Création de votre première page Web

Imaginer un temps, avant l'invention de l'Internet ou les sites n'existaient pas, et les livres, imprimés sur papier, étaient votre principale source d'information. Il vous fallait beaucoup d'efforts et de lecture pour trouver l'information exacte que vous cherchiez.

Aujourd'hui vous pouvez ouvrir un navigateur Web, allez sur le moteur de recherche de votre choix, et faire votre recherche. Toute l'information imaginable est au bout des doigts. Avec un peu de chances quelqu'un quelque part aura construit un site Web qui pourra apporter une réponse à votre recherche.

Nous allons étudier comment construire vos propres sites Web en utilisant les deux techno nécessaires que sont les langages HTML et CSS.

Avant de commencer à apprendre comment construire un sites Web avec HTML et CSS, il est important de comprendre les différences entre les deux langages, leurs syntaxes, et leurs terminologies.

Qu'est ce que sont HTML & CSS?

Le HTML ou HyperText Markup Language, donne la structure et le sens du contenu en définissant par exemple,titres,paragraphes ou les images.

Le CSS ou Cascading Style Sheets, est un langage de présentation créé pour modifier l'apparence du contenu -par exemple les polices ou les couleurs-.

HTML et CSS sont indépendants l'un de l'autres et le resteront. On verra que dans les bonnes pratiques le CSS ne doit pas être écrit à l'intérieur d'un document HTML. En règle générale, HTML représentera toujours le contenu, et CSS représentera toujours l'apparence de ce contenu.

Cette règle inébranlables de la différence entre HTML et CSS comprise, nous allons plonger dans le code HTML de façon plus détaillée.

Comprendre les termes du HTML

Les trois termes HTML que vous devrez apprendre sont les *éléments*, *balises* et *attributs*.

Les éléments

Les éléments désignent ce qui définit la structure et le contenu des objets dans une page. Certains des éléments les plus fréquemment utilisés incluent plusieurs niveaux d'entêtes (les éléments <h1> à <h6>) des paragraphes (identifiés par l'élément <p>) sachant que la liste contient aussi les éléments <a>, , , et , et bien d'autres encore...

Les éléments sont identifiés par l'utilisation des signes moins-que et plus-que -inférieur / supérieur- "**<**" "**>**" entourant le nom de l'élément. Ainsi, un élément sera libellé ainsi :

```
1<a>
```

Les balises

L'utilisation des signes moins-que et plus-que entourant le contenu d'un élément créé une *balise (tag en anglais).* Les balises vont le plus souvent par paires avec une balise d'ouverture et fermeture.

La *balise d'ouverture* marque le début d'un élément. Il se compose d'un signe inférieur suivi du nom d'un élément, et se termine par un signe supérieur, par exemple .

La *balise de fermeture* marque la fin d'un élément. Elle se compose d'un signe inférieur suivi d'une barre oblique et le nom de l'élément, puis se termine par un signe supérieur, par exemple .

Le contenu qui se situe entre l'ouverture et la fermeture de balises est le contenu de cet élément. Un lien d'ancrage, par exemple, aura une balise ouverture de *<a>* et une balise de fermeture de **. Ce qui est entre ces deux balises sera le contenu, ou libellé, affiché du lien sur lesquel on va cliquer:

```
1<a>...</a>
```

Les attributs

Les attributs sont les propriétés utilisées pour fournir des informations supplémentaires sur un élément. Les attributs les plus communs incluent *id*, l'attribut qui identifie un élément unique, l'attribut *class*, qui qualifie

une classe css que l'on va retrouver sur un ou des éléments, *src*,l'attribut qui spécifie une source de contenu intégrable, comme une image, et *href*,l'attribut qui fournit une référence de lien hypertexte vers une ressource liée.

Les attributs sont définis dans la balise d'ouverture, après le nom d'un élément. Généralement les attributs comprennent un nom et une valeur. Le format de ces attributs se compose du nom de l'attribut suivi d'un signe égal et une valeur d'attribut entourée par des doubles quotes " " ". Par exemple, un élément <a> incluant un attribut href ressemblera à ce qui suit :

```
1<a href="http://google.com/">Google</a>
```

Démo "Termes HTML communs"

Google

Le code précédent affiche le texte "Google" sur la page Web et mènera les utilisateurs à http://google.com lorsque vous cliquez sur le texte "Google". L'élément d'ancrage est déclaré dans les balises d'ouverture `<a>` et de fermeture `</a>` englobant le texte. L'attribut de référence de lien hypertexte et sa valeur sont déclarées dans la balise d'ouverture avec `href="http://google.com"`, `href` étant le nom et `http://google.com` étant la valeur.

Maintenant que vous savez ce que éléments HTML,balises et attributs sont, nous allons mettre sur pied notre première page web.

Mise en place de la structure du document HTML

Les documents HTML sont des fichiers de texte brut enregistrés avec une extension `.html` plutôt qu'une `.txt`. Pour commencer à écrire du HTML, vous devez donc utiliser un éditeur de texte brut. Cela ne comprend pas Microsoft Word ou Pages qui sont des éditeurs de texte riche. il existe de nombreux éditeurs de texte pour écrire du HTML et CSS dont les plus populaires sont Notepad++ et Sublime Text. Il existe de nombreuses alternatives comme Komodo Edit ou PSPad pour Windows et TextWrangler ou Smultron pour Mac. Il existe aussi des logiciels d'éditions encore plus avancés comme ATOM, Bracket, Visual Studio Code qui sont des éditeurs plus récents disponibles pour tous les systèmes.

Tous les documents HTML ont une structure nécessaire qui comprend la déclaration `<!DOCTYPE html>` et à minima les éléments `<html>`,`<head>`et`<body>`.

La déclaration de type de document ou `<!DOCTYPE html>` informe le navigateur web sur quelle version de HTML sera utilisée dans le document et doit être placé au début du document HTML. Parce que nous allons utiliser la dernière version de HTML, notre déclaration de type de document est simplement `<!DOCTYPE html>`. Suit la déclaration de type de document, l'élément `<html>`signifie que c'est le début du document.

A l'intérieur de l'élément `<html>` l'élément `<head>` identifie l'entête du document. Le contenu du `<head>` n'est pas affiché dans la page Web elle même. Il inclut le titre du document (qui est affiché sur la barre de titre de la fenêtre du navigateur), les liens vers des fichiers externes, ou tout autre métadonnées utile.

Tout le contenu visible dans la page Web va se trouver dans le `<body>` élément. Une rupture d'une structure typique de document HTML ressemble ceci:

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Bonjour le monde</title>
  </head>
  <body>
    <h1>Bonjour le monde</h1>
    <p>Ceci est une page Web.</p>
  </body>
</html>
```

Structure du document HTML exemple

Le code précédent montre le document commençant par la déclaration du type de document, `<!DOCTYPE html>`, suivie directement par l'élément `<html>`. A l'intérieur de l'élément `<html>` on a un élément `<head>` et un élément `<body>`. L'élément `<head>` comprend le codage des caractères de la page via l'élément `<meta charset="utf-8">` et le titre du document via l'élément `<title>`. L'élément `<body>` comprend un titre, l'élément `<h1>` et un paragraphe, l'élément `<p>`. Parce qu'à la fois le titre et le paragraphe sont imbriqués dans l'élément `<body>`, ils seront visibles sur la page Web.

Quand un élément est placé à l'intérieur d'un autre élément, il est de bonne pratique de le mettre en retrait pour maintenir la structure du document bien organisé et lisible. Dans le code précédent, les éléments `<head>` et `<body>` ont été imbriqués et indentés à l'intérieur de l'élément `<html>`.

Eléments auto-fermant

Dans l'exemple précédent, l'élément `<meta>` à une seule étiquette et n'inclut pas de balise de fermeture. Cela était intentionnel. Tous les éléments sont constitués d'ouverture et de fermeture de balises. Certains éléments reçoivent simplement leur contenu ou le comportement d'attributs dans une seule balise "auto-fermante". L'élément `<meta>` est un de ces éléments.

D'autres éléments de auto-fermant sont :

- `<br>`
- `<hr>`
- `<img>`
- `<input>`
- `<link>`
- `<meta>`
- `<wbr>`

La structure décrite ici, `<!DOCTYPE html>`, `<html>`, `<head>` et `<body>` est commune à l'ensemble des fichiers HTML. Nous garderons cette structure de document à portée main car nous allons l'utiliser dès que nous créerons de nouveaux documents HTML.

Validation du code

Quel que soit le soin que nous allons apporter à l'écriture de notre code, nous allons inévitablement faire des erreurs. Heureusement, pour nous aider lors de l'écriture du HTML et du CSS nous avons des validateurs pour vérifier notre travail. Le W3C a construit deux validateurs HTML et CSS qui vont scanner le code pour détecter les erreurs. Valider notre code permet non seulement de le rendre correct pour tous les navigateurs, mais contribue également à nous enseigner les meilleures pratiques pour écrire le code.

En pratique

Nous allons maintenant construire un site web que nous améliorerons et complèterons tout au long des leçons suivantes. C'est parti !

1. Ouvrons notre éditeur de texte, créons un nouveau fichier nommé `index.html`, et enregistrons le dans un endroit que nous n'oublierons pas. Je vais créer un dossier sur mon bureau nommé "bcma" et enregistrez le fichier à cet endroit.

2. Dans le fichier index.html, ajoutons la structure du document, avec le type de document `<!DOCTYPE html>` et les éléments `<html>`, `<head>` et `<body>`.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
  </head>
  <body>
  </body>
</html>
```

3. Dans le `<head>`, ajoutons les éléments `<meta>` et `<title>`. L'élément `<meta>` doit inclure l'attribut `charset` approprié et sa valeur, tandis que l'élément `<title>` doit contenir le titre de la page, à savoir "Cours HTML / CSS".

```
<head>
  <meta charset="utf-8">
  <title>Cours HTML / CSS</title>
</head>

```

4. Dans `<body>` ajoutons un titre `<h1>` et un paragraphe `<p>`. Dans le titre `<h1>` nous allons utiliser de nouveau "Cours HTML / CSS" et dans le paragraphe `<p>` nous allons inclure un texte simple pour présenter le cours.

```
<body>
  <h1>Cours HTML / CSS</h1>
  <p>Chaque année le cours HTML/CSS prépare de brillants
  développeurs front-end. Rejoignez nous dès Septembre
prochain !
  </p>
</body>
```

5. Maintenant il est temps de voir ce que nous avons fait ! Allons trouver notre fichier index.html (le mien est dans le dossier "bcma" sur mon bureau). Double-cliquez sur ce fichier ou faites le glisser dans un navigateur Web qui ouvrira la page en visualisation.

Fig 1 Nos premiers pas dans la construction du site

Une fois notre structure mise en place, nous allons maintenant jeter un oeil du côté du CSS.

Rappelez vous, HTML définit le contenu et la structure de nos pages web, alors que CSS définit le style visuel et l'aspect de nos pages web.

Comprendre les termes CSS

En complément des termes HTML, il y a quelques termes CSS avec lesquels vous devrez vous familiariser. Ces termes sont *sélecteurs*, *propriétés* et *valeurs.* Comme avec la terminologie HTML, plus vous travaillerez avec CSS, plus ces termes deviendront une seconde nature.

Les sélecteurs

Tous les éléments ajoutés à une page Web peuvent être “stylés” à l’aide de CSS. Un *sélecteur* désigne exactement un ou plusieurs élément-s au sein du HTML. On utilise un sélecteur pour cibler et appliquer un ou des style-s (tels que la couleur, taille et la position…). Les sélecteurs peuvent inclure une combinaison de différents qualificatifs pour sélectionner des éléments uniques, tout dépend de la précision de sélection que l’on souhaite atteindre. Par exemple, on peut vouloir sélectionner chaque paragraphe sur une page, ou nous pouvons vouloir choisir un seul paragraphe spécifique sur une page.

Les sélecteurs ciblent généralement une valeur d'attribut, comme un `id` ou une `class`,ou va cibler un type d'élément tels qu’un titre `<h1>` ou un `<p>`.

En CSS, les sélecteurs sont suivis d’accolades, `{}`,qui englobent les styles à appliquer à l'élément sélectionné. Le sélecteur ci-dessous vise tous les éléments `<p>`.

```
p{ ... }
```

Les propriétés

Une fois qu’un élément est sélectionné, une propriété détermine les styles qui seront appliqués à cet élément. Les noms de propriété se mettent après un sélecteur, dans les accolades,`{}`,et immédiatement après le sélecteur on va mettre deux points pour le séparer de sa valeur.

Il existe de nombreuses propriétés que nous pouvons utiliser, comme `background`, `color`, `font-size`, `height`, et `width`… Il y’en a de nombreuses autres et de nouvelles propriétés sont souvent ajoutés au fil des années. Dans le code suivant, nous définissons les propriétés `color` et `font-size` à appliquer à tous les éléments `<p>`.

```
p{
  color: ...;
  font-size: ...;
}
```

Valeurs

Jusqu'à présent nous avons sélectionné un élément avec un sélecteur et déterminé quel style nous aimerions appliquer avec une propriété. Maintenant nous pouvons déterminer le comportement de cette propriété avec une valeur.

Les valeurs peuvent être identifiés comme étant le texte entre les deux point `":"` et le point virgule `";"`.

Ici nous sélectionnons tous les éléments `<p>` et nous allons donner comme valeur à la propriété *couleur* `-color- orange` et la valeur de la propriété *taille de la police* `-font-size-` à `16px -pixels-` .

```
1 p{
2  color: orange;
3  font-size: 16px;
4 }
```

Pour résumer, en CSS une ***règle***, *un ensemble de propriété par sélecteur*, commence avec le sélecteur, qui est immédiatement suivie par des accolades. Aux sein de ces accolades sont déclarées des paires propriétés / valeurs.
Chaque déclaration commence par une propriété, qui est suivi par deux points et la valeur de la propriété délimitée par un point virgule.
Il est une pratique courante d'indenter la paire propriété et valeur dans les accolades.

Comme pour le HTML, ces indentations aide à garder notre code organisé et lisible.

Connaître la syntaxe générale du CSS est un bon début, mais nous aurons quelques autres leçons à apprendre avant d'aller au tréfond du sujet.

Nous allons voir comment il faut travailler avec les sélecteurs au sein d'une CSS.

Travailler avec sélecteurs

Les sélecteurs, comme mentionnés précédemment, indiquent quels éléments HTML sont stylés. Il est important de bien comprendre comment utiliser ces sélecteurs et comment en tirer profit. La première étape est de se familiariser avec les différents types de sélecteurs.

Nous allons commencer avec les sélecteurs les plus courant *type*, *class* et *Id*.

Sélecteur de type

Les sélecteurs *type* ciblent les éléments par leur type. Par exemple, si nous voulons cibler tous les éléments de paragraphe,`<p>`,nous devrons utiliser un sélecteur de type `p`.

CSS

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

HTML

```
1<p>...</p>
2<p>...</p>
```

Sélecteur de classe *class*

Les sélecteurs *class* nous permettent de sélectionner un élément en fonction de la valeur de l'attribut `class` de l'élément. Les sélecteurs de classe sont un peu plus précis que les sélecteurs de type, ils sélectionnent un groupe particulier d'éléments plutôt que tous éléments d'un seul type.

Les sélecteurs de classe nous permettent d'appliquer les mêmes styles à différents éléments à la fois en utilisant la même valeur de classe.

En CSS, les classes sont désignées par un point "`.`" précédent le libellé et suivi des valeurs de propriétés de l'attribut de classe entre accolades. Ici le sélecteur de classe sélectionnera tous les éléments qui ont pour valeur de l'attribut `class` la valeur `magnifique`.

Dans l'exemple ci dessous les deux éléments div et p seront sélectionnés mais pas le h1.

CSS

```
1 .magnifique{ ... }
```

HTML

```
1 <div class="magnifique">...</div>
2 <h1 class="titre">...</h1>
3 <p class="magnifique">...</p>
```

Sélecteur par identifiant *ID*

Les sélecteurs *d'ID* sont encore plus précis que les sélecteurs de classe, car ils visent seulement un unique élément à la fois.

Quel que soit type d'élément représenté par un identifiant cet attribut ne peut être utilisé qu'une seule fois par page. Si vous l'utilisez, ils doit être réservé à des éléments très significatifs.

En CSS, les sélecteurs d'ID sont indiqués par un signe dièse, "`#`", suivi du libellé de l'identifiant et des déclarations des valeurs de propriétés entre accolade.

Ici le sélecteur d'ID ne sélectionnera que l'élément div contenant cet `id` et ayant pour valeur l'attribut " `monidentifiant` " à savoir la ligne 2 du code HTML.

CSS

```
1 #mondidentifiant{ ... }
```

HTML

```
1 <div id="whatelse">...</ div>
2 <div id="mondidentifiant">...</ div>
3 <div>...</ div>
```

Sélecteurs supplémentaires

Les sélecteurs sont extrêmement puissants, et les sélecteurs décrits jusqu'à présent sont les sélecteurs les plus courants que nous allons rencontrer. Il y'en a de nombreux autres, plus avancés, qui existent. Lorsque nous serons à l'aise avec ces sélecteurs, on verra dans un prochain cours des sélecteurs plus avancés, sachant que l'on pourra aussi combiner tous ces sélecteurs…

Si l'on résume tout commence à se mettre en place. Nous ajoutons des éléments à une page dans le HTML, et nous pouvons alors sélectionner ces éléments et appliquer des styles sur ces propriétés CSS. Maintenant nous allons relier nos éléments HTML et leur correspondances dans le CSS, et faire fonctionner les deux ensemble.

Lier le CSS et le HTML

Afin d'appliquer notre CSS à notre HTML, nous avons besoin de faire référence à notre CSS au sein de notre fichier HTML. La meilleure pratique pour insérer du CSS est d'inclure tous nos styles en une seule feuille de style externe, un fichier, qui sera référencé à partir du `<head>` de notre document HTML. L'utilisation d'une feuille de style externe unique nous permet d'utiliser les mêmes styles sur tout le site Web sans avoir à apporter des modifications au niveau de chacune des pages du site.

Les autres options pour ajouter le CSS au HTML

Les autres options pour le référencement du CSS comprennent l'utilisation de la balise de styles internes `<style>` ou directement dans l'attribut style d'une balise, c'est à dire en "inline", de cette façon avec l'attribut "`style`" par exemple `<p style="...">` . Vous pouvez rencontrer ces options dans la vie, mais ils sont généralement mal vues, car elles rendent la mise à jour des sites lourde et difficile. Ce sont de mauvaises pratique, mais parfois indispensable !

La méthode à privilégier

Pour créer notre feuille de style CSS externe, nous allons utiliser à nouveau notre éditeur de texte pour créer un nouveau fichier texte avec l'extension de fichier `.css`. Notre fichier CSS doit être enregistré dans le même dossier ou un sous dossier, où notre fichier HTML a été enregistré .

Dans la `<head>` du document HTML, l'élément `<link>` est utilisé pour définir la relation entre le fichier HTML et le fichier CSS. Parce que nous relions à la CSS, nous utilisons l'attribut `rel` avec la valeur `stylesheet` pour spécifier leur type de relation. En outre, l'attribut `href` (ou référence de lien hypertexte) est utilisé pour identifier l'emplacement, ou chemin, du fichier CSS.

Prenons l'exemple suivant d'un document HTML`<head>`qui fait référence une feuille de style externe unique.

```
<head>
  <link rel="stylesheet" href="main.css">
</head>
```

Pour le que le CSS soit rendu correctement, le chemin de la valeur de l'attribut `href` doit directement adresser l'endroit où notre fichier CSS est enregistré. Dans l'exemple précédent, le `main.css` fichier est stocké dans le même emplacement que le fichier HTML, également connu sous le nom de répertoire racine.

Si notre fichier CSS est dans un sous répertoire ou sous dossier, la valeur de l'attribut `href` doit correspondre à ce chemin en conséquence . Par exemple, si notre fichier `main.css` a été stocké dans un sous répertoire nommé `"css"`, la valeur de l'attribut `href` sera `"css/main.css"`, utilisant une barre oblique pour indiquer le déplacement dans un sous répertoire.

À ce stade nos pages commencent à ressembler à quelque chose, lentement mais sûrement. On n'a pas encore investigué encore beaucoups dans le CSS, mais vous avez pu remarquer que certains éléments ont des styles par défaut que nous n'avons pas déclarés dans notre CSS.

En effet le navigateur impose ses propres styles CSS par défaut pour les éléments. Heureusement nous pouvons remplacer ces styles assez facilement, ce que nous allons faire d'abord en utilisant une feuille de réinitialisation CSS.

Expliquons la "réinitialisation" CSS

Chaque navigateur Web a ses propres styles par défaut pour différents éléments. La façon dont Google Chrome rend les titres, paragraphes, listes, et ainsi suite peut être différente de la façon dont Firefox, Edge ou Internet Explorer le font. Pour assurer la compatibilité cross-browser -*pour tous les navigateurs*-, il faut utiliser ce que l'on appelle une réinitialisations CSS.

Les réinitialisations CSS prennent chaque élément HTML commun avec un style prédéfini et fournissent un seul style unifié pour tous navigateurs. Ces réinitialisations impliquent généralement la suppression des dimensions, marges, padding ou styles supplémentaires et unifie ces valeurs. Parce que dans CSS, comme son nom l'indique il y'a une cascade des définitions qui s'applique de haut en bas, il faut pour assurer la réinitialisation, il faut que la première de notre fichier HTML soit la feuille de style de réinitialisation. Cela assure que ces styles seront lus en premier et que tous les différents navigateurs web fonctionneront à partir d'une base commune.

Il y a un tas de différentes réinitialisations disponibles à utiliser, qui ont tous leurs propres qualités. L'une des CSS de réinitialisations les plus populaires est la remise à zéro de Eric Meyer, qui a été adapté pour inclure tous les styles des nouveaux éléments HTML5.

Si vous vous sentez un peu plus aventureux, il y a aussi Normalize.css, créé par Nicolas Gallagher. Normalize.css ne se concentre pas d'une réinitialisation de tous éléments communs, mais plutôt de la redéfinition de styles communs pour tous les éléments. Cette méthode exige une grande compréhension des CSS, ainsi que la conscience de ce que vous souhaitez avoir dans vos styles.

Cross-compatibilité des navigateurs et essais

Comme mentionné précédemment, les différents navigateurs rendent les éléments de différentes manières. Il est important de reconnaître la valeur de compatibilité et les tests multi-navigateur. Les sites ne doivent pas exactement être les mêmes dans tous navigateurs, mais ils doivent être proches. Savoir avec quels navigateurs vous souhaitez être compatible, et à quel degré, est une décision vous aurez besoin de faire sur la base de ce qui sera le mieux pour votre site web.

En pratique

Exerçons nous à ajouter un peu de CSS dans notre fichier HTML.

1. Dans notre dossier "bcma", nous allons créer un nouveau dossier nommé "assets". Nous stockerons dans ce dossier toutes les ressources pour notre site Web, tels que nos feuilles de style, images, vidéos, et ainsi de suite. Pour nos feuilles de style, nous allons aller avant et ajouter un autre dossier nommé "css" dans le dossier "assets".

2. Grâce à notre éditeur de texte, nous allons créer un nouveau fichier nommé `styles.css` et enregistrez le dans le dossier "css" que nous venons créer.

3. En regardant notre fichier `index.html` dans un navigateur Web, nous pouvons voir que les éléments `<h1>` et `<p>` ont chacun un style CSS par défaut. Précisément, ils ont chacun une taille de police unique et un espacement autour eux. En utilisant la css de réinitialisation d'Eric Meyer, nous pouvons harmoniser ces styles, permettant que chacun d'entre eux partes de la même base. Pour ce faire nous allons accéder à la css sur [le site Web d'Eric Meyer] ,copiez sa css, et collez le dans la partie supérieure de notre fichier `styles.css.`

4. Notre fichier `styles.css` commence à prendre forme, nous allons maintenant le connecter à notre fichier `index.html.` Ouvrir le fichier `index.html` dans votre éditeur de texte, ajouter l'élément `<link>` dans l'élément `<head>,` juste après l'élément <title>.

5. Nous allons référencer une feuille de style dans l'élément `<link>,` nous allons ajoutez l'attribut de relation, `rel`,avec pour valeur `stylesheet`.

6. Nous allons également inclure une référence de lien hypertexte en utilisant l'attribut `href` pour notre fichier `styles.css`. Rappelez vous que `styles.css` est enregistré dans le dossier “`css`”, qui se trouve dans le dossier “`assets`” par conséquent, la valeur de l'attribut `href` qui est le chemin à notre fichier `styles.css` sera “`assets/css/styles.css`”.

```
<head>
  <meta charset="utf-8">
  <title>Cours Html</title>
  <link rel="stylesheet" href="assets/stylesheets/main.css">
</head>
```

Il est temps de vérifier notre travail et voir si le HTML et le CSS sont corrects.
Maintenant ouvrons notre fichier `index.html` (ou le rafraîchir si la page est déjà ouverte) dans un navigateur Web devrait afficher résultats légèrement différents que précédemment.

Résumé

Nous avons pris quelques grandes étapes de cette leçon. Vous savez maintenant les bases du HTML et CSS.

Pour rappel, jusqu'à présent nous avons découvert :

- La différence entre HTML et CSS
- Se familiariser avec éléments HTML,balises et attributs
- Mis en place la structure de votre première page web
- Se familiariser avec sélecteurs CSS, propriétés et valeurs
- Travailler avec les sélecteurs CSS
- Référencer le CSS dans votre code HTML
- Tester les CSS de réinitialisation

maintenant nous allons regarder de plus près au HTML et en apprendre un peu sur la sémantique.

Récupérer le cours

https://hohll.github.io/BCMA/HTML-CSS/niveau-1/