
Préparer son environnement de travail

Installer son environnement de travail IDE & extensions

Objectif : Créer un environnement de travail complet pour coder sereinement.



Installer son environnement de travail (1/2 journée)

Programme :

- Installation de VS Code (éditeur de code)
- Créer son premier fichier HTML depuis VS Code
- ESLint (aide à repérer les erreurs de code)
- Prettier (pour formater son code proprement)
- Live Server (prévisualisation en direct)
- Chrome Dev Tools (l'outil des dev sur Chrome)
- QCM
- **Bonus:** Installer NPM + Node + Webpack + Créer son fichier package.json

Matinée 18/03



Balsamiq Mockup

<https://www.afecdax.ovh/ressources/balsamiq/licence.txt>



Filezilla



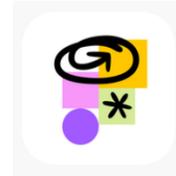
Figma



**Node JS
(option)**



VS Code



Figjam

Pas d'install nécessaire



Afec Dax :

<https://discord.gg/5PBF5Qe6RC>



Wamp



Gitbash

<https://git-scm.com/downloads>



Créer un compte sur Github.com

Réfléchir à votre projet fil rouge : notez des idées de ce que vous souhaitez développer comme projet dans l'année

Les différents IDE

C'est quoi un IDE ?

Un IDE (Environnement de Développement Intégré) c'est un logiciel qui regroupe tout ce dont on a besoin pour écrire, organiser et tester du code. C'est comme une boîte à outils spéciale pour les développeurs.

Ça permet de :

- D'écrire du code avec des couleurs (coloration syntaxique) pour mieux s'y retrouver.
- D'avoir des suggestions automatiques pour aller + vite.
- De repérer les erreurs avant même de tester.
- D'installer des extensions pour simplifier la vie (comme un correcteur de code ou un prévisualisateur HTML).

Différents IDE :

VS Code	PhpStorm
Phpstorm	9,90€ / mois
AI Code Editor	Gratuit & payant selon utilisation IA.
Sublime Text	Gratuit
Notepad++	Gratuit

Installation de VS Code

Pourquoi VS Code ?

- Gratuit.
- Fonctionne sur Windows, macOS et Linux.
- Léger (par rapport à d'autres IDE plus lourds).
- Grosse communauté, donc en cas de souci, il y a plein de solutions en ligne.
- Personnalisable avec des extensions

Installer VS Code

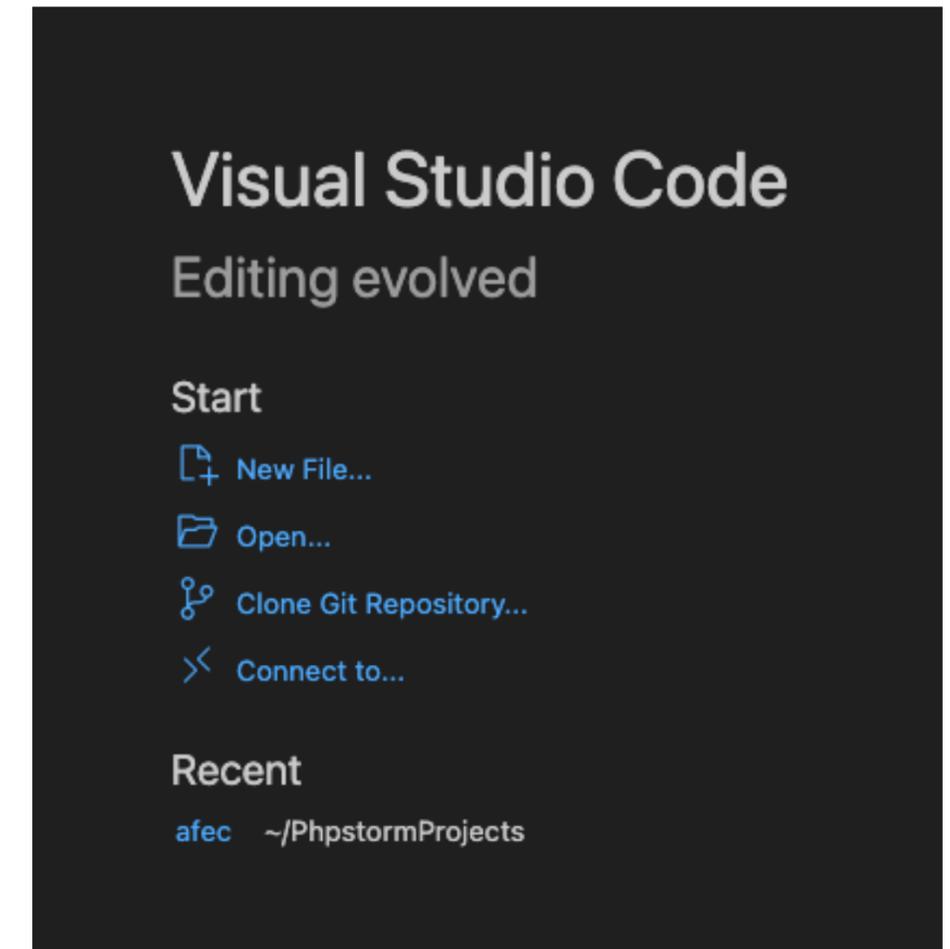
- Allez sur le site officiel : <https://code.visualstudio.com>
- Téléchargez la version adaptée à votre système (Windows, Mac ou Linux).
- Installez le fichier téléchargé.
- Suivez les étapes d'installation (vous pouvez laisser les options par défaut).
- Une fois installé, lancez VS Code.



Créer son premier fichier HTML

Créer son premier fichier depuis VS Code

1. Ouvrez VS Code.
2. Cliquez sur Start (ou directement sur l'icône de fichier en haut à gauche) puis sur Nouveau fichier (ou New File si l'interface est en anglais).
3. Nommez le fichier : **index.html**.
4. Une fenêtre va s'ouvrir pour enregistrer le fichier.
5. → Choisissez un emplacement (par exemple, le Bureau) et créez un nouveau dossier pour votre projet si besoin.
6. → Important : pour le nom du dossier, utilisez uniquement des lettres en minuscule, des chiffres, mais pas d'accents, d'espaces ou de caractères spéciaux.
7. Exemple de nom correct : **premier_projet** ou **exercice1**.
8. Cliquez sur Enregistrer.



Installation extension ESLint sur VS Code

ESLint est une extension qui vérifie la qualité de votre code JavaScript. Son but, c'est de vous éviter les erreurs bêtes et d'harmoniser votre style de code.

En clair, c'est un peu comme un correcteur orthographique, mais pour votre JavaScript.



Ce qu'ESLint permet :

- Il repère les erreurs de syntaxe (oubli de point-virgule, variables non déclarées, etc.).
- Il vous alerte sur les mauvaises pratiques (par exemple, utiliser une variable avant de l'avoir définie).
- Il vous aide à normaliser l'écriture de votre code, pour qu'il soit plus lisible et plus propre.

Installation

- Dans VS Code, ouvrez l'onglet Extensions (icône sur la barre de gauche ou raccourci). 
- Recherchez ESLint.
- Cliquez sur Installer.

ESLint ne fonctionne pas automatiquement :

il faudra par la suite le configurer dans chaque projet.

Installation extension Prettier sur VS Code

Prettier est un formateur de code qui s'occupe de la mise en page de vos fichiers : il indente correctement, aligne proprement les balises et ajoute les espaces là où il faut. Idéal pour apprendre à écrire du code lisible dès le début.



Ce que Prettier permet :

- Il rend le code plus lisible.
- Il est compatible avec HTML, CSS et JavaScript.
- Il s'intègre à ESLint
- Il fait gagner du temps
- Il rend le code compatible

Installation

1. Dans VS Code, ouvrez l'onglet Extensions (icône sur la barre de gauche ou raccourci). 
2. Recherchez Prettier.
3. Cliquez sur Installer.
4. (Optionnel) Dans les Paramètres (VS Code > Preferences > Settings, tu peux rechercher "Format On Save" et l'activer pour que Prettier reformate automatiquement chaque fois que tu sauvegardes

Installation extension Live Server sur VS Code

Live Server permet de lancer un petit serveur local qui affiche votre site dans le navigateur et rafraîchit automatiquement la page dès qu'on enregistre un fichier. Pour les débutants, c'est super pratique, car ça évite de devoir rouvrir la page à chaque modification.



Ce que Live Server permet :

- Aperçu en temps réel des changements.
- Facile à utiliser : un clic et ça marche.
- Parfait pour les projets HTML/CSS/JS simples.

Installation

1. Dans VS Code, ouvrez l'onglet Extensions (icône sur la barre de gauche ou raccourci). 
2. Recherchez Live Server.
3. Cliquez sur Installer.
4. Toujours Ouvrir un Dossier et non un fichier seul pour que Live Server fonctionne
5. Toujours S'assurer que "Go Live" est activé dans la barre de tâche du bas.

Chrome Dev Tools

Chrome DevTools est une suite d'outils intégrée à Google Chrome qui permet d'inspecter, modifier et déboguer les pages web en temps réel.

Ce que Chrome Dev Tools permet :

- Comprendre comment le navigateur interprète le HTML, CSS et JavaScript.
- Déboguer et corriger les erreurs.
- Tester des styles CSS en direct.
- Analyser les performances d'un site.
- Simuler des appareils mobiles.

Comment l'ouvrir ?

Raccourci clavier :

- Windows/Linux : F12 ou Ctrl + Shift + I
- Mac : Cmd + Option + I

Via le menu Chrome : Menu (:> "Plus d'outils" > "Outils de développement"



Chrome Dev Tools (suite)

Panneau "Elements" (HTML & CSS en live)

- Inspecter et modifier le HTML/CSS en direct.
- Tester des styles CSS sans modifier le code source.
- Ajouter/supprimer des classes et voir les changements en temps réel.

Exercice 1:

- Ouvrir DevTools sur une page web.
- Modifier un texte (<h1>, <p>) directement.
- Modifier une couleur de fond via CSS.
- Ajouter une nouvelle classe CSS et tester son effet.

Panneau "Console" (Débogage JavaScript)

- Afficher les erreurs JavaScript.
- Tester du code JavaScript en direct.
- Interagir avec le DOM via JavaScript.

Exercice 2 :

- Ouvrir la console avec F12 > Console.
- Exécuter du JavaScript simple :
`document.body.style.background = "lightblue";`
- Repérer et comprendre une erreur dans la console.

Chrome Dev Tools (suite)

Panneau "Network" (Suivi des requêtes HTTP)

- Voir quelles ressources sont chargées (images, scripts, CSS).
- Analyser le temps de chargement.
- Comprendre les requêtes API.

Exercice 3 :

- Charger une page et ouvrir Network.
- Filtrer par type (JS, CSS, images...).
- Observer les requêtes API sur un site dynamique.

Panneau "Performance" (Optimisation du chargement)

- Mesurer le temps de chargement du site.
- Détecter les ralentissements.
- Analyser les performances JavaScript et CSS.

Exercice 4 :

- Ouvrir Performance et enregistrer un profil.
- Analyser les résultats.

Chrome Dev Tools (suite)

Mode “Device Toolbar” (Simulation mobile)

- Tester la réactivité (responsive design).
- Simuler des appareils mobiles.
- Simuler une connexion lente (3G, 4G).

Exercice 5 :

- Activer le mode “Device Toolbar” (Ctrl + Shift + M / Cmd + Shift + M).
- Sélectionner un modèle d'appareil (iPhone, Galaxy...).
- Tester les tailles d'écran et voir l'adaptation CSS.

C'est quoi NODE.JS ?

NODE.js

Node.js est un environnement d'exécution JavaScript qui permet d'exécuter du code JavaScript côté serveur, en dehors du navigateur.

Avec Node.js, vous pouvez :

- Exécuter du code JavaScript côté serveur.
- Gérer des requêtes HTTP pour créer des API et des applications web.
- Lire, écrire et manipuler des fichiers sur le système.
- Gérer des tâches asynchrones comme les bases de données et les websockets.

Installation

Télécharger et installer NODE.js sur le site officiel nodejs.org

Vérifiez que tout fonctionne en lançant les lignes de commandes :

```
node -v
```

C'est quoi NPM ?

NPM

NPM (Node Package Manager) est un gestionnaire de paquets.

En gros, c'est une bibliothèque géante où sont stockées des milliers d'outils, de librairies et de modules prêts à l'emploi pour faciliter le développement.

Avec NPM, vous pouvez :

- Installer facilement des bibliothèques JavaScript (comme Webpack, Lodash, ESLint, etc.).
- Mettre à jour vos outils.
- Gérer toutes les dépendances de vos projets.

Installation

Téléchargez la version LTS (recommandée).

Installez Node.js en suivant les étapes.

Vérifiez que tout fonctionne en lançant les lignes de commandes :

```
npm -v
```

C'est quoi webpack ?

Webpack, c'est un outil qui assemble tous vos fichiers (JavaScript, CSS, images...) en un seul "bundle" optimisé.



Concrètement, Webpack :

- Combine vos fichiers en un seul fichier final.
- Optimise les performances (minification, gestion des images...).
- Gère automatiquement les imports entre vos fichiers JS ou CSS.
- Peut aussi actualiser automatiquement la page à chaque modification (webpack-dev-server).

Installation et configuration de Webpack

1) Initialiser un projet Node.js

- Assurez-vous que Node.js et npm sont installés (node -v et npm -v).
- Dans un dossier de projet, ouvrez un terminal et tapez :

```
npm init -y
```

Cela crée un fichier package.json pour gérer les dépendances du projet.

2) Installer Webpack et Webpack CLI

Installez Webpack en tant que dépendance de développement s'il n'existe pas :

```
npm install webpack webpack-cli --save-dev
```

3) Créer et configurer Webpack

- Créez un fichier webpack.config.js à la racine du projet.
- Ajoutez la configuration de base en remplaçant le contenu de webpack.config.js par celui ci :

```
https://github.com/nextader/afec/blob/main/webpack.config.js
```

4) Définir un script "build" dans package.json

- Ouvrez package.json et copiez le contenu du fichier package.json présent ici :

```
https://github.com/nextader/afec/blob/main/package.json
```

Installation et configuration de Webpack (suite)

5) Lancez le script :

`npm run build`

La commande **npm run build** est utilisée pour exécuter un script défini dans le fichier package.json sous "scripts".
(fonctionne que si un script "build" est défini dans package.json, sinon elle ne fait rien.)

Ressources

Webpack Starter Kit Afec

<https://github.com/nextader/afec>

VS Code

<https://code.visualstudio.com>