





- Framework Javascript développé par Google
- Développé facilement et rapidement des solutions web



## Historique:

- AngularJS (Angular 1.x) 2010
- Angular2 (2016) Révolution
- Angular4 (2017) TypeScript 2
- Angular5 (2017) HttpClient
- Angular6 (2018) Toolchain+
- Angular7 (2018) Toolchain,drag and drop,virtual scrolling



# Les outils:

**Node js (8 ou 10):** <https://nodejs.org/en/>

**npm:** <https://www.npmjs.com/>

**Angular Cli:** <https://angular.io/cli>



# Les outils:

```
npm install -g @angular/cli
```



# Utilisation angular cli

`ng new monAppli`



```
C:\Users\Gadpasou\angularApp>ng new monAppli
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE monAppli/angular.json (3786 bytes)
CREATE monAppli/package.json (1308 bytes)
CREATE monAppli/README.md (1025 bytes)
CREATE monAppli/tsconfig.json (435 bytes)
CREATE monAppli/tslint.json (2824 bytes)
CREATE monAppli/.editorconfig (246 bytes)
CREATE monAppli/.gitignore (576 bytes)
CREATE monAppli/src/favicon.ico (5430 bytes)
CREATE monAppli/src/index.html (295 bytes)
CREATE monAppli/src/main.ts (372 bytes)
CREATE monAppli/src/polyfills.ts (3571 bytes)
CREATE monAppli/src/test.ts (642 bytes)
CREATE monAppli/src/styles.css (80 bytes)
CREATE monAppli/src/browserslist (388 bytes)
CREATE monAppli/src/karma.conf.js (980 bytes)
CREATE monAppli/src/tsconfig.app.json (166 bytes)
CREATE monAppli/src/tsconfig.spec.json (256 bytes)
CREATE monAppli/src/tslint.json (314 bytes)
CREATE monAppli/src/assets/.gitkeep (0 bytes)
CREATE monAppli/src/environments/environment.prod.ts (51 bytes)
CREATE monAppli/src/environments/environment.ts (662 bytes)
CREATE monAppli/src/app/app-routing.module.ts (245 bytes)
CREATE monAppli/src/app/app.module.ts (393 bytes)
CREATE monAppli/src/app/app.component.html (1152 bytes)
CREATE monAppli/src/app/app.component.spec.ts (1101 bytes)
CREATE monAppli/src/app/app.component.ts (212 bytes)
CREATE monAppli/src/app/app.component.css (0 bytes)
CREATE monAppli/e2e/protractor.conf.js (752 bytes)
CREATE monAppli/e2e/tsconfig.e2e.json (213 bytes)
CREATE monAppli/e2e/src/app.e2e-spec.ts (300 bytes)
CREATE monAppli/e2e/src/app.po.ts (204 bytes)
npm WARN deprecated circular-json@0.5.9: CircularJSON is in maintenance only, flattened is its successor.

> node-sass@4.10.0 install C:\Users\Gadpasou\angularApp\monAppli\node_modules\node-sass
> node scripts/install.js

Cached binary found at C:\Users\Gadpasou\AppData\Roaming\npm-cache\node-sass\4.10.0\win32-x64-67_binding.node
```



.git	15/01/2019 09:29	Dossier de fichiers	
e2e	15/01/2019 09:26	Dossier de fichiers	
node_modules	15/01/2019 09:29	Dossier de fichiers	
src	15/01/2019 09:26	Dossier de fichiers	
.editorconfig	15/01/2019 09:26	Fichier source Edit...	1 Ko
.gitignore	15/01/2019 09:26	Fichier source Git l...	1 Ko
angular.json	15/01/2019 09:26	Fichier source JSON	4 Ko
package.json	15/01/2019 09:26	Fichier source JSON	2 Ko
package-lock.json	15/01/2019 09:29	Fichier source JSON	379 Ko
README.md	15/01/2019 09:26	Fichier source Mar...	2 Ko
tsconfig.json	15/01/2019 09:26	Fichier source JSON	1 Ko
tslint.json	15/01/2019 09:26	Fichier source JSON	3 Ko

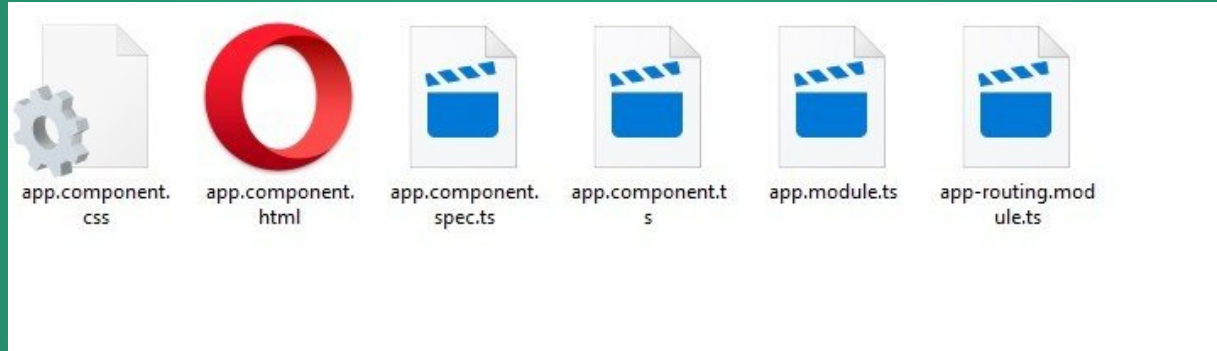




app	15/01/2019 09:26	Dossier de fichiers	
assets	15/01/2019 09:26	Dossier de fichiers	
environments	15/01/2019 09:26	Dossier de fichiers	
browserslist	15/01/2019 09:26	Fichier	1 Ko
favicon.ico	15/01/2019 09:26	Icône	6 Ko
index.html	15/01/2019 09:26	Opera Web Docu...	1 Ko
karma.conf.js	15/01/2019 09:26	Fichier de JavaScript	1 Ko
main.ts	15/01/2019 09:26	Fichier TS	1 Ko
polyfills.ts	15/01/2019 09:26	Fichier TS	4 Ko
styles.css	15/01/2019 09:26	Document de feui...	1 Ko
test.ts	15/01/2019 09:26	Fichier TS	1 Ko
tsconfig.app.json	15/01/2019 09:26	Fichier source JSON	1 Ko
tsconfig.spec.json	15/01/2019 09:26	Fichier source JSON	1 Ko
tslint.json	15/01/2019 09:26	Fichier source JSON	1 Ko

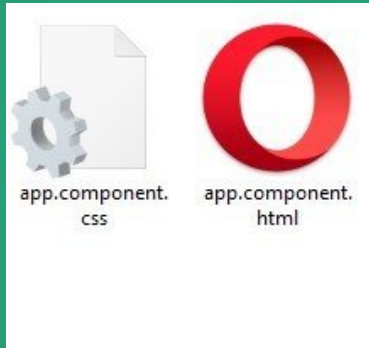


## Liste des fichiers sources générés





# Vue





# Composant



app.component.  
spec.ts



app.component.t  
s



# Modules



app.module.ts



app-routing.module.ts



# TypeScript

- Langage compilé et typé de microsoft
- Libre et opensource
- Transcompilé en javascript
- <https://www.typescriptlang.org>



# Typescript

Langage



# TypeScript

- Permet le typage forte dans javascript
- Type existant:  
number,string,boolean,any,array





# TypeScript

Exemple:

```
nombre :number;
```

```
test : boolean;
```

```
tab : number[] = [1,2,3];
```

```
tab : Array<number> = [1,2,3]
```



# TypeScript: Tuples

-Definition de tuple:

```
dictionnaire: [number,string];
```

```
dictionnaire =[1,"Hello"];
```



# TypeScript: Enum

```
enum Color { rouge,vert,bleu }
```

```
couleur : Color = Color.rouge;
```



# TypeScript: Any

```
whatever : any;
```

```
whatever=1;  
whatever="Hello";
```



# TypeScript: attributs

-Déclaration d'attribut (variable de classe)  
[public/private/protected] nom:type

Exemple: nombreDeClient : number=5;



# TypeScript: constructeur et attributs

```
nom:string;  
constructor(){  
  this.nom ="Hello";  
}
```

Le this n'est pas facultatif!



# TypeScript: var, let, const

Exemple:

```
let hello : string = "Hello";  
var hello1 : string = "Hello";  
const hello2: string ="Hello";
```



# TypeScript: classe et interface

```
class nom {
```

```
}
```

```
interface nom {
```

```
}
```

Relation heritage possible via extends

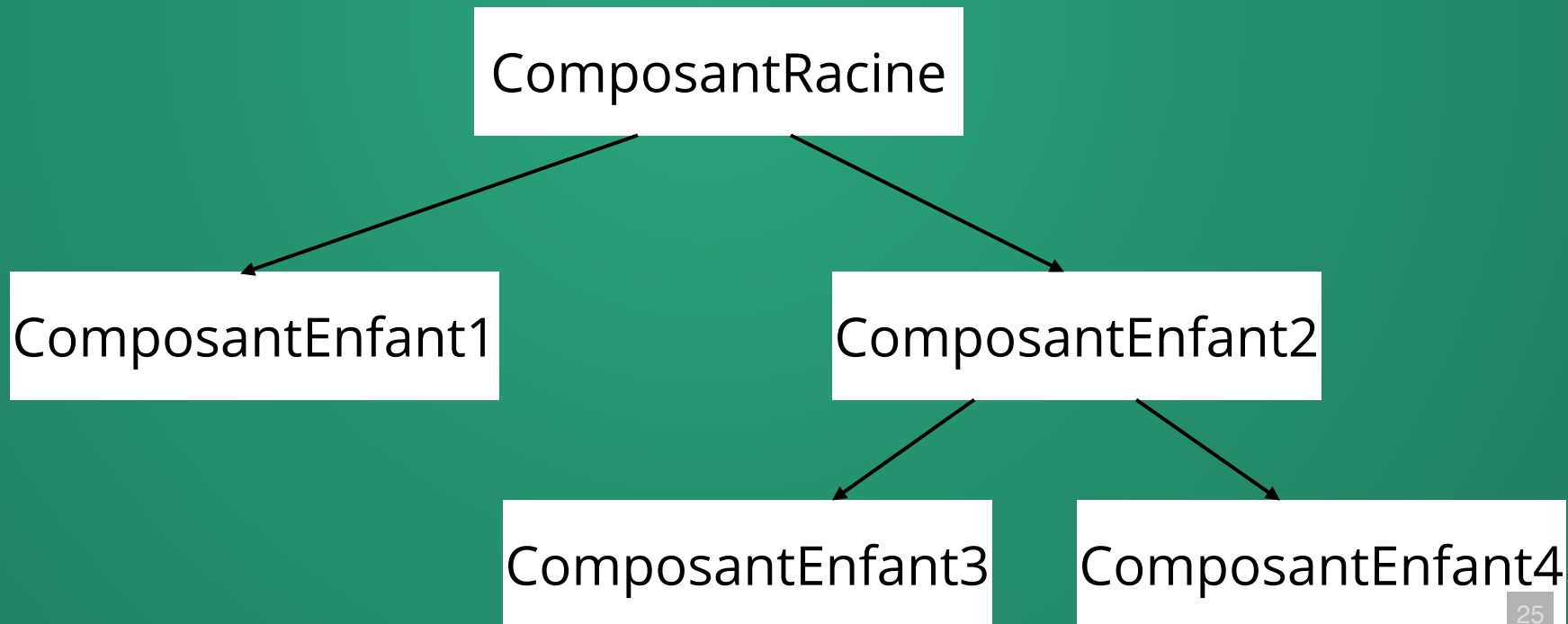




# Angular: Component

```
import {Component} from '@angular/core';
```

Decorator @Component





# Angular: Component

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title:string = 'Hello';
  nombreDeClient:number=5;
  constructor(){

  }
}
```



# Angular: Component

```
import { Component } from '@angular/core';
```



# Angular: Component

```
import { Component } from '@angular/core';

@Component({

})

export class AppComponent {
  title:string = 'Hello';
  nombreDeClient:number=5;
  constructor(){

  }
}
```



# Angular: Component

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',

export class AppComponent {
  title:string = 'Hello';
  nombreDeClient:number=5;
  constructor(){

}

}
```



# Angular: Component

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
})

export class AppComponent {
  title:string = 'Hello';
  nombreDeClient:number=5;
  constructor(){

  }
}
```



# Angular: Component

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title:string = 'Hello';
  nombreDeClient:number=5;
  constructor(){

  }
}
```



# Angular: Component

One way binding: La double moustache

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  title:string = 'Hello';  
  nombreDeClient:number=5;  
  
  constructor(){  
  
  
  
  
  
  
  
  
  
}
```

Injection de component vers vue

```
<h1>{{title}}</h1>  
<h1>{{nombreDeClient}}</h1>  
<app-heroes></app-heroes>
```





# Angular: Component

One way binding: les crochets

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  title:string = 'Hello';  
  status:boolean=true;  
  nombreDeClient:number=5;  
  
  constructor(){  
  
  
  }  
}
```

Injection de component vers vue

```
<h1>{{title}}</h1>  
<h1>{{nombreDeClient}}</h1>  
<button [disabled]="status" >Save</button>  
<app-heroes></app-heroes>
```



# Angular: Component

One way binding: les crochets

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  title:string = 'Hello';  
  status:boolean=true;  
  nombreDeClient:number=5;  
  
  constructor(){  
  
  
  }  
}
```

Injection de component vers vue

```
<h1>{{title}}</h1>  
<h1>{{nombreDeClient}}</h1>  
<button [disabled]="status" >Save</button>  
<app-heroes></app-heroes>
```



# Angular: Component

Two way binding: les parenthèses crochétées

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title:string = 'Hello';
  status:boolean=true;
  nombreDeClient:number=5;

  constructor(){

  }
}
```

```
1 <h1>{{title}}</h1>
2 <h1>{{nombreDeClient}}</h1>
3 <button [disabled]="status" >Save</button>
4 <input type="checkbox" name="info" [(ngModel)]="status" />
5 <app-heroes></app-heroes>
```

Injection de composant vers vue



# Angular: Component

Two way binding: les parenthesés crochétés: Import!

```
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3  import { FormsModule } from '@angular/forms';
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { HeroesComponent } from './heroes/heroes.component';
7
8  @NgModule({
9    declarations: [
10     AppComponent,
11     HeroesComponent
12   ],
13   imports: [
14     BrowserModule,
15     AppRoutingModule,
16     FormsModule
17   ],
18   providers: [],
19   bootstrap: [AppComponent]
20 })
21 export class AppModule { }
22
```



# Angular: Component

eventbinding: les parentheses

```
1 <h1>{{title}}</h1>
2 <h1>{{nombreDeClient}}</h1>
3 <button [disabled]="status" (click)="change()" >Save</button>
4 <input type="checkbox" name="info" [(ngModel)]="status" />
5
6
7 <app-heroes></app-heroes>
8
```

```
23 change(): void {
24   this.status=!this.status;
25
26 }
```



# Defi Component

Créer un component permettant de gérer une todo list  
(création et affichage)

ng generate component todolist

```
<ul>
  <li *ngFor="let todo of todos">
    {{todo}}
  </li>
</ul>
```



# Template: pipe

-Formatage via | (pipe)

Exemple: `<p> today {{annif | date}} </p>`



# Directives:interagir avec le DOM

3 types de directives:

- Directive d'attributs : [ngClass]
- Directive structurelle: \*ngFor (unique!)
- Les composants: des directives complexe:

@Output() sortie =new EventEmitter<type>();

Crée un event (sortie) dans le component





# Directives:interagir avec le DOM

Input et Output

Entrée: @Input()

Sortie: @Output() sortie =new EventEmitter<type>();

Crée un event (sortie) dans le component



# Template: pipe

-Peuvent être paramétrés:

Exemple `{{annif | date:'dd/MM/yyyy' }}`

ou encore

`{{annif | date:format }}`



# Template: pipe

Les plus utilisé: uppercase, lowercase,date,percent,slice

Création possible de nos propres custom pipe via l'annotation  
@Pipe et l'interface PipeTransform



# Template: async pipe

Pattern observer appliqué à un pipe.

Exemple:

Affichage d'un message à réception de celui ci



# Template: Angular Material

<https://material.angular.io/guide/getting-started>



# Services

Composant: Manipulation des données

Service: Logique métier/technique/fonctionnel

Service est une boîte noire utilisé par le composant



# Template: Angular Material

Drag and drop: nouveauté angular 7

<https://material.angular.io/cdk/drag-drop/overview>

<https://material.angular.io/cdk/drag-drop/overview>



# Defi Angular Material

Décorer une application déjà réalisé de composants material





# Services

Un service angular est une classe  
Décorée @Injectable()

Les services sont utilisés via Injection au besoin via le constructor



# Services

Service Synchrone  
Service Asynchrone



# Services

Service Asynchrone:

- Promise: Gestion d'un event via succès ou erreur
- Observable: gestion d'un flux d'event, annulable, possède des méthodes de parse avancées.



# Services: Promise

Promesse d'exécution d'une fonction une fois la méthode asynchrone terminée:

```
promise.then( succes, error)
```

Exemple:

```
monService.getQuelqueChose().then(  
  (succes) => {}, error => {});
```



# Services: Observable

**RxJS**

<https://rxjs-dev.firebaseapp.com/>

<https://www.learnrxjs.io/operators/creation/>

Création d'un observable via RxJS ( of,interval,from )

Lecture d'un observable: obs.subscribe(instruction);



# Routing

App Routing module



# Services: Subject

```
private subject = new Subject<any>();

constructor(){

    const source = interval(1000);
    const subscribe = source.subscribe(val => {
        this.sendMessage(" "+val);
        console.log(val);});

}

sendMessage(message: string) {
    this.subject.next(message);
}

clearMessage() {
    this.subject.next();
}

getMessage(): Observable<any> {
    return this.subject.asObservable();
}
```



# Services: Filter

```
const source = from([1, 2, 3, 4, 5]);  
  
const example = source.pipe(filter(num => num % 2
```





# Services: Pipe

```
const source = from([1, 2, 3, 4, 5]);  
  
const example = source.pipe(filter(num => num % 2 === 0));
```



# Services: Map

```
const nums = of(1, 2, 3);  
  
const squareValues = map((val: number) => val * val);  
  
const squaredNums = squareValues(nums);  
  
squaredNums.subscribe(x => console.log(x));
```



# Routing

## App Routing module

```
const routes: Routes = [  
  { path: 'chemin', component: MonComponent }  
];
```

## app-root

```
<router-outlet></router-outlet>
```



# Routing

## Vue html

```
<a routerLink="/chemin" > lien </a>
```



# Routing

## Route avec param

```
{ path: 'user/:name', component: UserComponent }
```

## Récupération du param via injection

```
constructor(  
  private route: ActivatedRoute){}  
  
this.route.snapshot.paramMap.get('name');
```



# Routing

## ActivatedRoute: Service injecté

```
constructor( private ar:ActivatedRoute)
```

```
constructor(  
  private route: ActivatedRoute){}  
  
this.route.snapshot.queryParamMap.get( 'name' );
```



# HttpClient

Nouvelle librairie depuis Angular 6

```
HttpClientModule
```

Utilisation via injection

```
constructor( private http : HttpClient )
```



# HttpClient

## Récupération d'un Observable

```
http.get(url)
```

## Récupération Typée

```
http.get<Utilisateur>(url)
```





# HttpClient

## Souscription

```
http.get(url).subscribe( val => quelqueChose)
```



# Deploiement

## Build

```
ng build
```

.git	15/01/2019 09:29	Dossier de fichiers	
dist	23/01/2019 18:04	Dossier de fichiers	
e2e	15/01/2019 09:26	Dossier de fichiers	
node_modules	23/01/2019 17:53	Dossier de fichiers	
src	15/01/2019 09:26	Dossier de fichiers	
.editorconfig	15/01/2019 09:26	Fichier source Edit...	1 Ko
.gitignore	15/01/2019 09:26	Fichier source Git l...	1 Ko
angular.json	20/01/2019 17:26	Fichier source JSON	4 Ko
package.json	21/01/2019 08:28	Fichier source JSON	2 Ko
package-lock.json	20/01/2019 17:26	Fichier source JSON	381 Ko
README.md	15/01/2019 09:26	Fichier source Mar...	2 Ko
tsconfig.json	15/01/2019 09:26	Fichier source JSON	1 Ko
tslint.json	15/01/2019 09:26	Fichier source JSON	3 Ko



# Deploiement: optimisation

Appel `enableProdMode()` dans `main.ts`

Build

```
ng build --prod
```



# Deploiement:

- Attention au base ref
- Attention aux regles de redirections
- Attention aux CORS (Cross origin resource sharing) Coté serveur