Les positionnements en CSS.

Un des aspects important mais assez délicat du travail de l’intégrateur est le positionnement des éléments. C’est pour cette raison que je vais tenter d’éclaircir ce point ici.

Pour ce faire, je vais prendre l’exemple du header de Webtuts.

Il faut distinguer trois propriétés CSS ici.

* Display : [ block – none – inline – inline-block ]
* Position : [ static – absolute – relative – fixed ]
* Float : [ none – left – right ]

La première propriété ( display ) permet de donner à l’élément son rôle dans la page. Lui assigner « block » lui imposera de prendre par défaut toute la largeur de son conteneur. L’élément pourra aussi implémenter les propriétés « width » (largeur de l’élément si elle doit être autre que la largeur du conteneur), « height » (la hauteur de la div), margin (marge extérieur de l’élément) et padding (marge intérieur de l’élément). A savoir, tel quel, un élément de type « block » empêchera l’alignement horizontal.

Lui assigner « none » « effacera » en quelque sorte votre élément. Il sera bien présent dans le DOM mais vous ne le verrez pas et les autres éléments agiront comme s’il n’existait pas.

Lui assigner « inline » permettra d’aligner plusieurs éléments sur la même ligne mais dans ce cas, il perdra toutes les propriétés propres aux éléments de type block énoncés plus haut (width, height, margin et padding).

Quant au petit dernier, « inline-block » est un type hybride entre « inline » et « block » (Cap’tain obvious !), apparu au début du CSS3. Il permet de combiner le comportement de la valeur « inline » tout en acceptant les propriétés de « block ». Génial n’est-ce pas ? Mais il y a un hic… (Il y en a toujours un…). Comme toute propriété du CSS3, « inline-block » est mal interprété par certain navigateurs (pour ne citer personne…).

Mais alors comment faire pour aligner des éléments de type block sans utiliser cette dernière valeur me direz-vous ? (Dites-le !)

Eh bien j’y viens justement…

La deuxième propriété (float) s’avère d’une grand aide lorsqu’il est question d’alignement de plusieurs éléments. Par défaut, la valeur est « none » et n’implique donc aucun comportement supplémentaire. Ainsi, lui assigné « left » mettra l’élément le plus à gauche du conteneur. Et « right »… (Devinez ?) Le placera le plus à droite !

Trois choses importantes à savoir sur cette propriété… La première est que si un conteneur contient plusieurs éléments en float, ils seront adjacents dans leur ordre de déclaration du DOM. Ainsi :

<style>

div {

float: left;

width: 150px;

height: 150px;

}

.green{

background: green;

}

.pink {

background: pink;

}

.blue {

background: blue;

}

</style>

<div class="green">

</div>

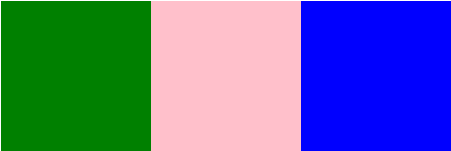
<div class="pink">

</div>

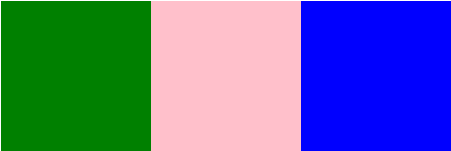
<div class="blue">

</div>

Ressemblera à ceci :



Tandis que remplacer « left » à la propriété « float » par « right » donnera ceci :



Ensuite, il faut savoir qu’un élément en « float » ne pourra pas recevoir la propriété « margin :auto » (si vous ne savez pas ce que c’est laissez ça pour plus tard).

La dernière chose à savoir est qu’un conteneur ne prendra pas en compte la hauteur de ses éléments fils étant en « float ». Ceci est assez gênant mais il y a une solution vous vous en doutez bien… Il suffit de mettre

<div style="clear:both">

Juste après la fin de vos éléments en « float » et le tour est jouée.

Cette propriété est bien pratique pour insérer des images au milieu d’un texte ou même pour le responsiv design, mais ça c’est pour plus tard !

Ce qui nous amène à notre dernière propriété… « Position » !

Cette propriété prend par défaut la valeur « static ».

Lui assigner « absolute » retire votre élément du flux du DOM. Pratique lorsque vous souhaitez faire chevaucher des éléments par exemple. Dans le cas d’un positionnement en « absolute », votre élément se mettra par défaut dans le coin haut gauche de votre premier parent étant en « position » « relative » et il se positionnera via les propriétés « top » et « left » ou « right »

Et position « relative » c’est quoi exactement ?

Eh bien en position relative, un élément peut être décalé verticalement et/ou horizontalement via les propriétés « top », « left » ou « right », contrairement aux flux normal du DOM. De plus, il sert de référence aux fils étant en position « absolute »