

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

UE : Développement frontend

Modules : Application côté client – Intégration web

Enseignants : UP Web

Classes : 4TWIN

Documents autorisés : OUI

☐ NON

☒ Nombre de pages : 12

Calculatrice autorisée : OUI

☐ NON

☒ Internet autorisée : OUI NON ☐ ☒

Date : 30/01/2021

Heure: 11h Durée : 1h30

Une seule réponse est correcte. Choisissez la bonne proposition sur la feuille des réponses

1. Le TypeScript utilisé par Angular est-il interprété par les navigateurs ?
 - A. Non il faut compiler d'abord le TP en JS
 - B. Oui les nouveaux navigateurs comprennent TS
 - C. Oui grâce à un plugin Microsoft
 - D. Aucune réponse correcte
2. Où pouvons-nous utiliser les directives ?
 - A. Dans le Template du composant
 - B. Dans la classe du composant
 - C. Dans le fichier CSS du composant
 - D. Dans le fichier HTML et TS du composant
3. Dans quel cas un composant parent peut accéder aux propriétés et méthodes de son composant fils ?
 - A. Quand le composant fils possède une propriété précédé par le décorateur @ViewChild()
 - B. Quand le composant parent possède une propriété précédé par le décorateur @ViewChild()
 - C. Quand le composant parent possède une variable de référence dans son template
 - D. Quand le composant fils possède une variable de référence dans son template
4. Soit un composant B fils du composant A. Le composant B envoie un message de type chaîne de caractères au composant A. Que faut-il déclarer au niveau du composant B pour qu'il peut envoyer un message à A ?
 - A. @Input message : string ;
 - B. @Input() message : string ;
 - C. @Output() message = new EventEmitter() ;
 - D. @Output message = new EventEmitter<string>() ;
5. Depuis quel package importer la classe EventEmitter pour déclarer une propriété de sortie ?
 - A. events
 - B. protractor
 - C. @angular/core

D. @angular/events

6. C'est quoi le résultat du code suivant ?

```
<input type="text" value="Ahmed" name="nom" #p /> <h1> {{ p }} </h1>
```

A. Undefined

B. Ahmed

C. [object HTMLInputElement]

D. rien

7. Quelle est la bonne déclaration d'une variable référence de Template ?

A. Cette variable est définie n'importe où dans le Template du composant

B. Cette variable est accessible depuis la classe du composant

C. Cette variable est accessible par le composant parent

D. Les propositions a, b et c sont correctes

8. Soit une classe CSS "myStyle" définie dans le fichier CSS du composant. Nous voulons appliquer cette classe sur une division. Laquelle des propositions suivantes est correcte ?

A. <div [ngStyle]="myStyle">....</div>

B. <div [ngClass]=" 'myStyle' ">....</div>

C. <div ngClass="myStyle">....</div>

D. <div [class]="myStyle">....</div>

9. Quel est le type de binding suivant ?

```
<button [style.background-color] ="getcolor()">Save</button>
```

A. event Binding

B. propertyBinding

C. Two way Data Binding

D. Css Binding

10. C'est quoi le résultat du code suivant si le champ id n'est pas rempli sachant que la propriété Product est déclarée dans le ts ?

```
<form> id: <input type="number" [(ngModel)]="product.id" #i="ngModel"
required > <div [hidden]="!i.invalid"> id is required </div>
</form>
```

A. id is required

B. Rien ne s'affiche

C. id is required avec erreur dans la console.

D. Rien ne s'affiche avec erreur dans la console

11. C'est quoi le rôle de la directive NgForm?

A. NgForm permet de suivre tous les FormControl dans un formulaire

B. NgForm permet de suivre chaque FormControl, dans une collection de FormControl, attaché à la directive NgModel

C. NgForm permet de suivre tous les FormControl dans une collection de FormControl

D. Les réponses a,b et c sont correctes

12. Dans une application Angular, on peut déclarer un sous-ensemble d'éléments d'un formulaire. Parmi les propositions suivantes, laquelle est correcte?

A. <div formGroup="group1">.....</div>

- B. <div [formGroup]="group1">.....</div>
- C. <div formGroupName="group1">.....</div>
- D. <div [formGroupName]="group1">.....</div>

13. Dans l'approche ReactiveForms, on peut lier un élément HTML d'un **formulaire** avec un élément FormControl d'un FormGroup donné. Parmi les propositions suivantes, laquelle est correcte?

- A- <input type="text" [formControl]="product.name">
- B- <input type="text" [formControlName]="product.name">
- C- <input type="text" formControl="product.name">
- D- <input type="text" formControlName="product.name">

14. Quel est le service prédéfini par Angular ?

- A. ActivatedRoute
- B. HttpClient
- C. FormBuilder
- D. Toutes les réponses sont correctes

15. Quelle annotation permet de décorer un service ?

- A. @Injectable
- B. @injection
- C. @Service
- D. @ProvidedIn

16. Soit un service défini avec "providedIn:root". Quels sont les composants dans lesquels nous pouvons injecter ce service ?

- A. Uniquement les composants du module racine
- B. Uniquement les composants de AppModule
- C. N'importe quel composant de l'application
- D. Uniquement le composant AppComponent

17. L'objectif du code ci-dessous est de supprimer un produit à l'aide d'un service appelé ProductService. Complétez les zones numérotées du code suivant avec la bonne proposition.

```
export class ListProductComponent implements OnInit {
  listProducts: [1];
  product: Product;
  constructor(private productService: ProductService) {
  }
  ngOnInit(): void {
    this.productService.getProductsWS().subscribe([2]) }
  delete(id) {
    this.[3].deleteProduct(id).[4] (
      () => this.listProducts = this.listProducts.filter(product => product.id != id),
    );
  }
}
```

- A. 1 : Product, 2 : (data: Product[]) => this.listProducts = data, 3: productService, 4: subscribe
- B. 1 : Product, 2 : (data: Product) => this.listProducts = data, 3: productService, 4: observe
- C. 1 : Product[], 2 : (data: Product[]) => this.listProducts = data, 3: productService, 4: subscribe

D. 1 : Product[], 2 : (data: this.listProducts = data), 3: productService, 4: observe

18. Parmi les propositions suivantes, laquelle est le résultat du code suivant ?

```
@Component({
  selector: 'my-selector',
  template: `<h1>{{ name }}</h1>
  Name : <input [(ngModel)]="name">`
})
export class TestComponent implements OnInit {
  constructor() { }
  ngOnInit(): void { }
}
```

A.

Name :

B.

Undefined

Name :

C.

undefined Name :

D. Erreur

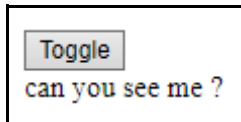
19. C'est quoi le résultat du code suivant parmi les propositions fournies ?

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

@Component({
  selector: 'my-selector',
  template: `<div>
    <button (click)='clicked()'>Toggle</button>
    <div *ngIf="show"> can you see me ? </div></div>
  `
})
export class TestComponent implements OnInit {
  show : boolean = false;
  clicked(){this.show= !this.show};
}
```

A.

B.



C. Page blanche

D. Erreur

20. Soit la méthode suivante :

```
function addNumber(input:boolean){  
  let a=100;  
  if(input){  
    let b=a+1;  
  }  
  return b;  
}
```

En tapant dans un fichier HTML `{{addNumber(true)}}`, c'est quoi le résultat attendu ?

A. 100

B. 101

C. erreur

D. 1

21. Quelle est la méthode qui n'appartient pas au cycle de vie d'un composant ?

A. `ngOnDestroy()`

B. `ngOnDie()`

C. `ngOnInit()`

D. `ngOnChanges()`

22. Soit trois composants `AppComponent`, `HomeComponent` et `ProductsComponent`. Le composant `AppComponent` contient un lien, en cliquant dessus le composant `HomeComponent` est chargé au niveau du composant `AppComponent`. Le composant `HomeComponent` contient un lien, en cliquant dessus le composant `ProductsComponent` est chargé au niveau du composant `HomeComponent`. Quelle est la bonne proposition qui permet de réaliser le traitement souhaité ?

A. Définir la directive « router-outlet » au niveau du composant `AppComponent` et configurer les routes comme suit :

```
{path:"home", component: HomeComponent},  
{path:"home/products ", component: ProductsComponent }
```

B. Définir la directive « router-outlet » au niveau du composant `AppComponent` et configurer les routes comme suit :

```
{path:"home", component: HomeComponent, children:[  
  {path:"products ", component: ProductsComponent }  
]}
```

C. Définir la directive « router-outlet » au niveau du composant `HomeComponent` et configurer les routes comme suit :

```
{path:"home", component:HomeComponent},  
{path:"products", component: ProductsComponent }
```

D. Définir une directive « router-outlet » au niveau du composant AppComponent et une autre directive « router-outlet » au niveau du composant HomeComponent et configurer les routes comme suit :

```
{path:"home", component: HomeComponent, children:[
  {path:"products", component: ProductsComponent }
]}
```

23. Sélectionnez le code qui permet de déclarer un formulaire d'authentification en utilisant l'approche ReactiveForms. Sachant que le formulaire contient deux champs : le login est un champ obligatoire d'une longueur minimale égale à 3 et le Password est un champ obligatoire qui contient 8 caractères (Lettre et chiffre)

A. `registerForm = new FormGroup({
 login: new FormControl("", Validators.required, Validators.minLength(3)),
 password: new FormControl("", Validators.required, Validators.pattern('[a-zA-Z]+[a-zA-Z]{8}')),
});`

B. `registerForm = new FormGroup({
 login: new FormControl("", Validators.required, Validators.minLength(3))],
 password: new FormControl("", Validators.required, Validators.pattern('[a-zA-Z]+@[a-zA-Z]{8}'))],
});`

C. `registerForm = new FormGroup({
 login: new FormControl("", Validators.required, Validators.minLength(3)),
 password: new FormControl("", Validators.required, Validators.pattern('[a-zA-Z]+@[a-zA-Z]{8}')),
});`

D. `registerForm = new FormControl({
 login: new FormControl("", [Validators.required, Validators.minLength(3)]),
 password: new FormControl("", [Validators.required, Validators.pattern('[a-zA-Z]+@[a-zA-Z]{8}'))],
});`

24. Nous voulons créer un lien HTML qui permet de pointer sur un path «reclamation» tout en envoyant la valeur d'un id. Quelle est la bonne déclaration du lien?

A. `Réclamer`

B. `Réclamer`

C. `<a [routerLink]=["/reclamation",p.id]>Réclamer`

D. `<a [routerLink]=["/reclamation",p.id]>Réclamer`

25. Soit le comportement souhaité suivant : « En tapant le mot 'home' dans l'URL, un composant HomeComponent doit se charger. En tapant n'importe quel autre mot, un composant PageNotFoundComponent est chargé ». Complétez les zones numérotées du code du fichier de routage « app.routing.module.ts » suivant par la bonne proposition.

```
import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';
import {Routes, RouterModule} from '[1]';
import {HomeComponent} from '../home/home.component';
import {PageNotFoundComponent} from '../page-not-found/page-not-found.component';
const ROUTES: Routes = [
  {path: 'home', component: HomeComponent},
  [2]
];
@NgModule({
```

- A. 1: @angular/router', 2: {path: '**', component: PageNotFoundComponent}, 3: RouterModule.forRoot(ROUTES), 4: exports: [RouterModule]
- B. 1: @angular/router', 2: {path: '', component: PageNotFoundComponent}, 3: RouterModule.forRoot(ROUTES), 4: exports: [RouterModule]
- C. 1: @angular/core', 2: {path: '**', redirectTo: PageNotFoundComponent}, 3: RouterModule.forRoot(ROUTES), 4: providers: [RouterModule]
- D. 1: @angular/core/router', 2: {path: '**', component: PageNotFoundComponent}, 3: RouterModule.forRoot(ROUTES), 4: exports: [RouterModule]

26. Comment résoudre l'erreur suivante?

" « No directive found with exportAs 'ngForm' »"

- A. Il faut mettre ngModel entre []
- B. Il faut exporter le contenu de ngModel dans une variable de référence
- C. Il faut ajouter un name à l'input correspondant
- D. Il faut importer le module FormsModule dans la liste des imports

27. Vous pouvez créer une référence locale HTML d'une balise HTML en utilisant une variable. Par quel caractère doit commencer cette variable ?

- A. @
- B. #
- C. *
- D. &

28. Soit une propriété 'available' de type Boolean déclarée dans un composant Angular et qu'on va l'utiliser pour changer la couleur d'un élément p du Template de ce composant : si available= true alors color = vert, si available = false alors color = rouge. Sélectionnez le code qui permet d'implémenter ce traitement.

- A. <p [ngStyle] = "{ 'color':(available==true ? 'green' : 'red') }">Contenu</p>
- B. <p [ngStyle] = "{ 'color':(available===true then 'green' else 'red') }">Contenu</p>
- C. <p [style.color] = "{ 'color':(available==true ? 'green' : 'red') }">Contenu</p>
- D. A et B sont correctes