer autant de niveau d'imbrication de conteneurs et d'éléments flexibles que souhaité

→ ajouter `display:flex` à un flex-item pour qu'il serve de conteneur flexible à ses propres enfants.

`Display:flex` → propriétés de type block

`Display:inline-flex` → propriétés de type inline

Les propriétés `float`, `clear` et `vertical-align` ne fonctionnent pas sur les éléments flexibles

AXE PRINCIPAL ET SECONDAIRE

- La propriété flex-direction va nous permettre de définir quel va être l'axe principal pour un conteneur flexible et ses éléments flexibles ainsi que la direction des éléments le long de cet axe. Le deuxième axe sera ainsi appelé axe secondaire. (cf: flexbox2.html)

GÉRER LA DIRECTION DES FLEX-ITEMS

- L'axe principal et la propriété flex-direction :
 - Row : Valeur par défaut. L'axe principal est l'axe horizontal et les éléments flexibles vont se placer en ligne, dans le sens de la lecture (de gauche à droite pour un français) ;
 - Row-reverse : L'axe principal est l'axe horizontal et les éléments vont se placer en ligne. Cette fois-ci, les éléments se placent dans le sens inverse de la lecture ;
 - Column : L'axe principal est l'axe vertical et les éléments vont se placer en colonne, en partant du début du conteneur vers la fin (du haut vers le bas par défaut) ;
 - Column-reverse : L'axe principal est l'axe vertical et les éléments vont se placer en colonne, en partant de la fin du conteneur vers le début (du bas vers le haut par défaut). (cf: [flexbox3.html](#))

GÉRER LES DÉPASSEMENTS DES ÉLÉMENTS DU CONTENEUR

Pour éviter ce comportement souvent non souhaité, nous allons pouvoir utiliser la propriété flex-wrap qui va pouvoir donner la possibilité aux éléments flexibles d'aller à la ligne ou en colonne s'ils dépassent de leur conteneur.

- Flex-wrap : nowrap (valeur par défaut)
- Flex-wrap : wrap (Les éléments qui dépassent du conteneur vont avoir la possibilité d'aller à la ligne ou de se positionner sur une nouvelle colonne à partir du début du conteneur et en allant vers la fin de celui-ci)
- Flex-wrap : wrap-reverse (Les éléments qui dépassent du conteneur vont avoir la possibilité d'aller à la ligne ou de se positionner sur une nouvelle colonne à partir cette fois-ci de la fin du conteneur et en allant vers le début) (cf : flexbox4.html & flexbox4bis)

Notation raccourcie avec flex-flow : column wrap; (=flex-direction & flex-wrap)

L'ORDRE D'AFFICHAGE DES ÉLÉMENTS FLEXIBLES

Appliquer la propriété `order` aux éléments flexibles

- La valeur `order` d'un élément va être comparée à celle des autres et l'élément possédant la plus petite valeur sera affiché en premier, celui possédant la deuxième plus petite valeur en deuxième et etc.
- Si plusieurs éléments possèdent la même valeur pour la propriété `order`, les éléments seront affichés dans l'ordre d'écriture du code.

(cf : `flexbox5.html`)

GÉRER L'ALIGNEMENT DES ÉLÉMENTS FLEXIBLES

Aligner les éléments d'un conteneur flexible par rapport à leur axe principal ou secondaire en nous servant des propriétés suivantes :

- `Justify-content;`
- `Align-items;`
- `Align-content;`
- `Align-self;`

GÉRER L'ALIGNEMENT DES ÉLÉMENTS SELON L'AXE PRINCIPAL AVEC JUSTIFY-CONTENT

`justify-content` : aligner les éléments d'un conteneur flexibles selon l'axe principal.

- `Flex-start` : Valeur par défaut. Les éléments vont être concentrés au début du conteneur (selon leur axe principal) ;
- `Flex-end` : Les éléments vont être concentrés à la fin du conteneur (selon leur axe principal) ;
- `center` : Les éléments vont être centrés dans le conteneur (selon leur axe principal) ;
- `Space-between`

GÉRER L'ALIGNEMENT DES ÉLÉMENTS SELON L'AXE PRINCIPAL AVEC JUSTIFY-CONTENT

- Space-between : Les éléments vont être régulièrement distribués dans le conteneur. Les éléments se trouvant contre un bord du conteneur vont être collés au bord ;
- Space-around : Les éléments vont être régulièrement distribués dans le conteneur. Chaque élément va posséder le même espace et les espaces vont être cumulatifs entre deux éléments, ce qui fait que la taille de l'espace entre le conteneur et un élément contre le bord du conteneur sera deux fois plus petite qu'entre deux éléments ;
- Space-evenly : Les éléments vont être régulièrement distribués dans le conteneur. L'espace entre le bord du conteneur et un élément sera le même que celui entre deux éléments. (cf : [flexbox6.html](#))

GÉRER L'ALIGNEMENT DES ÉLÉMENTS SELON L'AXE SECONDAIRE

3 Propriétés :

- align-items : permet de gérer l'alignement des éléments au sein d'une ligne (ou d'une colonne selon les axes principal et secondaires choisis) dans l'axe secondaire ;
- align-content : permet de gérer l'alignement des lignes (ou des colonnes selon l'axe) les unes par rapport aux autres dans l'axe secondaire et ne va donc avoir un effet que si nous avons plusieurs lignes (ou colonnes) d'éléments flexibles.
- align-self : permet de gérer l'alignement d'un élément flexible en particulier dans l'axe secondaire.

GÉRER L'ALIGNEMENT DES ÉLÉMENTS DANS UNE LIGNE OU DANS UNE COLONNE SELON L'AXE SECONDAIRE AVEC ALIGN-ITEMS

Align items → justify-content (mais sur l'axe secondaire)

- Stretch : Valeur par défaut. Les éléments vont s'étirer dans leur axe secondaire jusqu'à remplir tout l'espace disponible ;
- Flex-start : Les éléments vont être placés au début de leur conteneur en fonction de l'axe secondaire ;
- Flex-end : Les éléments vont être placés à la fin de leur conteneur en fonction de l'axe secondaire ;
- Center : Les éléments vont être placés au milieu de leur conteneur en fonction de l'axe secondaire ;
- Baseline : Les éléments vont être alignés dans leur axe secondaire de telle sorte à ce que leurs lignes de base (ligne imaginaire sur laquelle est écrit le texte) soient alignées. (cf : flexbox7.html & flexbox8.html)

GÉRER L'ALIGNEMENT DES LIGNES OU DES COLONNES LES UNES PAR RAPPORT AUX AUTRES SELON L'AXE SECONDAIRE AVEC ALIGN-CONTENT

- Stretch : Valeur par défaut. Les lignes ou colonnes vont s'étirer dans leur axe secondaire jusqu'à remplir tout l'espace disponible ;
- Flex-start : Les lignes ou colonnes vont être placés au début de leur conteneur en fonction de l'axe secondaire ;
- Flex-end : Les lignes ou colonnes vont être placés à la fin de leur conteneur en fonction de l'axe secondaire ;
- Center : Les lignes ou colonnes vont être placés au milieu de leur conteneur en fonction de l'axe secondaire ;
- Space-between & space-around (cf : [flexbox9.html](#))

GÉRER L'ALIGNEMENT D'UN ÉLÉMENT EN PARTICULIER DANS SON AXE SECONDAIRE AVEC ALIGN-SELF

- Ne pas appliquer la propriété au conteneur mais bien aux éléments flexibles et lui passer les valeurs : stretch / flex-start / flex-end / center / baseline
- Cette propriété va surcharger la valeur de align-items pour les éléments pour lesquels elle a été définie. Cela signifie en d'autres termes que ce sera la valeur de align-self qui sera retenue pour l'alignement par-dessus la valeur de align-items (cf : flexbox10.html)

GÉRER LA TAILLE ET LA FLEXIBILITÉ DES ÉLÉMENTS FLEXIBLES – FLEX-BASIS

- Définir une taille de départ des éléments flexibles avec la propriété flex-basis (exprimée généralement en %) surcharger la valeur width ou height
- Cf : [flexbox1 1_basis.html](#)

GÉRER LA TAILLE ET LA FLEXIBILITÉ DES ÉLÉMENTS FLEXIBLES – FLEX-GROW

- Définir la capacité des éléments à s'étirer dans leur conteneur pour remplir l'espace vide.
- On passe un nombre positif à flex-grow qui va définir la quantité d'espace disponible qu'un élément doit récupérer par rapport aux autres.
- La valeur par défaut de flex-grow est 0 ce qui signifie que les éléments n'ont pas le droit de grossir dans leur conteneur.
- Reprenons l'ex d'avant avec flex-grow à 1 → (Cf : [flexbox11_basis.html](#))

GÉRER LA TAILLE ET LA FLEXIBILITÉ DES ÉLÉMENTS FLEXIBLES – FLEX-SHRINK

- gérer leur capacité à rétrécir pour éviter que des éléments ne dépassent du conteneur avec la propriété flex-shrink.
- Même principe que flex-grow hormis sa valeur par défaut qui est 1
- Notez par ailleurs qu'un élément ne va pouvoir rétrécir que jusqu'au point où son contenu est prêt à dépasser de l'élément.
- Notation raccourcie avec flex → flex-grow flex-shrink flex-basis
→ ex : flex : 1 1 40%;

CAS D'UTILISATION ET LIMITES

- Effectuer un centrage parfait (dans les 2 axes)
- Créer un menu avec le flexbox
- Grouper des éléments (aligner d'un côté ou de l'autre du conteneur)
- Styliser des formulaires (input & bouton envoyer sur la même ligne pour l'ergonomie)
- Marges des éléments par rapport au conteneur à contrebalancer en valeurs négatives sur le conteneur lui même

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a blue gradient background, resembling a circuit board or a neural network.

MERCI À VOUS !
AVEZ-VOUS DES QUESTIONS ?