

[Accueil](#) > [Cours](#) > [Débutez avec Angular](#) > Quiz : Structurer un document avec des directives

Débutez avec Angular

 10 heures  Moyenne

Mis à jour le 13/06/2022



Structurer un document avec des directives

Vous n'avez pas validé ce quiz.



Vous n'avez pas atteint le seuil de validation de cet exercice, c'est-à-dire 70%. Ce n'est pas très grave car vous pourrez refaire le quiz dans 24h.

Compétences évaluées

 Structurer un document avec des directives

Question 1

Qu'est-ce qu'une directive ?

-  ☒ Un nouvel attribut HTML5
- ☐ Un décorateur que l'on ajoute à la déclaration d'un component
-  ☐ Une classe TypeScript qui ajoute du comportement supplémentaire à un élément HTML
- ☐ Une norme de nomenclature Angular

Une directive est une classe TypeScript qui ajoute du comportement supplémentaire à un élément HTML (qui peut aussi être un component).

Question 2

À quoi sert la directive `*ngIf` ?

- ☐ À montrer/cacher des éléments du DOM selon une condition donnée
- ✓ ☐ À ajouter ou non des éléments au DOM selon une condition donnée
- ✗ ☒ À appeler ou non une méthode selon une condition donnée
- ☐ À ajouter des classes CSS selon une condition donnée

La directive `*ngIf` sert à ajouter ou non un élément au DOM selon une condition donnée. Elle ne "cache" pas les éléments qui ne remplissent pas la condition : elle ne les ajoute simplement pas au DOM.

Question 3

Nous sommes dans un component où `photoUrl` est une propriété de type `string`.

Quelle syntaxe ci-dessous est correcte pour ajouter cette `<div>` au DOM uniquement si la valeur de `photoUrl` est égale à la valeur d'une deuxième variable appelée `maPhoto`, aussi de type `string` ?

- ☐

html

```
1 <div *ngIf="let photoUrl of photos"></div>
```
- ☐

html

```
1 <div *ngIf="photoUrl === 'maPhoto'"></div>
```
- ✓ ☒

html

```
1 <div *ngIf="photoUrl === maPhoto"></div>
```
- ☐

html

```
1 <div *ngIf="photoUrl && maPhoto"></div>
```

On doit passer une condition valable TypeScript à la directive `*ngIf` : c'est la condition selon laquelle l'élément sera ajouté ou non au DOM. `photoUrl` et `maPhoto` sont des variables, et donc n'ont pas besoin d'être entourées par `'`.

Question 4

À quoi sert la directive `*ngFor` ?

- ☐ À ajouter ou non un élément au DOM selon une condition donnée
- ✓ ☐ À ajouter un élément au DOM pour chaque élément d'un tableau
- ✗ ☒ À modifier tous les éléments d'une liste affichée dans le DOM
- ☐ À ajouter ou non des classes CSS à un élément du DOM

La directive `*ngFor` permet d'ajouter un élément au DOM pour chaque élément du tableau passé en argument. La première réponse correspond à `*ngIf`, la dernière à `[ngClass]`. Il n'y a pas de directive pour la troisième réponse.

Question 5

Nous nous trouvons dans un composant qui a une propriété `transactions` qui est un tableau d'objets de type `Transaction` (un type qui existe dans cette application fictive). On veut ajouter une instance de `DelayedTransactionComponent` pour chaque élément de ce tableau. `DelayedTransactionComponent` a une propriété `delayedTransaction` à laquelle on doit lier les `Transaction` du tableau.

Quelle syntaxe fonctionne correctement ?

✗ ☒

html

```
1 <app-delayed-transaction *ngFor="let transaction in transactions"
  [delayedTransaction]="transaction">
2 </app-delayed-transaction>
```

☐

html

```
1 <app-delayed-transaction *ngFor="let transactionElement of transactions"
  [delayedTransaction]="transaction">
2 </app-delayed-transaction>
```

✓ ☐

html

```
1 <app-delayed-transaction *ngFor="let transactionElement of transactions"
  [delayedTransaction]="transactionElement">
2 </app-delayed-transaction>
```

☐

html

```
1 <app-delayed-transaction *ngFor="let transactionElement of transactions"
  [delayedTransaction]="transaction">
2 </app-delayed-transaction>
```

La syntaxe valable est `<balise *ngFor="let element of array" [property]="element" >` .

Question 6

Que signifie l'astérisque dans `*ngIf` et `*ngFor` ?

- ☐ Que ce sont des directives
- ☐ Que ce sont des directives par attribut
- ✓ ☒ Que ce sont des directives structurales
- ☐ Que ce sont des composants

L'astérisque signifie que ce sont des directives **structurales**, qui modifieront donc la structure du document.

Question 7

Quelle est la différence entre `[ngStyle]` et `[ngClass]` ?

- ✓ ☐ `[ngStyle]` applique des valeurs dynamiques directement à des styles ; `[ngClass]` applique des classes CSS selon des conditions données.
- ☐ `[ngStyle]` ajoute des styles à une classe CSS donnée ; `[ngClass]` ajoute ensuite cette classe CSS selon une condition donnée.
- ☐ `[ngStyle]` applique des styles selon des conditions données ; `[ngClass]` applique des classes CSS qui tirent leurs noms de valeurs dynamiques.
- ✗ ☒ `[ngStyle]` applique des styles à des éléments sélectionnés par une méthode donnée ; `[ngClass]` applique des classes CSS sélectionnées par une méthode donnée.

`[ngStyle]` applique des valeurs dynamiques directement à des styles ; `[ngClass]` applique des classes CSS selon des conditions données.

Question 8

Dans un composant qui contient une propriété `size` de type `number` compris entre 0 et 1, on souhaite que :

- la largeur de cette `<div>` soit proportionnelle à `size` ,
- où une valeur de 0 correspond à une largeur de 0 % et une valeur de 1 correspond à une largeur de 100 %.

Quelle syntaxe `[ngStyle]` nous permet d'implémenter cette fonctionnalité ?



html

```
1 <div [ngStyle]="width: size * 100 + %"></div>
```



html

```
1 <div [ngStyle]="{ width: size + % }"></div>
```



html

```
1 <div [ngStyle]="{ width: size + '%' }"></div>
```






html

```
1 <div [ngStyle]="{ width: size * 100 + '%' }"></div>
```

`[ngStyle]` prend un objet où les clés sont les styles à modifier et les valeurs correspondent aux valeurs qui doivent prendre ces styles. Pour passer la bonne valeur, il faut multiplier `size` par 100 et y ajouter le caractère `%` .

Question 9

À quoi correspondent les valeurs de l'objet passé à `[ngClass]` ?

- ☐ Les valeurs qui doivent être appliquées aux styles
- ☐ Les noms des classes qui doivent être appliquées
-   Les éléments auxquels les classes doivent être appliquées
-  ☐ Les conditions selon lesquelles les classes doivent être appliquées

Les valeurs de l'objet passé à `[ngClass]` sont les conditions selon lesquelles les classes doivent être appliquées.

[Mettez de la classe](#)[Changez la casse](#)

Le professeur

Will Alexander

Scottish developer, teacher and musician based in Paris.

OPENCLASSROOMS 

OPPORTUNITÉS 

AIDE 

POUR LES ENTREPRISES 

EN PLUS 

 Français 



