http://taistoietcode.com/?p=34

Positionnement css

Posted on: octobre 21st, 2013 *by* R&L No Comments

Il existe en css 5 types de positionnement de bloc différents :

1_ par défaut qui ne nécessite aucune déclaration particulière dans le css,

2_ relatif qui permet d'ajouter un décalage de position sans affecter les autres éléments de la page,

3_ absolu, qui permet de spécifier la position d'un élément par rapport à son conteneur,

4_ fixe, qui permet de spécifier la position d'un élément par rapport à la fenêtre du navigateur,

5_ float qui permet aux éléments suivants le bloc positionné de se placer à sa gauche ou sa droite.

Nous allons détailler ces types de positionnement en utilisant toujours la même structure html :

```
1     <html>
2         <head>
3             <link href="css/styles.css" rel="stylesheet" type="text/css" />
4         </head>
5         <body>
6             <div id="global">
7                 <div id="bloc1"></div>
8                 <div id="bloc2"></div>
9                 <div id="bloc3"></div>
10            </div>
11        </body>
12
13    </html>
```

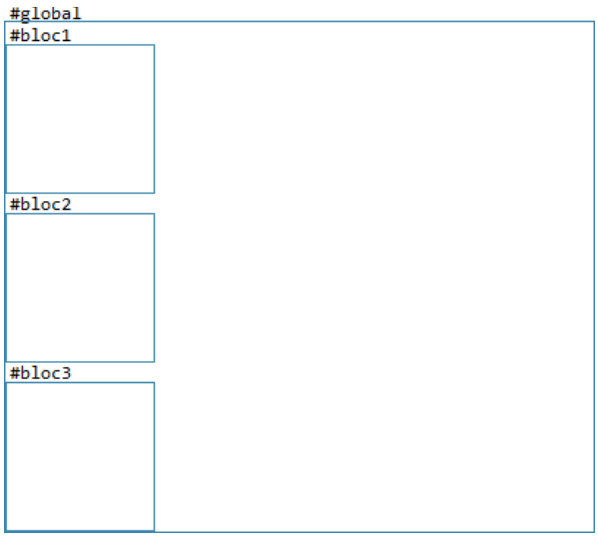
qui contient un bloc « global » englobant trois blocs « bloc1 », « bloc2 », « bloc3 ».

1. Positionnement par défaut

Avec le positionnement par défaut, les éléments de type bloc d'une page html, tel que les éléments se placent les un en dessous des autres, dans leur ordre d'apparition au sein du code html. Le code css suivant :

```
1#global div{
2    width:100px;
3    height:100px;
4}
```

où l'on définit uniquement une taille aux éléments contenus dans l'élément dont l'id est « global », appliquera la mise en forme suivante à la page html :



Les blocs se placent dans ce que l'on appelle le flux de la page. Les autres types de positionnement sortent les éléments de ce flux par défaut.

2. Positionnement relatif

Le positionnement relatif se spécifie dans le code css grâce à la déclaration

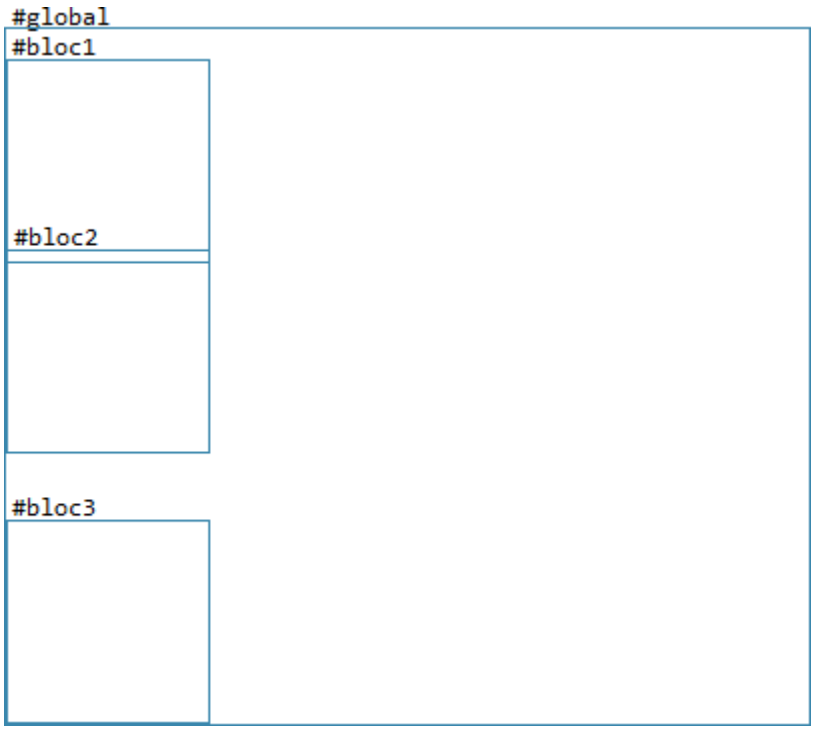
```
position:relative;
```

La seule différence entre le positionnement relatif et le positionnement par défaut est que le positionnement relatif permet de « fixer » le bloc dans le flux. L'affichage place d'abord le bloc dans le flux, puis, s'il y a des déclarations de positions supplémentaires dans les règles css qui lui sont attribuées, il ajoute ces positions à sa position actuelle dans le flux. Le positionnement relatif peut également servir à positionner des éléments en absolu à l'intérieur d'un autre élément (voir plus bas, positionnement absolu).

Le code css

```
#global div{
    width:100px;
    height:100px;
}
#global #bloc2{
    position:relative;
    bottom:20px;
}
```

décalera le deuxième bloc de 20px vers le haut à partir de sa position dans le flux et appliquera la mise en forme :



On peut constater que le décalage opéré sur l'élément dont l'id est « bloc2 » n'influe pas le positionnement des éléments qui le suivent dans le flux.

3. Positionnement absolu

Le positionnement absolu se spécifie dans le code css grâce à la déclaration

```
position:absolute;
```

Et permet à l'élément d'être placé à des coordonnées précises par rapport à son bloc conteneur.

Le point d'origine de ces coordonnées se spécifie grâce aux propriétés css

```
bottom
top
left
right
```

Dans le code css suivant, nous allons attribuer des positions absolues aux éléments se trouvant à l'intérieur de l'élément dont l'id est « global ». Le premier bloc (#bloc1) se placera en haut à gauche du conteneur (#global), le second (#bloc2) en haut à droite et le troisième (#bloc3) en bas à droite. Pour placer certains blocs (#bloc2 et #bloc3) en prenant comme références le côté inférieur et le côté droit du conteneur (#global), il va falloir donner des dimensions précises à celui-ci.

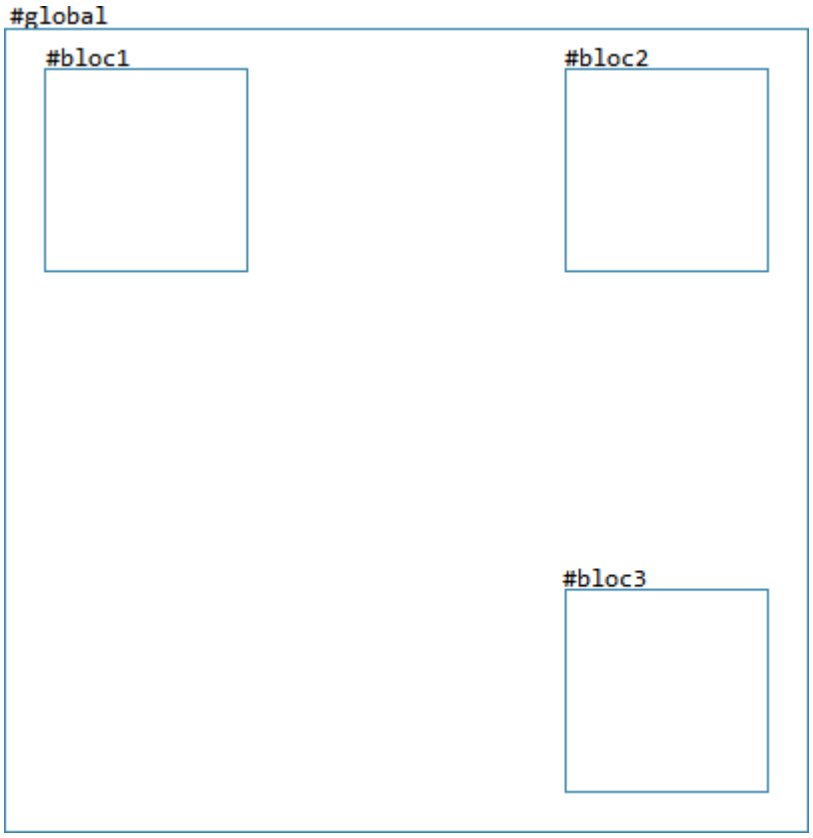
```
/*le conteneur a des dimensions précises et est "fixé" dans le flux
grâce au positionnement relatif*/
#global{
    width:400px;
    height:400px;
    position:relative;
}
/*tous les div contenus dans le conteneur #global se positionneront
en absolu et auront des dimensions précises*/
#global div{
    width:100px;
    height:100px;
    position:absolute;
}
/*le div #bloc1 se positionne à 20 pixels du bord supérieur de #global
et 20 pixels du bord gauche*/
#global #bloc1{
    top:20px;
    left:20px;
}
/*le div #bloc2 se positionne à 20 pixels du bord supérieur de #global
et 20 pixels du bord droit*/
#global #bloc2{
    top:20px;
    right:20px;
}
/*le div #bloc3 se positionne à 20 pixels du bord inférieur de #global
et 20 pixels du bord droit*/
#global #bloc3{
    bottom:20px;
    right:20px;
}
```

La déclaration

```
position:relative;
```

qui est attribuée à l'élément conteneur (#global) est nécessaire pour que les éléments qu'il contient se placent en prenant comme référence sa position. Sans cette déclaration, ces éléments se placeront par rapport à l'élément qui contient le conteneur (#global), dans ce cas, l'élément <body>.

Le code css appliquera donc la mise en forme :



5. Positionnement fixe

Le positionnement fixe se spécifie dans le css grâce à la déclaration

```
position:fixed;
```

et permet à l'élément d'être positionné par rapport à la fenêtre du navigateur. Un élément fixe dans la fenêtre ne disparaitra pas en suivant le scroll du navigateur, il restera toujours visible et au même endroit de la fenêtre. Ce positionnement peut-être utilisé pour placer un menu dans une page, qui sera de fait toujours visible. Ce positionnement n'est pas pris en compte par les smartphones et tablettes, ce qui peut engendrer des problèmes d'affichage sur ces appareils.

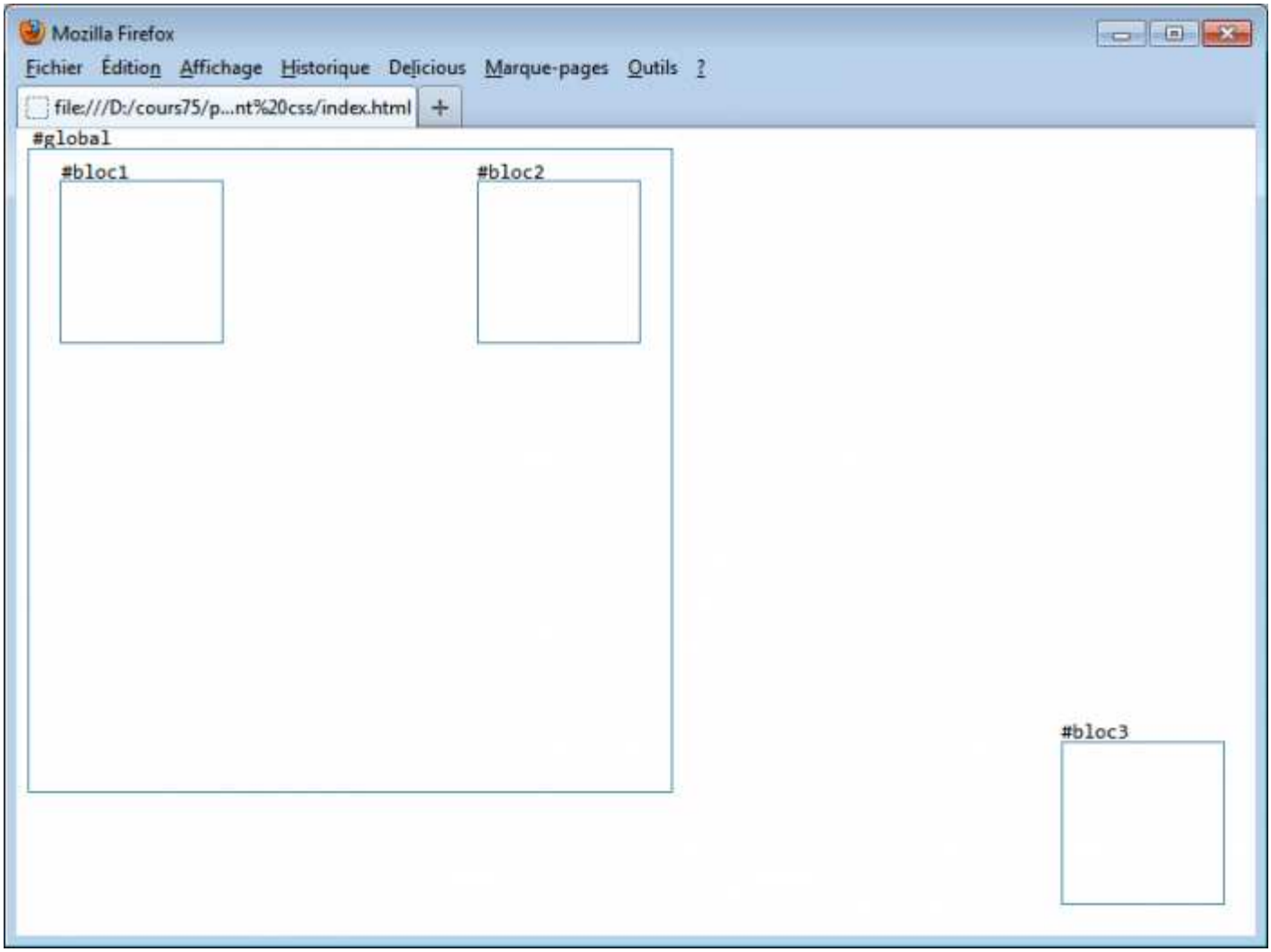
le code css suivant :

```
#global{
    width:400px;
    height:400px;
    position:relative;
}

#global div{
    width:100px;
    height:100px;
    position:absolute;
}

#global #bloc1{
    top:20px;
    left:20px;
}
#global #bloc2{
    top:20px;
    right:20px;
}
#global #bloc3{
    position:fixed;
    bottom:20px;
    right:20px;
}
```

où l'on précise que le bloc dont l'id est « bloc3" est en position fixe, sortira l'élément (#bloc3) de son conteneur et le placera non plus par rapport à celui-ci mais par rapport au coin inférieur droit de la fenêtre du navigateur.


Mozilla Firefox
Fichier Édition Affichage Historique Delicious Marque-pages Outils ?
file:///D:/cours75/p...nt%20css/index.html
#global
#bloc1
#bloc2
#bloc3

6. Positionnement flottant

Le positionnement flottant se spécifie grâce à la déclaration

```
float:left;
```

ou

```
float:right;
```

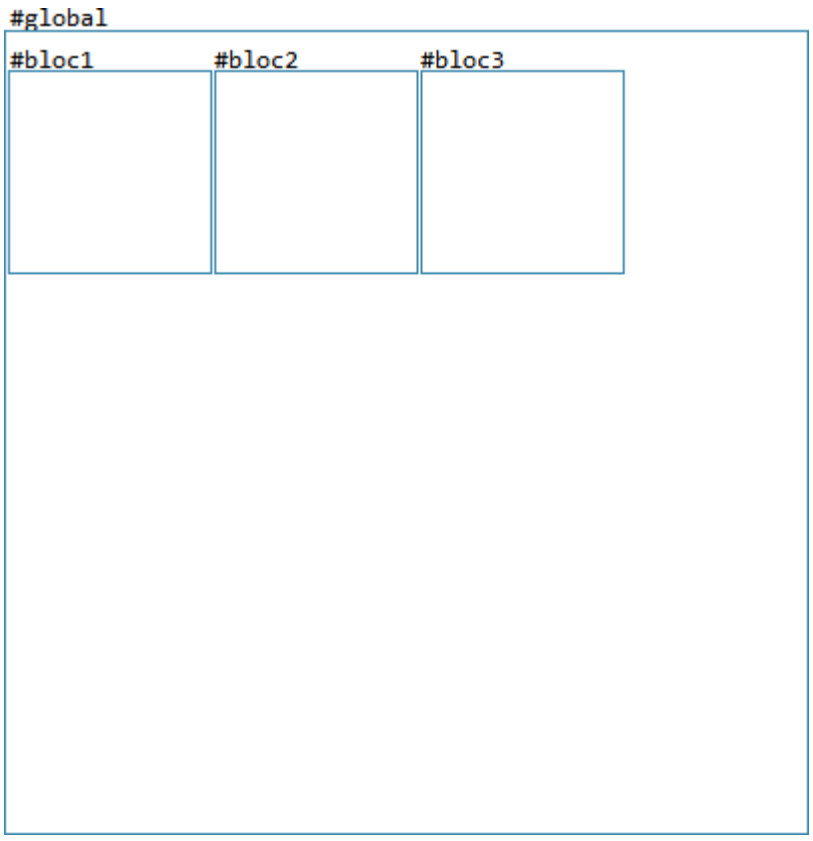
Il indique dans le premier cas que l'élément se positionnera à gauche de l'élément suivant et dans le second qu'il se positionnera à droite de l'élément suivant. Si le conteneur n'est pas assez large pour que les deux éléments se mettent côte à côte, le deuxième élément s'affichera en dessous du premier.

Le code css suivant :

```
#global{
    width:400px;
    height:400px;
    position:relative;
}

#global div{
    width:100px;
    height:100px;
    margin-top:10px;
    float:left
}
```

indique que tous les éléments contenus dans global tenteront de se placer les uns à côté des autres.



Lorsque l'on veut insérer des éléments en dessous d'une ligne d'éléments positionnés en float, il faut indiquer dans l'html que l'on reprend le cours du flux de la page, que l'on sort du positionnement float. Cette indication correspond à la déclaration css

```
clear:left;
```

ou

```
clear:right;
```

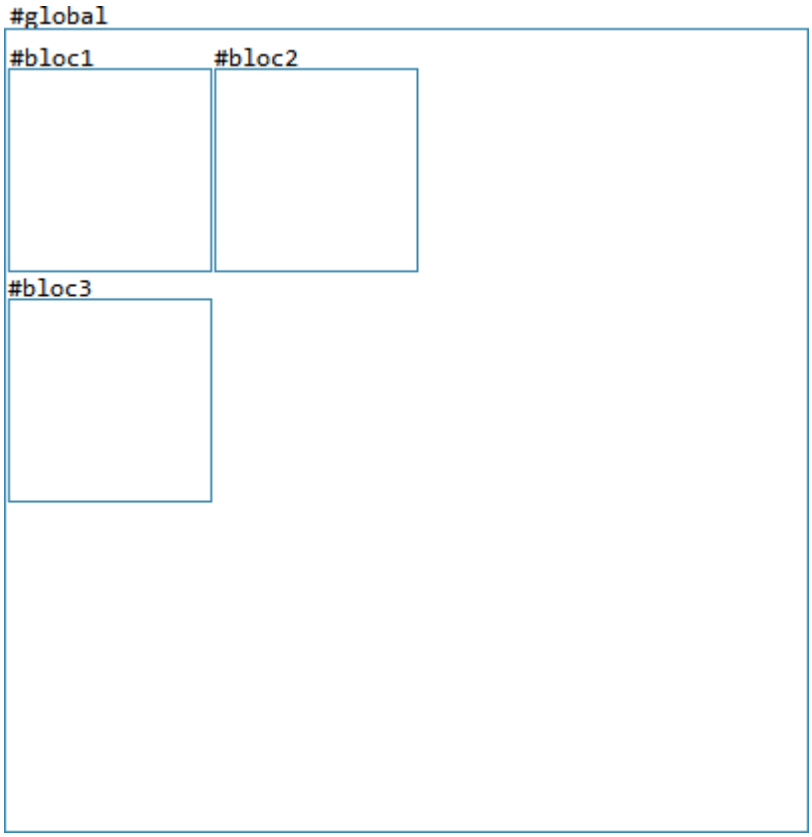
ou

```
clear:both;
```

selon le positionnement flottant que l'on veut désactiver, qui se réalisait via float:left (on utilise clear:left) ou float:right (on utilise clear:right), ou si l'on veut désactiver tous les positionnement flottants pour reprendre l'affichage en dessous (on utilise clear:both). Cette propriété doit être attribuée à l'élément qui suit les éléments positionnés en float, donc l'élément à partir duquel il faut passer à la ligne.

```
1     #global{
2         width:400px;
3         height:400px;
4         position:relative;
5         clear:both;
6     }
7
8     #global div{
9         width:100px;
10        height:100px;
11        margin-top:10px;
12        float:left
13    }
14    #bloc3{
15        clear:left;
16    }
```

Ici, le troisième bloc ne s'affichera donc plus à côté du deuxième mais en dessous :



Une autre façon de résoudre ce problème est de se créer un « outil » sous forme de balise à insérer dans la page html à chaque fois qu'il faut revenir à la ligne après des positionnements flottants. Pour ce faire, il faut modifier la page html en insérant cette balise aux endroits où il faut passer à la ligne, donc pour notre exemple :

```
<html>
    <head>
        <link href="css/styles.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <div id="global">
            <div id="bloc1"></div>
            <div id="bloc2"></div>
            <br class="clear" />
            <div id="bloc3"></div>
        </div>
    </body>

</html>
```

où la balise

```
<br class="clear" />
```

sert de passage à la ligne et fonctionne grâce à une classe css (voir article « sélecteurs css ») que l'on ajoute dans la feuille de style, qui devient :

```
#global{
    width:400px;
    height:400px;
    position:relative;
    clear:both;
}

#global div{
    width:100px;
    height:100px;
    margin-top:10px;
    float:left
}
.clear{
    clear:both;
}
```

La première solution est certainement la plus « correcte » dans le sens où l'on garde la mise en forme strictement dans la feuille de style et la structuration du contenu dans la page html. La deuxième solution implique d'insérer un élément de mise en forme directement dans la page html (notre <br class= »clear » />). Elle peut malgré tout être utile dans certains cas.