

Cours 3
Déblocage web

Les titres marqués d'un ↷ peuvent être sautés.

Révisions web • Sommaire



1. Un site web, c’est quoi ?
2. Avec quels outils on développe?
3. Notions à connaître
4. Documentation

1 · Un site web, c’est quoi ?

Définition en termes de design

Un site web est un type de média interactif accessible par le réseau mondial Internet qui permet de montrer de l'information sur l'écran d'un ordinateur, d'une tablette ou d'un smartphone et de naviguer entre les différentes pages via des hyperliens.



UX · User eXperience



Un **dossier d'UX research** contient les étapes suivantes :

1. **Interviews sur le terrain**
   → avec de vraies personnes utilisatrices potentielles.

2. **Élaboration de *persona***
   → personnages fictifs représentant la population interviewée.

3. **Produire au moins 1 scénario UX** (écrit, schéma UML, BD…)
   → mise en scène d'un *persona* lors de son utilisation et résultat de son expérience sur le site, améliorations par rapport à avant).

UI · User Interface



Les logiciels suivants sont les plus plébiscités du moment pour **réaliser des maquettes de site web** avec interactions :

1. Figma
2. Sketch
3. Adobe XD

Définition en termes de développement

Un site web est un dossier sur un serveur – *on appelle ce dossier de base la “racine”, ou “root” en anglais* – avec au minimum un fichier HTML (Hyper Text Markup Language) nommé `index.html` qui sera chargé par le navigateur web lorsqu’on consulte le site.



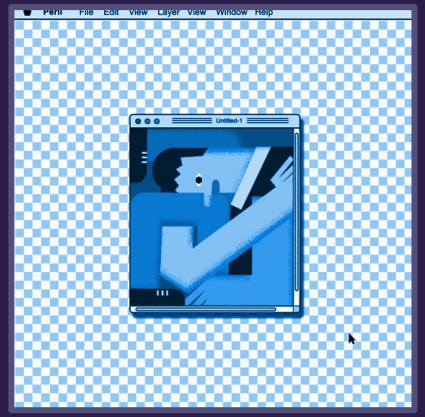
↷ Dans un dossier de site web…



On y trouve généralement d'autres pages, autrement dit d'autres **fichiers html** nommés librement, par exemple `contact.html` (*pas d'espaces, pas de caractères spéciaux ou accentués, pas de majuscules si possible*), entre lesquels on peut **naviguer grâce à des liens** (balise `<a>`).

C'est la structure et le contenu du site.

↷ Dans un dossier de site web...



Au moins un fichier CSS (Cascading Style Sheet = feuille de style en cascade) par exemple `style.css`, qui permettent de gérer l'apparence et la mise en forme des pages ainsi que les différents cas de responsive en fonction des devices (tailles d'écran, tactile...).

C'est la description de l'apparence du site.

↷ Dans un dossier de site web…



Éventuellement un ou plusieurs fichiers Javascript, par exemple `script.js`, qui permettent de créer des interactions complexes ou des animations programmatiques (*exemple ci-contre où l'animation doit être différente en fonction du chiffre à afficher, qui doit être calculé au préalable*).

C'est la couche de vernis (facultative mais appréciable) qui permet de créer une expérience utilisateur + élaborée.

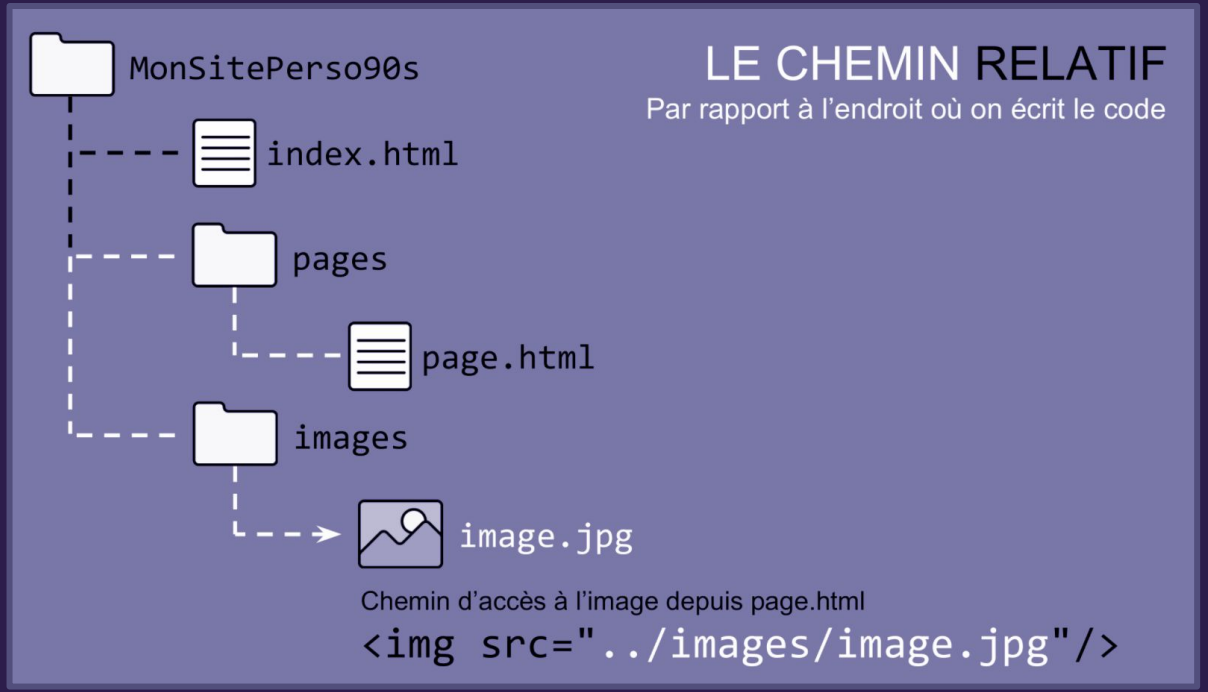
↷ Dans un dossier de site web…



Des **fichiers médias** qui pourront être intégrés dans les pages, sous forme de contenu (directement dans le HTML) ou d'arrière-plan (uniquement les images, via la CSS) :

- **images** (*fichiers .jpg, .png, .gif, .svg, etc.*)
  — balise auto-fermante `<img/>`,
- **vidéos** (*fichiers .mp4, .webm, etc.*) ou **sons** (*fichiers .mp3, .ogg*)
  — balises `<video>` et `<audio>` ou intégration avec `<embed>`,
- atteindre tout **autre type de fichier** (*par exemple des fichiers .pdf*) qui peuvent être affichés par le navigateur ou téléchargés
  — balise `<a>`.

↷ Chemin de fichier relatif



↷ Chemin de fichier absolu



2 • Avec quels outils on développe ?

↷ Contexte

Tous les fichiers informatiques sont composés de 0 et de 1, langage que l'on appelle binaire. Les **langages informatiques** sont qualifiés de bas niveau s'ils sont plutôt proches du fonctionnement de la machine, ou de haut niveau s'ils sont plutôt proches du langage humain. Le binaire est au niveau le plus bas qui existe!

Afin de le rendre compréhensible pour l'humain, le binaire est traduit dans des langages de plus haut niveau qui permettent d'**écrire du code facilement** et de travailler avec. Un langage de programmation sert donc à communiquer avec l'ordinateur! Qui dit langage dit vocabulaire, grammaire, syntaxe, orthographe, etc.



Workflow de développement

À l'école, **nous utilisons** Visual Studio Code pour éditer le code.

Notre outil principal est donc un logiciel de traitement de texte spécialement conçu pour travailler dans les différents langages de programmation, qu'on appelle **éditeur de code**.

Pour **créer l'arborescence du site**, on peut simplement utiliser le système de dossiers de son ordinateur.

3 • Notions à connaître ?

Fondamentaux

🧱 HTML

Fichier .html (ou **.php**)

- Le squelette HTML5 (aka *boilerplate*)
- Balise (aka *tag*) normale / auto-fermante
- Attributs
- Classes et ID
- Liens `<a>`
- URL et chemin de fichier absolu / relatif
- L'indentation
- *Facultatif: Bootstrap*

🎨 CSS

Fichier .css (ou balise `<style>` dans le HTML)

- Sélecteur
- Propriété
- Valeur
- L'**inspecteur**
- Le Box Model
- Display
- Position
- Flexbox (guide ultime)
- Grid (guide ultime)
- Media query
- *Facultatif: SASS*

⚙️ JS

Fichier .js (ou balise `<script>` dans le HTML)

- Variable
- Condition
- Opérateur logique
- Boucle
- La **console**
- Fonction
- Événement
- *Facultatif: la POO (concept clé avancé mais bon à savoir)*
- *Facultatif: jQuery*

Le débug

🦟 **L'inspecteur** de votre navigateur permet d'analyser votre site (*si vous êtes sur Safari, cette fonctionnalité doit être activée d'abord dans les paramètres*) comme suit :

1. Pour ouvrir cette fenêtre, il faut clic droit sur l'élément visé et sélectionner "*Inspecter l'élément*".
2. L'inspecteur possède 2 volets et plusieurs onglets pour analyser les différents langages utilisés par votre site web :
    a. Les volets de départ affichent l'onglet "Éléments" qui montre le code HTML et l'onglet "Styles" qui montre le code CSS correspondant à la sélection.
    b. L'onglet pour le JS s'appelle **la console** et a un usage spécifique.
3. Il est possible de modifier le code dans l'inspecteur pour tester des idées et rechercher les solutions aux bugs. Ces modifications sont temporaires: elles ne modifient pas le fichier de code original et seront réinitialisées en rechargeant la page.
4. Quand vous atteignez un résultat qui vous convient, il faut donc recopier les lignes de code qui ont fonctionné dans votre fichier source. Enregistrer puis recharger la page pour voir le résultat réel.

💡 *Astuce : on peut tester facilement des idées plus élaborées sur codepen.io et les sauvegarder en ligne pour plus tard si on crée un compte.*

↷ Avancé

🛰️ L'infrastructure

- Localhost 💻
- Serveur Apache ou Nginx[1]
- Nom de domaine 🌐
- Hébergement 🏠

📫 Registrar et hébergeur

Ces entreprises[2] fournissent un service en ligne qui permet de louer un nom de domaine, un espace de stockage pour son site, et même des adresses e-mail associées :

- Infomaniak (CH)
- OVH (FR)
- Amazon AWS (US)
- Microsoft Azure (US)
- …

🔑 Ce qu'il faut pour…

- Se connecter en **FTP / SFTP** :
    1. Adresse du serveur
    2. Utilisateur FTP
    3. Mot de passe de l'utilisateur

👉 Logiciel : Cyberduck, WinSCP, Fetch…
Ex : mettre en ligne / màj un site en ligne.

- Se connecter à une **base de données** MySQL :
    1. Adresse du serveur
    2. Nom de la base de données
    3. Utilisateur MySQL
    4. Mot de passe de l'utilisateur

👉 Où : fichiers .php, formulaire web…
Ex : installation d'un site Wordpress.

↷ Avancé

💽 Les bases de données

Les langages pour communiquer avec une base de données sont le **PHP** et le **MySQL**.

*Les CMS utilisent ces langages pour fonctionner. La plupart du temps dans votre métier, vous n'aurez donc pas directement à coder dans ces langages, mais il est utile de savoir qu'**ils ne fonctionnent que lorsqu'ils passent par des serveurs qui les supportent**.*

- Pour pouvoir **travailler en local** (par exemple sur un projet de site Wordpress), contrairement aux 3 langages principaux qui peuvent être interprétés directement par votre navigateur, les fichiers .php et les bases de données ont besoin que l'on installe **MAMP** pour Mac, WAMP pour Windows, ou LAMP pour Linux.
- Il faut placer les fichiers de son site dans le **dossier qui sera "transformé en serveur"** par le logiciel. Chercher où se situe ce dossier dans les paramètres de MAMP.
- Ensuite, on peut afficher son site en ouvrant son navigateur et en accédant à localhost (habituellement, écrire **localhost:8888** dans la barre d'URL permet d'y accéder, sinon, chercher l'info dans les paramètres de MAMP).

🏗 Quelques CMS

Wordpress, Shopify, Prestashop, Wix, Squarespace, Cargo, Joomla, Drupal, Magento, Typo3…

Même si un site est géré par un CMS, il est toujours constitué des langages cités précédemment. Sauf que ce n'est pas vous qui l'écrivez!

4 • Documentation

Incontournables

📕 Documentation

- Encyclopédie officielle du W3C avec tutoriels et exemples toujours à jour: w3schools.com
- Dictionnaire des langages web: https://devdocs.io
- Quelqu'un a forcément déjà rencontré le même problème que vous sur Stack Overflow
- Déterminez si une nouvelle notion CSS est utilisable à l'aide de Can I use?

🧠 Tutos en ligne

- Alsacréations: tutoriels gratuits, en Français, avec exemples. Actus et bonnes pratiques.
- Openclassrooms (*ancien Site du Zéro pour ceux qui connaissent!*): tutoriels gratuits en Français, avec exercices pratiques et plan de cours progressif.
- Supports du cours "Web! The bootcamp." de Simon Pinkas à télécharger.

Outils et astuces en ligne

🔗 **Liens utiles**

- Codepen.io pour bricoler des prototypes web, vite là, avant d'oublier son idée géniale!
- Le squelette HTML5 sur Alsacréations.
- Guide ultime Flexbox et Guide ultime Grid sur CSS Tricks.
- Générateur de dégradés de couleurs à copier-coller dans sa CSS.
- Amp-what.com: TOUS les caractères spéciaux, ASCII, Emoji, etc.
- On ne présente plus Fonts.google.com
- Loremipsum.io quand tu n'as pas le temps d'inventer un texte de remplissage marrant.
- Wordpress: plan de la hiérarchie du thème (pour bien comprendre le système de fallback entre les différents fichiers).
- Wordpress: cheatsheet ultime pour tout comprendre sur la construction de thème.

Bonus: comptes Instagram et Newsletters à suivre

💜 = les meilleurs !

📷 Code

1. 💜 baby_wolf_codes

📷 Tendances & astuces UX / UI

1. 💜 figma
2. 💜 thewilsonwings
3. ux.base
4. uiuxbunker
5. uxbrainy
6. ui_gradient
7. uix.osm
8. design.gems

📷 Designers partageurs d'expériences freelance et astuces crafty

1. 💜 made.by.james
2. 💜 zanderwhitehurst
3. 💜 thechrisdo
4. uiuxcreative
5. michael.filipiuk
6. _hadi.ai
7. sehbancreates
8. kalypsodesigns
9. the.ewk
10. harshil_designer

📩 Newsletters Design

1. 💜 UX Design Weekly (Kenny Chen)
2. 💜 Dribbble (s'inscrire puis cocher les cases Weekly Newsletters qui vous intéressent)
3. Google Design (newsletter ici)

📩 Newsletters Code / Tech

1. 💜 Smashing Magazine
2. Codepen Spark