

# Programme formation DWWM

Programme avec les **8 blocs pédagogiques** du référentiel RNCP DWWM RNCP37674, **réorganisés dans l'ordre le plus logique pour les apprenants**, avec **un aperçu de leur contenu**.

---

## Bloc 1 – Installer et configurer son environnement de travail

 *Objectif : Préparer le poste de travail pour développer efficacement.*


- Installation des outils indispensables : éditeur de code (VS Code), terminal, navigateur, extensions.
  - Configuration de Git et GitHub pour la gestion de version.
  - Compréhension des chemins relatifs/absolus, du système de fichiers et de la structure d'un projet web.
  - Installation de Node.js, gestion de dépendances avec npm.
  - Introduction à la ligne de commande.
- 

## Bloc 2 – Maquetter des interfaces utilisateur web ou web mobile

 *Objectif : Concevoir l'apparence et l'organisation des interfaces avant leur codage.*

- Bases du design d'interface : UX/UI, ergonomie, responsive design.
  - Outils de maquettage (ex : Figma).
  - Création de wireframes et prototypes.
  - Bonnes pratiques d'accessibilité numérique (RGAA).
  - Définition de personas et parcours utilisateurs.
- 

## Bloc 3 – Réaliser des interfaces utilisateur statiques web ou web mobile

 *Objectif : Transformer une maquette en site statique fonctionnel.*

- Langages HTML5 et CSS3 : balisage sémantique, mise en page, media queries.
- Grilles CSS, Flexbox, Grid Layout.
- Utilisation d'un framework CSS (Bootstrap ou Tailwind CSS).
- Mise en place du responsive design.
- Bonnes pratiques de code (indentation, noms de classes, conventions BEM...).

---

## **Bloc 4 – Développer la partie dynamique des interfaces utilisateur web ou web mobile**

 *Objectif : Ajouter de l'interactivité côté navigateur.*

- Langage JavaScript (ES6+), logique conditionnelle, boucles, fonctions.
- Manipulation du DOM (Document Object Model).
- Événements utilisateur (clic, clavier, formulaires...).
- Requêtes HTTP avec fetch ou axios (APIs REST).
- Introduction à un framework JS (React.js ou Vue.js).

---

## **Bloc 5 – Mettre en place une base de données relationnelle**

 *Objectif : Structurer et stocker des données de manière fiable.*

- Modélisation conceptuelle (MLD, MCD).
- Création de schémas SQL (tables, types de données, clés).
- Introduction à un SGBD (MySQL, PostgreSQL...).
- Commandes SQL de base : SELECT, INSERT, UPDATE, DELETE.
- Notions de normalisation et contraintes d'intégrité.

---


## **Bloc 6 – Développer des composants d'accès aux données SQL et NoSQL**

 *Objectif : Relier le back-end aux bases de données pour y lire et écrire des données.*

- Connexion à une base de données via un langage serveur (ex : Node.js avec Sequelize, MongoDB avec Mongoose).
- ORM vs requêtes brutes.
- Introduction à NoSQL (MongoDB) : collections, documents, schéma flexible.
- Requêtes filtrées, jointures, pagination.
- Sécurisation des requêtes (ex : injection SQL).

---


## **Bloc 7 – Développer des composants métier côté serveur**

 *Objectif : Construire la logique applicative et les API.*

- Introduction à l'architecture client/serveur et au modèle MVC.
- Création d'un serveur web (Express.js, PHP).
- Routage, middlewares, validation des données.








- API RESTful : endpoints, statuts HTTP, routes CRUD.
  - Sécurité : gestion des erreurs, authentification (JWT), CORS.
- 

## **Bloc 8 – Documenter le déploiement d’une application dynamique web ou web mobile**

 *Objectif : Finaliser et publier une application professionnelle.*

- Versionnage Git et GitHub, branches, merge, pull requests.
  - Création d’un fichier README technique.
  - Utilisation de Postman ou Swagger pour documenter une API.
  - Déploiement front (Netlify, Vercel) et back-end (Render, Railway, VPS).
  - Introduction à Docker (containers simples).
- 

### **Résumé visuel de l’ordre logique**

1. Environnement de travail 
2. Maquettage 
3. Interfaces statiques 
4. Interfaces dynamiques 
5. Base de données relationnelle 
6. Accès aux données SQL/NoSQL 
7. Composants métier serveur 
8. Déploiement & documentation 