

03. Construire et structurer son portfolio

Objectifs de la séance

- Savoir organiser l'information de manière claire et attractive.
- Développer une première version fonctionnelle.

1. Storytelling et hiérarchisation

Un bon portfolio n'est pas une simple liste de projets : c'est un **récit structuré** qui guide le lecteur.

Le storytelling consiste à présenter vos projets comme des **histoires brèves** avec un début (le problème), un déroulement (la solution) et une fin (l'impact).

Exemple de structure pour chaque projet :

- **Contexte** : quel besoin, pour qui ?
- **Rôle** : qu'avez-vous fait précisément ?
- **Technologies** : stack utilisée, outils clés.
- **Résultat** : démo, captures, repo, chiffres (performances, utilisateurs).

Hiérarchisez vos contenus :

- Mettez **3 projets majeurs** en avant, pas 10.
- Ajoutez éventuellement une section "autres projets" plus concise.
- Commencez par ce qui est le plus pertinent pour vos objectifs pro (ex. : un projet scolaire pour un stage, un projet OSS pour un poste de dev back-end).

2. Accessibilité et design responsive

Un portfolio doit être **lisible et inclusif**. L'accessibilité et le responsive ne sont pas des options.

Bonnes pratiques :

- Couleurs contrastées, taille de texte lisible.
- Navigation claire, titres structurés (`h1` , `h2` , etc.).
- Texte alternatif (`alt`) sur les images et captures.
- Responsive design : lisible sur ordinateur, tablette, smartphone.

Astuce : testez votre site avec les outils intégrés aux navigateurs (Chrome/Firefox DevTools) et avec un lecteur d'écran si possible.

3. Introduction aux outils

GitHub Pages

- Solution gratuite et simple pour héberger un portfolio statique.
- Compatible avec Jekyll, Hugo, ou un site HTML/CSS/JS classique.

Générateurs de site statique

- **Hugo**, **Gatsby**, **Next.js** (pour profils plus avancés).
- Avantage : rapidité, sécurité, bonne gestion du contenu.
- Inconvénient : nécessite un peu de configuration.

CMS Headless ou générateurs spécialisés

- Exemples : **Netlify CMS**, **Zivee**, **Contentful**.
- Avantage : interface simple, moins de code à gérer.
- Inconvénient : dépendance à un service externe.

4. Mini-atelier

1. Créez une **page d'accueil simple** pour votre portfolio.
2. Ajoutez un premier projet (titre, description, capture, lien GitHub).

3. Publiez-le via GitHub Pages (ou un équivalent).
4. Vérifiez que la page est accessible sur mobile et ordinateur.

5. Livrable de la séance

Une première version en ligne d'un portfolio minimal, contenant :

- Une brève présentation (bio ou tagline).
- Un projet mis en avant avec description et lien.
- Hébergement en ligne (GitHub Pages, Netlify, ou autre).

6. Pour aller plus loin

- *A List Apart – The Art of Storytelling in UX.*
- *Google Web.dev – Accessibility fundamentals.*
- *GitHub Pages Documentation* : déploiement rapide.
- *Netlify* et *Vercel* : alternatives modernes de déploiement.

7. Mini-FAQ

- **Dois-je coder mon portfolio à la main ?**

Pas forcément. L'important est la clarté et la mise à jour. Un générateur peut suffire.

- **Puis-je utiliser un template ?**

Oui, mais personnalisez-le. Les recruteurs voient immédiatement un template générique non adapté.

- **Un seul projet, ce n'est pas trop peu ?**

Non. Un projet bien expliqué vaut mieux qu'une dizaine mal présentés.