Machine Translated by Google

Questions à choix multiples (QCM) - Réponses et justifications

1. Composants angulaires :

Réponse : b) Pour créer des éléments d'interface utilisateur réutilisables

- Justification : les composants sont les éléments fondamentaux d'Angular.
  applications. Ils encapsulent les données, le balisage HTML et la logique d'une partie particulière de l'interface utilisateur, les rendant réutilisables et maintenables.

2. Liaison de données :

Réponse : c) Liaison de propriété

- Justification : la liaison de propriété est utilisée pour envoyer des données du composant parent au composant enfant. Il lie une propriété d'un élément DOM à un champ de la classe de composant.

3. Injection de dépendances :

Réponse : c) Pour injecter des services dans des composants ou d'autres services

- Justification : L'injection de dépendances (DI) est un modèle de conception utilisé dans Angular pour
  fournir aux composants les services ou autres choses dont ils ont besoin. Cela permet aux classes d'être plus découplées et modulaires.

4. Directives :

Réponse : b) Pour afficher conditionnellement un élément

- Justification : ngIf est une directive structurelle qui ajoute ou supprime conditionnellement un élément du DOM en fonction d'une condition donnée.

5. Prestations :

Réponse : a) @Injectable()

- Justification : Le décorateur @Injectable() est utilisé pour définir une classe de service dans Angular qui peut être injectée dans des composants et d'autres services.

6. Crochets de cycle de vie :

Réponse : c) ngAfterViewInit()

- Justification : ngAfterViewInit est appelé une fois qu'Angular a complètement initialisé la vue et les vues enfants d'un composant.

7. Acheminement :

Réponse : a) /route/:param

Machine Translated by Google

- Justification : Dans Angular, les paramètres de route dynamique sont spécifiés par deux points (:) suivis du nom du paramètre.

8. Formes réactives :

Réponse : b) FormControl

- Justification : FormControl suit la valeur et le statut de validation d'un individu contrôle des formulaires.

9. Injection de dépendances :

Réponse : d) Favorise un couplage lâche entre les classes

- Justification : L'injection de dépendances (DI) favorise le couplage lâche en permettant aux classes de recevoir leurs dépendances à partir de sources externes plutôt que de les créer en interne.

10. Tuyaux :

Réponse : a) datePipe

- Justification : Le datePipe est un tube intégré dans Angular utilisé pour formater les dates selon un format spécifié.

11. Modules angulaires :

Réponse : a) Pour encapsuler et regrouper des composants, des services et des directives

- Justification : NgModule aide à organiser une application en blocs cohérents de fonctionnalités. Il encapsule des composants, des services et des directives, rendant l'application modulaire.

12. Observables et RxJS :

Réponse : b) Un flux continu de données auquel vous pouvez vous abonner

- Justification : les observables prennent en charge la transmission de messages entre les éditeurs et les abonnés dans votre application. Ils sont largement utilisés dans Angular pour les opérations asynchrones.

13. Détection des changements :

Réponse : c) En utilisant une zone pour détecter les opérations asynchrones

- Justification : Angular utilise des zones pour intercepter les activités asynchrones, puis effectue une détection des modifications une fois ces tâches terminées.

Questions directes - Réponses et justifications

Machine Translated by Google

1. Directives :

   - Réponse : Les directives structurelles modifient la disposition en ajoutant, supprimant ou manipulant éléments (par exemple, ngIf, ngFor). Les directives d'attribut modifient l'apparence ou le comportement d'un élément existant (par exemple, ngStyle, ngClass).
   - Justification : Comprendre la différence est crucial pour manipuler le DOM et contrôler efficacement le comportement et l'apparence des éléments.

2. Prestations :

   - Réponse : Les services sont des objets singleton qui fournissent des fonctionnalités partagées entre les composants. Ils sont utiles pour réutiliser le code, maintenir l'état et implémenter une logique de niveau de service comme les requêtes HTTP.
   - Justification : l'utilisation de services favorise une architecture propre en déléguant des tâches spécifiques à des classes de services, en rendant les composants allégés et axés sur la présentation des données.

3. Routage :

   - Réponse : Configurez une route dans Angular en définissant un tableau Route dans votre @NgModule décorateur, spécifiant le chemin et le composant. Utilisez le RouterModule.forRoot() méthode pour l'enregistrer.
   - Justification : Une configuration appropriée des itinéraires est essentielle pour naviguer entre les composants et créer une application monopage (SPA).