

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module : Angular

Enseignants : UP Web

Classes : 5GL & 5INFOB

Documents autorisés : OUI ☐ NON ☒

Calculatrice autorisée : OUI ☐ NON ☒

Date : 02/12/2020

Heure: 11h

Nombre de pages : 08

Internet autorisée : OUI ☐ NON ☒

Durée : 1h30

Partie 1 : Question Cours(7 pts)

Une seule réponse est correcte

- 1- Soit la configuration de routes suivante : (1pt)

```
const routes: Routes = [
  {path: '**', component:
  ProductsComponent},
  {path: 'randonnees', component:
  ContentComponent},
  {path: 'home', component:
  ArticleComponent},
];
```

En tapant l'url
localhost :4200/randonnees ,
lequel des composants suivants sera
affiché ?

- A- ProductsComponent
B- ContentComponent
C- ArticleComponent
D- Le composant AppComponent
- 2- Comment peut-on partager les données
entre composants qui ne sont pas
imbriqués? (1pt)
- A- Via un autre composant.
B- Via un service.
C- Via des événements.
D- Via les directives

- 3- Quelle est la solution proposée par
Angular qui permet aux développeurs
d'enrichir les balises HTML? (1pt)

A- Les promises
B- Les directives
C- Les observables
D- Les décorateurs

- 4- Dans quels cas ce titre apparaîtra-t-il
dans le Template? (1pt)

<h1 *ngIf="show">Classe 5eme!</h1>

A. show = true
B. show = false
C. show= 'true '
D. show = 'false'

- 5- Quelle directive vous permet d'assurer
le Two way data binding.? (1pt)

A. ngModel
B. ngForm
C. ngStyle
D. ngSubmit

6. Dans le Template d'un composant ListTodo, vous avez le sélecteur suivant :

`<app-todo [todo]='t'> </app-todo>`. Sachant que **app-todo** est le sélecteur d'un composant Todo.

a. Quelle est la relation entre le composant ListTodo et le composant Todo (1pt)

ListTodo est le composant parent du composant Todo

Ou bien

Le composant Todo est le composant fils du composant ListTodo

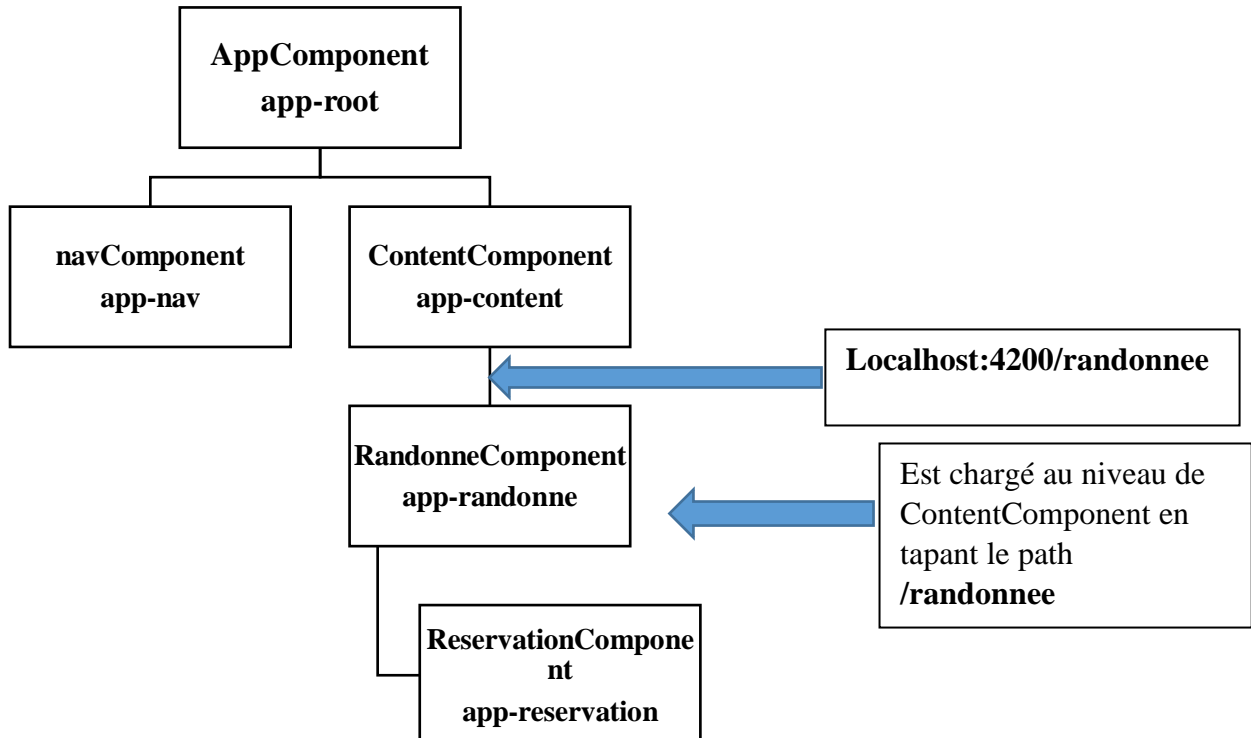
b. Quelle est la propriété qui sera déclarée dans le composant Todo ? écrire le code nécessaire pour la déclarer. (1pt)

La propriété à déclarer : todo (facultatif, s'il déclare cette propriété c'est suffisant)

@Input() todo : any ou bien @Input() todo : any ou bien @Input() todo

Partie 2 : Étude de cas (13 pts)

Nous allons développer la partie front-end d'une application web pour les passionnés des randonnées avec Angular 9. Pour cela, nous proposons la hiérarchie de composants suivante que nous allons créer tout au long de notre étude de cas.



Soit les deux modèles de données « **Randonnee** » et « **Reservation** » suivants :

```
export class Randonnee {  
  title : string;  
  description: string;  
  price:number;  
  date:string;  
  nbPlaces:number;}
```

```
export class Reservation{  
  identite : string;  
  telephone : string;  
  nb:number;  
}
```

Le composant **navComponent** contient dans son Template le code suivant :

```
<a href="#">Home</a>  
<a routerLink="/randonnees">Randonnées</a>  
<a href="#">Sports</a>  
<a href="#">Contact</a>
```

En cliquant sur le lien Randonnées, un composant **RandonneComponent** est chargé affichant le texte « **Nos randonnées** : ».

1- Définissez le contenu des templates des composants **AppComponent**, **ContentComponent** et **RandonneComponent**.

app.component.html (0.25 pt)

```
<app-nav></app-nav>  
<app-content></app-content>
```

content.component.html (0.25 pt)

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

randonne.component.html (0.25 pt)

Nos randonnées :

2- Au niveau du fichier app-routing.module.ts , configurez la route qui permet de pointer sur le composant RandonneComponent et dont le lien correspondant est déjà défini en haut dans navComponent **(0.25 pt)**

```
const routes: Routes = [  
  {path : 'randonnees', component: RandonneComponent} 0.25  
];
```

3- Soit la classe **RandonneService** suivante. La méthode getListRandonnees() permet de retourner la liste des randonnées contenues un fichier json. Complétez le code qui manque dans chaque numéro. (1.5 pt)

```
import { Injectable } from '@angular/core';
import { HttpClient } from [ 1 ];

import { Randonnee } from './model/randonnee';
@Injectable({
  providedIn: 'root'
})
export class RandonneService {
  constructor(private http : [ 2 ] ) { }
  getListRandonnees()
  {
    return this._http.get< [ 3 ] >("../assets/randonnees.json");
  }
}
```

- [1] '@angular/common/http' 0.5
- [2] HttpClient 0.5
- [3] Randonnee[] 0.5

Une fois le composant RandonneComponent est chargé, la liste des randonnées contenues dans le fichier json s'affiche comme mentionnée dans la **Figure 1**



Figure 1

4- Complétez par le code nécessaire pour la récupération de la liste des randonnées à partir de la méthode getListRandonnees() du service RandonneeService. (2 ,5 pts)

```
import { Component, OnInit } from '@angular/core';
import { RandonneeService } from '../randonnee.service';
@Component({
  selector: 'app-randonnee',
  templateUrl: './randonnee.component.html',
  styleUrls: ['./randonnee.component.css']
})

export class RandonneeComponent implements OnInit {
  1
  constructor( 2 ) { }
  ngOnInit(): void { 3 }
}
```

```
1 .. list : Randonnee[]; (0.5)
2 private srandonnee:RandonneeService (1)
3 this.srandonnee.getListRandonnees().subscribe(res=>this.list=res); (1)
```

RQ: l'étudiant peut appeler sa propriété comme il veut, list et srandonnée sont des propositions de correction.

5- Complétez le code html suivant pour afficher la liste des randonnées comme indiqué dans la Figure 1 sachant que les images possèdent l'extension « .jpg », sont stockées sous le répertoire « ../assets/images » et le nom de chaque image est le titre de la randonnée. (1.5pts)

```
<div *ngFor="let r of 1 ; let i=index">
  2      3      <button>Details</button>
</div>
```

1 : list (Le nom de la propriété donné par l'étudiant doit être le même que dans la question précédente) (0.25)

```
2: <img [src]=" '../././assets/images/'+r.title+'.jpg' " /> (1)
```

```
3: {{r.title}} (0.25)
```

```
*****
```

En cliquant sur le bouton « Details » de la Figure 1 :

- Le reste des informations de la randonnée courante s'affichent comme c'est indiqué dans la **Figure 2**.
- Un nouveau composant ReservationComposant sera affiché par la suite. Voir **Figure 3**

[HomeRandonnéesSportsContact](#)

Nos randonnées :



Randonnée Aveyron [Details](#)

- Paradis pour les amateurs de randonnées pédestres, l'Aveyron est parcouru de circuits et de sentiers balisés.
- 30 dt
- 15/01/2021
- 50



Randonnée Pédestre [Details](#)



Trekking [Details](#)

Figure 2

6- Sachant que en cliquant sur le bouton Details, une méthode ayant comme signature **showDetails(i :number)** est appelée, écrivez alors le code de cette méthode et mettez à jour, si c'est nécessaire, les propriétés de la classe RandonneComponent. **(1 pt)**

```
show: boolean = false; (0.25) //les noms des propriétés est au choix
num:number ;(0.25) //les noms des propriétés est au choix
showDetails(i:number){

    this.show=true; (0.25)
    this.num=i; (0.25) }
```

7- Complétez le code html suivant du composant **RandonneComponent** pour aboutir au résultat de la **Figure 2**. Il suffit de rajouter la condition nécessaire pour l’affichage du reste des informations d’une randonnée. **(1 pt)**

```
<div *ngFor="let r of list; let i=index">
  (Contenu déjà demandé dans la question 5)
  <button (click)="showDetails(i)">Details</button>
  <div * >
    <ul>
      <li>{{r.description}}</li>
      <li>{{r.price}} dt</li>
      <li>{{r.date}}</li>
      <li>{{r.nbPlaces}}</li>
    </ul>
  </div>
</div>
```

* *ngIf="show && i==num" (1)

Un nouveau composant appelé **ReservationComponent** est créé et ajouté dans la vue du composant **RandonneComponent**. Ce composant contient un formulaire de réservation à la randonnée correspondante et s’affiche aussi en cliquant sur le bouton **Details**. Voir **Figure 3**

Nos randonnées :



Randonnée Aveyron [Details](#)

- Paradis pour les amateurs de randonnées pédestres, l'Aveyron est parcouru de circuits et de sentiers balisés.
- 30 dt
- 15/01/2021
- 50

Identité

Veuillez écrire votre nom et prénom

N° Téléphone

N° de Téléphone obligatoire

Nombre de places à réserver

Veuillez saisir le nombre de place à réserver

[Réserver](#)



Randonnée Pédestre [Details](#)



Trekking [Details](#)

Figure 3

8- Le champ « Identité » du formulaire de réservation est **obligatoire**. Si ce champ est vide alors le message « Veuillez écrire votre nom et prénom » s'affiche. Complétez alors le code HTML suivant : (2pt)

```
<form #f="ngForm">
  <label>Identité</label>
  <input type="text"
  <div
    >
      Veuillez écrire votre nom et prénom
  </div>
</form>
```

1required (0.25) name="identite" (0.25) ngModel (0.5) #i="ngModel" (0.5)

=> le name et le nom de la variable de référence sont au choix

2 *ngIf="i.invalid" (0.5) => le nom de la variable de référence doit le
meme

que dans la réponse 1

10- Le bouton « Réserver » de la Figure 3 est **désactivé** si **le formulaire de réservation est invalide**. Mettez à jour le code html du composant **ReservationComponent** en ajoutant le code nécessaire au niveau du bouton « Réserver » **(1 pt)**

```
<button type="submit" * > Réserver</button>
```

```
* .....[disabled]="f.invalid" (1)
```

11- En cliquant sur le bouton « Réserver », cette réservation est ajoutée dans une liste des réservations défini dans le composant **RandonneComponent**, **listReservations : Reservation[]**. Mettez à jour la classe du composant **ReservationComponent** en indiquant uniquement le code utile pour répondre à cette question. **(1,5 pts)**

```
@Output() reserved (0.5) = new EventEmitter<Reservation (0.25)>();  
//le nom de la propriété est au choix  
participate(){ //le nom de la méthode est au choix  
  this.reserved.emit(this.maReservation); (0.75) - 0.5 si pas de réservation à  
  emettre  
}
```

Bon Travail ☺