

1- Quelle est la différence entre `let` et `var` pour déclarer des variables en JavaScript ?

- a. Aucune différence
- b. **`let`** a une portée de bloc, tandis que **`var`** a une portée de fonction
- c. **`var`** a une portée de bloc, tandis que **`let`** a une portée de fonction

2- Qu'est-ce qu'une fonction de rappel (callback) en JavaScript ?

- a. Une fonction qui rappelle une autre fonction tous les X millisecondes.
- b. Une fonction qui est passée comme argument à une autre fonction et qui est exécutée ultérieurement.
- c. Une fonction qui renvoie une autre fonction.

3- Comment éviter le callback hell en JavaScript ?

- a. En utilisant uniquement des promesses.
- b. En évitant complètement l'utilisation des fonctions de rappel.
- c. En organisant le code de manière modulaire et en utilisant des techniques telles que les promesses ou `async/await`.

4- Comment déclarer une fonction fléchée avec un seul paramètre nommé `x` ? a. `x => {}`

- b. `(x) => {}`
- c. `function(x) {}`

5- Quel langage est principalement utilisé pour écrire le noyau de Node.js (runtime) ?

- a. Java
- b. JavaScript
- c. C++

d. Python

6- Qu'est-ce que le bytecode dans le contexte de l'interprétation JavaScript ?

- a. Une forme compressée du code source JavaScript.
- b. Une représentation intermédiaire du code générée après la compilation.
- c. Un format binaire utilisé pour stocker les chaînes de caractères en JavaScript.
- d. le bytecode n'existe pas en javascript

7- Comment utiliser filter pour obtenir un tableau ne contenant que les éléments supérieurs à 10 dans un tableau donné ?

- a. `array.filter(element > 10);`
- b. `filter(array, element => element > 10);`
- c. `array.filter(element => element > 10);`

8- Quelle est la syntaxe correcte pour accéder à la propriété name d'un objet person ?

- a. `person.name;`
- b. `person[name];`
- c. `person['name'];`

9- Comment ajouter une nouvelle propriété age à un objet user ?

- a. `user.addProperty('age', 25);`
- b. `user.age = 25;`
- c. `user.setProperty('age', 25);`

10- Quelle méthode est utilisée pour vider tous les enfants d'un nœud dans l'arbre DOM ?

- a. `clearChildren()`
- b. `removeAllChildren()`
- c. `node.innerHTML = '';`

11- Comment utiliser `innerHTML` pour ajouter un lien hypertexte à un élément avec l'ID "myDiv" ?

- a. `document.getElementById('myDiv').addLink('Lien');`
- b. `document.getElementById('myDiv').innerHTML = 'Lien';`
- c. `addLink('myDiv', 'Lien');`

12- Comment attacher un gestionnaire d'événements "click" à un bouton avec l'ID "myButton" en JavaScript ?

- a. `myButton.addEventListener('click', myFunction);`
- b. `myButton.attachEvent('click', myFunction);`
- c. `onClick(myButton, myFunction);`

13- Comment sélectionner tous les éléments avec la classe "myClass" en JavaScript ?

- a. `selectElementsByClass('myClass');`
- b. `getElementsByClass('myClass');`
- c. `document.getElementsByClassName('myClass');`
- d. `document.querySelectorAll("myClass");`

14- Quel en-tête HTTP est utilisé pour indiquer le type de contenu dans une réponse ?

- a. Content-Type
- b. Content-Encoding
- c. Type

15- Quel est le rôle du troisième paramètre (booléen) dans la méthode `xhr.open` lors de la création d'une requête `XMLHttpRequest` ?

- a. Il spécifie le type de requête (GET, POST, etc.).
- b. Il indique si la requête doit être asynchrone (true) ou synchrone (false).
- c. Il détermine le format des données de la requête (JSON, XML, etc.).
- d. Il définit si la requête doit être sécurisée (true pour HTTPS, false pour HTTP).

16- Qu'est-ce que Node.js ?

- a. Un serveur de bases de données.
- b. Un environnement d'exécution JavaScript côté serveur.
- c. Un framework JavaScript pour le développement front-end et back-end.

17- Quel est le rôle PRINCIPAL du fichier `package.json` dans un projet Node.js ?

- a. Définir les scripts de build.
- b. Spécifier les dépendances du projet et configurer des métadonnées.
- c. Indiquer le chemin du point d'entrée du programme.

18- Comment utilisez-vous un middleware spécifique pour une route particulière dans Express.js ?

a. `app.route('/path').applyMiddleware(myMiddleware)`

b. `app.use('/path', myMiddleware)`

c. `app.applyMiddleware(myMiddleware, '/path')`