

# Angular

---

## Outils

- Angular n'apporte pas d'opinion CSS (pas de framework), en choisir un.
- Installer :
  - nodejs + npm
  - @angular/cli : `npm install --global @angular/cli`
- Si VS Code : Angular Essentials
  - Sinon installer `Angular Language Service`
- Augury pour Chrome
- `@compodoc/compodoc`

## Outils : @compodoc/compodoc

### Installation

```
npm i @compodoc/compodoc
```

### Modifier le `package.json`

```
{
  "scripts": {
    "create:doc": "compodoc -p tsconfig.app.json -s -w"
  }
}
```

### Générer la doc

```
npm run create:doc
```

## Outils : npm

Il est préférable d'installer les dépendances en **local**

### Installation globale :

```
npm i -g PACKAGE_NAME@VERSION
```

Installation locale :

```
npm i PACKAGE_NAME@VERSION
```

## Outils : @angular/cli

Créer un espace de travail (avec une application par défaut) :

```
ng new WORKSPACE_NAME
```

Fichiers de configuration importants :

- **angular.json** - @angular/cli.
- **package.json** - npm (description, scripts, dependances)
- **tsconfig.json** - typescript.

Générer des éléments applicatifs:

```
ng g SCHEMATIC PATH/NAME
ng g d DIRECTIVE_NAME
ng g s SERVICE_NAME
ng g p PIPE_NAME
ng g m MODULE_NAME
ng g c COMPONENT_NAME
```

Lancer l'application en dev:

```
cd WORKSPACE_PROJECT
ng serve [-o --open --prod]
```

Créer un livrable :

```
cd WORKSPACE_PROJECT
ng build [--prod]
```

## Concepts

Angular est un framework de composant Web Composant : brique d'interface ré utilisable

### Concepts : Components

Structuration de la UI en fonctionnalités.

- RECETTE - Moteur de Rendu
- Lifecycle
- HTML
- CSS
- JS -> Logique encapsulée

Un module Angular est un regroupement de **Component**, **Pipe**, **Directive**. Il y a toujours au moins un module. (le premier root ou **bootstrap** module).

### Concepts : Components Angular

Identifier par le décorateur **@Component**

Penser a identifier les "SMART" vs "DUMB".

- **SMART** - Est en relation avec les services.
- **DUMB** - Est configuré.

## Lifecycle

- constructor
- parfois : **ngOnChanges**
- **ngOnInit**
- **ngDoCheck**
  - Possède de sous étapes
- **ngOnDestroy**

Syntaxe de template :

```
<div>
  <!-- Binding textuel -->
  {{expressionOrClassValueOrMethodCallToString}}
  <ng-content select="CSSSelector">
    <!-- Permet la récupération de contenu FILTRE PAR LE CSSSelector projeté -->
  </ng-content>
  <ng-content>
    <!-- Permet la récupération de conetnu projeté -->
  </ng-content>
```

```

<input [value]=" evaluatedExpression ">
<button (click)=" evaluatedExpression " >Click Me I am Famous</button>
<div *ngIf=" evaluatedBooleanExpression " >
  Ce message serait conditionnel !
</div>
<ul>
  <li *ngFor="let myVar of myIterableCollection">{{myVar}}</li>
  <!-- for(let myVar of myIterableCollection) {} -->
</ul>
<div [ngClass]=" {
  'one-css-class-name-to-set': evaluatedBooleanExpression,
  'other-css-class-name-to-set': evaluatedBooleanExpression,
} ">
</div>
</div>

```

## Concepts JavaScript

Tous les composants clés **Component**, **Pipe**, **Directive**, **service**, **module** sont des **class**.

## ES6 pour angular

- **modules**
- **Promise**
- **=>**
- **class**
- block scope **let, const**

## Observable **rxjs**

Une représentation de l'asynchronicité (en javascript).

1. Créer un **Observable**
  - Sur un **sujet**
  - on peut créer canal de traitement
    - **.pipe( operator() )**
    - **.pipe( operator() )**
    - Les opérateur sont fournis par **rxjs** et indique la nature de la transformation.
2. Y souscrire **.subscribe(callback)**

## 5 Concepts clés Angular

- **NgModule** - regroupement logique
  - **Component**
  - **Directive** - enrichissement
  - **Pipe** - Transformation
- **Service** - Logique Métier

## 19 Décorateurs

- @NgModule
- @Component
- @Injectable
- @Pipe
- @Directive
- @Inject
- @HostBinding
- @HostListener
- @Input
- @Output
- @ViewChild
- @ViewChildren
- @ContentChild
- @ContentChildren

## Méthodologie :

1. Définir les interfaces.
2. Définir les features (composant fonctionnels)
3. A. Définir les services
4. B. Organiser les modules.
  - Créer les composant
  - Les directives.
  - Les pipe.