

# Flexbox vs CSS Grid

# Deux technologies de mise en page CSS



# BLEND WEB MIX

[BILLETTERIE](#)[Programme](#) ▾ [BlendWebMix ?](#) [Blog](#) [Participants](#) [Job Board](#) [Infos pratiques](#)

24 et 25  
octobre  
2018

**LES DEUX JOURS LES +  
WEB DE L'ANNÉE !**

au Centre des Congrès à Lyon !

**2000 participants attendus  
pour l'édition 2018 !**

De quoi se régaler en networking !

Tech, Design, Business : plus de **70**  
**conférences** à découvrir !

[Découvre le programme !](#)



# Speakers

Deux Freelances, passionnés de CSS

Thomas Zilliox

Expert CSS,  
a écrit un livre sur Flexbox  
aux éditions le train de 13h37



Malik Baba Aïssa

Scrum master,  
travaille tous les jours avec un  
kanban (~ une grosse grille)

# La structure HTML

```
<div class="parent">
    <div class="enfant"></div>
    <div class="enfant"></div>
    <div class="enfant"></div>
    <div class="enfant"></div>
    ...
</div>
```

# 1. Faire des colonnes

# 1. Faire des colonnes

Avant

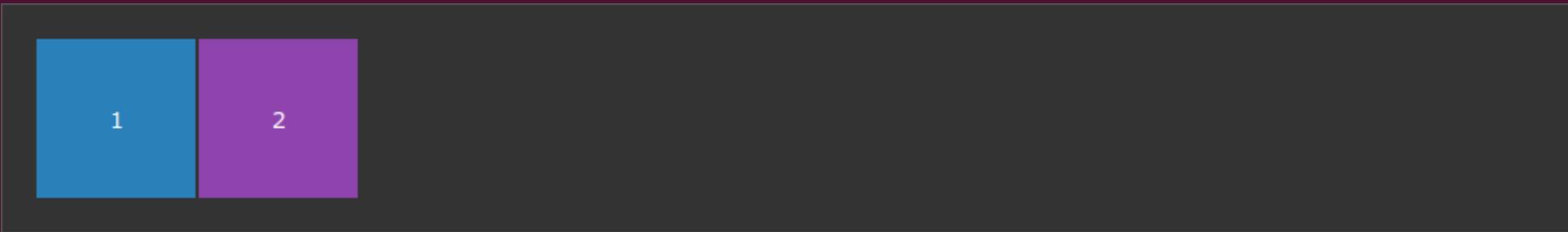
Des specs inadaptés :

- Float
- Inline-block
- ...

On a recours à des outils type Bootstrap.

# 1. Faire des colonnes

## Avec Flex



```
.parent {  
  display: flex;  
}  
  
.enfant {  
  /* Deviennent automatiquement des éléments flexibles */  
  width: 100px;  
}
```

# 1. Faire des colonnes

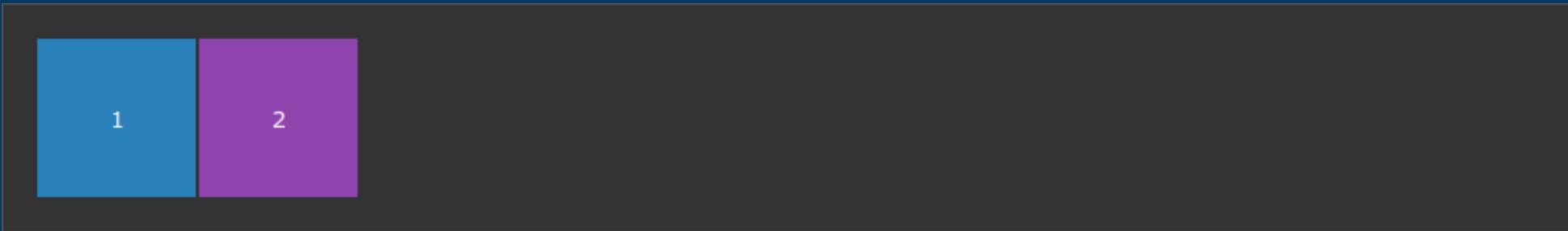
## Avec Flex



```
.parent {  
    display: flex;  
}  
.enfant {  
    width: 100px;  
}  
.enfant:nth-child(5) {  
    width: 300px;  
}
```

# 1. Faire des colonnes

## Avec Grid



```
.parent {  
    display: grid;  
    grid-template-columns: 100px 100px;  
}  
  
.enfant {  
    /* Deviennent automatiquement des éléments cellules */  
}
```

# 1. Faire des colonnes

## Avec Grid



```
.parent {  
    display: grid;  
    grid-template-columns: repeat(4, 100px);  
}  
  
.enfant {  
    /* Deviennent automatiquement des éléments cellules */  
}
```

# 1. Faire des colonnes

## Avec Grid



```
.parent {  
    display: grid;  
    grid-template-columns: repeat(4, 100px) 300px;  
}  
  
.enfant {  
    /* Deviennent automatiquement des éléments cellules */  
}
```

# 1. Faire des colonnes

## Avec Grid



```
.parent {  
    display: grid;  
    grid-template-columns: repeat(4, 100px) 300px;  
    grid-gap: 20px;  
}  
  
.enfant {  
    /* Deviennent automatiquement des éléments cellules */  
}
```

# Fight

Feature

Colonnes

Flex

1

Grid

2

## 2. Alignment

# 2. Alignement

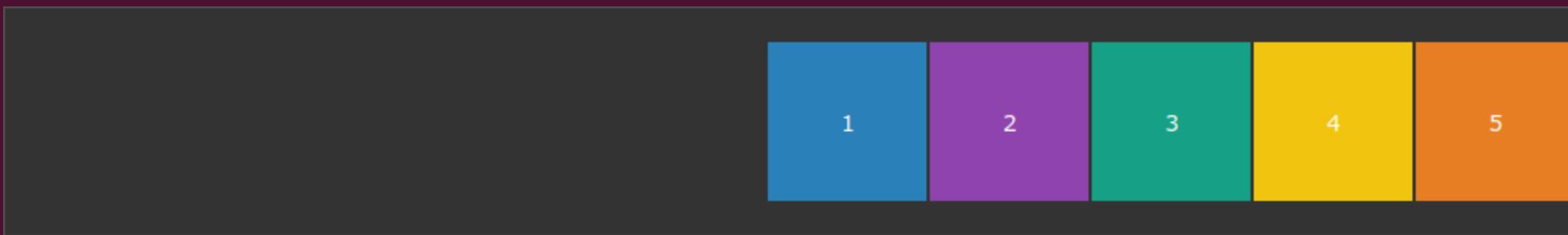
## Avec Flex



```
.parent {  
    justify-content: flex-start;  
}
```

# 2. Alignement

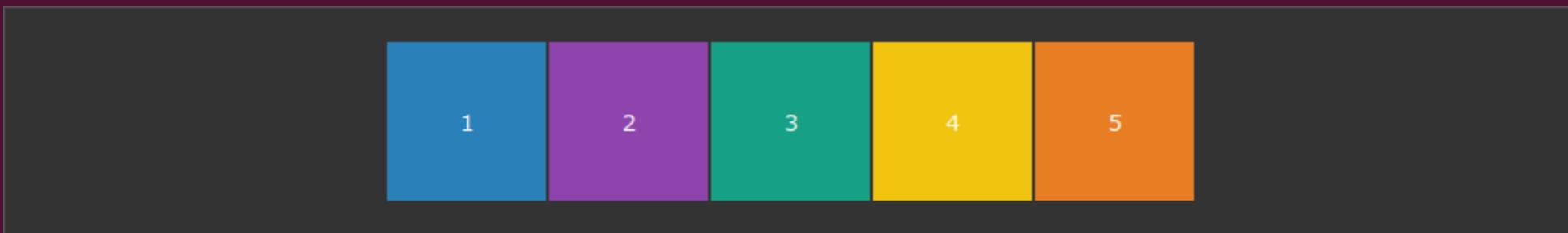
## Avec Flex



```
.parent {  
    justify-content: flex-end;  
}
```

# 2. Alignement

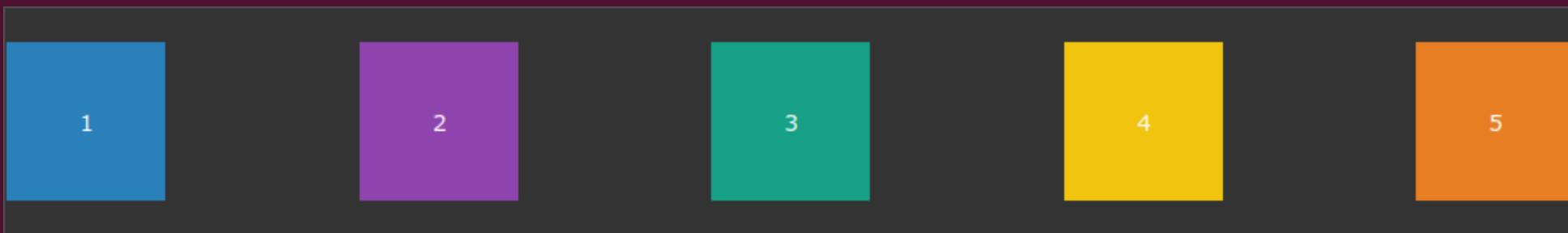
## Avec Flex



```
.parent {  
    justify-content: center;  
}
```

# 2. Alignement

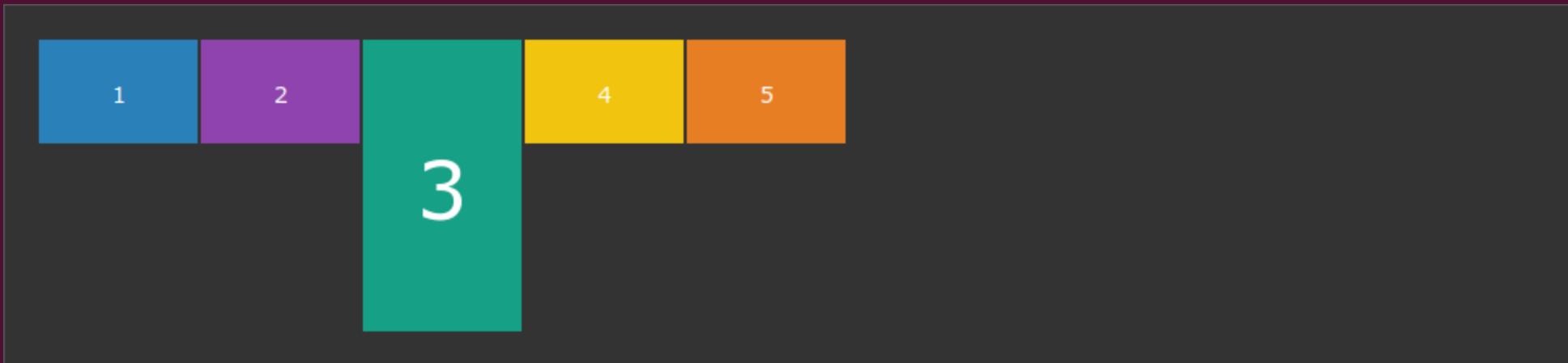
## Avec Flex



```
.parent {  
    justify-content: space-between;  
}
```

# 2. Alignement

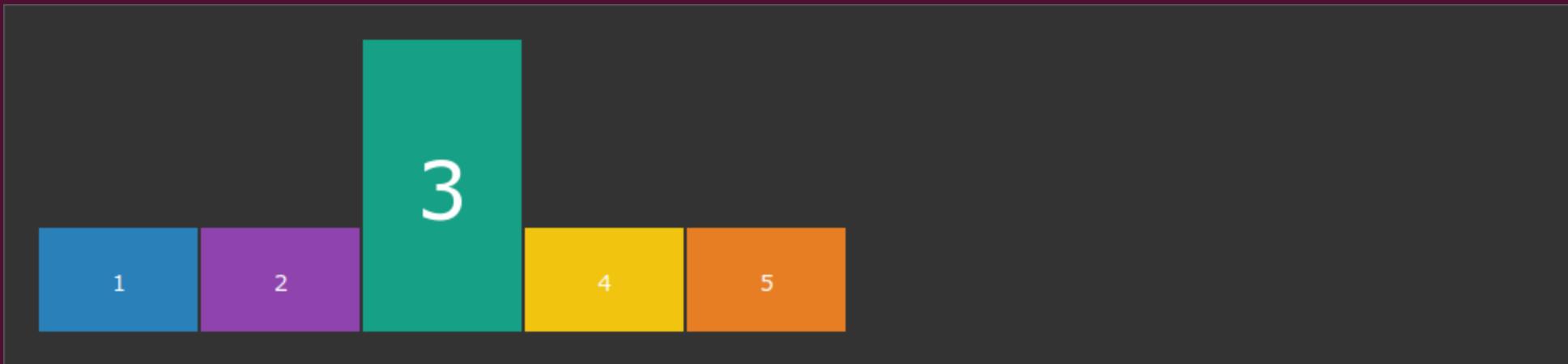
## Avec Flex



```
.parent {  
    align-items: flex-start;  
}
```

# 2. Alignement

## Avec Flex



```
.parent {  
    align-items: flex-end;  
}
```

# 2. Alignement

## Avec Flex



```
.parent {  
    align-items: center;  
}
```

# 2. Alignement

## Avec Flex



```
.parent {  
    align-items: stretch;  
}
```

# 2. Alignement

## Avec Grid

C'est pareil,  
mais sans les préfixes !

# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1

### 3. Elasticité

# 3. Elasticité

## Avec Flex

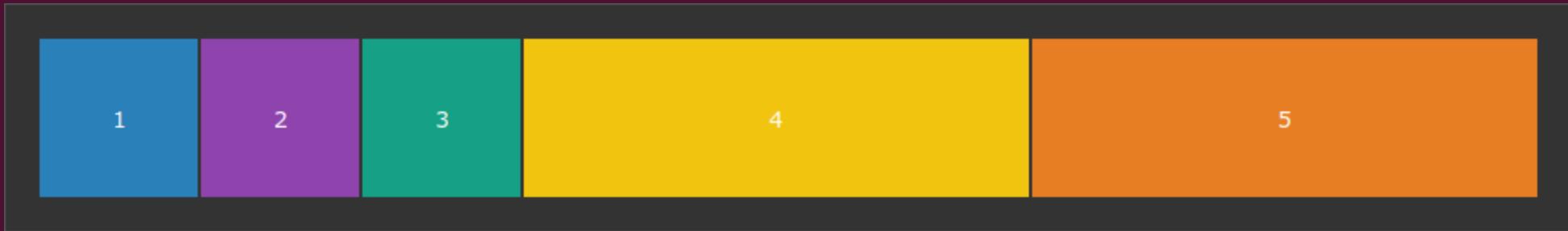


```
.parent { display: flex; }
.enfant { width: 100px }

.enfant:nth-child(4) {
  flex: 1;
}
```

# 3. Elasticité

## Avec Flex

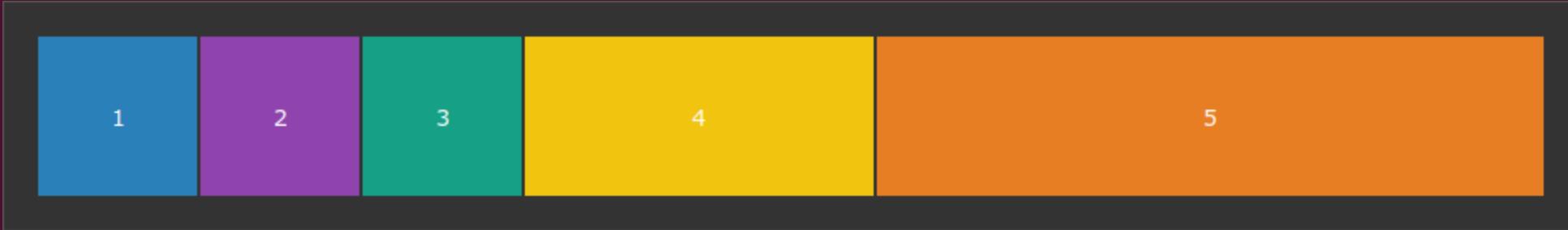


```
.parent { display: flex; }
.enfant { width: 100px }

.enfant:nth-child(4) {
  flex: 1;
}
.enfant:nth-child(5) {
  flex: 1;
}
```

# 3. Elasticité

## Avec Flex



```
.parent { display: flex; }
.enfant { width: 100px }

.enfant:nth-child(4) {
  flex: 1;
}
.enfant:nth-child(5) {
  flex: 2;
}
```

# 3. Elasticité

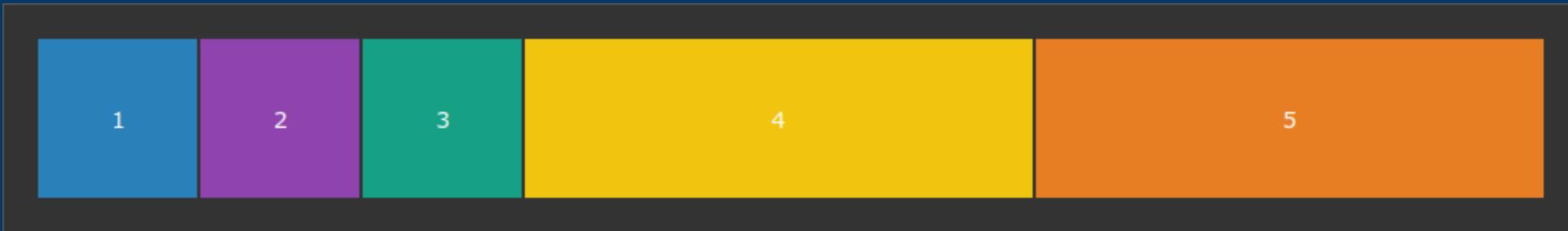
## Avec Grid



```
.parent {  
  display: grid;  
  grid-template-columns: 100px 100px 100px 1fr 100px;  
}  
  
.enfant {  
  /* rien */  
}
```

# 3. Elasticité

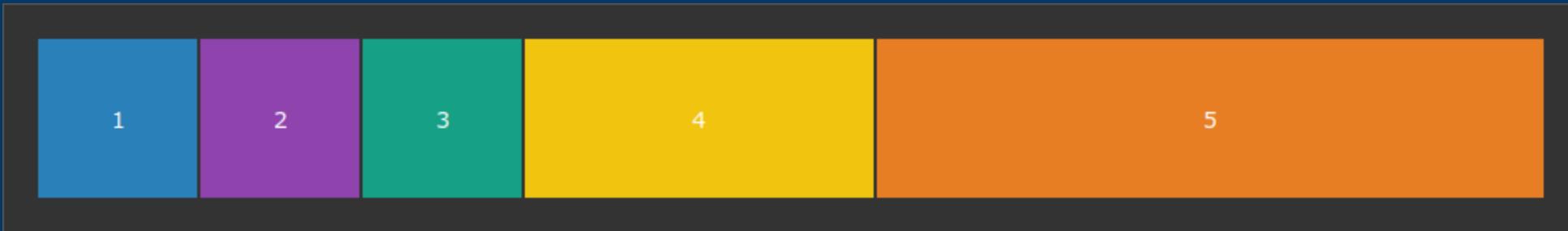
## Avec Grid



```
.parent {  
    display: grid;  
    grid-template-columns: 100px 100px 100px 1fr 1fr;  
}  
  
.enfant {  
    /* rien */  
}
```

# 3. Elasticité

## Avec Grid



```
.parent {
  display: grid;
  grid-template-columns: 100px 100px 100px 1fr 2fr;
}

.enfant {
  /* rien */
}
```

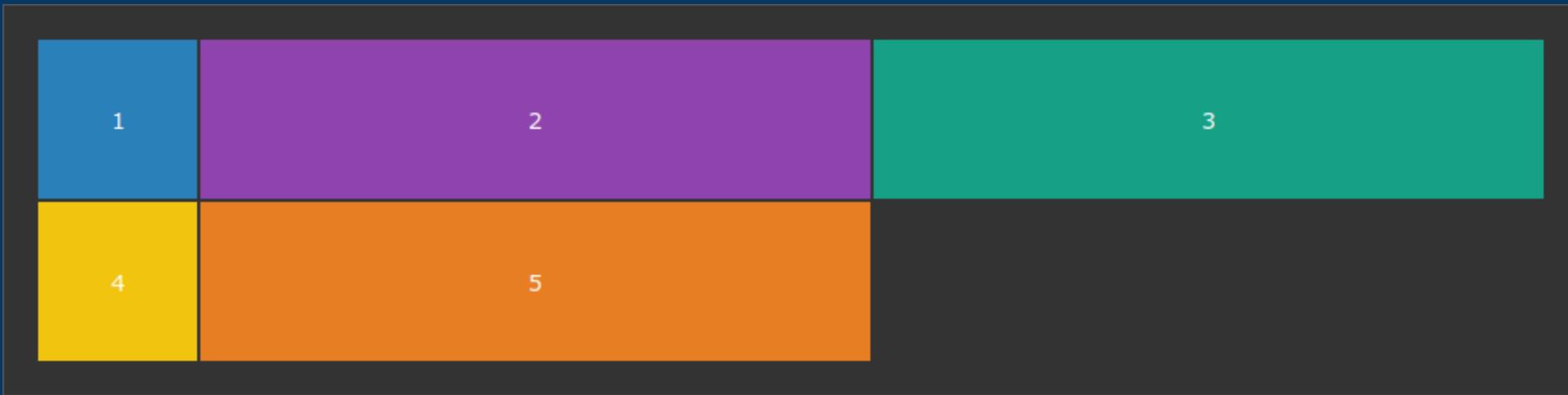
# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1

# 4. Multi-lignes

# 4. Multi-lignes

## Avec Grid



```
.parent {  
  display: grid;  
  grid-template-columns: 100px 1fr 1fr;  
}
```

# 4. Multi-lignes

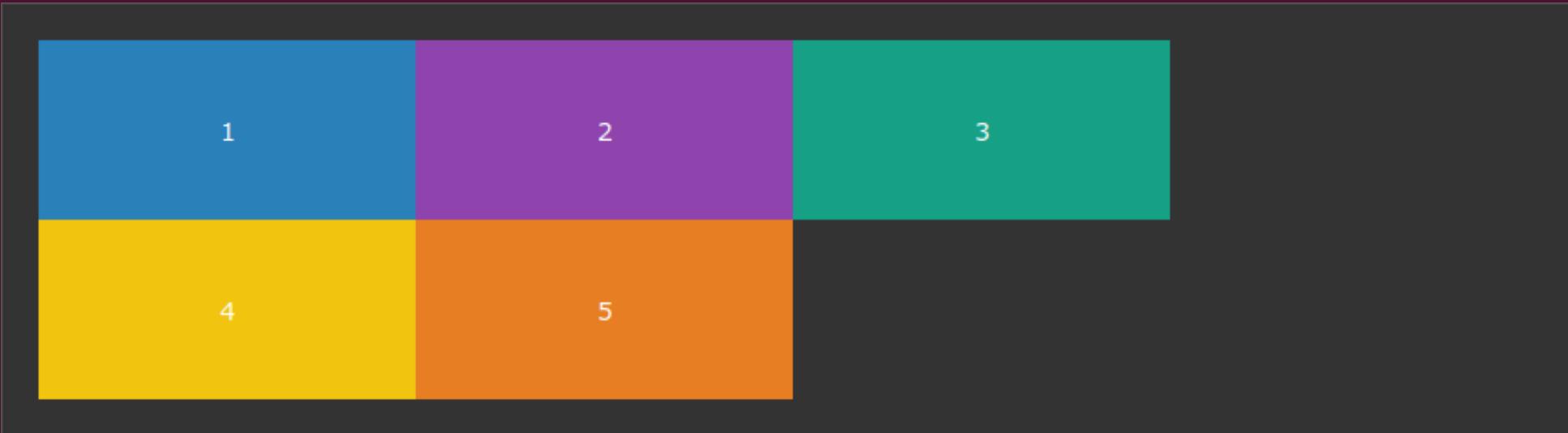
## Avec Grid



```
.parent {  
  display: grid;  
  grid-template-columns: 100px 1fr 1fr;  
  grid-template-rows: 100px 200px;  
}
```

# 4. Multi-lignes

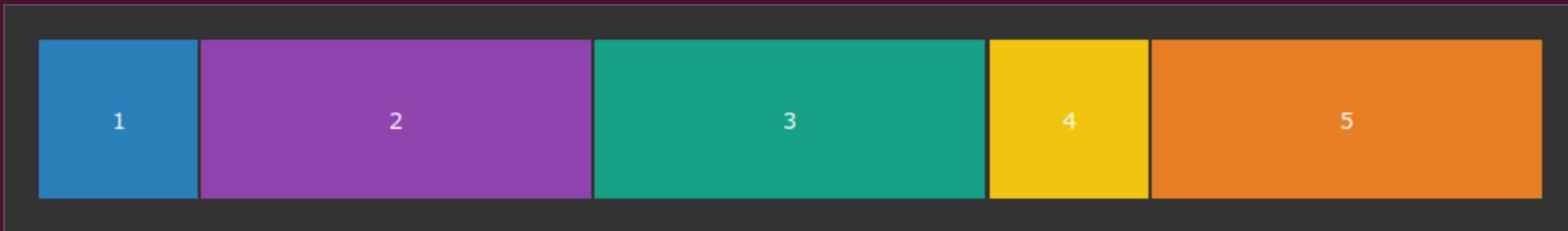
## Avec Flex



```
.parent {  
    display: flex;  
    flex-wrap: wrap;  
}  
.enfant {  
    width: 200px;  
}
```

# 4. Multi-lignes

## Avec Flex



```
.parent {  
    display: flex;  
    flex-wrap: wrap;  
}  
.enfant { width: 100px; }  
  
.enfant:nth-child(3n + 2),  
.enfant:nth-child(3n + 3) {  
    flex: 1;  
}
```

# 4. Multi-lignes

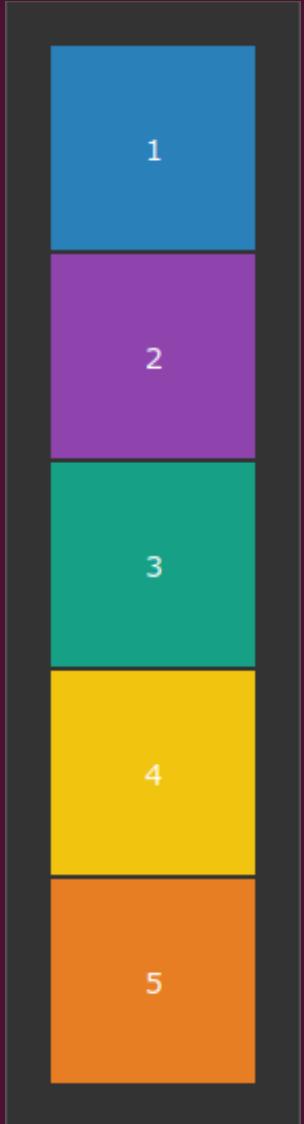
## Avec Flex



```
.parent { display: flex; flex-wrap: wrap; }
.enfant { width: 100px; }
.enfant:nth-child(3n + 2),
.enfant:nth-child(3n + 3) {
  flex: 1;
  min-width: 40%;
}
```

# 4. Multi-lignes

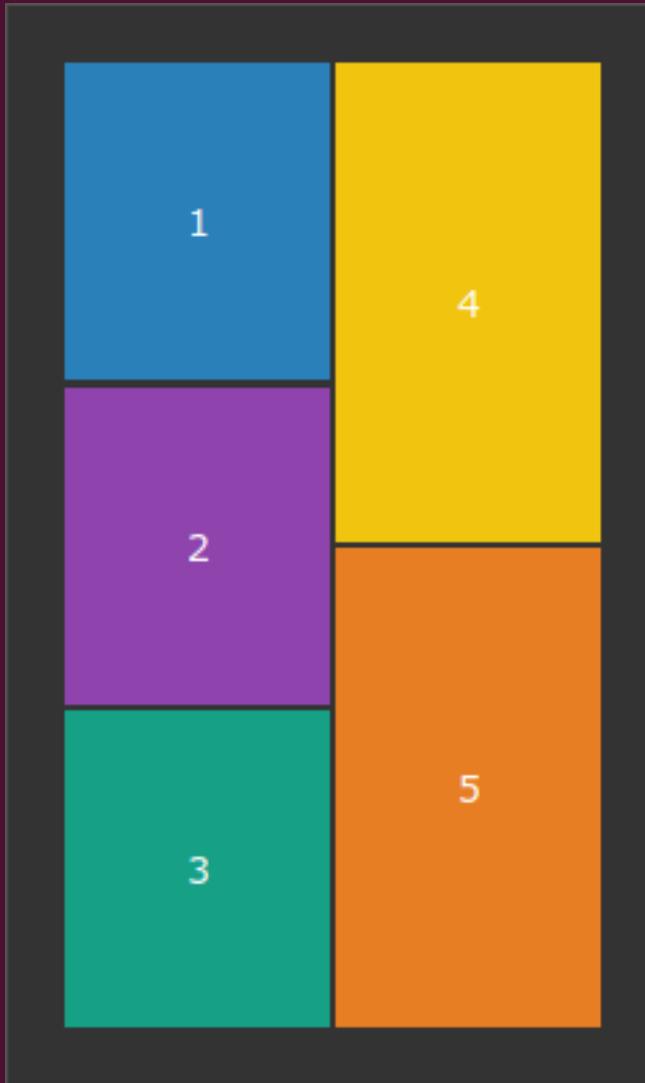
Avec Flex



```
.parent {  
    display: flex;  
    flex-direction: column;  
}  
.enfant {  
    height: 100px;  
}
```

# 4. Multi-lignes

Avec Flex



```
.parent {  
    display: flex;  
    flex-direction: column;  
    flex-wrap: wrap;  
}  
.enfant {  
    min-height: 100px;  
    flex: 1;  
}
```

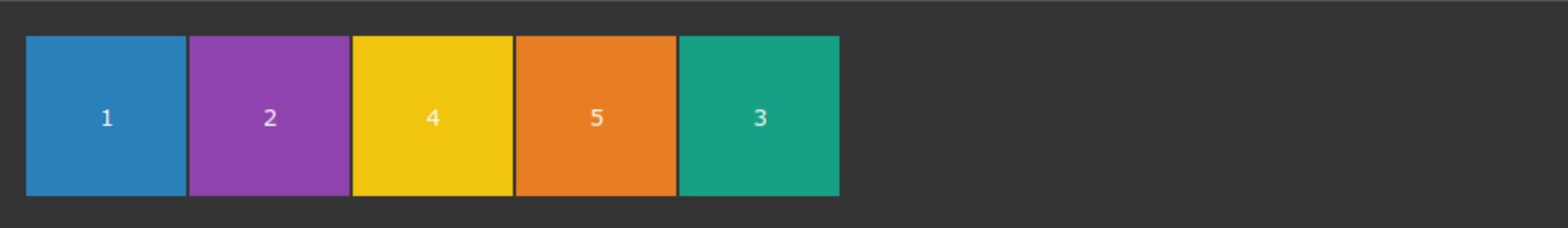
# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1

# 5. Positionnement

# 5. Positionnement

## Avec Flex



```
.parent {  
    display: flex;  
}  
  
.enfant:nth-child(3) {  
    order: 1;  
}
```

# 5. Positionnement

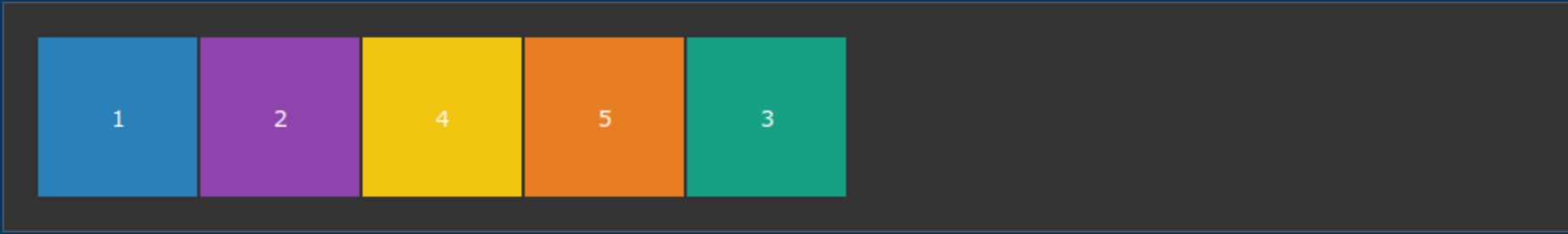
## Avec Flex



```
.parent {  
    display: flex;  
}  
  
.enfant:nth-child(3) {  
    order: -1;  
}
```

# 5. Positionnement

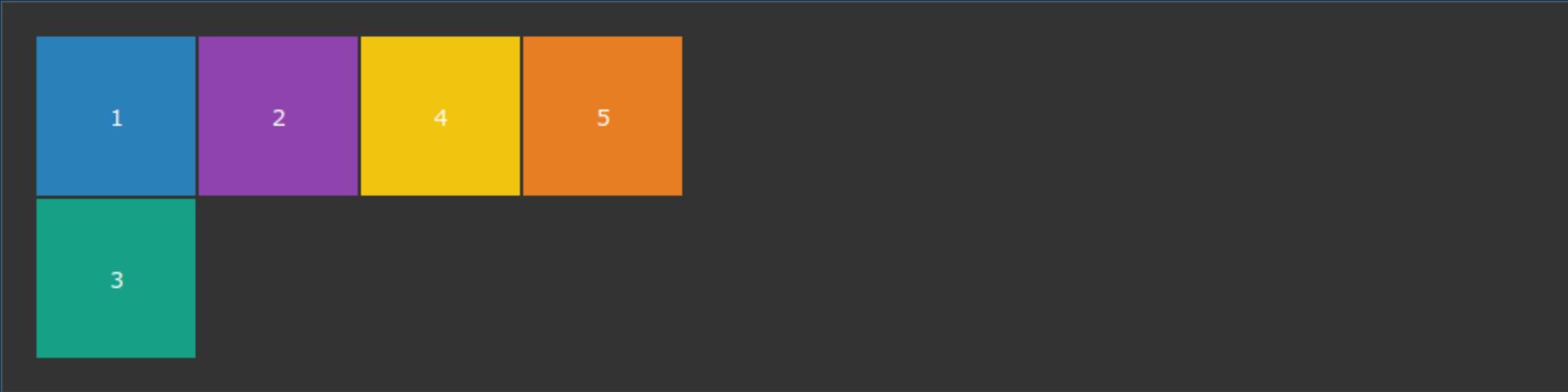
## Avec Grid



```
.parent {  
    display: grid;  
}  
  
.enfant:nth-child(3) {  
    order: 1;  
}
```

# 5. Positionnement

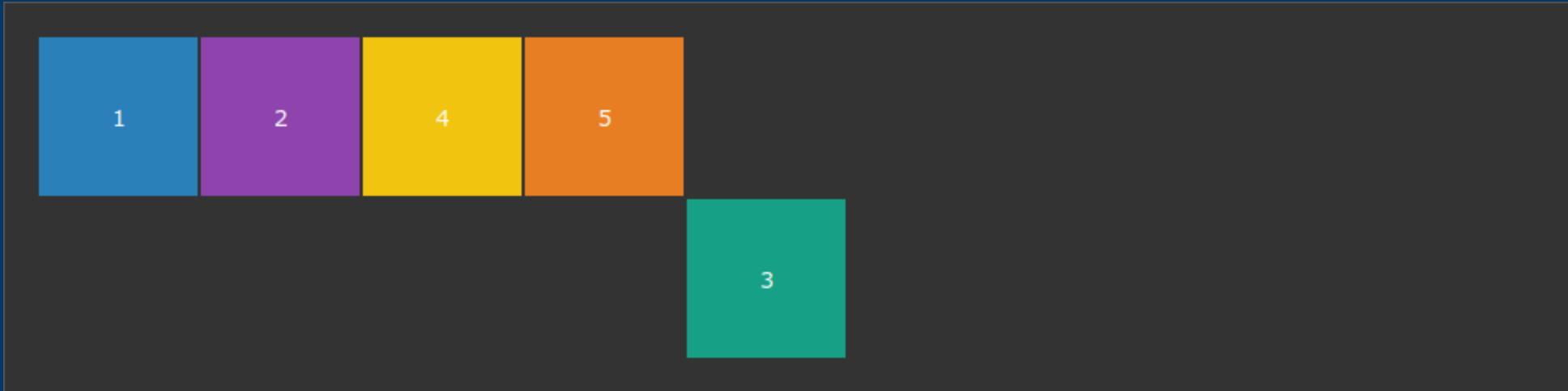
## Avec Grid



```
.parent { display: grid; }
.enfant:nth-child(3) {
    grid-row: 2;
}
```

# 5. Positionnement

## Avec Grid



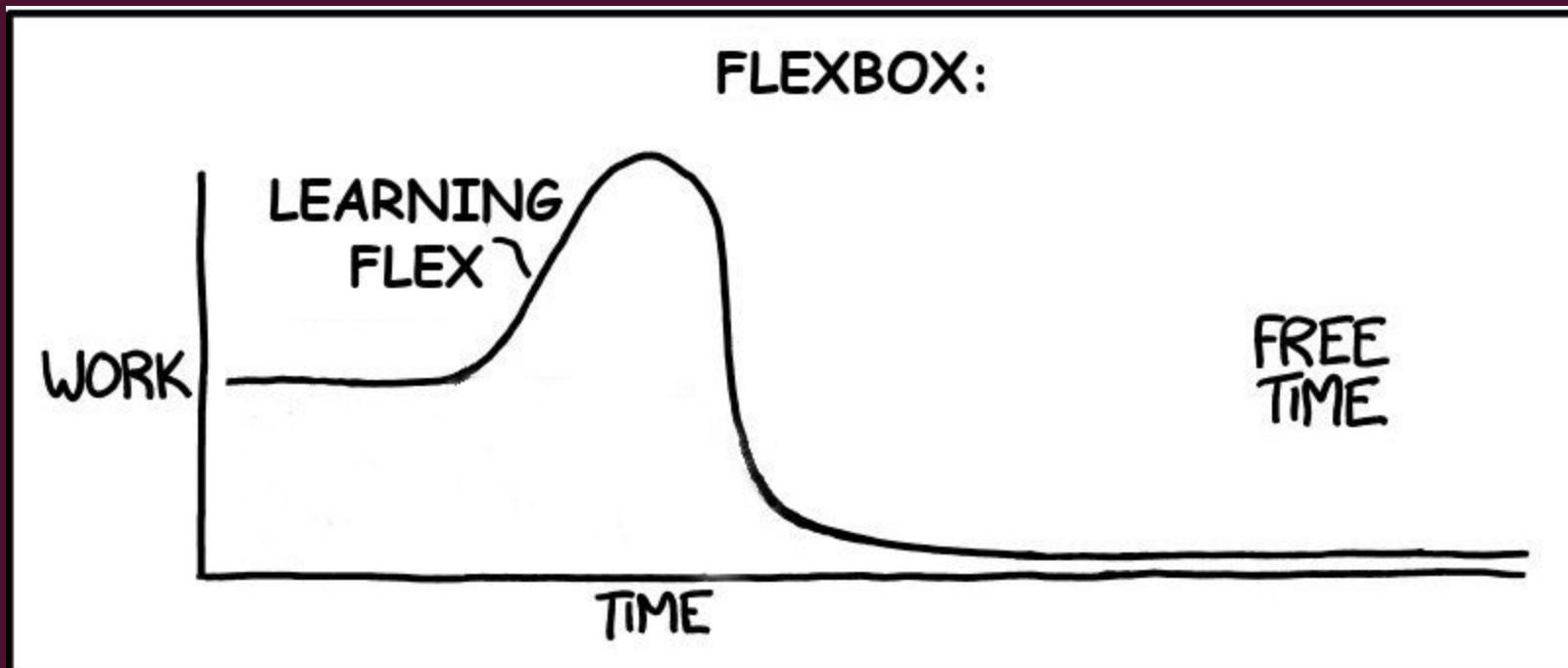
```
.parent { display: grid; }
.enfant:nth-child(3) {
  grid-row: 2;
  grid-column: 5;
}
```

# Fight

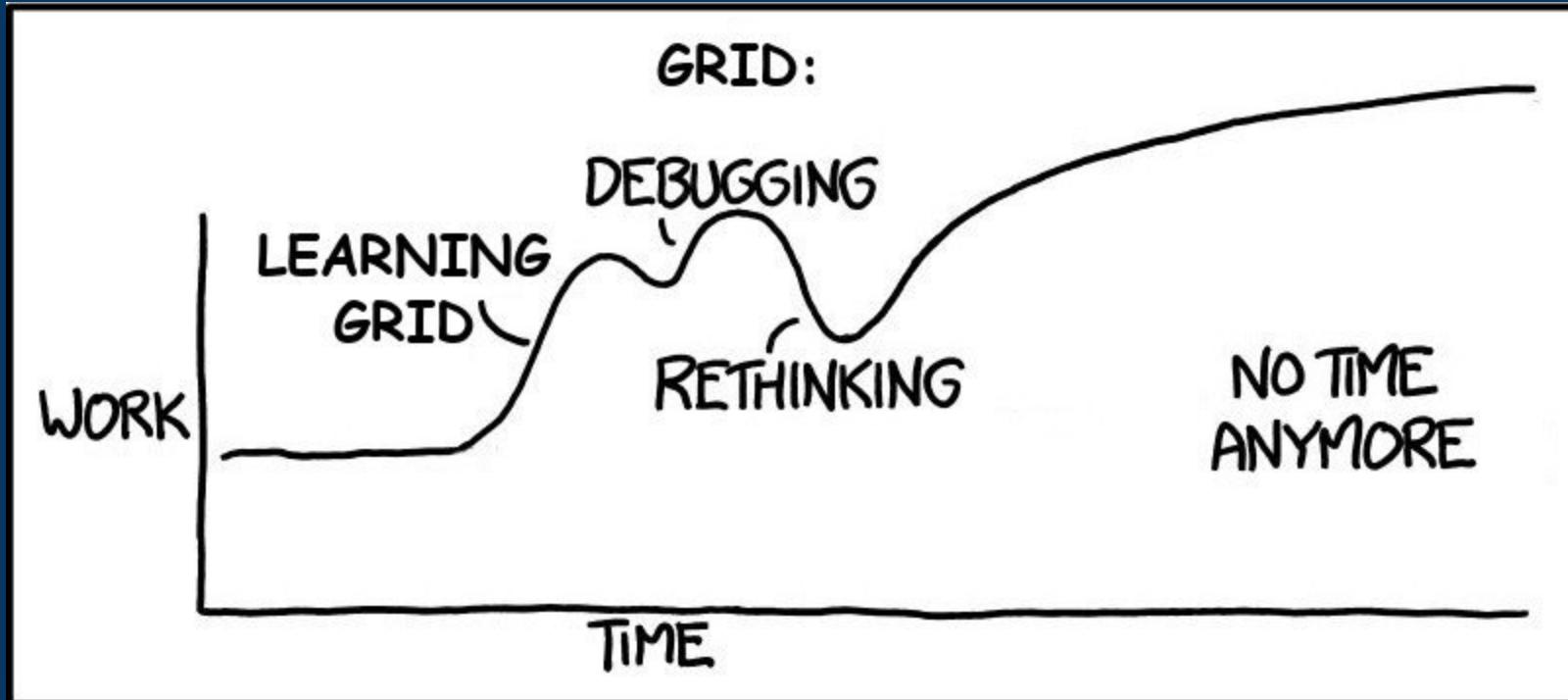
Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1
Positionnement	0	1

# 6. Courbe d'apprentissage

# 6. Courbe d'apprentissage De Flex



# 6. Courbe d'apprentissage De Grid

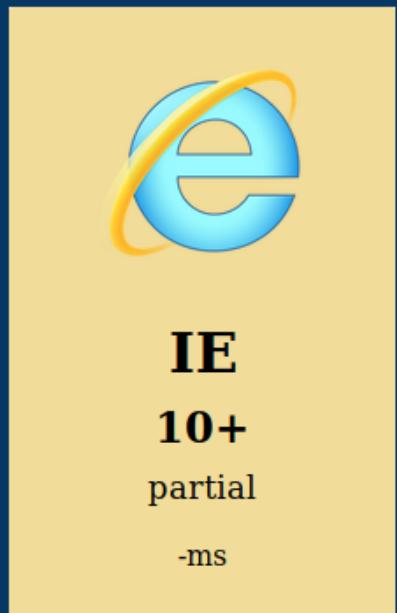


# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1
Positionnement	0	1
Simplicité	1	0

# 7. Support

# 7. Support De Grid



# 7. Support De Flex


<b>IE</b>
<b>11</b>
partial


<b>Edge</b>
<b>12+</b>
full


<b>Safari</b>
<b>9+</b>
no prefix


<b>Firefox</b>
<b>28+</b>
no prefix


<b>Chrome</b>
<b>29+</b>
no prefix

# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1
Positionnement	0	1
Simplicité	1	0
Support	1	0.5

# 8. Dev Tools

# 8. Dev Tools

## De Grid

About membership

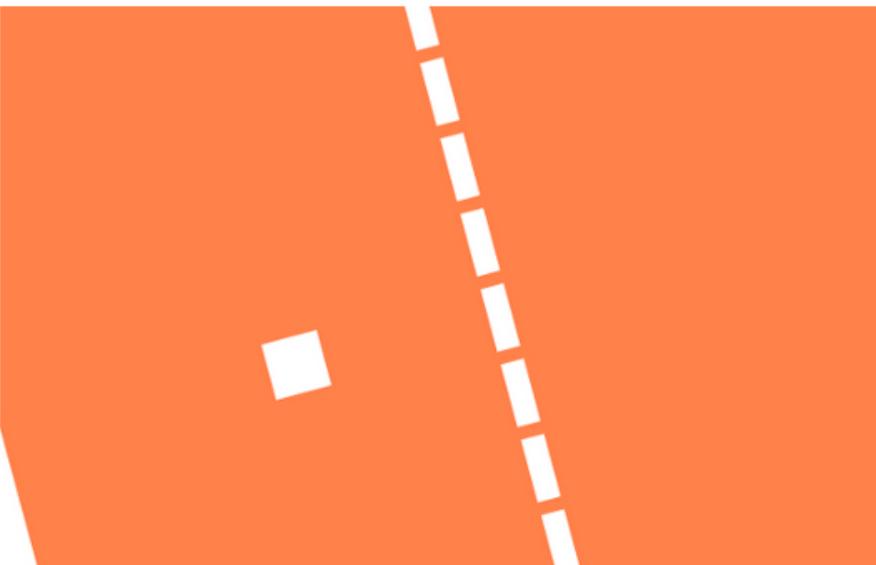
# Medium

Sign in Get started

HOME UNRULY BODIES TECHNOLOGY CULTURE ENTREPRENEURSHIP CREATIVITY SELF POLITICS MEDIA PRODUCTIVI >

FEATURED

**Son of Pong**  
Coming of age as the child of a video game junkie  
John DeVore • ★ 15 min read



**The Many Faces of Bitcoin**  
By Murad Mahmudov and Adam Taché.  
Adam Taché

**The Body on the Edge of Womanhood and Queerness**  
On learning to accept love after losing it  
Your Fat Friend • ★

**We Were Content Then, And Young**  
A Decade on the Garbage Hell Site  
Daniel P. Shannon

---

**How the Democrats could still lose the 2018 midterms**  
Kent Kroeger • ★

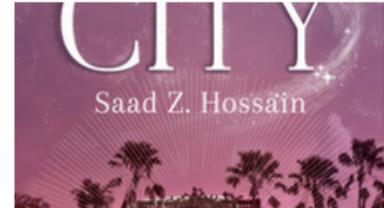
**The Body's Evolving Desire**  
Larissa Pham • ★

**The Secrets of Telling Your Future**  
Matthew Charles Davis • ★

**See all featured >**

---

**CITY**  
Saad Z. Hossain



**DJINN CITY**  
A NOVEL



**10 Book Designers Discuss the Book Covers They Rejected, And Why**  
Jo Lou • 12 min read

# 8. Dev Tools

## De Grid

About membership

# Medium

Sign in Get started

HOME UNRULY BODIES TECHNOLOGY CULTURE ENTREPRENEURSHIP CREATIVITY SELF POLITICS MEDIA PRODUCTIVI >

FEATURED

**Son of Pong**  
Coming of age as the child of a video game junkie  
John DeVote ★ 15 min read

**The Many Faces of Bitcoin**  
By Murad Mahmudov and Adam Taché  
Adam Taché

**The Body on the Edge of Womanhood and Queerness**  
On learning to accept love after losing it  
Your Fat Friend ★

**We Were Content Then, And Young**  
A Decade on the Garbage Hell Site  
Daniel P. Shandon

**How the Democrats could still lose the 2018 midterms**  
Kent Kroeger ★

**The Body's Evolving Desire**  
Carissa Pham ★

**The Secrets of Telling Your Future**  
Matthew Charles Davis ★

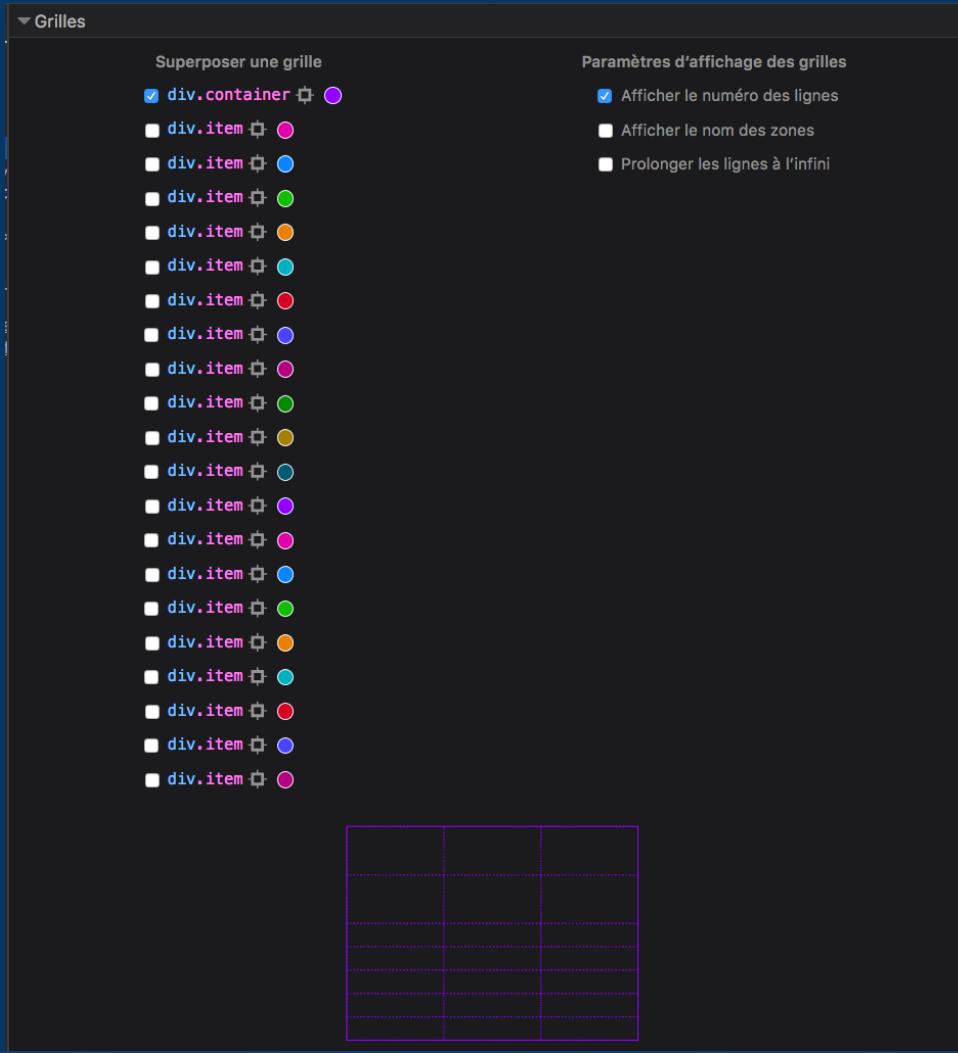
**See all featured >**

**CITY**  
Saad Z. Hossain

**DROWN CITY**  
A NOVEL  
Jo Lou ★ 12 min read

# 8. Dev Tools

## De Grid



# 8. Dev Tools

## De Flex

The screenshot shows a browser developer tools window with the 'Layout' tab selected. At the top, there's a visual representation of a flexbox container with several items. The container has a dashed purple border and a light blue background. Inside, there are ten circular items of various colors (purple, green, red, blue, brown) arranged in two rows of five. Each item is surrounded by a thin blue border. Below the visual, the browser's developer tools interface is visible, including the navigation bar with tabs like Inspector, Console, Debugger, Style Editor, Performance, Memory, Network, Storage, and Layout.

**Visual Representation:** A horizontal row of ten circular items, each with a thin blue border, arranged in two rows of five. The container has a dashed purple border and a light blue background.

**Developer Tools:**

- Inspector:** Shows the DOM structure:

```
<section class="initial">
  <h1>flex: initial;</h1>
  <h2>No Wrap</h2>
  <div class="flex">...</div> flex
  <h2>Wrap</h2>
  <div class="flex wrap"> flex
    
    
    
    
    
    
```
- Style Editor:** Shows the CSS rules for the selected element:

```
element { inline }
.wrap { flex-wrap: wrap; }
.flex { display: flex; align-items: center; justify-content: space-between; }
div { margin: 0 2rem; }
```
- Layout Tab:** Shows the layout analysis for the selected element, indicating it's an Overlay Flexbox.
- Computed:** Shows the computed styles for the element.
- Animations:** Shows the available animations for the element.
- Grid:** Shows a message: "CSS Grid is not in use on this page".
- Box Model:** Shows a message: "Box Model".

# Fight

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1
Positionnement	0	1
Simplicité	1	0
Support	1	0.5
Dev Tools	0	1

9. Think outside the (flex)box

# 9. Think outside the (flex)box

**Flexbox proche**  
des applications mobiles / natives

**Flex porté**  
en C/C++ par Facebook pour React Native



# Le bilan

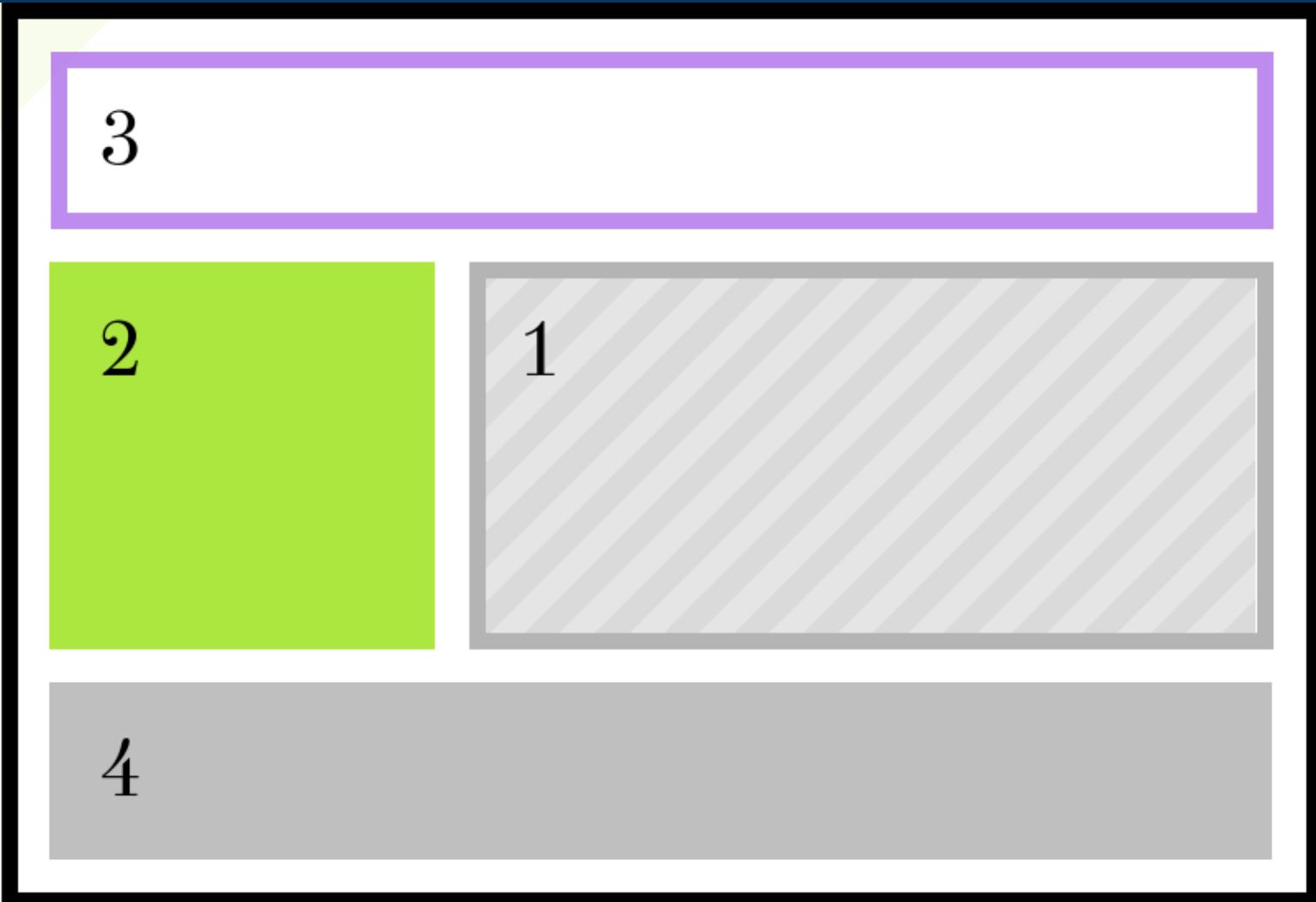
# Le Bilan

Feature	Flex	Grid
Colonnes	1	2
Alignement	1	1
Elasticité	1	1
Multi-lignes	0	1
Positionnement	0	1
Simplicité	1	0
Support	1	0.5
Dev Tools	0	1
Outside the web	2.5 	0
<b>Total</b>	<b>7.5</b>	<b>7.5</b>

Mais...

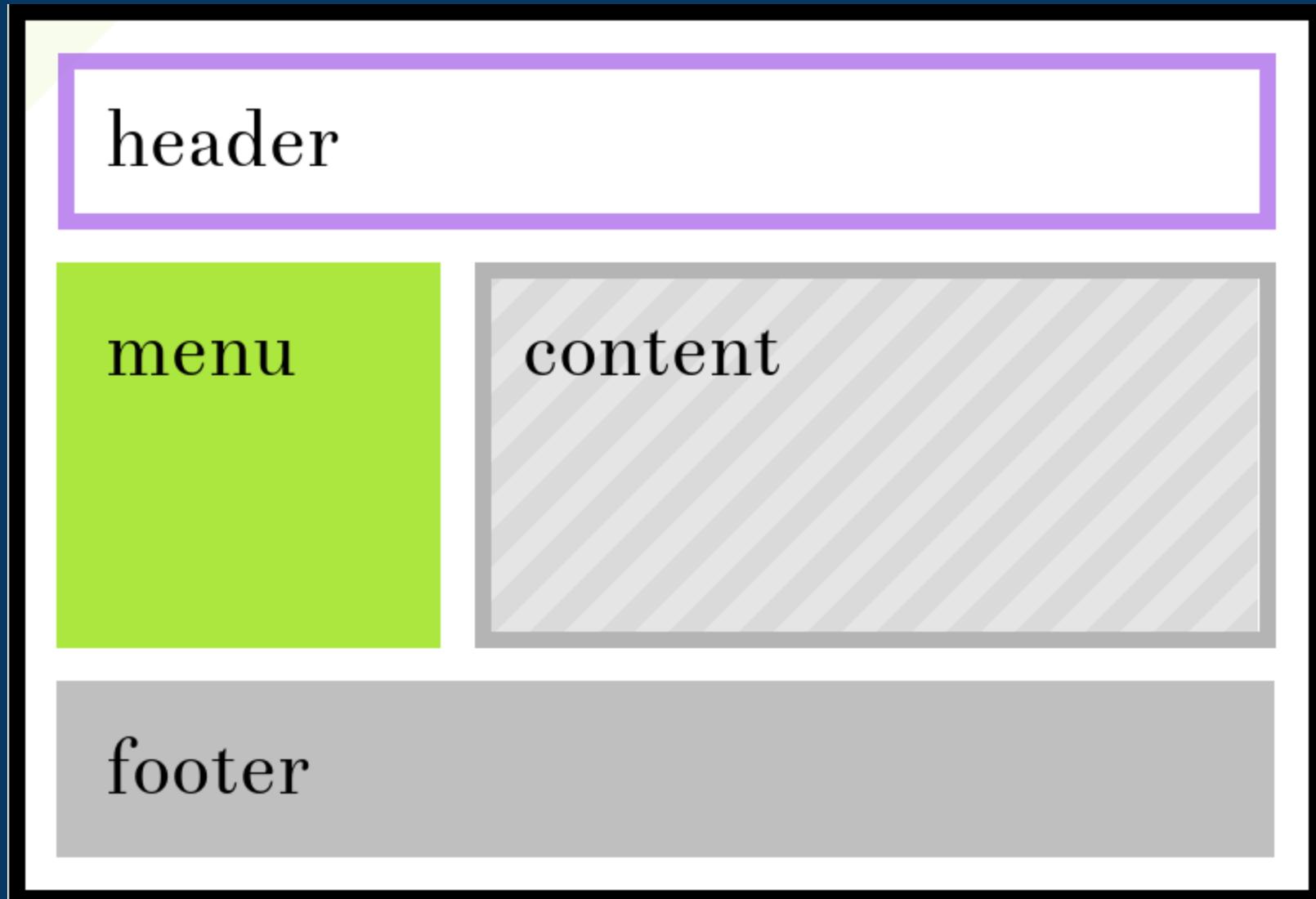
# Fusion de cellules

```
.enfant3 {  
    grid-column: span 2;  
}
```



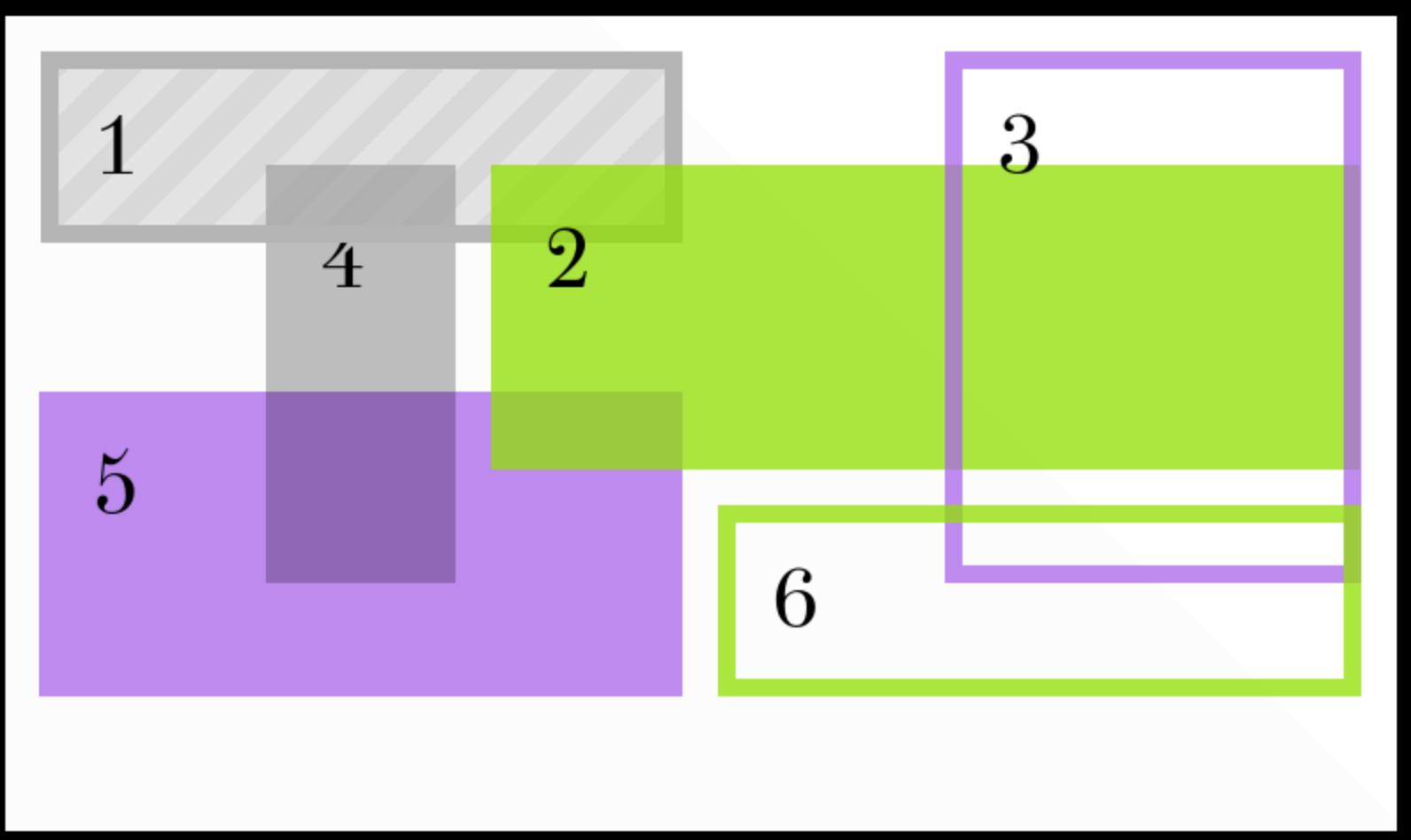
# Nommer des zones

```
.parent {  
  grid-template-areas:  
    "header header"  
    "menu content"  
    "footer footer"  
}  
  
.enfant1 {  
  grid-area: content;  
}  
  
.enfant2 {  
  grid-area: menu;  
}  
  
.enfant3 {  
  grid-area: header;  
}  
  
...
```

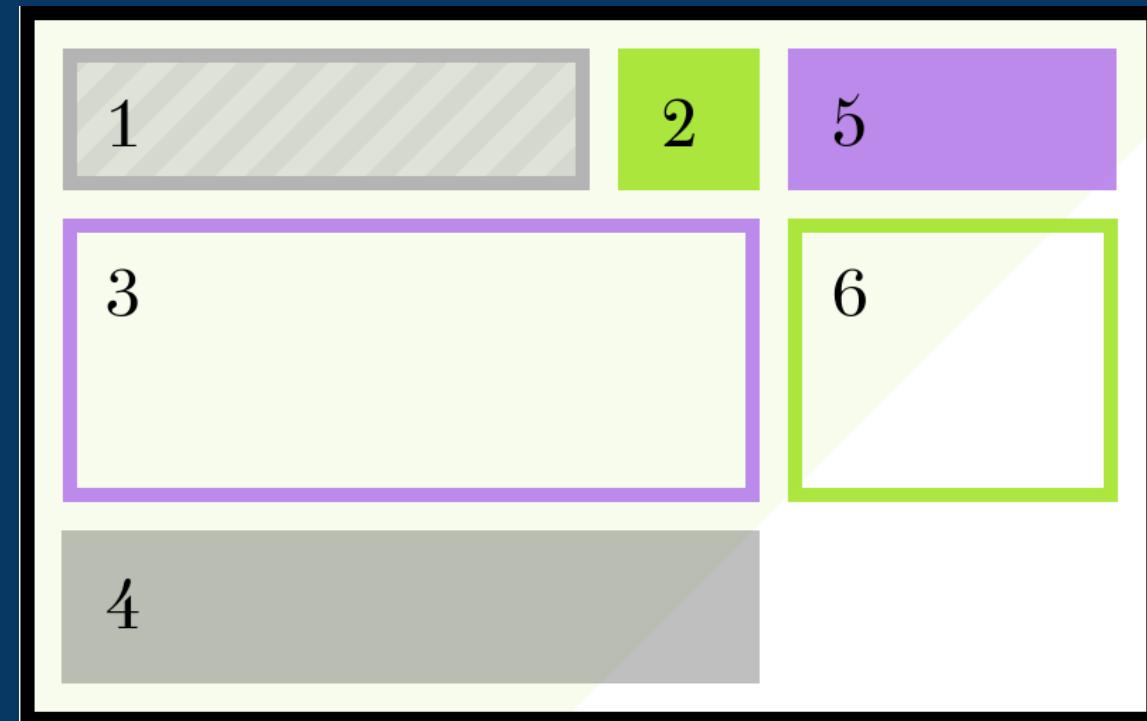
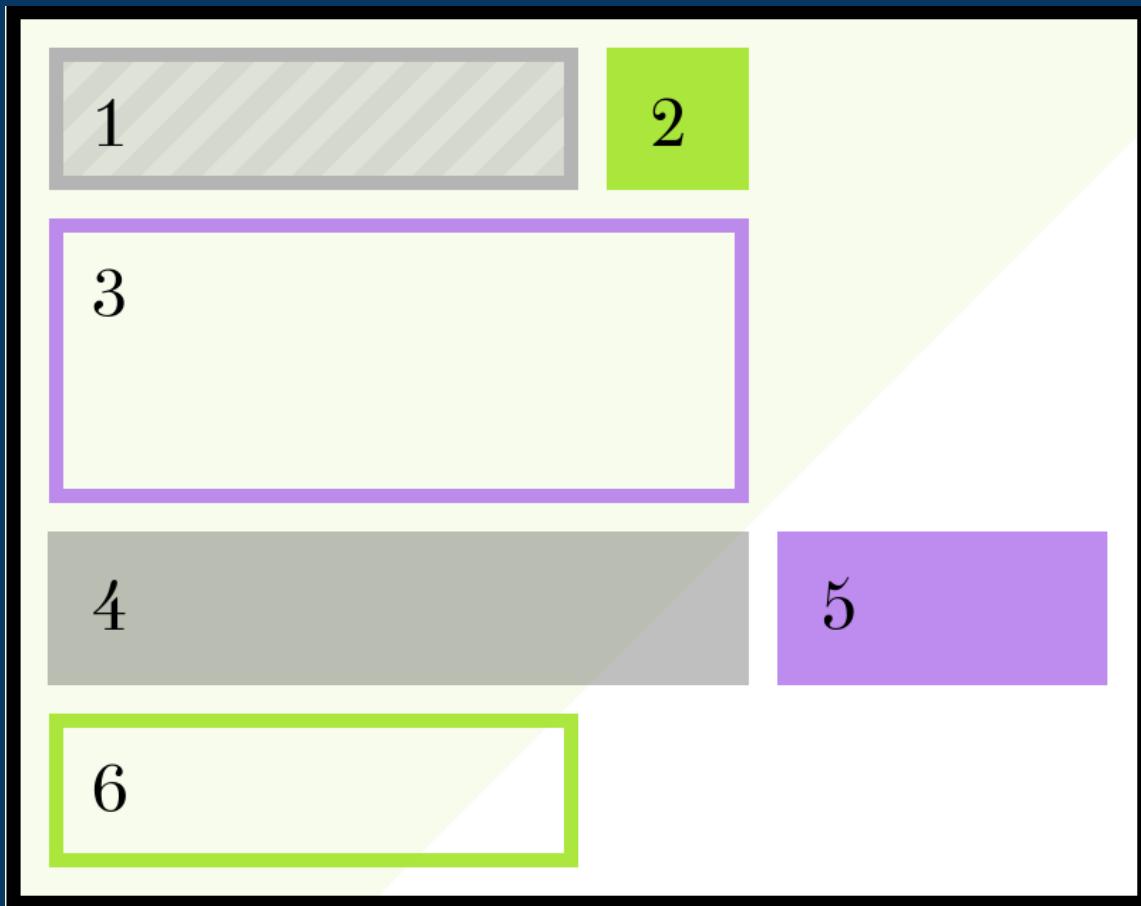


# Superposition

```
.enfant2 {  
    grid-column: 3 / span 2;  
    grid-row: 2;  
}  
  
.enfant3 {  
    grid-column: 4;  
    grid-row: 1 / span 3;  
}
```



# Algorithme de répartition



```
.parent {  
  grid-auto-flow: row/column/dense  
}
```

# Flex ou Grid ?

Grid  
pour les gabarits complexes

Flex  
pour les gabarits simples ou les  
composants (si application)



# BLEND WEB MIX

[BILLETTERIE](#)[Programme](#) ▾ [BlendWebMix ?](#) [Blog](#) [Participants](#) [Job Board](#) [Infos pratiques](#)

24 et 25  
octobre  
2018

**LES DEUX JOURS LES +  
WEB DE L'ANNÉE !**

au Centre des Congrès à Lyon !

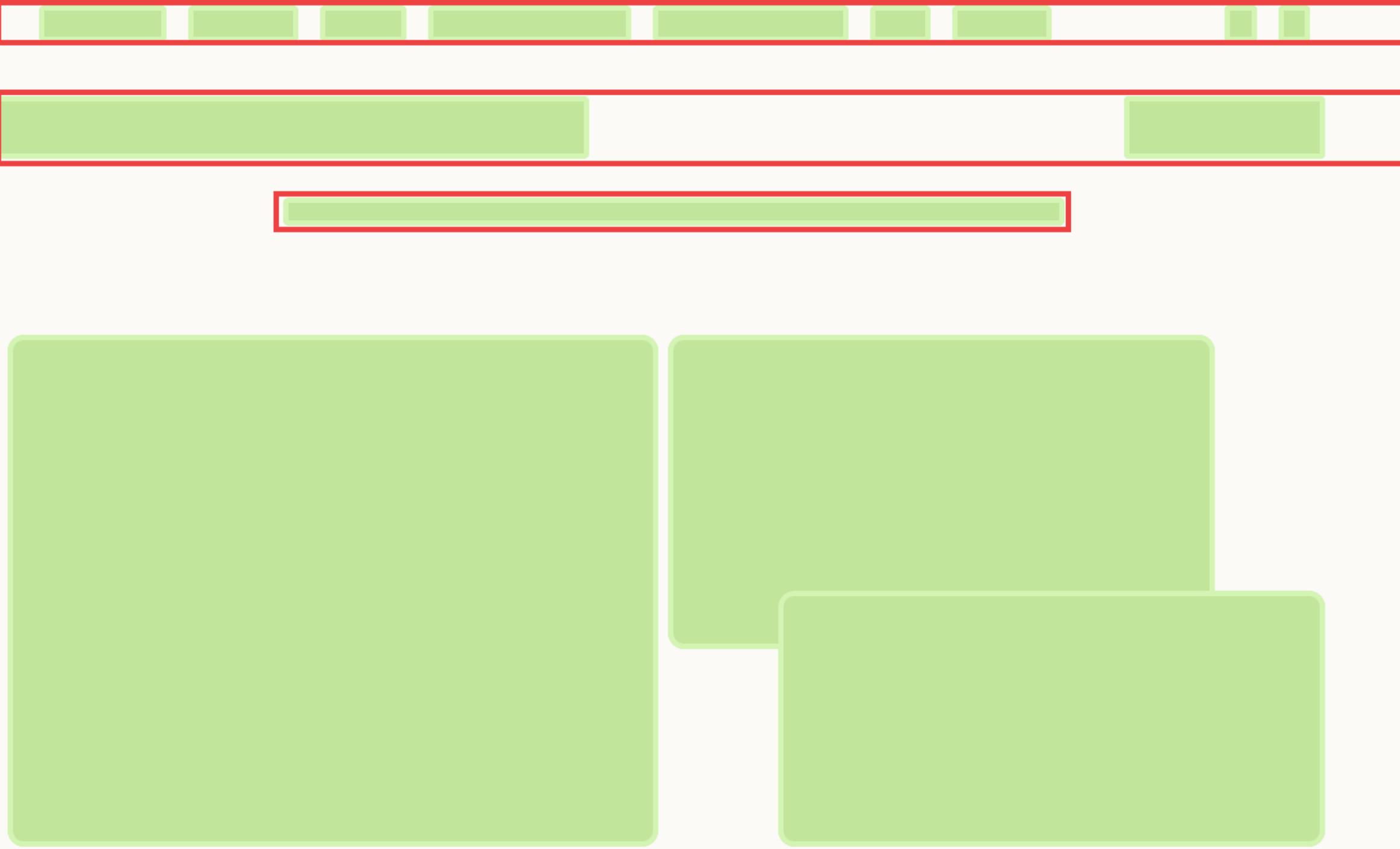
**2000 participants attendus  
pour l'édition 2018 !**

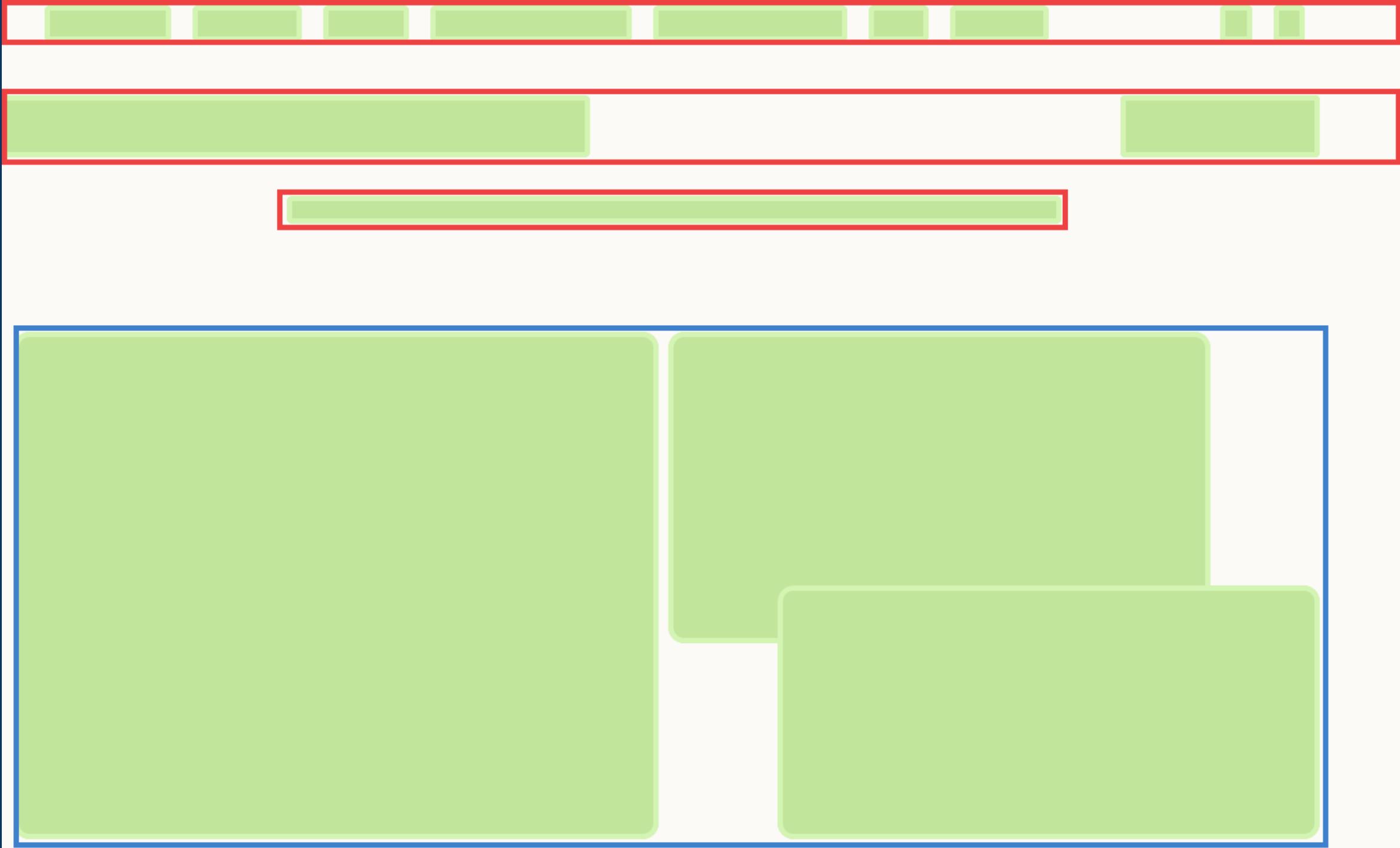
De quoi se régaler en networking !

Tech, Design, Business : plus de **70**  
**conférences** à découvrir !

[Découvre le programme !](#)







Mais...

Surtout dans le futur...

Aujourd'hui seul Flexbox  
peut être utilisé sur IE10+

CSS Grid est un "bonus"

# Merci !

N'hésitez pas à venir au Village (espace animations à 10h) pour discuter de Flex et de Grid

Thomas Zilliox

@iamtzi

<https://tzi.fr>



Malik Baba Aïssa

@malik\_ba

<https://codepen.io/malikba>

Des ressources pour s'y mettre

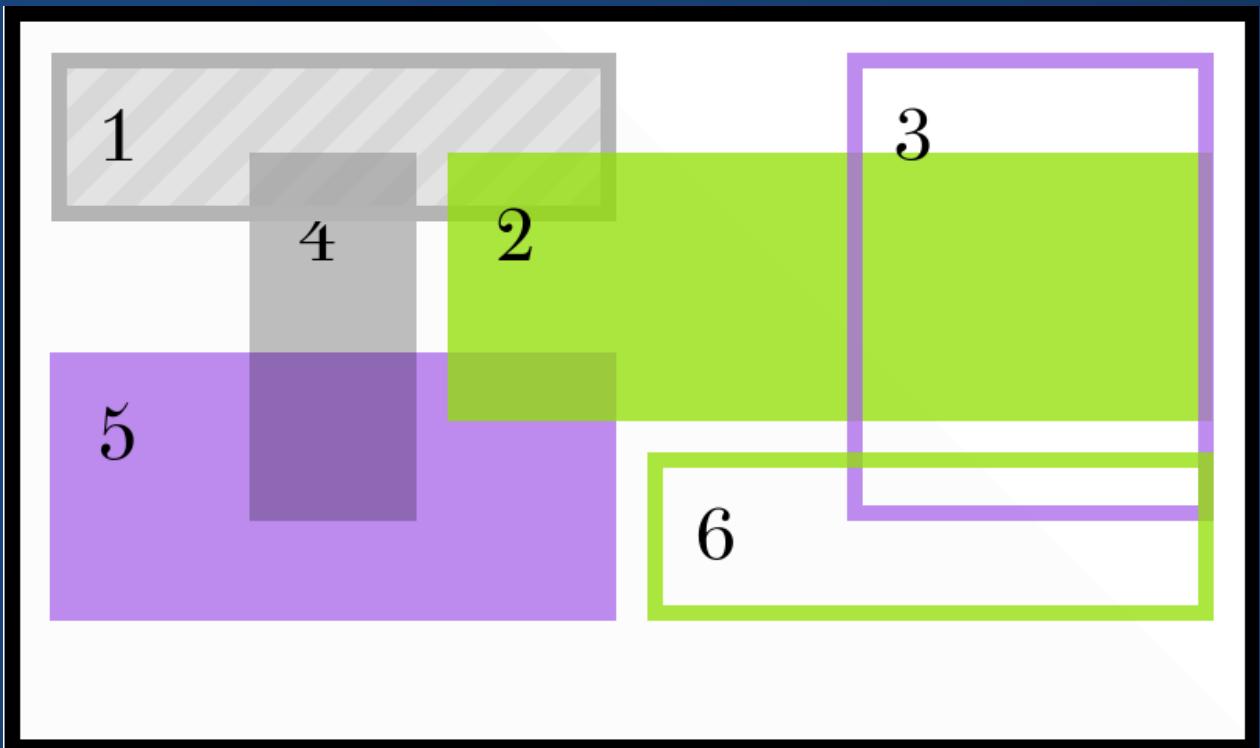
# Pour explorer : Les bacs à sable

**Flexbox :**

<https://bit.ly/play-flex>

**Grid :**

<https://bit.ly/play-grid>



# Pour essayer : Les jeux

Flexbox :

<https://flexboxfroggy.com/#fr>

Grid :

<https://cssgridgarden.com/#fr>

**GRID GARDEN**

Niveau 1 de 28 ▶

Bienvenue dans Grid Garden, où vous écrivez du code CSS pour cultiver votre jardin de carotte ! Arrosez seulement les régions qui ont des carottes en utilisant la propriété `grid-column-start`.

Par exemple, `grid-column-start: 3;` arrosera la région commençant à la 3ème ligne verticale de la grille, une autre manière de le dire, c'est la 3ème bordure verticale en partant de la gauche dans la grille.

Grid Garden est créé par [Codepiper](#) • [GitHub](#) •



# Pour comprendre : Les tutoriels MDN

## Flexbox :

<https://bit.ly/mdn-flex>

The screenshot shows the MDN web docs website. The header features the MDN logo (a white fox head icon) and the text 'MDN web docs' next to it. Below the logo is the Mozilla logo ('moz:lla'). To the right of the logo are three navigation items: 'Technologies ▾', 'Guides et références ▾', and 'Votre avis ▾'. The main title 'Les concepts de base pour flexbox' is displayed prominently in large, bold, dark font.

## Grid :

<https://bit.ly/mdn-grid>

The screenshot shows the MDN web docs website. The header features the MDN logo (a white fox head icon) and the text 'MDN web docs' next to it. Below the logo is the Mozilla logo ('moz:lla'). To the right of the logo are three navigation items: 'Technologies ▾', 'Guides et références ▾', and 'Votre avis ▾'. The main title 'Les concepts de base des grilles CSS' is displayed prominently in large, bold, dark font.

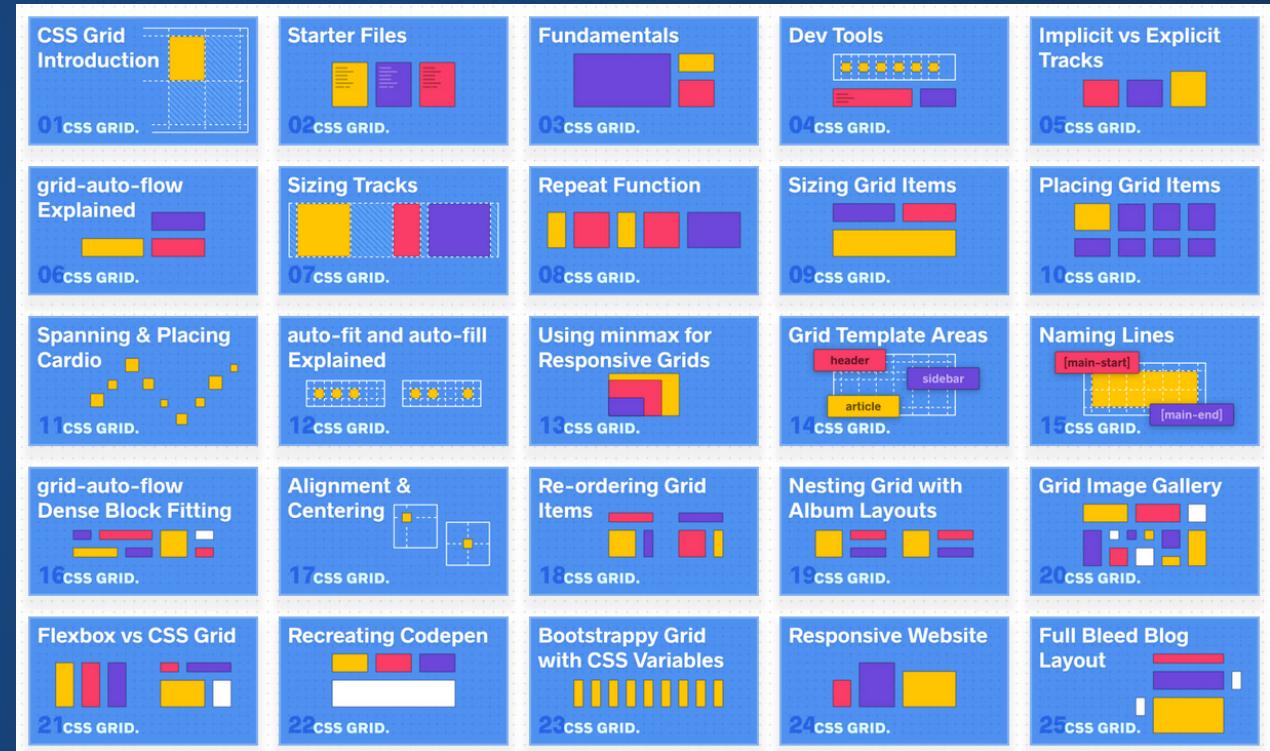
# Pour retenir : Les exercices (vidéos et code)

Flexbox :

<https://flexbox.io/>

Grid :

<https://cssgrid.io/>



# Merci (encore) !

Des questions ?

Thomas Zilliox

@iamtzi

<https://tzi.fr>



Malik Baba Aïssa

@malik\_ba

<https://codepen.io/malikba>