

Université			
		de Strasbourg	

HTML



CSS



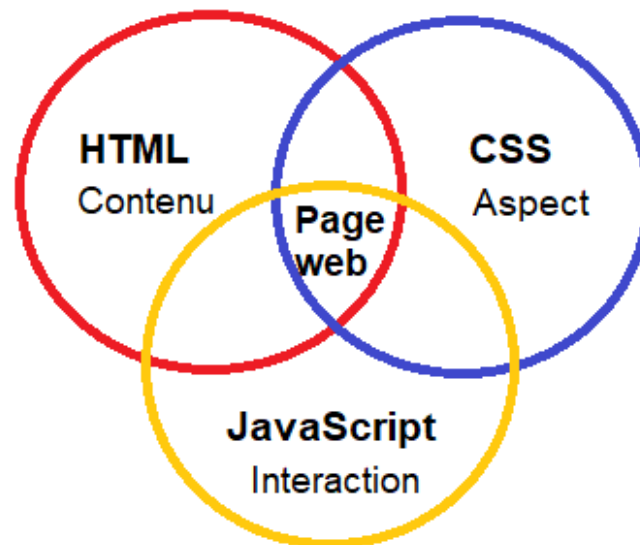
JS



Conception et Architecture Web & IHM

Master GPI

Nous avons déjà vu comment créer un fichier HTML qui a pour objectif de stocker le contenu du site. Chaque contenu doit être positionné dans la bonne balise ce qui permet de lui donner du sens (notion de balise sémantique apportée par HTML5). Nous allons maintenant voir comment mettre en forme ce contenu par le langage CSS (cascading style sheets).



Université			
		de Strasbourg	

HTML



CSS



JS



CSS

Plan

1. Le principe général

1 – Le principe général

- L'idée c'est d'appliquer un style à un ou plusieurs éléments HTML. Pour cela on va utiliser un sélecteur :

Principe et formalisme	Exemple
<pre>unSelecteur { proprieteA: valeur; proprieteB: valeur; ... }</pre>	<pre>h3 { font-style: italic; }</pre>

1 – Le principe général

Il existe 3 possibilités pour placer ces règles de style :

- Dans le code HTML directement (déconseillé)
- Dans l'entête de la page web en utilisant la balise `<style>`
- Dans un ou plusieurs fichiers distincts ayant pour extension `.css` (méthode conseillée)

Plan

1. Le principe général
2. Les sélecteurs d'éléments

2 – Les sélecteurs d'éléments

- Pour illustrer les sélecteurs je vais prendre un exemple fictif. Admettons qu'il existe une balise `<amphi>` et une balise `<etudiant>`. La représentation de cet amphi serait donc la suivante :

```
<amphi>  
  <etudiant>...</etudiant>  
  <etudiant>...</etudiant>  
  ...  
</amphi>
```

Et moi j'ai le super pouvoir de changer en CSS la couleur des vêtements des étudiants avec la propriété `color`...

2 – Les sélecteurs d'éléments

- Sélecteur d'un type de balise

Problème : Comment faire pour que tous les éléments d'un même type de balise soient concernés par la mise en forme souhaitée ?

- **Exemple fictif** : je veux que tous les étudiants de cet amphi soient habillés en bleu :

Solution : Utiliser un sélecteur de type de balise

Côté CSS	Côté HTML
<pre>etudiant { color: blue; }</pre>	Rien à préciser, le style va s'appliquer à tous les éléments de type <etudiant>

2 – Les sélecteurs d'éléments

- **Exemple réel** : Je veux que les éléments de type `<p>` soient affichés en bleu

Côté CSS	Côté HTML
<pre>p { color: blue; }</pre>	Rien à faire, tous les éléments de type <code><p></code> auront leur texte en bleu

Un même style peut concerner plusieurs balises

Côté CSS	Côté HTML
<pre>p, div { color: blue; }</pre>	Rien à faire, tous les éléments de type <code><p></code> et <code><div></code> auront leur texte en bleu

2 – Les sélecteurs d'éléments

- Sélecteur de classe

Problème : Comment faire pour qu'une partie seulement des éléments d'un type de balise soient concernés par une mise en forme ?

- **Exemple fictif** : Je veux que les étudiants du dernier rang de l'amphi soient habillés en rouge

Solution : Créer une classe.

Côté CSS	Côté HTML
<pre>etudiant.dernierRang { color: red; }</pre>	<pre><etudiant class="dernierRang"> ... </etudiant></pre>

2 – Les sélecteurs d'éléments

- **Exemple réel** : Je veux qu'une partie des éléments de type `<p>` soient affichés en rouge :

Côté CSS	Côté HTML
<pre>p.enRouge { color: red; }</pre>	<pre><p class="enRouge"> ... </p></pre>

Attention au formalisme : **p.enRouge**

2 – Les sélecteurs d'éléments

- Sélecteur de classe (suite)

Une même classe peut concerner plusieurs types de balise :

Côté CSS	Côté HTML
<pre>.enRouge { color: red; }</pre>	<pre><p class="enRouge"> ... </p> <h4 class="enRouge"> ...</h4></pre>

Attention au formalisme : **.enRouge**

2 – Les sélecteurs d'éléments

- Sélecteur par identifiant

Problème : Comment faire si parmi les éléments de type d'une balise, il y en a un (et un seul) qui doit être concerné par une mise en forme ?

- **Exemple fictif** : Je veux que l'étudiant dont l'INE = 5871 soit habillé en vert

Solution : Créer un identifiant

Côté CSS	Côté HTML
<pre>etudiant#INE5871 { color: green; }</pre>	<pre><etudiant id="INE5871"> ... </etudiant></pre>

2 – Les sélecteurs d'éléments

- Sélecteur par identifiant (suite)
- **Exemple réel** : Un des éléments de type `<p>` doit être affiché en vert

Côté CSS	Côté HTML
<pre>p#leParagEnVert { color: green; }</pre>	<pre><p id="leParagEnVert"> ... </p></pre>

Attention au formalisme : `p#leParagEnVert`

2 – Les sélecteurs d'éléments

- Sélecteur par placement dans son conteneur

On peut sélectionner le premier ou le dernier élément dans son conteneur

- **Exemple fictif** : Je veux que le premier étudiant placé dans l'amphi soit habillé en noir :

Côté CSS	Côté HTML
<pre>etudiant:first-child { color: black; }</pre>	<pre><amphi> <etudiant>...</etudiant> <etudiant>...</etudiant> ... </amphi></pre>

Idem pour la propriété :last-child

2 – Les sélecteurs d'éléments

- Sélecteur par placement dans son conteneur (suite)
 - **Exemple réel** : Je veux que le premier élément du conteneur <body> soit affiché en italique :

Côté CSS	Côté HTML
<pre>p:first-child { font-style: italic; }</pre>	<pre><body> <p>...</p> <section> <article> ... </article> </section> </body></pre>

2 – Les sélecteurs d'éléments

- Uniquement les éléments inclus dans un élément
 - **Exemple réel:** Je veux que les éléments de type `<p>` des `<article>` soient affichés en italique (et pas les autres) :

Côté CSS	Côté HTML
<pre>article p { font-style: italic; }</pre>	<pre><body> <p>...</p> <section> <article> <p> ... </p> </article> </section> </body></pre>

2 – Les sélecteurs d'éléments

- Uniquement les éléments enfants directs
 - **Exemple réel:** Je veux que les éléments de type `<p>` enfants directs de la section soient affichés en vert :

Côté CSS	Côté HTML
<pre>section > p { color: green; }</pre>	<pre><body> <section> <p>Paragraphe section</p> <article> <p>Paragraphe article</p> </article> </section> </body></pre> <div><p>Paragraphe section</p><p>Paragraphe article</p></div>

2 – Les sélecteurs d'éléments

- Pseudo-classes

Il est possible d'appliquer un style uniquement lorsqu'un état particulier est actif.

- **Exemple fictif** : Lorsque je m'adresse à un étudiant, je veux qu'il soit momentanément habillé en blanc

Côté CSS

```
etudiant:repondreQuestionDuProf {  
    color: white;  
}
```

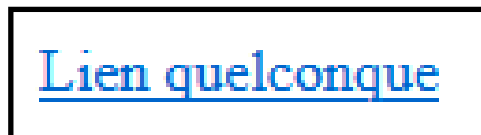
2 – Les sélecteurs d'éléments

- Pseudo-classe (suite)

Une pseudo-classe fréquemment utilisée est :hover qui permet de modifier un style au passage de la souris.

- **Exemple réel** : Lorsque je passe la souris sur un lien hypertexte il doit être surligné en jaune

Côté CSS	Côté HTML
<pre>a:hover { background-color: yellow; }</pre>	<pre>Lien quelconque</pre>



2 – Les sélecteurs d'éléments

Dans cette initiation, on est loin d'avoir fait le tour des sélecteurs et des possibilités offertes par le CSS, il faudra se référer à la documentation officielle pour rechercher les cas particuliers dont vous aurez besoin.

Plan

1. Le principe général
2. Les sélecteurs d'éléments
3. Règles conflictuelles

3 – Règles conflictuelles

- **Attention**, lorsqu'un style est appliqué à un élément HTML, il est appliqué par héritage à tous ses éléments enfants.

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>section { color: red; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

3 – Règles conflictuelles

- **Attention**, les styles sont appliqués dans l'ordre de la feuille CSS. Si un élément se voit attribuer plusieurs styles par le même sélecteur, c'est le dernier style qui s'applique.

Code HTML	Code CSS
<code><h1>Initiation CSS</h1></code>	<pre>h1 { color: red; } h1 { color: blue; }</pre>

Initiation CSS

3 – Règles conflictuelles

- **Attention**, si un élément se voit attribuer plusieurs styles par des sélecteurs différents, une priorité sur le type de sélecteur s'applique.

Code HTML	Code CSS
<pre><h1 class="titre">Initiation CSS</h1></pre> <div>Initiation CSS</div>	<pre>.titre { color: red; } h1 { color: blue; }</pre>

Retenons : identifiant > classe > élément

Plan

1. Le principe général
2. Les sélecteurs d'éléments
3. Règles conflictuelles
4. Mettre en forme le texte

4 – Mettre en forme le texte

- Le choix de la police de caractères

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; } p { font-family: Consolas, sans- serif; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

4 – Mettre en forme le texte

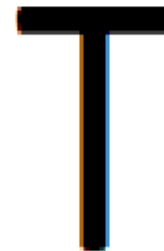
- Le choix de la police de caractères

serif (avec empattement)



VS

sans-serif (sans empattement)



4 – Mettre en forme le texte

- Le choix de la taille du texte

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; } p { font-family: Consolas, sans- serif; font-size: 30px; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

4 – Mettre en forme le texte

- Le choix de l'inclinaison

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; } p { font-family: Consolas, sans- serif; font-style: italic; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

4 – Mettre en forme le texte

- Le choix de l'épaisseur du texte

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; } p { font-family: Consolas, sans- serif; font-weight: bold; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

4 – Mettre en forme le texte

- Le soulignement

Code HTML	Code CSS
<pre><section> <article> <h3>Initiation CSS</h3> <p> Lorem ipsum dolor sit amet, consectetur adipiscing.</p> </article> </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; } p { font-family: Consolas, sans- serif; text-decoration: underline; }</pre>

Initiation CSS

Lorem ipsum dolor sit amet, consectetur adipiscing.

4 – Mettre en forme le texte

- Le positionnement du texte

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; }</pre> <p> Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariatur ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.</p>

Par défaut aligné à gauche

4 – Mettre en forme le texte

- Le positionnement du texte

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: right; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariatur ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

4 – Mettre en forme le texte

- Le positionnement du texte

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: justify; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariat ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

4 – Mettre en forme le texte

- Espacement entre les lignes

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: justify; line-height: 1.5; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariatur ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

Par défaut 1

4 – Mettre en forme le texte

- Espacement entre les lignes

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: justify; line-height: 0.5; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo
 officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non
 voluptates pariatur ab voluptatem alias enim iustum autem et tempore
 perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium
 debilis et consequatur magni quo voluptatem animi est autem
 dignissimos. Qui nihil excepturi non dolores aperiam 33 magni
 architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

Par défaut 1

4 – Mettre en forme le texte

- La couleur du texte

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: justify; color: green; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariat ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

On peut utiliser

Le nom des couleurs, leur valeur hexadécimale et les triplets RGB (plus de détails à venir dans le cours).

4 – Mettre en forme le texte

- Des coins arrondis
 - Par défaut les coins sont carrés

Code HTML	Code CSS
<pre><section> <h3>Initiation CSS</h3> ... </section></pre>	<pre>h3 { font-family: "Times New Roman", sans-serif; text-align: center; background-color: orange; border-radius: 50px; }</pre>

Initiation CSS

4 – Mettre en forme le texte

- Des coins arrondis (suite)



`border-radius: 50px 51px 54px 55px / 57px 52px 53px 56px;`

4 – Mettre en forme le texte

- Les bordures

`border: 4px solid black;`

↑
épaisseur

↑
style

↑
couleur

| solid (plein)

| dotted (pointillés)

| dashed (tirets)

4 – Mettre en forme le texte

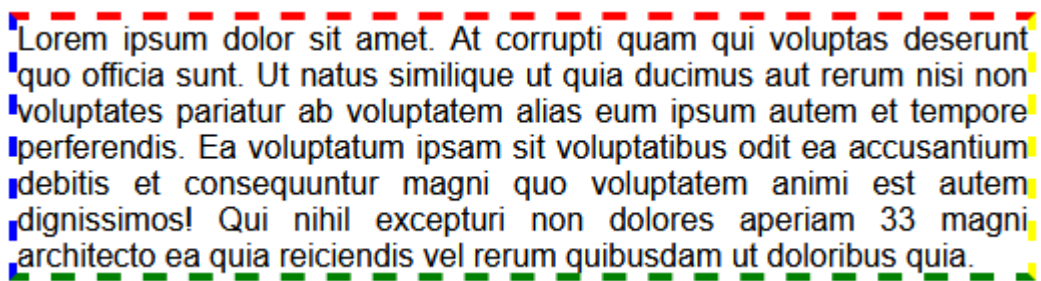
- Les bordures

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans- serif; text-align: justify; border: 5px dashed red; }</pre>

Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt
quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non
voluptates pariat ab voluptatem alias eum ipsum autem et tempore
preferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium
debitis et consequuntur magni quo voluptatem animi est autem
dignissimos! Qui nihil excepturi non dolores aperiam 33 magni
architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.

4 – Mettre en forme le texte

- Les bordures

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans-serif; text-align: justify; border-top: 4px dashed red; border-bottom: 4px dashed green; border-left: 4px dashed blue; border-right: 4px dashed yellow; }</pre> 

4 – Mettre en forme le texte

- La couleur en général

On peut utiliser le nom des couleurs, leur valeur hexadécimale ou le code RGB. Par exemple pour le vert :

code nom : **green**

code hexadécimal : **#008000**

code RGB : **rgb(0,128,0)**

Le **code** couleur **RGB (red-green-blue)** se présente sous la forme d'un triplet de nombres compris entre 0 et 255

4 – Mettre en forme le texte

- Mettre un arrière-plan
 - Une couleur

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans-serif; text-align: justify; background-color: yellow; }</pre> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt quo officia sunt. Ut natus similique ut quia ducimus aut rerum nisi non voluptates pariat ab voluptatem alias eum ipsum autem et tempore perferendis. Ea voluptatum ipsam sit voluptatibus odit ea accusantium debitis et consequuntur magni quo voluptatem animi est autem dignissimos! Qui nihil excepturi non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia.</p>

4 – Mettre en forme le texte

- Mettre un arrière-plan
 - Une image

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans-serif; text-align: justify; background-image: url('image.png'); background-repeat: no-repeat; }</pre>

4 – Mettre en forme le texte

- Mettre un arrière-plan
 - Une image

Code HTML	Code CSS
<pre><section> <article> <p>Lorem ipsum dolor sit amet. At corrupti quam qui voluptas deserunt ... non dolores aperiam 33 magni architecto ea quia reiciendis vel rerum quibusdam ut doloribus quia. </p> </article> </section></pre>	<pre>p { font-family: Arial, sans-serif; text-align: justify; background-image: url('image.png'); background-repeat: repeat; }</pre>

Plan

1. Le principe général
2. Les sélecteurs d'éléments
3. Règles conflictuelles
4. Mettre en forme le texte
5. Positionner les blocs

5 – Positionner les blocs

- Chaque élément est vu comme une boîte rectangulaire !
- On considère 3 types d'élément qui vont avoir un comportement de placement différent :
 1. Les éléments **block** qui se placent naturellement les uns en dessous des autres dans le flux ;
 2. Les éléments **inline** qui se placent naturellement les uns après les autres (en ligne) dans le flux ;
 3. Les éléments **inline-block** se comportent comme les éléments inline mais peuvent être dimensionnés.

Remarque : On peut changer le comportement par défaut d'un élément grâce à la propriété **display** !

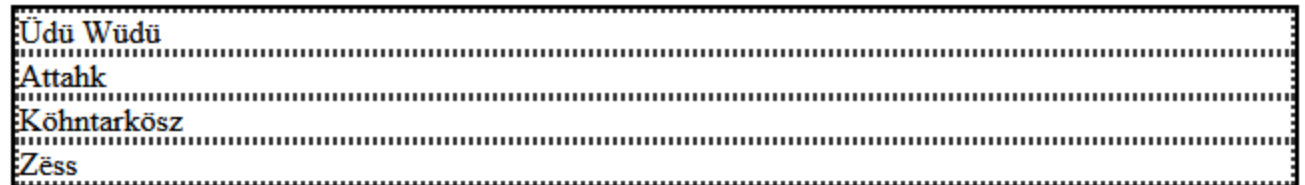
5 – Positionner les blocs

- Les éléments **block**
 - Quelques éléments HTML sont par défaut de type block : `<p>`, `<section>`, `<article>`; `<header>`, `<nav>`, ...
 - Comportement d'un élément block dans son conteneur :
 - Se place automatiquement verticalement (en colonne)
 - Occupe toute la largeur disponible
 - Peut être dimensionné

5 – Positionner les blocs

- Exemple éléments block

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: block; border: 2px dotted black; }</pre>



- Se place automatiquement verticalement ☒
- Occupe toute la largeur disponible ☒

5 – Positionner les blocs

- Exemple éléments block (suite)

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: block; border: 2px dotted black; width: 200px; }</pre>



- Peut être dimensionné ☒

5 – Positionner les blocs

- Les éléments **inline**

- Quelques éléments HTML sont par défaut de type inline : `<a>`, ``; `<header>`, ...
- Comportement d'un élément inline dans son conteneur :
 - Se place automatiquement horizontalement (en ligne)
 - N'occupe que la largeur nécessaire
 - Ne peut pas être dimensionné

5 – Positionner les blocs

- Exemple éléments inline

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: inline; border: 2px dotted black; }</pre>

Üdü Wüdü Attahk Köhntarkösz Zëss

- Se place automatiquement horizontalement ✓
- N'occupe que la largeur nécessaire ✓

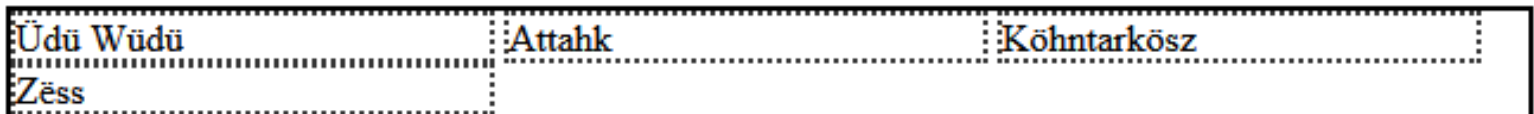
5 – Positionner les blocs

- Les éléments **inline-block**
 - Quelques éléments HTML sont par défaut de type inline-block : `<button>`, `<textarea>`, `<input>`, ...
 - Comportement d'un élément inline-block
 - Se place automatiquement horizontalement (en ligne)
 - N'occupe que la largeur nécessaire
 - Peut être dimensionné

5 – Positionner les blocs

- Exemple élément inline-block

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: inline-block; border: 2px dotted black; width: 200px; }</pre>

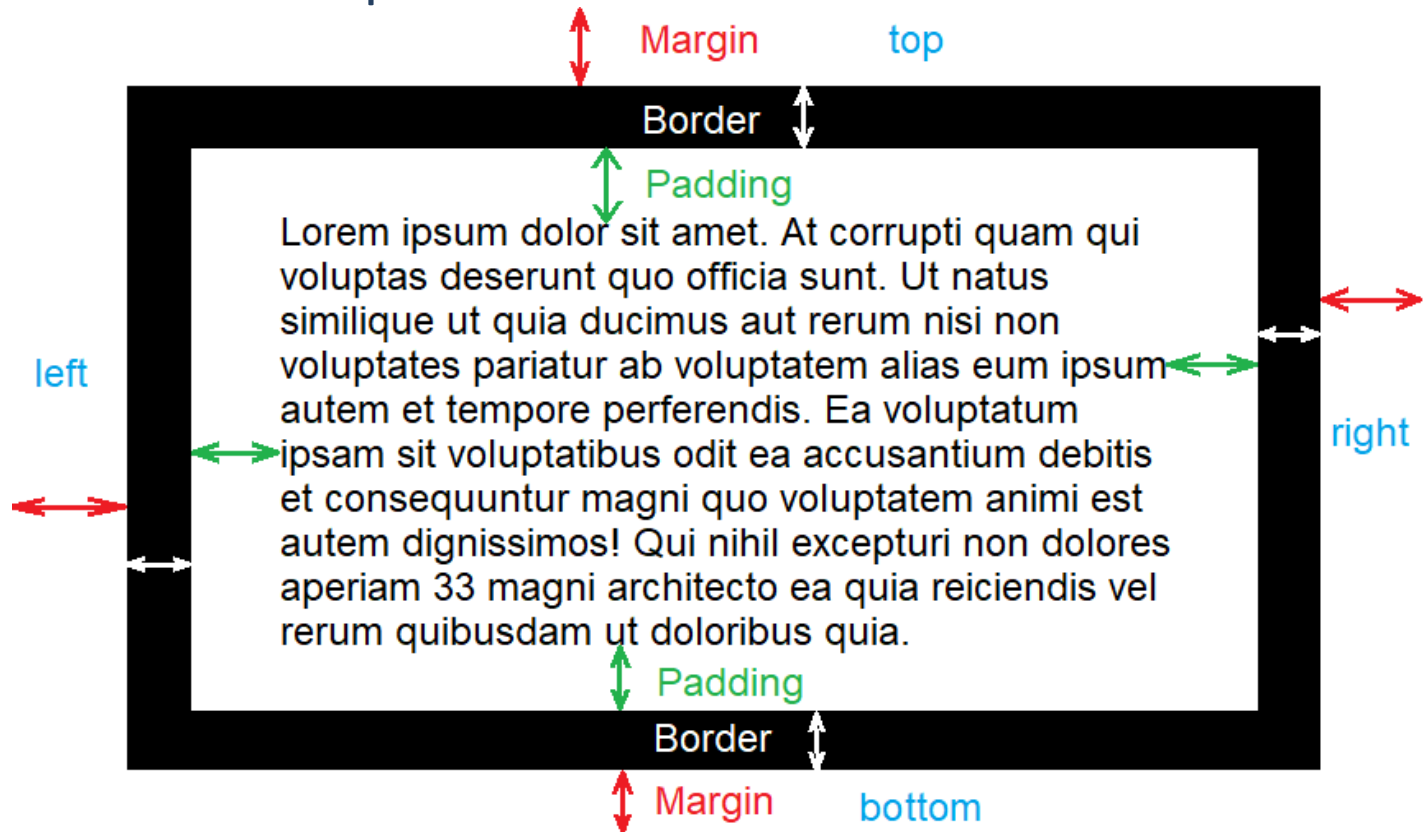


- Se place automatiquement horizontalement ☒
- Peut être dimensionné ☒

5 – Positionner les blocs

- Les espacements (margin-padding)

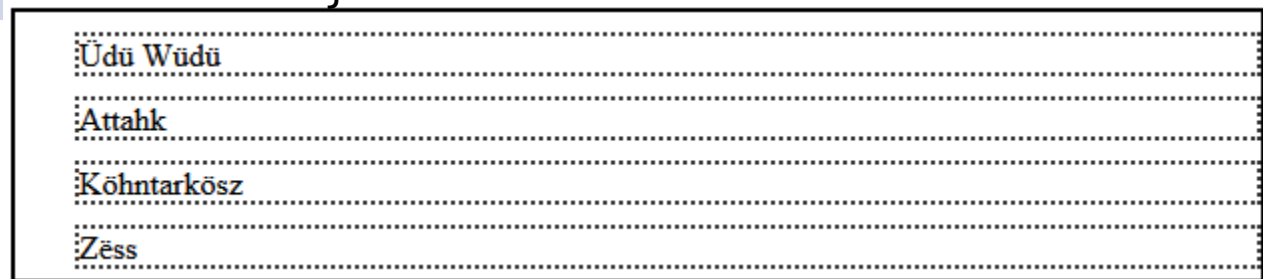
Ce schéma résume les possibilités de dimensionner les différentes marges :



5 – Positionner les blocs

- Margin gère l'espacement entre son parent et/ou ses voisins

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: block; border: 2px dotted black; margin: 10px 0 5px 30px; }</pre>

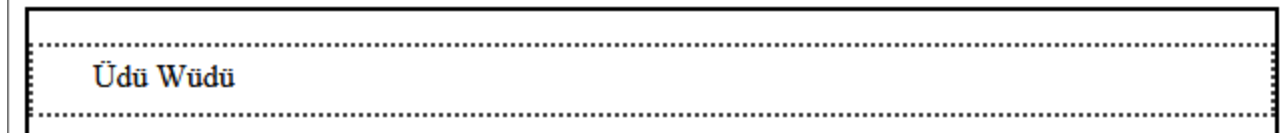


Placé dans le sens : top-right-bottom-left

5 – Positionner les blocs

- Padding gère l'espacement dans le bloc entre le contenu et la bordure

Code HTML	Code CSS
<pre><section> <p> Üdü Wüdü </p> <p> Attahk </p> <p> Köhntarkösz </p> <p> Zëss </p> </section></pre>	<pre>section { border: 2px solid black; } p { display: block; border: 2px dotted black; padding: 10px 0 5px 30px; }</pre>



Placé dans le sens : top-right-bottom-left

Plan

1. Le principe général
2. Les sélecteurs d'éléments
3. Règles conflictuelles
4. Mettre en forme le texte
5. Positionner les blocs
6. Les unités de taille

5 – Les unités de taille

Les unités les plus utilisées :

- **Unités relatives (à privilégier)**
 - rem : relative à la taille de police de l'élément racine `<html>`
 - em : relative à la taille de police héritée
 - % : 100% = 1em
- **Unités absolues**
 - px : en pixels (un pixel = 1 point sur l'écran)
 - cm : centimètres
 - in : inch (1 in = 2,54 cm = 96 pixels)

Il en existe plein d'autres...

6 – Les unités de taille

- Un exemple

Code HTML	Code CSS
<pre><section> <h2>Une liste</h2> <p>Voici une belle liste...</p> Elément 1 Elément 1.1 </section></pre>	<pre>html { font-size: 16px; } section { font-size: 20px; } h2 { font-size: 2em } p { font-size: 2rem; } li { font-size: 1.5em; }</pre>

6 – Les unités de taille

- Un exemple – aperçu du résultat :

Une liste 40 px (2em = 2 x 20)

Voici une belle liste ... 32 px (2rem = 2 x 16)

- Élément 1
 - Élément 1.1 30 px (1.5em = 1.5 x 20)

Plan

1. Le principe général
2. Les sélecteurs d'éléments
3. Règles conflictuelles
4. Mettre en forme le texte
5. Positionner les blocs
6. Les unités de taille
7. Les flexbox

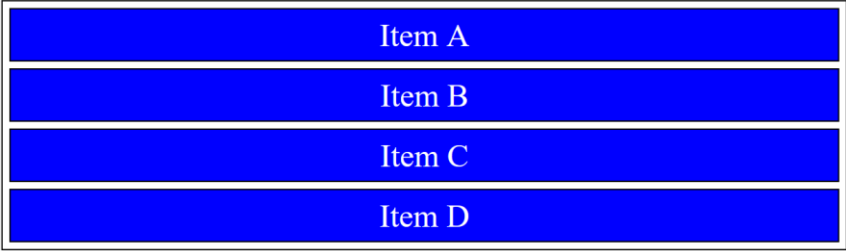
7 – Les flexbox

- Flexbox peut se traduire par "boite flexible", c'est un module qui permet de distribuer l'espace entre les objets d'un conteneur pour les placer comme on le souhaite. Cette technique permet d'aller beaucoup plus loin que le comportement par défaut des éléments (block, inline, inline-block).

Le fonctionnement de la propriété **flex-direction** est importante à comprendre pour commencer.

7 – Les flexbox

- Situation de départ sans flexbox

Code HTML	Code CSS
<pre><section> <p>Item A</p> <p>Item B</p> <p>Item C</p> <p>Item D</p> </section></pre> 	<pre>section { border: 2px solid black; } p { border: 2px solid black; font-size : 3rem; background-color: blue; color: white; text-align: center; margin: 10px 10px 10px 10px; padding: 10px 10px 10px 10px; }</pre>

Remarque : comportement normal puisque `<p>` par défaut est de type block (positionnement en colonne)

7 – Les flexbox

- On rajoute une classe container

- `display: flex;` //Indique un comportement en flexbox
- `flex-direction: row;` //Indique un placement horizontal

Code HTML	Code CSS
<pre><section class="container"> <p>Item A</p> <p>Item B</p> <p>Item C</p> <p>Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; }</pre>

Item A

Item B

Item C

Item D

- `flex-direction: row-reverse;` //placement horizontal inversé

Item D

Item C

Item B

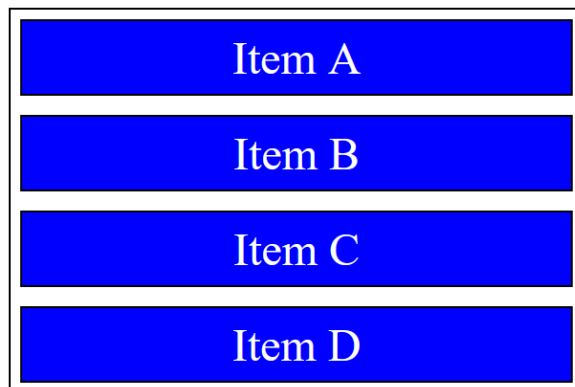
Item A

7 – Les flexbox

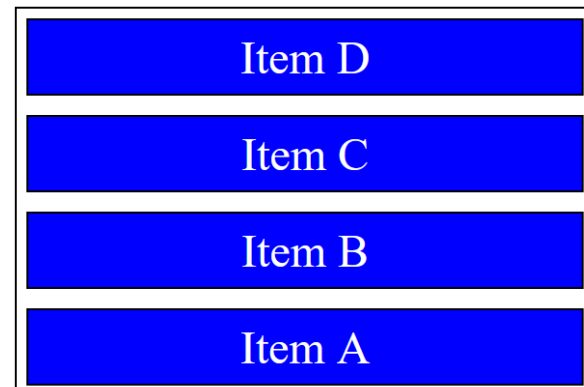
- `flex-direction: column;` //Indique un placement vertical

Code HTML	Code CSS
<pre><section class="container"> <p>Item A</p> <p>Item B</p> <p>Item C</p> <p>Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: column; }</pre>

`flex-direction: column;`



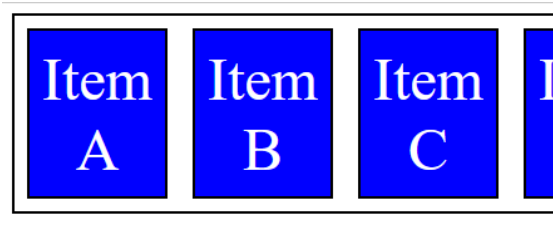
`flex-direction: column-reverse;`



7 – Les flexbox

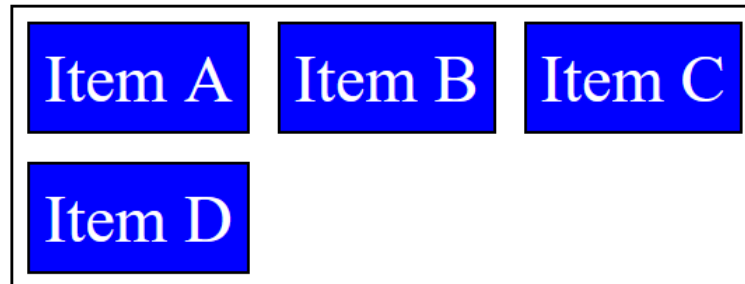
- Le comportement du passage à la ligne
 - On veut que les items soient disposés sur une seule ligne

- flex-wrap: nowrap;



- On autorise les items à passer automatiquement à la ligne

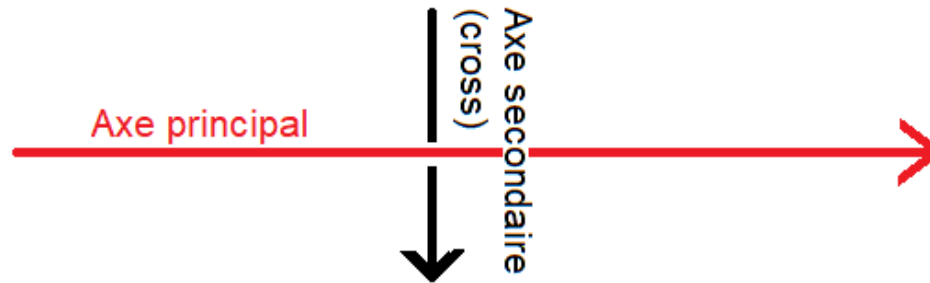
- flex-wrap: wrap;
 - flex-wrap: wrap-reverse;



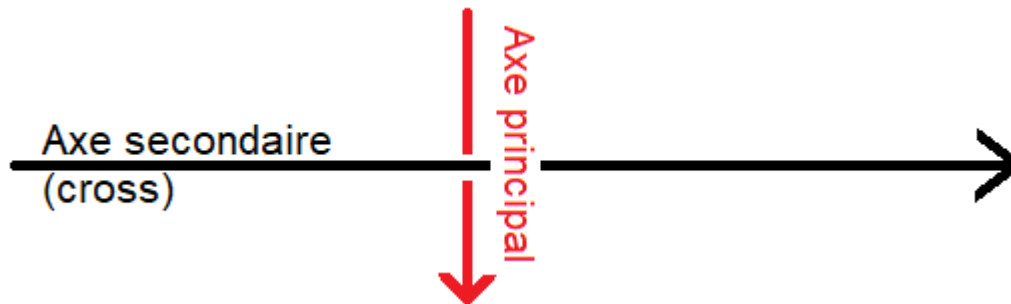
7 – Les flexbox

- Les notions d'axe principal et d'axe secondaire

Si le container est configuré en **flex-direction: row;** (alignement horizontal) :



Si le container est configuré en **flex-direction: column;** (alignement vertical) :



7 – Les flexbox

- Comment occuper l'espace disponible dans le conteneur ?

2 approches :

- On laisse le container gérer avec les propriétés suivantes :
 - justify-content
 - align-content
 - align-item
- On laisse les items se gérer eux-mêmes :
 - order
 - align-self

7 – Les flexbox

- Approche container
 - La propriété `justify-content` permet de gérer les espacements sur l'axe principal

Code HTML	Code CSS
<pre><section> <p>Item A</p> <p>Item B</p> <p>Item C</p> <p>Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; justify-content: flex-start; }</pre>

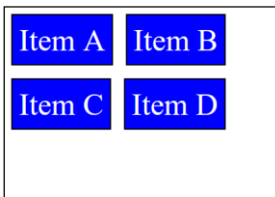
flex-start	<div>Item AItem BItem CItem D</div>
flex-end	<div>Item AItem BItem CItem D</div>
center	<div>Item AItem BItem CItem D</div>
space-between	<div>Item AItem BItem CItem D</div>
space-around	<div>Item AItem BItem CItem D</div>
space-evenly	<div>Item AItem BItem CItem D</div>

7 – Les flexbox

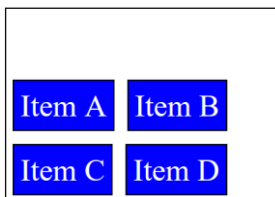
- Approche container
 - La propriété align-content permet de gérer les espacement sur l'axe secondaire

Code HTML	Code CSS
<pre><section> <p>Item A</p> <p>Item B</p> <p>Item C</p> <p>Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; flex-wrap: wrap; /* Obligatoire */ height: 300px; align-content: flex-start; }</pre>

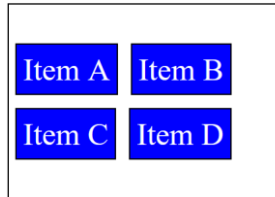
flex-start



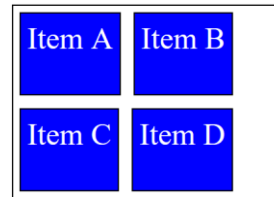
flex-end



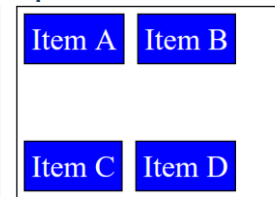
center



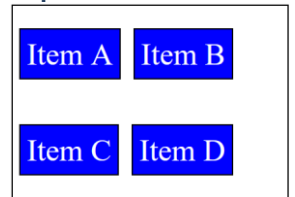
stretch



space-between



space-around

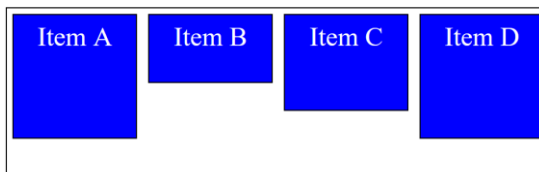


7 – Les flexbox

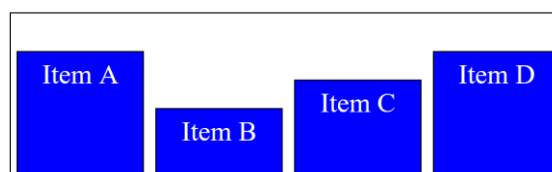
- Approche container
 - La propriété align-item permet de gérer l'alignement des items sur l'axe secondaire

Code HTML	Code CSS	
<pre><section> <p class="moyen">Item A</p> <p class="grand">Item B</p> <p class="petit">Item C</p> <p class="moyen">Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; flex-wrap: wrap; height: 300px; align-item: start; }</pre>	<pre>.grand { height: 200px; } .moyen { height: 150px; } .petit { height: 100px; }</pre>

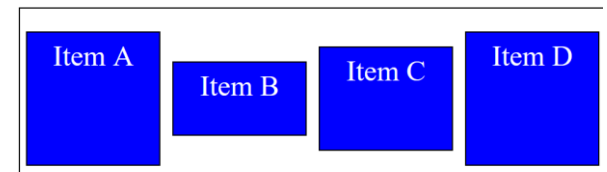
start



end



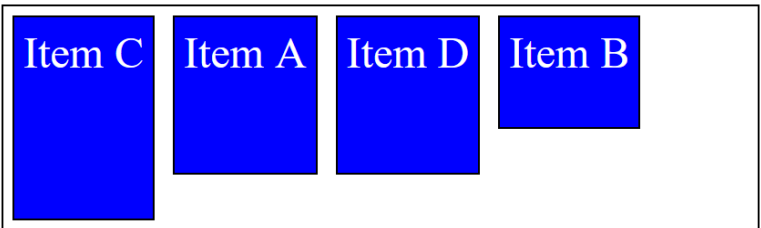
center



7 – Les flexbox

- Approche item
 - La propriété `order` permet de préciser un ordre d'affichage

Code HTML	Code CSS	
<pre><section> <p class="moyen">Item A</p> <p class="grand">Item B</p> <p class="petit">Item C</p> <p class="moyen">Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; flex-wrap: wrap; height: 300px; }</pre>	<pre>.grand { height: 200px; order: 1; } .moyen { height: 150px; order: 2; } .petit { height: 100px; order: 3; }</pre>

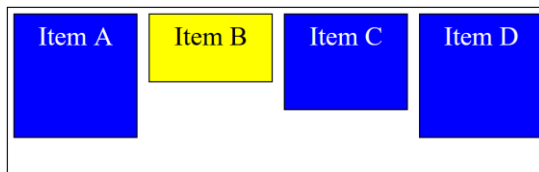


7 – Les flexbox

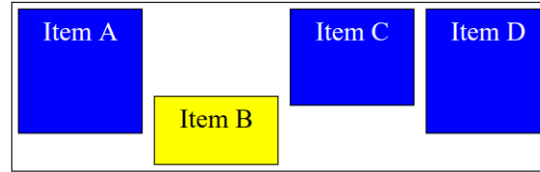
- Approche item
 - La propriété align-self permet à un item de s'aligner lui-même dans le conteneur

Code HTML	Code CSS	
<pre><section> <p class="grand">Item A</p> <p class="petit">Item B</p> <p class="moyen">Item C</p> <p class="grand">Item D</p> </section></pre>	<pre>.container { display: flex; flex-direction: row; flex-wrap: wrap; height: 300px; }</pre>	<pre>.grand { height: 200px; } .moyen { height: 150px; } .petit { height: 100px; background-color: yellow; color: black; align-self: start; }</pre>

start



end



center

