Module Angular

Parti 1

Maxime Tournier

Sommaire

Présentation

Angular est un FrameWork

Traduction

Framework = Cadre de travail

En tant que développeur on fait souvent la meme chose

Exemple:
Valider les formulaires — Gerez la navigation — traiter les erreurs

Et pour regler ce problème au lieu d'allez recupere les fonction d'autre projet on à créer les framework

Avantage : Tout le monde travaille sur les même fonction

Présentation

Pour comprendre le fonctionnement d'angular

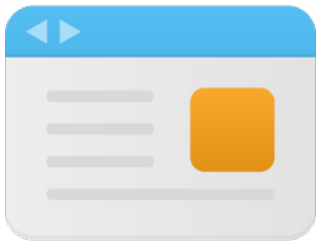
Il faut comprendre le web aujourd'hui

Site Web / Application Web

Il s'agit bien de deux chose différente

Présentation

Avant ça comment ça marche :

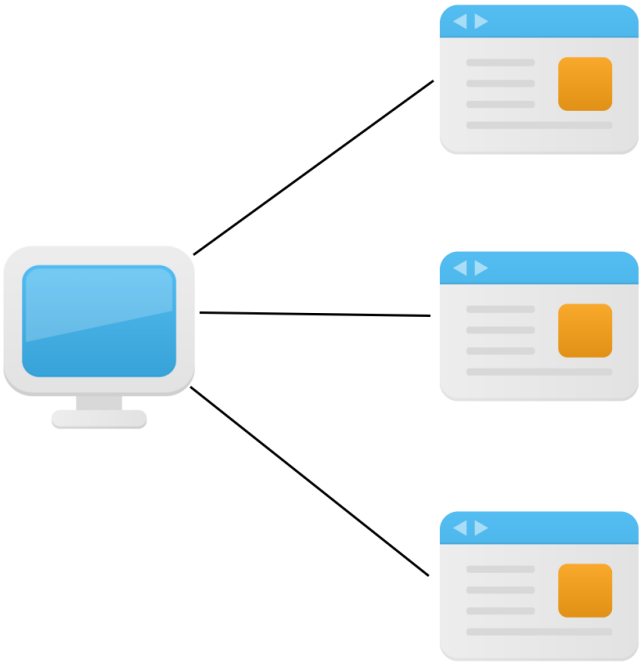


- Fichier client (CSS, HTML, JS)

- Fichier serveur (php, java) (ici qu'on fait des requetes sql)

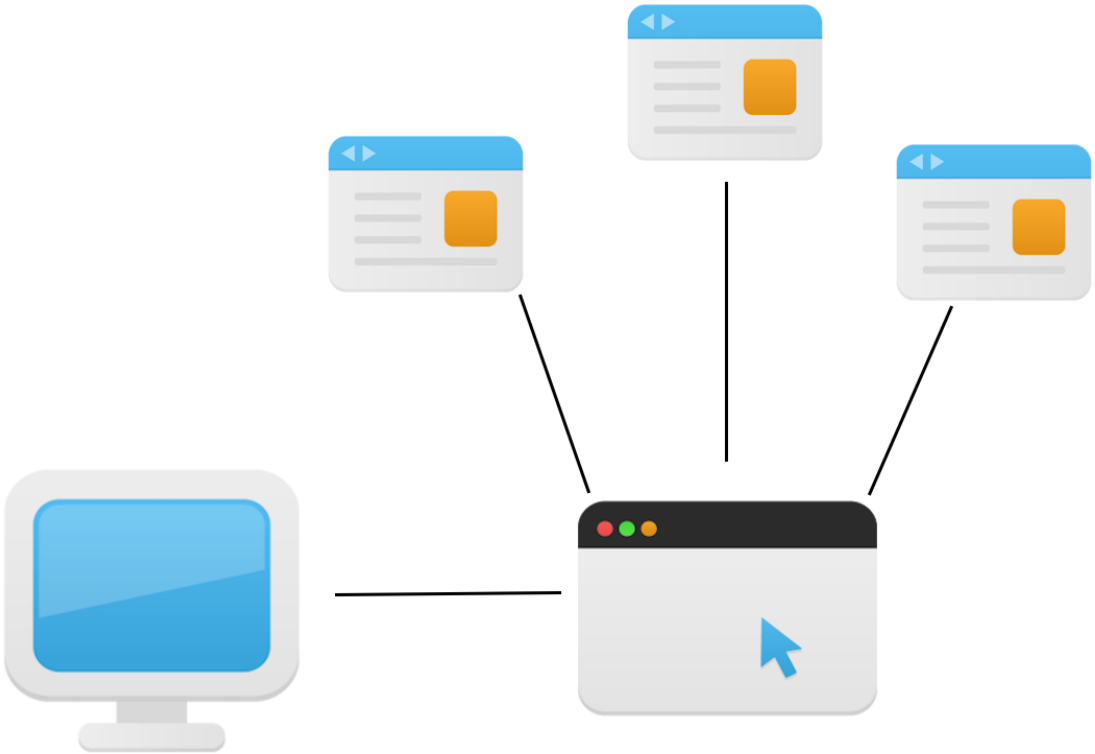
Présentation

Et donc un site web fonctionne comme ça :



Présentation

Et une application web marche comme ça:



Présentation

Resumé :

le client doit faire une requête au serveur a chaque fois dans un site web

alors que dans une application web, nous allons deleguer le travaille a JavaScript qui vas désactivé ou activé une parti du code html/css

Information

Cette méthode s'appel SPA (Single Page Application)

Histoire

AngularJS n'est pas egal a Angular

AngularJS à été créer par Google

AngularJS avait comme architecture MVC

AngularJS n'est plus maintenu depuis 2018

MVC

Module Vue Controlleur (Comme Symfony)

Histoire

Angular est un framework orienté composant

Nous allons codé une multitude de petit composant qui formeront une application

Un composant est une parti du site qui fonctionne de manière autonome sur une applicaiton

Histoire

Angular est il difficiel à apprendre ? :

Angular à une réputation de framework difficile

Car Angular est basé sur
TypeScript un languge basé sur Javascript qui permet de Typé vos variable

Angular est basé sur la version de Javascript ES6

Information

Regarder la documentation S'IL VOUS PLAIT
Angular.io

Démarrage Angular

1. Installer un environnement de developpement
2. Genere un socle angular (Grâce à Angular-CLI)
3. Nous allons jouer avec le composant racine =)

Information

un composont c'est comme une parti de l'application web. Mais il y a un premier composant a l'origine de tout, que nous appelons le composant racine

Démarrage Angular

Ce qu'on a besoin :

D'abors on vas avoir besoin du moteur javascript (NodeJS) et d'un gestionnaire de packet (NPM)

D'un Editeur de code TypeScript

Un outil qu'on appel Angular CLI (Qui vas nous permet de gérez notre projet Angular)

NodeJS et NPM

NodeJS et NPM sont utiliser pour la majorité des projet JavaScript moderne

NodeJS permet d'excute du code JS coté serveur

NPM
permet de gere les dépence de paquet de javascript (angular est un paquet)

NodeJS integre par defaut NPM

NodeJS et NPM

Pour installer NodeJS (et donc aussi NPM)

il vous suffit d'installer sur le site officiel https://nodejs.org/fr

Information

Prenez la version LTS

IDE

On vas avoir besoin d'un IDE

Un IDE est un environnement de developpement (Editeur de code amélioré)

- Visual Studio Code : Recommander par Mircorsoft (Gratuit)
- WebStorm : Beaucoup plus puissant et recommander par Google, de la suite JetBrain (Payant)

Choissier celui que vous préfèrer ou que vous êtes le plus a l'aise

Information

La suite JetBrain (WebStorm) est gratuit pendant tout au long de votre formation, inscription sur leurs site internet

Terminal

Pour votre terminal :

Souvent il peut avoir des problème avec le terminal Windows

Essayer de toujours d'utiliser PowerShell

Démarrage Angular

Activité :

- Installer NodeJS et NPM sur le site officiel (https://nodejs.org/fr)
Prenez la version LTS pour NodeJS

- Installer un IDE : Visual Studio Code ou WebStorm

Pour tester si NodeJS et NPM son bien installer, Ouvrez un terminal

```
$ node -v
v18.15.0
$ npm -v
9.5.0
```

TypeScript

TypeScript est une surcouche de JavaScript

TypeScript est developed par Microsoft depuis 2012

TypeScript support les dernier version de JS ES6

TypeScript

TypeScript a besoin d'être compilé (Build) pour fonctionner

Il sera transformé en un fichier JS

L'extension de TS est .ts

Pour compilé TypeScript on excutera cette commande

```
tsc file.ts
```

Cette commande nous créara un fichier JS du nom de file.js

Typescript ajoute la déclaration de variable:

```
myVar: number;
myVarAssigned: string = "Maxime";
myCIient: IClient = new Client();
myPersons: Array<Person> = []
myObject: any;
```

Typescript ajoute la déclaration de variable:

number = Int ou Float
string = Une chaint de caracter
IClient = Un object de type ICLient
Array<Person> = Un tableau de "Person"
any = Autre

TypeScript

Activité :

- Installer de quoi compile TypeScript avec cette comande

```
npm install -g typescript
```

Activité :

- Refaire l'activité 1 du rappel JavaScript
En typant les variable avec TypeScript

- Puis compilé le fichier TypeScript et tester le site

Activité :

Mini-Quiz à Choix Multiples

Premier pas avec Angular

Angular-CLI :

Cette application vas installer tout ce qu'on a besoin
en une seul ligne de commande

et avec cette même outils on vas pouvoir piloté certaine fonctionnalité
d'angular directement en ligne de command

Information

Ce n'est pas la seul méthode pour démarré un projet Angular, mais bien la plus connue

Premier pas avec Angular

Activité :

- Installer Angular-CLI

Grace à NPM :

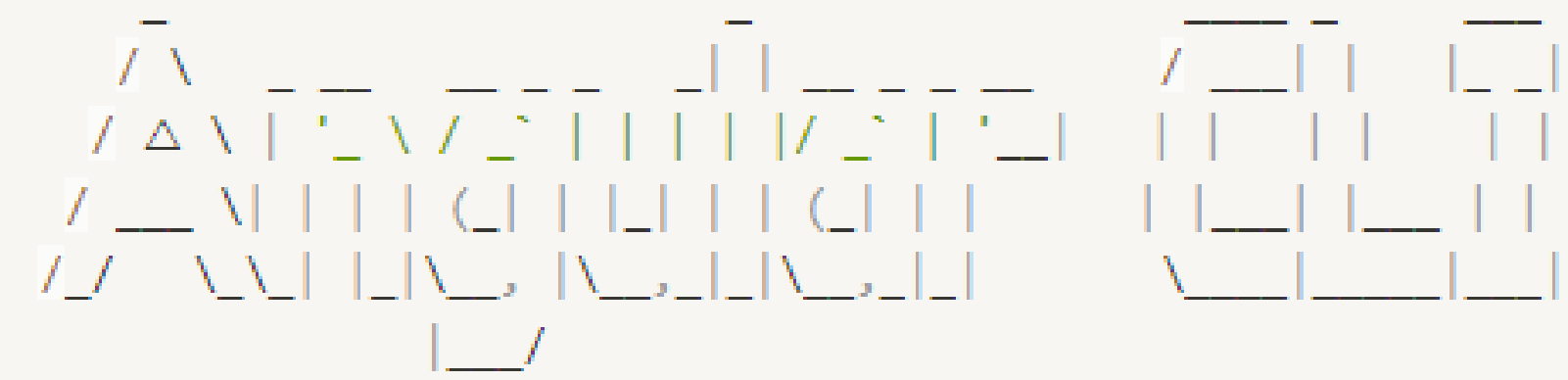
npm install -g @angular/cli

Premier pas avec Angular

Activité :

- Verifier l'installation de Angular-CLI

```
$ ng version
```



```
Angular CLI: 16.2.0
Node: 18.15.0
Package Manager: npm 9.5.0
```

Premier pas avec Angular

Nous alons generez notre premier projet Angular

```
$ ng new pokemon-app --minimal --style=css
```

Le nom du projet = pokemon-app
le type de ficher de style = css

Premier pas avec Angular

Activité :
Créer notre projet Angular

```
$ ng new pokemon-app --minimal --style=css
```

Premier pas avec Angular

```
$ ng new ng-pokemon-app --minimal --style=css
? Would you like to add Angular routing? Yes
CREATE ng-pokemon-app/angular.json (2989 bytes)
CREATE ng-pokemon-app/package.json (790 bytes)
CREATE ng-pokemon-app/README.md (1066 bytes)
CREATE ng-pokemon-app/tsconfig.json (901 bytes)
CREATE ng-pokemon-app/.gitignore (548 bytes)
CREATE ng-pokemon-app/tsconfig.app.json (263 bytes)
CREATE ng-pokemon-app/.vscode/extensions.json (130 bytes)
CREATE ng-pokemon-app/.vscode/launch.json (297 bytes)
CREATE ng-pokemon-app/.vscode/tasks.json (531 bytes)
CREATE ng-pokemon-app/src/main.ts (214 bytes)
CREATE ng-pokemon-app/src/favicon.ico (948 bytes)
CREATE ng-pokemon-app/src/index.html (298 bytes)
CREATE ng-pokemon-app/src/styles.css (80 bytes)
CREATE ng-pokemon-app/src/app/app-routing.module.ts (245 bytes)
CREATE ng-pokemon-app/src/app/app.module.ts (393 bytes)
CREATE ng-pokemon-app/src/app/app.component.ts (1489 bytes)
CREATE ng-pokemon-app/src/assets/.gitkeep (0 bytes)
✓ Packages installed successfully
```

Premier pas avec Angular

Que fait "Ng New"

- Il créer tout le socle d'Augular
- Il installe toute les dépendance d'Angular (Il effectue des simple NPM INSTALL)
- Il initialise un .git
- Setup des confugurations pour des IDE

Présentation du socle

Premier pas avec Angular

Lancé Angular :

```
$ ng server
? Would you like to share pseudonymous usage data about this project
at Google under Google's Privacy Policy at https://policies.google.co
details and how to change this setting, see https://angular.io/analyt
Global setting: enabled
Local setting: disabled
Effective status: disabled
✓ Browser application bundle generation complete.

Initial Chunk Files   | Names         |  Raw Size
vendor.js             | vendor        |   2.33 MB |
polyfills.js          | polyfills     | 333.19 kB |
styles.css, styles.js | styles        | 230.46 kB |
```

Inspection premier composant

Inspection du module racine

Configuration TypeScript

```
8: "strict": true,
9: "strictPropertyInitialization": false,
```

Composants

Quest-ce qu'un composants ? :

Un composants est une parti de l'écran

Cette portion de l'écran qu'on vas controller,
on appel ça une vue (ligne 5)

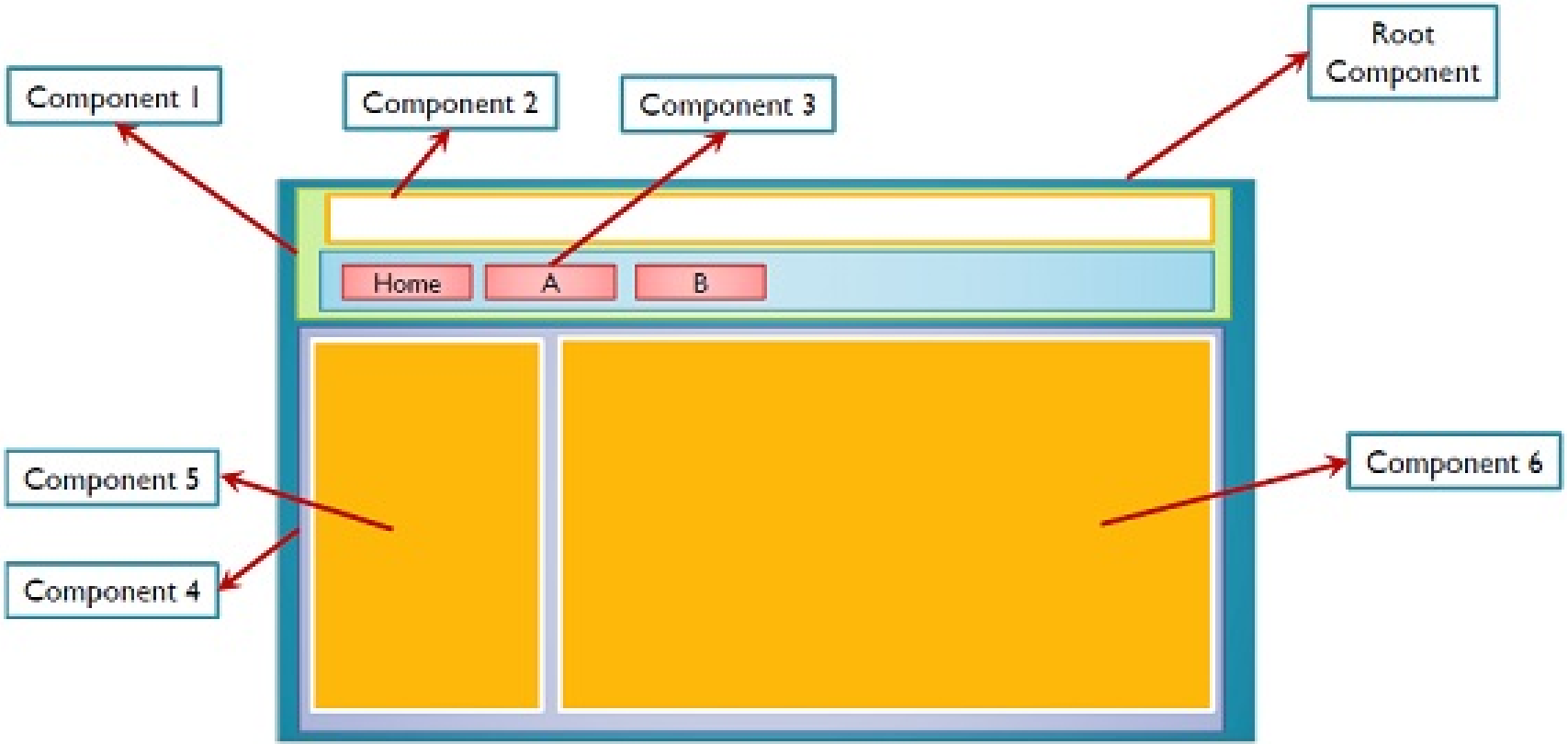
cette vue est deffini dans le template et ça peut-etre beaucoup chose

Donc un composant est une class + une vue

Composants

Quest-ce qu'un composants ? :

le template (la vue) c'est le rendu pour les utilisateurs



Composants

Première variable :

```
export class AppComponent {
  pokemons = ['Moustillons', 'Salameche', 'tortank'];
}
```

Composants

Activité :

Afficher un pokémon dans votre template !

Composants

Affichage de notre première variable :

```
template: `<h1> Welcome to {{ pokemons[1] }}! </h1>`
```

Composants

Activité :

- Créer deux variable de type number
- Mettez deux nombre aléatoire
- Afficher dans le template le resultat de l'addition de ces deux nombre

Information

Pensez a typé vos variable

Composants

Cycle de vie :

ngOnChanges	C'est la méthode appelée en premier lors de la création d'un composant, avant même *ngOnInit*, et à chaque fois que Angular détecte que les valeurs d'une propriété du composant sont modifiées. La méthode reçoit en paramètre un objet représentant les valeurs actuelles et les valeurs précédentes disponibles pour ce composant.
ngOnInit	Cette méthode est appelée juste après le premier appel à *ngOnChanges*, et elle initialise le composant après que Angular ait initialisé les propriétés du composant.
ngDoCheck	On peut implémenter cette interface pour étendre le comportement par défaut de la méthode *ngOnChanges*, afin de pouvoir détecter et agir sur des changements qu'Angular ne peut pas détecter par lui-même.
ngAfterViewInit	Cette méthode est appelée juste après la mise en place de la vue d'un composant (et des vues de ses composants fils s'il en a).
ngOnDestroy	Appelée en dernier, cette méthode est appelée avant qu'Angular ne détruise et ne retire du DOM le composant. Cela peut se produire lorsqu'un utilisateur navigue d'un composant à un autre par exemple. Afin d'éviter les fuites de mémoire, c'est dans cette méthode que nous effectuerons un certain nombre d'opérations afin de laisser l'application « propre » (nous détacherons les gestionnaires d'événements par exemple).

Composants

NgOnInit :

On vas en chargant ce composont
vouloir afficher le tableau pokemon dans un console.log

On vas devoir importé la fonctionnalité nessesaire

Composants

NgOnInit :

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

Composants

NgOnInit :

```
export class AppComponent implements OnInit {
  pokemons = ['Moustillons', 'Salameche', 'tortank'];
  ngOnInit() {
  }
}
```

Composants

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

@Component({
  selector: 'app-root',
  template: `<h1> Welcome to {{ pokemonList[1] }}! </h1>`
})
export class AppComponent implements OnInit {
  pokemonList = ['Moustillons', 'Salameche', 'tortank'];

  ngOnInit(){


    console.table(this.pokemonList);


  }
}
```

Composants

Activité :

- Créer une alert (Function JavaScript) au démarrage
- Cette alert dira "bonjour" au Utilisateur

- Puis faite un "console.log" d'une nouvelle variable de type String
- La variable dira "Vous n'est pas polie"

- Afficher dans un console.log l'additions de l'activité precedente

Composants

Interaction Utilisateur :

On vas simplement créer une function qui affiche notre pokémon dans la console

Une fois cette fonctionne créer, on vera dans le template comment activé cette fonction

Interaction Utilisateur :

```
export class AppComponent implements OnInit {
  pokemonList = ['Moustillons', 'Salameche', 'tortank'];

  ngOnInit(){
    console.table(this.pokemonList);
  }

  selectPokemon(pokemonName: string) {
    console.log(`vous avez cliqué sur le pokémon ${pokemonName}`)
  }
}
```

Interaction Utilisateur :

Pour tester cette fonction sans le template, on vas lancer la fonction au lancement du composant

```
ngOnInit(){
    console.table(this.pokemonList);
    this.selectPokemon('Salameche');
  }
```

Composants

Activité :

- Créer une fonction "multiplication"
- Cette fonction besoin de deux variable number pour foncitonner
- Cette fonction doit multiplier les deux variable et l'afficher dans un console.log

- Créer cette fonction et lancé la fonciton au démarrage du composant en utilisant les deux variable "data1" et "data2"

Composants

Gerez de la donnée :

nous allons créer deux fichier

- ./src/app/pokemons.ts
- ./src/app/mock-pokemons-list.ts

Le premier est un model (Class D'Object en php)

Le Deuxième un tableau basé sur le model qui contient de la donnée

Composants

Activité :

- Afficher en titre "Liste de pokémon"
- Recupéré dans la pokemonList toute les donnée du mock
- Dans la fonction selectPokemon recupéré tout un model pokémon plutot qu'un String

Template

Laisser la vue au même endroit que la logique
C'est absolument pas recommander

Alors dans notre composant on vas changer ça,
en créant un ficher appart

Template

On vas créer un fichier :
- ./src/app/app.component.html

et modifier le composant :

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

Template

Dans app.component.html :

```
<h1>Liste des pokémon</h1>
<p>Paragraphe</p>
<p>Paragraphe</p>
<p>Paragraphe</p>
```

Template

On peut constater que la vue ce charge bien par le biais de notre fichier html

Et il y a bien toujours notre logique dans la console JavaScript

Donc maintenant on a fichier Logique (Composant) et un fichier Graphique (Template)

c'est ce systeme qu'on vas utiliser tout au long du developpement Angular

Template

L'interpolation :

Nous avons actuellement un fichier HTML static

Et nous voulons le rendre dynamique

On vas utiliser la syntaxe d'interpolation

Template

L'interpolation :

```
<h1>Liste des pokémon</h1>
<p>{{ pokemonList[0] }}</p>
<p>{{ pokemonList[1] }}</p>
<p>{{ pokemonList[2] }}</p>
<p>{{ pokemonList[3] }}</p>
<p>{{ pokemonList[4] }}</p>
<p>{{ pokemonList[5] }}</p>
<p>{{ pokemonList[6] }}</p>
<p>{{ pokemonList[7] }}</p>
<p>{{ pokemonList[8] }}</p>
<p>{{ pokemonList[9] }}</p>
<p>{{ pokemonList[10] }}</p>
<p>{{ pokemonList[11] }}</p>
```

Template

Activité :

- Regler le souci !

Template

L'interpolation :

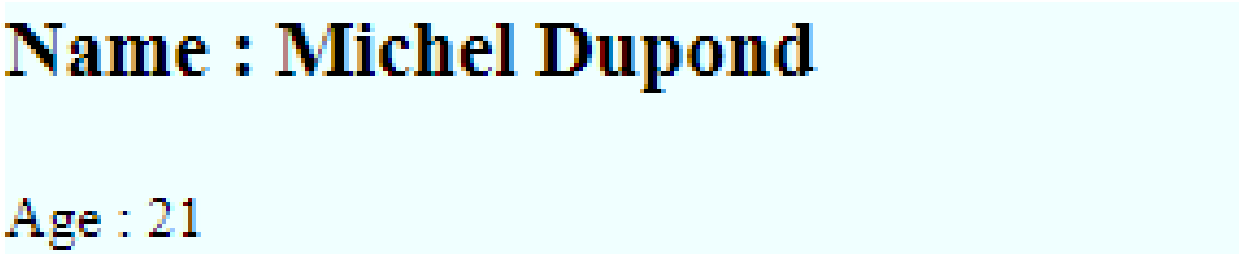
```
<h1>Liste des pokémon</h1>
<p>{{ pokemonList[0].name }}</p>
<p>{{ pokemonList[1].name }}</p>
<p>{{ pokemonList[2].name }}</p>
<p>{{ pokemonList[3].name }}</p>
<p>{{ pokemonList[4].name }}</p>
<p>{{ pokemonList[5].name }}</p>
<p>{{ pokemonList[6].name }}</p>
<p>{{ pokemonList[7].name }}</p>
<p>{{ pokemonList[8].name }}</p>
<p>{{ pokemonList[9].name }}</p>
<p>{{ pokemonList[10].name }}</p>
<p>{{ pokemonList[11].name }}</p>
```

Template

Activité :

- Créé un tableau nommé "Personne"
- Dans ce tableau on aura un nom et un prénom et age
- Puis afficher

dans le template avec la balise article le Nom, le Prenom et l'Age du tableau



Template

Interaction Utilisateur :

Pour l'instant tout ce qu'on a vue c'est de trasmettre des donnée de la Logique à la Vue

mais ce qu'on vas vouloir faire c'est l'inverse

Donc ce qu'on vas vouloir faire :
C'est quand on clic
sur un nom de pokemon excuter la fonction qu'on a créer "selectPokemon"