

# Cours Angular : Écosystème, Prérequis et Démarrage d'un Projet

## 1. Introduction à Angular

Angular est un **framework open-source** développé par Google pour créer des applications web **modernes, modulaires et performantes**.

- Langage principal : **TypeScript**
  - Architecture : **MVC / composants**
  - Basé sur la **programmation réactive** (RxJS)
  - Avantages : maintenable, testable, scalable
- 

## 2. Écosystème Angular

### 2.1 Angular CLI

- Outil en ligne de commande pour créer, configurer et gérer un projet Angular
- Commandes importantes :
  - `ng new` : créer un projet
  - `ng serve` : exécuter l'application
  - `ng generate` : générer composants, services, modules
  - `ng build` : compiler le projet pour production

### 2.2 Angular Modules

- `AppModule` : module principal
- Modules fonctionnels : lazy-loaded, feature modules
- Modularité = meilleure organisation et maintenance

### 2.3 Packages et dépendances

- `@angular/core` : cœur du framework
- `@angular/router` : routage
- `@angular/forms` : gestion des formulaires
- `@angular/common/http` : communication HTTP
- RxJS : programmation réactive

### 2.4 Environnement et tooling

- Node.js (>=18) et npm ou yarn
  - IDE recommandé : VS Code
  - Extensions utiles : Angular Language Service, Prettier
-

## 3. Prérequis

### 3.1 Logiciels nécessaires

- Node.js et npm (ou yarn)
- Angular CLI (`npm install -g @angular/cli`)
- Navigateur moderne (Chrome, Firefox, Edge)

### 3.2 Connaissances recommandées

- HTML / CSS / JavaScript
  - TypeScript
  - Concepts de programmation orientée objet
  - Concepts de programmation réactive (Observables)
- 

## 4. Création d'un projet Angular

### 4.1 Commande de création

```
ng new mon-projet-angular
```

- Options proposées : routing (oui/non), CSS/SCSS/SASS/LESS

### 4.2 Structure du projet

```
mon-projet-angular/  
|-- src/  
|   |-- app/          # composants, services, modules  
|   |-- assets/       # images, fichiers statiques  
|   |-- environments/ # configuration dev/prod  
|-- angular.json      # configuration Angular  
|-- package.json      # dépendances et scripts npm
```

---

## 5. Configuration du projet

### 5.1 Fichiers importants

- `angular.json` : configuration globale du projet
- `tsconfig.json` : configuration TypeScript
- `package.json` : dépendances et scripts
- `environments/` : variables pour dev et prod

## 5.2 Ajout de modules et packages

```
ng add @angular/material # Material Design
npm install rxjs          # RxJS (si nécessaire)
```

## 5.3 Paramètres CLI utiles

- `ng config` : configurer le projet globalement
  - `ng lint` : vérifier la qualité du code
  - `ng test` : exécuter les tests unitaires
- 

# 6. Exécution et compilation

## 6.1 Exécution en développement

```
ng serve --open
```

- Compile et démarre un serveur local (par défaut `http://localhost:4200`) - `--open` : ouvre automatiquement dans le navigateur

## 6.2 Compilation pour production

```
ng build --prod
```

- Génère les fichiers optimisés dans le dossier `dist/`

## 6.3 Autres commandes utiles

- `ng generate component mon-composant` : créer un composant
  - `ng generate service mon-service` : créer un service
  - `ng test` : exécuter les tests
  - `ng e2e` : tests end-to-end
- 

# 7. Bonnes pratiques

- Toujours utiliser Angular CLI pour générer composants et services
  - Organiser le code par modules (feature modules, shared modules)
  - Séparer les environnements (dev/prod)
  - Versionner le projet avec Git
  - Suivre les conventions de nommage Angular
-

## 8. Résumé

- Angular = framework moderne pour applications web modulaires
- CLI = outil essentiel pour créer, configurer et gérer le projet
- Connaissances préalables : TypeScript, HTML, CSS, RxJS
- Développement : `ng serve`, production : `ng build --prod`
- Architecture modulable et bonnes pratiques = maintenabilité et scalabilité

---

👉 Prochaine étape recommandée : **cours détaillé sur la structure d'un projet Angular et son control flow.**