

Créer sa landing page de A à Z (#1)

Objectifs pédagogiques :

- Comprendre ce qu'est une landing page et connaître des exemples
- Savoir comment organiser ses fichiers de code
- Connaître le rôle de la balise <head> et son contenu
- Savoir mettre en forme du texte
- Savoir comment utiliser les couleurs en CSS
- Comprendre ce qu'est GitHub et à quoi cela sert

Livrable :

- Une version de base de la landing page, avec un titre, du texte, des couleurs, une balise <head> complète ;
- Un dépôt sur Github avec le code dessus ;

Plan de travail :

#	Challenges	Durée conseillée (minutes)
1	C'est quoi une landing page ? Introduction sur les landing pages	20
2	Préparons nos fichiers et Hello World Ou comment démarrer un projet web	20
3	La balise <head>	20
4	Une page de base en HTML/CSS Codons !	25
5	Les couleurs Comment mettre des couleurs sur un site ?	15
6	Introduction au code versioning Très important :)	60
	<i>total</i>	160



Travail à faire pour la prochaine fois : voir à la fin du document.

1. C'est quoi une landing page ?

Ce qu'il faut faire

Allez voir quelques exemples de landing page :

- <https://www.squarespace.com/>
- <http://framesforsketch.com/>
- <https://evernote.com/>
- <https://runcloud.io/>
- <https://www.apple.com/iphone-x/>
- <https://flowmapp.com/>
- <https://www.mozilla.org/en-US/firefox/quantum/>

Et travaillez sur ces questions :

- Qu'est-ce qu'une Landing Page ? Quel est son but ?
- Qu'est-ce que ces landing page ont en commun ?
- Repérez des éléments intéressants dans ces pages que vous aimeriez refaire.

2. Préparons nos fichiers et Hello World

Ce qu'il faut faire

Dans cette partie nous allons créer et organiser nos fichiers de code de manière à débiter n'importe quel projet de site web.

	Etape	Description
1	Où ranger ses projets de sites web sur son ordinateur ?	<p>La première chose à savoir est dans quel dossier vous allez placer votre site web sur votre ordinateur. Il faut savoir deux choses :</p> <ul style="list-style-type: none">- Si votre site web ne contient pas de PHP, alors il peut être lu directement par votre navigateur (dans Chrome par exemple en faisant simplement Fichier -> Ouvrir le fichier). Dans ce cas-là, la localisation de votre fichier sur votre ordinateur n'importe peu.- Si votre site web contient du code PHP, votre navigateur ne saura pas le lire et l'ouvrir directement. Il faudra passer par un serveur local, Mamp, qui interprètera le code PHP. Dans ce cas-là, il faut placer votre code dans un dossier bien spécifique, appeler htdocs. <p>Pour ne pas que vous ayez trop de dossiers de travail différents, on vous conseille de placer tous vos futurs projets à un seul et même endroit, en l'occurrence le dossier htdocs.</p>
2	1 dossier = 1 projet	<p>On se trouve donc désormais dans le dossier htdocs. Pour ne pas mélanger les fichiers de code de plusieurs projets différents, on va ranger un peu tout ça et créer un dossier pour notre projet.</p> <p>On peut appeler ce dossier landing_page par exemple. Attention, évitez de mettre des accents, caractères spéciaux (autres que - et _) et espaces dans le nom de votre dossier.</p> <p>On devrait donc avoir une arborescence comme cela :</p> <pre>--htdocs -- landing_page -- autre_projet_1 -- autre_projet_2 -- ...</pre>
3	Les fichiers de base	<p>On peut donc maintenant aller dans le dossier de notre projet. Nous nous trouvons donc dans htdocs/landing_page.</p> <p>Au commencement de tout site web, il y a deux fichiers : index.html et style.css. Pourquoi ces noms ? Peut-on choisir ? Ce sera votre travail pour la prochaine fois de le découvrir ;)</p> <p>Comment créer ces fichiers ? La manière la plus propre de créer ces fichiers est de passer par Atom :</p> <ol style="list-style-type: none">1. Lancez Atom2. Faites Fichier -> Ouvrir le dossier afin d'ouvrir notre dossier landing_page.3. Dans la colonne de gauche, faites Clic droit -> Nouveau fichier et donnez-lui un nom.

4	Le code minimal du index.html	<p>Vous avez donc vos deux fichiers index.html et style.css de créés mais qui sont encore vides. C'est normal ;)</p> <p>Alors que le fichier style.css peut pour le moment rester vide, le fichier index.html nécessite un code minimal, qui correspond à la structure de base de tout site web.</p> <p>Ce code est disponible ici. Copiez-collez le dans votre index.html ;) Si ce code n'est pas clair, faites des recherches sur ce que chaque terme/ balise signifie.</p> <p>Voilà, notre code de base est prêt !</p>
5	Hello World	<p>Quand les développeurs commencent un projet, ils démarrent quasiment tout le temps avec un « Hello World », un petit « coucou » dans le code permettant de s'assurer que tout marche bien.</p> <p>Ecrivez simplement « Hello World » dans votre index.html, mais attention à l'écrire au bon endroit ! ;) Vérifiez ensuite que cela s'affiche bien dans votre navigateur et vérifiez avec le mentor que le code est juste.</p>

3. La balise <head>

Ce qu'il faut faire

La balise <head> contient tous les paramètres et informations pour que votre page s'affiche de manière optimale. Par exemple, c'est dans cette balise qu'on dira où se trouve notre feuille de style CSS, notre script JavaScript (s'il y en a un), où l'on précisera les mots clefs pour optimiser le SEO, où on écrira le titre de la page (qui s'affiche dans l'onglet du navigateur) etc.

Contrairement à ce qui se trouve dans la balise <body>, tout le contenu de <head> ne s'affichera pas sur la page. Ce sont des *métadonnées*.

	<i>Etape</i>	<i>Description</i>
1	Qu'y-a-t-il dans l'en-tête ? Métadonnées en HTML	Lecture du très bon article de Mozilla à ce sujet, <u>ici</u> .
2	Relier le fichier de style CSS au HTML	A vous de jouer ;)

4. Une page de base en HTML/CSS

Ce qu'il faut faire

Allons plus loin que le Hello World ;) On vous donne 25 minutes pour personnaliser votre page !

Ici, on ne vous dit pas quelles balises utiliser, c'est à vous de chercher avec Google ;)

	<i>Etape</i>	<i>Description</i>
1	Créons un titre	<p>Commencez déjà par écrire un titre dans la page. Attention on parle bien du contenu de la page ici, pas le titre global de la page (qui lui s'écrit avec la balise <title>).</p> <p>Ensuite, comment fait-on un sous-titre ? Et un sous-sous-titre ? Et un sous-sous-sous-titre ? ;)</p> <p>Par défaut, cette balise HTML a du style : en gras, texte assez grand etc. C'est un style par défaut donné par votre navigateur. Essayez de changer un peu ça en CSS, en donnant à votre titre la taille que vous voulez.</p>
2	Créons un paragraphe	<p>Idem, créez un paragraphe en HTML et modifiez le style avec CSS ! :) N'hésitez pas à aller voir <u>de ce côté</u> pour essayer plusieurs propriétés de style.</p>

5. Les couleurs

Ce qu'il faut faire

Il y a de nombreuses manières de coder des couleurs en CSS. On va voir les plus courantes ici ;)

Pour rappel, on peut utiliser les couleurs avec - entre autres - deux de ces propriétés CSS : **color** et **background-color**.

	Etape	Description
1	Les noms de couleurs	<p>Par défaut, on peut écrire une bonne centaine de couleurs juste avec le nom anglais. Déjà pour les couleurs de base : red, pink, orange, yellow, purple, green, blue, brown, white, gray.</p> <p>Essayez de mettre en couleur de fond sur votre site une de ces couleurs de base ;)</p> <p>Au-là de ces 10 couleurs de base, on peut aussi utiliser des noms plus poussés, comme indianred, gold, lavender etc. La liste complète est ici.</p>
2	Les codes hexadécimaux	<p>Si on veut être encore plus précis par rapport aux couleurs, on peut s'aider de ce qu'on appelle un Color Picker : un outil qui pour une couleur donnée nous renvoie un code de 6 caractères commençant par un #. Ce code est appelé hexadécimal. Par exemple, le code #FFFFFF permet d'encoder la couleur blanche, qui d'ailleurs peut aussi s'écrire avec le nom white.</p> <p>Exemple d'un color picker ici.</p> <p>Choisissez une couleur, copiez le code hexadécimal et ajoutez-le dans votre code CSS ;)</p>
3	Les code RGB	<p>On peut aussi écrire les couleurs en suivant la norme RGB. Chaque couleur est alors décrite d'après les trois couleurs primaires : Red, Green et Blue.</p> <p>Par exemple, le code rgb(149, 102, 92) permet de coder une couleur marron-gris.</p> <p>A vous d'essayer ! Toujours avec le Color Picker, copiez les valeurs R, G et B et renseignez les dans votre code CSS en suivant la syntaxe : rgb(R, G, B).</p>
4	L'opacité d'une couleur	<p>L'avantage d'utiliser les couleurs en RGB, c'est qu'on peut rajouter un 4ème paramètre, appelé alpha, qui peut régler l'opacité de la couleur. Par exemple, si je veux que ma couleur marron-gris soit un peu transparente, je peux écrire : rgba(149, 102, 92, 0.8)</p> <p>Ici, 0.8 signifie 80%. Ma couleur sera visible à 80%. Si je mets le alpha à 0, ma couleur sera totalement transparente (invisible), et si je passe à 1, elle sera totalement visible.</p> <p>A vous d'essayer ! Jouer avec l'opacité d'une couleur ! ;)</p>

6. Versioner son code avec Git

Ce qu'il faut faire

Tout le monde fera cette partie en même temps avec le mentor.

Note

On reviendra pleinement sur Git / Github dans un cours ultérieur. Donc si ce n'est pas encore très clair pour vous, ce n'est pas grave (et c'est normal). Essayez malgré tout de comprendre le principe et l'utilisation de base.

Travail à faire pour la prochaine fois

Durée estimée : 1h-1h30
Ce travail est susceptible d'être noté.

1. Qu'est-ce qu'un code propre ?

Faites des recherches notamment sur :

- Commenter son code
- Indenter son code
- Bien nommer ses fichiers
- Bien nommer ses classes

Livrable :

- Vos notes de recherche ;
- La mise en application sur votre code ;
- Des questions seront posées à l'oral au début du prochain cours

2. Pensez à votre design

Choisissez un thème pour votre landing page parmi les deux suggérés :

- **Thème 1 - Service** : Plateforme de streaming pour des séries TV. La landing page s'adresse à
- **Thème 2 - Produit** : Nouveau casque de Réalité Virtuelle.

Pensez aussi au design de votre site, de manière schématique. Réalisez une maquette simple, où l'on visualise la position des principaux blocs.

Livrable :

- Une maquette **simple**, sur papier ou via un logiciel (au choix), où l'on visualise la position et le rôle des principaux blocs de votre page.

