

HTML 5 et CSS 3 (partie 2)

Objectifs

- Connaître quelques notions avancées de CSS 3,
- Appréhender l’affichage,
- Introduction au Responsive Web Design.

1. La notion de cascade

On constate que l’on peut avoir plusieurs définitions de style soit en ligne, interne(s) ou externe(s). En cas de concurrence entre plusieurs éléments de style, intervient alors la notion de "**cascade**" (le Cascading de Cascading Style Sheets) ou d’ordre de priorité.

Le navigateur prend d’abord en considération les spécifications des feuilles de style externes (avec l’extension css), ensuite celles de style internes (soit celles à l’intérieur des balises <head>) et ensuite des feuilles de style en ligne (celles liées à un élément HTML 5).

Ainsi, en cas de conflit entre une spécification de style définie à la fois dans une feuille de style externe et dans une feuille de style interne, c’est la spécification de la feuille de style interne qui sera retenue par le navigateur. De même, en cas de conflit entre une feuille de style interne et en ligne, c’est cette dernière qui l’emportera.

L’ordre croissant de priorité (de la plus basse à la plus haute) est :

1. Propriétés du navigateur.
2. Les feuilles de style externes.
3. Les feuilles de style internes.
4. Les feuilles de style en ligne.

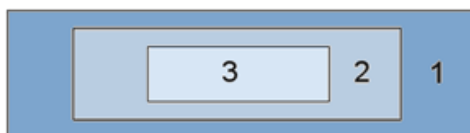
Les styles déclarés dans la feuille de style en ligne ont donc bien la plus haute priorité. La règle de priorité, pour l’affichage du document par le navigateur, sera d’utiliser la feuille de style la plus proche de l’élément.

2. La notion d’héritage

Le principe de l’héritage peut être expliqué par un rappel du HTML et de ses balises imbriquées. Soit le bout de code :

Le langage HTML <i>Version 5,<code>initiation</code></i>

« Version 5, initiation » sera non seulement sous une police à pas fixe (les balises <code> ... </code>) mais aussi en italique (compris dans des balises <i> ... </i>) et également en gras (balises ...). On pourrait dire que « Version 5, initiation » a hérité du formatage italique de la balise <i> et du formatage gras de la balise . Il en est de même pour les styles.



Le style 2 se caractérise par son propre style et celui du style 1 dont il a hérité. Le style 3 se caractérise par son propre style et celui du style 1 et 2 dont il a hérité. Dans la même logique, on utilisera le terme de « parent » et « enfant ».

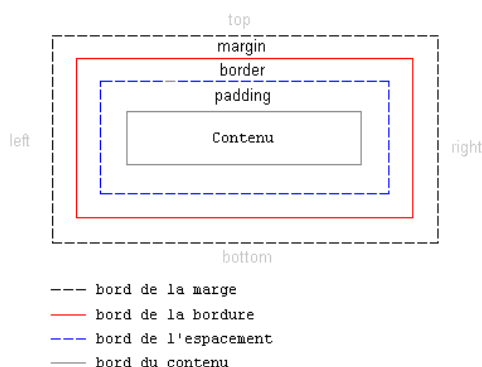
Le style 1 est bien entendu parent du style 2. De même, le style 2 et le style 1 sont parents du style 3. Ou encore, le style 2 est enfant du style 1 et le style 3 est enfant des styles 2 et 1.

3. Concept du modèle de boîte

L'élément boîte est une notion importante en CSS. Cette notion, un peu abstraite, doit être correctement intégrée pour la suite de notre étude et tout particulièrement en vue du positionnement (chapitre Les propriétés d'affichage - Positionnement).

Cet élément boîte est aussi appelé un conteneur ou un bloc (block). Le W3C définit un élément boîte comme une zone rectangulaire constituée :

- d'un contenu,
- d'une marge intérieure (padding),
- d'une bordure (border),
- et d'une marge extérieure (margin).



A savoir que :

- Le contenu peut être, par exemple, le texte d'un paragraphe.
- Celui-ci est entouré d'un espace allant du texte aux bordures du cadre, que l'on appellera la **marge intérieure** ou interne.
- Les bordures du cadre (facultatives).
- Un espace entourant le tout, appelé **marge extérieure** ou externe.

Les feuilles de style vont reprendre et étendre cette notion de boîte pour donner au concepteur le contrôle total de la marge intérieure et extérieure et de la bordure.

Avec les feuilles de style CSS, on pourra, par exemple, régler distinctement :

- Les quatre marges extérieures et ce dans toutes les directions,
- Les quatre bordures (dimension, style, couleur),
- Les quatre marges intérieures et ce dans toutes les directions,
- Les dimensions (largeur et hauteur) du contenu.

La source principale de confusion est que, après avoir précisé la largeur et/ou la hauteur du contenu, on n'a pas explicitement défini la dimension totale de l'élément boîte, qui comprend la dimension du contenu, la marge intérieure, la dimension de la bordure et la marge extérieure.

Ainsi, la propriété **width** appliquée au contenu (le petit rectangle central) n'est qu'un élément de la dimension de l'élément boîte, il faut y ajouter la marge intérieure, la bordure et la marge extérieure pour obtenir l'espace total réel de l'élément boîte. C'est bien entendu la même chose pour la hauteur déterminée par la propriété **height**.

La dimension totale de l'élément boîte se calcule donc ainsi : contenu + marges intérieures + bordures + marges extérieures.

Exemple

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 <style type="text/css">
7   p { width: 200px;
8       border: 10px solid rgb(125,165,205);
9       padding: 20px;
10      margin: 20px;
11      background-color: rgb(215,230,245);}
12 </style>
13 </head>
14 <body>
15 <p>Notion de boîte en CSS</p>
16 </body>
17 </html>
```

4. Élément bloc ou en ligne

La propriété de style display permet de redéfinir un élément en ligne comme étant un élément bloc et inversement.

display:	inline; (en ligne)
	block; (de type bloc)

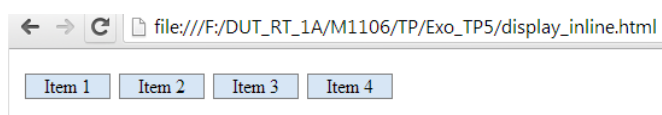
Cette propriété est très riche en possibilités de présentation et est souvent utilisée pour des feuilles de style CSS complexes.

Les nouvelles balises sémantiques sont des éléments en ligne. Pour leur attribuer une fonction de présentation, il est impératif de les définir comme des éléments de type bloc. Application à la balise <nav>.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 <style type="text/css">
7   nav { display: block;
8         width: 350px;
9         border: 1px solid gray;
10        background-color : rgb(215,230,245);
11        padding-left: 4px;}
12   a { text-decoration: none;}
13 </style>
14 </head>
15 <body>
16 <nav>
17 <a>Home</a> |
18 <a>Produits</a> |
19 <a>Solutions</a> |
20 <a>Points de vente</a> |
21 <a>Contact</a>
22 </nav>
23 </body>
24 </html>
```

Les items de listes sont des éléments de type bloc. Ils apparaissent d'ailleurs les uns en dessous des autres. Pourtant avec la propriété **display: inline**, il est possible de les afficher sur une ligne.

Réaliser le menu en ligne suivant (balises ul et li)



5. Les propriétés d’affichage

Un élément peut se positionner de quatre façons différentes.

- Positionnement statique

C’est le positionnement normal de l’élément selon la façon habituelle de procéder du navigateur.

position: static;	
-------------------	--

- Positionnement relatif

C’est le positionnement d’un élément par rapport à sa position normale ou statique. Cet élément reste dans le flux des données mais est en quelque sorte excentré par rapport à sa position normale.

position: relative;	left: valeur ou % top: valeur ou %; right: valeur ou %; bottom: valeur ou %;
---------------------	---

- Positionnement absolu

Le positionnement absolu crée un élément indépendant du reste du document. Les éléments définis en position absolue sont retirés du flux normal et se positionnent à l’emplacement exact défini par le concepteur. La position est définie par les coordonnées (x,y).

position: absolute;	left: valeur ou % top: valeur ou %; right: valeur ou %; bottom: valeur ou %;
---------------------	---

- Positionnement fixe

Le positionnement fixe crée aussi un élément indépendant dont on peut définir la position. Ici l’élément reste fixe même lorsque l’on fait défiler le document.

position: fixed;	left: valeur ou % top: valeur ou %; right: valeur ou %; bottom: valeur ou %;
------------------	---

6. Flottement

La propriété **float** retire un élément boîte du flux normal pour la placer le plus à droite ou le plus gauche possible dans son élément parent, soit son conteneur.

float:	right; ou left; ou none;
--------	--------------------------------

- La valeur **right** aligne l’élément concerné à droite, poussant les autres éléments à s’aligner à gauche.
- La valeur **left** aligne l’élément concerné à gauche, poussant les autres éléments à s’aligner à droite.
- La valeur **none** ne spécifie rien et rend la main au navigateur.
- La position **float** ne peut s’appliquer en cas de positionnement absolu.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 </head>
7 <body>
8 <div style="float: right;">
9 
10 </div>
11 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
12 non risus. Suspendisse lectus tortor, dignissim sit amet,
13 adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
14 diam. Maecenas ligula massa, varius a, semper congue, euismod non,
15 mi.</p>
16 </div>
17 </body>
18 </html>
```

7. Dégagement

La propriété `clear` permet d'annuler le flottement introduit par la propriété `float`.

<code>clear:</code>	<code>right;</code> <code>ou left;</code> <code>ou both;</code> <code>ou none;</code>
---------------------	--

- La valeur **right** annule le flottement à droite.
- La valeur **left** annule le flottement à gauche.
- La valeur **both** annule le flottement des deux côtés.
- La valeur **none** annule toute propriété de flottement.

Reprendre l'exemple précédent et annuler le flottement.

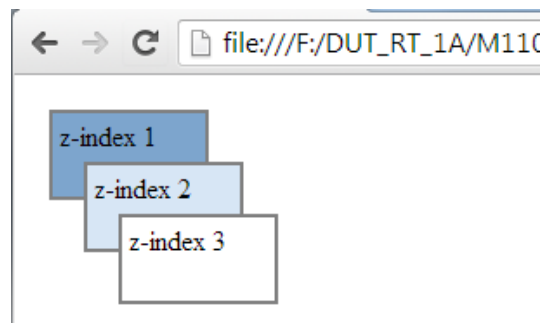
8. Superposition

La propriété `z-index` ajoute un axe en profondeur permettant de positionner des éléments au-dessus ou en dessous d'un autre élément. Ainsi, l'élément avec une propriété `z-index: 2` apparaîtra avant ou au-dessus de l'élément avec une propriété `z-index: 0`.

<code>z-index:</code>	un nombre entier (positif);
-----------------------	-----------------------------

Cette propriété ne fonctionne qu'avec un positionnement absolu des éléments.

Réaliser la superposition de 3 boîtes.



9. Découpage

Cette propriété `clip` détermine la partie visible de l'élément, généralement une image. Ainsi, l'image entière sera bien présente dans le document mais seulement une partie (rectangulaire) de celle-ci sera visible.

<code>clip:</code>	<code>rect(sup_gauche sup_droit inf_droit inf_gauche);</code> <code>auto;</code>
--------------------	---

Soit par exemple : `clip: rect(10px 270px 190px 50px);`

10. Les feuilles de style CSS 3

Les bords arrondis

Les bords arrondis des quatre extrémités sont introduits par la propriété :

border-radius:	x y; où x et y sont une valeur ou un pourcentage
----------------	---

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  <style type="text/css">
7  #round { width: 200px; height: 80px;
8           background-color: rgb(185,205,225);
9           border: 1px solid gray;
10          border-radius: 1em;}
11 </style>
12 </head>
13 <body>
14 <div id="round"></div>
15 </body>
16 </html>|

```

Particularité pour les feuilles de style CSS3. Les navigateurs n'ont souvent pas attendu la recommandation officielle du W3C pour implémenter les propriétés CSS3. Ainsi pendant tout un temps, on retrouve ces propriétés précédées d'un préfixe ; **-moz-** pour Firefox, **-ms-** pour Internet Explorer, **-webkit-** pour Safari et Google Chrome, **-o-** pour Opera. Au fil du temps et des versions, les propriétés CSS3 sont reconnues sans préfixe.

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  <style type="text/css">
7  #round { width: 200px; height: 80px;
8           background-color: rgb(185,205,225);
9           border: 1px solid gray;
10          border-radius: 1em;
11          -moz-border-radius: 1em;
12          -webkit-border-radius: 1em;}
13 </style>
14 </head>
15 <body>
16 <div id="round"></div>
17 </body>
18 </html>|

```

Il est également possible de définir l'arrondi de chacun des angles, à l'aide d'une écriture raccourcie qui se lit comme à l'accoutumée dans le sens des aiguilles d'une montre en débutant par le haut (top, right, bottom, left).

border-radius: liste valeurs / liste valeurs

Les bords arrondis ne s'appliquent pas qu'aux divisions <div> mais aussi dans un tableau et aux images.

Les ombres

Avec les propriétés CSS3, il est à présent possible d'ajouter un effet d'ombre sur le texte ou les éléments boîte.

Les ombres sur le texte

text-shadow:	x y z couleur où - x est le déport de l'ombre vers la droite. - y est le déport de l'ombre vers le bas. - z est l'intensité du dégradé ou du flou (facultatif, par défaut 0). - couleur est la couleur de l'ombre.
--------------	--

Les paramètres x et y admettent des valeurs négatives. Le déport de l'ombre est alors respectivement vers la gauche et vers le haut.

Les ombres sur les éléments boîte

box-shadow:	x y z couleur où - x est le déport de l'ombre vers la droite. - y est le déport de l'ombre vers le bas. - z est l'intensité du dégradé ou du flou (facultatif, par défaut 0). - couleur est la couleur de l'ombre.
-------------	--

Les paramètres x et y admettent des valeurs négatives. Le déport de l'ombre est alors respectivement vers la gauche et vers le haut. La propriété s'applique sur la boîte de l'élément et non sur sa bordure. L'ombrage n'affecte donc pas la taille de la boîte de l'élément.

Redimensionner un élément

La propriété CSS3 **resize** permet de redimensionner un élément.

resize:	none; both; horizontal; vertical;
---------	--

- none : l'utilisateur ne peut redimensionner l'élément.
- both : l'utilisateur peut modifier l'élément en hauteur et en largeur.
- horizontal : l'utilisateur peut modifier la largeur de l'élément.
- vertical : l'utilisateur peut modifier la hauteur de l'élément.

```
1 <!doctype html>
2 <html>
3 <head>
4 <title>Html5</title>
5 <meta charset="utf-8">
6 <style type="text/css">
7 .bloc { resize:vertical;
8         overflow:hidden;
9         width:270px; height:60px;
10        border:1px solid silver;}
11 </style>
12 </head>
13 <body>
14 <br>
15 <div class="bloc">
16 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
17 risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
18 nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas
19 ligula massa, varius a, semper congue, euismod non, mi. Proin
20 porttitor, orci nec nonummy molestie, enim est eleifend mi, non
21 fermentum diam nisl sit amet erat.
22 </div>
23 <p><b>Agrandir la division pour lire la suite.<b></P>
24 </body>
25 </html>
```

L'opacité ou la transparence

Avec les CSS3, il est possible de faire varier l'opacité ou la transparence d'un élément.

opacity:	Valeur comprise entre 1 et 0. Avec la valeur 1, l'opacité est complète et la transparence nulle. Avec 0, l'opacité est nulle et l'élément complètement transparent.
----------	---

- Le résultat de la propriété opacity est assez proche de celui de la notation de couleur RGBA avec transparence. Il faut cependant noter que la propriété opacity s'applique à l'élément dans son intégralité, soit à tous ses descendants. RGBA ne s'applique uniquement qu'à l'élément sélectionné.
- L'élément doté de la propriété opacity est translucide. Il laisse apparaître l'élément qui se trouve éventuellement situé en dessous de lui (voir exemple ci-après).

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 <style type="text/css">
7   div.background { width: 250px; height: 100px;
8                     background: url(cup.jpg);
9                     border: 2px solid black;}
10  div.box { width: 200px; height: 60px;
11            margin: 24px 25px;
12            border: 1px solid black;
13            background-color: #ffffff;
14            opacity: 0.5;
15            text-align: center;}
16 </style>
17 </head>
18 <body>
19 <p></p>
20 <div class="background">
21 <div class="box">
22 <h2>Lorem ipsum dolor</h2>
23 </div>
24 </div>
25 </body>
26 </html>

```

Modifier l'opacité de l'image au survol du curseur de la souris.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 <body>
7 
10 </body>
11 </html>

```

Les colonnes multiples

Présenter un contenu en plusieurs colonnes comme avec Adobe InDesign est une autre nouveauté des CSS3.

column-count:	Entier qui détermine le nombre de colonnes dans lesquelles sera affiché le contenu de l'élément.
column-width:	Valeur qui décrit la largeur optimale de chaque colonne (facultatif).
column-gap:	Valeur qui détermine l'espace ou le padding entre les colonnes.
column-rule:	Définit une bordure entre les colonnes.

-moz-column-count:	-webkit-column-count:
-moz-column-width:	-webkit-column-width:
-moz-column-gap:	-webkit-column-gap:
-moz-column-rule:	-webkit-column-rule:


```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  <style type="text/css">
7  #box { width:420px; height: 240px;
8         text-align: justify;
9         column-count: 2;
10        column-gap: 2em;
11        column-rule: 1px solid silver;
12        -moz-column-count: 2;
13        -moz-column-gap: 2em;
14        -moz-column-rule: 1px solid silver;
15        -webkit-column-count: 2;
16        -webkit-column-gap: 2em;
17        -webkit-column-rule: 1px solid silver;}
18 </style>
19 </head>
20 <body>
21 <div id="box">
22 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
23 risus. Suspendisse lectus tortor, dignissim sit in amet,
24 adipiscing nec, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
25 risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
26 nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas
27 ligula massa, varius a, semper congue, euismod non, mi. Proin
28 porttitor, orci nec nonummy molestie, enim est eleifend mi, non
29 fermentum diam nisl sit amet erat.|
30 </div>
31 </body>
32 </html>

```

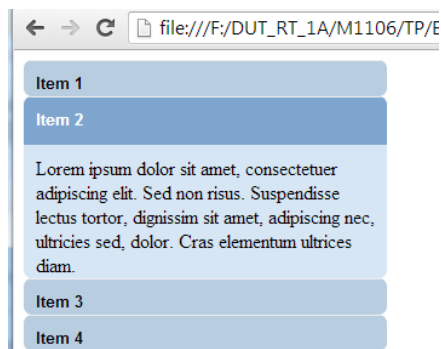
Les transitions

Les propriétés CSS3 transition apportent du mouvement dans le design de la page, à l'instar de ce qui se réalise avec du JavaScript. Les animations se réalisent principalement à partir de quatre propriétés :

transition-property	Précise les propriétés CSS à animer, par exemple, la couleur (color) et la largeur (width). Ces propriétés sont précisées en les listant, séparées par des virgules. Le mot-clé all (défaut) désigne toutes les propriétés animables de l'élément.
transition-duration	Précise la durée de la transition. Celle-ci est exprimée en s (secondes) ou ms (millisecondes).
transition-timing-function	Précise la fonction de transition à utiliser : ease : rapide sur le début et ralenti sur la fin. linear : vitesse constante sur toute la durée de l'animation. ease-in : lent au début et accélère de plus en plus vers la fin. ease-out : rapide sur le début et décélère sur la fin. ease-in-out : le départ et la fin sont lents.
transition-delay	Précise le retard (ou l'avance) du départ de la transition. Celui-ci est exprimé en s (secondes) ou ms (millisecondes).

La présentation de contenu sous forme d'accordéon connaît un franc succès sur la toile. La propriété CSS3 **transform** permet de réaliser cet effet uniquement par des feuilles de style, sans faire appel à du code JavaScript.

Réaliser cette présentation accordéon.



Les filtres graphiques

Avec la propriété **filter**, il est possible d'appliquer sur des images ou sur tout autre élément Html des effets de filtres jusque-là réservés aux logiciels graphiques comme Photoshop, Corel Paint Shop Pro ou Gimp pour ne citer que ceux-là.

Ne vous réjouissez pas trop vite car cette spécification du W3C est très récente (Editor's Draft du 16 novembre 2012) et n'est prise en charge, pour le moment, que par Google Chrome 18+ et Safari 6+ avec le préfixe -webkit-.

Sa syntaxe est : -webkit-filter: filtre(valeur);

Les filtres disponibles sont :

- grayscale(valeur) Niveau de gris où valeur est un nombre compris entre 0 et 1 ou un pourcentage.
- sepia(valeur) Effet sépia où valeur est un nombre compris entre 0 et 1 ou un pourcentage.
- saturate(valeur) Saturation où valeur est exprimée en un nombre compris entre 0 et 10 ou un pourcentage.
- hue-rotate(angle) Teinte où angle est exprimée en degrés.
- invert(valeur) Effet négatif où valeur est un nombre compris entre 0 et 1 ou un pourcentage.
- opacity(valeur) Opacité où valeur est un nombre compris entre 0 et 1 ou un pourcentage.
- brightness(valeur) Luminosité où valeur est un nombre compris entre 0 et 1 ou un pourcentage.
- contrast(valeur) Contraste où valeur est un nombre compris entre 0 et 10 ou un pourcentage.
- blur(valeur) Flou où valeur est exprimée en pixels.
- drop-shadow(ombre) Ombre portée par exemple drop-shadow(6px 6px 20px black).

Pour visualiser ces différents effets, reportez-vous aux sites <http://jsbin.com/uziyih> ou <http://html5-demos.appspot.com/static/css/filters/index.html>

Une multitude d'autres filtres sont annoncés comme halftone, motion-blur, posterize, bump, affine-transform, bloom, gloom, mosaic, displace, edge-detect, pinch, twirl.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 </head>
7 <body>
8 
9 
10 
11 
12 </body>
13 </html>
```

Il existe de nombreux autres propriétés du CSS 3 telles que :

Les polices personnalisées, les lettres creuses, la césure automatique, les arrières plans multiples, les dégradés de couleur, les transitions, les masques, les effets de reflet, les fondus d'images...

La balise CANVAS

L'élément <canvas> est basé sur la spécification canvas du WHATWG, elle-même basée sur la balise <canvas> d'Apple implémentée dans Safari. Celui-ci peut être utilisé pour afficher des graphes, des éléments d'interface, et d'autres éléments graphiques personnalisés sur le client.

<canvas> crée une surface de dessin de taille fixe, ou canevas, exposant un ou plusieurs contextes de rendu. Nous nous concentrerons sur le contexte de rendu 2D (c'est d'ailleurs le seul contexte de rendu actuellement défini). Dans le futur, d'autres contextes peuvent fournir différents types de rendu. Par exemple, il est probable qu'un contexte 3D basé sur OpenGL ES sera ajouté à la spécification <canvas>.

La balise <canvas> affiche dynamiquement une image mise à jour à l'aide d'un code javascript.

11. Site Web adaptatif : Responsive Web Design

On appelle Responsive Design ou Responsive Web Design (littéralement conception web adaptive) la faculté d'un site web à s'adapter au terminal de lecture.

Comprendre les media queries

Cette pratique (RWD) consiste à concevoir un design graphique adaptatif pour les périphériques de consultation des pages HTML, mobiles ou non, et ce en appliquant des feuilles de styles séparées et adaptées en fonction des résolutions visées.

Déclarer vos feuilles de styles

Depuis CSS2, il est possible de spécifier le type de média auquel appliquer une feuille de styles spécifique. Ainsi on peut :

- Déclarer les feuilles de styles au sein de votre page HTML,
- Déclarer les feuilles de styles au sein d'une autre CSS,
- Importer les feuilles de styles au sein d'une autre CSS.

Toutes ces règles peuvent être combinées. Elles agiront en cascade (sur le schéma parent-enfant).

Identifier les résolutions courantes

Il y a 4 grandes familles de résolutions standards :

- Les ordinateurs de bureau (1024 pixels et plus),
- Les ordinateurs portables et tablettes (1024 à 768 pixels),
- Les tablettes et smartphones (768 à 480 pixels),
- Les téléphones mobiles de petite taille (480 pixels).

Déclarer la balise viewport

Dans la partie <head></head>, on déclare la balise méta **viewport**. Cette balise permettra de définir les dimensions de la page HTML pour les périphériques nomades de faible résolution. Les valeurs de l'attribut CONTENT vont initialiser les dimensions de la page HTML.

- Width=device-width redimensionne le site sur toute la largeur de l'écran du mobile,
- Initial-scale=1.0 profondeur initiale du zoom,
- Maximum-scale=5.0 le taux de zoom est égal à 5 fois la valeur initiale,
- User-scalable=yes on peut zoomer et dézoomer sur la page.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width,initial-scale=1.0, maximum-scale=5.0, user-scalable=yes"/>
7 |
8 </head>
```

Déclarer la feuille de style standard

L'attribut media="screen" signifie que les styles employés ne s'appliqueront qu'aux terminaux de type écran. Mais il existe de nombreux autres attributs déterminant le type d'environnement dans lequel certains styles doivent s'appliquer.

Déclarer la feuille de styles dédiée tablette 7 pouces

Déclaration d'une seconde feuille de styles avec l'ajout de l'attribut média "screen and (max-width: 1024px)". Cela signifie que cette feuille de styles ne s'appliquera que dans un contexte où la résolution n'excèdera pas 1024 pixels de large.

Structurer la page HTML

Cinq éléments structurels principaux seront déterminants :

- Le conteneur principal <container></container>
- L'en-tête <header></header>
- Le menu principal <nav></nav>
- La partie de gauche <content_01></content_01>
- La partie de droite <content_02></content_02>

Chacune de ces parties inclura des éléments enfants, tels que textes, images...

Identifier les principaux éléments structurels

Dans la feuille de style principal, on trouve :

- #container {width: 940px;}
- #content_01 {width: 600px; padding-left: 20px;}
- #content_02 {width: 300px; padding-left: 20px;}
- content_01_block {width 280px; margin-left:20 px;}

Le conteneur principal fait 940 pixels de large. Dans ce dernier, 2 parties : une à gauche (#content_01) qui fait 600 pixels de large, largeur à laquelle viennent se rajouter 20 pixels de marge intérieure à gauche et une partie à droite (#content_02) qui fait 300 pixels de large (+20 pixels de marge intérieure à gauche). En additionnant les 2 parties (620 + 320), on arrive bien à 940 pixels.

Dans la seconde feuille de styles, on retrouve les éléments présents dans la feuille de styles principale mais avec les valeurs suivantes :

- #container {width: 640px;}
- #content_01 {width: 280px; padding: 40px;}
- #content_02 {width: 300px; padding-left: 20px;}
- content_01_block {width 280px; margin-left:0 px;}

Il n'y a plus de marge à droite pour les éléments de la classe **.content_01_block**. Il y a une marge intérieurs supplémentaire à l'élément #content_01. Tous les blocs de la classe **.content_01_block** viendront s'empiler les uns en dessous des autres.

Changer le texte présent dans le header

Pour chaque feuille de styles, on déclare que la balise est visible ou non selon son identifiant. Pour la feuille de styles principale :

- #contexte_01 {display :block ;}
- #contexte_02, #contexte_03 {display :none ;}

On affiche le contexte_01 mais pas les autres.

Quelques sites

<http://brolik.com/blog/responsive-web-design-examples-with-css-tips-and-tricks/>

<http://blog.soat.fr/2013/04/introduction-au-responsive-design-comment-rendre-son-site-responsive-sans-trop-defforts/>

<http://lehollandaisvolant.net/?d=2012/11/24/20/37/08-css-faire-un-theme-mobile-avec-html5-et-responsive-design>