



Développement Web

- CSS-



FlexBox



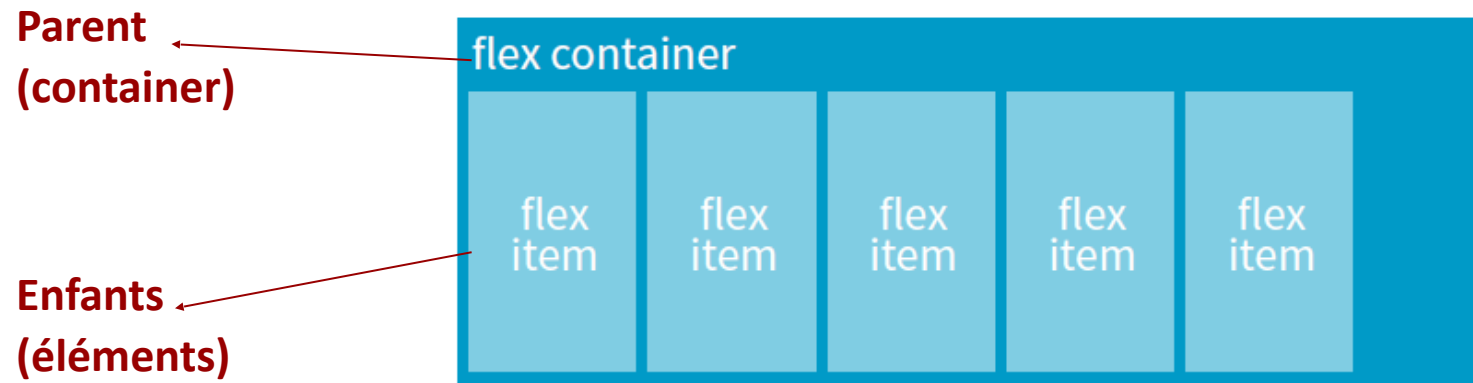
Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web



Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web
- **Principe** :
 - Définition d'un conteneur
 - Insertion de plusieurs éléments dans le conteneur



Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web

- **Principe :**

- Définition d'un conteneur
- Insertion de plusieurs élément dans le conteneur



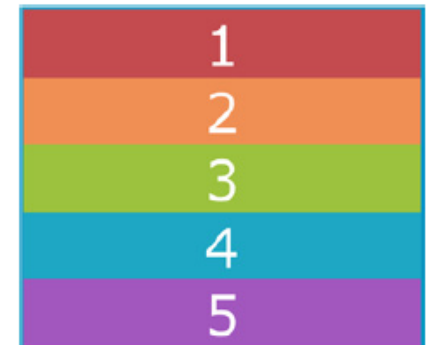
- **Déclaration :**

```
#container {  
    display: flex;  
}
```

```
<div id="container">  
  <div class="box box1">1</div>  
  <div class="box box2">2</div>  
  <div class="box box3">3</div>  
  <div class="box box4">4</div>  
  <div class="box box5">5</div>  
</div>
```



flex layout mode



block layout mode (sans flex)

La direction : flex-direction

- **rôle** : définir la façon d'agencement des éléments
- **valeurs** : row | column | row-reverse | column-reverse
- **valeur par défaut** : row
- **appliqué aux** : éléments du conteneur flex

flex-direction: **row**; (default)



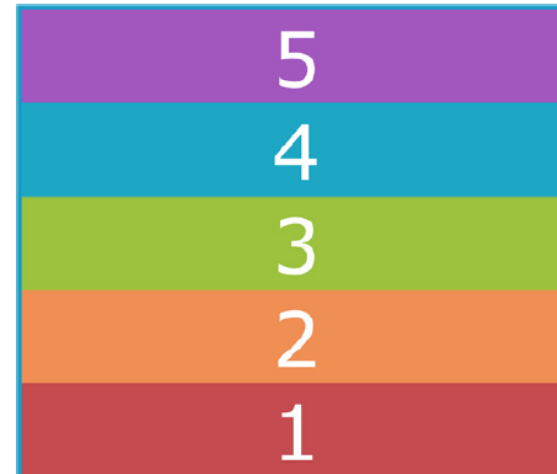
flex-direction: **row-reverse**;



flex-direction: **column**;



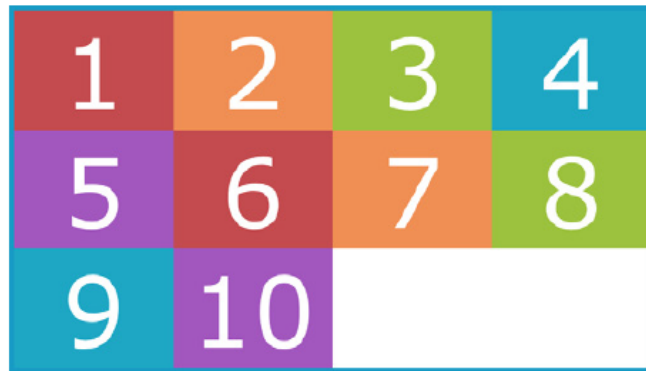
flex-direction: **column-reverse**;



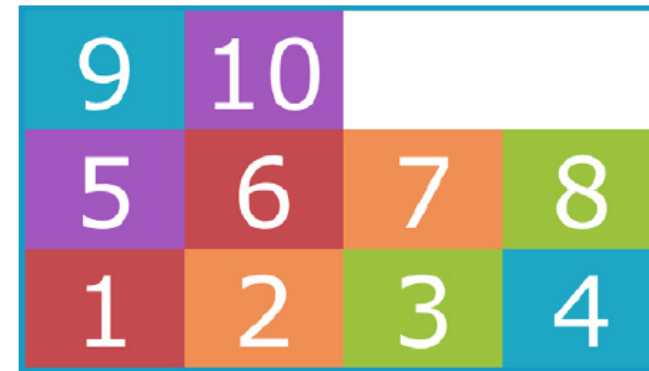
Le retour à la ligne : flex-wrap

- **rôle** : définir la façon d'affichage des éléments (une seule ligne ou plusieurs lignes)
- **valeurs** : nowrap | wrap | wrap-reverse
- **valeur par défaut** : nowrap
- **appliqué aux** : éléments du conteneur flex

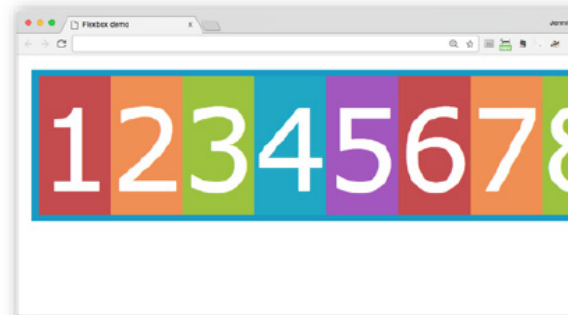
flex-wrap: wrap;



flex-wrap: wrap-reverse;



flex-wrap: nowrap; (default)



Alignement selon l'axe principale : justify-content

- **rôle** : définir la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe principale
- **valeurs** : flex-start | flex-end | center | space-between | space-around
- **valeur par défaut** : flex-start
- **appliqué aux** : éléments du conteneur flex

justify-content: flex-start; (default)



justify-content: flex-end;



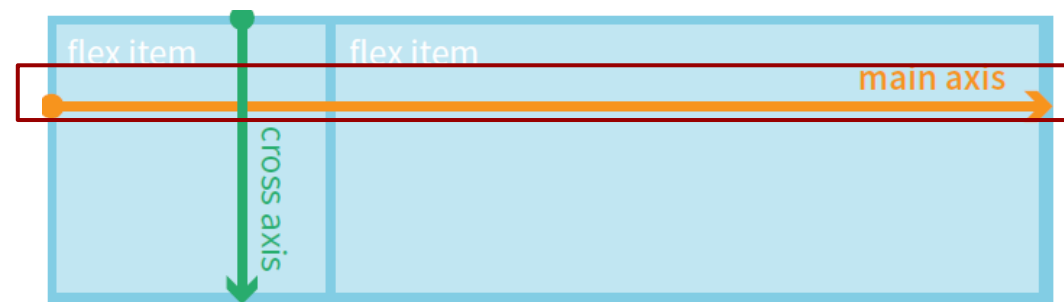
justify-content: space-between;



justify-content: center;

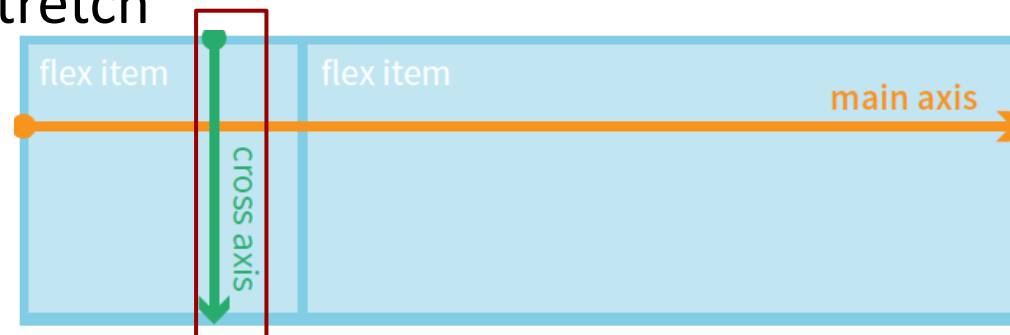


justify-content: space-around;

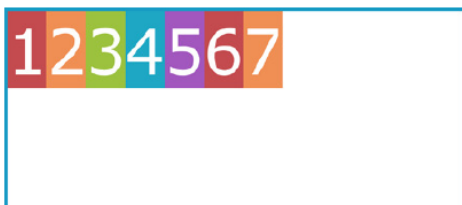


Alignement selon l'axe secondaire : align-items

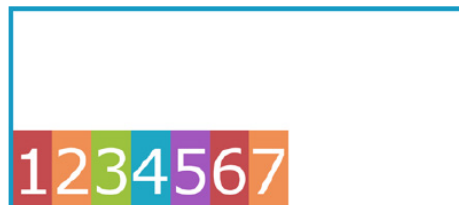
- **rôle** : définir la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe secondaire
- **valeurs** : flex-start | flex-end | center | baseline | stretch
- **valeur par défaut** : stretch
- **appliqué aux** : éléments du conteneur flex



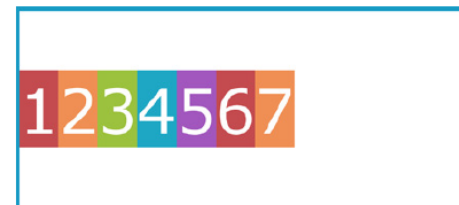
`align-items: flex-start;`



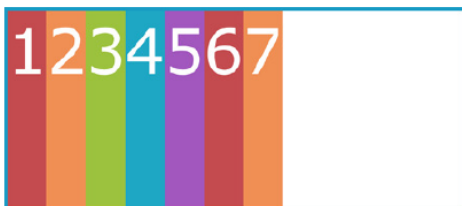
`align-items: flex-end;`



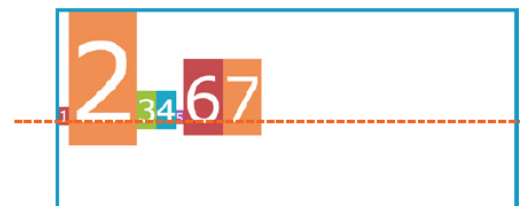
`align-items: center;`



`align-items: stretch;` (default)

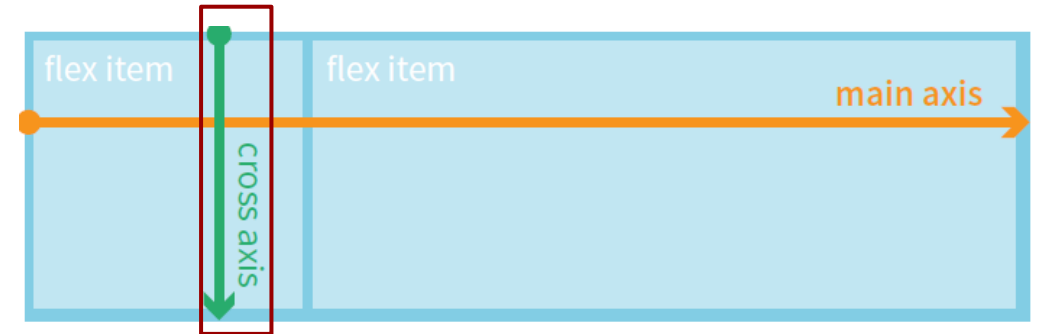


`align-items: baseline;`



Alignement d'un seul élément : align-self

- **rôle** : définir la façon dont l'espace doit être réparti pour un élément selon l'axe secondaire
- **valeurs** : flex-start | flex-end | center | baseline | stretch
- **valeur par défaut** : stretch
- **appliqué aux** : éléments du conteneur flex



```
.box4 {  
  align-self: flex-end;  
}
```

align-self: flex-end;



Expansion d'un élément flex : flex-grow

- **rôle** : définir le facteur d'expansion d'un élément
- **valeurs** : nombre
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

flex-grow : 0



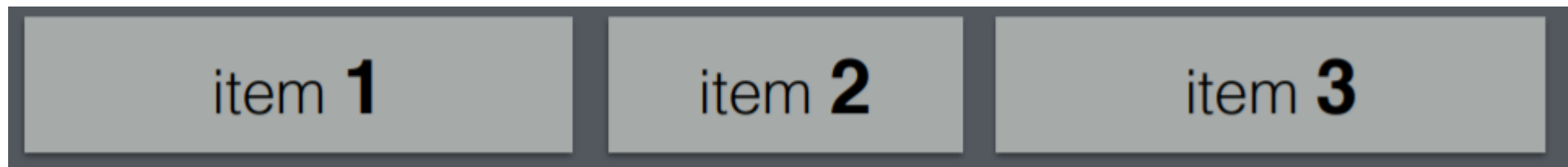
flex-grow : 1



Compression d'un élément flex : flex-shrink

- **rôle** : définir le facteur de compression d'un élément
- **valeurs** : nombre
- **valeur par défaut** : 1
- **appliqué à** : un élément flex

```
#item1, #item3 {  
    display: flex;  
    flex-shrink: 2;  
}
```



Taille initiale d'un élément flex : flex-basis

- **rôle** : définir la taille initiale d'un élément flex
- **valeurs** : length | percentage | content | auto
- **valeur par défaut** : auto
- **appliqué à** : un élément flex



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box3 {  
  order: 1;  
}
```



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box2, .box3 {  
  order: 1  
}
```



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box5 {  
  order: -1  
}
```





Bootstrap 4 utilise flexbox

[bootstrap](#) / dist / css / bootstrap-grid.css

```
.row {  
  display: flex;  
  flex-wrap: wrap;  
  margin-right: -15px;  
  margin-left: -15px;  
}
```

```
.col {  
  flex-basis: 0;  
  flex-grow: 1;  
  max-width: 100%;  
}
```

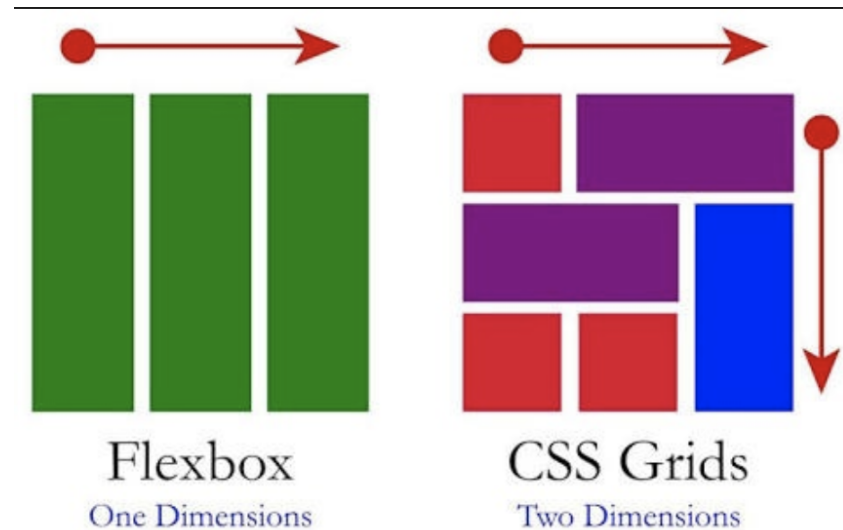
Source : <https://github.com/twbs/bootstrap/blob/master/dist/css/bootstrap-grid.css>

Grid



Définition

- **Grid** est un module pour la présentation des composants d'une page web
- **Principe :**
 - Définition d'un conteneur
 - Insertion de plusieurs éléments en lignes et en colonnes du conteneur

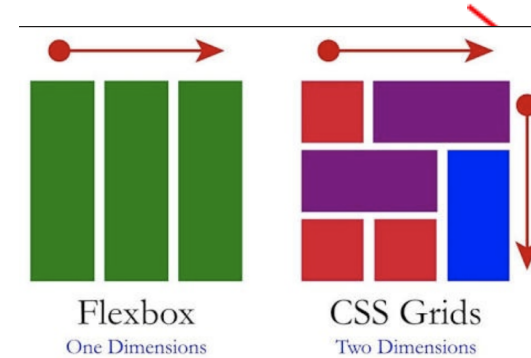


Définition

- **Grid** est un module pour la présentation des composants d'une page web

- **Principe :**

- Définition d'un conteneur
- Insertion de plusieurs éléments en lignes et en colonnes du conteneur

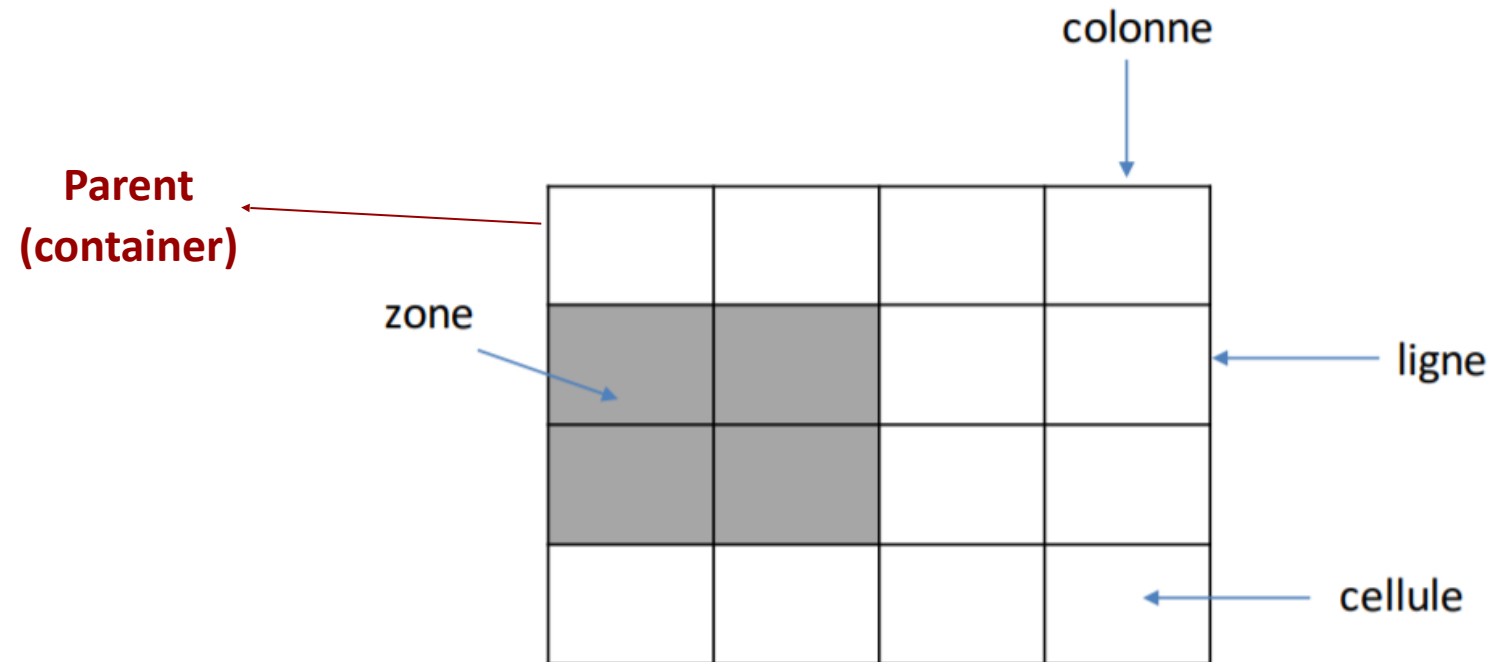


- **Déclaration :**

```
#layout {  
    display: grid;  
}
```

```
<div id="layout">  
  <div id="one">One</div>  
  <div id="two">Two</div>  
  <div id="three">Three</div>  
  <div id="four">Four</div>  
  <div id="five">Five</div>  
</div>
```

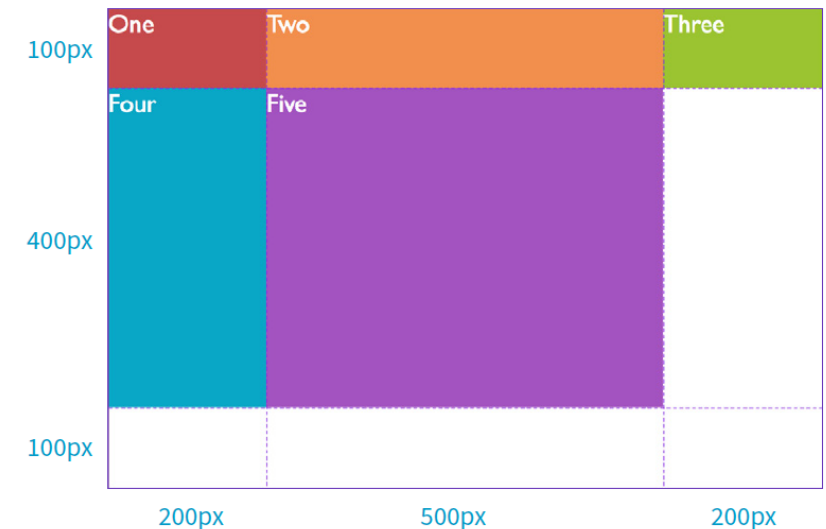
Terminologie



Définition des lignes et colonnes

- **grid-template-rows :**
 - **rôle** : définir le nombre et la taille de chaque ligne
 - **valeurs** : none | taille
 - **valeur par défaut** : none
 - **appliqué aux** : conteneurs grid
- **grid-template-columns :**
 - **rôle** : définir le nombre et la taille de chaque colonne
 - **valeurs** : none | taille
 - **valeur par défaut** : none
 - **appliqué aux** : conteneurs grid

```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}
```



Définition des lignes et colonnes

■ grid-template-rows :

- **rôle** : définir le nombre et la taille de chaque ligne
- **valeurs** : none | taille
- **valeur par défaut** : none
- **appliqué aux** : conteneurs grid

```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 1fr 200px;  
}
```

```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}
```

fractional unit

■ grid-template-columns :

- **rôle** : définir le nombre et la taille de chaque colonne
- **valeurs** : none | taille
- **valeur par défaut** : none
- **appliqué aux** : conteneurs grid



Emplacement des grilles : utilisation des lignes et des colonnes

grid-row-start

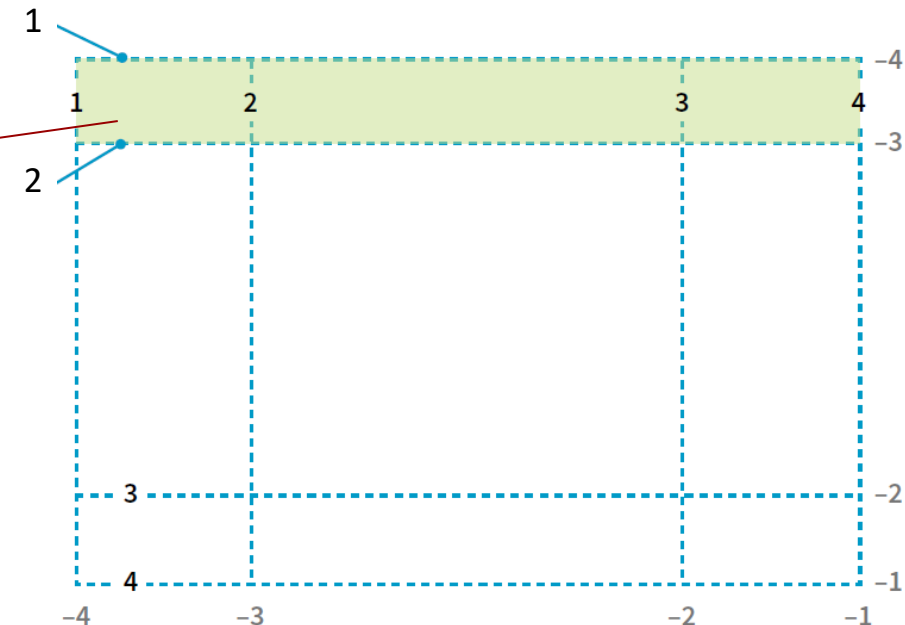
grid-row-end

grid-column-start

grid-column-end

- **rôle** : définir le début et la fin d'un élément en utilisant les lignes et les colonnes
- **valeurs** : auto | grid line | span number | span 'line name' | number 'line name'
- **valeur par défaut** : auto
- **appliqué aux** : éléments grid

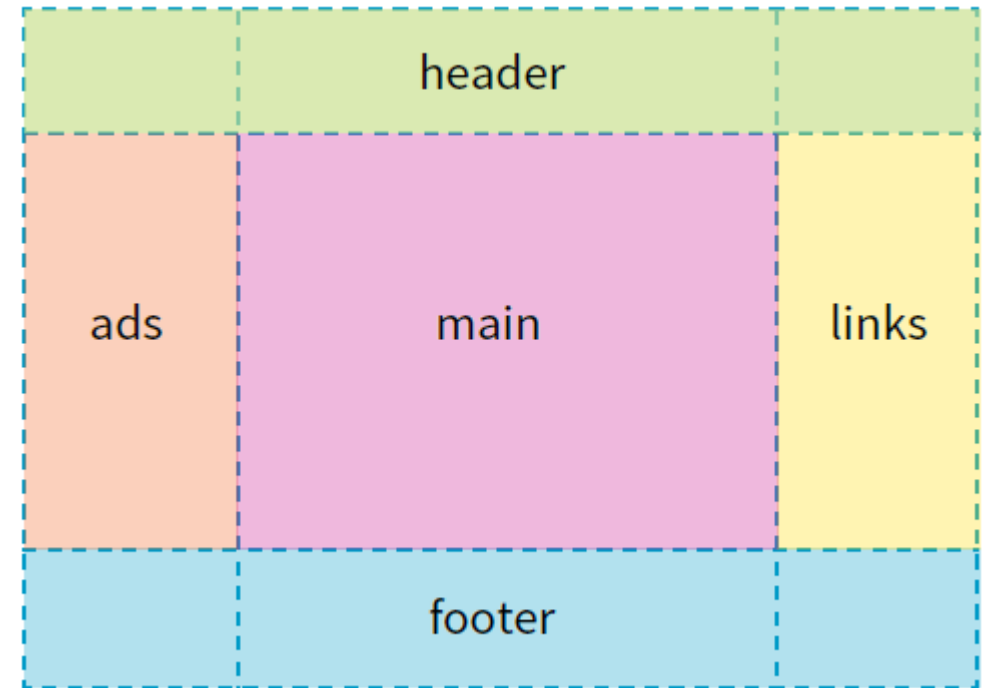
```
#one {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```



Affectation des noms aux grilles : grid-template-area

- **rôle** : définir des noms aux grilles
- **valeurs** : none | noms
- **valeur par défaut** : none
- **appliqué aux** : éléments grid

```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px  
  200px;  
  grid-template-areas:  
    "header header header"  
    "ads main links"  
    "footer footer footer";  
}
```

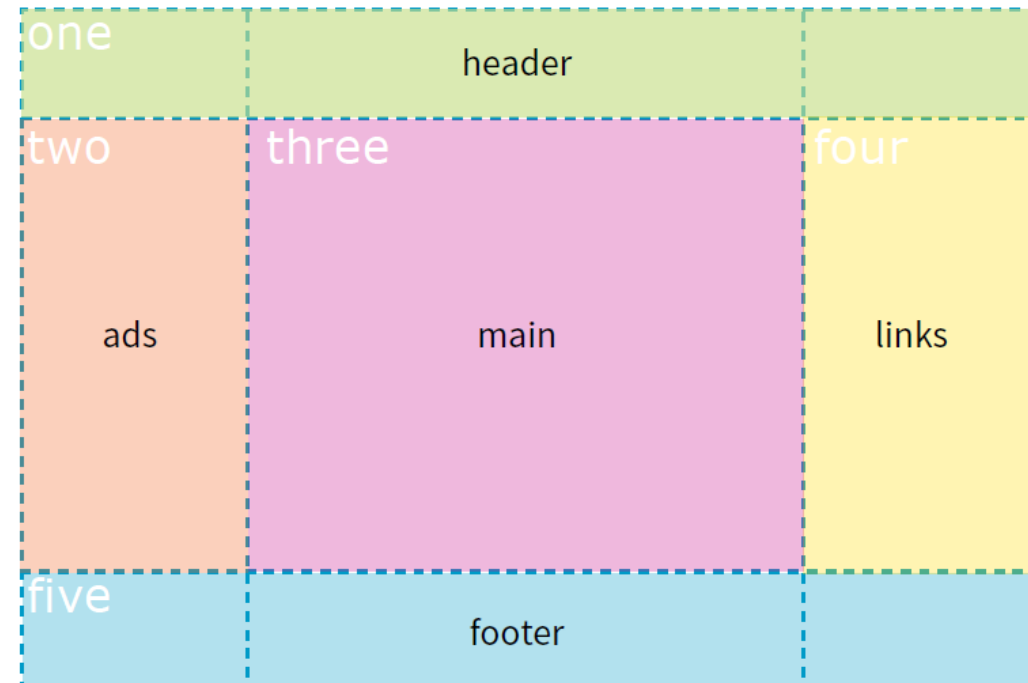


Emplacement des grilles : utilisation des noms des grilles

grid-area

- **rôle** : définir le début et la fin d'un élément en utilisant les noms des grilles
- **valeurs** : nom de la grille
- **valeur par défaut** : -
- **appliqué aux** : éléments grid

```
#one { grid-area: header; }  
#two { grid-area: ads; }  
#three { grid-area: main }  
#four { grid-area: links; }  
#five { grid-area: footer; }
```

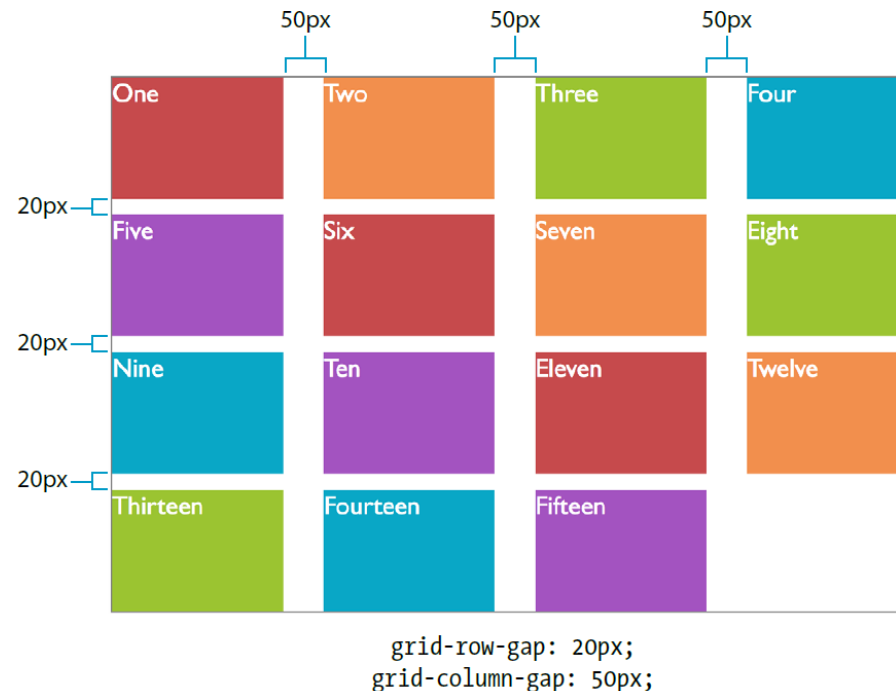


Espaces entre les lignes et les colonnes

grid-row-gap

grid-column-gap

- **rôle** : définir les espaces entre les lignes et entre les colonnes
- **valeurs** : longueur positive
- **valeur par défaut** : 0
- **appliqué au** : conteneur grid



Responsive Web Design



Comment rendre un site responsive ?

- Images flexibles ✓
- Bootstrap ✓
- CSS flexbox ✓
- CSS grid ✓
- **Configuration du viewport**
- **Media queries**



Configuration du viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Partie visible
d'une page web

largeur de la page
=
largeur de l'appareil

Niveau de zoom



- **Définition** : Media Queries (= Requêtes média) sont des styles CSS conditionnels qui permettent de définir différents styles CSS pour différents types d'appareils media et selon différents critères
- **Syntaxe** :

type de média (all, screen, ...)

caractéristique et valeur du média

```
@media not|only mediatype and (mediafeature and|or|not  
mediafeature) {  
    /* Code CSS */  
}
```

■ Exemple :

```
body {  
  background-color: tan;  
}  
  
@media screen and (max-width: 992px) {  
  body {  
    background-color: blue;  
  }  
}  
  
@media screen and (max-width: 600px) {  
  body {  
    background-color: olive;  
  }  
}
```

