

fixed

R102

DÉVELOPPEMENT D'INTERFACES WEB

Jean Carle jean.carle@univ-lille.fr

POSITIONNEMENT DES BLOCS

et mise en page

https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout

PROPRIÉTÉS **position**

Elle définit la façon dont un élément est positionné.

Positionnements possibles

static	Dans le flux normal (placement par défaut)
relative	Place réservée dans le flux, décalé selon top, right, bottom, left
fixed	Hors flux (place non réservée), placé par rapport à l'écran selon top, right, bottom, left
absolue	Hors du flux (place non réservée), placé par rapport à la page selon top, right, bottom, left
sticky	position relative en dessous d'une limite, fixed au dessus. Limite définie avec top, right, bottom, left

<https://developer.mozilla.org/fr/docs/Web/CSS/position>

PROPRIÉTÉS Position

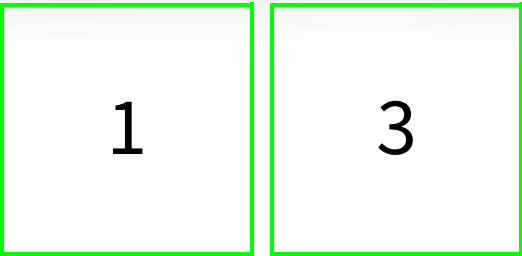
Exemples

abs

position: static

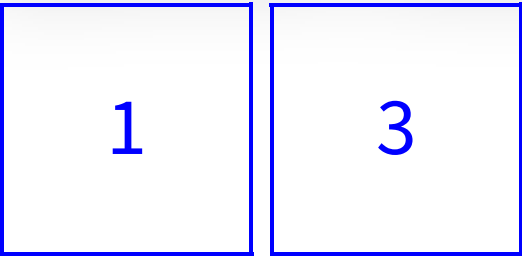


position: fixed



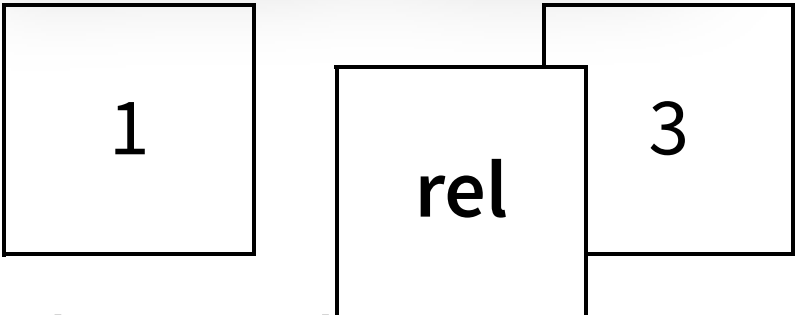
Déplacement par rapport
à la fenêtre

position: absolute



Déplacement par rapport
au conteneur (page web)

position: relative



position: sticky

Pour voir l'effet, il faut essayer de sortir le
bloc au bord de l'écran.



LES BOITES FLOTTANTES

Ce sont des éléments qui sortent du flux pour être placé au début ou à la fin de la ligne.

Les éléments voisins se placent autour.

Propriété `float`

none dans le flot standard

right l'élément est placé à droite

left l'élément est placé à gauche

<https://developer.mozilla.org/fr/docs/Web/CSS/float>

LES BOITES FLOTTANTES

Propriété `clear`

none	Peut se placer sur la même ligne qu'un élément flottant
left	Se place en dessous des éléments flottant à gauche. Comble l'espace à gauche des éléments flottant à droite.
right	Se place en dessous des éléments flottant à droite. Comble l'espace à droite des éléments flottant à gauche.
both	interdit le placement au même niveau que des éléments flottants (droite ou gauche).

<https://developer.mozilla.org/fr/docs/Web/CSS/clear>

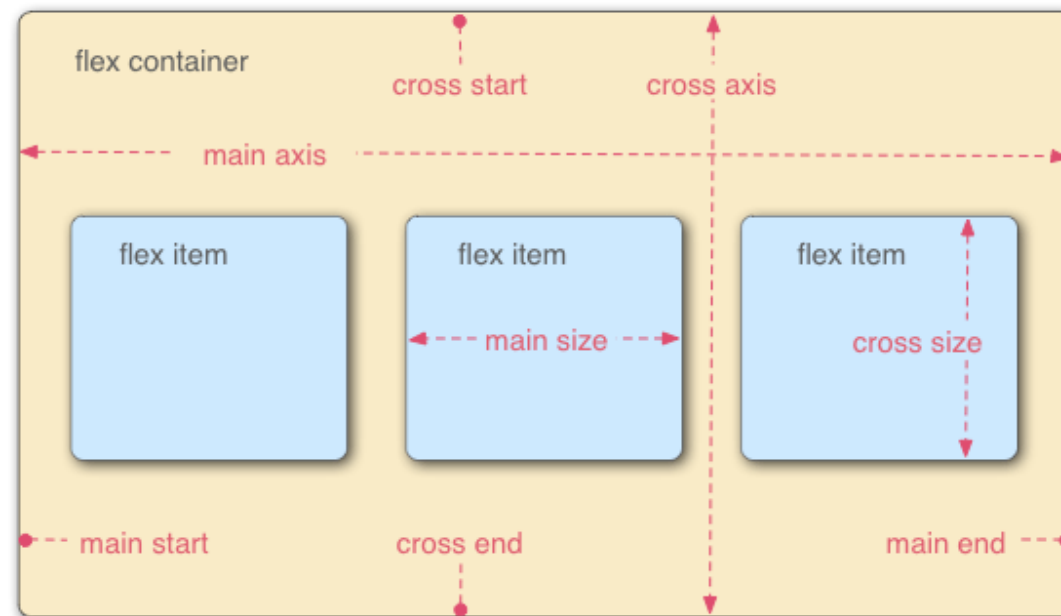
- i** Cette propriété n'a d'effet que sur les éléments de type bloc et uniquement en présence d'éléments flottants.
- i** Avec les techniques modernes comme `Flexbox` et `Grid`, l'utilisation de `float` et `clear` est devenue moins utile.

FLEXBOX

- Facilite grandement l'alignement et la répartition unidimensionnelle (horizontale ou verticale) des éléments voisins au sein d'un conteneur parent.
- Utilise un axe principal et un sens de placement (à partir du début ou de la fin)
- `flexbox` est activée avec la propriété **display** sur le parent (qui sert de conteneur) par la valeur **flex** ou **inline-flex** selon le positionnement extérieur du conteneur.

FLEXBOX

https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout



FLEXBOX - EXEMPLE

CSS

```
.conteneur {  
  display: flex;  
  flex-direction: row-reverse;  
}  
.two {  
  flex-grow: 1;  
  flex-shrink: 1;  
  flex-basis: auto;  
}  
.three { flex: 2 1 auto; }  
.four { flex: 1 1 auto; }
```

html

```
<div class="conteneur">  
  <p class="one">Un</p>  
  <p class="two">Deux</p>  
  <p class="three">Trois</p>  
  <p class="four">Quatre</p>  
</div>
```

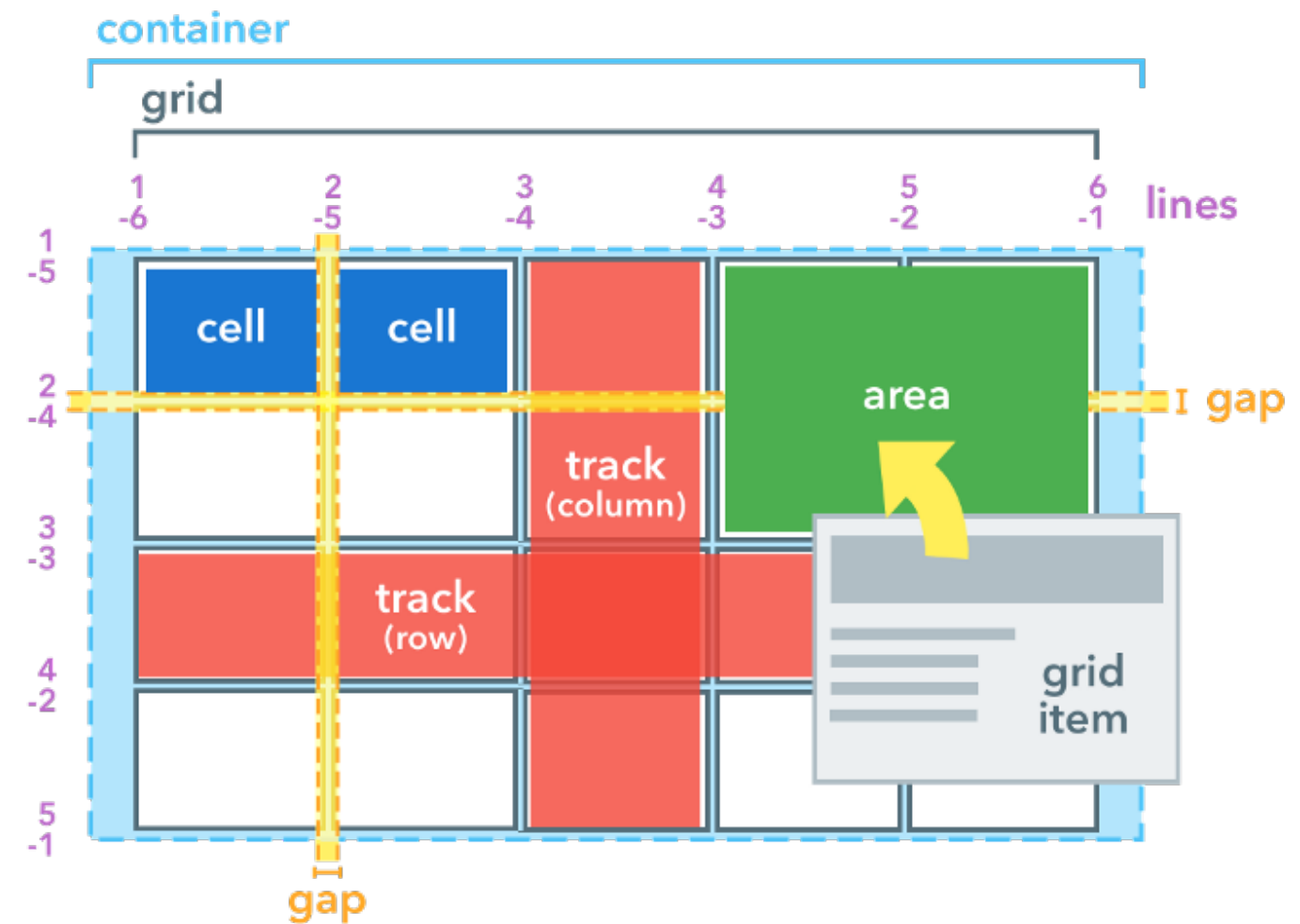
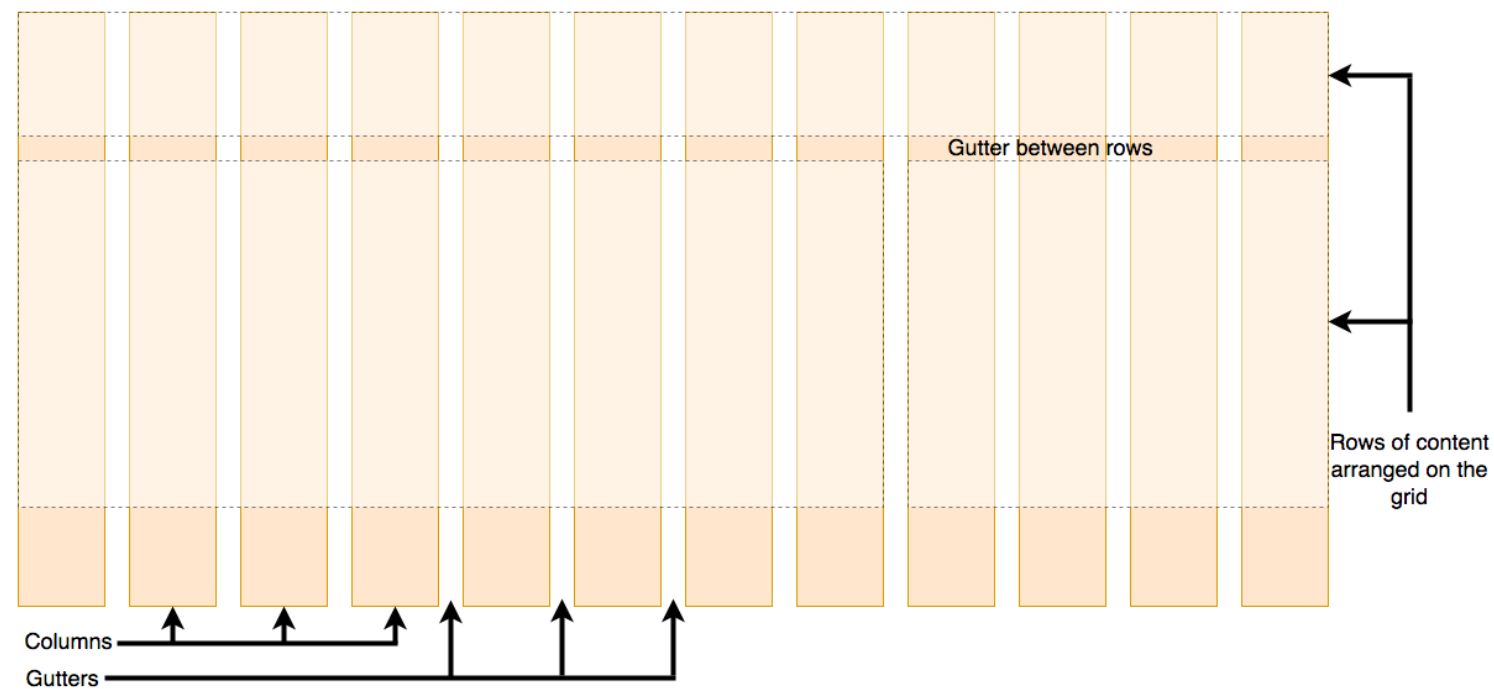
Résultat

Quatre	Trois	Deux	Un
--------	-------	------	----

GRID

- Facilite grandement les mises en page bidimensionnelles (horizontale ET verticale).
- Multiple méthodes de placement des zones utilisées
- Unité et fonctions spécifique (**fr**, **repeat()**, **minmax()**) pour simplifier la création et la répartition et la taille des cellules.
- `grid` est activée avec la propriété **display** sur le parent (qui sert de conteneur) par la valeur **grid** ou **inline-grid** selon le positionnement extérieur du conteneur.

GRID - VOCABULAIRE



https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Grids

<https://cssgrid-generator.netlify.app/>, <https://griddy.io/>

GRID - EXAMPLE

CSS

```
.conteneur {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 10px;  
  grid-auto-rows: minmax(100px, auto);  
}  
.one {  
  grid-column-start: 1 ;  
  grid-column-end: span 2;  
  grid-row: 1;  
}  
.two { grid-area: 1 / 2 / 3 / 4; }  
.five { grid-area: 4 / 2; }
```

```
.three {  
  grid-column: 1;  
  grid-row-start: 2;  
  grid-row-end: 5;  
}  
.four {  
  grid-column: 3;  
  grid-row: 3;  
}  
.six {  
  grid-column: 3;  
  grid-row: 4;  
}
```

html

```
<div class="conteneur">  
  <div class="one">Un</div>  
  <div class="two">Deux</div>  
  <div class="three">Trois</div>  
  <div class="four">Quatre</div>  
  <div class="five">Cinq</div>  
  <div class="six">Six</div>  
</div>
```

GRID - EXAMPLE

Un	Deux	
Trois		
		Quatre
	Cinq	Six

C'EST FINI

La documentation est indispensable pour avancer et s'améliorer

- spécification : <https://html.spec.whatwg.org/multipage/>
- Références, guides, tutoriaux :
 - [Mozilla Developer Network](#)
 - [W3 School](#)

