

SCI6005 Information numérique et informatique documentaire (H2022)

Christine Dufour, EBSI, UdeM

H2022 24 février 2022

Cours 7 : Développement de sites Web : CSS, ergonomie et accessibilité

Table des matières

I - Cours 7 - Développement de sites Web : CSS, ergonomie et accessibilité	3
1. + Au programme aujourd'hui	3
2. + Alignement pédagogique	3
3. Introduction	4
4. Feuilles de styles CSS	4
4.1. Syntaxe de base CSS	5
4.2. Intégration des styles	8
4.3. Application des styles	10
4.4. Validation	11
5. Ressources en lien avec le cours.....	11
Glossaire	12
Index	13
Crédits des ressources	14

Cours 7 - Développement de sites Web : CSS, ergonomie et accessibilité



1. + Au programme aujourd'hui

- CSS

2. + Alignement pédagogique



Objectifs visés, matériel du cours et évaluation : Examen mi-session

Lien entre les objectifs, les compétences à développer et le matériel du cours 7

Objectif général : Comprendre la place des technologies et de l'information numérique en contexte documentaire		
Objectifs spécifiques	Compétences à développer	Matériel associé
Décrire les principales technologies utilisées en milieu documentaires	Comprendre l'utilité des pages CSS (séparation contenu-traitements)	Section <i>Feuilles de styles CSS > Introduction</i>
	Connaître la syntaxe d'une règle CSS (déclaration, propriété, valeur)	Section <i>Feuilles de styles CSS > Syntaxe de base CSS</i>
	Savoir utiliser des propriétés CSS pour faire de la mise en page	Section <i>Feuilles de styles CSS > Syntaxe de base CSS > Propriétés CSS</i> <u>Glossaire des propriétés CSS</u> ¹ TP Site Web
	Distinguer l'utilisation d'une classe de l'utilisation d'une règle s'appliquant à un élément précis	Section <i>Feuilles de styles CSS > Syntaxe de base CSS > Sélecteurs</i>
	Expliquer l'importance de la validation des feuilles de styles CSS	Section <i>Feuilles de styles CSS > Validation</i>

¹https://cours.ebsi.umontreal.ca/glossaireweb/index_css.php?cours=sci6005



Objectifs visés, matière du cours et activités associées

Lien entre les objectifs, la matière du cours 7 et les activités associées

Objectif général : Concevoir et implanter des systèmes et services d'information numérique		
Objectifs spécifiques	Matière	Activités
Exploiter la structuration de l'information numérique dans différents contextes	Propriétés CSS	TP Site Web
	Propriétés CSS	<u>Pratique CSS sur StudiUM</u> ¹

3. Introduction

La séparation que l'on retrouve pour un format de document structuré entre son contenu et sa mise en forme permet de faciliter la **réutilisation** des contenus. Cette mise en forme peut se faire par l'utilisation des feuilles de styles. Il existe différents langages de feuilles de styles dont CSS (*Cascading Style Sheets*) qui peut être utilisé avec HTML, XHTML et XML. XSL (*eXtensible Style Language*) et XSLT (*eXtensible Style Language with Transformation*) sont deux autres exemples de formats de feuilles de styles qui peuvent être utilisés avec XML.

Dans le cadre du cours, nous verrons les bases du langage CSS pour la mise en forme de pages HTML, afin de bien en comprendre la logique. CSS permet d'aller toutefois bien plus loin comme le site [css Zen Garden](http://www.csszengarden.com/)² permet de le découvrir!



Si vous désirez développer vos connaissances des feuilles de styles et de l'ergonomie du développement Web, le cours [INU3051 Information et sites Web](https://admission.umontreal.ca/cours-et-horaires/cours/inu-3051/)³ est pour vous!

4. Feuilles de styles CSS

Le langage CSS est une recommandation du W3C depuis 1996. Il est préconisé pour le stylage des documents XHTML et HTML depuis HTML 4.0. On peut y voir plusieurs avantages :

- Il est **compatible**, du moins en partie, avec tous les **navigateurs récents**.
- C'est un langage qui se base sur la **structure des documents**.
- La **séparation des contenus et des traitements** qu'il permet augmente l'efficacité de la **réutilisation** de l'information et de la **gestion des documents**.
- Il est possible de définir **plus d'une feuille de styles** pour un site en fonction du médium utilisé (imprimé, écran, etc.). Il est ainsi possible, par exemple, d'améliorer l'**accessibilité** des sites pour les personnes malvoyantes en leur définissant une feuille de styles leur facilitant la lecture.
- Cela permet aussi une plus grande **uniformité** de présentation pour l'ensemble des pages d'un site.
- C'est un langage très **riche** et **flexible**.

¹<https://studium.umontreal.ca/mod/quiz/view.php?id=3218504>

²<http://www.csszengarden.com/>

³<https://admission.umontreal.ca/cours-et-horaires/cours/inu-3051/>

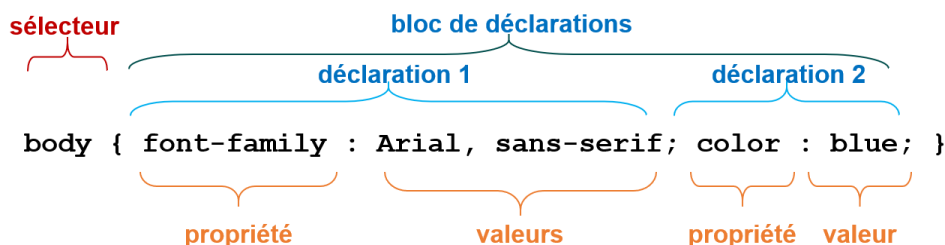
- En sortant les styles des fichiers HTML pour les conserver dans une feuille de styles externe, les fichiers HTML s'en trouvent **allégés**.

4.1. Syntaxe de base CSS

Règles CSS : Principes de base

Une feuille de styles CSS est une séquence de **règles CSS** qui s'appliquent soit (1) à un *élément HTML en particulier* (par exemple pour changer le visuel d'un paragraphe `<p>`) ou (2) à une *classe d'éléments qui peut être utilisée avec différents éléments HTML*.

Une règle CSS est composée d'un **sélecteur** qui indique ce qui est modifié suivi d'un **bloc de déclarations** qui définissent les *propriétés modifiées* et leurs *valeurs*. Par exemple :



Dans l'exemple ci-dessus, le **sélecteur** est l'élément `<body>`. Le **bloc de déclarations** est indiqué entre accolades `{ }`. Le bloc de déclarations de l'exemple comporte **deux déclarations** (chaque déclaration est séparée par un *point-virgule*). Chacune des déclarations permet d'associer une **caractéristique de mise en forme** à une **valeur spécifique**. Plus précisément :

- La première déclaration précise le visuel pour la **police de caractères** (`font-family`) pour la police spécifique *Arial* ou pour la famille générique des polices de caractères *sans empattement* (*sans serif*) (si la police *Arial* n'est pas installée sur le poste).
- La deuxième déclaration précise la **couleur de la police** de caractères (`color`) qui sera bleue.

On retient donc, au niveau de la syntaxe pour une règle CSS :

- La règle débute par le **sélecteur**.
- Le bloc de déclarations est indiqué entre **accolades**.
- Les déclarations sont séparées par un **point-virgule**.
- La propriété d'une déclaration est séparée de sa valeur par un **deux-points**.

On peut définir des **sélecteurs multiples** pour un **même bloc de déclarations** en les séparant par une virgule. Par exemple la règle ci-dessous permet la mise en gras à la fois des titres de niveau 2 `<h2>` et 3 `<h3>` :

```
1 h2, h3 {font-weight: bold;}
```

Sélecteurs

Les **sélecteurs** dans une règle CSS permettent de **limiter la portée** du bloc de déclarations. On retrouve quatre principaux types de sélecteurs :

1. Un **élément HTML**;
2. Une **classe d'éléments HTML**;
3. Un **élément HTML qualifié par une classe**;
4. L'élément **a** avec ses différents statuts.

L'exemple ci-dessous illustre ces différents types de sélecteurs qui sont décrits par la suite :

```

1 strong {color: blue;} /* emphase forte à écrire en bleu */
2
3 p>strong {color: blue;} /* emphase forte à l'intérieur des paragraphes à écrire en bleu */
4
5 *.copyright {font-size: 80%;} /* classe pour réduire la taille de la police de caractères qui peut s'appliquer à
   n'importe quel élément */
6
7 p.avertissement {color: red;} /* classe nommée « avertissement » applicable uniquement à des paragraphes
   pour les écrire en rouge */
  
```

```

8
9 a:link {color: blue; text-decoration: none;} /* liens écrits en bleu sans soulignement */
10 a:visited {color: #33CC33; text-decoration: line-through;} /* liens visités écrits en vert et barrés */
11 a:hover {color: blue; text-decoration: underline; background-color: #eff8ad;} /* liens survolés écrits en bleu
soulignés et avec fond jaune */

```

Les lignes 1 et 3 de l'exemple ci-dessus illustrent le premier cas d'une règle qui s'applique sur un **élément** en général (`strong`) ou un élément dans un contexte particulier (`strong` à l'intérieur de `p`).

Les lignes 5 et 7 définissent une règle pour des **classes**. La première classe (*copyright*) peut être associée, dans la page HTML, à n'importe quel élément (c'est l'astérisque devant le nom de la classe qui indique qu'elle peut s'appliquer à tous les éléments; cet astérisque pourrait être omis). Elle pourrait être associée, par exemple, à un élément `p` tel qu'illustré ci-dessous :

```
1 <p class="copyright">&copy; John Smith</p>
```

La ligne ci-dessus est une ligne qui se trouverait **dans le fichier HTML**. La manière d'appliquer une classe consiste à ajouter à la balise d'ouverture de l'élément que l'on veut modifier, ici `p`, un **attribut class** et d'indiquer comme valeur de cet attribut le **nom de la classe**. Notez l'absence ici du point qui précède le nom de la classe. Ce point n'est présent que dans le fichier CSS pour indiquer qu'il s'agit d'une classe.

La deuxième classe (*avertissement*) ne peut être associée qu'à des éléments `p`. Si vous l'associez par erreur dans votre fichier HTML à un autre élément, la règle ne serait pas appliquée. On pourrait ainsi l'utiliser, dans le fichier HTML, pour modifier la couleur de la police de caractères d'un paragraphe en particulier comme suit :

```
1 <p class="avertissement">Attention : N'oubliez pas de faire valider vos pages avant de les mettre sur le Web!
</p>
```

Il est important de bien comprendre la différence entre l'utilisation d'une règle pour un élément en particulier et une règle pour une classe. Dans le premier cas (*règle pour un élément*), si la propriété `color: red` est définie pour l'élément `p`, **tous les éléments `p` de toutes les pages Web** associées à la feuille de styles seront en rouge. Dans le deuxième cas (*règle pour une classe*), si la propriété `color: red` est associée à la classe *avertissement*, **seuls les éléments `p` où on a ajouté `class="avertissement"` seront en rouge**.

Finalement, les lignes 9 à 11 illustrent les différents états possibles d'un lien hypertextuel : visuel de base (`link`), visuel d'un lien déjà visité (`visited`) et visuel si on survole un lien avec la souris (`hover`).

Commentaires et casse

On peut ajouter des commentaires dans un fichier CSS en les faisant précéder par `/*` et suivre par `*/`. Par exemple :

```
1 /* Règles pour les tableaux */
2 table {font-family:Arial;}
```

Cela permet de **documenter la feuille de styles** ainsi qu'à **masquer** certains éléments de mise en page.

La casse (majuscules/minuscules) n'est pas prise en compte. Les espaces et les sauts de ligne avant et après les signes de ponctuation sont ignorés.

Couleurs

Il y a plusieurs façons de **spécifier une couleur**, par exemple :

- Par **mots-clés** : `black`, `white`, `gray`, `yellow`, `red`, `blue`, `green`, etc.
- Par **triplet de composantes rouge/vert/bleu (RGB)** dans une des quatre formes suivantes
 - `#rrggbb` (ex. : `#008000`)
 - `#rgb` (ex. : `#080`)
 - `rgb(a,b,c)` où `a`, `b` et `c` sont des nombres entiers entre 0 et 255 (ex. : `rgb(0,128,0)`)
 - `rgb(d%,e%,f%)` où `d`, `e` et `f` sont des nombres entre 0 et 100 (pourcentages) (ex. : `rgb(0%,50%,0%)`)
- Par **triplet de composantes teinte (hue)/saturation/lumière (lightness) (HSL) (CSS3)** : `hsl(x,y,z)` où `x` est un nombre entre 0 et 360 (un degré sur la roue des couleurs), et `y` et `z` sont des pourcentages (un nombre entre 0 et 100) (ex. : `hsl(120,100%,25%)`)

La question des couleurs en est une délicate. Il est en effet bien facile de faire des mauvais choix! Heureusement, on retrouve plusieurs ressources sur le Web pour aider celles et ceux qui n'ont pas l'œil pour les couleurs comme, par exemple, <http://www.w3schools.com/colors/default.asp>.

Unités de mesure

On peut vouloir préciser la **taille de différents éléments** (par exemple, la taille d'une image ou d'une police de caractères). Pour ce faire, il est possible d'utiliser des *valeurs absolues* ou des *valeurs relatives* :

- Les unités de mesure en **valeurs absolues** ont une **mesure fixe** comme, par exemple, des centimètres (*cm*), des millimètres (*mm*), des points typographiques (*pt*) et des pixels en relation avec la résolution (*px*).
- Les unités de mesure en **valeurs relatives** sont **proportionnées par rapport à d'autres éléments**. C'est le cas par exemple de l'unité *em* qui correspond à la largeur de la lettre "m" dans la police courante et de l'unité *ex*, qui est la hauteur de la lettre "x" dans la police courante. Il est aussi parfois possible d'utiliser un pourcentage qui fait référence à une autre valeur telle que définie dans la propriété (par exemple pour redimensionner une image). Le pourcentage se calcule le plus souvent en relation avec la taille de la police courante ou la taille de l'élément parent.

Bien que les navigateurs gèrent maintenant de manière plus intelligente les valeurs absolues lorsque l'on fait un zoom, il n'en demeure pas moins qu'elles sont déconseillées dans certains cas pour laisser plus de flexibilité à l'utilisateur pour redimensionner une page (par exemple pour la taille de la police).

CSS et Médias

Depuis la deuxième version de CSS, on peut spécifier comment représenter un document en fonction du **média utilisé par le lecteur**. Les principaux types de médias sont : *all*, *print*, *screen* et *speech* (https://www.w3schools.com/cssref/css3_pr_mediaquery.asp).

On peut ainsi associer à une même page HTML plusieurs feuilles de styles pour différents médias en faisant le lien, dans l'entête, vers les différentes feuilles grâce à des éléments `link`. Par ex. :

```
1 <link type="text/css" rel="stylesheet" href="css/screen.css" media="screen" />
2 <link type="text/css" rel="stylesheet" href="css/print.css" media="print" />
```

On peut aussi inclure, dans une même feuille de styles, des blocs de déclarations selon les médias :

```
1 /** Print styles **/
2 @media print { ... }
```

Propriétés CSS

Il existe un **très grand nombre** de propriétés dans le langage CSS. Dans le cours, nous explorerons les propriétés les plus importantes :

- *Arrière-plan*
 - background-color
 - background-image
- *Bordure*
 - border-color
 - border-style
 - border-width
- *Espacement/visibilité/positionnement/taille*
 - display
 - float
 - margin
 - padding
 - width
- *Liste*
 - list-style-type
- *Police*

- font-family
- font-size
- font-style
- font-weight
- *Texte*
 - color
 - text-align
 - text-decoration
 - text-indent

Ces propriétés sont décrites dans le [glossaire des propriétés CSS](#)¹.

Pour des **listes plus complètes**, voir, par exemple :

- [W3Schools : CSS Tutorial](#)²
- [Complete CSS Guide](#)³

4.2. Intégration des styles

Les styles CSS peuvent être **intégrés** dans une page Web de trois manières :

1. **À même les balises d'ouverture** des éléments dans une page Web (=intraligne)
2. Dans l'**en-tête d'une page Web** (=interne)
3. Dans un **fichier externe** (=externe)

Chacune de ces méthodes possède ses avantages et désavantages, comme vous le constaterez à la lecture des sections qui suivent. Dans le cadre du *TP Site Web*, la **troisième méthode** (*fichier externe*) est privilégiée.

a) Intégration des styles dans les balises d'ouverture des éléments

Il est possible d'intégrer des règles CSS directement dans la page HTML dans la balise d'ouverture de l'élément HTML dont on veut modifier le visuel. Cela se fait en lui ajoutant un attribut `style` qui aura pour valeur les règles CSS désirées comme illustré ci-dessous. La couleur de la police de caractères de l'élément `p` y est redéfinie en argenté.

```
1 <p style="color: silver">titre</p>
```

Cette méthode permet de **modifier à la pièce**, pour des *besoins particuliers*, une instance d'un élément. Elle est toutefois **déconseillée** lorsqu'il s'agit de définir le **visuel pour toutes les instances** d'un élément en particulier dans l'ensemble d'un site Web. En effet, en ce cas, il faudrait ajouter l'attribut `style` à chacune des instances de l'élément, ce qui représente bien du travail. De plus, si vous décidez de modifier le visuel de votre site Web, il faudra modifier toutes les valeurs des attributs `style` ainsi ajoutés.

b) Intégration des styles dans l'en-tête d'une page Web

Il est possible d'insérer des règles CSS dans l'entête des pages HTML (section `head`) à l'aide d'un élément `style` comme l'illustre l'exemple ci-dessous :

```
1 <head>
2 <style type="text/css">
3   body {background-color: gray;}
4 </style>
5 </head>
```

¹https://cours.ebsi.umontreal.ca/glossaireweb/index_css.php?cours=sci6005

²<http://www.w3schools.com/css/>

³http://www.westciv.com/style_master/academy/css_tutorial/

Cette méthode permet de **modifier l'ensemble du visuel d'une page Web**. Elle est à privilégier uniquement pour des **cas particuliers** où vous désirez, pour une page, un visuel différent du reste du site Web. Tout comme la première méthode présentée, elle est **déconseillée** pour définir les **styles de base du site Web au complet** comme il faudra faire l'insertion de l'élément `style` sur toutes les pages du site, ce qui peut devenir fastidieux si le site comporte plusieurs pages. De plus, si vous désirez changer le visuel de votre site Web, il faudra modifier les entêtes de toutes les pages Web.

c) Intégration des styles dans un fichier externe

La troisième et dernière méthode présentée est celle qui est **conseillée** lorsque l'on définit le **visuel de base d'un site Web**. Il s'agit d'inclure toutes les règles CSS dans un **fichier séparé** et de pointer vers ce fichier à partir des pages du site Web à l'aide d'un élément `link` que l'on inclut dans l'entête des pages (`head`) de cette manière :

```
1 <link rel="stylesheet" type="text/css" href="css/styles.css" />
```

On indique, dans cet élément `link`, le chemin vers le fichier CSS à l'aide de l'attribut `href`. Pour l'exemple ci-dessus, il s'agit d'un chemin relatif, comme le fichier CSS est sur le même serveur que la page HTML (ce qui est habituellement le cas). Ce fichier CSS, par rapport à la page HTML, se trouve dans un dossier `css` et se nomme `styles.css`. C'est la méthode que vous utiliserez pour le *TP Web*.

Exemple d'une feuille de styles externe



```
1 @charset "utf-8";
2
3 @media all {
4
5   body {
6     font-family: Arial, Helvetica, sans-serif;
7   }
8
9   h1 {
10    color: #FF6600;
11    font-weight: bold;
12    font-size: 1.8em;
13  }
14
15  ul {
16    list-style-type: square;
17  }
18
19  *.notes {
20    font-size: 0.7em;
21    margin-bottom: 0em;
22  }
23 }
```

Si on examine l'exemple de plus près, on remarque :

- L'indication, en première ligne, du **jeu de caractères** utilisé;
- La précision, en deuxième ligne, du **media** auquel s'appliquent les styles (ici, cela s'applique à tous les médias);
- La présence de **propriétés CSS** pour la définition du **visuel** pour les éléments `body` (polices de caractères), `h1` (couleur de la police de caractères, emphase, taille) et `ul` (type de puces);
- La présence d'une **classe** `notes` qui pourra s'appliquer à n'importe quel élément pour diminuer la taille de la police de caractères et enlever la marge inférieure;
- L'utilisation de l'**indentation** et de **retours de ligne** pour séparer les propriétés pour faciliter la lecture de la feuille de styles.

4.3. Application des styles

Comme nous l'avons vu, les caractéristiques de mise en forme peuvent provenir de **différentes sources** :

- Des feuilles de styles externes ou internes;
- Des attributs `style` dans les balises d'ouverture d'éléments HTML;
- Des paramètres du navigateur
 - Paramétrage personnalisé par l'utilisateur
 - Style par défaut du navigateur pour ce type d'élément

Il est ainsi possible, pour un élément, de retrouver plusieurs règles CSS définies. Qu'arrive-t-il en ce cas? Les deux sections qui suivent présentent quelques principes de base pour comprendre les différentes possibilités.

a) Héritage et fusion des règles CSS

Tous les éléments d'un document HTML n'ont pas besoin d'être présents dans une feuille de styles. Si aucune règle n'est spécifiée pour un élément, il y a deux possibilités :

- Dans certains cas, l'élément peut hériter (notion d'*héritage*) des règles spécifiées pour l'élément parent (c'est-à-dire l'élément dans lequel il se trouve).
- Si pour un élément il n'y a pas d'héritage ou si aucune règle CSS n'est précisée pour son élément parent, il sera restitué en fonction des paramètres par défaut du navigateur.

Par exemple :

```
1 body {color: red; font-family: sans-serif; font-size: 12pt;}
2 h1 {color: blue;}
```

Dans l'exemple ci-dessus, un élément `h1`, comme il est un enfant de l'élément `body`, serait présenté avec :

- `color: blue` en raison de la règle définie à la ligne 2.
- `font-family: sans-serif ; font-size: 12pt` en raison de la règle définie à la ligne 1. La propriété `color: red` de la ligne 1 ne serait pas appliquée, comme une autre couleur est définie par la suite à la ligne 2.

Donc, si plus d'une règle s'appliquent au même élément, les propriétés sont "fusionnées". Par ex. :

```
1 h1, h2, h3 {color: red;}
2 *.important {font-size: 200%;}
```

Dans l'exemple ci-dessus, un élément `h2` avec `class="important"` serait présenté avec `font-size:200%` ET `color: red`.

b) Priorité des règles CSS

Si **plusieurs sources spécifient des valeurs différentes** pour la même propriété, la **priorité** suivante s'applique :

1. Attribut `style` dans la balise de début de l'élément
2. Feuille de styles interne
3. Feuille de styles externe
4. Paramétrage personnalisé du navigateur
5. Style par défaut du navigateur

Par exemple, si vous voulez "surpasser" une règle définie dans une feuille de styles externe dans un cas particulier, il suffit de préciser les règles CSS soit dans l'entête de votre page HTML (*feuille de styles interne*) ou directement dans la balise d'ouverture de l'élément à modifier en lui ajoutant un attribut `style`.

Si des valeurs différentes pour la même propriété sont définies à l'intérieur d'une **même feuille de styles** (interne ou externe), la priorité dépend de l'ordre d'**apparition** : la règle apparaissant **en dernier** aura préséance sur les autres.

4.4. Validation

Validation CSS

Tout comme pour la validation HTML, la validation CSS vise à s'assurer du **respect de la norme CSS utilisée**. Une feuille de styles valide augmente l'**accessibilité** d'un site. La validation peut se faire **manuellement** ou à partir d'un **validateur** comme celui du W3C : <http://jigsaw.w3.org/css-validator/>. Il est ici aussi recommandé de **valider au fur et à mesure** de la construction et d'**alterner** entre *validation* et *corrections* (une erreur pouvant avoir des incidences multiples).

Autres types de validation d'un site Web

En sus de la validation HTML et de la validation CSS, d'autres types de validation peuvent être effectués pour augmenter la qualité et l'accessibilité d'un site Web :

- Détection des **liens brisés** (url pointant vers un fichier qui n'est plus accessible) sur une page, par exemple <http://validator.w3.org/checklink>
- Vérification de l'**accessibilité d'une page**, particulièrement pour les personnes handicapées, par exemple <http://wave.webaim.org/>
- Vérification de l'**accessibilité d'une page** pour les personnes daltoniennes, par exemple <https://www.toptal.com/designers/colorfilter> [pour valider une page] et <http://www.color-blindness.com/coblis-color-blindness-simulator/> [pour valider une image]

5. Ressources en lien avec le cours

Matériel de cours

- *Notes de cours (cf. sci6005_h2022_c7_notes_cours)*

Ressources complémentaires

- [Glossaire des propriétés CSS vues dans le SCI6005](#)¹
- [Pratique CSS sur StudiUM](#)²

Protocole du TP Site Web³

¹https://cours.ebsi.umontreal.ca/glossaireweb/index_css.php?cours=sci6005

²<https://studium.umontreal.ca/mod/quiz/view.php?id=3880099>

³https://studium.umontreal.ca/course/view.php?id=212980§ion=3#tp_web [accès restreint]

Glossaire



Empattement

On retrouve en typographie deux familles de police de caractères, soit les **polices avec empattement** (*serif*) comme Times New Roman et les **polices sans empattement** (*sans serif*) comme Arial. L'empattement est la petite ligne horizontale que l'on retrouve dans le dessin d'une lettre au bas de ses traits verticaux. Voir par exemple [https://fr.wikipedia.org/wiki/Empattement_\(typographie\)](https://fr.wikipedia.org/wiki/Empattement_(typographie)) pour des exemples.

Index



Commentaires.....	5
Couleurs.....	5
Déclaration CSS.....	5
Feuille de styles externe.....	8
Feuille de styles interne.....	8
Fusion.....	10
Héritage.....	10
Médias.....	5
Priorités.....	10
Propriétés CSS.....	5
Ressources en lien avec le cours.....	11
Sélecteurs.....	5
Style intraligne.....	8
Syntaxe CSS.....	5
Unités de mesure.....	5
Validation.....	11
Validation CSS.....	11

Crédits des ressources



p. 3

<http://creativecommons.org/licenses/publicdomain/4.0/fr/>, johnny_automatic

p. 4

<http://creativecommons.org/licenses/publicdomain/4.0/fr/>, maoriveros