D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER — Université de Toulouse

Faculté Sciences et Ingénierie

Positionnement avancé en CSS

Mathieu RAYNAL

mathieu.raynal@irit.fr

http://www.irit.fr/~Mathieu.Raynal

D[i] Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER
Université de Toulouse
Faculté Sciences et Ingénierie

Positionnement avancé en CSS

Affichage multi-colonnes

Flexible Box Layout

Grid Layout

Définir la largeur de colonnes

- *column-width* définit la longueur optimale d'une colonne
  - Le nombre de colonne est calculé en fonction de la longueur du conteneur et de la longueur d'une colonne
  - Si la longueur demandée n'est pas un multiple de celle du conteneur :
    - On prend la valeur en dessous
    - La largeur des colonnes est réadaptée

Définir le nombre de colonnes

- *column-count* définit le nombre de colonnes souhaité
  - La longueur des colonnes est calculée en fonction de la longueur du conteneur et du nombre de colonne

- *columns* permet de définir l'une ou l'autre des deux propriétés précédentes
  - Si les 2 sont définies, column-count déterminera le nombre maximum de colonnes

Espacement entre les colonnes

- *column-gap* définit l'espacement entre 2 colonnes

- *column-rule* définit une « bordure » entre 2 colonnes. C'est une concaténation des règles :
  - *column-rule-color*
  - *column-rule-style*
  - *column-rule-width*

Mettre un élément sur plusieurs colonnes

- *column-span* étend ou non un élément sur l'ensemble des colonnes
  - *all* : étendre sur l'ensemble des colonnes
  - *none* : ne pas étendre

- *column-fill* réparti ou non le contenu sur les différentes colonnes
  - *auto*
  - *balance*

Saut de colonne

- Ajout de saut de colonne
  - *break-before*
    - *auto*
    - *column*
    - *avoid-column*
  - *break-after*
    - *auto*
    - *column*
    - *avoid-column*
  - *break-inside*
    - *auto*
    - *avoid-column*

D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER
Université de Toulouse
Faculté Sciences et Ingénierie

Positionnement avancé en CSS

Affichage multi-colonnes

Flexible Box Layout

Grid Layout

Flexbox

- Pour arranger les éléments HTML sur la page
  - Modification de l'ordre d'affichage des éléments
  - Ajustement de la taille des éléments
  - Alignement des blocs

- Pour utiliser flexbox dans un élément :
  - *display: flex;*

Définition de l'axe principal

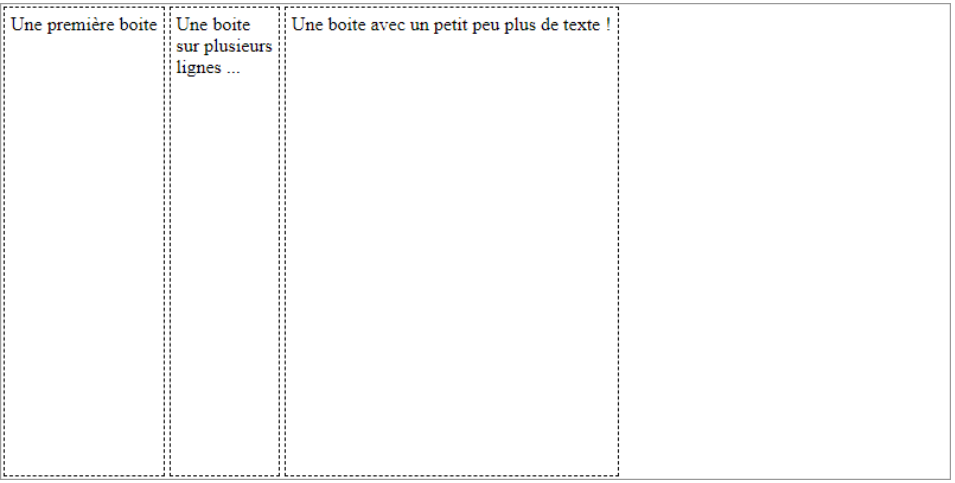
- Tous les éléments contenus dans un élément sont ajoutés selon un axe principal

- Axe principal défini grâce à *flex-direction*
  - *row*
    - l'axe principal sera aligné avec la direction « en ligne », et le sens d'écriture
    - Les éléments seront ajoutés les uns à la suite des autres
  - *row-reverse*
    - Dans le sens inverse du sens d'écriture

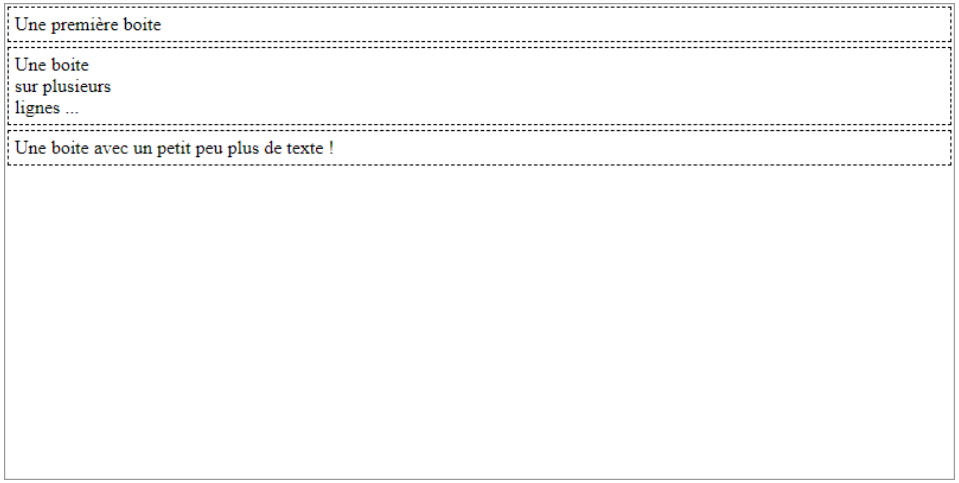
  → Attention, il n'y a pas de retour à la ligne automatique

  - *column*
    - l'axe principal suivra la direction « en bloc »
    - Les éléments seront ajoutés les uns en dessous des autres, de haut en bas
  - *column-reverse*
    - Les éléments seront ajoutés les uns au dessus des autres, de bas en haut

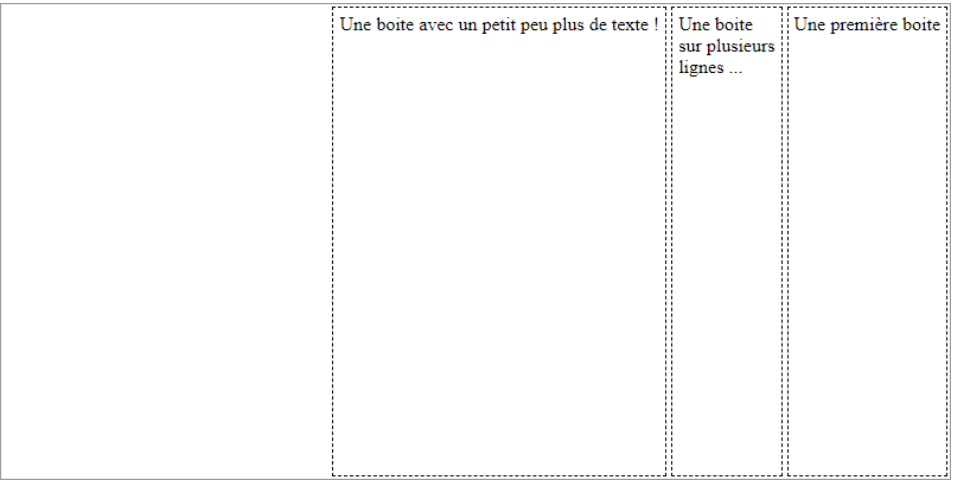
Définition de l'axe principal



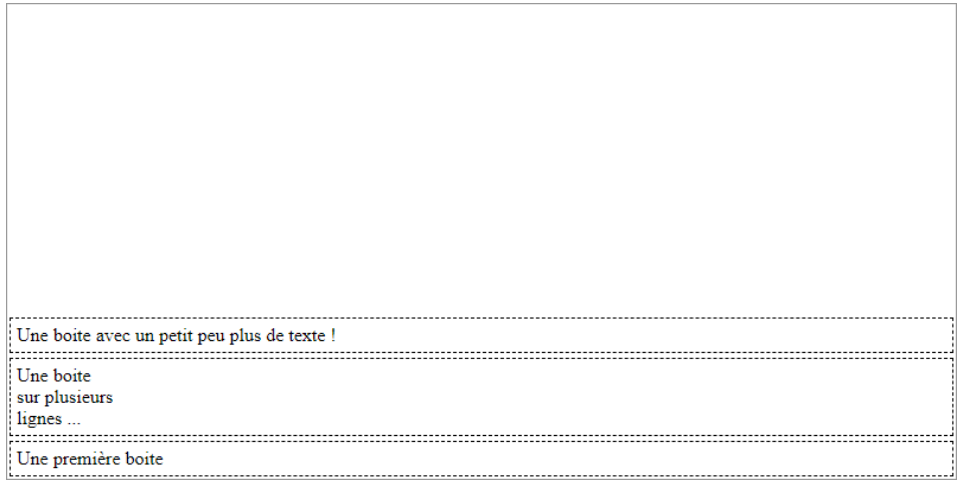
flex-direction: *row*;



flex-direction: *column*;



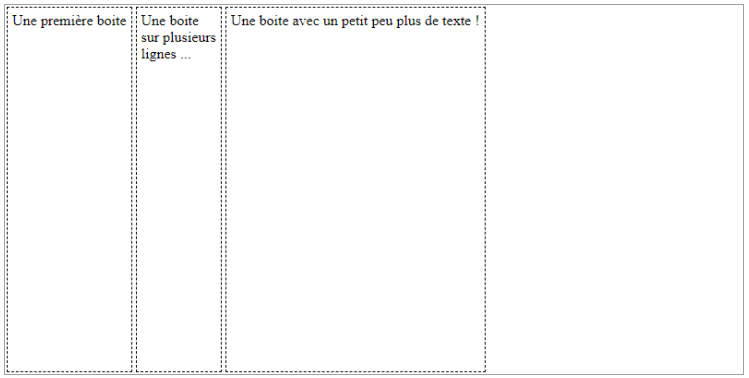
flex-direction: *row-reverse*;



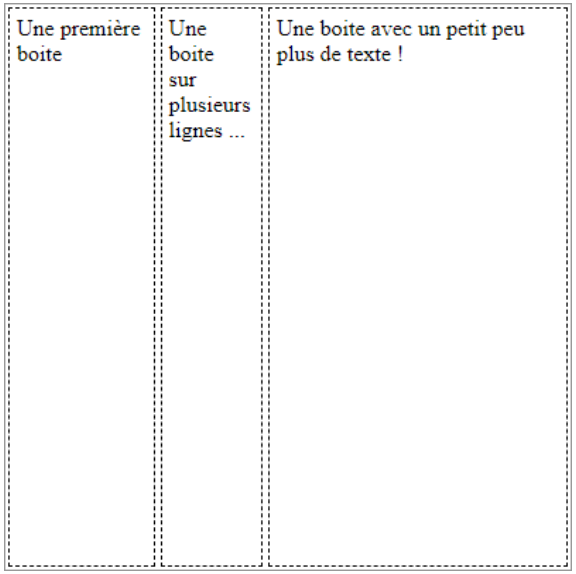
flex-direction: *column-reverse*;

Retour à la ligne ou non

- Les éléments dans un conteneur sont ajoutés sur le même axe en fonction de l'espace disponible
  - Les éléments essayent de prendre leur taille par défaut

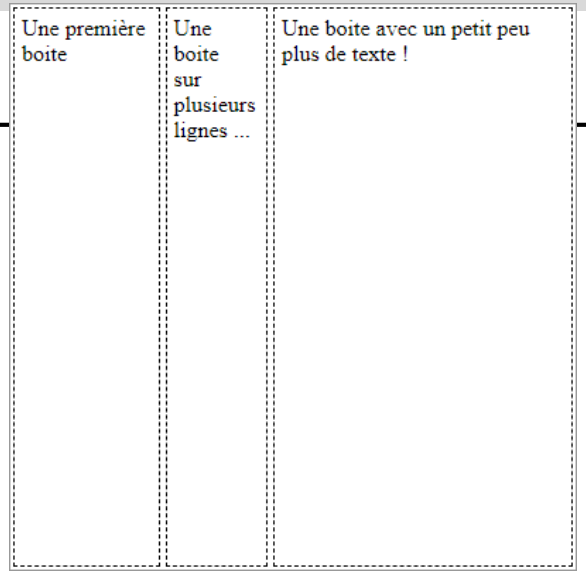


- Si la taille du conteneur est trop petite
  - Par défaut, tous les éléments restent sur la même ligne
  - les éléments sont redimensionnés



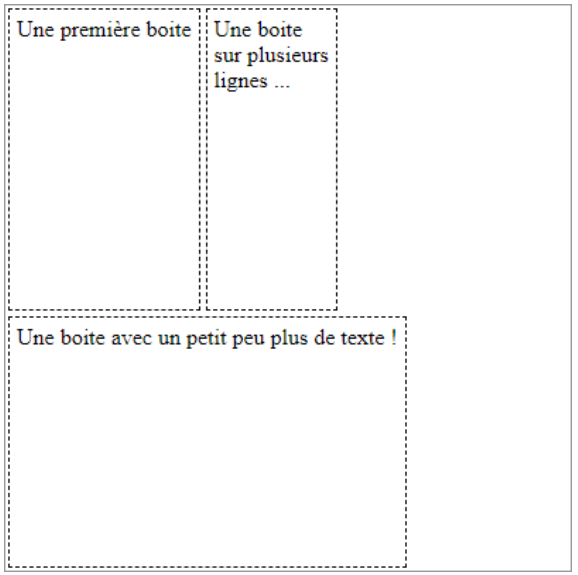
Retour à la ligne ou non

- Possibilité de demander un retour à la ligne pour conserver la taille des éléments :
  - *flex-wrap*: *wrap*;



flex-wrap: *no-wrap*;

- *flex-flow* est la propriété pour synthétiser les propriétés *flex-direction* et *flex-wrap*
  - La première valeur de cette propriété sera utilisée pour *flex-direction* et la seconde pour *flex-wrap*.



flex-wrap: *wrap*;

Ordre des éléments

- La propriété *order* permet de définir l'ordre d'apparition des éléments

Définition de la taille des éléments

- Attribut *flex-basis*
  - Définit la taille de « base » de l'élément, avant la répartition de l'espace
  - Valeur possible
    - *auto* : taille par défaut de l'élément
    - *0* : taille minimale
    - Une valeur relative (% ou vw) ou absolue (px)

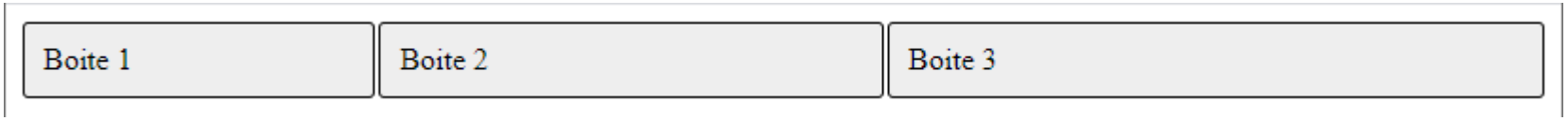
Répartition de l'espace disponible

- Les éléments prennent leur taille
- S'il reste de l'espace disponible, celui-ci peut être réparti entre les différents éléments
- *flex-grow* permet de répartir cet espace disponible
  - valeur qui définit un ratio d'espace disponible pris par l'élément

```
div.boite1{
    flex-grow: 0;
}

div.boite2{
    flex-grow: 1;
}

div.boite3{
    flex-grow: 2;
}
```



Répartition de l'espace manquant

- Les éléments prennent leur taille
- S'il manque de l'espace, les éléments peuvent être redimensionné
- *flex-shrink* permet de répartir l'espace à supprimer
  - valeur qui définit un ratio d'espace à enlever à l'élément

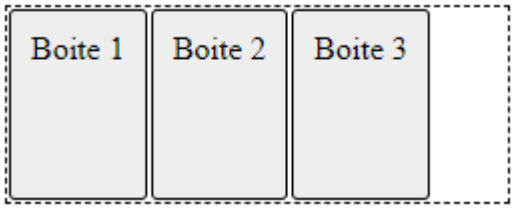
Gestion de l'espace

- La propriété *flex* permet de définir successivement les valeurs de flex-grow, flex-shrink, flex-basis

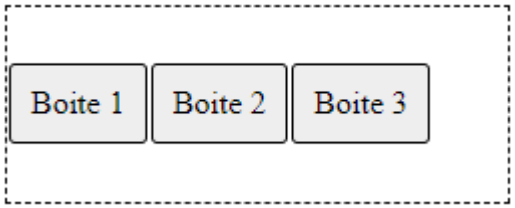
- Quelques valeurs spécifiques
  - *flex*: *initial* équivalent à *flex*: *0 1 auto*
  - *flex*: *auto* équivalent à *flex*: *1 1 auto*
  - *flex*: *none* équivalent à *flex*: *0 0 auto*
  - *flex*: *<nombre-positif>* équivalent à *flex*: *1 1 0*

Alignements des boites

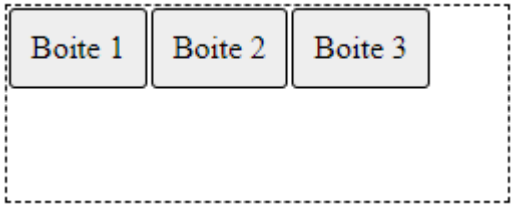
- *align-items* aligne les éléments le long de l'axe secondaire



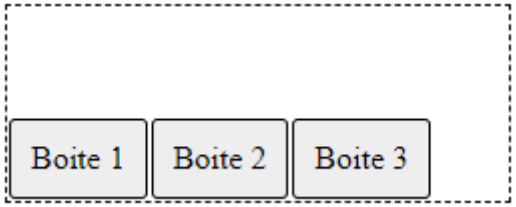
stretch



center



flex-start



flex-end

Justifier les éléments sur un axe

- *justify-content* : aligne les éléments le long de l'axe principal
  - *flex-start*
  - *flex-end*
  - *center*
  - *space-between* répartit l'espace disponible de façon égale entre chaque élément
  - *space-around* répartit l'espace disponible de façon égale autour de chaque élément (y compris au début et à la fin)
  - *space-evenly* : espace également réparti (y compris au début et à la fin)