



Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

<http://cellierfranck.alwaysdata.net>



MS LANGAGE & DEVELOPPEMENT POUR LE WEB

HTML 5 & CCS 3 – SYNTAXE & LEXIQUE.

A. PRINCIPES

Maintenant que nous savons à quoi ressemble le HTML5 et ce que nous pouvons faire avec, il est important de **comprendre les différences entre ce langage et le CSS3, leur syntaxe, et une certaine terminologie commune.**

Le **HTML5** est un langage créé pour structurer et donner du sens au contenu. Le **CSS3**, également connu sous le nom de « feuilles de style en cascade », est un langage de présentation créé pour donner du style au contenu, pour jouer sur son apparence. En résumé, le HTML5 détermine la structure et la signification du contenu sur une page Web pendant que le CSS3 détermine le style et l'apparence de ce contenu. Ces deux langages sont indépendants l'un de l'autre.

Le CSS3 ne doit pas résider dans un document HTML5 et vice-versa (fortement conseillé).

Prenons un exemple, l'élément HTML5 *p* est utilisé pour afficher un paragraphe de texte sur une page web. Le CSS3 utilise un sélecteur pour intervenir sur l'élément *p* en changeant la couleur, la taille de police, le poids de la police, et d'autres propriétés ...

B. LE LEXIQUE HTML 5

Les trois termes les plus importants : **Élément**, **Balise** et **Attribut**.

Eléments

Les éléments sont des indicateurs définissant des objets dans une page, comprenant la structure et le contenu. Quelques-uns des éléments les plus populaires sont les titres (*h1* à *h6*), les liens (*a*), les mots en gras (*strong*) ...

1 | ``

Les balises ou Tags (en anglais)

Les éléments sont souvent faits de plusieurs ensembles de balises, identifiés comme étant d'ouverture et de fermeture. La balise d'ouverture marquera le début d'un élément, tel que `<div>`. La balise de clôture marquera la fin d'un élément et commencera par une barre oblique, comme `</div>`.

```
1 | <p> .... </p>
```

Pour ne pas confondre balises et éléments, considérons l'exemple suivant :

```
1 | <p> L'élément P contient des mots en <strong>Gras</strong>. </p>
```

L'élément *p* contient du texte et des balises. L'élément *strong* ne contient que du texte. Si en CSS3 on applique un style à l'élément *p*, par exemple une couleur de texte rouge, alors tout le contenu de l'élément *p* sera affiché en rouge. Même le texte en Gras.

L'élément *p* contient également la balise d'ouverture `` et la balise de fermeture ``.

Les Attributs

Les attributs sont des propriétés utilisées pour fournir des instructions supplémentaires aux éléments donnés. Plus généralement, les attributs sont utilisés pour attribuer un **id** (identifiant), un **class** (sélecteur CSS3), une **description** d'élément (`alt`), une **source** (`src`), ou pour fournir une référence de **lien** hypertexte (`href`).

```
1 | 
```

Dans la ligne de code ci-dessus, la balise `` (qui se ferme toute seule : Comprendre le principe de balises imbriqués) possède deux attributs : `src` et `alt`.

L'attribut `src` indique à la balise `` l'adresse de l'image à afficher tandis que l'attribut `alt` indique un texte alternatif décrivant l'image s'affichant au cas où celle-ci ne serait pas accessible.

C. STRUCTURE ET SYNTAXE D'UN DOCUMENT HTML 5

Tous les documents HTML ont une structure minimale qui contient les éléments suivant : **doctype**, **html**, **head** et **body**.

Suite à la déclaration doctype (En savoir plus sur [le Doctype HTML5](#)), l'élément `<HTML>` signifie le début et la fin du document.

L'élément *head* du document est utilisé pour décrire toutes les méta-données, le titre du document et des liens vers des fichiers externes (relier notre document à une feuille de style CSS3 par exemple). Tout ce qui est inclus dans les balises `<head>` n'est pas visible pour les internautes. Il s'agit uniquement de renseignements indiqués au navigateur.

En général, la structure d'un document HTML5 se présente comme suit :

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <title>Titre de ma page</title>
6   </head>
7   <body>
8     <h1>Bonjour à tous</h1>
9     <p>Ceci est ma page web</p>
10  </body>
11 </html>
```

Les balises META

C'est aussi entre ces balises `<head>` que vous placerez les fameuses balises `<meta>` censées améliorer votre référencement. Bien que leur importance diminue au fil du temps, je vous conseille malgré tout de les laisser. Parmi toutes les balises `<meta>`, les plus utilisées sont les suivantes :

CONTENT-TYPE : Indique la nature du document et le jeu de caractères utilisés.

TITLE : Indique le nom de votre page web. Bien qu'il soit préférable d'utiliser la balise `<title>`.

DESCRIPTION : Contient la description de votre page. Bien qu'elle ne soit pas affichée, elle sera retrouvée par les moteurs de recherches, juste sous le titre de votre page. Il est conseillé de ne pas dépasser les 200 caractères pour cette description.

KEYWORDS : Mettre dans cette balise tous les mots-clés ou expressions-clés de la page. Ils devront être séparés d'une virgule. Cette balise à une importance très relative. Une liste de 15 à 20 mots sera le maximum conseillé.

ROBOTS : La balise meta robots sert à « communiquer » avec les robots des moteurs de recherches. Ces derniers ont des robots (petits programmes) qui parcourent le web et enregistrent les pages web des sites. Ils vont de site en site en suivant les liens. Dans cette balise `<robots>` vous pourrez justement indiquer à ces robots si vous souhaitez qu'ils enregistrent ou non votre page, s'ils doivent ou non suivre les liens de la page. A noter qu'il est possible de définir individuellement chaque lien et son suivi grâce à l'attribut « NoFollow » de la balise de lien `<a>` dont il sera question un peu plus loin.

```

1 <head>
2
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4
5 <meta name="title" content="Titre de la page" />
6 <meta name="description" content="description de la page"
7 <meta name="keywords" content="mots-clé1, mots-clé2, ..."
8
9 <!-- Indexer et suivre -->
10 <meta name="robots" content="index,follow" />
11
12 </head>

```

Ajouter le CSS

A l'intérieur des balise `<head>`, il est possible d'ajouter différents styles CSS. Le CSS est un langage autre que le HTML5. Il permet de mettre en page le document.

Pour relier une Feuille de Style – CSS pour Cascading Style Sheet, il faut placer la balise `<link>` entre les balises `<head>`. Donnée à cette balise les attributs correspondant et leurs informations relatives. Dans l'exemple ci-dessous, **Link** est la balise et, **Rel**, **href** et **type** sont des attributs.

```

1 <head>
2
3 <link rel='stylesheet' href='votre-fichier.css' type='text/css' />
4
5 </head>

```

Relier une feuille de Style – CSS – indépendante est plus pratique, car si le site possède plusieurs pages WEB, une seule ligne de code suffit pour l'insérer dans chacune. Cela facilite les modifications ultérieures. Il est cependant possible d'écrire du code CSS sur une page html. Ce code ne sera donc valable que pour cette page. Il s'écrit de la manière suivante :

```

1 <head>
2
3 <style type="text/css" media="screen">
4 p { color:red; }
5 </style>
6
7 </head>

```

Pour rappel, le HTML5 fonctionne par balises imbriquées. Entre les balises `<p>` et `</p>` se trouve un paragraphe. Dans l'exemple ci-dessous, le style sera appliqué à tous les paragraphes de la page. Ces paragraphes s'écriront donc de la couleur rouge.

Relier les Scripts

Entre les balises `<head>`, il est possible également d'insérer des scripts.

Comme pour le CSS, la manipulation est quasi identique pour les scripts. Il est permis de, soit relier un script indépendant, soit d'écrire sur votre page html. Exemple :

```
1 | <script type="text/javascript" src="votre-script.js"></script>
```

Le JavaScript est un langage de script, comme son nom l'indique, qui agit avec le navigateur. Par exemple, ouvrir une petite fenêtre d'alerte qui demande de confirmer le choix lors d'un clic sur un lien.

Il faut penser à toujours refermer la balise `<script>` même si elle ne contient rien en apparence. Contrairement au CSS, les scripts ne s'incorporent pas toujours entre les balises `<head>`. Tout dépend de leur fonctionnement relatif au DOM (comprendre ceci par chargement et exécution de la page).

```
1 | <!-- Exemple d'un script JavaScript incorporer -->
2 | <script type="text/javascript">
3 |     alert("Voici un message d\alerte!");
4 | </script>
5 | <noscript>La balise SCRIPT s'accompagne souvent
6 | de sa balise inverse : NOSCRIPT. Cette dernière contient
7 | un message à afficher si le script n'est pas supporté.
8 | Vous n'est pas obligé de mettre cette balise.
9 | </noscript>
```

D. CORPS D'UN DOCUMENT HTML5

Après avoir détaillé l'en-tête situé entre les balise `<head>` et `</head>`, il faut lancer le corps de la page web, le gros du travail.

Représentée entre les balises `<body>` et `</body>`, il faut placer **tout le contenu de la page web à l'intérieur** de celle-ci. Contrairement à l'en-tête, le corps du document s'affichera sur le navigateur.

Cette partie du cours enseigne à placer des textes structurés en paragraphe, avec des titres, des images, de la musique et de la vidéo. Il sera également question d'architecture spécifique au HTML5 avec ses nouvelles balises telles que ASIDE, SECTION, ARTICLE, NAV, etc.

Etudions le code source suivant :

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5
6 <!-- En-tête du document -->
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8
9 <!-- Balise meta -->
10 <meta name="title" content="Titre de la page" />
11 <meta name="description" content="description de la page"
12 <meta name="keywords" content="mots-clé1, mots-clé2, ..."
13
14 <!-- Indexer et suivre -->
15 <meta name="robots" content="index, follow" />
16
17 <!-- Relier une feuille CSS externe -->
18 <link rel='stylesheet' href='votre-fichier.css' type='text/css' />
19
20 <!-- Incorporez du CSS dans la page -->
21 <style type="text/css" media="screen">
22 p { color:red; }
23 </style>
24
25 <!-- Relier un fichier JavaScript -->
26 <script type="text/javascript" src="votre-script.js"></script>
27
28 <!-- Incorporez du JavaScript dans la page -->
29 <script type="text/javascript">
30 alert("Voici un message d\'alerte!");
31 </script>
32 <noscript>La balise SCRIPT s\'accompagne souvent
33 de sa balise inverse : NOSCRIPT. Cette dernière contient
34 un message à afficher si le script n\'est pas supporté.
35 Vous n\'est pas obligé de mettre cette balise.
36 </noscript>
37
38 </head>
39
40
41 <body>
42
43 <!-- CORPS DE LA PAGE -->
44 <p>Bienvenue sur ma page web</p>
45
46 </body>
47 </html>
```

E. LEXIQUE CSS3

En complément des termes HTML5, il y a quelques termes courant en CSS3 qu'il faut connaître.

Les sélecteurs

Un sélecteur détermine exactement à quel(s) élément(s) les styles correspondants seront appliqués. Les sélecteurs peuvent inclure une combinaison de différents identifiants : les classes, les types ...

Les sélecteurs sont situés avant la première accolade, {

```
1 | p { ... }
```

Dans l'exemple ci-dessus, nous appliqueront un style aux éléments *p*. Ce style, qui prend la place des trois points, s'appliquera à tous les paragraphes, ou éléments *p*, de la page web.

Les Propriétés

Une propriété détermine le style qui sera appliquée à un élément. Il y a un très grand nombre de propriétés possibles et de nouvelles sont constamment ajoutées.

```
1 | p {  
2 |     font-family: Arial;  
3 | }
```

Dans l'exemple ci-dessus, la propriété `font-family`.

Les valeurs

C'est la valeur qui détermine le comportement d'une propriété. Les valeurs sont identifiées comme le texte après les deux points de la propriété jusqu'au point-virgule. Ce dernier sert à indiquer la fin de la ligne de commande suivant le schéma suivant :

```
1 | p {  
2 |     propriété : valeur ;  
3 | }
```

Les espaces entre les deux-points (:) et le point-virgule (;) ne sont pas obligatoires. Il est possible de mettre des espaces si cela améliore la lisibilité, cela n'affectera en rien le résultat.

Dans l'exemple ci-dessous, la valeur Arial est donnée à la propriété font-family. Soit la police de caractères Arial à l'élément *p*.

```
1 | p {  
2 |     font-family: Arial;  
3 | }
```

Le CSS3 fonctionne en utilisant des sélecteurs qui permettent d'appliquer des styles aux éléments HTML5. Tous les styles CSS3 permettent à plusieurs éléments **d'hériter** de différent style.

Un exemple, il est possible de définir un style pour l'ensemble du texte sur une page afin qu'il soit d'une couleur spécifique :

```
1 | p {  
2 |     color : red;  
3 | }
```

L'ensemble de texte de notre document HTML5 appartenant à un élément *p* sera affiché en rouge.

```
1 | <p>Ce texte sera affiché en <strong>rouge</strong> </p>  
2 | <p>Ce deuxième paragraphe sera aussi en rouge</p>
```

Le texte ci-dessus sera affiché en rouge, même l'élément STRONG (le texte entre les balises ** et **). Pour appliquer une couleur différente à l'élément STRONG ou à un seul paragraphe (élément *p*), nous allons avoir recours aux sélecteurs.

F. LES SÉLECTEURS

Les sélecteurs, comme mentionné précédemment, servent à déterminer le style des éléments. Dans l'immédiat, il est important de bien comprendre comment les utiliser sans pour autant tous les connaître. Les sélecteurs les plus utilisés comprennent des éléments, des identifiants et des classes, ou une combinaison des trois.

Les sélecteurs de type

Les sélecteurs de type sont les sélecteurs les plus élémentaires. Privilégiez les sélecteurs de type lorsque cela est possible car ils nécessitent moins de code et sont faciles à gérer. Le style sera appliqué à l'ensemble des éléments du même type :

```
1 | p {  
2 |     color : red;  
3 | }  
  
1 | <p>Ce texte sera affiché en rouge</p>  
2 | <p>Ce deuxième paragraphe sera aussi en rouge</p>
```

Les sélecteurs de classe

Les sélecteurs de classe permettent d'appliquer le même style à un groupe d'éléments en leur donnant à tous le même attribut de class. En CSS3, Les classes sont notées suivant le schéma ci-dessous :

Franck CELLIER, Formateur en informatique - Tout droit réservé - 2019

2 rue Gabriel FAURÉ - 59150 WATTRELOS - fcellier34@gmail.com - ☎ 06 41 12 70 19 - www.linkedin.com/in/formateur-franck-cellier


```
1 | .class { ... }
```

Un point (.) précède le nom de la classe. Il est permis d'utiliser l'attribut de classe même sur de multiples éléments sur une page.

```
1 | .rouge {  
2 |     color : red;  
3 | }  
  
1 | <p class="rouge">Ce texte sera affiché en rouge</p>  
2 | <p>Ce deuxième paragraphe ne le sera PAS</p>
```

Les sélecteurs d'ID (identifiant)

Les sélecteurs d'ID sont semblables à des sélecteurs de class mais sont utilisés pour cibler un élément UNIQUE sur le document. En CSS3, Les classes sont notées suivant le schéma ci-dessous :

```
1 | #identifiant { ... }
```

Un symbole dièse (#) précède le nom de l'identifiant. Les ID sont uniquement utilisés une fois par page et devrait idéalement être réservés à des éléments significatifs.

```
1 | #rouge {  
2 |     color : red;  
3 | }  
  
1 | <p id="rouge">Ce texte sera affiché en rouge</p>  
2 | <p>Ce deuxième paragraphe ne le sera PAS</p>
```

Combiner plusieurs sélecteurs

La performance du CSS3 provient de sa capacité à combiner des sélecteurs et d'hériter des styles. Cela permet de commencer avec un sélecteur générique et de continuer à travailler de façon plus spécifique. En outre, vous il est possible de combiner des sélecteurs différents pour être le plus précis possible :

```
1 <p id="rouge">Le mot en gras est en <strong>ROUGE</strong> </p>.
2 <p>Le mot en gras est en <strong class="vert">Vert</strong> </p>.
3 <p>Le mot en gras est en <strong>Bleu</strong> </p>.

1 strong {
2     color : blue ;
3 }
4 #rouge strong {
5     color : red ;
6 }
7 strong.vert {
8     color : green ;
9 }
```

Sélecteurs supplémentaires

Les sélecteurs peuvent être extrêmement puissants et les sélecteurs décrits ci-dessus n'en sont que le début. Beaucoup de sélecteurs plus avancés existent et sont facilement disponibles. Il convient également de mentionner que tous les sélecteurs ne sont pas compatibles avec tous les navigateurs, en particulier avec les nouveaux sélecteurs introduit en CSS3.

G. LES COMMENTAIRES EN HTML5 ET CSS3

Le HTML5 et le CSS3 donnent la possibilité de commenter le code. Ces commentaires peuvent être utilisés pour aider à gérer plus efficacement le code. Les commentaires sont particulièrement utiles quand plusieurs personnes travaillent sur le même projet. Tout contenu enveloppé dans les commentaires ne sera pas affiché sur la page, ni interprété par les navigateurs.

Commentaires en HTML5

```
1 <!-- Ceci est un commentaire en HTML5,
2 il peut tenir su plusieurs lignes,
3 et ne sera ni affiché, ni interprété
4 par le navigateur -->
5 <p>Ceci nest plus un commentaire</p>
```

Commentaires en CSS3

```
1 /* Ceci est un commentaire en CSS3
2 il peut tenir su plusieurs lignes,
3 et ne sera ni affiché, ni interprété
4 par le navigateur */
5 p { ... }
```

Les commentaires en **HTML5** sont encadré par les symboles <!-- et -->

Les commentaires en **CSS3** sont encadré par les symboles /* et */