



# Formation ANGULAR

## Chapitre 1

Animé par Michel BOCCIOLESI

# Table des matières

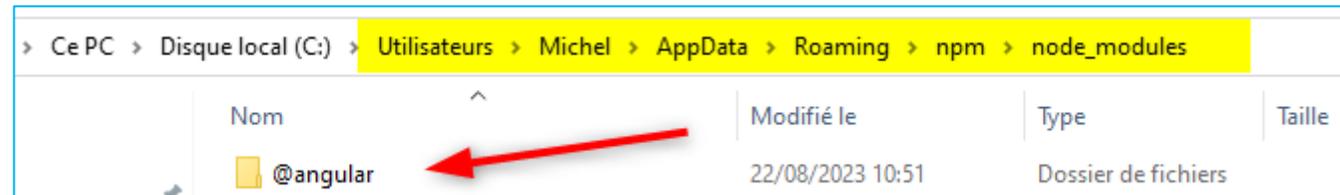
- **Intérêt du Framework Angular**  
Framework à base de modules et de web-components - webpack  
Architecture à base de web-components
- **« Nouveautés » Ecmascript 2015+**  
La révolution Javascript
- **Le Framework Angular - Historique**  
Angular JS à Angular 2+  
Les différentes versions Angular 2 à 16  
Les versions marquantes
- **Documenter son projet – fichiers MD**
- **Les composants :**  
TS – HTML – CSS  
Binding – décorateurs  
composants parents-enfants  
directives - pipes
- **Les modules :**  
« structurels » - architecture de projet  
« fonctionnels » - partie spécifique de l'application (compte-client, panier)
- **Mise en place du projet « fil rouge » de la formation**  
Application des acquis
- **Injection de dépendances (service Web ou autres)**
- **Les cycles de vie du composant**  
constructor – ngOnInit – la gestion des datas
- **Le routage Angular « navigation et liens »**  
Concepts de base et avancés
- **Communication entre composants parents et enfants**  
@Input() @Output()
- **Les formulaires Angular**  
Reactive Forms vs Template
- **La programmation RXJS et les observables**
- **Test Unitaires**  
Introduction à Karma et Jasmine

# Installation et configuration

<https://nodejs.org/fr>  
<https://angular.io/>  
<https://www.npmjs.com/package/@angular/cli>

- `npm install -g @angular/cli` (global dans le path utilisateur)

`/home/michel (linux)`



- `npm install @angular/cli@12`
- `npm install @angular@15`

# Documenter son projet



The image shows a side-by-side comparison of a code editor and a PDF viewer. The code editor on the left displays the content of a file named 'COMMANDES.md'. The PDF viewer on the right shows the rendered version of this file as 'COMMANDES.pdf'. The PDF content is a structured list of Angular CLI commands and options, with each section clearly separated by horizontal lines.

```
formation-angular-2023-06 > Documentation > COMMANDES.md > # Generate objets > ## composant
1 Pour générer le fichier lisible : CTRL + SHIFT + v
2 ou bouton droit
3
4 # Commandes Angular NG
5
6 # New projet
7
8 ng new nomDuProjet
9
10 ## options :
11
12 # Generate objets
13
14 ## composant
15 - ng g(enerate) c(omposant) nomDuComposant
16 - options :
17 --export (exporte le composant depuis le module)
18 --skip-tests : omet la création du fichier de spec.ts (tests
unitaires)
19 --skip-import : ne déclare pas le composant dans un module
20 --flat : crée le composant sans dossier
21
22 ## module
23 - ng g(enerate) m(odule) nomDuModule
24 - options :
25 --routing : crée le fichier de routing forChild pour le module
concerné
26
27 ## service
```

COMMANDES.pdf

COMMANDES.md

Pour générer le fichier lisible : CTRL + SHIFT + v ou bouton droit

---

## Commandes Angular NG

---

### New projet

ng new nomDuProjet

options :

---

### Generate objets

---

#### composant

- ng g(enerate) c(omposant) nomDuComposant
- options : --export (exporte le composant depuis le module) --skip-tests : spec.ts (tests unitaires) --skip-import : ne déclare pas le composant dans un composant sans dossier

# Documenter son projet



The screenshot shows the Visual Studio Code interface with the Markdown PDF extension installed. The left pane displays the content of the 'COMMANDES.md' file, which includes the text 'Commandes Angular NG' and 'New projet'. A yellow highlight is placed over the text 'CTRL + SHIFT + v ou bouton droit'. The right pane shows the extension's details, including its icon (a circle with 'M↓' and 'PDF'), the name 'Markdown PDF', version 'v1.5.0', author 'yzane', download count '1747619', and a 5-star rating. There are buttons for 'Désactiver', 'Désinstaller', and a settings gear icon. A status message at the bottom indicates 'Cette extension est activée globalement.'

COMMANDES.md | Prévisualiser COMMANDES.md × | ... | COMMANDES.pdf | Extension : Markdown PDF ×

Pour générer le fichier lisible : CTRL + SHIFT + v ou bouton droit

## Commandes Angular NG

---

## New projet

---

### Markdown PDF v1.5.0

yzane | 1747619 | ★★★★★ (105)

Convert Markdown to PDF

Désactiver | Désinstaller | ⚙️

Cette extension est activée globalement.

```
PS C:\Users\Michel> node -v
v18.12.1
PS C:\Users\Michel> ng version
```

# Angular CLI

```
Angular CLI: 16.2.4
Node: 18.12.1
Package Manager: npm 9.7.2
OS: win32 x64
```

```
Angular: undefined
...
```

Package	Version
@angular-devkit/architect	0.1602.4 (cli-only)
@angular-devkit/core	16.2.4 (cli-only)
@angular-devkit/schematics	16.2.4 (cli-only)
@schematics/angular	16.2.4 (cli-only)

```
PS C:\Users\Michel> ng new TP01-les-fondamentaux
```

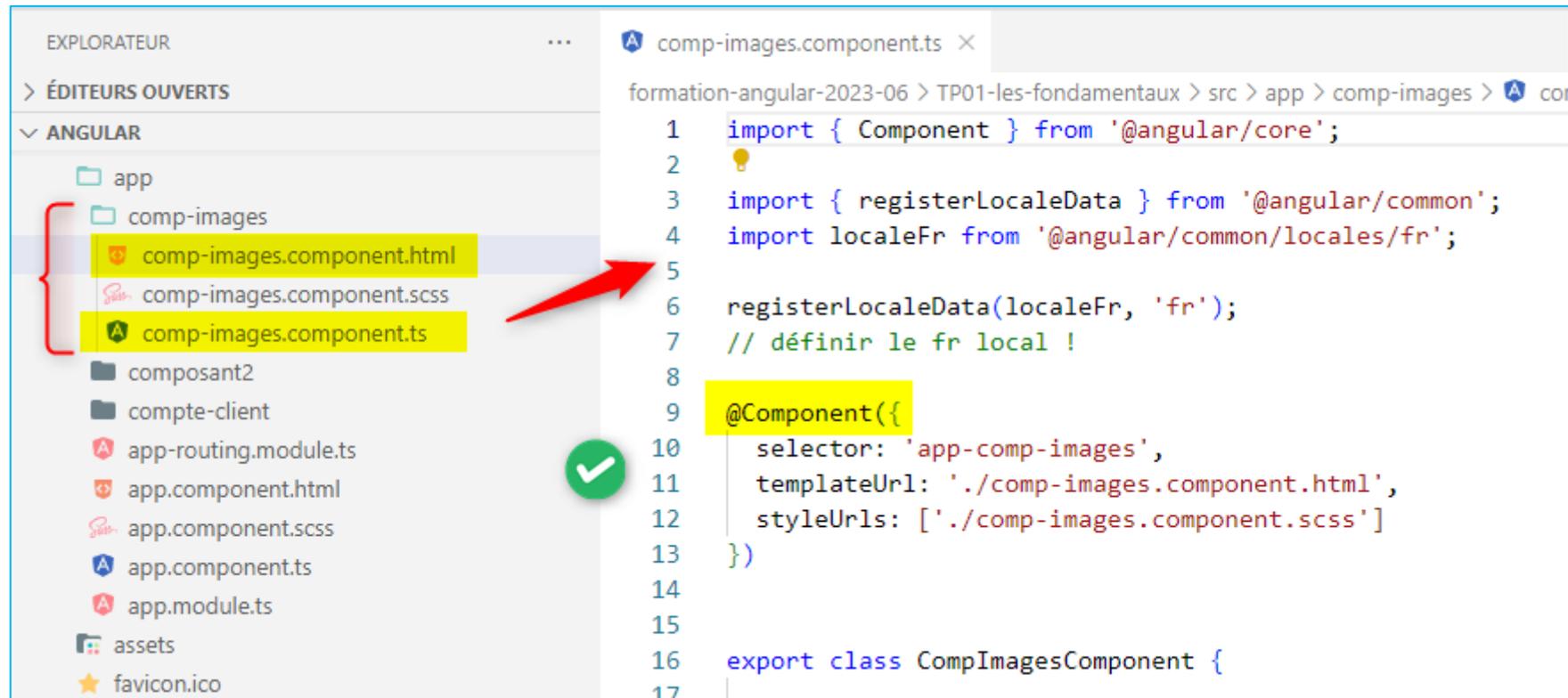
## Prise en main de l'architecture d'un projet NG

```
package.json index.html angular.json app.component.html main.ts
fondamentaux > angular.json > {} projects > {} TP01-les-fondamentaux > {} architect > {} build > {} options
9   "@schematics/angular:component": {
10     "style": "scss"
11   }
12 },
13 "root": "",
14 "sourceRoot": "src",
15 "prefix": "app",
16 "architect": {
17   "build": {
18     "builder": "@angular-devkit/build-angular:browser",
19     "options": {
20       "outputPath": "dist/tp01-les-fondamentaux",
21       "index": "src/index.html",
22       "main": "src/main.ts",
23       "polyfills": [
TP01-les-fondamentaux > src > main.ts
1 import { platformBrowserDynamic } from '@angular/platform-brows
2
3 // import LOCAL (import ES 2015)
4 import { AppModule } from './app/app.module';
5
6 // est le point d'entrée du projet
7 // charger ou bootstrapper le tout 1er module
8
9 platformBrowserDynamic().bootstrapModule(AppModule)
10 | // .catch(err => console.error(err));
11
app.component.scss app.module.ts app.component.ts
TP01-les-fondamentaux > app > app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6
7 // un décorateur transforme la class en objet NG (comp, module,
  service ...)
8 @NgModule({
9   declarations: [
10     // liste de tous les composants du module
11     AppComponent,
12     // PlanDeCoursComponent,
13     // DatesCoursComponent
14   ],
15   imports: [
16     // toutes les librairies nécessaires à ce module
17     BrowserModule,
18     AppRoutingModule
19   ],
20   providers: [
21     // services
22   ],
23   bootstrap: [
24     // quel est le 1er composant chargé ?
25     AppComponent
26   ]
27 })
TP01-les-fondamentaux > src > app > app.component.ts > ...
1 import { Component } from '@angular/core';
2
3 // un décorateur transforme la class en objet NG (comp, module,
4 @Component({
5   // props ou directives
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.scss']
9 })
10
11 // export et class sont des mots ECMAScript 2015
12 export class AppComponent {
13
14   // propriétés
15   title = 'TP01-les-fondamentaux';
16 }
17
```

# Intérêts du Framework Angular

## Les web-components (composants web)

1 composant est constitué de 3 fichiers : TS, la vue HTML et le fichier de style CSS\*



The screenshot displays the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure under 'ANGULAR'. A folder named 'app' contains a sub-folder 'comp-images'. Inside 'comp-images', three files are listed: 'comp-images.component.html', 'comp-images.component.scss', and 'comp-images.component.ts'. A red bracket groups these three files, and a red arrow points from this group to the code editor on the right. The code editor shows the content of 'comp-images.component.ts'. The code includes imports for '@angular/core', '@angular/common', and '@angular/common/locales/fr'. It also shows the registration of locale data for 'fr' and the definition of the '@Component' decorator with properties for selector, templateUrl, and styleUrls. The component class 'CompImagesComponent' is partially visible at the bottom.

```
1 import { Component } from '@angular/core';
2
3 import { registerLocaleData } from '@angular/common';
4 import localeFr from '@angular/common/locales/fr';
5
6 registerLocaleData(localeFr, 'fr');
7 // définir le fr local !
8
9 @Component({
10   selector: 'app-comp-images',
11   templateUrl: './comp-images.component.html',
12   styleUrls: ['./comp-images.component.scss']
13 })
14
15
16 export class CompImagesComponent {
17
```

# Intérêts du Framework Angular



## Les web-components (composants web)

le composant web permet de travailler une partie de la page globale index.html tant au niveau des données (le TS) que de la vue HTML+CSS\*

```
app.component.ts ×
formation-angular-2023-06 > TP01-les-fondamentaux > src > app > app.component.ts > ...
1 import { Component } from '@angular/core';
2
3 // un décorateur : son rôle : transformer la
  classe en ... (composant)
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.scss']
9 })
10
11 // une classe (ES2015) exportée
12 export class AppComponent {
13
14   // 1- déf des propriétés
15   public title: string = 'Angular';
16   public version: number;
17
18   // 2- constructeur de classe
19   constructor() {
20     this.version = 16;
21   }
22 }
23

app.component.html ×
formation-angular-2023-06 > TP01-les-fondamentaux > src > app > app.component.html > ...
1
2 <!-- string interpolation de variables {{ }} -->
3 <!-- Binding : le TS est lié à la vue HTML(template) -->
4 <!-- Single way Binding: TS => HTML -->
5 <h1>Formation {{title}} {{version}}</h1>
6
7
8 <!-- composant enfant de AppComponent -->
9 <app-composant2></app-composant2>
10
11 <hr>
12
13 <app-comp-images></app-comp-images>
14
15 <hr>
16 <h2>Espace Client ...</h2>
17 <app-landing-page></app-landing-page>
```

# Webpack

Le webpack est un environnement de développement NODEJS, composé de plusieurs fichiers de configuration permettant de « customiser » le développement (serve) et le build (compilation)

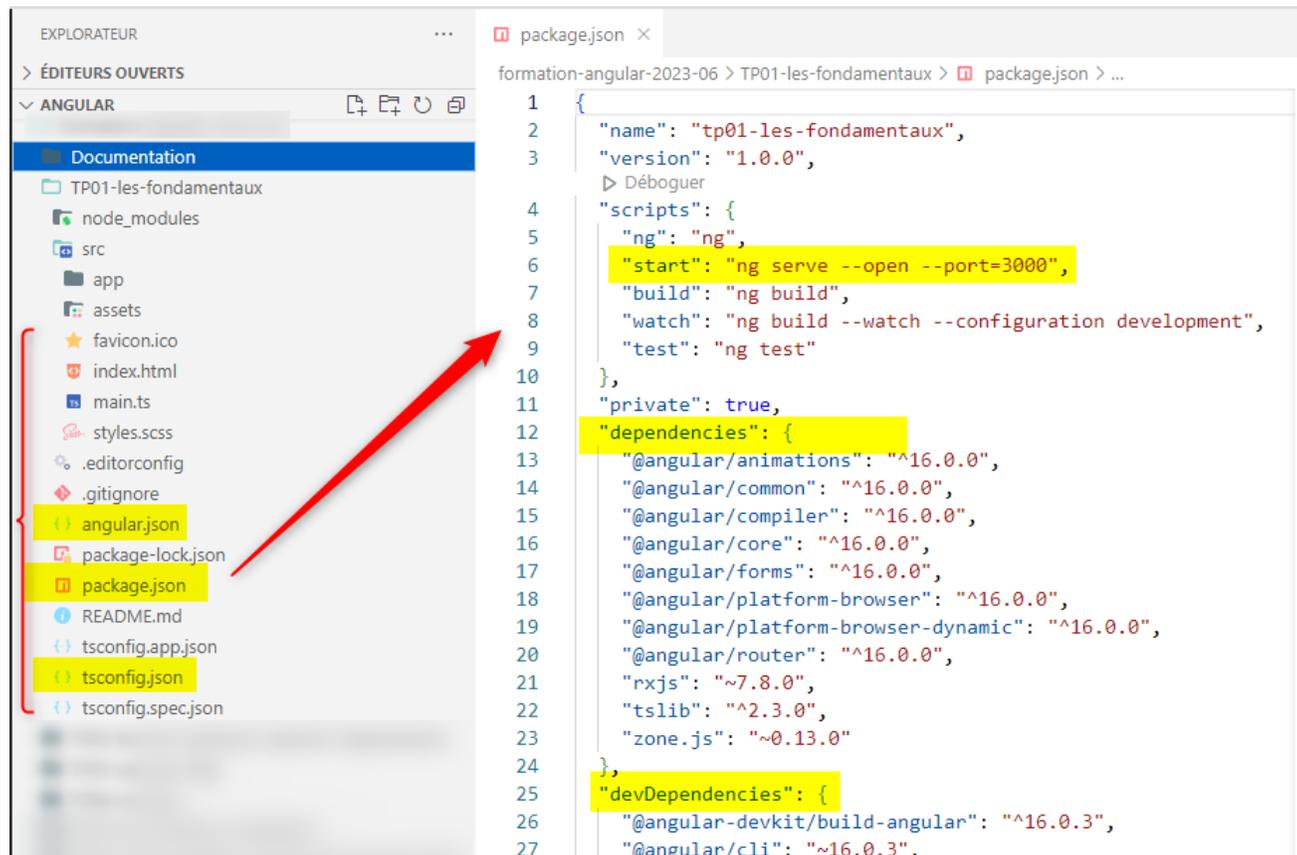
- package.json
- angular.json
- tsconfig.json
- répertoire node\_modules \*

node\_modules \* contient toutes les librairies (dépendances) nécessaires au projet webpack.

Il y a au moins 32k fichiers dans un simple projet Angular.

npm init -y : crée un projet webpack

npm update : met à jour les paquets selon les règles du package.json



```
1 {
2   "name": "tp01-les-fondamentaux",
3   "version": "1.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve --open --port=3000",
7     "build": "ng build",
8     "watch": "ng build --watch --configuration development",
9     "test": "ng test"
10  },
11  "private": true,
12  "dependencies": {
13    "@angular/animations": "^16.0.0",
14    "@angular/common": "^16.0.0",
15    "@angular/compiler": "^16.0.0",
16    "@angular/core": "^16.0.0",
17    "@angular/forms": "^16.0.0",
18    "@angular/platform-browser": "^16.0.0",
19    "@angular/platform-browser-dynamic": "^16.0.0",
20    "@angular/router": "^16.0.0",
21    "rxjs": "~7.8.0",
22    "tslib": "^2.3.0",
23    "zone.js": "~0.13.0"
24  },
25  "devDependencies": {
26    "@angular-devkit/build-angular": "^16.0.3",
27    "@angular/cli": "~16.0.3",
```

# Composants et modules

Création de composants : composant2 – comp-images

- Le sélecteur de composant est placé dans la vue HTML d'un composant « parent »
- Le composant appelé par le sélecteur devient le composant enfant.

Un module « structurel » ou « fonctionnel » déclare ses composants et les exporte afin que leurs contenus soient exploitables dans les vues de composants d'autres modules. (import-export)

Ressources disponibles :

<https://dev.webjs.fr/1-Angular/ressources-formation/>



# Composants parents et enfants

The image shows a development environment with three code files and a browser preview:

- app.component.html**: Contains the main layout with comments and component tags. Lines 8-9 show `<app-composant2>` and line 13 shows `<app-comp-images>`. Red arrows point from these tags to the browser preview.
- composant2.component.html**: Contains `<h1>Composant #2</h1>`. A red arrow points from this line to the browser preview.
- comp-images.component.html**: Contains a loop using `*ngFor` to render images. A red arrow points from the `<img>` tag to the browser preview.

The browser preview at `localhost:3000` displays the rendered output:

- Header: **Formation Angular 16**
- Section: **Composant #2**
- Section: **Les Avatars ...**
- Content: Five circular avatars of Star Wars characters: Stormtrooper, Chewbacca, Rey, Han Solo, and Darth Vader.

# Directives Binding Pipes

```
1 <h1>{{title | titlecase}}</h1>
2
3 <!-- structural directives Angular (déjà faites...)
4 test IF -- boucles *ngIf *ngFor *ngSwitch .. -->
5
6
7 <!-- les directives s'écrivent (presque tout le temps) avec des crochets -->
8 <div *ngFor="let valeur of imagesArray" >
9   <span>
10     
19     <!-- la directive ngClass affiche une classe nommée si la condition se réalise -->
20   </span>
21 </div>
22
23 <!-- *ngIf : affiche (ou pas) ce qui est contenu dans l'élément en fonction d'un paramètre-->
24 <div *ngIf="flag===true">
25   <p>{{formation}}</p>
26 </div>
27
28 <!-- *ngSwitchCase ==> gère plusieurs tests (conditions) -->
29 <div [ngSwitch]="codeCours">
30
31   <p *ngSwitchCase="'ng'">Formation Angular</p>
32   <p *ngSwitchCase="'react'">Formation React</p>
33   <p *ngSwitchDefault>Formation non trouvée.</p>
34
35 </div>
36
37 <p>{{maDate | date:'dd/M/Y H:m:s'}}</p>
38 <!-- <p>{{total | number:'1.0-2':'fr'}}</p> -->
39 <p>{{total | currency:'EUR':'symbol':'1.0-2':'fr'}}</p>
```

```
app.component.html ×
formation-angular-2023-06 > TP01-les-fondamentaux > src > app > app.component.html > ...
1
2 <!-- string interpolation de variables {{ }} -->
3 <!-- Binding : le TS est lié à la vue HTML(template) -->
4 <!-- Single way Binding: TS => HTML -->
5 <h1>Formation {{title}} {{version}}</h1>
6
7
8 <!-- composant enfant de AppComponent -->
9 <app-composant2></app-composant2>
10
11 <hr>
12
13 <app-comp-images></app-comp-images>
14
15 <hr>
16 <h2>Espace Client ...</h2>
17 <app-landing-page></app-landing-page>

app.module.ts ×
formation-angular-2023-06 > TP01-les-fondamentaux > src > app > app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { NgModule } from '@angular/core';
5
6 // -----
7 // IMPORT LOCAL (ressources fichiers)
8 import { AppRoutingModule } from './app-routing.module';
9 import { AppComponent } from './app.component';
10 import { FormsModule } from '@angular/forms';
11 import { Composant2Component } from './composant2/composant2.component';
12 import { CompImagesComponent } from './comp-images/comp-images.component';
13 import { HeaderComponent } from './compte-client/composants/header/header.component';
14 import { CompteClientModule } from './compte-client/compte-client.module';
15
16 // un décorateur son rôle : transformer la classe en ... (module)
17 @NgModule({
18 // objet de description
19
20 })
21
22 compte-client.module.ts ×
formation-angular-2023-06 > TP01-les-fondamentaux > src > app > compte-client > compte-client.module.ts > ...
13 HeaderComponent,
14 FooterComponent,
15 LandingPageComponent
16 ],
17 imports: [
18   CommonModule
19 ],
20 exports: [
21   HeaderComponent,
22   BodyComponent,
23   FooterComponent,
24   LandingPageComponent
25 ]
26 })
27 export class CompteClientModule { }
28
```

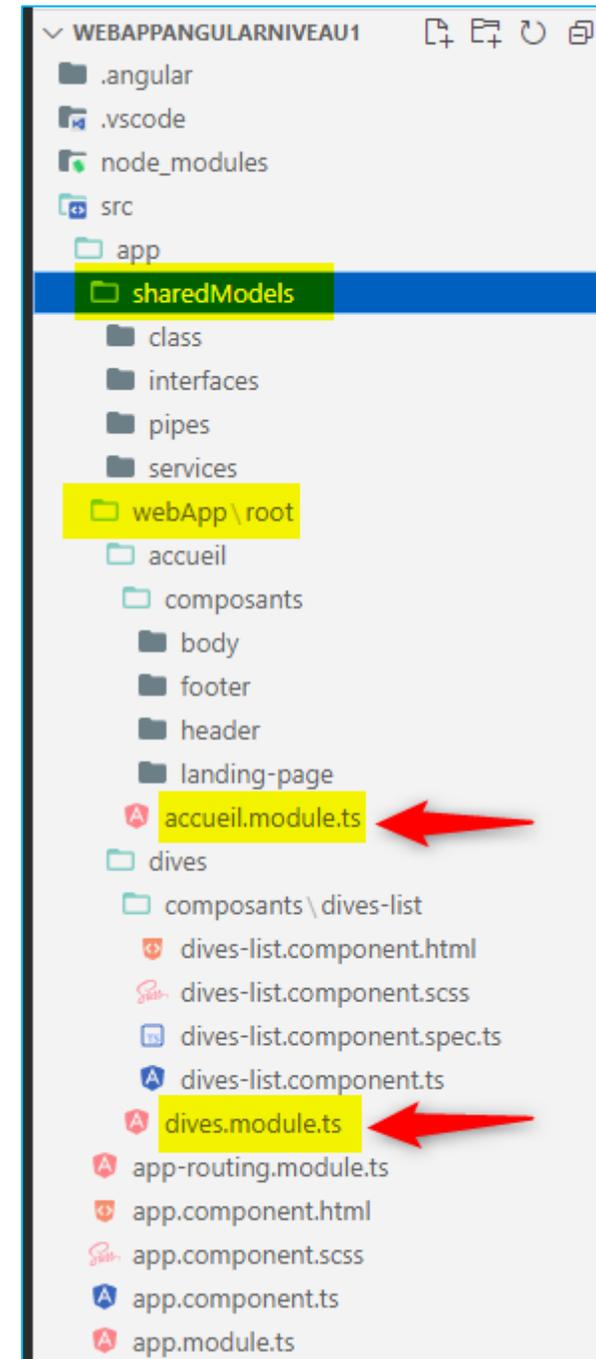
Export des composants depuis le module « compte-client »  
Import du module « compte-client » dans « app-module »

# Mise en place du TP fil rouge

- Création d'un nouveau projet – avec le routage et SCSS
- Installation des dépendances Bootstrap et bootstrap-icons  
<https://www.npmjs.com/package/bootstrap>  
<https://www.npmjs.com/package/bootstrap-icons>
- Installation de la « dev » dépendance json-server  
npm install -D json-server

```
9   "test": "ng test",  
10  "json-server": "json-server --watch src/assets/json/bdd.json -p 3001"  
11
```

- Bonnes pratiques « structure et architecture » de projet
- injection de dépendances ...



# Mise en place du TP fil rouge

```
PS C:\Users\Michel\Desktop\formation-angular-sparks-10-2023\TP03-services-HTTP> npm run json-server  
  
> tp02-bonnes-pratiques-injection-dependances@0.0.0 json-server  
> json-server --watch src/assets/json/bdd.json -p 3001  
  
\{^_^}/ hi!  
  
Loading src/assets/json/bdd.json  
Done  
  
Resources  
http://localhost:3001/dives  
http://localhost:3001/produits  
  
Home  
http://localhost:3001  
  
Type s + enter at any time to create a snapshot of the database  
Watching...  
□
```

# Mise en place du TP fil rouge

```
package.json U x header.component.html U ... angular.json U x
TP02-bonnes-pratiques > package.json > {} dependencies
1 {
2   "name": "tp02-bonnes-pratiques",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve --open --port=3000",
7     "build": "ng build",
8     "watch": "ng build --watch --configuration developer",
9     "test": "ng test"
10  },
11  "private": true,
12  "dependencies": {
13    "@angular/animations": "^16.2.0",
14    "@angular/common": "^16.2.0",
15    "@angular/compiler": "^16.2.0",
16    "@angular/core": "^16.2.0",
17    "@angular/forms": "^16.2.0",
18    "@angular/platform-browser": "^16.2.0",
19    "@angular/platform-browser-dynamic": "^16.2.0",
20    "@angular/router": "^16.2.0",
21    "bootstrap": "^5.3.2",
22    "bootstrap-icons": "^1.11.1",
23    "rxjs": "~7.8.0",
24    "tslib": "^2.3.0",
25    "zone.js": "~0.13.0"
26  },
27  "devDependencies": {
28    "@angular-devkit/build-angular": "^16.2.0",
29    "@angular/cli": "~16.2.0",
30    "@types/jasmine": "~5.1.0",
31    "jasmine-core": "~5.1.0",
32    "karma": "~6.4.0",
33    "karma-chrome-launcher": "~3.1.0",
34    "karma-coverage": "~2.2.0",
35    "karma-jasmine": "~5.1.0",
36    "karma-jasmine-html-reporter": "~2.0.0",
37    "typescript": "~5.1.0"
38  }
39 }
40

TP02-bonnes-pratiques > angular.json > {} projects > {} TP02-bonnes-pratiques > {} architect > {} build > {}
16 "architect": {
17   "build": {
18     "builder": "@angular-devkit/build-angular:browser",
19     "options": {
20       "outputPath": "dist/tp02-bonnes-pratiques",
21       "index": "src/index.html",
22       "main": "src/main.ts",
23       "polyfills": [
24         "zone.js"
25       ],
26       "tsConfig": "tsconfig.app.json",
27       "inlineStyleLanguage": "scss",
28       "assets": [
29         "src/favicon.ico",
30         "src/assets"
31       ],
32       "styles": [
33         "node_modules/bootstrap/dist/css/bootstrap.min.css",
34         "node_modules/bootstrap-icons/font/bootstrap-icons.css",
35         "src/styles.scss"
36       ],
37       "scripts": [
38         "node_modules/bootstrap/dist/js/bootstrap.min.js"
39       ]
40     }
41   }
42 }
```

app.module.ts

```

1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { AccueilModule } from './webApp/accueil/accueil.module';
7
8 @NgModule({
9   declarations: [
10    AppComponent
11  ],
12  imports: [
13    BrowserModule,
14    AppRoutingModule,
15    AccueilModule
16  ],
17  providers: [],
18  bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21

```

landing-page.component.html

```

1 <app-header></app-header>
2 <!-- <app-body></app-body -->
3 <app-dives-list></app-dives-list>
4 <app-footer></app-footer>

```



accueil.module.ts

```

1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { LandingPageComponent } from './composants/landing-page/landing-page.component';
4 import { HeaderComponent } from './composants/header/header.component';
5 import { FooterComponent } from './composants/footer/footer.component';
6 import { BodyComponent } from './composants/body/body.component';
7 import { DivesModule } from '../dives/dives.module';
8
9 @NgModule({
10  declarations: [
11    LandingPageComponent,
12    HeaderComponent,
13    FooterComponent,
14    BodyComponent
15  ],
16  imports: [
17    CommonModule,
18    DivesModule
19  ],
20  exports: [
21    LandingPageComponent,

```

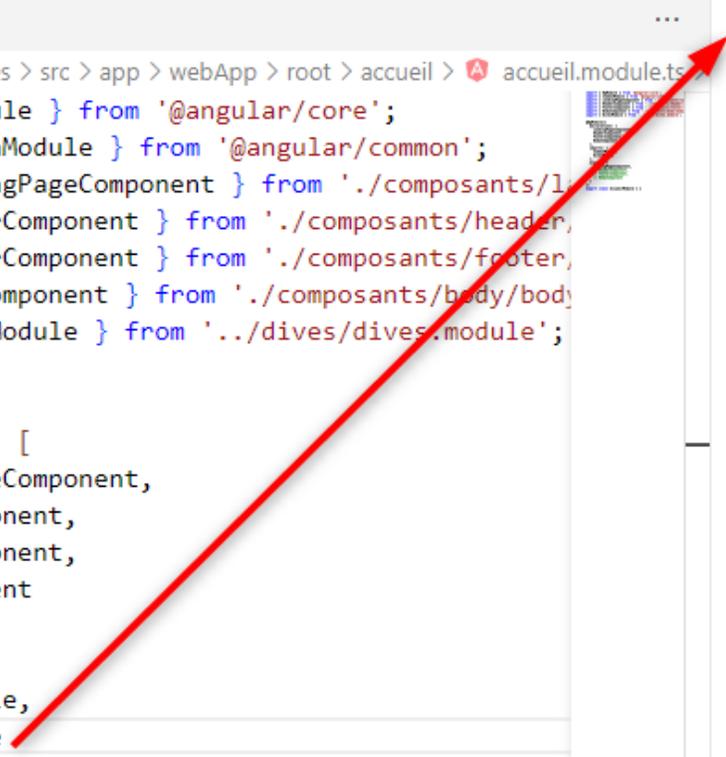


dives.module.ts

```

1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { DivesListComponent } from './dives-list/dives-list.component';
4
5 @NgModule({
6   declarations: [
7     DivesListComponent
8   ],
9   imports: [
10    CommonModule
11  ],
12   exports: [
13     DivesListComponent
14  ]
15 })
16 export class DivesModule { }
17
18

```



# Life Cycle du composant constructor & ngOnInit

```
diver-list.component.ts 3  diver.service.ts X
TP02-bonnes-pratiques-injection-dependances > src > app > sharedModel
1 import { Injectable } from '@angular/core';
2 import { Dives } from '../class/dives';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7
8 export class DivesService {
9
10   // 1- props
11   private dives: Dives[];
12
13   // 2- const
14   constructor() {
15     this.dives = [
16       {
17         "id": 1,
18         "name": "Trou Aux Biches - L'Epave du
19         "location": "Ile Maurice - Trou aux B
```

```
constructor(
  private _service: DivesService,
  private _title: Title,
  private _meta: Meta
) {
  console.warn('Constructor');
  this.dives = [];
  this._title.setTitle('Liste des Plongées SEO ....');
  this._meta.addTags([
    { name: 'description', content: 'Texte de description ...' },
    { name: 'author', content: 'MB CREATION WEB' }
  ]);
}

// -----
// 3- lifeCycle
// -----
ngOnInit(): void {
  // Chargement des datas
  console.warn('NgOnInit');
  // console.table(this._service.getDives());
  // -- on passe les datas du service à la prop dives du composant
  this.dives = this._service.getDives();
}
```

# L'injection de dépendances Angular

Au moment de la création du composant (constructor)  
Le composant reçoit un service ou une « fonctionnalité » requise

```
dives.service.ts | dives-list.component.ts
```

```
81     "location": "Cozumel - Mexique - Départ de Pl",
82     "level": 2,
83     "description": "Plongée de niveau 2 FFESSM ou",
84     "latitude": 20.4317585,
85     "longitude": -86.9202746,
86     "evaluation": [
87       7,
88       9,
89       8
90     ],
91     "photo": "https://dev.webjts.fr/images/fond5.jpg"
92   };
93 };
94 }
95
96 // Méthodes
97 public getDives = () => {
98   return this.dives;
99 }
100 }
101 }
102
```

```
1  import { Component, OnInit } from '@angular/core';
2  import { DivesService } from 'src/app/sharedModels/services';
3
4  @Component({
5    selector: 'app-dives-list',
6    templateUrl: './dives-list.component.html',
7    styleUrls: ['./dives-list.component.scss']
8  })
9  export class DivesListComponent implements OnInit {
10
11    // 1- Props
12
13    // 2- Constructor
14    constructor(
15      // injection de dépendances requises
16      // au moment de la construction (création)
17      // du composant
18      private _service:DivesService
19    ) {
20      console.warn('Constructor');
21    }
22
23    // 3- Lifecycle
24    ngOnInit(): void {
25      console.warn('NgOnInit');
26      // Rôle : chargement des datas
27      console.table(this._service.getDives());
28    }
29  }
```

# TP : renseigner les valeurs manquantes

```
dives-list.component.ts 9+ x
1 import { Component, OnInit } from '@angular/core';
2 import { Dives } from 'src/app/sharedModels/class/dives';
3 import { DivesService } from 'src/app/sharedModels/services/dives-service';
4
5 @Component({
6   selector: 'app-dives-list',
7   templateUrl: './dives-list.component.html',
8   styleUrls: ['./dives-list.component.scss']
9 })
10 export class DivesListComponent implements OnInit {
11
12   // 1- Props
13   ~~~~~
14
15   // 2- Constructor
16   constructor(
17     // injection de dépendances requises
18     // au moment de la construction (création)
19     // du composant
20     private _service:DivesService
21   ) {
22     console.warn('Constructor');
23     ~~~~~
24   }
25
26   // 3- Lifecycle
27   ngOnInit(): void {
28     console.warn('NgOnInit');
29     // Rôle : chargement des datas
30     console.table(this._service.getDives());
31     ~~~~~
32
33 }
34
```

```
dives-list.component.html x
1 <h1 class="alert alert-primary">Liste des Plongées</h1>
2
3 <div class="row">
4   <div class="col-md-3" *ngFor="<div class="card">
5     <!-- récupérer chaque image -->
6     <img src="" class="card-img-top">
7     <div class="card-body">
8       <h3 class="card-title">Nom : </h3>
9       <p class="card-text">Description : </p>
10
11     <!-- cibler un élément du dom en template référence # -->
12     <button class="btn btn-primary" #btnPlus>En Savoir Plus</button>
13
14   </div>
15 </div>
16 </div>
17
18 </div>
19
20 </div>
21
22
```



# Life Cycle du composant

## AfterViewInit – AfterViewChecked - OnDestroy

```
// -----  
//     cycles de vie  
// -----  
ngOnInit():void {  
  // exploitation des datas  
  console.warn('OnInit');  
  console.table(this._service.getDives());  
  this.dives=this._service.getDives();  
  console.log('dans le init : ', this.elBtn);  
  // output => undefined  
}
```

```
// -----  
ngAfterViewInit(): void {  
  console.warn('ngAfterViewInit');  
  // quand le DOM (document object model) est chargé  
  // toute la vue HTML et tous ces éléments sont prêts  
  console.log('dans le After View Init : ', this.elBtn);  
  // output => ElementRef {nativeElement: button.btn.btn-primary}  
  let texte='AfterViewInit';  
  // this.elBtn.nativeElement.innerHTML = `Plus d'infos ${texte}`;  
  
  console.log(this.elCollectionBtn);  
  
  this.elCollectionBtn.forEach(  
    (elt:ElementRef) => {  
      elt.nativeElement.innerHTML=`Plus d'infos`;  
      // elt.nativeElement.className='btn btn-warning';  
      elt.nativeElement.classList.replace('btn-primary', 'btn-success');  
    }  
  );  
}
```



# Rappels – Révisions Ecmascript



# Ecmascript 2015+



Ecmascript, la normalisation du langage Javascript a révolutionné Javascript en 2015 (ES6) en le dotant de fonctionnalités et spécificités dignes d'un « langage de haut niveau ».

Cette transformation totale et globale devenait nécessaire au vu des besoins des développements front web et mobile. *Angular utilise Ecmascript.*

Les améliorations les plus importantes du langage concernent :

1. La définition des variables et leur portée : VAR, LET et CONST
2. La possibilité de créer des modules de code exportable et importable : IMPORT, EXPORT
3. L'écriture simplifiée des FAT ARROWS `<script src="js/index.js" type="module" defer></script>`
4. La string interpolation des variables avec `${maVariable}` et la concatenation avec les back stits
5. Les itérateurs et les boucles FOR OF, FOR IN et FOR OF ENTRIES
6. Les spread operators pour déstructurer les tableaux
7. La notion de promesses afin de gérer les événements et les procédures asynchrones
8. Les objets et les CLASS\*

# Ecmascript 2015+



```
> for (var i=0; i<=3 ; i++ ) { console.log(i); }
0 VM5352:1
1 VM5352:1
2 VM5352:1
3 VM5352:1
< undefined
> console.log(i);
4 VM5411:1
< undefined
> for (let j=0; j<=3 ; j++ ) { console.log(j); }
0 VM5535:1
1 VM5535:1
2 VM5535:1
3 VM5535:1
< undefined
> console.log(j)
✖ Uncaught ReferenceError: j is not defined VM5613:1
  at <anonymous>:1:13
>
```

# EcmaScript 2015+



```
> for ( let val of tab ) { console.log(val); }
ok VM6316:1
cool VM6316:1
< undefined
> for ( let i in tab ) { console.log(i); }
0 VM6332:1
1 VM6332:1
< undefined
> for (let[i,val] of tab.entries() ) { console.log(i, val);}
0 'ok' VM6348:1
1 'cool' VM6348:1
< undefined
>
```

# EcmaScript 2015+



```
1 // nouveautés (fonctions)
2 function direBonjour(nom = "SkyWalker") {
3     // passage d'arguments par défaut
4     // console.log("Bonjour " + nom);
5
6     // string interpolation des variables
7     console.log(`Bonjour ${nom}`);
8     document.querySelector("#resultat").innerHTML = `Bonjour ${nom}`;
9 }
10
11 const direBonjour2 = (prenom, nom) => {
12     console.log(`Bonjour ${prenom} ${nom}`);
13     document.querySelector("#resultat").innerHTML = `Bonjour ${prenom} ${nom}`;
14 }
15
16 direBonjour(); // javascript s'écrit en camelCase majuscule à chaque mot sauf le 1er
17 direBonjour2("Dark", "Vador");
18
```

# Ecmascript 2015+

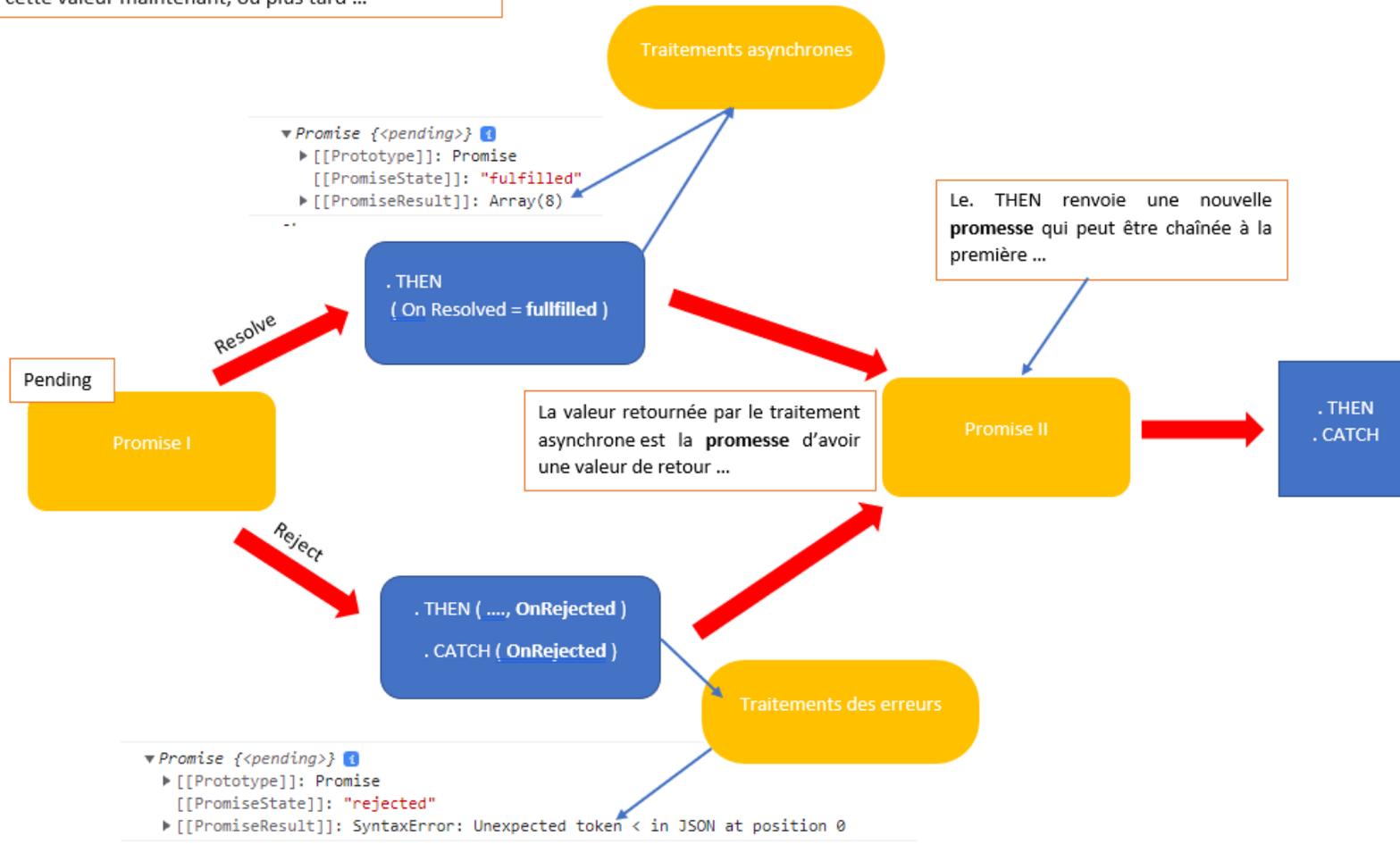


```
58 // -----  
59 let personne = {  
60   prenom: "Luke",  
61   nom: "Skywalker",  
62   // tableaux  
63   langages: ["JS", "React", "NG", "Vue"],  
64   // méthodes  
65   nomComplet() {  
66     return `${this.prenom} ${this.nom}`;  
67   }  
68 }  
69 console.log(personne.nomComplet());  
70  
71 // ---- Nouveautés class (PascalCase = maj à chaque mot )  
72 class Personne {  
73  
74   // Constructor  
75   constructor(nom, age) {  
76     this._nom = nom;  
77     this._age = age;  
78   }  
79   // Méthodes  
80   infosPersonne() {  
81     return `Bonjour je m'appelle ${this._nom} et j'ai ${this._age} ans.`;  
82   }  
83 }  
84  
85 let p1 = new Personne("Emilie", 35);  
86 console.log(p1);
```

# Ecmascript 2015+



La valeur retournée par le traitement **asynchrone** associé aux promesses est une **promesse** d'avoir cette valeur maintenant, ou plus tard ...



# Ecmascript 2015+



```
1 export const fnFilm = () => {
2   // ES6 : fonction fetch
3   // simplification d'écriture d'Ajax
4   // amélioration du process JS ( Promises )
5
6   // 1- Request Query (requête Infos):
7   // const _url = 'https://test.webjs.fr/films.json';
8   const _url = 'https://dev.webjs.fr/films.json';
9   // const _url='http://127.0.0.1:3000/json/films.json'; // en local avec un serveur json
10
11  // 2- Request Init Query
12  // définir la méthode (get ou post) ... ou patch ou put ou delete)
13  const _method = 'get';
14  // définir nos entêtes HTTP
15  const _headers = new Headers();
16
17  _headers.append('Content-Type', 'text/json'); //type mime (ex:image/png)
18  // _headers.append('Access-Control-Allow-Origin', 'cors');
19
20  // 3- Ecriture du FETCH
21  // fetch renvoie une promesse
22  fetch(
23    _url,
24    {
25      method: _method,
26      headers: _headers
27    }
28  ).then(
29    // si on arrive à avoir une réponse du serveur
30    // état onFullFilled
31    (responseHTTP) => {
32      // le then récupère une réponse 200 ok 404
33      console.warn('On Fullfilled du 1er Then');
34      console.log(responseHTTP);
35      console.log(responseHTTP.body);
```

# Les versions marquantes !



# Les versions marquantes Angular

- Version 9 (6 février 2020)
  - Nouveau compiler et nouveau moteur de rendu Ivy
  - Réduction notable de la taille des bundles (build)
  - Optimisation de la compilation (temps, debug) AOT  
*activé par défaut avec ng serve => remontée des erreurs en mode dev ...*
  - composant YouTube  
<https://github.com/angular/components/tree/main/src/youtube-player>
  - composant Google Maps  
<https://github.com/angular/components/blob/main/src/google-maps/README.md>

EXPLORATEUR

ÉDITEURS OUVERTS

- package.json TP-NG9

ANGULAR-RELEASES

- TP-NG9
  - e2e
  - src
    - protractor.conf.js
    - tsconfig.json
  - src
    - .editorconfig
    - .gitignore
    - angular.json
    - browserslist
    - karma.conf.js
    - package.json
    - README.md
    - tsconfig.app.json
    - tsconfig.json
    - tsconfig.spec.json
    - tslint.json

package.json X

TP-NG9 > package.json > ...

```
1 {
2   "name": "tp-ng9",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "test": "ng test",
9     "lint": "ng lint",
10    "e2e": "ng e2e"
11  },
12  "private": true,
13  "dependencies": {
14    "@angular/animations": "~9.1.13",
15    "@angular/common": "~9.1.13",
16    "@angular/compiler": "~9.1.13",
17    "@angular/core": "~9.1.13",
18    "@angular/forms": "~9.1.13",
19    "@angular/platform-browser": "~9.1.13",
20    "@angular/platform-browser-dynamic": "~9.1.13",
21    "@angular/router": "~9.1.13",
22    "rxjs": "~6.5.4",
23    "tslib": "^1.10.0",
24    "zone.js": "~0.10.2"
25  },
```

PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS

PS C:\Angular-releases> npx @angular/cli@9 new TP-NG9

EXPLORATEUR

ÉDITEURS OUVERTS

- tsconfig.json TP-NG9

ANGULAR-RELEASES

- TP-NG9
  - e2e
    - src
    - protractor.conf.js
    - tsconfig.json
  - src
  - .editorconfig
  - .gitignore
  - angular.json
  - browserslist
  - karma.conf.js
  - package.json
  - README.md
  - tsconfig.app.json
  - tsconfig.json
  - tsconfig.spec.json
  - tslint.json
  - ng-9.png

TP-NG9 > tsconfig.json > ...

```
1 {
2   "compileOnSave": false,
3   "compilerOptions": {
4     "baseUrl": "./",
5     "outDir": "./dist/out-tsc",
6     "sourceMap": true,
7     "declaration": false,
8     "downlevelIteration": true,
9     "experimentalDecorators": true,
10    "module": "esnext",
11    "moduleResolution": "node",
12    "importHelpers": true,
13    "target": "es2015",
14    "lib": [
15      "es2018",
16      "dom"
17    ]
18  },
19  "angularCompilerOptions": {
20    "fullTemplateTypeCheck": true,
21    "strictInjectionParameters": true
22  }
23 }
24
```



# Les versions marquantes Angular

- Version 10 (24 juin 2020)

- Mode `strict` (typage obligatoire)  
`ng new TP01 --strict`

*Rappels :*

- basic : pas de vérification de types
  - full : les props utilisées dans la Vue doivent être correctement définies dans le TS
- 
- nouveau datePicker dans Angular Material

ÉDITEURS OUVERTS

- × package.json TP-NG10
- ANGULAR-RELEASES
  - TP-NG9
  - TP-NG10
    - e2e
    - src
      - .browserslistrc
      - .editorconfig
      - .gitignore
      - angular.json
      - karma.conf.js
      - package.json
      - README.md
      - tsconfig.app.json
      - tsconfig.json
      - tsconfig.spec.json
      - tslint.json
      - ng-9-config.png
      - ng-9.png

TP-NG10 > package.json > ...

```
1 {
2   "name": "tp-ng10",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "test": "ng test",
9     "lint": "ng lint",
10    "e2e": "ng e2e"
11  },
12  "private": true,
13  "dependencies": {
14    "@angular/animations": "~10.2.4",
15    "@angular/common": "~10.2.4",
16    "@angular/compiler": "~10.2.4",
17    "@angular/core": "~10.2.4",
18    "@angular/forms": "~10.2.4",
19    "@angular/platform-browser": "~10.2.4",
20    "@angular/platform-browser-dynamic": "~10.2.4",
21    "@angular/router": "~10.2.4",
22    "rxjs": "~6.6.0",
23    "tslib": "^2.0.0",
24    "zone.js": "~0.10.2"
25  }
}
```

PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS

```
PS C:\Angular-releases> npx @angular/cli@10 new TP-NG10
```

# Les versions marquantes Angular

- Version 11 (17 novembre 2020)

- Mode `strict` est proposé lors de la création de l'appli

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL  PORTS

PS C:\Angular-releases> npx @angular/cli@11 new TP-NG11
Need to install the following packages:
  @angular/cli@11.1.2
Ok to proceed? (y) y
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated source-map-codec@1.4.8: Please use @jridgewell/source-map-codec instead
npm WARN deprecated @npmcli/ci-detect@1.4.0: this package has been deprecated, use `ci-info` instead
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() i
/math-random for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() i
/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/
npm WARN deprecated @schematics/update@0.1101.2: This was an internal-only Angular package up through Angular v
pendency.
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict (y/N) 
```

# Les versions marquantes Angular

- Version 12 (19 mai 2021)

- Abandon progressif de la compatibilité et support IE  
*validé avec NG13*
- Fin du support de **e2e, protractor**..
- TSlint déprécié depuis 2019 est remplacé par ESLint
- script watch qui remplace et améliore build
- build est désormais le mode prod

EXPLORATEUR

ÉDITEURS OUVERTS

- × package.json TP-NG12

ANGULAR-RELEASES

- TP-NG9
- TP-NG10
- TP-NG11
- TP-NG12
  - node\_modules
  - src
    - .browserslistrc
    - .editorconfig
    - .gitignore
    - angular.json
    - karma.conf.js
    - package-lock.json
    - package.json
    - README.md
    - tsconfig.app.json
    - tsconfig.json
    - tsconfig.spec.json
    - ng-9-config.png
    - ng-9.png
    - ng-10.png
    - ng-11.png

package.json

```
TP-NG12 > package.json > ...
1 {
2   "name": "tp-ng12",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "watch": "ng build --watch --configuration development",
9     "test": "ng test"
10  },
11  "private": true,
12  "dependencies": {
13    "@angular/animations": "~12.2.0",
14    "@angular/common": "~12.2.0",
15    "@angular/compiler": "~12.2.0",
16    "@angular/core": "~12.2.0",
17    "@angular/forms": "~12.2.0",
18    "@angular/platform-browser": "~12.2.0",
19    "@angular/platform-browser-dynamic": "~12.2.0",
20    "@angular/router": "~12.2.0",
21    "rxjs": "~6.6.0",
22    "tslib": "^2.3.0",
23    "zone.js": "~0.11.4"
24  }
}
```

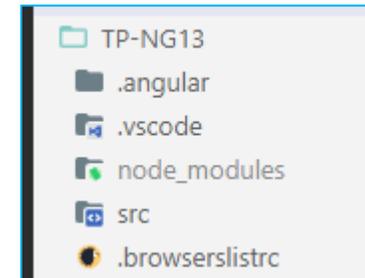
PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS

```
PS C:\Angular-releases> npx @angular/cli@12 new TP-NG12
```

# Les versions marquantes Angular

- Version 13 (3 novembre 2021)

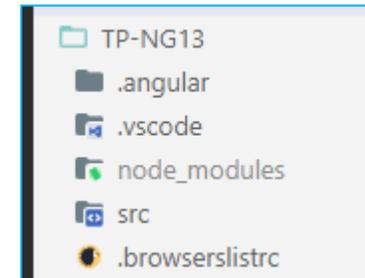
- Arrivée du cache .angular => accélération des temps de compilation
- démarrage en mode dév plus rapide
- Travail continu d'optimisation de IVY et ViewEngine



# Les versions marquantes Angular

- Version 14 (11 juin 2022)

- FormControl enfin typé
- composants *standalone* (*validé avec NG15*)
- optimisation des messages d'erreurs
- erreur banana in a box [( )] ou ([ ])

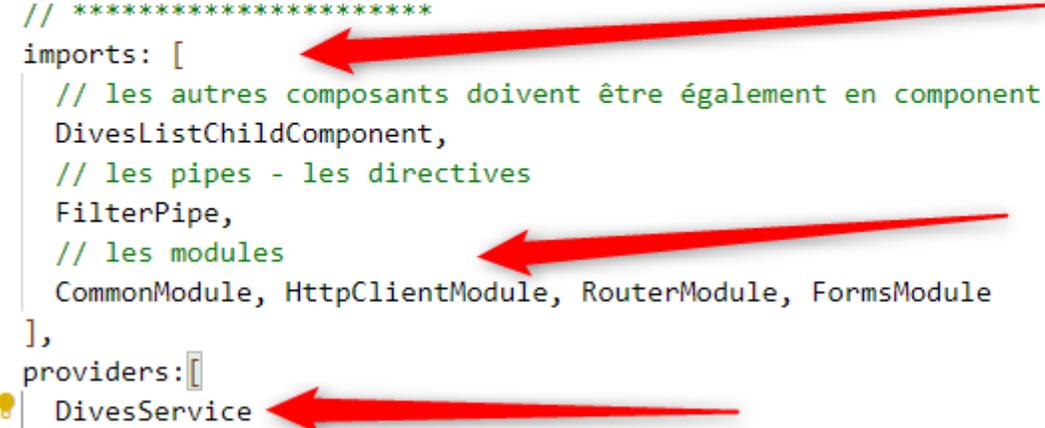


# Les versions marquantes Angular

- Version 15 (16 novembre 2022)

- composants **standalone**  
Prop : standalone:true
- Omission des ngModules ?
- API autonomes ?

```
@Component({
  selector: 'app-root',
  // *****
  standalone: true,
  templateUrl: './dives-list.component.html',
  styleUrls: ['./dives-list.component.scss'],
  // *****
  imports: [
    // les autres composants doivent être également en component standalone !!
    DivesListChildComponent,
    // les pipes - les directives
    FilterPipe,
    // les modules
    CommonModule, HttpClientModule, RouterModule, FormsModule
  ],
  providers: [
    DivesService
  ]
})
```



EXPLORATEUR

ÉDITEURS OUVERTS

- GRUPE 1
  - main.ts src
- GRUPE 2
  - dives-list.component.ts src\app\web...

STANDALONE-NG14

- .angular
- node\_modules
- src
  - app
    - webApp
      - components
        - dives-list
          - dives-list.component.html
          - dives-list.component.scss
          - dives-list.component.ts
        - dives-list-child
          - dives-list-child.component.html
          - dives-list-child.component.scss
          - dives-list-child.component.ts
        - dives.service.ts
        - ~~app.component.ts~~
        - ~~app.module.ts~~
        - app.component.html
        - app.component.scss
        - routes.ts

main.ts

```
src > main.ts > ...
1 import { enableProdMode } from '@angular/core';
2
3 import { environment } from './environments/environment';
4
5 import { bootstrapApplication } from '@angular/platform-browser';
6 import { DivesListComponent } from './app/webApp/components/dives-
7 import { provideRouter } from '@angular/router';
8 import { routes } from './app/routes';
9
10 if (environment.production) { enableProdMode();}
11
12 bootstrapApplication(
13   DivesListComponent ,
14   { providers:
15     [ provideRouter(routes)]}
16 );
17
18 // platformBrowserDynamic().bootstrapModule(AppModule)
19
20
```

# Les versions marquantes Angular

- Version 16 (12 mai 2023)

- Les **signals**
- detection change
- encore besoin de Zone.JS ?
- encore besoin de RXJS ?