

Examen Angular 21/10/2021

1. Que permet de faire la décoration **@Input** ?

Communication entre component parent (mère) et le component fille

2. Je souhaite créer un formulaire par le **template**, je dois déclarer quelle directive Angular sur la balise form afin de lui dire que j'utilise un formulaire par le template ?

ngForm

3. Lors d'un formulaire par le code, que permet de faire la directive « **formControlName** » ?

Elle fait la liaison entre l'input HTML et le formControl dans le typescript

4. J'ai créé une route vers un component avec un paramètre, afin de pouvoir récupérer ce paramètre dans ce component, je dois utiliser un objet Angular précis, lequel (juste le nom) ?

ActivatedRoute

5. Que permet de faire cette ligne de code :

```
[class.is-selected]="dice.isSelected"
```

On applique la class « is-selected » si la condition est validée, ici la condition est que « **dice.isSelected** »

6. Je viens de faire un « **git clone** » sur un projet Angular, afin que mon projet fonctionne correctement, je dois lancer une commande, laquelle ?

« **npm i** »

7. Lors de la réalisation d'un formulaire (peut-importe lequel), je peux utiliser des propriétés proposées par Angular, afin de vérifier l'état des inputs (champs du formulaire), il en existe 4, lesquelles et que font-elles ?

Dirty : modifié au moins une fois par l'utilisateur

Touched : sélectionné au moins une fois par l'utilisateur

Invalid : l'input ne respecte pas les validations html

Valid : l'input respecte les validations html

8. Que représente le mot clé « **this** » ?

La classe/l'objet courant(e)

9. A quoi sert l'**injection de dépendance** dans Angular ?

Laisser le framework créer lui-même les objets dont on a besoin. L'injection de dépendance se fait uniquement dans un constructeur.

10. Détaillez cette ligne de code :

```
let clermontFd = new Commune() ;
```

On déclare une variable, modifiable, de nom `clermontFd` que l'on instancie comme étant un objet de type `Commune`.

11. Il existe **3 mots clés pour représenter la visibilité des classes/attributs**, quels sont-ils et donner une définition pour chacun d'entre eux.

Private : accessible seulement dans la classe courante

Public : accessible partout

Protected : accessible seulement dans la classe courante et les classes filles

12. Dans le cadre de l'héritage, **je suis dans une classe fille**, quel mot-clé me permet de rappeler les comportements des méthodes de la classe mère ?

Le mot clé : **super**

13. Quelle(s) différence(s) y a-t-il entre un « **for..in** » et un « **for..of** » ?

« **for..in** » : itère dans un tableau et renvoie l'index

« **for..of** » : itère dans un tableau et renvoie la valeur

14. Quelle(s) différence(s) y a-t-il entre une **classe** et une **interface** ?

L'interface ne s'instancie pas et ne peut pas avoir de méthode avec un comportement, seulement une signature de méthode.

15. Un component est constitué de 4 fichiers distinct, quels sont leur rôle respectif ? Préciser leur extension de fichier à chaque fois.

.ts : la partie typescript, c'est le « controller », il gère les évènements de la vue

.html : la vue, l'interface utilisateur

.scss : le fichier de style, c'est là que l'on définit le « design » de la vue

.spec.ts : le fichier de tests unitaires du component

16. Dans l'html, je souhaite créer un bouton avec une action au clic, comment puis-je créer cette action ? (Donner le code)

```
<button (click)= « uneFonction »>OK</button>
```

17. Comment puis-je créer, ou instancier, une classe ?

On utilise le mot clé « **new NOM_CLASS** »

18. Avec quelle commande crée-t-on un service ? A quoi sert un **Service** en Angular ?

```
« ng g s NOM_SERVICE »
```

Il est là pour rendre un service, entre autres, donner des données aux components (il peut être utilisé plusieurs fois) ou simplement service de classe de « stockage » de méthode, afin d'éviter des répétitions.

19. Quelle est la particularité d'une **classe abstraite** (**abstract class**) ?

Elle ne s'instancie pas

20. Dans l'html d'un component, que fait cette ligne de code :

```
{{oneCase.piece?.nom}}
```

Il s'agit d'un « if » (le « ? »), si l'objet piece existe, alors on affiche sa propriété « nom »

(Question Bonus : Quelle commande Git permet de récupérer les changements du dépôt distant ?)

git pull