Séquence de cours

WEB ET WEB MOBILE

Séquence VI: Superposition de spécifications de style

Objectifs pédagogiques

Dans ce cours vous apprendrez à :

- comprendre l'ordonnancement des bases de styles

Plan

1. Le float
2. Positionnement absolu et relatif
3. Interaction avec le fond et positionnement flottant

A retenir

I. FLOT

Sans les spécifications de positionnement CSS, le webmaster était obligé de recourir à des expédients html, ce qui mélange l'information et la structure avec la mise en forme.

Ce qu'il faut comprendre de la notion de Flot c'est que les informations qui constituent une page forment un flot qui est la suite des conteneurs de type bloc à mesure qu'ils sont lus par le navigateur.

Ce flot est d'ailleurs à plusieurs niveaux car, dans un conteneur de plus haut niveau, il y a des conteneurs secondaires et des éléments et ainsi de suite.

Lorsqu'on ne spécifie aucun positionnement, les conteneurs successifs de type bloc viennent les uns en dessous des autres.

Dans un conteneur, les éléments en ligne successifs viennent de gauche à droite sur la ligne et on passe au-dessus lorsque la largeur du conteneur va être dépassée .

Dans cette succession, les deux conteneurs sont séparés non pas par la somme marge basse du premier plus marge haute du deuxième mais la plus grande des deux;cela s'appelle la fusion des marges.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Flot</title>
    <style type="text/css">
        p{ border: : 1px dashed black; }
        #d1 p {margin: 50px 10px 30px 10px;}
        #d2 p { margin: 20px 10px 30px 10px; }
        div { border: 2px solid black; }
    </style>
</head>
<body>
    <div id="d1">
        <p>Ces paragraphes sont séparés par la plus grande.</p>
        <p> de leur marge haute (50px) et basse (30px)</p>
    </div>
    <div id="d2">
        <p>Ici marge basse (30px) > haute (20px)</p>
        <p>donc vous aurez séparation=basse</p>
    </div>
</body>
</html>
```

Avec des conteneurs emboîtés, les choses sont plus complexes: l***es marges latérales ne fusionnent jamais ; les marges horizontales fusionnent.*** Le rendu était différent avec les anciennes versions d'Internet Explorer, mais Internet Explorer 9 + se conforme aux autres navigateurs.

Dans l'exemple ci-dessous ci-après, la **div** extérieur sert de conteneur global. Si nous imposons une bordure à la div milieu, les marges intérieures et de milieu ne sont plus en contact, donc il n'y a aucune fusion, on ajoute la règle ***border:1px solid black;*** à la **div milieu** et on constate que les marges sont bien distinctes.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flots conteneur emboîtés</title>
    <style type="text/css">
        #exterieur { border: 1px dashed black; }
        #milieu{ margin: 20px 20px 20px 20px; background-color: silver; }
        #interieur{ margin: 1px dashed black; background-color: white;  }
    </style>
</head>
<body>
    <div id="exterieur">  <div id="milieu">      <div id="interieur">
    <p>Contenu du paragraphe</p>
    </div> </div> </div>
</body>
</html>
```

Pour les éléments en-ligne (**inline**), il n'y a que les marges latérales et il n'y a jamais de fusion.

Propriété display

- Appliquée à un **élément en ligne(inline)**, la règle ***display : block ;*** fait de cet élément un élément bloc, donc capable d'obéir à des styles bloc et les éléments successifs sont affichés les uns en dessous des autres.

- Appliquer à un élément bloc, la règle ***display : inline ;*** le transforme en élément en ligne, donc éléments successifs afficher les uns à côté des autres. Il en résulte que la frontière entre éléments bloc (in bloc) et en ligne (inline) est très floue d'autant que la propriété display admet aussi la valeur ***online-block*** : éléments successifs affichés les uns à côté des autres mais redimensionnables.

La propriété display admet aussi comme valeur ***list-item*** qui fait comme si l'élément avait la balise <li>, et ***none*** qui fait que l'élément n'est pas affiché (comme s'il était absent de flot).

CSS3 ajoute la valeur ***flexbox*** ; on associe les paramètres :

flex–direction (sens d'affichage,

- **lr** de gauche à droite,
- **rl** l'inverse,
- **tb** de haut en bas,
- **bt** l'inverse,
- **inline,**
- **block,**

etc.),

flex-order (ordre d'affichage),

flex-pack (distribution, start, end, center, justify) et

flex-align (alignement vertical, auto, vaseline).

Exemple:

#page{display : flexbox; flex-direction:bt;}

II. POSITIONNEMENT ABSOLU ET RELATIF

- **Utiliser les positions absolues**

Le positionnement absolu a des désavantages du fait de sa rigidité : il ne permet pas (ou difficilement) le centrage du site, ni son adaptation aux différentes résolutions la plupart du temps.

Sachez cependant que c'est le seul moyen de superposer deux blocs (avec la propriété z-index).

Lorsqu'il est en position **absolue**, le bloc est dit "positionné". Il est retiré du "flux" du code html : son positionnement sera le même quelle que soit l'emplacement de la balise dans le conteneur (défini par les propriétés **top** et **left** par rapport au coin supérieur gauche du conteneur). Un des inconvénients est que le contenu dans son conteneur peut passer par dessous (le DIV étant retiré du flux).

Le bloc est placé par rapport à son conteneur s'il est lui-même positionné, ou alors par rapport à la page entière (body).

** ... **

- **Utiliser les positions relatives**

Le positionnement relatif permet d'inscrire un contenu en flux normal, ***puis de le décaler horizontalement et/ou verticalement***. Le contenu suivant n'est pas affecté par ce déplacement, qui peut donc entraîner des chevauchements.

Par exemple

Ce texte est normal mais celui-ci est dans un span et retour au normal

Dans ce dernier cas, le span a comme style :

"position:relative; bottom:6px; left:25px; background-color:gold;".

Vous pouvez remarquer que si le point d'insertion de la boîte SPAN est placé correctement dans la suite de la phrase et qu'il y a un espace correspondant à la largeur du texte dans le SPAN, il y a en plus un déplacement relatif à ce point d'insertion (et correspondant aux propriétés **bottom** et **left**) qui provoque alors un chevauchement du texte.

III. INTERACTION AVEC LE FOND ET POSITIONNEMENT FLOTTANT

- **Interaction avec le fond**

Positionnement fixe

Le positionnement fixe est semblable au positionnement absolu sauf qu'ici, le bloc englobant correspond au *viewport (C'est une zone d'affichage qui représente la zone actuellement visible sur l'appareil. Pour un navigateur web, la zone d'affichage correspond généralement à la fenêtre du navigateur sans les éléments d'interface du navigateur (barre de menu, etc.). Bref, sur le Web, la zone d'affichage correspond la plupart du temps à la région à l'intérieur de la fenêtre dans laquelle vous consultez un site ou une application.)* si aucun ancêtre de l'élément ne possède une propriété ***`transform`, `perspective` ou `filter`*** qui est différente de `none`. On utilise souvent ce positionnement pour créer un élément flottant qui reste à la même position, même lorsqu'on fait défiler la page.

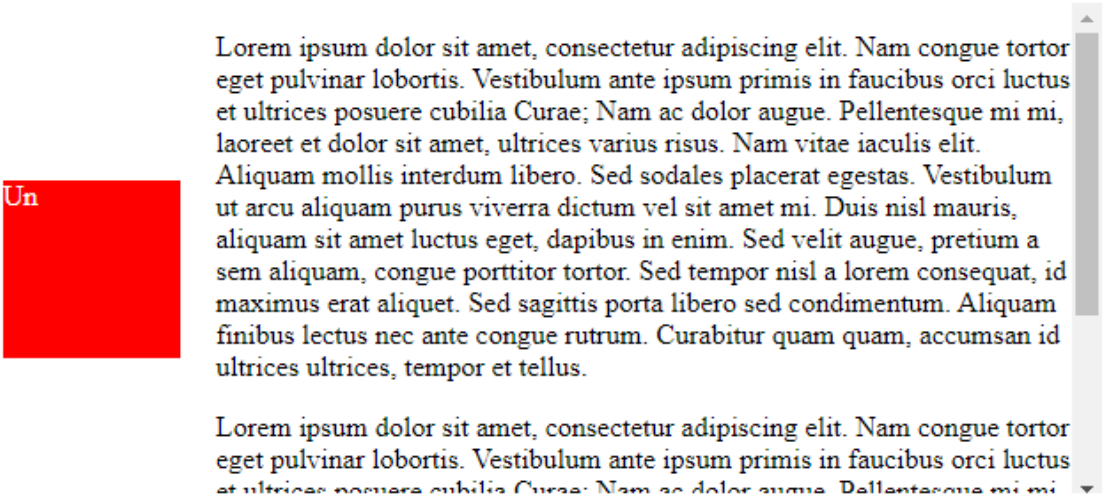
Dans l'exemple qui suit, la boîte « Un » est fixée à 80 pixels du haut de la page et à 20 pixels du bord gauche.

```
</body>
 <div class="outer">
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam congue tortor eget pulvinar lobortis.
    Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nam ac dolor augue.
    Pellentesque mi mi, laoreet et dolor sit amet, ultrices varius risus. Nam vitae iaculis elit.
    Aliquam mollis interdum libero. Sed sodales placerat egestas. Vestibulum ut arcu aliquam purus viverra dictum vel sit amet mi.
    Duis nisl mauris, aliquam sit amet luctus eget, dapibus in enim. Sed velit augue, pretium a sem aliquam, congue porttitor tortor.
    Sed tempor nisl a lorem consequat, id maximus erat aliquet. Sed sagittis porta libero sed condimentum.
    Aliquam finibus lectus nec ante congue rutrum. Curabitur quam quam, accumsan id ultrices ultrices, tempor et tellus.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam congue tortor eget pulvinar lobortis.
    Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nam ac dolor augue.
    Pellentesque mi mi, laoreet et dolor sit amet, ultrices varius risus. Nam vitae iaculis elit.
    Aliquam mollis interdum libero. Sed sodales placerat egestas. Vestibulum ut arcu aliquam purus viverra dictum vel sit amet mi.
    Duis nisl mauris, aliquam sit amet luctus eget, dapibus in enim. Sed velit augue, pretium a sem aliquam, congue porttitor tortor.
    Sed tempor nisl a lorem consequat, id maximus erat aliquet. Sed sagittis porta libero sed condimentum.
    Aliquam finibus lectus nec ante congue rutrum. Curabitur quam quam, accumsan id ultrices ultrices, tempor et tellus.
  </p>
  <div class="box" id="un">Un</div>
</div>
```

```
<style type="text/css">
            .box {
              background: red;
              width: 100px;
              height: 100px;
              margin: 20px;
              color: white;
            }
            #un {
              position: fixed;
              top: 80px;
              left: 10px;
            }
            .outer {
              width: 500px;
              height: 300px;
              overflow: scroll;
              padding-left: 150px;
            }
</style>
```

CSS

Lorsqu'on regarde le haut de la page, la boîte apparaît en haut à gauche, même après avoir défilé, elle reste à la même place par rapport au viewport :



- **Positionnement flottant**

La propriété **FLOAT** permet de positionner un bloc à gauche ou à droite dans un conteneur (et non plus l'un en dessous de l'autre). Le reste du conteneur occupera alors l'espace laissé libre.

Comme le float sort du flux courant, il n'est pas compté dans le calcul de la hauteur du conteneur, si celle-ci n'est pas spécifiée.
L'utilisation courante consiste à aligner une image à gauche ou à droite d'un texte de contenu :

```
<div class="conteneur">
        <p>Ici se trouve mon texte.</p>
        <img class="gauche" alt="..." src="..." />
</div>
```

Code CSS :

```
.gauche {
float: left;
}
```

A retenir

- La propriété position définit la façon dont un élément est positionné dans un document. Les propriétés t***op, right, bottom*** et ***left*** déterminent l'emplacement final de l'élément positionné.

- Un élément positionné est un élément dont la propriété de position calculée est *relative, absolute ou fixed*.

- Un élément positionné de façon relative est un élément dont la propriété de position calculée est relative. Dans ce cas, les propriétés top ou bottom indiquent le décalage vertical à appliquer et left ou right indiquent le décalage horizontal.

- Un élément positionné de façon absolue est un élément dont la propriété de position calculée est absolute ou fixed. Dans ce cas, les propriétés top, bottom, right et left indiquent les distances entre les bords de l'élément et les bords du bloc englobant (c'est-à-dire l'ancêtre par rapport auquel l'élément est positionné). Si l'élément possède des marges, elles sont ajoutées aux décalages.