

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

Ce TD est une initiation à la création de feuilles de styles CSS. Vous allez d'abord découvrir la syntaxe du langage, puis voir comment mettre en forme les différentes balises HTML, changer le formatage par défaut, utiliser les différents types de sélecteurs et découvrir les propriétés les plus utiles de CSS.

### 1. Introduction

Nous avons vu au TD précédent comment créer une page HTML, structurer son contenu, ajouter des liens, des images et des tableaux. C'est bien mais cela ne suffit pas ! En effet, la page que vous avez créée n'est pas mise en forme. Pas de style de titre, pas de couleur, pas de formatage !

C'est là qu'intervient CSS ! Pour mettre en forme votre contenu, définir la taille du texte, la couleur, la position des images, des blocs de texte, l'affichage ou le masquage de zones, les arrières plans, etc.

**Souvenez vous qu'idéalement, HTML ne s'occupe jamais de la mise en forme. C'est le rôle de CSS. Et c'est son unique mission, d'ailleurs.**

En CSS, on peut quasiment tout changer de la présentation d'une page HTML. C'est très puissant, et donc assez complexe.

#### 1.1. Outils requis

Pour réaliser correctement ce TD (et les suivants), je vous recommande fortement d'utiliser la dernière version de Firefox pour tester vos réalisations, et de découvrir le raccourci clavier **Shift-F7** et son onglet « éditeur de styles CSS » qui vous permet de visualiser votre feuille de style, de la modifier et de visualiser l'impact des changements en direct !

Concernant l'éditeur de texte, pas de changement par rapport à HTML. Si vous avez un éditeur de prédilection, utilisez-le ! Sinon, je vous recommande Brackets, Gedit ou Notepad++.

#### 1.2. Références

Vous n'avancerez pas bien vite avec CSS si vous n'avez pas un minimum de documentation à disposition. Le web propose énormément de sites de qualité, donc n'hésitez pas à chercher des solutions lorsque vous êtes perdus. A toute fin utile, je vous recommande :

<http://www.w3schools.com/cssref/> et <https://developer.mozilla.org/fr/docs/CSS/Reference>.

### 2. CSS, c'est quoi ?

CSS est un acronyme pour Cascading Style Sheet. Il s'agit donc d'un langage permettant de créer des feuilles de style (en cascade... mais laissons cela de côté !).

Tout comme HTML, CSS est un langage qui se matérialise par du texte, mais ici, pas de balise mais des **sélecteurs**, qui permettent de choisir sur quelle balise HTML nous allons appliquer les **valeurs** souhaitées au **propriétés** voulues.

#### 2.1. Un petit exemple des possibilités de CSS ?

1. Faites une recherche de votre choix sur Google, avec Firefox.
2. Lorsque la liste de résultats apparaît, appuyez sur **Shift-F7**, puis allez dans l'onglet « éditeur de styles » et cliquez sur les « yeux » en bas à gauche, pour tous les désactiver (et donc désactiver toutes les feuilles CSS de cette page).
3. Regardez l'état de la page HTML...
4. Convaincus par l'utilité et la puissance et l'utilité de CSS ?

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 2.2. Simplicité, cohérence, mutualisation

Un des avantages majeurs de CSS, c'est que cela sert à définir des styles qui sont réutilisables à volonté.

Tout comme vous avez pu le voir avec les traitements de textes, cela évite par exemple, de spécifier à chaque titre sa taille, sa couleur, l'espace avant et après. Il suffit d'appliquer la feuille de style, et le tour est joué.

Nous avons donc avec CSS la possibilité de gérer le style d'un site web, de manière globale, dans devoir faire un seul copier-coller. Et la cohérence de présentation est garantie, puisque les propriétés et caractéristiques ne sont définies qu'une fois ! Ainsi, si votre site web a tous ses titres en rouge et que subitement vous voulez les passer en bleu, il n'y a qu'une seule et unique ligne à changer dans tout le site : celle qui gère la couleur du titre dans la feuille CSS !

En effet, une feuille de style CSS n'est pas liée à une page HTML en particulier. Elle peut être réutilisée sur toutes les pages d'un site, offrant ainsi la simplicité des évolutions futures ! Imaginez un site de vente en ligne avec plusieurs centaines de milliers de produits ! Si la taille du bandeau en haut change... 100 000 pages à modifier à la main ?

### 2.3. Mise en place d'une feuille de style

Il existe différentes manières d'associer des styles CSS à une page HTML, mais la plus efficace consiste à coder le CSS dans un fichier `.css` séparé dont le nom est fourni dans l'entête `<head>` du fichier HTML.

La syntaxe HTML est alors la suivante :

```
<link type="text/css" rel="stylesheet" href="chemin/fichier.css" />
```

Note : pour tous ceux qui ont déjà utilisé l'attribut « `style=...` » dans des balises HTML, vous pouvez (et vous devez !) l'oublier !

### 2.4. Exercice n°1 : mon premier document CSS

1. Créez un nouveau dossier nommé TD8 dans votre espace DropBox et copiez tous les fichiers et dossiers du TD précédent dedans. (Si vous n'avez pas fini le TD précédent, utilisez les éléments fournis pour ce TD)
2. Créez un sous dossier nommé CSS, et créez un fichier CSS à l'intérieur, nommé `06-style.css`
3. Ouvrez votre fichier `06-style.css` et écrivez ceci :

```
1  body
2  {
3      background : black;
4      color : #FFFFFF;
5      text-align : center;
6  }
```

4. Dans le fichier `01-hello_world.html`, faites ce qu'il faut pour que la feuille de style et la page HTML soient associées
5. Sauvegardez vos documents
6. Ouvrez votre fichier HTML avec votre navigateur et comparez avec le fichier du TD précédent.
7. Bravo :-)

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 2.5. La syntaxe CSS

Analysons le contenu de cette première page CSS et découvrons les possibilités du langage.

#### 2.5.1 Les sélecteurs

La première chose à définir dans un style c'est sur quoi il s'applique. Cela se fait, en utilisant un **sélecteur**. Dans notre exemple, c'est le mot « body ». Comme vous le voyez, un sélecteur peut être tout simplement un nom de balise HTML. Auquel cas, toute balise HTML rencontrée dans la page et portant ce nom se verra affectée du style qui est défini. C'est aussi simple que cela !

##### 2.5.1.1 L'ordonnement des sélecteurs

Le problème des sélecteurs, c'est justement qu'ils sont trop simples. Ils s'appliquent à toutes les balises qui respectent leurs critères.

Il est donc possible de préciser le contexte, afin d'être plus... sélectif ! Voici un comment :

- `baliseA baliseB` : applique le style à la baliseB uniquement si elle est contenue de près ou de loin dans une baliseA (baliseB est un descendant de baliseA dans l'arborescence des balises HTML)
- `baliseA > baliseB` : applique le style à la baliseB uniquement si elle est immédiatement incluse dans la baliseA (B est l'enfant immédiat de A dans l'arborescence des balises HTML)
- `baliseA + BaliseB` : applique le style à la baliseB uniquement si elle est immédiatement adjacente à la balise A (B est la sœur immédiate de A dans l'arborescence des balises HTML)
- `baliseA[attribut]` : applique le style à la baliseA uniquement si elle possède l'attribut spécifié.
- `baliseA[attribut="valeur"]` : applique le style à la baliseA uniquement si elle possède l'attribut spécifié et qu'il contient la valeur indiquée.

Notes :

- il existe d'autres sélecteurs, mais la liste est beaucoup trop longue. Voir : <https://developer.mozilla.org/fr/docs/CSS/Reference#S.C3.A9lecteurs>
- il existe un piège sur la syntaxe des sélecteurs, c'est la virgule. En effet,
  - `baliseA, BaliseB` : ne fait pas référence à l'ordre des balises, mais sert à **factoriser** du code. En fait, cela permet de regrouper les propriétés des baliseA et baliseB dans le même bloc de code, et éviter les copier-coller si on veut que ces balises aient la même mise en forme.

##### 2.5.1.2 Les id et les classes : des sélecteurs aux petits oignons

En fait, même avec l'ordonnement des sélecteurs, nous sommes un peu limités. Il est donc utile de pouvoir cibler une balise en lui donnant un nom, puis en la nommant, tout simplement, précisément et explicitement !

Les **id** sont exactement faits pour cela : lorsqu'il existe une balise spécifique, qui n'apparaît qu'UNE fois dans la page HTML (c'est une contrainte à respecter), on peut lui donner un attribut « id » et une valeur de type chaîne de caractères, qui sera son identifiant. C'est très utile notamment pour les images (logos et autres) qui n'apparaissent généralement qu'une fois.

Les **classes** fonctionnent de la même manière mais peuvent être affectées à plusieurs balises : lorsqu'il existe une balise qui existe en plusieurs exemplaires, dont on veut qu'elle se comporte d'une certaine façon, mais pas TOUTES les balises avec ce même nom, alors on peut lui donner un attribut « class » et une valeur de type chaîne de caractères, qui sera son identifiant de classe, et cela servira à définir son style.

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

Coté **HTML**, cela se passe donc ainsi :

- pour une balise unique identifiée par un ID (exemple avec un paragraphe) :
  - `<p id="mon_ID">...</p>`
- pour une balise à laquelle on donne une classe de style (exemple avec un paragraphe) :
  - `<p class="ma_classe">...</p>`

Coté **CSS**, les **sélecteurs** s'écrivent ainsi :

- pour un ID sur une balise paragraphe : `p#mon_ID { /* CSS ici...*/ }`
- pour une classe sur une balise paragraphe : `p.ma_classe { /* CSS ici...*/ }`
- pour décrire le style d'une classe qui peut s'appliquer à n'importe quelle balise, on supprime le nom de la balise et on écrit le sélecteur tout simplement : `.ma_classe { /* CSS ici...*/ }`

### 2.5.1.3 les pseudo-classes et pseudo-éléments

Comme si cela ne suffisait pas (et cela ne suffit pas !), il existe des notions supplémentaires dans les sélecteurs : les pseudo-classes et pseudo-éléments. Ils sont liées à des événements, des propriétés bien particulières ou un état courant de la balise.

La syntaxe consiste à ajouter « : » puis le nom de la pseudo-classe à un sélecteur.

Voir : <https://developer.mozilla.org/fr/docs/CSS/pseudo-classes>

et <https://developer.mozilla.org/fr/docs/CSS/Pseudo-%C3%A9l%C3%A9ments>

Il en existe de nombreux (et de plus en plus car CSS3 en ajoute !), mais la plus célèbre c'est la pseudo-classe qui gère les liens.

Exemple avec un lien de la classe `mes_liens` :

```
a.mes_liens:link { color : blue ; } /* lien non visité : bleu */
a.mes_liens:visited { color : blue ; } /* lien visité : bleu quand même ! */
a.mes_liens:hover { color : red ; } /* survol de la souris sur le lien */
a.mes_liens:active { color : black ; } /* clic en cours sur le lien */
```

### 2.5.2 Les propriétés

Les **propriétés** sont définies pour certains types de balises, et ne sont donc pas valides pour toutes. Il en existe énormément, et CSS3 vient en ajouter encore.

La syntaxe pour affecter une valeur à une propriété est très simple et se résume à ceci :

```
propriété : valeur ;
```

Dans notre exemple : `background : black ;`

Nous définissons la propriété « background » (fond de la page) à la valeur « black » la couleur noire.

### 2.5.3 Les valeurs

Les **valeurs** s'expriment différemment selon qu'il s'agit de définir la taille d'un texte, la couleur d'une page, la taille d'une image, l'alignement d'un texte, ...

Il serait beaucoup trop long de présenter toutes les possibilités et il sera utile de lire la documentation ou de se servir d'un éditeur de texte avec assistance à la saisie, mais quelques grands principes à respecter peuvent être résumés ci-dessous.

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 2.5.3.1 Texte et blocs de textes (paragraphes etc)

On peut exprimer la taille d'une police en pixels, en point, en picas, en pouces, en centimètres, en pourcentage, ...  
<http://www.w3.org/Style/Examples/oo7/units.fr.html>

La meilleure unité est le « em » : 1 em est équivalent à la taille de la police lorsqu'elle est à sa taille par défaut. Ainsi, vous laissez au navigateur (et donc à l'internaute !) le choix de la taille de la police. Et grâce à l'unité em, lorsque vous voulez qu'un titre soit plus gros ou qu'une légende soit plus petite, vous vous contentez de changer le style concerné, avec la bonne valeur de taille, exprimée en em relatifs.

Exemples :

```
font-size : 1.4em ; /* 1,4 fois la taille par défaut */  
font-size : 0.7em ; /* 0,7 fois la taille par défaut */
```

Il en est de même avec tout ce qui contient du texte ou se dimensionne comparativement au texte : il est largement préférable d'utiliser une mesure en em pour éviter que le texte ne déborde de votre conteneur.

### 2.5.3.2 Les couleurs

Les couleurs sont particulières, elles peuvent également s'exprimer de nombreuses façons, mais là, c'est selon votre goût et vos préférences.

On peut indiquer une couleur :

- par son nom (s'il est défini, et attention, seules les 16 premières couleurs sont validées par la norme !)
- avec une notation hexadécimale : 1 ou 2 caractères pour chaque composante RVB.
- utiliser les fonctions de conversion : rgb, rgba, hsl, hsla

Pour les détails et exemples, voir :

[http://www.w3schools.com/cssref/css\\_colors\\_legal.asp](http://www.w3schools.com/cssref/css_colors_legal.asp)  
[http://developer.mozilla.org/fr/docs/CSS/color\\_value](http://developer.mozilla.org/fr/docs/CSS/color_value)

### 2.5.3.3 Taille des images

Là, encore, plusieurs unités possibles : les pixels, les pourcentages et les em notamment.

En général, les pixels (px) représentent l'unité la plus cohérente. Parfois, l'image doit prendre une proportion définie de la page (la moitié par exemple), le pourcentage est alors le bienvenu. Enfin, il existe des images qui méritent de se redimensionner automatiquement lorsque la taille du texte change dans le navigateur, pour s'ajuster à la taille d'un paragraphe par exemple. Pour ces dernières, utilisez les em.

## 2.5.4 Norme / standard / conventions

### 2.5.4.1 Définition et conformité à la norme

Comme nous l'avons vu pour HTML, tout langage respecte une grammaire (ou syntaxe, ou norme). En CSS, tout comme HTML, elle est dictée par le W3C.

Mais contrairement à HTML, où les navigateur essaient de s'en sortir en cas de faute, lorsqu'il y a une erreur de syntaxe en CSS, la propriété est généralement ignorée.

### 2.5.4.2 Exercice n°2 : validation de page web.

Allez sur <http://jigsaw.w3.org/css-validator/> et validez que votre première feuille CSS est correctement structurée.

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 2.5.4.3 Conventions de codage

Une fois encore, et comme en HTML, il ne s'agit pas ici d'éléments obligatoires pour respecter le langage, mais de bonnes pratiques afin de mieux lire (et relire) son code. Un bon code est avant tout un code lisible. Pour qu'il le soit, nous allons appliquer 5 grands principes :

- les noms de propriétés et d'attributs sont forcément en minuscule.
- l'indentation du code, c'est à dire les espaces en début de ligne. Chaque fois qu'un sélecteur est défini, les propriétés et valeurs qui lui sont associées sont présentées après une tabulation. Cela permet de rapidement voir dans quel sélecteur est placé chaque propriété .
- Les sauts de ligne : après la définition d'un sélecteur, on saute une ligne, idem après chaque affectation de propriété à une valeur. (une seule affectation par ligne)
- Les accolades sont toujours seules sur leur ligne.
- lorsqu'on fait quelque chose d'un peu compliqué, (et c'est très souvent, en CSS !), ou pour ajouter de la clarté dans le code, on ajoute des commentaires !
  - La syntaxe des commentaires est la suivante : pour démarrer un commentaire, on utilise /\* et pour le terminer, on utilise \*/
  - un commentaire peut donc s'étendre plusieurs ligne, tant qu'on n'a pas rencontré \*/

## 2.6. Héritage des styles

Les balises HTML héritent des styles selon l'arborescence HTML. Lorsque vous définissez un style pour une balise, ce style est hérité à toutes ses sous balises (filles), en général.

Et si vous définissez un style à une balise fille, ce qui est déjà défini dans la balise mère reste valide et appliqué (sauf si la propriété est redéfinie dans la fille), et les deux styles se combinent.

## 3. CSS : Mise en pratique

Comme pour HTML, après cette très longue (mais nécessaire) présentation des concepts et de la syntaxe, nous allons (enfin !) passer aux exercices pratiques.

Vous allez vite constater qu'il existe énormément de propriétés et autant de contraintes sur les valeurs. Il est donc utile de les manipuler pour voir leur fonctionnement et mieux mémoriser ce qu'elles permettent.

L'objectif va, à nouveau, être de vous faire découvrir les propriétés par vous même, et qu'à la fin des exercices votre page de référence HTML du précédent TD y ait gagné en convivialité.

Et n'oubliez pas de regarder le résultat de votre travail dans un navigateur, après chaque modification de votre fichier CSS, en rechargeant la page (F5 ou Ctrl-R).

### 3.1. Exercice n°3 : Les propriétés générales de la page HTML

1. Dans votre dossier CSS, créez une nouvelle feuille de style que vous nommerez 07-style\_refcard.css.
2. Ouvrez la page HTML 02-refcard.html et faites ce qu'il faut pour que la feuille de style qu'on vient de créer y soit liée correctement. Tous les exercices de ce TD utilisent ces deux fichiers.
3. Définissez un style pour la balise body, et à l'aide de la propriété background-color, choisissez la couleur de fond silver.
4. Définissez couleur #000 pour le texte, grâce à la propriété color.
5. Les documents sont beaucoup plus lisibles à l'écran avec une police sans-serif. Grâce à la propriété font-family, faites les modifications adéquates.
6. Sauvegardez vos documents sur votre espace DropBox.

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 3.2. Exercice n°4 : Formatage du texte

1. Faites un style pour le titre h1. Il devra être italique, d'une dimension de 3em et centré. Utilisez les propriétés `font-size`, `font-style` et `text-align`.
2. Faites un style pour les titres h2, qui seront en dimension 2em en vert, et souligné. Utilisez notamment la propriété `text-decoration`.
3. Faites un style pour les paragraphes, qui seront en alignement justifié.
4. Modifiez le style des citations pour qu'elles apparaissent en petites majuscules, utilisez la propriété `font-variant`. Améliorez la lisibilité mettant 0.1em dans la propriété `letter-spacing`.
5. Modifiez le style des termes de définition pour qu'ils soient en gras. Utilisez la propriété `font-weight`.
6. Faites en sorte que la première ligne des paragraphes soit indentée de 2em, avec `text-indent`.
7. Sauvegardez vos documents sur votre espace DropBox.

### 3.3. Exercice n°5 : Reformatage d'éléments

1. Modifiez le style des abréviations pour que lorsqu'on les survole, le curseur (`CURSOR`) soit un point d'interrogation.
2. Modifiez le style des textes « mis en valeur », pour qu'ils soient en gras en plus d'être en italique.
3. À l'aide de la propriété `list-style-type` (et de l'ordonnancement des sélecteurs !), faites en sorte que les éléments de liste non ordonnée (et uniquement ceux-ci) ait un cercle plein (disque).
4. Modifiez la couleur du surlignage pour qu'il soit rose (ne vous posez pas trop de questions sur la propriété, simplement, réfléchissez bien !)
5. Sauvegardez vos documents sur votre espace DropBox.

### 3.4. Exercice n°6 : Formatage des liens, id, classes et pseudo-classes

1. Il y a 6 liens dans la page HTML sur laquelle nous travaillons. Faites en sorte que les 2 premiers liens soient affichés en blanc (utilisez un sélecteur de type `CLASSE`).
2. Le 3<sup>e</sup> lien doit être rouge, avec un soulignement qui n'apparaît que lorsqu'on survole le lien à la souris (utilisez un sélecteur de type `id`).
3. L'avant dernier lien sera d'une taille fixée à 5 pixels.
4. Les 4<sup>e</sup> et 6<sup>e</sup> liens seront en jaune, et passeront orange uniquement lorsqu'on clique dessus, tout en doublant de taille de police.
5. Sauvegardez vos documents sur votre espace DropBox.

### 3.5. Exercice n°7 : Formatage des images

1. Redimensionnez l'image pour qu'elle ne puisse pas déborder d'un rectangle de 150x100px. Utilisez les propriétés `max-width` et `max-height`.
2. Ajoutez une bordure de 3 pixels en pointillés et de couleur orange. Regardez les propriétés `border` et associées (`border-size`, etc).
3. Sauvegardez vos documents sur votre espace DropBox.

### 3.6. Exercice n°8 : Formatage des tableaux

1. Faites en sorte que le tableau soit affichée ainsi :

## Création et gestion de documents (C2i)

### TD 8/12 : CSS

Nom	Prénom	Age
Bob	L'éponge	8
McQueen	Flash	21

*L'age de certaines personnes*

Note : Voici les caractéristiques requises :

- le texte est calé à droite
- la bordure est un trait noir, d'un pixel de large.
- la marge intérieure (padding) des cellules est de 0,5em
- la légende est en bas, de taille 0,7em, en italique.

Aide : Pour y arriver, vous devrez notamment vous renseigner et utiliser les propriétés suivantes :

- padding
  - caption-side
  - border-collapse
2. Validez que votre feuille CSS est valide (avec l'outil du W3C) , tout comme votre page HTML.
  3. Sauvegardez vos documents sur votre espace DropBox.

### 3.7. CSS avancé

#### 3.7.1 Exercice n°9 : Pseudo-éléments

1. A l'aide du pseudo élément `first-letter`, faites en sorte que la première lettre de chaque titre de niveau 2 soit en blanc et avec une taille de 150 %.
2. Sauvegardez vos documents sur votre espace DropBox.

#### 3.7.2 Exercice n°10 : Image de fond

1. Faites en sorte que la page ait le logo de l'université en image de fond :
  - (a) modifiez le style de la balise `body`, à l'aide de la propriété `background-image`, puis de la fonction `url()`, choisissez dans le dossier `images`, votre image `05-logo_uns`. Faites attention aux chemins des images ! Vous devez travailler en relatif par rapport à l'emplacement de votre fichier CSS !
2. Sauvegardez votre feuille de style et visualisez le résultat. C'est illisible !
3. À l'aide de la propriété `background-repeat`, désactivez l'effet de mosaïque.
4. Mieux ? Oui mais pas parfait. Trouvez la hauteur de votre image en pixels. A l'aide de la propriété `padding-top` (marge intérieure supérieure), libérez un espace de la hauteur de votre image, exprimé en pixels.
5. Sauvegardez vos documents sur votre espace DropBox.

#### 3.7.3 Exercice n°11 : Changer la police (premier exercice en CSS3 !)

1. Allez sur <http://www.google.com/fonts>, et cherchez la police nommée Gochi Hand, puis choisissez l'option « quick use ».
2. Appliquez les instructions pour faire en sorte que votre balise `h1` soit affichée avec la police Gochi Hand .
3. Validez que votre feuille CSS est valide (avec l'outil du W3C) , tout comme votre page HTML.
4. Sauvegardez vos documents sur votre espace DropBox.

# Création et gestion de documents (C2i)

## TD 8/12 : CSS

### 4. Liens et références

Sources bien utiles pour créer ce TD, ainsi que des références très intéressantes pour les webmasters :

<http://www.w3.org>

<http://validator.w3.org/>

<http://jigsaw.w3.org/css-validator/>

<http://www.w3schools.com/cssref/>

<https://developer.mozilla.org/fr/docs/CSS/Reference>

<http://fr.openclassrooms.com/informatique/html/cours>

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

<http://www.w3schools.com/html/default.asp>

<http://www.alsacreations.com/tutoriels/>

<http://www.html5rocks.com/en/resources>

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

<http://diveintohtml5.info/>

<http://html5doctor.com/>