



# Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

<http://cellierfranck.alwaysdata.net>



## MS LANGAGE & DEVELOPPEMENT POUR LE WEB

### LA PROPRIÉTÉ POSITION

#### A. POSITIONNEMENT

La propriété sert à définir la **position des éléments dans une page HTML**

**Héritage** : Il n'y pas de notion d'héritage pour les marges ([voir notion d'héritage sur ce lien](#))

**Syntaxe** : `position: valeur;`

Valeurs possibles:

1. `static`
2. `relative`
3. `absolute`
4. `fixed`

La propriété de feuille de style position **CSS** est généralement utilisée pour la création de calques (Layer).

Valeurs possibles:

1. `top`
2. `left`
3. `right`
4. `bottom`

**Exemple** : `position: absolute; top: 10px; right: 10px;`

Le principe du positionnement en CSS fait qu'il est possible de positionner un block n'importe.

#### Un exemple avec un titre H2

Prenons un exemple avec un titre H2 à 50px du haut et 20px de la gauche de la page HTML, le code CSS pourrait est le suivant : `h2 { position: absolute; top: 50px; left: 20px; }`

## B. LE POSITIONNEMENT ABSOLU

Une balise HTML avec une position absolue ne laisse aucun espace vide après qu'il est positionné. On place une balise HTML en position absolue en fixant la valeur de la propriété position à "absolute". On peut alors utiliser les propriétés right, left, top et bottom pour placer la balise HTML.

### Un exemple avec 4 div

```
#div_1 { position:absolute; top: 50px; left: 50px;}  
#div_2 { position:absolute; top: 50px; right: 50px;}  
#div_3 { position:absolute; bottom: 50px; right: 50px;}  
#div_4 { position:absolute; bottom: 50px; left: 50px;}
```

Dans cet exemple, vous allez obtenir une div à chaque coin de votre écran.

## C. LE POSITIONNEMENT RELATIF

On place une div en position relative en avec la valeur de la propriété : relative. La différence entre position absolue et position relative est à la façon dont la position est définie. Pour une div en position relative, elle est calculée d'après sa position originale dans la page.

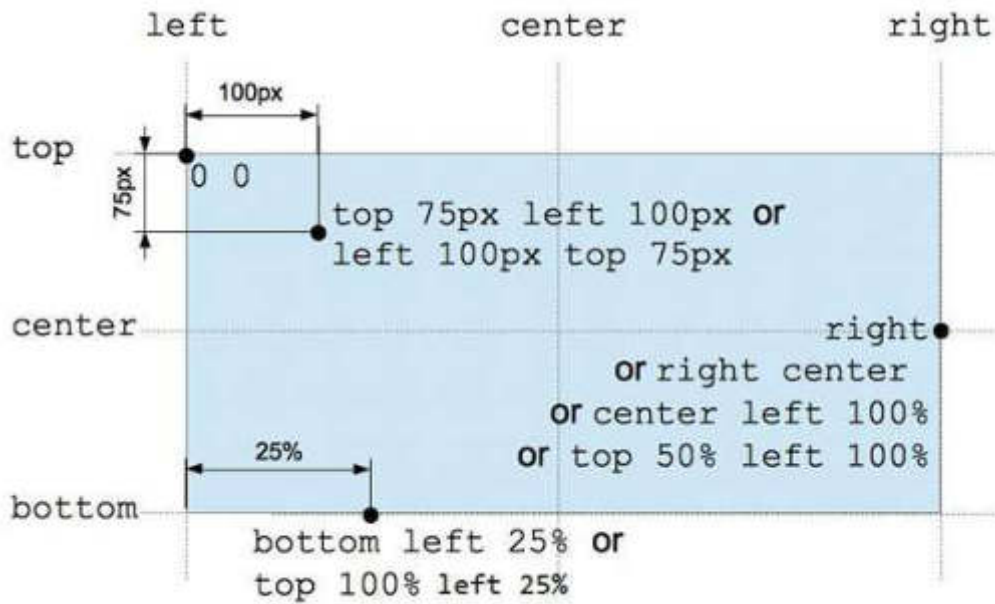
En d'autres termes, si on déplace l'élément vers la droite, la gauche, le haut ou le bas. Ainsi, l'élément obtient toujours un espace dans la page après qu'il est positionné. Comme exemple de positionnement relatif, nous pouvons positionner trois images relativement à leur position originale dans la page. Remarquez les espaces vides laissés par elles à leurs positions originales dans le document :

```
#div1 { position:relative; left: 35px; bottom: 15px;}  
#div2 { position:relative; left: 15px; bottom: 50px;}  
#div3 { position:relative; left: 5px; bottom: 70px;}
```

## D. UN PEU PLUS DE DETAIL

La propriété POSITION n'est pas simple à conceptualiser. Pour cela, il est nécessaire de rattacher à chaque valeur un schéma.

Schéma général :



## E. La position statique

La position statique est la **valeur par défaut** quand on ne spécifie rien dans cette propriété. L'élément statique ne fait que suivre la structure HTML pour pouvoir se positionner par rapport aux autres éléments. Comme vu plus haut, cela dépend de son type et d'où il est placé dans le code HTML.



En ne se basant que sur cette valeur, la façon dont on positionne nos éléments est assez limitée et pas pratique du tout. On peut toujours jouer sur la propriété `display` qui permet de changer le type d'une balise ou sur la propriété `float`, mais cela est peu pratique.

De plus, la valeur statique ne permet pas l'utilisation des autres propriétés de position comme `z-index`, `top`, `bottom`, `left` et `right` ce qui limite d'autant plus les possibilités.

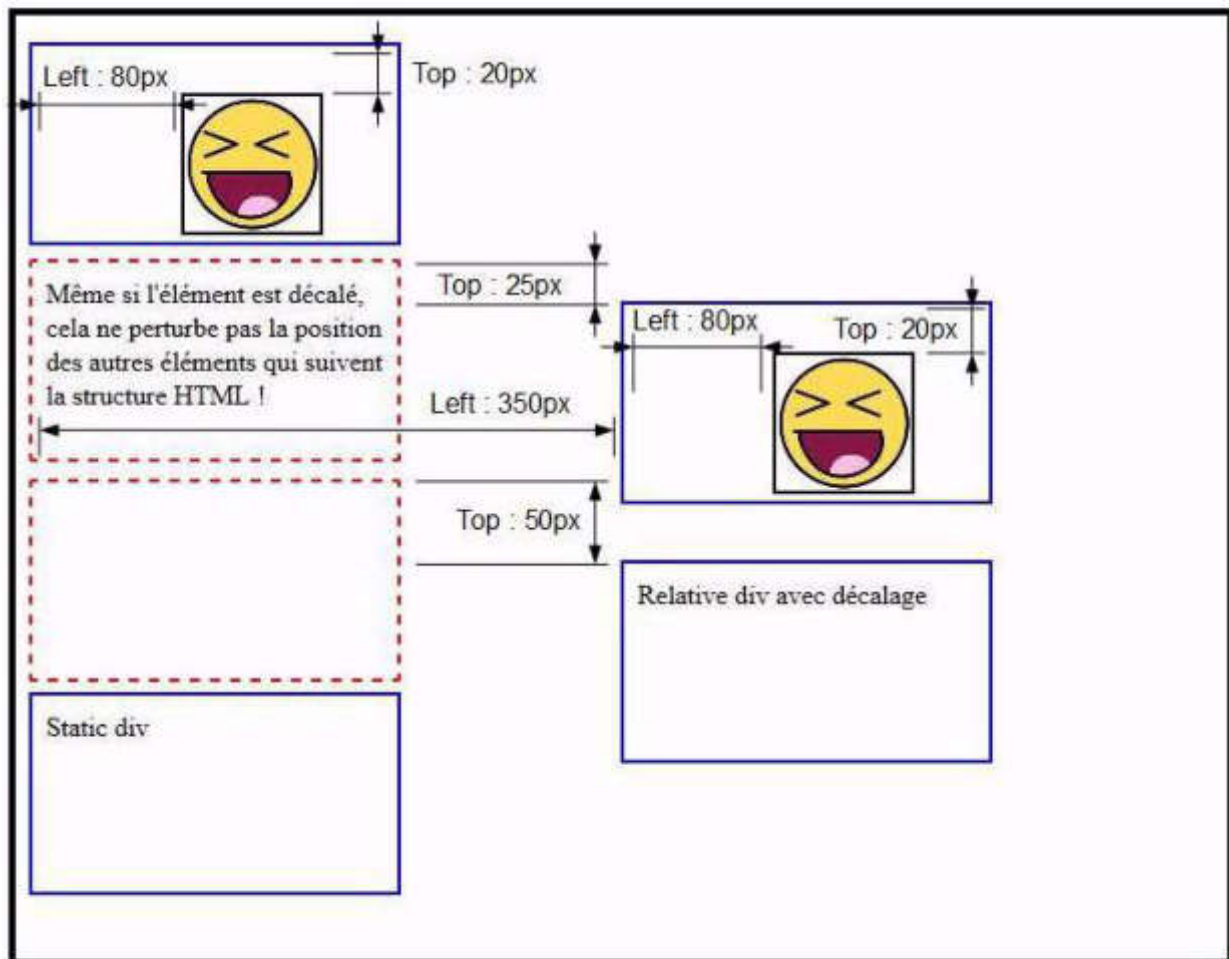
Vous n'aurez normalement pas à spécifier cette valeur très souvent, mais voilà comment on fait :

```
1 position: static
```

## F. La position relatif

L'élément relatif se **comporte exactement comme la position statique**, elle prend en compte le type de balise et la hiérarchie. Mais cette valeur rajoute la possibilité de déplacer un élément grâce aux **quatre propriétés de position** qui est `top`, `left`, `right` et `bottom`.

Comme son nom l'indique, la position d'un élément qui est relatif **dépend de la position de son élément parent sur l'écran**. Un décalage de 50px vers le haut correspondra toujours au même décalage par rapport à son parent. Mais si l'élément père se déplace de l'écran, alors l'élément relatif **se déplacera lui aussi avec son père**, tout en conservant son décalage de 50px avec lui !



Un autre concept à saisir pour comprendre les positions est le concept de **référence**. En utilisant les quatre propriétés de position, le décalage sur l'écran ne sera pas pareil en fonction de quel élément est la référence. Dans le cas d'un élément en position relative, c'est **l'élément parent qui est la référence**. Le décalage se fait donc en fonction de la position de la référence sur l'écran.

Un autre truc à comprendre c'est que si un élément en position relative est déplacé, cela ne perturbera pas la position des autres éléments de la page web.

Voilà comment on met un élément en position relatif :

```
1 position:relative
```

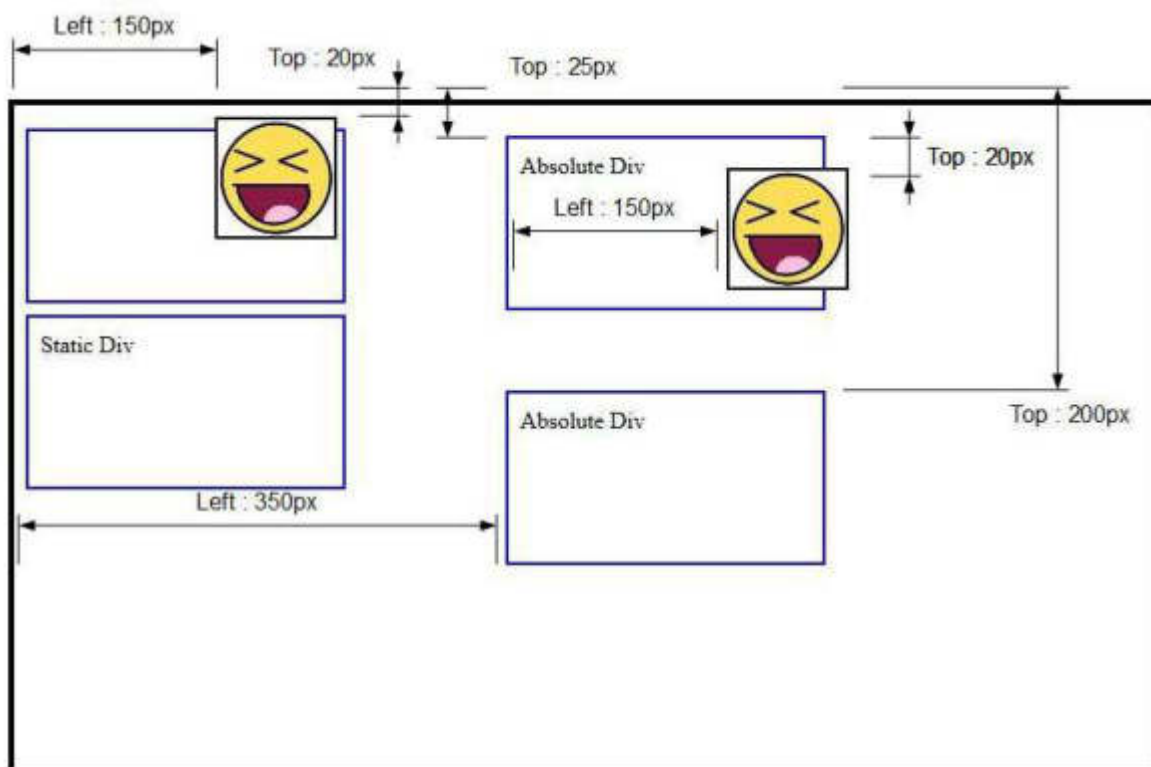
## G. La position absolue

La position absolue se comporte exactement comme la position relative, mais a **deux différences** notables.

La 1ere est que comparée à la position relative, l'élément en absolu **sort de la structure HTML** au niveau du positionnement. C'est comme s'il n'existait plus pour les autres éléments qui viendront s'adapter en conséquence. Du coup, tous les éléments absolus se retrouveront

dans le coin en haut à gauche de sa référence et s'empileront les uns sur les autres si aucune des quatre propriétés de positions n'est spécifiée.

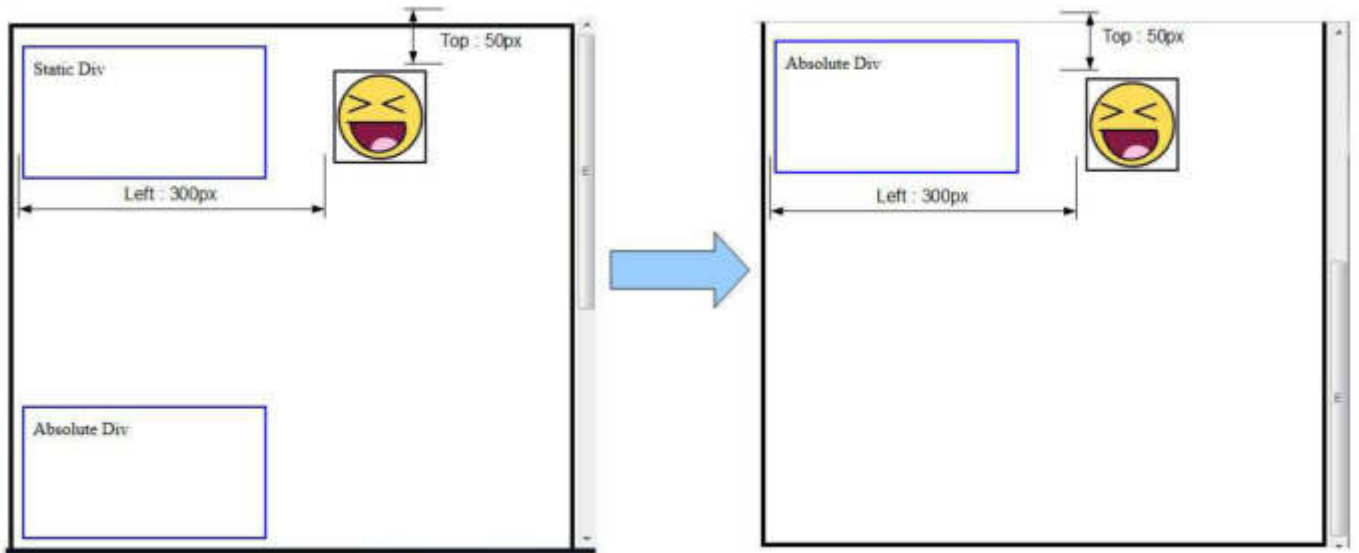
La 2e c'est que **sa référence sera toujours la balise `body`**, sauf si son élément parent à une position **autre que statique**, dans ce cas alors, ça sera **son élément parent la référence**.



1 position: absolute

## H. La position fixe

La position fixe **partage les mêmes caractéristiques que la position absolue**, mais avec une différence subtile. La référence n'est pas son élément père, ni la page web, mais **la fenêtre de votre navigateur !**



L'élément reste dans un endroit fixe de la page web et n'y bouge pas, même si vous défilez la page vers le haut ou le bas comparé aux autres positions.

1 `position: fixed`

## I. La position héritée ...

Elle permet d'hériter la position de son élément père, voilà !