




# CSS

Seconde partie




## ● CSS, mise en page : Compléments sur la Flexbox, la propriété align-self (1)

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "flexboxextra.html" (avec le HTML minimum) et "flexboxextra.css"
2. Modifiez votre fichier "flexboxextra.html" pour qu'il se voit appliquer les styles de "flexboxextra.css"
3. Dans "flexboxextra.css":
  - 3.1. Créez une règle pour la classe "container" qui lui appliquera les propriétés suivantes:
    - 3.1.1. display: flex
    - 3.1.2. background-color: lightgreen
    - 3.1.3. height: 150px
    - 3.1.4. width: 50vw
    - 3.1.5. align-items: center
    - 3.1.6. margin-bottom: 2vh
    - 3.1.7. justify-content: space-around;
  - 3.2. Créez une règle pour la classe "child" qui lui appliquera les propriétés suivantes:
    - 3.2.1. background-color: blueviolet
    - 3.2.2. height: 100px
    - 3.2.3. width: 100px
    - 3.2.4. border: solid red 1px



## ● CSS, mise en page : Compléments sur la Flexbox, la propriété align-self (2)

4. Dans "flexboxextra.html":
  - 4.1. Créez une balise <div> avec la classe "container"
  - 4.2. Dans cette balise, créez 3 balises <div> avec la classe "child"
5. Affichez la page dans votre navigateur. Les 3 carrés violets doivent être alignés horizontalement
6. Maintenant, appliquez la propriété css "align-self":
  - 6.1. au premier carré violet, avec la valeur "flex-start"
  - 6.2. au troisième carré violet, avec la valeur "flex-end"
7. Rafraîchissez la page. Les carrés violets doivent être maintenant décalés sur l'axe horizontal




# CSS, mise en page : Compléments sur la Flexbox, la propriété align-self

- La propriété "align-self" sert modifier le comportement d'un enfant par rapport à la propriété "align-items" définie par le parent
- Valeurs possibles:
  - flex-start
  - flex-end
  - center
  - stretch



# ● CSS, mise en page : Compléments sur la Flexbox, la propriété flex-wrap

1. Dans "flexboxextra.css":
  - 1.1. Créez une règle pour la classe "child2" qui lui appliquera les propriétés suivantes:
    - 1.1.1. border: solid red 1px
    - 1.1.2. background-color: yellow
    - 1.1.3. height: 80%
    - 1.1.4. width: 300px
2. Dans "flexboxextra.html":
  - 2.1. Créez une nouvelle balise <div> avec la classe "container"
  - 2.2. Dans cette balise, créez 3 balises <div> avec la classe "child2"
3. Affichez la page dans votre navigateur. Les 3 carrés jaunes ne doivent pas déborder du container vert
4. À présent, dans "flexboxextra.css", modifiez la règle pour l'identifiant "container" en ajoutant la propriété "flex-wrap" avec la valeur "wrap"
5. Rafraichissez la page dans le navigateur. Le 3ème carré jaune doit être renvoyé à la ligne. Notez que les carrés jaunes ont maintenant la taille prévue par leurs propriétés css alors qu'ils étaient compressés à l'étape 3



# CSS, mise en page : Compléments sur la Flexbox, la propriété flex-wrap

- La propriété "flex-wrap" sert à définir si les enfants dans une flexbox peuvent retourner à la ligne s'ils ne peuvent pas contenir dans le parent
- Valeurs possibles:
  - nowrap (valeur implicite par défaut)
  - wrap (retour à la ligne autorisé)
  - wrap-reverse (retour à la ligne dans le sens inverse)



# CSS, mise en page : la Grille

- Grid: modèle de disposition en grille
- C'est une autre manière de mettre en page les documents HTML
- Contrairement à la flexbox qui est plutôt uni-dimensionnelle, la grille est bi-dimensionnelle
- Les Flexbox et les Grid peuvent néanmoins être utilisées de concert
- Pour activer la grille d'une balise, il faut lui donner la propriété "display" avec la valeur "grid". Cette balise devient le container de grille
- Les balises enfants vont s'organiser selon le modèle Grid. On peut les appeler éléments de grille
- On introduit une nouvelle unité: "fr", qui représente une fraction de l'espace disponible dans le container de grille



# CSS, la Grille : propriétés du container


Propriété (container)	Description	Ex. de valeurs
grid-template-columns	Définit les colonnes de la grille	1fr 2fr 1fr, 300px 200px 300px, repeat(3 , 1fr), repeat(auto-fill, minmax(100px, 1fr)
gap	Espace entre les lignes et colonnes	20px, 1em
row-gap column-gap	Espace entre les lignes Espace entre les colonnes	30px, 1em, 1vh 25px, 1em, 2vw
grid-auto-rows	Taille des lignes automatiques	100px minmax(100px, auto)





# CSS, la Grille : propriétés du container

- `minmax(min, max)`: permet de spécifier une taille minimum et une taille maximum
- `minmax(100px, auto)`: permet de spécifier une taille variable s'adaptant au contenu grâce au mot clé `auto` pour éviter les débordements
- `repeat(nb, taille)`: permet de répéter *nb* fois une *taille* donnée. On peut utiliser le mot clé "auto-fill" pour *nb* pour répéter tant qu'il y a de la place
- On peut combiner `repeat` et `minmax`:
  - `grid-template-columns: repeat(auto-fill, minmax(200px, 1fr))` : permet de mettre autant de colonne avec une taille minimum de 200px et une taille maximum de 1fr



# CSS, la Grille : propriétés des éléments de grille

Propriété (élément)	Description	Ex. de valeurs
grid-column	Colonne de début et de fin de l'élément	2, 1 / 3 span 2 4 / span 2
grid-row	Ligne de début et de fin de l'élément	2, 1 / 4 span 3 4 / span 2



# ● CSS : la Grille (1)

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "simplegrid.html" (avec le HTML minimum) et "simplegrid.css"
2. Modifiez votre fichier "simplegrid.html" pour qu'il se voit appliquer les styles de "simplegrid.css"
3. Générez une grille (<div>) qui contient 15 balises <div> numérotées de 1 à 15, avec fond violet et bordure rouge:
  - 3.1. la grille doit comporter 5 colonnes de taille égale
  - 3.2. la taille de chaque ligne doit être au moins de 10vh
  - 3.3. l'espacement entre les lignes et les colonnes doit être de 1vh
4. (résultat à obtenir visible page suivante)

(Vous n'avez pas besoin des propriétés grid-column et grid-row pour cet exercice)



# ● CSS : la Grille (1)

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15



## ● CSS : la Grille (2)

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "grid.html" (avec le HTML minimum) et "grid.css"
2. Modifiez votre fichier "grid.html" pour qu'il se voit appliquer les styles de "grid.css"
3. Créez, au sein d'une balise <div> qui sera votre conteneur de grille:
  - 3.1. Un titre <h1> (élément de grille)
  - 3.2. Une balise <div> (élément de grille) qui contient:
    - 3.2.1. un titre <h2>
    - 3.2.2. un paragraphe <p> de plusieurs lignes
  - 3.3. Une seconde balise <div> (élément de grille) qui contient:
    - 3.3.1. un titre <h2>
    - 3.3.2. un paragraphe <p> de plusieurs lignes
    - 3.3.3. un second paragraphe <p> de plusieurs lignes
  - 3.4. Un titre <h3> (élément de grille) qui fera office de pied de page
4. Utilisez une grille pour mettre en page ces balises (résultat à obtenir visible page suivante, vous aurez besoin des propriétés grid-column et grid-row pour cet exercice)



# ● CSS : la Grille (2)

Résultat à obtenir:

## Gros titre

### Article 1

Duis vestibulum elit finibus nulla consequat, vel hendrerit odio mattis. Quisque vitae mauris lacus. Nulla gravida lacus sit amet velit venenatis, auctor luctus odio facilisis. Aliquam sem erat, luctus lacinia blandit vel, hendrerit eu leo. Nam accumsan metus non lacus ultrices congue. Proin nec risus eget nunc vulputate porttitor. Etiam ut neque semper felis consequat facilisis id id turpis. Nulla iaculis tincidunt velit non scelerisque. Donec ac felis eu lacus rhoncus dictum. Vestibulum scelerisque ac odio eu condimentum. In feugiat ante tellus, sed semper nisi tincidunt non. In vehicula vitae nisl quis consectetur. Maecenas scelerisque est at quam vehicula, ut accumsan justo tempor. Aenean pretium orci rhoncus, interdum massa ultricies, aliquam augue.

### Article 2

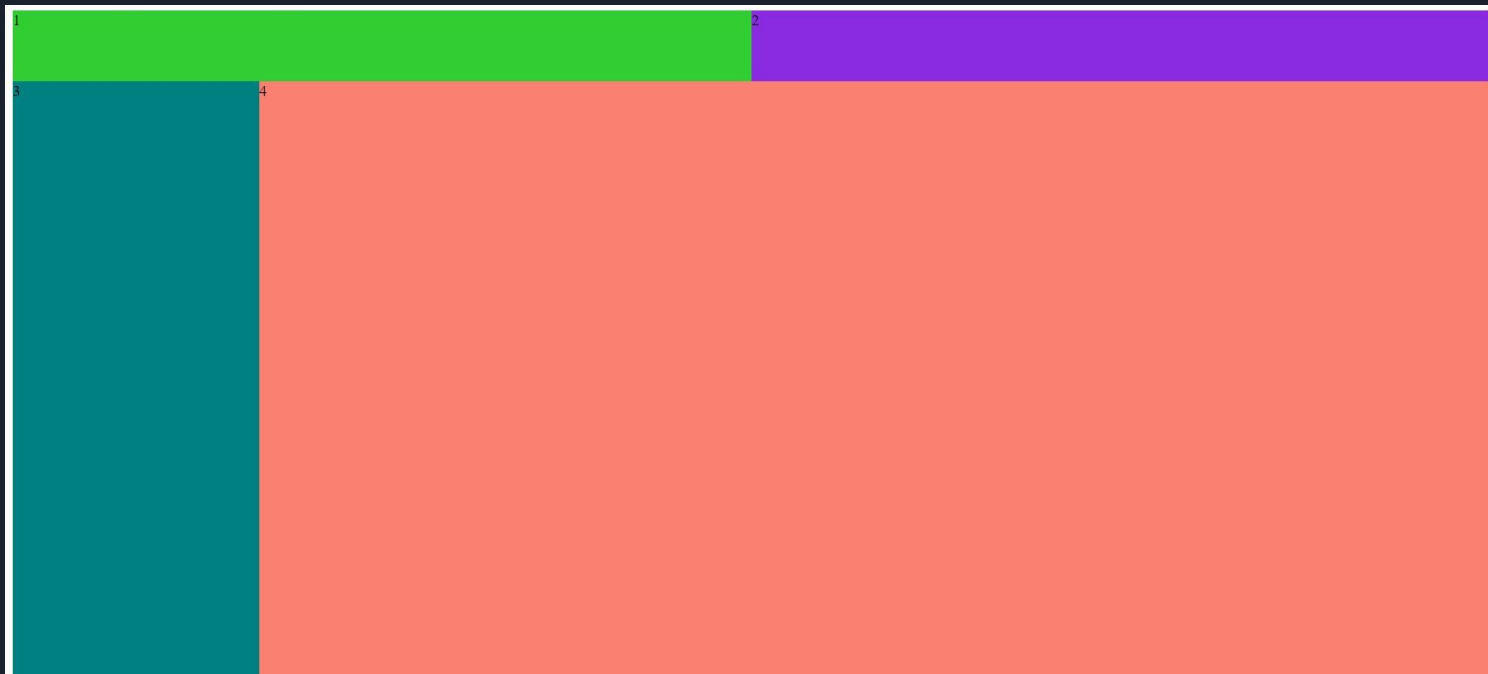
Duis vestibulum elit finibus nulla consequat, vel hendrerit odio mattis. Quisque vitae mauris lacus. Nulla gravida lacus sit amet velit venenatis, auctor luctus odio facilisis. Aliquam sem erat, luctus lacinia blandit vel, hendrerit eu leo. Nam accumsan metus non lacus ultrices congue. Proin nec risus eget nunc vulputate porttitor. Etiam ut neque semper felis consequat facilisis id id turpis. Nulla iaculis tincidunt velit non scelerisque. Donec ac felis eu lacus rhoncus dictum. Vestibulum scelerisque ac odio eu condimentum. In feugiat ante tellus, sed semper nisi tincidunt non. In vehicula vitae nisl quis consectetur. Maecenas scelerisque est at quam vehicula, ut accumsan justo tempor. Aenean pretium orci rhoncus, interdum massa ultricies, aliquam augue.

Duis vestibulum elit finibus nulla consequat, vel hendrerit odio mattis. Quisque vitae mauris lacus. Nulla gravida lacus sit amet velit venenatis, auctor luctus odio facilisis. Aliquam sem erat, luctus lacinia blandit vel, hendrerit eu leo. Nam accumsan metus non lacus ultrices congue. Proin nec risus eget nunc vulputate porttitor. Etiam ut neque semper felis consequat facilisis id id turpis. Nulla iaculis tincidunt velit non scelerisque. Donec ac felis eu lacus rhoncus dictum. Vestibulum scelerisque ac odio eu condimentum. In feugiat ante tellus, sed semper nisi tincidunt non. In vehicula vitae nisl quis consectetur. Maecenas scelerisque est at quam vehicula, ut accumsan justo tempor. Aenean pretium orci rhoncus, interdum massa ultricies, aliquam augue.

### Pied de page

## ● CSS : la Grille (3)

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "divgrid.html" (avec le HTML minimum) et "divgrid.css"
2. Modifiez votre fichier "divgrid.html" pour qu'il se voit appliquer les styles de "divgrid.css"
3. Produisez le résultat suivant (page suivante pour quelques indices):





## ● CSS : la Grille (3)

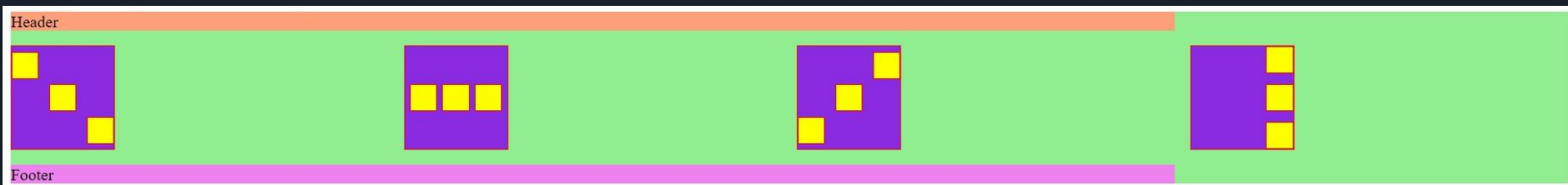
Instructions:

- On peut tout faire avec des balises <div>
- On utilise une grille à 12 colonnes de tailles égales
- On définit la taille des lignes à 10vh
- Les couleurs importent peu, faites juste en sorte qu'on puisse les distinguer



# ● CSS : la Grille (4) : mélange Grille / Flexbox

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "flexgrid.html" (avec le HTML minimum) et "flexgrid.css"
2. Modifiez votre fichier "flexgrid.html" pour qu'il se voit appliquer les styles de "flexgrid.css"
3. Produisez le résultat suivant:



- Instructions:
  - On peut tout faire avec des balises <div>
  - On utilise une grille à 4 colonnes de tailles égales pour placer le header, le footer et les 4 carrés violets (le container vert étant la grille)
  - Chaque carré violet est un container type flexbox, les 3 carrés jaunes sont donc disposés en utilisant le modèle des Flexbox
  - Les couleurs importent peu, faites juste en sorte qu'on puisse les distinguer

# CSS : la Grille (5)

1. Dans votre répertoire de travail, créez deux nouveaux fichiers "finalgrid.html" (avec le HTML minimum) et "finalgrid.css", qui s'applique à "finalgrid.html"
2. Créez la structure HTML suivante dans "finalgrid.html" (allez voir la slide suivante pour les contraintes)

## Gros titre

### Sous titre

#### Lorem ipsum 1

Nulla tempor est nec turpis varius maximus. Nam nec dictum ante, a ornare turpis. Nullam ultricies tortor ex, rhoncus molestie libero ullamcorper eget. Etiam quis pellentesque neque, et ultricies erat. Suspendisse pellentesque faucibus diam. Aenean ac tempor velit. Mauris feugiat iaculis nulla, at interdum arcu vestibulum ut. Mauris enim est, gravida ut malesuada sed, cursus at ante. Proin tincidunt dapibus odio. Morbi congue libero purus, vitae porttitor metus placerat vitae. Vestibulum imperdiet at lorem vel scelerisque. Donec porttitor ipsum vel hendrerit ornare. Praesent vitae felis eget nunc aliquet aliquet sit amet eget nulla. Quisque elit dui, venenatis id dolor et, rhoncus posuere diam. Quisque ac viverra lectus, et fermentum tellus. Praesent ornare ornare dui, eu pulvinar dolor vestibulum id.

#### Lorem ipsum 2

Nulla tempor est nec turpis varius maximus. Nam nec dictum ante, a ornare turpis. Nullam ultricies tortor ex, rhoncus molestie libero ullamcorper eget. Etiam quis pellentesque neque, et ultricies erat. Suspendisse pellentesque faucibus diam. Aenean ac tempor velit. Mauris feugiat iaculis nulla, at interdum arcu vestibulum ut. Mauris enim est, gravida ut malesuada sed, cursus at ante. Proin tincidunt dapibus odio. Morbi congue libero purus, vitae porttitor metus placerat vitae. Vestibulum imperdiet at lorem vel scelerisque. Donec porttitor ipsum vel hendrerit ornare. Praesent vitae felis eget nunc aliquet aliquet sit amet eget nulla. Quisque elit dui, venenatis id dolor et, rhoncus posuere diam. Quisque ac viverra lectus, et fermentum tellus. Praesent ornare ornare dui, eu pulvinar dolor vestibulum id.

#### Lorem ipsum 3

A turpis eleifend, eget Nulla tempor est nec turpis varius maximus. Nam nec dictum ante, a ornare turpis. Nullam ultricies tortor ex, rhoncus molestie libero ullamcorper eget. Etiam quis pellentesque neque, et ultricies erat. Suspendisse pellentesque faucibus diam. Aenean ac tempor velit. Mauris feugiat iaculis nulla, at interdum arcu vestibulum ut. Mauris enim est, gravida ut malesuada sed, cursus at ante. Proin tincidunt dapibus odio. Morbi congue libero purus, vitae porttitor metus placerat vitae. Vestibulum imperdiet at lorem vel scelerisque. Donec porttitor ipsum vel hendrerit ornare. Praesent vitae felis eget nunc aliquet aliquet sit amet eget nulla. Quisque elit dui, venenatis id dolor et, rhoncus posuere diam. Quisque ac viverra lectus, et fermentum tellus. Praesent ornare ornare dui, eu pulvinar dolor vestibulum id.

#### Lorem ipsum 4

Etiam interdum laoreet diam at auctor. Sed Nulla tempor est nec turpis varius maximus. Nam nec dictum ante, a ornare turpis. Nullam ultricies tortor ex, rhoncus molestie libero ullamcorper eget. Etiam quis pellentesque neque, et ultricies erat. Suspendisse pellentesque faucibus diam. Aenean ac tempor velit. Mauris feugiat iaculis nulla, at interdum arcu vestibulum ut. Mauris enim est, gravida ut malesuada sed, cursus at ante. Proin tincidunt dapibus odio. Morbi congue libero purus, vitae porttitor metus placerat vitae. Vestibulum imperdiet at lorem vel scelerisque. Donec porttitor ipsum vel hendrerit ornare. Praesent vitae felis eget nunc aliquet aliquet sit amet eget nulla. Quisque elit dui, venenatis id dolor et, rhoncus posuere diam. Quisque ac viverra lectus, et fermentum tellus. Praesent ornare ornare dui, eu pulvinar dolor vestibulum id.

- Menu 1
- Menu 2
- Menu 3
- Menu 4
- Menu 5
- Menu 6
- Menu 7
- Menu 8
- Menu 9



## ● CSS : la Grille (5)

Vous respecterez les contraintes suivantes:

- Comme nous avons beaucoup abusé des classes et identifiants dans les exercices précédents, n'utilisez pas de classe ni d'identifiant
- Le gros titre est un `<h1>`, le sous-titre un `<h2>`, chaque titre de paragraphe un `<h3>`, le pied de page un `<h4>`
- Utilisez une grille
- Pas de flexbox

Quelques suggestions:

- Utilisez une grille à 16 colonnes de taille égale
- Encadrez chaque paragraphe et son titre avec un `<div>`
- Comme vous n'avez pas accès aux classes et identifiants, utilisez par exemple des sélecteurs sur la pseudo-classe `:nth-child()` pour placer vos éléments sur la grille
- Les couleurs choisies importent peu

# CSS : la Grille (5)

## 1. En utilisant une grille, obtenez le résultat suivant:

