Formation HTML / CSS CNAM 2019

Formateur J.VIDAL

Sommaire

Table des matières

CHAPITRE 1 : Introduction html / versions

Court historique du HTML

Avant Internet

À la fin des années 1960, le **Pentagone** a demandé à son agence de recherche et développement, **DARPA**, de concevoir un réseau **invulnérable** pour les forces armées américaines. Le mot "invulnérable" est peut-être un grand mot car la demande était plutôt de concevoir *un réseau décentralisé qui se dégraderait avec élégance si ses noeuds étaient attaqués et détruits: il ralentirait mais ne s'arrêterait pas*.

Sa première démonstration officielle remonte à 1972 mais sa gestation s'est faite sous le nom d'**Arpanet** qui permettait aux agences gouvernementales et aux universités de communiquer entre-elles. Arpanet a servi de banc d'essai à plusieurs technologies et a donné naissance à l'**Internet** en 1992 en devenant **public**. Cependant, la contribution la plus importante de DARPA a été l'invention d'une série de protocoles de mise en ligne des pages reliées entre elles par des hyperliens: la suite **TCP/IP** qui est l'ensemble des protocoles utilisés pour le transfert des données sur Internet. TCP s'occupe du transport des données alors que IP s'occupe des adresses réseaux.

Sur ce site, je vais considérer seulement une partie de l'Internet: le **World Wide Web**, une des multiples applications de l'Internet qui comprennent aussi le *courrier électronique*, la *messagerie instantanée* et *Usenet*.

Le Web

Il existe une grande confusion entre l'Internet et le Web. Le Web a été mis au point plusieurs années après l'Internet et il s'agit seulement d'une des applications de l'Internet qui permet de naviguer entre des documents reliés entre eux par des **liens hypertextes** via un protocole de transmission de données appelé **HTTP** . Il met en jeu deux participants:

- un **serveur** qui produit les pages Web; et
- un **client** qui examine ces pages grâce à un logiciel local appelé **fureteur**.

Un site web est habituellement construit autour d'une page centrale, appelée « **page d'accueil** » et proposant des liens vers un ensemble d'autres pages hébergées sur le même serveur.

À la base de la programmation Web, on retrouve **HTML** (**H**yper**t**ext **M**arkup **L**anguage), un *langage de balisage* conçu pour afficher des pages Web. Sorti vers 1993 et alors appelé **SGML** (quoique les fichiers de données affichaient le suffixe .html). Cet HTML primitif contient déjà divers éléments comme le titre du document, les hyperliens, la structuration du texte en titres, sous-titres, listes ou texte brut. Tout est prêt mais il manque quelque chose: des instruments qui permettraient au usagers de voir et consulter les pages - il fallait des **fureteurs** (en anglais: **browsers**): *logiciels clients qui, à partir du code HTML qui leurs sont présentés gênèrent un affichage sur un écran*.

Vers la même époque, **NCSA Mosaic** remplit le vide, c'est le **premier fureteur**: il affiche déjà le texte et les images. **Netscape** apparaît en 1994 et il affiche des cadres (frames). **Internet Explorer,** mis sur le marché un peu plus tard,lui aussi affiche le texte, les images et les cadres. L'état de HTML correspond alors à ce que l'on pourrait appeler **HTML 1.0**. Il n'existe cependant aucune spécification portant ce nom, notamment parce que le langage est alors en pleine évolution. En 1995, **Netscape Navigator** ajoute le support de nombreux éléments de présentation: attributs de texte, clignotement, centrage, etc. Déjà, à ce moment, les développeurs ressentent un besoin pour une façon de styliser l'affichage à l'extérieur du HTML.

Standardisation du Web

Il y a alors un besoin de standardisation car l'évolution technologique est divergente. Heureusement, il y a eu la **W3C** (**World Wide Web Consortium**), un organisme de normalisation sans but lucratif fondé en octobre 1994 ayant pour mission de promouvoir des normes ouvertes et d'assurer la compatibilité des technologies du Web.

En début de mandat, le *W3C* travaille sur le développement d'un modèle d'objets de document qui a rapidement évolué vers une standardisation des fureteurs. il propose alors **HTML+**, un brouillon de **HTML 3.0** avec support des tables, des figures et des expressions mathématiques. Au début de 1997, le W3C publie la norme **HTML 3.2** qui décrit la pratique courante en 1996, un ramassis de ce que les fureteurs du moment peuvent faire. Il contient des éléments en prévision du support des styles et des scripts.

Le Web actuel

HTML 4.0 suit peu après à la fin de 1997 et apporte différentes améliorations pour l'accessibilité des contenus dont principalement la possibilité d'une séparation plus explicite entre structure et présentation du document. De plus, il définit trois variantes de documents qui sont définies en début de page par le DTD (Document type définition):

- **strict** - exclut des éléments et attributs dits « de présentation », destinés à être remplacés par les styles CSS, ainsi que les éléments applet et frame qui sont remplacés par l'élément "object" réputé plus apte à l'interopérabilité et à l'accessibilité;
- **transitionnelle** - étend la variante stricte en reprenant les éléments et attributs dépréciés de HTML 3.2, dont les éléments de présentation sont couramment utilisés par les éditeurs HTML de l'époque; et
- **frameset** - normalise la technique des jeux de cadres composant une ressource unique à partir de plusieurs pages web assemblées par le fureteur.

HTML 4.01, une révision de HTML 4.0, est entré en vigueur en 1999.

HTML 5

Aujourd'hui, *HTML 4.1 est* ***encore la norme*** quoique **HTML 5.0** soit actuellement dans sa phase finale de validation. Elle va remplacer la version 4.01 datant de 1999 et introduire de nouvelles balises, de nouvelles API et de nouveaux attributs qui vont étendre les capacités du langage HTML au delà du simple objectif d'afficher des informations.

Comme pour la version HTML 4.1, la version HTML 5 nécessite un DOCTYPE qui permet d'indiquer au navigateur la méthode de rendu à appliquer pour afficher le document. Celui-ci ne fait plus référence à un DTD (Document Type Definition) et se déclare comme ceci :

```
<!DOCTYPE html>
```

La détection du type d'encodage du document HTML se fait en vérifiant les éléments suivant dans l'ordre indiqué :

1. recherche d'une **balise meta** de type http-equiv="Content-Type" dans le document;
2. détection de la marque d'ordre des octets (BOM) en début de fichier indiquant l'encodage; et
3. recherche de la balise meta charset.

Remarque
Ce site opère déjà avec ce doctype suivi de
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
dans l'entête

Les langages auxiliaires

Avec l'évolution de HTML, d'autres langages se sont aussi formés. Afin d'améliorer la réactivité des pages Web, Netscape a introduit **Javascript** avec **Navigator 2.0** en 1996. **JScript** de Microsoft a suivi peu après. Ces langages qui s'exécutent sur l'ordinateur client, permettent aux développeurs de donner aux pages une certaine interactivité. Le support pour **DHTML** et **CSS** fut ajouté en 1997.

Les feuilles de style en cascades sont le résultat d'une demande quasi-universelle de la part des développeurs d'une façon de séparer le *contenu* de la *présentation*. Elles servent à décrire la présentation des documents HTML et XML. Elles deviennent couramment utilisées dans la programmation Web au début des années 2000.

PHP et ASC s'exécutent sur le serveur. Ils permettent au développeur d'injecter dans une page Web non seulement les balises HTML mais aussi tous les codes *Javascript* (ou *JScript*) nécessaire à son interactivité:

- **PHP** fut créé en 1994. Ce langage exécuté sur le serveur sert principalement à produire des pages Web dynamiques en injectant dans la page des codes Javascript en plus des balises HTML. Tout fichier avec l'extension .php est en effet un fichier où les commandes PHP seront interprétées.
- **ASP** est un ensemble de logiciels développés par Microsoft et utilisés dans la programmation Web. Elle nécessite pour fonctionner une plate-forme **Windows** avec **IIS** installé, ou encore une plate-forme **Linux** ou **Unix** avec une version modifiée d'**Apache**. ASP est une structure composée d'objets

accessibles par deux langages principaux : le **VBScript** et le **JScrip**t [*Wikipedia*]. ***Je n'en sais pas plus sur ASP car je ne l'ai jamais utilisé***.

CHAPITRE 2 : Structure d'un document

Les bases d'un document HTML

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  ...
  <!-- Le reste du contenu -->
  ...
</body>
</html>
```

Le Doctype

http://41mag.fr/liste-des-balises-html5/balise-doctype-html5

Les balises html, head, body

Les commentaires

- HTML

Les commentaires commencent par <!-- et se terminent par -->.

- CSS

mettez /* avant le LA PROPRIETE et */ après le point-virgule;

- PHP

```
<?php
   echo 'Ceci est un test'; // Ceci est un commentaire sur une seule ligne, style c++
   /* Ceci est un commentaire sur
      plusieurs lignes */
   echo 'Ceci est un autre test';
   echo 'Et un test final'; # Ceci est un commentaire style shell sur une seule ligne
?>
```

- JAVASCRIPT

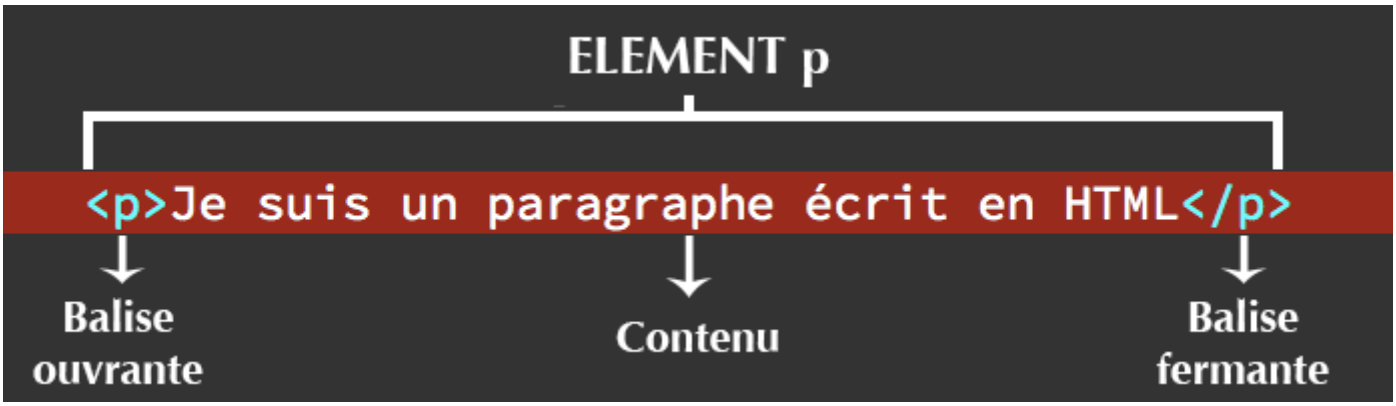
```
alert("message"); // Commentaire expliquant le rôle de l'instruction
/* Commentaire sur une ligne */
/* Commentaire
sur plusieurs
lignes */
alert("message"); // Cette ligne est exécutée
```

Structures des balises et des attributs

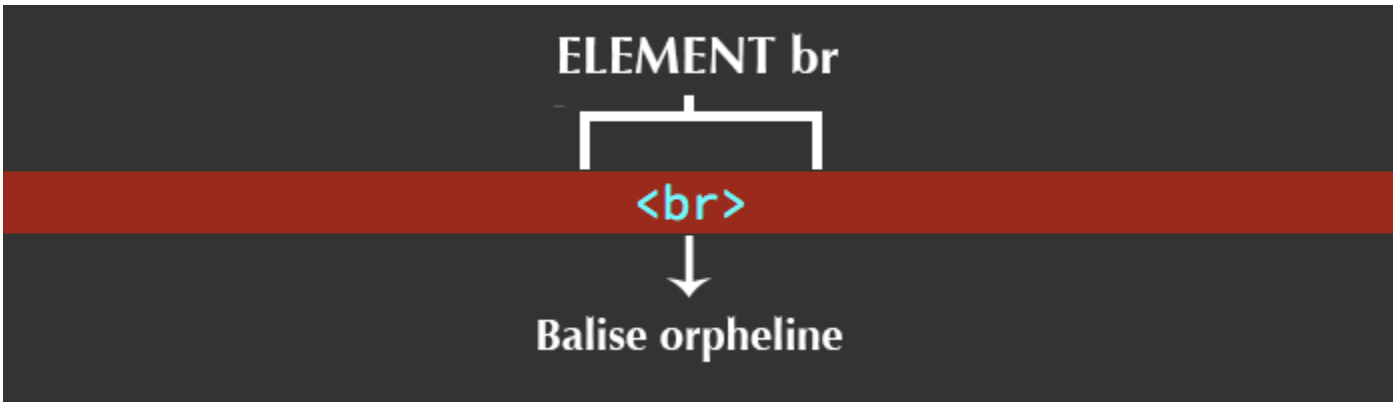
Les balises en HTML

Un élément HTML peut être soit constitué d'une **paire de balises et d'un contenu**, soit (plus rarement) d'une **balise unique qu'on dit alors orpheline**.

L'élément `p` ci-dessous est constitué d'une balise ouvrante, d'un balise fermante (notez la présence du slash), et d'un contenu (textuel) entre les balises.



L'élément `br` ci-dessous (servant à créer un retour à la ligne) n'est lui constitué que d'une balise orpheline.



Sur le web, vous trouverez peut être des éléments `br` écrits avec un slash après le nom de l'élément, comme ceci :
. Les deux syntaxes sont acceptées en HTML, la seule différence est que la syntaxe utilisant le slash est également reconnue par le langage XML.

Liste des balises HTML 5

https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1608357-memento-des-balises-html

Les attributs en HTML

Finalement, **la balise ouvrante d'un élément HTML peut contenir des attributs**, qui sont parfois même obligatoires.

Les attributs vont venir compléter les éléments en les définissant plus précisément ou en leur apportant des informations supplémentaires.

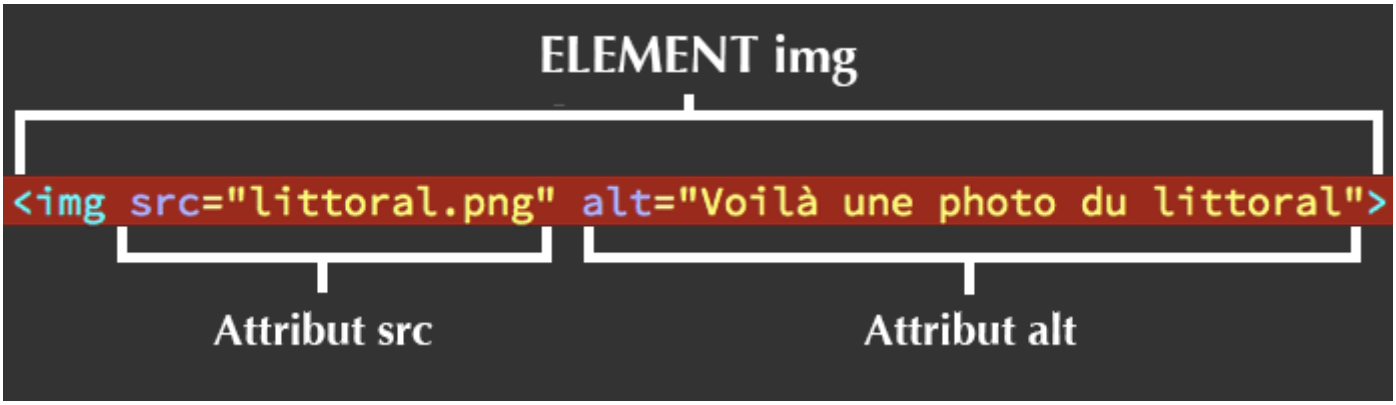
Un attribut contient toujours une valeur, qu'on peut cependant parfois omettre dans le cas des attributs ne possédant qu'une seule valeur (la valeur est considérée comme évidente).

Par exemple, l'élément `a` (pour "anchor") servant à créer des liens vers d'autres sites ou d'autres pages, va avoir besoin d'un attribut `href` ("hypertexte reference") qui va prendre comme valeur l'adresse (relative ou absolue) de la page vers laquelle on souhaite faire un lien.

L'élément `img`, servant à insérer une image dans une page HTML, va lui demander deux attributs : `src` et `alt`.

L'attribut `src` va prendre comme valeur le nom et l'emplacement de l'image tandis que l'attribut `alt` va afficher un texte alternatif dans le cas où l'image ne serait pas disponible (pour les non voyants par exemple).

Notez que cet élément n'est constitué que d'une seule balise orpheline, tout comme l'élément `br`.



Notez bien que les balises et les attributs ne seront jamais affichés par le navigateur : ils vont servir à indiquer au navigateur comment il doit traiter chaque contenu. Bien construire sa page web sera aussi hautement bénéfique pour votre référencement dans les moteurs de recherche.

Liste des attributs

https://developer.mozilla.org/fr/docs/Web/HTML/Attributs

Différences entre les balises inline et block

3 modes d'affichage pour les éléments HTML

Par défaut, les éléments HTML sont affichés selon l'un des modes suivants :

1- Bloc / block

 Occupe toute la largeur disponible. Lorsque 2 éléments blocs se suivent dans une page, ils sont positionnés (par défaut) l'un **sous** l'autre.

Exemple typique d'élément bloc : l'élément (le paragraphe).

2- En-ligne / inline

 N'occupe que la largeur indispensable à l'affichage du contenu et ne provoque pas de retour à la ligne. Lorsque 2 éléments en-ligne se suivent dans une page, ils sont positionnés l'un **à côté** l'autre (si la largeur de page le permet).

Exemple typique d'élément en-ligne : l'élément <img> (image).

3- Invisible / not displayed

 Certains éléments ne servent qu'à apporter des informations invisibles pour l'internaute.

Exemple typique d'élément non affiché : l'élément <meta>.

Exemples d'éléments Block et inline-block

https://www.alsacreations.com/tuto/lire/530-La-structure-des-balises-bloc-et-en-ligne.html

CHAPITRE 3 : Les balises de contenu

Les titres (h1, h2, h3, etc)

<h1>Titre de niveau 1</h1>

<h2>Titre de niveau 2</h2>

<h3>Titre de niveau 3</h3>

<h4>Titre de niveau 4</h4>

<h5>Titre de niveau 5</h5>

<h6>Titre de niveau 6</h6>

Les paragraphes

L'élément HTML ** contenu ** représente un paragraphe de texte. Les paragraphes sont généralement représentés comme des blocs et séparés par un espace vertical, leur première ligne est également parfois indentée. Les paragraphes sont des éléments blocs.

Mise en gras et en italique

Cette rubrique va vous montrer comment mettre un texte en gras, en italique et le souligner.

Pour cela, il vous suffit d'encadrer le texte concerné par les balises ****, **** et/ou **<u></u>**.

https://www.alsacreations.com/article/lire/552-strong-b-em-i-quelle-balise-utiliser-et-pourquoi.html

Les liens hypertexte

Liens vers un site (liens absolue)

```
<a href="https://openclassrooms.com">OpenClassrooms</a>
```

```
<p>Bonjour. Vous souhaitez visiter <a href="https://openclassrooms.com">OpenClassrooms</a> ?<br />
C'est un bon site ;o)</p>
```

Liens vers une page (Liens relatif)(valable uniquement si la page2.html se trouve dans le même dossier que page1.html)

```
<p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a> ?</p>
```

Lien relatif vers une page qui est rangée dans un sous-dossier « contenu »

```
<a href="contenu/page2.html">
```

Exemple avec plusieurs sous-dossiers

```
<a href="contenu/autredossier/page2.html">
```

La page 2 se trouve dans un dossier différents de la page 1 (../ on remonte d'un cran dans l'architecture)

```
<a href="../page2.html">
```

Autre lien : Les ancres nommées

Ajouter l'attribut « id » à la balise <h2>

```
<h2 id="mon_ancre">Titre</h2>
```

Puis créer l'ancre

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

ATTENTION il faut beaucoup de texte pour que cela fonctionne car les ancres fonctionnent en fonction des tailles d'écran

Pointer une ancre sur une autre page

```
<a href="ancres.html#rollers">
```

L'attribut « Title » (Une info bulle au-dessus du lien)

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com" title="Vous ne le regretterez pas !">OpenClassrooms</a> ?</p>
```

Un lien qui ouvre une nouvelle fenêtre

```
<p>Bonjour. Souhaitez-vous visiter <a href="https://openclassrooms.com" title="Vous ne le regretterez pas !" target="_blank">OpenClassrooms</a> ?</p>
```

Un lien « email »

```
<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>
```

Un lien pour télécharger un fichier

```
<p><a href="monfichier.zip">Télécharger le fichier</a></p>
```

Les différents types de liens

https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1604646-creez-des-liens

Les listes (ordonnées et non ordonnées)

Liste non ordonnées (puce)

```
<ul>
  <li>2 avocats (pelés et avec les noyaux retirés)</li>
  <li>le jus d'un citron</li>
  <li>¼ de concombre, coupé grossièrement</li>
  <li>1 petite tomate, coupée</li>
</ul>
```

Liste ordonnées (chiffre par défaut)

```
<ol>
  <li>2 avocats (pelés et avec les noyaux retirés)</li>
  <li>le jus d'un citron</li>
  <li>¼ de concombre, coupé grossièrement</li>
  <li>1 petite tomate, coupée</li>
</ol>
```

Plus d'infos sur les listes

https://developer.mozilla.org/fr/docs/Apprendre/HTML/Comment/Cr%C3%A9er_une_liste_d_%C3%A9l%C3%A9ments_avec_HTML

Les tableaux

Les balises de base :

<table> l'ensemble du tableau

<tr> la ligne du tableau

<td> la cellule du tableau

<th> le titre du tableau

Les tableaux en détails

https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1606851-ajoutez-des-tableaux

Chapitre 4 : Les Médias

Les images et format d'image

PNG	Portable Network Graphics
GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
SVG	Scalable Vector Graphics

ICO	FAVICON

Les attributs essentiels (src, alt, etc)

<img class="fit-picture"

src="/media/examples/grapefruit-slice-332-332.jpg"

alt="Grapefruit slice atop a pile of other slices" width="100%" height ="auto" />

Intégrer des sons et vidéos

Les sons

https://www.pierre-giraud.com/html-css/cours-complet/insertion-audio-html.php

Les vidéos

https://www.pierre-giraud.com/html-css/cours-complet/insertion-video-html.php

Le cas de Youtube/vimeo etc

https://www.pierre-giraud.com/html-css/cours-complet/integration-semantique-html.php

Chapitre 5 : Les Formulaires

http://formation.upyupy.fr/html-xhtml/formulaires-html/

Les balises de formulaire
Les attributs action et method
Les différents champs de formulaire
Les différentes formes du champs input
Les attributs pour les champs de formulaire (placeholder, required, etc)
Valider un formulaire

Chapitre 6 : Le CSS

Présentation et définition

CSS (Cascading Style Sheets) : Une feuille de style CSS est un langage informatique (langage CSS) qui décrit la présentation des documents HTML, XHTML et XML. Les standards définissant le code CSS sont publiés par le World Wide Web Consortium (W3C). L'utilisation du CSS est indispensable pour le developpement web (front end) afin de rendre le site esthétique et responsive design. Dernière version CSS : CSS3

3 manières d'appliquer un style css :

Style html

<table border="1" bgcolor="silver" cellpadding="3" cellspacing="0">

Css dans le corps html

 Cette balise div a du style !

Css dans l'entête de la page

```
<head>
<style type="text/css">
  div {
    background-color:#339;
    color:#fff;
    padding:15px;
    border-bottom:5px solid red;
    margin-bottom:15px;
      }
</style>
</head>

<body>
<div>
Cette phrase est présentée en fonction du style défini dans l'en-tête
</div>
<div>
Cette phrase aussi, et pourtant le style n'a été défini qu'une fois !
</div>
</body>
```

Attacher une css à une page html

La façon idéale de définir les CSS consiste à les enregistrer dans un document indépendant de vos pages HTML. Grâce à cette méthode, toutes les pages qui font référence à cette feuille de style externe hériteront de toutes ses définitions.

Cette méthode permet également de définir plusieurs feuilles de styles pour le même contenu et de basculer d'une feuille à l'autre en fonction du support sur lequel le contenu est affiché (écran, imprimante, etc.). Nous reviendrons plus tard sur cet aspect.

Une page HTML peut faire référence à plusieurs feuilles de styles en même temps. Dans ce cas, les définitions contenues dans ces différentes feuilles seront combinées entre elles. Nous examinerons ce point dans notre chapitre Héritages et cascades.

Voici un exemple de styles définis dans un document séparé :

Document "mes-styles.css"

```
body {
  background-color:#ccf;
  letter-spacing:.1em;
}
p {
  font-style:italic;
  font-family:times,serif;
}
```

Document "ma-page.html"

```
<head>
<link href="mes-styles.css" media="all" rel="stylesheet" type="text/css" />
</head>

<body>
<p>Voici un exemple de paragraphe.</p>
<p>Et voici un deuxième paragraphe.</p>
</body>
```

Les selecteurs (balise, id, class)

Comment utiliser des classes pour appliquer un style

Vous pouvez attribuer à chaque élément HTML une ou plusieurs classes. C'est vous qui définirez le nom de ces classes et qui déciderez de leurs styles.

Les styles définis dans les classes remplaceront les styles "normaux" des éléments auxquels ils s'appliquent.

Pour créer une classe, vous devez simplement faire figurer son nom précédé d'un point. Pour éviter toute ambiguïté, votre nom de classe ne doit pas comporter d'espace.

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
.mon-style {
  color:red;
}
```

Pour appliquer le style défini dans votre classe à un élément, ajouter la mention class="nom-du style" dans la définition de la balise :

```
<p class="mon-style">Le style s'applique à ce paragraphe </p>

<p>Mais pas à celui-là</p>
```

Cette façon de procéder est très pratique car elle permet d'appliquer les réglages de votre classes et de nombreux éléments, même s'ils ne sont pas du même type :

```
<p class="mon-style">Le style peut s'appliquer à ce paragraphe </p>

<div class="mon-style">Et aussi à cette balise ! </div>
```

Les éléments HTML peuvent avoir plusieurs classes

Chaque élément HTML peut avoir aucune, une ou plusieurs classes. Pour appliquer plusieurs classes au même élément, précisez simplement la liste de classes en séparant leurs noms par un espace :

Cette définition de styles est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
.mon-style1 {
  color:yellow;
}
.mon-style2 {
  background-color:#A0A0A0;
  font-weight:bold;
}

<p class="mon-style1 mon-style2">Les styles des deux classes s'appliquent à ce paragraphe</p>

<p class="mon-style2">Alors que ce paragraphe n'a qu'une seule classe </p>
```

Comment utiliser des "ID" pour appliquer un style

Les éléments HTML peuvent se voir attribuer un "ID" (identification) en plus ou à la place d'une classe.

Le principe de l'ID est très similaire à celui de la classe à une exception prés :

- Plusieurs éléments peuvent avoir la même classe.
- Il ne doit y avoir qu'un seul élément ayant un ID donné.

Vous pourriez parfaitement vous contenter d'utiliser les classes pour tous vos styles et oublier complètement l'existence des ID. Leur utilisation permet simplement de clarifier les choses et de mieux structurer vos pages :

- On utilise les classes pour définir l'aspect des mots, phrases et paragraphes.
- On utilise les ID pour placer des blocs dans la page (sachant qu'on a généralement jamais deux blocs placés exactement au même endroit).

Pour créer un ID, vous devez simplement faire figurer son nom précédé d'un dièse **#**. Pour éviter toute ambiguïté, votre nom d'ID ne doit pas comporter d'espace.

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
#mon-style {
  color:red;
}
```

Pour appliquer le style défini dans votre ID à un élément, ajouter la mention ID="nom-du style" dans la définition de la balise :

```
<p ID="mon-style">Le style s'applique à ce paragraphe </p>

<p>Mais pas à celui-là</p>
```

http://www.css-faciles.com/premiers-pas-css.php

Les pseudo-sélecteurs

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

Les pseudos classes

https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes

Les propriétés et valeurs

https://lesdocs.fr/principales-proprietes-et-valeurs-css/

Chapitre 7 : Les propriétés CSS

Les propriétés de texte (police, taille, alignement)

https://www.pierre-giraud.com/html-css/cours-complet/proprietes-css-text.php

Les couleurs hexadecimale, rgb, rgba et nommées

http://www.css-faciles.com/couleurs-css.php

Les propriétés d'arrière plan

https://developer.mozilla.org/fr/docs/Web/CSS/background

Les marges intérieurs et extérieur (margin, padding)

https://www.pierre-giraud.com/html-css/cours-complet/marges-css.php

Les bordures

https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1605694-creez-des-bordures-et-des-ombres

Les dimensions

https://www.pierre-giraud.com/html-css/cours-complet/dimensions-css.php

Les listes

https://www.alsacreations.com/article/lire/1210-donnez-du-style-a-vos-listes.html

Chapitre 8 : Le positionnement de bloc

Positionnement absolu ou relatif

https://www.pierre-giraud.com/html-css/cours-complet/css-position.php

Bloc flottant et fixe

https://www.pierre-giraud.com/html-css/cours-complet/float-clear-css.php

Les blocs flexibles

https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html

Les grilles

https://www.alsacreations.com/article/lire/1388-css3-grid-layout.html

Chapitre 9 : Les animations

https://www.pierre-giraud.com/html-css/cours-complet/animations-css.php

Les animations et transitions
Durée des animations
Les fonctions de transition
Les Keyframes

Chapitre 10 : Autres propriétés CSS3

Les transformations

https://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html

Les ombres de texte et de bloc

https://www.alsacreations.com/tuto/lire/910-creer-des-ombrages-ombres-css-box-shadow-text-shadow.html

Les bordures arrondies

https://www.alsacreations.com/tuto/lire/891-coins-arrondis-css-sans-images.html

Les colonnes

https://www.alsacreations.com/tuto/lire/1557-les-multicolonnes-en-css3.html

La propriété box-sizing

https://developer.mozilla.org/fr/docs/Web/CSS/box-sizing

Chapitre 11 : Construire un site web adaptatif

Responsive Web Design (RWD) : définition, composantes
https://www.pierre-giraud.com/html-css/cours-complet/introduction-responsive-design.php

Viewport : notion, meta, valeurs
https://www.pierre-giraud.com/html-css/cours-complet/viewport-html.php

https://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html

Unités relatives (%, em) vs. absolues (px)
http://www.css-faciles.com/unites-css.php

Chapitre 12 : Créer une grille d'affichage flexible

Tailles d'écran, résolution optimale
https://www.alsacreations.com/tuto/lire/547-faire-un-site-pour-toutes-les-resolutions.html

Modes portrait et paysage
https://developer.mozilla.org/fr/docs/Web/CSS/@media/orientation

Tailles minimales, maximales
https://www.cssdebutant.com/coder-en-css-largeur-et-hauteur.html

Blocs, approche contenu/contenant
https://developer.mozilla.org/fr/docs/Web/CSS/A_Propos_Du_Bloc_Conteneur

Principe des box, layout avec CSS3
https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/3298561-faites-votre-mise-en-page-avec-flexbox

Taille des fonts : fixer une base pour les tailles, conversions em
https://www.alsacreations.com/article/lire/563-gerer-la-taille-du-texte-avec-les-em.html

Eviter les débordements
https://www.alsacreations.com/tuto/lire/1038-gerer-debordement-contenu-et-cesures-css.html

Chapitre 13 : Utiliser des médias flexibles

Images flexibles : images de fond, adaptation HTML5
https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html

Marges et espaces flexibles
https://www.cssdebutant.com/coder-en-css-les-padding-en-css.html

Vidéos adaptées
https://la-cascade.io/video-en-background/

Support des propriétés CSS par les anciens navigateurs

http://www.css-faciles.com/styles-alternatifs.php

Chapitre 14 : Ecrire des CSS3 Media Queries

Adaptation de l'affichage en fonction de la résolution

https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1607616-utilisez-le-responsive-design-avec-les-media-queries

Types de médias

https://developer.mozilla.org/fr/docs/Web/CSS/@media

Choix des règles conditionnelles : orientation, device-width

https://developer.mozilla.org/fr/docs/Web/API/CSS_Object_Model/Managing_screen_orientation

Medias queries internes, externes

https://www.pierre-giraud.com/html-css/cours-complet/media-queries-css-responsive.php

Gestion des menus et sliders

Exemples de menus dynamique html/css

https://www.pierre-giraud.com/html-css/exercices-html-css/creation-menu-responsive-deroulant-html-css.php

Exemples de sliders dynamique

https://www.creativejuiz.fr/blog/tutoriels/css3-creer-slideshow-automatique-controlable-transition

Chapitre 15 : Frameworks RWD

Panorama des frameworks existants

https://www.codeur.com/blog/front-end-framework/

Bootstrap

https://getbootstrap.com/

Vidéo tutorial

https://www.youtube.com/watch?v=gm2RCfjXS3s

960 grid

https://960.gs/

tutorial

https://code.tutsplus.com/tutorials/prototyping-with-the-grid-960-css-framework--net-1787

vidéo HTML

https://code.tutsplus.com/articles/a-detailed-look-at-the-960-css-framework--net-2567

Vidéo intégration PHOTOSHOP / 960.gs

https://www.youtube.com/watch?v=XxhhkvP1w_c

Optimisation des ressources graphiques

https://lehollandaisvolant.net/tuto/pagespd/