

CSS : grid (tutoriel sur le placement)

Remarque. Ce tutoriel est extrait de la section de MDN intitulée : [Placement basé sur les lignes](#). Les sources des exemples accompagnent ce sujet.

Dans [l'article sur les concepts de base](#), nous avons vu comment positionner des éléments en utilisant des numéros de lignes. Nous allons désormais étudier cette fonctionnalité de positionnement plus en détail.

Commencer par utiliser les lignes numérotées est plutôt logique car toutes les grilles possèdent des lignes numérotées. Ces lignes forment les colonnes et les lignes horizontales de la grille, elles sont numérotées à partir de 1. On notera aussi que la numérotation des lignes varie selon le mode d'écriture du document. Dans un document écrit de gauche à droite comme le français, la ligne numéro 1 est située à l'extrémité gauche de la grille. Si l'écriture va de droite à gauche, la ligne numéro 1 sera celle qui est située le plus à droite. Nous explorerons ces notions sur les modes d'écriture dans [un prochain guide](#).

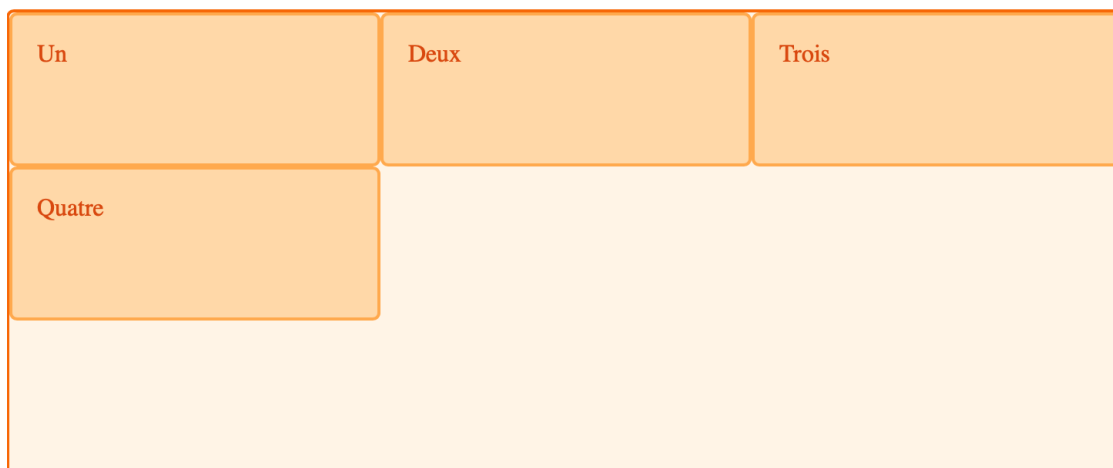
Un exemple simple

Dans cet exemple simple, on a une grille avec trois pistes pour les colonnes et trois pistes pour les lignes, on a donc 4 lignes pour chaque dimension.

Dans le conteneur, on a quatre éléments fils. Si aucune autre règle de placement n'est indiquée, ces éléments seront placés automatiquement et la grille remplira les quatre premières cellules. Si vous utilisez [l'outil de mise en évidence des grilles de Firefox](#) ou de Google Chrome, vous pouvez voir les colonnes et les lignes horizontales formées par la grille.



```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: repeat(3, 100px);  
}  
  
<div class="wrapper">  
  <div class="box1">Un</div>  
  <div class="box2">Deux</div>  
  <div class="box3">Trois</div>  
  <div class="box4">Quatre</div>  
</div>
```



Positionner les éléments d'une grille grâce au numéro de ligne

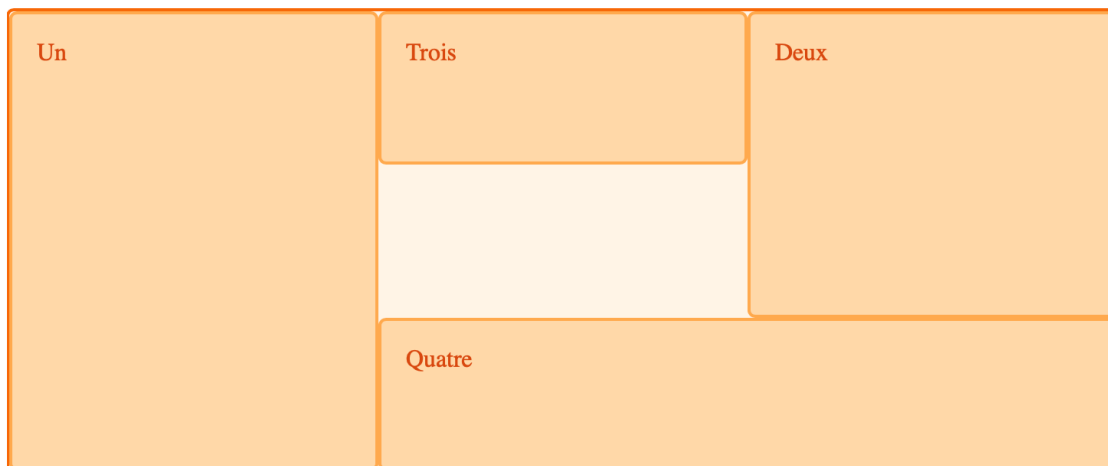
On peut placer les éléments d'une grille en utilisant les numéros de lignes qui définissent la zone allouée à l'élément. Si on souhaite que le premier élément commence tout à gauche et occupe une colonne, qu'il commence sur la première ligne et s'étale sur quatre lignes, on pourra utiliser les règles suivantes :

```
.box1 {
  grid-column-start: 1;
  grid-column-end: 2;
  grid-row-start: 1;
  grid-row-end: 4;
}
```

Lorsqu'on positionne des objets sur la grille, les autres continueront d'être placés selon les règles de placement automatique. Nous verrons ces règles dans [un prochain guide](#) mais grâce à cet exemple, on peut voir que les cellules vides sont remplies au fur et à mesure par les objets qui ne sont pas placés explicitement.

On peut placer chacun des éléments individuellement et on peut également choisir de laisser certaines cellules vides. Un des avantages de la grille CSS est qu'on peut créer des espaces sans avoir à utiliser des marges ou d'autres méthodes de contournement.

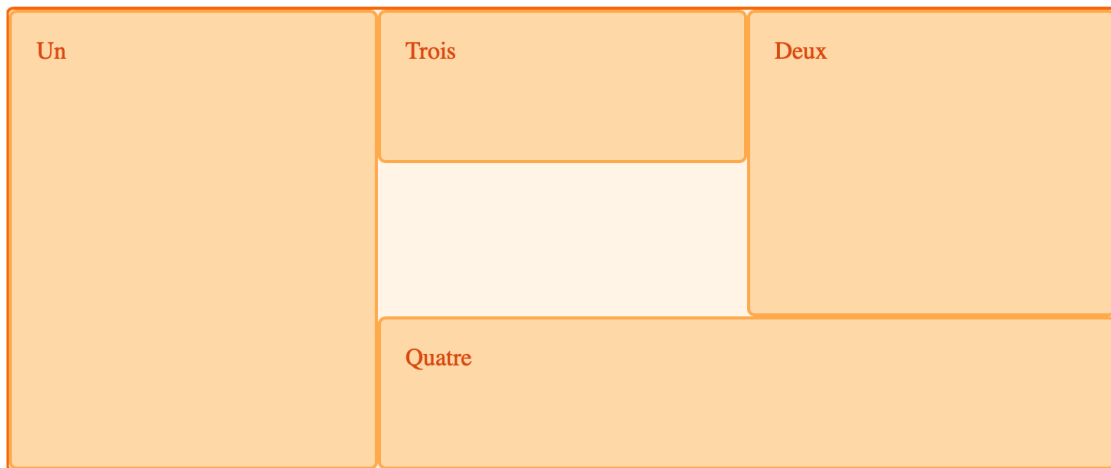
```
.box2 {
  grid-column-start: 3;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 3;
}
.box3 {
  grid-column-start: 2;
  grid-column-end: 3;
  grid-row-start: 1;
  grid-row-end: 2;
}
.box4 {
  grid-column-start: 2;
  grid-column-end: 4;
  grid-row-start: 3;
  grid-row-end: 4;
}
```



Les propriétés raccourcies `grid-column` et `grid-row`

On a écrit beaucoup de règles pour positionner chaque élément. Heureusement, il existe des propriétés raccourcies qui permettent d'avoir une syntaxe plus concise. Les propriétés `grid-column-start` et `grid-column-end` peuvent être combinées pour former la propriété raccourcie `grid-column` de la même façon, `grid-row-start` et `grid-row-end` peuvent être synthétisées avec `grid-row`.

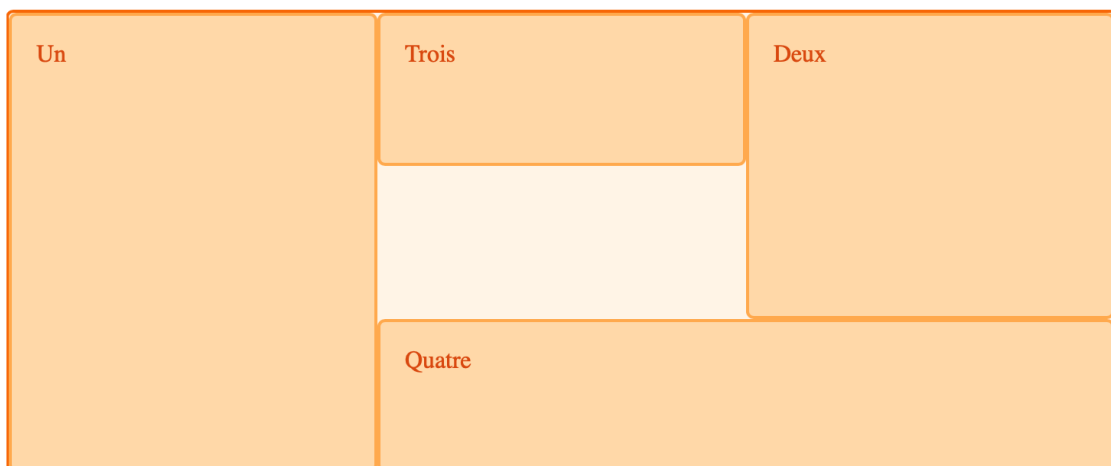
```
.box1 {  
  grid-column: 1 / 2;  
  grid-row: 1 / 4;  
}  
.box2 {  
  grid-column: 3 / 4;  
  grid-row: 1 / 3;  
}  
.box3 {  
  grid-column: 2 / 3;  
  grid-row: 1 / 2;  
}  
.box4 {  
  grid-column: 2 / 4;  
  grid-row: 3 / 4;  
}
```



Tailles par défaut avec les propriétés raccourcies

Avec les propriétés raccourcies, on obtient le code suivant (sans aucune barre oblique ni seconde valeur pour les éléments qui n'occupent qu'une seule piste).

```
.box1 {
  grid-column: 1 ;
  grid-row: 1 / 4;
}
.box2 {
  grid-column: 3 ;
  grid-row: 1 / 3;
}
.box3 {
  grid-column: 2 ;
  grid-row: 1 ;
}
.box4 {
  grid-column: 2 / 4;
  grid-row: 3 ;
}
```



La propriété `grid-area`

On peut aller plus loin et définir une zone pour chaque élément grâce à une seule propriété : `grid-area`. Cette propriété raccourcie permet d'utiliser les valeurs des propriétés suivantes (dans cet ordre) :

- `grid-row-start`
- `grid-column-start`
- `grid-row-end`
- `grid-column-end`

```
.box1 {  
  grid-area: 1 / 1 / 4 / 2;  
}  
.box2 {  
  grid-area: 1 / 3 / 3 / 4;  
}  
.box3 {  
  grid-area: 1 / 2 / 2 / 3;  
}  
.box4 {  
  grid-area: 3 / 2 / 4 / 4;  
}
```

