

# Plan de Formation – 120h (HTML, CSS, JS, Angular, Git)

## Module 1 : Introduction & Outils (6h)

- Présentation du web (client/serveur, HTTP, navigateurs).
  - Installation des outils : VS Code, Node.js, Git, Angular CLI.
  - Introduction à Git et GitHub :
    - `git init`, `clone`, `add`, `commit`, `push`, `pull`.
    - Organisation en branches.
- 

## Module 2 : HTML (12h)

- Structure de base d'une page HTML.
  - Les balises sémantiques (`header`, `nav`, `main`, `footer`).
  - Les formulaires et contrôles (`input`, `select`, `textarea`).
  - Liens, images, tableaux, listes.
  - Accessibilité et bonnes pratiques.
  - Mini-projet : Page web statique ( portfolio simple).
- 

## Module 3 : CSS (15h)

- Sélecteurs et spécificité.

- Box model, positionnement, display (flex, grid).
  - Couleurs, typographies, variables CSS.
  - Animations et transitions.
  - Responsive design (media queries).
  - Framework CSS (Bootstrap ou Tailwind en option).
  - Mini-projet : css pour le portfolio simple.
- 

## **Module 4 : JavaScript (30h)**

### **Fondamentaux (15h)**

- Variables, types, opérateurs.
- Conditions, boucles.
- Fonctions, portée, closures.
- Tableaux, objets.
- DOM : sélection, manipulation, événements.
- LocalStorage / SessionStorage.

### **Avancé (15h)**

- ES6+ (arrow functions, destructuring, modules).
- Promises, async/await.
- Fetch API (consommer une API REST).
- Classes & POO en JS.
- Notions de tests unitaires (Jest ou Jasmine).

- Mini-projet : To-do list avec persistance locale.
  - Mini-projet : ajouter animation a la portfolio.
- 

## **Module 5 : Git (6h)**

- Workflow complet avec GitHub.
  - Branches, merges, pull requests.
  - Gestion des conflits.
  - Gitflow (feature, develop, main).
  - Bonnes pratiques pour un projet collaboratif.
- 

## **Module 6 : Angular (36h)**

### **Introduction & Bases (12h)**

- Installation Angular CLI.
- Structure d'un projet Angular.
- Composants, templates, databinding.
- Directives (\*ngIf, \*ngFor, attributs).
- Services et injection de dépendances.

### **Avancé (12h)**

- Routing et navigation.
- Formulaire (Template-driven et Reactive).

- Communication entre composants (Input, Output).
- Pipes, lifecycle hooks.

## **Intégration avec API & CRUD (12h)**

- Consommer une API REST avec HttpClient.
  - Opérateurs RxJS (map, filter, subscribe).
  - CRUD complet avec backend simulé (json-server).
  - Authentification de base (login/logout).
- 

## **Module 7 : Projet Final – Gestion de Parking (15h)**

### **Objectifs pédagogiques**

- Mettre en pratique **HTML, CSS, JS, Angular, Git**.
- Réaliser une **application CRUD complète**.
- Gérer une logique métier réaliste (réservations, disponibilité, facturation).
- Déployer le projet en ligne (GitHub Pages).

## **Fonctionnalités à développer**

### **1. Gestion des utilisateurs**

- Authentification simple (login avec email/mot de passe).
- Deux rôles :
  - **Admin** : gère le parking.
  - **Client** : réserve une place.

## 2. Gestion des places de parking

- CRUD des places : créer, éditer, supprimer.
- Attributs d'une place : numéro, état (libre/occupée/réservée), type (standard, handicapé, premium).
- Vue en **grille visuelle** des places.

## 3. Gestion des véhicules

- Enregistrement d'un véhicule (plaque, marque, modèle).
- Association d'un véhicule à un utilisateur.

## 4. Réservations

- Réserver une place de parking pour un créneau horaire.
- Vérification de disponibilité.
- Annulation/modification de réservation.

## 5. Facturation / Paiement (simplifié)

- Génération d'un ticket de stationnement (temps + coût).
- Exemple de calcul : 2dt de l'heure, forfait max 10dt/jour.

## 6. Tableau de bord admin

- Statistiques :
  - Taux d'occupation.
  - Revenus générés.
  - Nombre de réservations.
- Graphiques avec **Chart.js** ou **ngx-charts**.

## 7. Bonus

- Géolocalisation des parkings (Google Maps API).
  - Notifications (par email ou affichage dans l'app).
- 

### Stack technique

- **Frontend** : Angular 16+, css.
  - **Backend (simulé)** : json-server.
  - **Gestion du code** : Git & GitHub.
  - **Déploiement** : GitHub Pages.
- 

### Organisation du projet (15h)

#### 1. Setup (2h) :

- Créer le repo Git.
- Initialiser Angular .

#### 2. Gestion des places (3h) :

- CRUD complet des places.
- Affichage en tableau et grille.

#### 3. Gestion véhicules & utilisateurs (2h).

#### 4. Réservations (3h) :

- Système de réservation avec validation.

**5. Facturation (2h) :**

- Calcul des coûts.
- Génération de tickets.

**6. Dashboard admin (2h) :**

- Statistiques + graphiques.

**7. Finition & déploiement (1h) :**

- Responsive design.
- Déploiement GitHub .