# Fiche du cours

55 h

**Titre :**

Web231 - JavaScript & Angular : Introduction

**Description :**

Découverte JavaScript moderne (ES6+) et Angular pour bâtir une SPA maintenable. Du langage (et bases TypeScript) aux composants, templates & data-binding, services/DI, routing, formulaires, HTTP/RxJS, tests et build/déploiement avec l'Angular CLI. Un mini-projet fil rouge consolide chaque notion.

**Objectifs :**

- Fondamentaux : JavaScript ES6+ et TypeScript utiles à Angular.*
- Vue : composants standalone, templates, liaisons (binding) et directives.*
- Architecture : services & injection de dépendances, routing, guards et lazy-loading.*
- Formulaires : template-driven & reactive forms avec validation.*
- Données & async : HttpClient, interceptors, environnements, RxJS (Observables) + aperçu Signals.*
- Qualité & prod : tests (Jest/Testing Library), ESLint/Prettier, build & déploiement

**Chapitres :**

1. JS moderne & TypeScript (types, interfaces) • Node/npm • Angular CLI & structure du projet*
2. Composants standalone • templates, data-binding (interpolation, property/event), directives & pipes*
3. Communication composants (@Input/@Output), services & DI, cycle de vie, pipes personnalisés*
4. Routing : Router, paramètres, navigation, guards, lazy-loading des features*
5. Formulaires I : template-driven, validation, messages d'erreur, UX*
6. Formulaires II : Reactive Forms, FormBuilder, validators custom*
7. HTTP & Data : HttpClient, interceptors (auth/log), gestion d'erreurs, environments*
8. RxJS & async : Observables, opérateurs usuels (map, switchMap, catchError), aperçu Signals*
9. UI & Qualité : Angular Material/Tailwind (au choix), accessibilité (a11y), tests (Jest/RTL), ESLint/Prettier*
10. Mini-projet : SPA "catalogue" (liste/détail + formulaire + API), build et déploiement d'un aperçu en ligne

| **À la fin :** |
| --- |
| Vous saurez créer une SPA Angular : composants propres, services/DI, routing, formulaires (reactive), HTTP/RxJS, tests et build/déploiement — prête à montrer en portfolio.<br>Suite recommandée (avancé) : NgRx ou Signals pour l’état global, Angular Universal (SSR), PWA, optimisation de performance (OnPush, pré-chargement de routes) et tests e2e (Cypress/Playwright). |