

HTML & CSS

INTRODUCTION :

Le HTML (HyperText Markup Language) et le CSS (Cascading StyleSheets) sont des langages informatiques conçus pour la création de sites web. Un site web est une collection de pages web hébergées sur un serveur et liées entre elles par des liens hypertextes, il est accessible via un navigateur.

Apprendre le HTML et le CSS signifie entrer dans le monde des programmeurs. Ces langages sont incontournables et universels ils se trouvent à la base de tous les sites web, les navigateurs (Google, safari...) aussi ont été conçus pour pouvoir lire du code HTML et maintenant ils ne peuvent lire que du code HTML, CSS et JavaScript. Ces deux langages n'ont aucun concurrent à l'heure actuelle, quel que soit le projet web à concevoir il faut forcément les utiliser.

Pour une explication plus détaillée de ces langages nous allons diviser ce document en deux grandes parties et chaque partie sera divisée en sous parties. Dans la première partie nous parlerons de HTML et dans la deuxième nous aborderons le CSS.

I. HTML

1. Historique

Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web.

Ce langage permet d'écrire de l'hypertexte (d'où son nom), de structurer sémantiquement une page web, de mettre en forme du contenu, de créer des formulaires de saisie ou encore d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques. L'HTML offre également la possibilité de créer des documents interopérables avec des équipements très variés et conformément aux exigences de l'accessibilité du web.

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). HTML est initialement dérivé d'un langage de description à balises : le SGML (Standard Generalized Markup Language). Il a été inventé par Tim Berners-Lee dans les années 1990 et n'a jamais cessé d'évoluer. Ainsi, on est passé du HTML version 1 en 1991 au HTML version 2 en 1994 au HTML3 et etc. jusqu'au HTML5 en novembre 2014.

HTML Version History

HTML Version	Year Of Release
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML 5	2012
XHTML5	2013

All-web-programing.Blogspot.Com

2. Les bases en html

Un fichier HTML est un fichier texte d'extension .html où l'on écrit ce qui doit être affiché sur la page. Pour l'utiliser il faut avoir un éditeur de texte (visual studio code, komodo, sublime text, notepad++...) pour écrire le code.

1) Afficher un fichier HTML dans le navigateur

Pour afficher un document en HTML5 dans le navigateur, la première chose à faire, une fois un nouveau fichier ouvert dans votre éditeur de texte, est d'en changer le type pour mettre votre document en « html ». Pour cela, deux solutions : soit vous trouvez l'onglet où vous pourrez définir le type de votre document, soit vous enregistrez sous votre document en lui donnant un nom du type : document.html. Cela aura pour effet de changer le type de document en un document HTML. Ensuite, pour visualiser votre fichier dans un navigateur, vous avez à nouveau deux choix. Soit votre éditeur possède une option vous proposant de pré-visualiser votre document dans le navigateur de votre choix (c'est le cas de Komodo), soit vous devrez enregistrer votre document (sur votre bureau, par exemple), puis double cliquer dessus, tout simplement, afin que la page s'ouvre dans votre navigateur par défaut. Pensez bien à enregistrer dans votre éditeur votre document à chaque fois que vous souhaitez l'ouvrir dans votre navigateur et à rafraîchir la page dans votre navigateur si votre document était déjà ouvert (ctrl+r ou cmd+r) sans quoi vous ne verrez aucun changement.

Le code de HTML se compose d'une série d'éléments, utilisés pour entourer, ou envelopper, les diverses parties du contenu pour les faire apparaître ou agir d'une certaine façon.

2) Structure de base d'un fichier HTML



Ci-dessus nous avons la structure de page d'un fichier HTML.

- En première ligne se trouve la balise `<!DOCTYPE html>` :

Au début de HTML, dans les années 1991-1992, les doctypes étaient utilisés pour faire référence à des ensembles de règles qu'on pouvait utiliser pour dire qu'un document était du HTML « valide » et détecter les erreurs de balisage. Cependant, ceux-ci ne sont plus utilisés aujourd'hui et sont juste présents pour s'assurer que la page puisse fonctionner y compris sur les anciens navigateurs. Pour le moment, c'est tout ce qu'il y a à savoir à propos des doctypes.

- sur la deuxième ligne nous avons la balise `<html></html>` :

Cet élément encadre tout le contenu de la page. Cet élément est parfois appelé l'élément racine.

- `<head></head>` est le troisième élément du fichier :

Il est utilisé comme un container pour toutes les choses qui font partie de la page HTML mais qui ne sont pas du contenu affiché. C'est dans cet élément qu'on mettra des mots-clés, une description de la page qui apparaîtra sur les moteurs de recherche, les liens vers les fichiers CSS à utiliser pour la mise en forme, les déclarations des jeux de caractères à utiliser et ainsi de suite.

C'est dans cet balise qu'on trouve les balise `<meta>` et `<title></title>` :

❖ `<meta charset="utf-8">` :

Cet élément définit le jeu de caractères qui devrait être utilisé pour le document et indique que c'est utf-8. utf-8 regroupe l'ensemble des caractères connus utilisés dans les différents langages humains. Généralement, utf-8 permet de gérer n'importe quel texte que vous pourriez utiliser sur la page. Il n'y a pas de raison de ne pas le définir, et il permet d'éviter certains problèmes plus tard.

❖ `<title></title>` :

L'élément `<title>` définit le titre de votre page. C'est ce titre qui apparaîtra sur l'onglet lorsque la page sera chargée. C'est également ce titre qui sera utilisé pour décrire la page lorsque vous la placez dans vos marques-pages.

● `<body></body>` :

C'est l'élément qui contient tout le contenu que vous souhaitez afficher pour qu'il soit vu par les visiteurs : cela peut être du texte, des images, des vidéos, des jeux, des pistes audio jouables, et ainsi de suite.

3) Les éléments en html

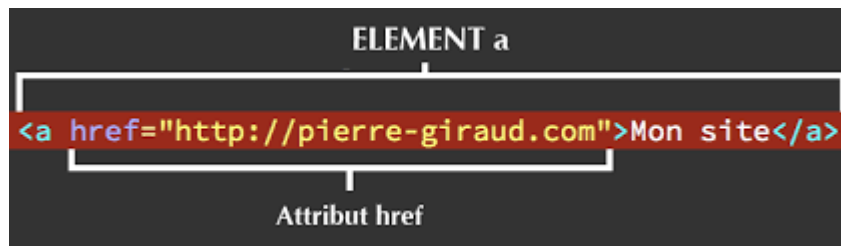
Un élément HTML est une suite de caractères textuels qui structure un document écrit en HTML, il peut être soit constitué d'une paire de balises (ouvrante et fermante) et d'un contenu, soit d'une balise unique qu'on dit alors « orpheline »:

- **Les balises :**

- Les balises « ouvrantes » indiquent le début de la portion de page web à délimiter. Elle se compose du nom de l'élément entre deux chevrons : `<exemple>`.
- Les balises « fermantes », qui en indiquent la fin. Ici on a également des chevrons et le nom de l'élément, auxquels on ajoute une barre oblique avant le nom de l'élément : `</exemple>`.
- Les balises « orpheline » servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

- **Le contenu :** C'est le contenu de l'élément. Ça peut être du texte.

Les éléments peuvent aussi avoir des « attributs », ce qui ressemble à :



Les attributs contiennent des informations supplémentaires qui portent sur l'élément et qu'on ne souhaite pas afficher avec le contenu. Dans cet exemple, l'attribut href permet d'utiliser un lien qui va nous diriger vers un autre site lorsque l'utilisateur clique sur Mon site.

Un attribut doit toujours avoir : un espace entre l'attribut et le nom de l'élément ou l'attribut précédent (s'il y a plusieurs attributs) , un nom (le nom de l'attribut), suivi d'un signe égal « = », des guillemets anglais (") pour encadrer la valeur de l'attribut.

Imbriquer des éléments : vous pouvez placer des éléments au sein d'autres éléments, c'est ce qu'on appelle l'imbrication.

Toutefois, il faut faire attention à ce que les éléments soient bien imbriqués les uns dans les autres. Les éléments doivent être ouverts et fermés correctement de façon à ce qu'ils soient clairement à l'intérieur ou à l'extérieur les uns des autres. S'ils se chevauchent, le navigateur essaiera de choisir la meilleure option, qui ne sera peut-être pas ce que vous vouliez dire et pourrait conduire à des résultats inattendus. Donc pour être d'avoir bien imbriquer son code il faut l'indenter : l'indentation, c'est tout simplement aérer le code en ajoutant des espaces au début de certaines lignes afin de le rendre plus lisible pour vous comme pour les autres. Penser aussi à commenter pour mieux vous repérer. Ces commentaires sont spéciaux : ils ne seront pas visibles par vos visiteurs (à moins que ceux-ci n'affichent le code source de la page).

4) Les balises html

Les balises en HTML sont nombreux nous pouvons citer:

`...` → pour mettre en italique

`...` → pour mettre en gras

`<mark>...</mark>` → pour surligner

`...` → pour une liste non ordonnée

`...` → pour une liste ordonnée

`<il>...</il>` → encadre chaque élément des balises `` ou ``.

`...` → pour mettre un lien la valeur se trouvant entre les guillemets représente le lien vers lequel on veut se déplacer.

`<h2 id="cuisine">La cuisine</h2>`

`cuisine` → pour accéder à la partie cuisine sur la même pages

`...` → vous ne le regretterez pas affichera quand vous pointez la souris sur le bouton

`...` → la page s'ouvrira sur une nouvelle onglet

`...` → génère l'envoi d'une mail

`...` → télécharger un fichier

Si on fait un lien vers site avec une adresse bizarre on remplace les & par &

< → par chevron ouvrant

> → par chevron fermant

& → par &

 → par espace

`` : src → chemin

image / alt → texte alternatif / title → sera au survol de l'image

Afficher la version miniature sur la page et faire un lien pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature:

On met d'abord les images montagne et montagne.mini dans le dossier img

`<p>`

Cliquez pour voir l'image dans sa taille d'origine `</br>`

``

`</p>`

`<figure>`

``

`<figcaption>Le logiciel bloc-notes</figcaption>`

`</figure>`

Figcaption → libellé qui se rapporte en bas de la figure

Extension d'image:

Jpeg → photo

Png → illustration, logo

Gif → image animé

II. CSS

1. Historique

De la même façon que HTML, CSS n'est pas vraiment un langage de programmation. C'est un langage de feuille de style, c'est-à-dire qu'il permet d'appliquer des styles sur différents éléments sélectionnés dans un document HTML. Par exemple, on peut sélectionner tous les éléments d'une page HTML qui sont paragraphes et afficher leurs textes.

Les feuilles de style étaient déjà là sous une forme ou une autre depuis le début de HTML dans les années 1990. Plusieurs navigateurs avaient leur langage de style propre qui pouvait être utilisé pour personnaliser les documents web. À l'origine, les feuilles de style étaient destinées aux utilisateurs finaux. Les premières versions de HTML ne possédaient pas beaucoup d'outils pour la mise en forme, et c'était donc souvent à l'utilisateur de décider comment le document allait s'afficher.

HTML se développait, et il permettait de répondre à l'attente des développeurs web. Avec ces fonctions naissantes, les feuilles de style devinrent moins importantes, et un langage externe permettant de définir le style n'a pas été entièrement accepté jusqu'au développement de CSS.

Le concept des feuilles de style a été à l'origine proposé en 1994 par Håkon Wium Lie ; Bert Bos travaillait au même moment sur un navigateur appelé Argo, utilisant les feuilles de style ; ils décidèrent alors de développer ensemble CSS.

Quelques langages de feuille de style avaient déjà été proposés, mais CSS était le premier à inclure l'idée de « cascade » (feuille de style en cascade) - la possibilité pour le style d'un document d'être hérité à partir de plus d'une « feuille de style ». Cela permettait d'outrepasser le style d'un site spécifique, en héritant, ou en « cascasant » le style du site dans d'autres régions. Cette fonction permet donc un gain de contrôle, à la fois pour l'auteur du site, et pour l'utilisateur. Elle permet aussi un mélange de préférences relatives de style.

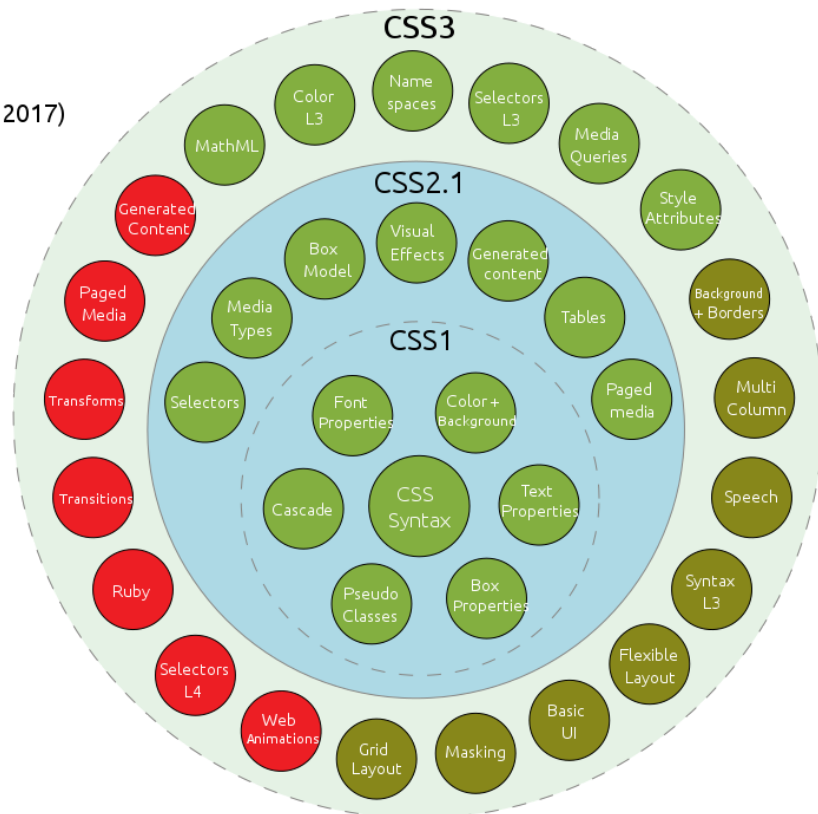
La proposition de Håkon a été présentée à la conférence « Mosaic and the Web » de Chicago en 1994, et par Bert Bos en 1995. À cette même époque, le World Wide Web Consortium (W3C) se créait, et le consortium se prit d'intérêt pour CSS, et organisa un travail autour du langage. Håkon et Bert étaient les responsables du projet. D'autres membres, comme Thomas Reardon (de Microsoft) y ont participé. Fin 1996, CSS était presque prêt à être officialisé. La norme CSS1 fut publiée en décembre 1996.

En 1997, CSS a été attribué à un groupe de travail au sein du W3C, présidé par Chris Lilley. Ce groupe commença par s'occuper des problèmes non réglés par CSS1, ce qui entraîna la parution de CSS2 comme une recommandation officielle en mai 1998. CSS3 est actuellement en développement.

CSS3

Taxonomy & Status (September 2017)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Obsolete or inactive



2. Les bases en CSS

Ce langage sert à définir un ensemble de règles stylistiques à l'intérieur de pages web codées en HTML ou en XML. CSS est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS (.css) et contient des éléments de codage. Il est venu compléter le HTML.

1) Où écrire le CSS?

Nous avons trois possibilités pour écrire notre CSS. Nous pouvons écrire le CSS :

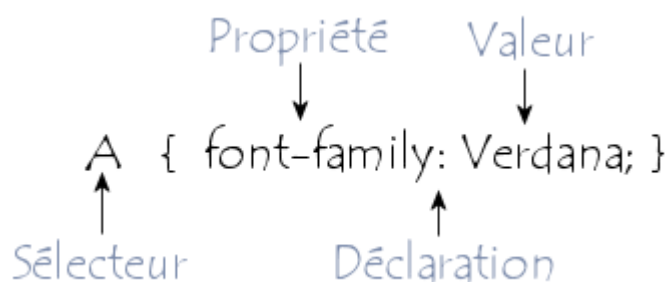
- À l'intérieur de l'élément head de notre document HTML en utilisant la balise style et de placer nos déclarations CSS à l'intérieur de cet élément style ;
- Dans la balise ouvrante de l'élément concerné dans le fichier HTML on utilise l'attribut style et lui affecter en valeur nos propriétés CSS ;
- Dans un nouveau fichier créé dans votre éditeur de texte portant l'extension .css. Une fois le fichier.css créé vous allez devoir lier les deux fichiers HTML et CSS à l'aide de l'élément link que l'on place dans l'élément head. L'élément link est représenté sous forme de balise orpheline et doit être accompagné de ses deux attributs « rel » et « href ». L'attribut rel sert à préciser le style du fichier lié (dans notre cas c'est une feuille de style, donc « stylesheet » en anglais). L'attribut href, que vous connaissez déjà, sert à faire le lien en soi.

2) Structure de CSS

Cette structure s'appelle un ensemble de règles (ou seulement « une règle »). La définition d'un style se fait à l'aide de règles en texte simple permettant de décrire l'aspect des éléments de la page. Une règle CSS est caractérisée par deux principaux éléments :

- Un sélecteur de balises est le nom de l'élément HTML situé au début de l'ensemble de règles. Il permet de préciser à quelles balises du document le style s'applique.
- Une déclaration de style, définie entre accolades, permettant de préciser le style à appliquer aux balises sélectionnées. La déclaration est elle-même constituée :
 - d'une ou plusieurs propriété(s), suivie du caractère « : » (double point),
 - d'une ou plusieurs valeur(s) associée(s) à droite de la propriété, après les deux points entourée de guillemets s'il s'agit de plusieurs mots ou séparés
- Chaque ensemble de règles, à l'exception du sélecteur, doit être entre accolades ({}).
- Pour chaque déclaration, il faut utiliser deux points (:) pour séparer la propriété de ses valeurs.
- Pour chaque ensemble de règles, il faut utiliser un point-virgule (;) pour séparer les déclarations entre elles.

Ainsi, si on veut modifier plusieurs propriétés d'un coup, on peut utiliser plusieurs déclarations dans une seule règle en les séparant par des points-virgules.



3) Les sélecteurs

On appelle « sélecteur de balise » (ou « sélecteur d'éléments ») le ou les mots clés précédant l'accolade et servant à indiquer le ou les balises du document auxquelles le style entre accolades s'applique. Il y a différents types de sélecteurs :

★ Sélecteur universel :

Grâce au sélecteur universel (« * ») il est possible de définir un style s'appliquant à tous les éléments HTML. La syntaxe du sélecteur universel est la suivante :

```
* { /* style */ }
```

★ Sélecteur Simple :

Ils correspondent à des éléments HTML seuls et sans attributs (par exemple le sélecteur p). Ce type de sélecteur doit être préféré tant que possible pour des raisons d'optimisation et de performance du code. En effet, ils requièrent moins de code et sont donc moins gourmands en énergie que des sélecteurs plus complexes. Votre page mettra ainsi moins de temps à charger.

```
body { /* style */ }
```

Le problème reste qu'on est quand même très limité avec des sélecteurs simples. Pour appliquer un style différent à deux éléments de même type, deux paragraphes par exemple, il faut créer un attribut class ou id.

Class et Id sont deux attributs HTML qui ont été créés pour pouvoir appliquer différents styles à des éléments de même type. L'attribut id assigne un nom à l'élément sur lequel il s'applique. Pour un balisage correct, il ne peut y avoir qu'un et un seul élément avec ce nom. L'attribut class assigne un nom de classe à un élément. Ce nom peut être utilisé sur plusieurs éléments dans la même page.

★ Sélecteur d'identifiant

L'élément d'une page qui possède l'ID fourni (pour une page HTML donnée, on ne peut avoir qu'un seul élément pour un ID donné).

```
#my-id { /* style */ }
```

★ Sélecteur de classe

Les éléments d'une page qui sont de la classe donnée (pour une page donnée, il est possible d'avoir plusieurs éléments qui partagent une même classe).

```
.my-class { /* style */ }
```

★ Sélecteur d'attribut

Les éléments de la page qui possèdent l'attribut donné.

```
img[src] { /* style */ }
```

★ Sélecteur regroupement

h1,h2,h3 { /* style */ }

★ Sélecteur d'éléments imbriqués

Permet de créer une règle ne s'appliquant que lorsque un élément Y est imbriqué dans un élément X. Sa syntaxe est la suivante :

selecteur_X selecteur_Y { /* style; */ }

★ Sélecteur d'éléments consécutifs

Permet de créer une règle ne s'appliquant que lorsque un élément Y suit immédiatement un élément X. Sa syntaxe est la suivante :

selecteur_X + selecteur_Y { /* style; */ }

★ sélecteur d'éléments père-fils

Permet de créer une règle ne s'appliquant que lorsque un élément Y est fils direct d'un élément X. La règle ne s'applique pas si Y est encapsulé dans une ou plusieurs autres balises intermédiaires. Sa syntaxe est la suivante :

selecteur_X > selecteur_Y { /* style; */ }

4) Quels propriétés de CSS

P

{

Colors: blue;

}

background-color: → mettre un couleur de fond

background-attachment: fixed; → fixe

background-attachment: scroll; → fait défiler avec le texte

background-size: cover; → redimensionne l'image afin qu'elle s'adapte à la taille de l'élément qui la contient.

background-repeat: no repeat; → pour empêcher le background de se répéter

background-position: top right; positionne le background en haut à droite

opacity: → faire une transparence

background-image: url("ynomimage") → image de fond

Font-size: valeur → la taille de texte

Font-family: → définir la police

Font-style: italic; → mettre en italique

Font-weight: bold; → mettre en gras

Text-decoration: underline; → souligner

Text-decoration: overline; → ligne au dessus

Text-decoration: line-through; → barré

Text-transform: uppercase; → mettre majuscule

Text-align: valeur → aligner le texte

valeur de Text-align :

Center → centrer le texte

Right → aligné à droit

Justify → justifier le texte(qu'il prenne tt la largeur

Float: left; → le texte entour l'image

Clear: right; → annule le flottant en droite

border: 3px blue solid; → mettre un bordure cde 3px avec un color blue et de type solide

Border-radius: valeur → bordure arrondi

Box-shadow : 6px(décalage horizontal de l'ombre) 6px(décalage vertical de l'ombre)

0px(adoucissement du dégradé) black(couleur de l'ombre) → faire un ombre

Text-shadow : 6px 6px 0px black → faire un ombre pour le texte

List-style-type: none; → éliminé les puce

a:hover

{

color:green; → lien au survol

}

P:link

{

Color: blue; → le lien en bleu quand on clique changement couleur

}

a:active

{

background-color: blue; → quand le visiteur clique sur le lien

}

input:focus

}

background-color: blue; → lorsque l'élément est sélectionné

}

a: visited

{

color:red; → si le visiteur a déjà vu la page concernée

}

Line-height → espace entre ligne

Letter-spacing → espace entre lettres

Word-spacing → espace entre mots

Position: static; → positionné selon le flux normal de la page.

Position: fixed; → fixer

Position: relative; → décaler un élément par rapport à sa position dans le flux au moyen des propriétés de décalage top, right, bottom et left .

Position: absolute;

Z-index → permet par exemple d'indiquer que pour deux éléments A et B partiellement ou totalement superposés, A sera placé au-dessus de B ou inversement.

Overflow: hidden ou scroll ou auto; → définit comment gérer le dépassement du contenu d'un élément dans son

P: first-child → les p qui sont les 1er enfant de n'importe quel éléments html

P: first-last → les p qui sont les derniers enfant de n'importe quel éléments html

P: nth-child(2) → tt les deuxième enfants des éléments html

5) Dégradé

Dégradés radiaux(closest-side:plus proche du centre, farthest-side: plus éloignée centre)

Nous avons deux types de dégradés à notre disposition : des dégradés linéaires ou radiaux.

Les fonds en dégradé sont considérés comme des images et donc, à ce titre, nous allons à nouveau utiliser la propriété background-image pour les créer. Il y a quelques années de cela, tous les navigateurs ne reconnaissaient pas les mêmes propriétés et ne les implémentaient pas de la même manière. En l'occurrence, chaque navigateur avait une méthode bien à lui d'implémenter un fond en dégradé. Les préfixes vendeurs ont alors été créés afin que chaque navigateur affiche une propriété de la même manière que les autres. Ce sont des petits mots clefs qui vont être placés avant la déclaration du type de dégradé souhaité. Même si aujourd'hui la majorité des navigateurs suivent des standards qui ont été établis et implémentent les propriétés de la même façon, cela peut toujours être utile (et ça ne coûte rien) de mentionner des préfixes vendeurs dans le cas où certains de vos visiteurs utiliseraient des vieilles versions de navigateurs.

Les préfixes vendeurs sont les suivants :

- Pour Chrome et Safari : -webkit-
- Pour Mozilla : -moz-
- Pour Internet Explorer : -ms-
- Pour Opera : -o-

a) Les dégradés de type linéaire

Pour créer un dégradé linéaire, on utilise les mots clefs linear-gradient. On peut ensuite spécifier autant de valeurs que souhaitées, chaque valeur correspondant à une couleur de votre dégradé. Afin que le tout fonctionne correctement, vous devez mentionner chaque préfixe vendeur et il est recommandé de finir avec le linear-gradient sans aucun préfixe (ce sera la valeur par défaut) comme ceci :

```
.box_gradient {
  background-color: #444444;
  background-image: -webkit-linear-gradient(linear, left top, left bottom, from(#444444), to(#999999)); /* Safari 4+, Chrome */
  background-image: -webkit-linear-gradient(top, #444444, #999999); /* Chrome 18+, Safari 5.1+, iOS 5+ */
  background-image: -moz-linear-gradient(top, #444444, #999999); /* Firefox 3.6-15 */
  background-image: -o-linear-gradient(top, #444444, #999999); /* Opera 11.10-12.00 */
  background-image: linear-gradient(to bottom, #444444, #999999); /* Firefox 16+, IE10, Opera 12.50+ */
}
```

Les dégradés linéaires peuvent aller du haut vers le bas, de bas en haut, de gauche à droite, de droite à gauche en encore en diagonale. Par défaut, cependant, les dégradés linéaires partent

du haut vers le bas d'un élément. On peut choisir l'inclinaison de nos dégradés en utilisant des mots clefs ou en spécifiant la valeur en degré de l'angle voulu avant d'indiquer les couleurs du dégradé. Pour que votre dégradé aille vers la droite, on utilisera la valeur `to right`, pour qu'il aille à gauche `to left` et pour qu'il aille de bas en haut `to top`.

b) Les dégradés de type radial

Un dégradé de type radial est un dégradé créé à partir d'un point central. Les dégradés de type radial fonctionnent sur le même schéma que ceux de type linéaire. Cette fois-ci, nous allons utiliser les mots clefs `radial-gradient`.

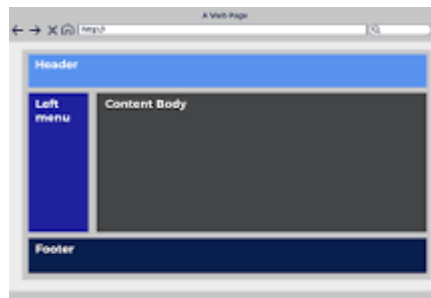
On peut choisir si l'on souhaite que notre dégradé ait la forme d'un cercle ou d'une ellipse en précisant les mots clefs `circle` et `ellipse` juste avant nos couleurs.

On peut également choisir avec précision les dimensions et la forme de notre dégradé en précisant deux chiffres (généralement en px) correspondant respectivement à la largeur et à la hauteur de notre dégradé.

6) Flex-box

Le flexbox est un modèle de disposition très puissant qui va nous permettre de contrôler facilement et avec précision l'alignement, la direction, l'ordre et la taille de nos éléments (ou plus précisément de nos boîtes). Une manière de répartir des espaces et des positionnements

Comment utiliser les Flexbox ?



Pour utiliser Flexbox, il suffit de mettre la propriété css `display:flex` sur l'élément parent

Conclusion

Les langages HTML et CSS sont véritablement le socle de tout projet de développement web. Que vous vouliez créer un site e-commerce, un blog, une application mobile ou quoique ce soit d'autre, vous serez obligé de passer par les langages HTML et CSS. Apprendre le HTML et le CSS signifie entrer dans le monde des programmeurs et c'est donc commencer à les comprendre et à parler comme eux.