

Positionnement CSS

Le positionnement en CSS permet :

- de **placer** les éléments où l'on souhaite (en ligne ou colonne)
- de construire la **structure** de sa page web

<header>

<nav>

<main>

<section>

<article>

<article>

<article>

<section>

<aside>

<section>

<section>

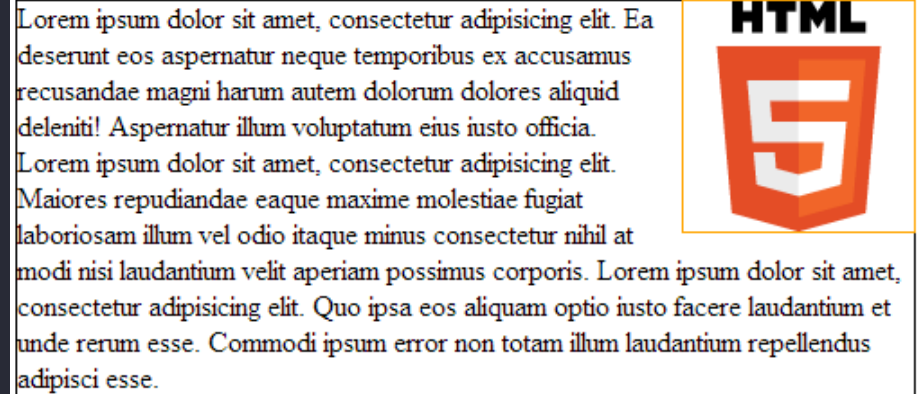
<footer>

Le flottant

Un flottant indique qu'un élément doit être retiré du flux normal et doit être placé sur le côté droit ou gauche **de son conteneur**

```
<article>  
    
  <p>Lorem ipsum dolor ...</p>  
</article>
```

```
article img {  
  float: right;  
}
```




Lorsqu'un élément est positionné en *"flottant"*, il sort du flux de la page, mais le contenu autour continue de tenir compte de sa présence.

Souci de débordement

Parfois, on souhaiterait pouvoir stopper l'écoulement après un élément précis

```
<article>
  
  <p>Lorem ipsum dolor ...</p>
</article>

<h2>Subtitle #2</h2>
<p>Lorem ipsum dolor ...</p>
```

| | |
|---|--|
| Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ea deserunt eos aspernatur neque temporibus ex accusamus recusandae magni harum autem dolorum dolores aliquid deleniti! Aspernatur illum voluptatum eius iusto officia. | HTML  |
| Subtitle #2 | |
| Lorem ipsum dolor sit amet, consectetur adipisicing elit. | |


Souci de débordement

La règle **clear** permet de stopper un écoulement

```
<article>
  
  <p>Lorem ipsum dolor ...</p>
</article>

<h2 class="clearfix">Subtitle</h2>
<p>Lorem ipsum dolor ...</p>
```

```
.clearfix {
  clear: both; ✓
}
```

| | |
|---|--|
| Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ea deserunt eos aspernatur neque temporibus ex accusamus recusandae magni harum autem dolorum dolores aliquid deleniti! Aspernatur illum voluptatum eius iusto officia. | HTML  |
| Subtitle #2 | |
| Lorem ipsum dolor sit amet, consectetur adipisicing elit. | |

Conclusion

Un flottant permet de disposer un élément à gauche ou droite dans le but que le contenu qui le suit s'écoule autour.

On ne doit pas s'en servir pour de la mise en page !
(d'autres modules sont là pour ça : *Flexbox*, *Grid*)

Inline-Block

Le mode de rendu *inline-block* est un mix des propriétés du *block* et du *inline*.

Il permet de placer des éléments côte-à-côte (inline), mais en ayant la possibilité de les dimensionner (block)

```

<div class="column">
  Lorem ipsum dolor ...
</div>
<div class="column">
  Lorem ipsum dolor ...
</div>
<div class="column">
  Lorem ipsum dolor ...
</div>

```

```

.column {
  display: inline-block;
  width: 300px;
}

```

| | |
|--|--|
| <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus suscipit et veritatis at iusto culpa nihil. Vel quia sed ad quaerat officiis iste maxime totam repellendus optio ipsa nostrum et! Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas explicabo tempora at delectus porro sunt eveniet alias aperiam numquam vero iusto natus error quis dolorum dignissimos ad veritatis architecto nemo?</p> | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus suscipit et veritatis at iusto culpa nihil. Vel quia sed ad quaerat officiis iste maxime totam repellendus optio ipsa nostrum et! Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sunt blanditiis quia incidunt asperiores ad fugiat officia?</p> |
|--|--|

Par défaut, les éléments s'affichent sur la *baseline*

Alignement vertical

La valeur **inline-block** de la propriété *display* permet l'utilisation d'une autre propriété pour **centrer verticalement les éléments**

```
1 .column {  
2   display: inline-block;  
3   width: 300px;  
4   vertical-align: top; /* "middle", "bottom" ou "baseline" */  
5 }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Accusamus suscipit et veritatis at iusto culpa nihil. Vel quia sed ad quaerat officiis iste maxime totam repellendus optio ipsa nostrum et!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Accusamus suscipit et veritatis at iusto culpa nihil. Vel quia sed ad quaerat officiis iste maxime totam repellendus optio ipsa nostrum et! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas explicabo tempora at delectus porro sunt eveniet alias aperiam numquam vero iusto natus error quis dolorum dignissimos ad veritatis architecto nemo?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Accusamus suscipit et veritatis at iusto culpa nihil. Vel quia sed ad quaerat officiis iste maxime totam repellendus optio ipsa nostrum et! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sunt blanditiis quia incidunt asperiores ad fugiat officia?

Alignement vertical

`vertical-align` ne fonctionne qu'avec
`display: inline-block`

(Sans cela il n'a aucun effet)

Relatif

Un positionnement relatif permet de déplacer un élément par rapport à sa position initiale

On utilise la propriété suivante :

```
position: relative;
```

La propriété `position` s'accompagne de 4 autres propriétés pour aider au positionnement :

- `top`
- `right`
- `bottom`
- `left`

Exemple

```
<p>Pas de doute, <strong>ce texte est important</strong>  
si on veut comprendre correctement le positionnement.</p>
```

```
1 strong {  
2   background-color: red;  
3   color: yellow;  
4   position: relative;  
5   left: 55px;  
6   top: 10px;  
7 }
```



On l'utilise généralement pour faire de légers ajustements.

Absolu

Un élément positionné en absolue est **retiré du flux de la page et peut être placé n'importe où**

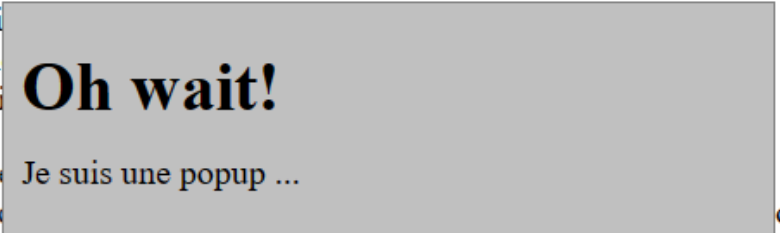
Il est très utile par exemple pour créer des *popups* qui se placent par dessus les autres éléments de la page

```
<div class="popup">
  <h1>Oh wait!</h1>
  <p>Je suis une popup ...</p>
</div>
```

```
1 .popup {
2   background-color: silver;
3   padding: 8px;
4   border: 1px solid gray;
5   position: absolute;
6   top: 100px;
7   left: 10%;
8   width: 80%;
9 }
```

Titre de la page

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nihil quis i
provi
animi

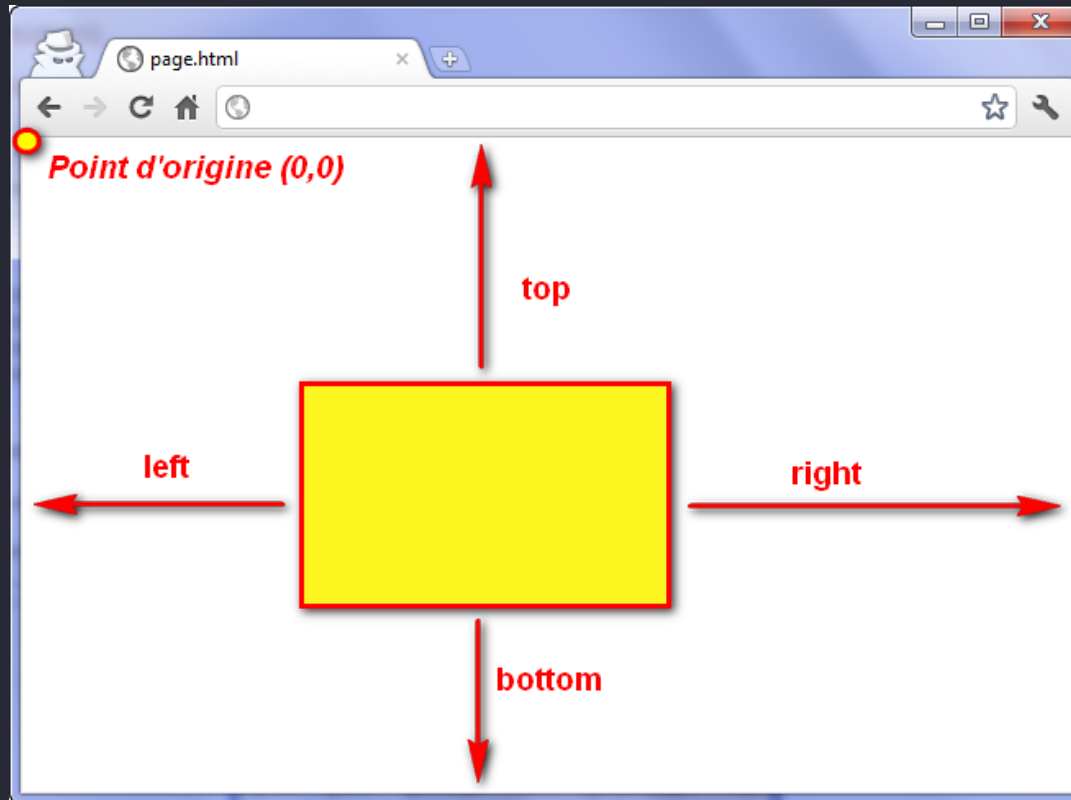


Oh wait!

Eius Je suis une popup ... o,
tempo
quod sed officia molestias nesciunt, eos, ipsum tenetur incidunt
cumque. Veniam, incidunt, optio.

Porro adipisci eveniet accusamus, earum recusandae alias, velit
nostrum magnam maxime quo iure sint culpa dolorem,
reprehenderit numquam nam saepe obcaecati soluta. Sint,
exercitationem. Natus eveniet atque debitis tenetur. Explicabo.

Une nouvelle fois, les propriétés **top**, **left**, **bottom**, **right** agissent sur la position de l'élément





Par défaut, un élément positionné en absolu se positionne **par rapport à son plus proche parent positionné en absolu ou relatif**

Sinon, il se positionne par rapport au **<body>**

Exemple

```
1 <div class="popup">
2   <h1>Hi there!</h1>
3   <p>Contenu de la popup ...</p>
4   <a class="close">✖</a>
5 </div>
```

```
1 .popup {
2   position: absolute;
3   ...
4 }
5
6 .popup .close {
7   position: absolute;
8   top: 0;
9   right: 0;
10 }
```

Titre de la page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sit

voluptas aspernatur consectetur adipiscing elit. Sit

consectetur adipiscing elit. Sit

Ad quod explicabo laboriosam consequuntur, culpa, molestias

explicabo laboriosam consequuntur, culpa, molestias

pariatur iste laudantium voluptatem?

Explicabo laboriosam consequuntur, culpa, molestias

assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Assumenda sapiente vitae iste. Distinctio maiores illum quam

Fixe

Un élément positionné en fixe fonctionne de la même manière qu'un élément absolu hormis le fait que le menu **reste fixe à l'écran lors du scroll**

```
.popup {  
  position: fixed;  
  ...  
}
```



Sticky

Ce style de positionnement est un mélange de *relative* et *fixed*

L'élément est considéré comme positionné de façon relative, jusqu'à ce qu'un seuil soit franchi **lors du scroll**

À partir de ce seuil, l'élément est positionné de façon fixe

Ce mode de positionnement peut être très pratique pour proposer un menu horizontal toujours visible à l'écran

```
<nav>
  <a href="#">Accueil</a>
  <a href="#">À propos</a>
  <a href="#">Contact</a>
</nav>
```

```
<main>
  ...
</main>
```

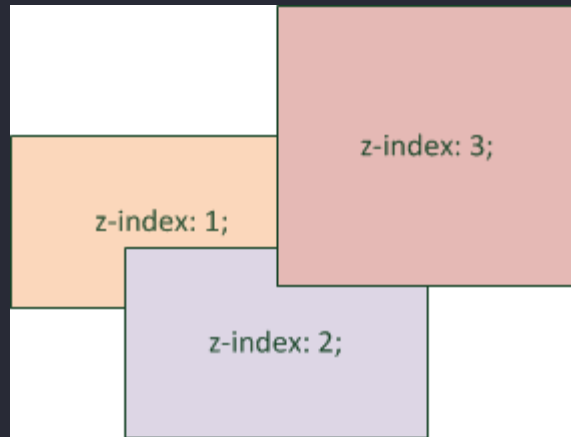
```
nav {
  position: sticky;
  top: 0;
}
```



z-index

Les positionnements que nous avons vus (*relatif*, *absolu*, *fixe* et *sticky*) permettent d'utiliser la propriété **z-index** pour gérer **l'ordre d'affichage**

```
#bloc-orange { z-index: 1; }  
#bloc-blue  { z-index: 2; }  
#bloc-red   { z-index: 3; }
```



La propriété prend une valeur numérique qui sert de référence pour l'ordre d'affichage.

Learn Layout



Apprendre les mises en page CSS

Apprendre les positionnements en CSS

 fr.learnlayout.com