TD#2 – CSS : à chacun son style

Toutes les questions du TD sont à répondre dans un document .html validé qui contiendra vos nom et prénom.

Même si nous adressons ici des problématiques d'affichage et de style, ce module n'a pas la prétention de faire de vous des designers. Veillez à vous concentrer sur la technique et non pas sur la beauté relative de vos œuvres.

CSS : Introduction

Les feuilles de style en cascade nommées CSS (Cascading Style Sheets) servent à décrire la présentation, ou style, des documents (x)HTML et XML.

Les standards définissant CSS sont publiés par le W3C. Introduit au fin 1996, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

L'un des objectifs majeurs de CSS est de séparer le contenu de la forme dans un document (x)HTML. Il est donc possible de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée.

Les styles sont appliqués au dernier moment, dans le navigateur web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de structure et de présentation, et de réduire la complexité de l'architecture d'un document.

Ainsi, les avantages des feuilles de style sont multiples :

- La structure et la présentation du document sont gérées dans des fichiers séparés.
- La conception d'un document se fait sans se soucier de la présentation.
- La présentation est uniformisée, *i.*e. plusieurs pages (x)HTML peuvent faire référence aux mêmes feuilles de styles.
- Un même document peut donner le choix entre plusieurs feuilles de style, par exemple une pour l'impression et une pour la lecture à l'écran.
- La lisibilité des pages (x)HTML est améliorée puisqu'elles ne contiennent plus de balises ni d'attributs de présentation.

Inclusion de CSS

Toutes les instructions de mises en forme seront contenues dans un fichier CSS identifiable par son extension `.css`. Le fichier (X)HTML contenant la structure du document doit alors contenir une référence vers le fichier CSS.

Par exemple, si vous possédez un fichier `style.css`, vous devrez ajouter la ligne suivante dans le **bloc header** du fichier HTML :

```
<head>
   <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

Ici, un chemin relatif avec href. Donc ma feuille de style n'est pas obligée d'être à côté de mon fichier xHTML ?

Rappel de l'épisode précédent : Si vous ne liez pas de feuille de style à un document xHTML et que vous l'ouvrez dans un navigateur, est-il affiché avec un style ?

La syntaxe de CSS

De manière générale, la syntaxe du CSS est très simple. Elle correspond au schéma suivant :

```
selecteur { propriété:valeur }
```

Chaque sélecteur (ci-dessous `body`) peut avoir plusieurs propriétés avec des valeurs indépendantes, il faut pour cela séparer les propriétés par un « ; ».

```
body {
    background: #eeeeee;
    font-family: "Trebuchet MS", Verdana, Arial, serif;
    }
```

Vous ~~pouvez~~ **DEVEZ** également insérer des commentaires dans votre code avec la syntaxe suivante :

```
 /* Commentaire ici */
```

Couleurs

Vous pouvez choisir d'utiliser :

- un nom standard de couleur (voir le site du W3C)
- de donner les couleurs sous la forme `rgb` qui offre alors un large choix de couleurs en combinant le rouge, le vert et le bleu
- de référencer une couleur par son code hexadécimal

Vous devriez jeter un oeil sur l'aide des couleurs du W3C.

A Faire :

- Combien de couleurs chacune de ces méthodes vous permettent d'atteindre ?
- Comparez en deux lignes ces méthodes.

La notion d'héritage/cascade

Si on traduit les trois mots de « Cascading Style Sheets » on obtient Feuille de style en cascade. Pourquoi « en cascade » ?

Car si vous définissez une police de type “Trebuchet MS” sur la balise <body>, l'ensemble des autres éléments du site (parce qu'inclus dans cette balise) prendra comme police `Trebuchet MS`, inutile de le redéfinir pour chaque élément.

Autre exemple, si vous définissez la balise `body` avec une couleur rouge (comme le montre la commande suivante), les autres éléments du `body` auront une couleur rouge par défaut :

```
 body { color: #FF0000; }
```

Si vous souhaitez une autre police pour les balises `<h1>` par exemple, il suffit de la définir à nouveau.

```
h1 {font-family: Georgia, sans-serif;}

p {font-family: Tahoma, serif;}
```

Les navigateurs appliquent donc à un élément la règle la plus **spécifique** qui le concerne, donc préférentiellement une règle qui cible directement cet élément, s'il n'y en a pas ce sera la règle de son élément père, puis de son grand-père, etc, jusqu'à la règle par défaut.

Un sélecteur de la forme sélecteur1 > sélecteur2, au contraire, ne désigne que le cas où sélecteur2 est directement fils de sélecteur1. Par exemple, ceci peut être utile si l'on souhaite que la règle s'applique à un `li` qui serait contenu dans une sous-liste de type `ul` seulement, mais pas `ol`.

```
ul > li { margin-left: 10em; }
```

A Faire :

- Visualiser le fichier `gardencss0.html` (fichier présent dans codes-TD-WEB.zip) avec un navigateur
- Vérifiez la validité du document. Votre objectif est de le rendre valide en séparant contenu et présentation.
- Recopier le fichier sous `gardenWithoutCss.html` puis séparer le style dans un fichier css.
- Validez votre fichier css.
- Conseil :
    1. vérifiez la validité de votre fichier css
    2. modifiez votre fichier html
    3. visualisez le html
    4. vérifiez la validité
    5. itérez
- Validez votre nouveau fichier html `gardenWithoutCss.html` résultant, il doit être conforme au fichier initial.

Quelques Tips

Type d'élément	**Eléments en css**
lien non visité	a:link {color: …}
lien visité	a:visited {color:}
lien actif (cliqué)	a:active {color:}
lien survolé	a:hover {color:}
Type de police	{font-family:“French Script MT”;
Taille de police	{font-size:xx-large; *par exemple*
Couleur du texte	color:
Texte en gras	font-weight:bold;
Texte en italique	font-style:italic;
Texte souligné	text-decoration:underline;
Couleur de fond	background-color:

Combinaison de sélecteurs

Vous pouvez aussi combiner les éléments HTML qui regroupent les mêmes caractéristiques. Voici un exemple permettant de modifier en même temps l’ensemble des balises <h*i*>

```
h1, h2, h3, h4, h5, h6 {
```

```
    color: #009900;
    font-family: Georgia, sans-serif;
}
```

En CSS, des règles de reconnaissance de motifs déterminent les règles de style qui s'appliquent aux éléments de l'arbre du document. Ces motifs, nommés sélecteurs, sont variés, allant du simple nom d'un élément jusqu'aux riches motifs contextuels. Quand toutes les conditions d'un motif sont vérifiées pour un élément donné, celui-ci est retenu par le sélecteur.

Dans un sélecteur, la sensibilité à la casse d'un nom d'élément du document dépend du langage utilisé pour le document. Par exemple, ceux-ci sont insensibles à la casse en HTML, par contre, ils le sont en XML. **Ils sont donc importants pour nous.**

- En savoir plus sur les sélecteurs
- Cascade CSS et priorité des sélecteurs … qui explique l'ordre de priorité des sélecteurs et l'utilisation de !important.

A Faire :

- Modifier la nouvelle feuille de style de l'exercice précédent pour que tous les titres s'écrivent dans la fonte : "Comic Sans MS", cursive, sans-serif
- Que se passe-t-il si vous ne modifiez pas vos définitions de styles pour h1 et h3 ?

Les polices de caractères
AVANT Le texte s'affiche avec les polices de caractères présentes sur l'ordinateur des visiteurs. Cela implique que vous ne pouvez pas utiliser n'importe quelles polices.

Pour cela vous pouvez utiliser des politiques dîtes génériques qui se trouvent sur presque tous les ordinateurs.

En savoir plus sur les fontes

Mais depuis CSS3 Vous pouvez définir des polices et les héberger sur votre propre serveur : elles seront téléchargées automatiquement au besoin.

http://www.w3schools.com/css/css3_fonts.asp

La notion de classe, de pseudo classe et d'id

Les classes

CSS ne se limite pas à la redéfinition du style des balises HTML. On peut également créer des classes, à l'aide de l'attribut `class=""`, comme le montre l'exemple ci-dessous :

```
<h1 class="nomDeLaClasse"> mon texte </h1>
```

Cela permet de regrouper plusieurs éléments, possiblement de natures différentes, en un ensemble. Par exemple, tous les éléments HTML de mon CV correspondant à de l'informatique peuvent utiliser une police particulière.

On notera que la définition d'un motif visant une classe commence par un « . ».

```
.nomDeLaClasse {
   font-size: small;
```

```
   color: #008080;
   font-weight: bold;
}
```

En résumé, une classe peut contenir zéro, un ou plusieurs éléments, à différents niveaux hiérarchiques de mon arbre xHTML.

Les pseudo-classes

HTML et CSS ont un mécanisme similaire aux classes, mais ad-hoc, pour faire le lien avec certaines fonctionnalités spécifiques, et en particulier les liens. Le HTML a deux utilisations pour l'élément <a> : faire un lien et poser une ancre. S'il est légitime de changer l'apparence d'un lien, il n'est probablement pas souhaitable que les ancres apparaissent de la même manière. Le sélecteur a:link désigne un lien par opposition à une ancre.

Vous avez sans doute remarqué que certains navigateurs affichent des couleurs différentes les liens déjà visités. CSS permet de changer ça en utilisant un sélecteur `a:visited`. Attention, `:visited` s'utilise à la place de `:link` : un lien déjà visité n'est pas reconnu par `a:link`.

Dans le même genre d'idée, il existe la pseudo-classe `:hover`, qui désigne un élément que l'utilisateur « touche », avec le pointeur de la souris par exemple. On trouve également les pseudo-classes `:focus` et `:active`, qui ont des significations proches. Ceci permet de rendre plus réactif le style de vos pages statiques en fonction de l'utilisation qui en est faite.

Le sélecteur id

Comme son nom l'indique, il s'agit d'un identifiant donné à un élément. Par définition, un identifiant est unique. Plusieurs éléments ne peuvent donc pas avoir le même identifiant. Cela permet de sélectionner très facilement un bloc xHTML spécifique de notre arbre qui n'est pas discriminé par un type de balise spécial. On trouvera par exemple dans la page html :

```
<div id="container">
   <div id="intro">
```

Et dans le fichier .css, le motif commençant par # suivant :

```
#container {
   background: url(/001/zen-bg.jpg) no-repeat top left;
   padding: 0 175px 0 110px;
   margin: 0;
   position: relative;
   }
#intro {
   min-width: 470px;
   }
```

Mise en Pratique

A Faire :

Nous voulons ajouter quelques citations sur notre site perso.
Voici quelques exemples pour les moins éclairés :

```
<h2> Citations </h2>

  <q>Programs must be written for people to read, and only incidentally for machin
es to execute.</q>

  <address>Harold Abelson</address>

  <q>Any sufficiently advanced technology is equivalent to magic.</q>

  <address>Sir Arthur C. Clarke</address>

  <q>The real danger is not that computers will begin to think like men, but that
men will begin to think like computers.</q>

  <address>Sydney J. Harris</address>

  <q>Si debugger, c’est supprimer des bugs, alors programmer ne peut être que les
ajouter.</q>

  <address>Edsger Dijkstra</address>
```

Remarquez les balises html5 utilisées; (en savoir plus).

La visualisation par défaut n'est pas satisfaisante. Modifier le fichier de style pour répondre aux critères suivants. Indice : "after".

- toutes les citations soient en **gras**, les auteurs en *italique*
- les citations en français sont en violet et suivie de *par*
- les citations en anglais sont en vert et suivie de *by*
- j'ai une préférence pour une des citations, je voudrais la faire apparaître en rouge, que proposez-vous ?

-