

TD2 — Premiers pas en CSS

1 Principes généraux

CSS est l'acronyme de *Cascading Style Sheets*, traduit par « Feuilles de style en cascade » ; c'est le langage usuel pour modifier l'apparence et la mise en forme des pages web.

Le HTML s'occupe du fond et de la sémantique des pages, CSS gère la présentation. C'est un principe fondamental en développement web : *la séparation du fond et de la forme*. On va écrire dans le fichier CSS des lignes de codes pour associer les balises HTML aux formats souhaités.

1.1 Bases de la syntaxe

La syntaxe est très simple et correspond au schéma suivant :

```
1 selecteur {  
2     propriete1: valeur1; /* Notez le point-virgule à la fin de la ligne */  
3     propriete2: valeur2a, valeur2b, valeur2c;  
4     propriete3: valeur3a, valeur3b;  
5 }
```

CSS

Par exemple, on peut changer la couleur et la taille de police dans les paragraphes en modifiant les propriétés du sélecteur **p** :

```
1 p {  
2     font-size: 12pt; /* Taille de police : 12 pt */  
3     color: grey; /* Couleur de police : gris */  
4 }
```

CSS

1.2 Styles en cascade

Les styles sont dits « en cascade », car un style appliqué sur un sélecteur seront automatiquement répercutés sur l'ensemble des éléments contenus dans les balises HTML correspondantes.

Considérons le code suivant :

```
1 body {  
2     background-color: #aaa; /* Arrière-plan gris */  
3     width: 800px; /* Largeur de 800px */  
4     font-family: Verdana, Arial, serif /* Police d'écriture */  
5 }
```

CSS

Ces instructions seront automatiquement appliquées à toutes les balises entre **<body>** et **</body>**... soit l'ensemble de la page !

1.3 Inclusion de CSS

Pour qu'un fichier CSS s'applique à vos pages HTML, il faut les relier entre eux. Supposons que vous disposiez d'un fichier `design.css`. Dans votre code HTML, on ajoute la ligne suivant *dans le bloc* `<header>...</header>` :

```
1 <header>
2     <!-- On retrouve les balises title et meta -->
3     <link rel="stylesheet" href="design.css"> <!-- Notez l'attribut href ! -->
4 </header>
```

2 Au travail !

Nous allons utiliser un fichier HTML fourni et modifier pas à pas son apparence en ajoutant un fichier CSS.

- Exercice 1** (Mise en place). 1) Placez-vous dans votre répertoire `Documents ▸ BUT_INFO ▸ R102_DIW` et créez un sous-dossier `TD2`.
- 2) À l'intérieur de ce sous-dossier, créez-en deux : `html` et `css`. Logiquement, nous placerons les fichiers HTML et CSS dans les dossiers qui leur correspondent.
- 3) Ouvrez QuickPHP et configurez-le pour qu'il cible le répertoire de votre projet, `Documents ▸ BUT_INFO ▸ R102_DIW ▸ TD2`.

Les dossiers et sous-dossiers dans un projet informatique

Dans vos projets, il sera très important de classer correctement vos divers codes sources et fichiers utiles. Un dossier pour les images, un autre pour les CSS, encore un autre pour le HTML, etc.

Pour le moment, cela semble inutile car les projets tout petits. Mais rapidement, vous aurez plusieurs dizaines de fichiers sources, de différents types, et il sera alors indispensable d'être bien organisées. Autant prendre immédiatement de bonnes habitudes.

- 4) Sur EPREL, téléchargez le fichier `index.html` du TD2 et enregistrez-le dans `html`.
- 5) Dans le répertoire `css`, créez un fichier `design.css` (vide pour le moment).
- 6) Dans l'en-tête du fichier HTML, ajouter les lignes nécessaires pour relier votre fichier CSS et votre page HTML.

À partir de maintenant, le fichier HTML ne doit plus être modifié.

- Exercice 2** (Structuration de la page). 1) Commençons par limiter la taille maximale du corps de page. Sur le sélecteur `body`, appliquez la propriété `width: 800px`. Vous devez obtenir le code suivant :

```
1 body {
2     width: 800px;
3 }
```

- 2) Pour centrer le contenu, on doit modifier les *marges* du sélecteur **body**. CSS permet de régler très précisément les marges, mais pour le moment nous allons simplement ajouter la propriété **margin: auto**; à notre sélecteur. Observez le résultat obtenu.
- 3) Enfin, nous allons modifier la couleur d'arrière-plan et choisir un gris clair, par exemple **#f7f7f7**. Le sélecteur pour modifier la couleur d'arrière-plan est **background-color: #f7f7f7**.

Les couleurs en CSS (et en informatique)

Il existe de très nombreuses manières de choisir les couleurs en CSS.

- Certaines couleurs sont prédéfinies : [en voici la liste](#).
- Pour les autres, il existe (au moins !) cinq manières différentes de choisir les couleurs, cf. [cette page](#).

Les deux plus utilisées sont RGB et HEX. En informatique, chaque couleur peut être représentée par un mélange de rouge (*red*), de vert (*green*) et de bleu (*blue*). Par exemple, **rgb(255 163 72)** est un mélange avec rouge (255), vert (163) et bleu (72). Chaque valeur est entre 0 et 255. On peut aussi utiliser la notation hexadécimale **#ffa348**.

Exercice 3 (Les paragraphes). 1) Nous allons justifier le texte de chaque paragraphe. Dans le sélecteur **p**, appliquez la propriété **text-align: justify**;

- 2) Maintenant, nous allons changer la couleur des liens (balise HTML **<a>...**) pour les mettre en **orange** (couleur CSS prédéfinie).

- a. Écrivez le code CSS nécessaire, sachant que la couleur de police correspond à la propriété CSS **color**. Votre code doit être de la forme suivante (remplacez les commentaires par les bons éléments) :

```
design.css
1  a {
2      /* Propriété pour la couleur */ : /* Valeur */;
3  }
```

- b. Quels sont les liens impactés par la modification ? Est-ce le résultat attendu ?
- c. En l'état actuel des choses, la modification est appliquée à *tous* les liens, même ceux qui ne sont pas à l'intérieur des balises **<p>...</p>** (notamment les liens en haut de la page). On souhaiterait appliquer les modifications uniquement aux liens dans des paragraphes. Pour ce faire, remplacez le sélecteur **a** par le sélecteur **p a**. Que remarquez-vous ? Expliquez.

- 3) Les liens sont soulignés par défaut. Dans la version finale de la page, le lien « *consectetur adipiscing elit* » où se trouve le pointeur de la souris, le soulignement a disparu. Pour faire cela, nous utilisons le mécanisme de *pseudo-classe*, qui permet de modifier le comportement d'un sélecteur uniquement dans certaines situations (par exemple le passage de la souris). Ajouter les lignes suivantes à votre code CSS :

```
design.css
1  a:hover { /* Observez le mot-clé hover. */
2      text-decoration: none;
3  }
```

En savoir plus sur les pseudo-classes

On peut utiliser les pseudo-classes sur toutes les balises HTML, pas uniquement pour les liens.

Par exemple, la pseudo-classes `nth:child` est très utilisée pour alterner les couleurs des lignes dans les tableaux.

Vous pouvez [consulter la documentation](#) pour faire des essais.

Exercice 4 (Les titres). Dans cet exercice, nous allons procéder à la mise en forme des titres.

1) Les titres `<h3>`.

- Écrivez le sélecteur CSS correspondant. En s'aidant des propriétés CSS utilisées dans les questions précédentes, mettez les titres en couleur `#520d35`.
- Ajoutez la ligne `margin-top: 45px;`. Qu'observez-vous ? Réglez la valeur à 15 pixels.
- Appliquez de même une marge de 5 pixels sous le titre.
- Enfin, nous allons modifier la taille de la police, avec `font-size: 1.4rem`. Faites varier la valeur `1.4rem`. Qu'observez-vous ?

Les unités de mesure en CSS

Le langage CSS propose plusieurs façons de mesurer les longueurs et distances.

Les unités absolues parmi lesquelles le centimètre (`cm`), le point (`pt`) et le pixel (`px`). Les unités absolues ne sont pas conseillées car elles ne s'adaptent pas correctement aux différentes tailles d'écrans. En règle générale, l'unité absolue à utiliser est le pixel (car c'est la plus petite longueur possible qu'un écran peut afficher), les autres sont à éviter.

Les unités relatives qui désignent un coefficient d'agrandissement (ou de réduction) par rapport à une valeur de référence. La valeur `1em` désigne simplement la taille de police de l'élément courant ; si celle-ci n'est pas définie, c'est la taille de l'élément parent. Ainsi, `1.3em` est la taille de police multipliée par 1,3. L'unité `rem` correspond à la taille de l'élément racine, elle est donc très pratique pour définir toutes les dimensions de manière relative.

2) Les titres `<h2>`.

- Mettre tous ces titres en couleur `#520d35` et en police taille `2rem`.
- En utilisant les propriétés `margin-top` et `margin-bottom`, réglez la marge haute à 40 pixels et la marge basse à 15 pixels.

3) Les titres `<h1>`. On vous fournit le code CSS suivant :

```

1 h1 {
2   background: linear-gradient(to right, #fff, #f3f3f3);
3   border-bottom: 1px solid #1088bf;
4   border-top: 1px solid #1088bf;
5   color: #520d35;
6   font-size: 2.6rem;
7   font-weight: 400;
8   line-height: 50px;
9   margin-bottom: 25px;
10  padding-bottom: 3px;
11  padding-left: 3%;
12 }

```

- a. En modifiant les valeurs des diverses propriétés, expliquer le rôle de chaque ligne.
- b. Que se passe-t-il si l'on remplace `padding-left` par `margin-left` ?

Exercice 5 (Le menu). Le code source HTML contient les balises en paires `<nav>...</nav>`. Cette paire de balises permet de délimiter les menus des sites Internet. Dans notre exemple, le menu est représenté avec une liste à puces, et les liens sont laissés vides à titre d'exemple. Cela n'empêche pas d'utiliser le sélecteur CSS `nav a` pour modifier l'apparence.

- 1) Mettre les liens du menu en couleur `#1088bf`.
- 2) Au passage de la souris, remplacer la couleur par `#520d35` et retirer le soulignement (cela se fait avec la directive `text-decoration: none`).
- 3) Enfin, on souhaite que les liens *déjà cliqués* gardent la couleur `#520d35`. Pour cela, utiliser la pseudo-classe `a:visited`.

Exercice 6 (Le pied de page). Les pieds de pages sont très utilisés sur le web : ce sont les informations présentes tout bas, et on les place entre les balises `<footer>...</footer>`. *Cet exercice est moins guidé, n'hésitez pas réutiliser ce qui a été vu dans les exercices précédents et à faire des recherches.*

- 1) Réglez la taille de police à `0.85em`.
- 2) Réglez la marge haute à `15px`.
- 3) Mettez les marges gauche et droite à la valeur `auto`. À quoi cela sert-il ?
- 4) Centrez le texte.

Exercice 7 (Validation). Comme pour le HTML, il existe un [validateur](#) CSS. Passez votre code au validateur et corrigez les éventuelles erreurs.

3 Le mot de la fin

Quelques bonnes pratiques

- 1) Les balises HTML ne gèrent que la sémantique et le code CSS ne gère que l'apparence. Les deux doivent pouvoir être modifiés de manière indépendante.
- 2) Appliquez toujours les styles en respectant la cascade : on gère les balises les plus extérieures en premiers, et on se dirige vers les balises les plus intérieures.
- 3) Au fur et à mesure, vos sites seront de plus en plus gros et auront plusieurs pages HTML. En revanche, il ne doit y avoir qu'un seul (ou éventuellement deux pour les *très* gros projets) fichiers CSS ^a. Cela permet d'harmoniser la mise en forme entre les différentes pages.
- 4) Pour mieux organiser les différents sélecteurs, vous pouvez essayer de les trier par ordre alphabétique.

^a. Par exemple, l'encyclopédie Wikipédia ne compte que deux fichiers CSS !