

Documentation

Projet : Réalisation d'un CV en HTML 5 et CSS 3

1. Conception de la page

A. Avant-propos sur l'orientation du projet

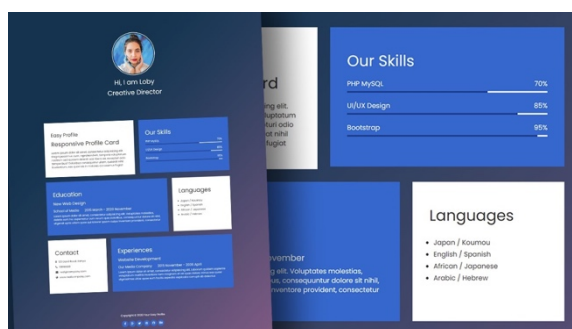
La thématique de l'évaluation sur le HTML et le CSS consiste en la réalisation d'une page Internet représentant notre Curriculum Vitae. Que le C.V soit fictif ou non, je souhaite que mon travail, même dans le cadre d'un projet de formation puisse être utile pour la suite de ma reconversion professionnelle. De ce fait, j'ai préféré m'orienter vers un C.V réaliste sur mon profil et mes compétences / expériences, avec comme challenge personnel de pouvoir le présenter plus tard à des professionnels (dans le cadre d'une recherche de stage par exemple).

B. Recherche et conception

Pour cela, j'ai d'abord pris mon C.V format PDF comme une référence. Le document possède un style visuel (couleurs etc...) qui lui est propre, des catégories, des logos et des icônes. Je considère qu'il s'agit d'une bonne base de réflexion car d'expérience, je sais que mon C.V attire le regard des recruteurs. Je souhaite donc que ce site Internet qui représente une vitrine des compétences en HTML et CSS acquises dans le cadre de ma formation chez Pop School puisse avoir une cohérence avec mon CV au format PDF.



Ensuite, j'ai recherché des idées de conception de design de site type CV (*resume* en anglais) principalement via Google et [Graphic River](https://www.graphicriver.net/) (un site que j'affectionne particulièrement pour trouver de l'inspiration). J'ai opté pour ce style :

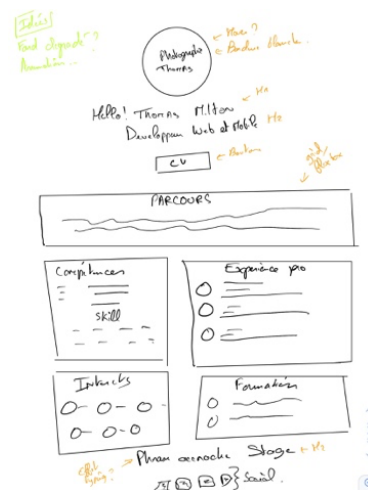


Source : [Online Web Ustaad](https://www.onlinewebustaad.com/)

C'est un style qui me plaît car je le trouve simple, clair et efficace. La photographie ronde centrée en haut de page avec l'identité de la personne et sa profession en titre. Ensuite, j'aime beaucoup la disposition des blocs de compétences format rectangulaire, qui est un bon exercice pour les fonctions grid et flexbox et un bas de page d'icônes de réseau social. Je précise que bien que l'auteur original ait réalisé un tutoriel de création sur YouTube, je ne l'ai pas visionné. Je me suis inspiré que du design et non de son code.

C. Maquettage

Le maquettage a consisté à une simple esquisse sur iPad pour positionner les éléments. En effet, je partais d'une bonne base avec le model design et mon C.V. De plus, je ne maîtrise pas Adobe XD ou Figma pour réaliser un projet de maquettage plus poussé.



2. Choix de la photographie

J'ai utilisé une photographie type profil professionnel « passe-partout », il s'agit d'une photographie me représentant en costume, je l'utilise sur LinkedIn et sur mon C.V. Je pense personnellement qu'elle représente un peu trop mon ancien travail de recruteur et je souhaite à l'avenir en refaire une nouvelle. Concernant la taille et le poids, elle ne fait que 11ko (elle se chargera facilement avec la page), c'est un format carré de 262*262px à 72ppp. C'est une photographie légère et son format me permet de la styliser facilement avec CSS.

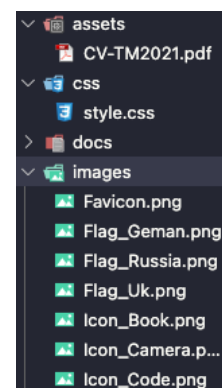
3. Technique HTML / CSS

Lors de mes recherches documentaires, j'ai constaté qu'il y a plusieurs méthodes. Certains développeurs conseillent de développer d'abord la partie HTML au complet et de procéder ensuite à la stylisation de la page par le CSS, d'autres procèdent par étape, en réalisant, par exemple le haut du site en HTML puis en stylisant cette partie, pour ensuite réaliser une autre partie. Je préfère cette seconde méthode, car cela me permet d'avoir un résultat « final » partie par partie et de faire évoluer ma vision et le style en fonction de l'avancement du site. Par exemple, si je me rends compte qu'il y a un espace libre, je peux y ajouter un nouvel élément.

A. Organisation de l'arborescence des fichiers

Concernant l'arborescences des fichiers dans Visual Studio Code, voici ma manière de procéder.

- assets (contient le CV.pdf à télécharger)
- css (contient style.css)
- docs (contient les consignes de l'examen et la documentation)
- images (contient les images qui commencent par le Type suivi du nom pour une meilleur organisation)
- index.html



B. Technique HTML

Je débute la partie HTML par délimiter le site en grandes parties (`<head>`, `<body>` (au sein du `body`, j'y ajoute les balises `<header>`, `<main>` et `<footer>`) puis j'ajoute des commentaires qui vont nommer les différentes sections de type : `<-- header section -->`. Cela afin d'y voir clair et d'avoir une meilleure organisation. Si j'ai besoin d'éditer en CSS une `class` ou un `id` spécifique, je peux le retrouver facilement et rapidement.

Ensuite je m'occupe de la partie `<head>` par l'ajout des balises `<meta>`, la modification du titre et des balises `<link>` vers le style.css et le flavicon. Concernant le flavicon, pour cet exercice, j'ai utilisé un modèle de TradeMark (™), car cela correspond à mes initiales.

J'ai lu que par convention, les noms des `id` et de `class` ainsi que les commentaires importants doivent être rédigé en anglais. C'est ce que j'applique également.

Concernant la méthodologie HTML appliquée à cet examen, j'utilise une balise `<h1>` pour le titre principal (*Hello ! je suis Thomas*) puis des balises `<h2>`, `<h3>` et `<h4>` pour les sous-titres, des balises `` si j'ai besoin de styliser un élément particulier d'un texte. J'utilise des balises `<div>` pour créer des conteneurs (*container*) et des balises `<section>` si j'ai besoin d'appliquer plusieurs paramètres à un ensemble de balises `<div>` (sur l'utilisation de la position Grid ou Flexbox par exemple). A l'intérieur des conteneurs, notamment sur les blocs de compétences/expériences, j'utilise des tableaux via les balises `<table>` ce qui permet d'adapter le contenu et d'ajouter facilement des informations (par l'ajout de lignes (*raw*)). J'utilise

également la balise `
` si j'ai besoin de revenir à la ligne et la balise `<progress>` pour faire des barres de progression ainsi que des balises d'ancrage `<a>` pour des liens ou des logos cliquables.

Je n'ai pas éprouvé de grosses difficultés sur cette partie.

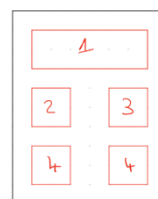
C. Technique CSS

Concernant le CSS, je commence par tester le lien entre le fichier `style.css` et `index.html`, par l'ajoute un background avec l'attribut `*` puis j'applique un « mini reset » avec les attributs : `margin:0 ; padding: 0;` et `box-sizing: border-box;` ce qui donne un meilleur rendu des éléments. Puis, comme pour le HTML, je divise les différentes parties par des commentaires pour une meilleure organisation. Pour plus de visibilité, j'utilise une arborescence de sélecteurs limité à 3 éléments (`#topsite h1...`).

Concernant les paramètres du body, `height: 100%;` permet au body de prendre la hauteur de son parent html ce qui empêche son contenu de changer dynamiquement. Concernant les couleurs, j'ai utilisé les mêmes que celles de mon C.V (`rgb(193, 162,109)`) pour la cohérence. Ensuite j'ai importé une police de [Google](#) font pour plus de personnalisation.

Pour styliser le header, qui contient la photographie, la phrase de présentation, le titre et le bouton, j'ai positionné les éléments avec les attributs `text-align`, `margin` ainsi que `padding`. J'utilise le `border-radius` pour la forme de la photo. Concernant le bouton, j'ai utilisé l'attribut `animation`, `cursor` et la pseudo classe `:hover` pour l'effet d'animation au passage de la souris.

Concernant les blocs de compétences et d'expériences, j'ai d'abord réalisé une première version avec les flexbox, afin de positionner les blocs d'éléments facilement. Avec la `class .container`, j'ai pu appliquer mes paramètres de style et de positionnement à l'ensemble des blocs puis pour ensuite modifier chaque bloc séparément. Chaque bloc à les mêmes paramètres (un `border-radius`, une couleur, une ombre et un `padding`). Concernant Flexbox, j'ai eu des difficultés dans le positionnement des éléments de manières précises. Je souhaitais, avoir le bloc (about) qui soit positionné en première position et qui prennent toute la largeur, puis avoir en ligne 2 et 3 une rangée de deux blocs pour faire un (2x2)



Or, avec Flexbox, ce n'était pas possible car cela entraînait des décalages alors j'ai plutôt utilisé la position Grid pour être plus précis (avec les attributs : **grid-template** et **grid**).



Flexbox(v1)

grid (v2)

Concernant la position Flexbox, je l'ai utilisé pour du positionnement simple sur des petits éléments comme la partie contact, le positionnement des icônes de réseaux sociaux et le responsive avec (media query).

D. Responsive (IphoneX)

Mode portrait



Mode paysage



4. Ressenti sur le domaine

J'éprouve beaucoup de plaisir à utiliser le HTML et CSS. Concernant le CSS, je manque d'automatismes dans CSS et de vision globale. Je mérite à mieux utiliser les **class** ou **id** uniques pour y intégrer des fonctions généralistes car je me suis rendu compte que j'applique souvent les mêmes fonctions pour différentes **class**, ce qui me fait perdre du temps.