

Exercice 8 (bonus) – Structure statique site web mobile

Dans cet exercice vous devez, en suivant les principes acquis dans l'exercice précédent, reproduire la mise en page d'un site web mobile.

Une aide est disponible à la fin des consignes lorsque ce sigle apparaît « @ ».

Une liste de quelques propriétés CSS pour les blocks est disponible à cette adresse :

http://www.ecoledunet.com/cours/css_modele_position.pdf

L'intégralité du design devra être produit avec des boîtes <div> et du CSS et éventuellement des attributs pour gérer la taille des images.

Sources : Une fois l'archive décompressée vous trouverez :

- Un aperçu du résultat à obtenir : « rendu.jpg »
- Une méthode, « methode.jpg », qui vous aidera à développer votre mise en page pas à pas.

Notions abordées : boîtes / tailles / marges intérieures et extérieures / couleur de fond / site web mobile

Mémento :

- Pensez à organiser votre code en effectuant des sauts de lignes et des tabulations.
- Pensez régulièrement à tester votre page sur les différents navigateurs. Touche F12 sous Dreamweaver ou icône en forme de planète terre.

PARTIE 1 : Pour commencer...

- ➔ Créer un nouveau dossier « exo8 » dans lequel vous stockerez votre site. Les dossiers et les noms de fichiers d'un site web ne doivent JAMAIS contenir d'espace ni de caractères accentués. Vous pouvez mettre un « _ » ou un « - » de substitution. Par convention nous écrirons toujours tout en minuscule.
- ➔ Créer un dossier « images » à l'intérieur du dossier de votre site, dans lequel vous placerez l'ensemble des images nécessaires à la réalisation.
- ➔ Créer une nouvelle page HTML avec Dreamweaver et enregistrez-la à la racine du dossier de votre site.
- ➔ Déclarer un nouveau site dans Dreamweaver.
- ➔ Définir le titre de la page « Restaurant Alsacien ».

PARTIE 2 : reproduire la maquette « methode.jpg »

- ➔ Définir une typographie pour l'ensemble du document HTML. @
- ➔ Reproduire la mise en page, conformément à la maquette graphique @.
- ➔ Organiser votre code, faites des sauts de lignes pour le rendre graphiquement lisible.
- ➔ Uploadez votre exercice en ligne, dans votre dossier FTP
- ➔ Affichez la page en ligne

AIDE

1. Définir une famille de typographie pour l'ensemble du document

Il est possible de définir une typographie générale pour l'ensemble du document HTML. Cela évite d'avoir à définir la famille de typo utilisée balise par balise...

Pour se faire nous emploierons la propriété CSS « font-family ». La valeur de cette propriété est le nom de la « font » employée (Verdana, Tahoma...). Lorsque le nom d'une typographie comporte des espaces, Comic sans MS par exemple, nous emploierons des guillemets simples pour la définir « ' ».

Si l'on prend en considération que cette propriété doit être valable pour l'ensemble du document il convient de placer cette déclaration dans la balise <body> car toutes les propriétés CSS sont héritées des balises parentes.

Attention, il ne faut pas recréer une balise <body>, mais placer la propriété CSS à l'intérieur.

Ex 1 : (sans espaces)

```
<body style="font-family:Tahoma;">
```

Ex 2 : (avec des espaces)

```
<body style="font-family:'Comic Sans MS';">
```

Il est possible de définir des typographies de secours au cas où la typographie ne serait pas installée sur la machine du consultant. Pour se faire nous emploierons une virgule suivie des noms des typographies de substitutions.

```
<body style="font-family:Tahoma, Verdana, 'Comic sans MS';">
```

Dans cet exemple la famille de typographie principale est Tahoma. Si celle-ci n'est pas installée sur l'ordinateur de l'internaute, les polices de substitutions viendront en remplacement. Dans l'ordre il s'agira de Verdana puis Comic sans MS.

2. Gestion du design pour mobile

Lorsque l'on développe un site web pour mobile il faut être le plus simple et le plus cohérent possible. La grande problématique du développement mobile vient d'une contradiction technologique évidente : la majeure partie des navigateurs internet mobiles sont développés pour interpréter des sites pour pc fixes, qui n'ont pas de versions mobiles. Et Dans le même temps les développeurs adaptent des sites dans une version mobile, pour des navigateurs mobiles qui n'y sont pas si bien préparés que ça...

Les idées maîtresses du développement de sites web mobiles s'axent autour de :

- La représentation des typographies, plus grosse et donc plus lisible. On compte 18px pour un corps de texte standard et 32px pour un niveau de texte 1 (<h1>).

- La superposition de blocs adaptatifs. On utilise beaucoup les pourcentages qui permettent d'adapter la représentation du site en fonction du format de consultation (landscape/portrait). Notez que par défaut un bloc mesure déjà 100% de la taille écran.
- On essaye de développer simple, sans conteneur, associant les propriétés CSS directement à la balise concernée et non à la balise parent.
- Il faut tester ses productions dans le navigateur du mobile. Il est également possible d'avoir un aperçu en redimensionnant sa fenêtre de son navigateur fixe à la taille d'un Smartphone.

3. Définir une couleur de fond

Pour définir une couleur de fond nous utilisons la même propriété CSS que celle qui permet de définir une image de fond.

Ex : `background :#ffcc00 ;`

4. Centrer une image en CSS

La balise `` est une balise de type « inline ». Pour centrer une image en CSS, il convient d'associer 2 propriétés CSS : « `display` », qui permet de redéfinir le type d'élément en « `block` », et « `margin` » qui permet de centrer.

Ex : centrage d'une image

```

```

NB : Il est également possible d'utiliser d'autres moyens détournés, pour centrer une image, mais ils sont généralement moins pratiques puisqu'ils font appelle à l'utilisation d'autres blocks. Dans cette optique on pourrait envisager de placer la balise image dans un conteneur ayant une propriété de centrage « `text-align:center ;` » par exemple.

5. Proportions des boîtes

Il existe principalement 2 unités de mesures qui permettent de déterminer la taille d'un élément :

Le pixel : L'unité de mesure de l'écran. **Ex :** 500px

Le pourcentage : La taille de l'élément est définie en fonction de la taille écran de l'utilisateur. **Ex :** 50%

Les 2 propriétés CSS qui permettent de définir une boîte s'intitulent « **width** » et « **height** ». Il est important de prendre en considération que la hauteur, « **height** » est également défini par la hauteur du contenu placé à l'intérieur de la boîte. Ainsi, la plupart des boîtes qui incluent le corps de texte n'ont pas de hauteur définie, puisque c'est le contenu qui déterminera la hauteur.

Ex :

```
<div style="width:250px;height:300px;">
  Contenu à encadrer
</div>
```

NB : Par défaut la largeur d'une boîte est toujours 100%.

NB : Il n'est pas possible de définir une hauteur en pourcentage. Excepté si le type de positionnement de l'élément est absolu « absolute »._Cf. Exercice suivants.

6. Centrer un élément et gérer les marges extérieures

En CSS, la propriété qui permet de centrer un élément et de gérer les marges extérieures est la même.

Il est possible de définir indépendamment une marge extérieure pour les 4 cotés d'un élément de la manière suivante :

```
<div style="margin-top:5px;"> </div> <!-- Marge haute -->
<div style="margin-right:10px;"> </div> <!-- Marge droite -->
<div style="margin-bottom:15px;"> </div> <!-- Marge bas -->
<div style="margin-left:20px;"> </div> <!-- Marge gauche -->
```

Bien entendu il est possible de coupler ces propriétés, « margin-bottom :5px ;margin-left :10px ».

```
<div style="margin-top:5px; margin-right:10px;"> </div>
<!-- Marge haute de 5px et marge droite de 10px -->
```

Lorsque l'on souhaite combiner plusieurs de ces propriétés de marges extérieures, on utilise la propriété « margin » qui est, souvenez-vous, également utilisée pour centrer un élément. Comment faire ?

La propriété margin permet de définir successivement la valeur des 4 marges que l'on va avoir en suivant le sens des aiguilles d'une montre.

```
<div style="margin: 5px 10px 15px 20px;"> </div>
```

Dans cet exemple, la boîte a comme propriété :

- une marge extérieure haute de 5px
- une marge extérieure à droite de 10px
- une marge extérieure en bas de 15px
- une marge extérieure à gauche de 20px

Souvenez vous également que pour centrer un élément il suffit d'attribuer la valeur « auto » à la marge gauche et à la marge droite.

```
<div style="margin: 5px auto 15px auto;"> </div>
```

Ainsi, la boîte dispose d'une marge extérieure haute de 5px, d'une marge extérieure basse de 15px et sera centrée dans la mise en page.

7. Marges intérieures (padding)

Il s'agit de la marge présente entre la zone de contenu et la bordure de la boîte. Elle est généralement utilisée pour décoller le texte du contour de la boîte.

La propriété CSS qui permet de gérer cette marge interne s'intitule « padding », elle obéit aux mêmes règles de compositions que « margin ».

```
<div style="padding-top:5px;"> </div>
<!-- Marge int. haute -->
```

```
<div style="padding:5px 10px 15px 20px;"> </div>
<!-- Marge int. haute droite bas gauche -->

<div style="padding:5px;"></div>
<!-- Marge int. de 5px de tous les cotés -->
```

NB : Il n'est pas possible d'utiliser un « padding » négatifs.

8. Supprimer les marges extérieures

Certains éléments comme les paragraphes, les niveaux de titres (<h1>), ont par défaut des marges qui sont graphiquement représentées dans le navigateur. Il peut s'avérer utile de supprimer ces marges. Pour se faire, nous utiliserons la propriété CSS « margin »

Ex : pour supprimer la marge haute

```
<p style="margin-top:0px;">
```

Ex : pour supprimer toutes les marges...

```
<p style="margin:0px;">
```