Examen blanc : Préparation au mi-temps 2025

Partie 1 : QCM à Choix Unique (50 questions)

1. **Quel mot-clé permet de déclarer une variable accessible uniquement dans le bloc où elle est définie ?**
   A. var
   B. let
   C. const
   D. function

2. **Quelle méthode permet de transformer un tableau en chaîne de caractères en insérant un séparateur ?**
   A. split()
   B. join()
   C. concat()
   D. reduce()

3. **Que retourne typeof null en JavaScript ?**
   A. "null"
   B. "undefined"
   C. "object"
   D. "string"

4. **Comment s'appelle la phase où JavaScript prépare les déclarations de variables et de fonctions en mémoire avant l'exécution ?**
   A. Héritage
   B. Initialisation
   C. Hoisting
   D. Instanciation

5. **Quelle est la valeur de NaN === NaN ?**
   A. true
   B. false
   C. Lève une erreur
   D. undefined

6. **Quelle méthode ajoute un ou plusieurs éléments à la fin d'un tableau ?**
   A. push()
   B. pop()
   C. unshift()
   D. shift()

7. **Lequel de ces types n'existe pas en JavaScript ?**
   A. number
   B. boolean
   C. symbol
   D. double

8. **Quelle méthode est utilisée pour convertir une chaîne en nombre entier ?**
   A. parseInt()
   B. parseFloat()
   C. Number()
   D. Math.floor()

9. **Quel est le résultat de '5' + 3 ?**
   A. 8
   B. 53
   C. NaN
   D. Lève une erreur

10. **Quelle structure permet de gérer les erreurs avec un bloc d'exécution et un bloc de récupération ?**
    A. if / else
    B. try / catch
    C. switch / case
    D. throw / return

11. **Quel mot-clé est utilisé pour stopper entièrement une boucle for ?**
    A. stop
    B. exit
    C. break
    D. return

12. **Que fait Array.prototype.map() ?**
    A. Il modifie le tableau d'origine.
    B. Il renvoie un nouveau tableau transformé.
    C. Il efface le contenu du tableau.
    D. Il trie le tableau.

13. **Quelle est la portée d'une variable déclarée avec var dans une fonction ?**
    A. Portée bloc
    B. Portée globale
    C. Portée de fonction
    D. Portée script uniquement

14. **Quelle est la sortie de console.log(2 == '2') ?**
    A. true
    B. false
    C. undefined
    D. Erreur de type

15. **Dans une fonction fléchée, que représente this ?**
    A. L'objet global (window en navigateur)
    B. Le contexte courant
    C. L'instance de la fonction
    D. Il n'existe pas de this dans les fonctions fléchées

16. **Quelle est la valeur par défaut de arguments dans une fonction fléchée ?**
    A. Un tableau vide

B. Tous les arguments passés
C. undefined
D. Les fonctions fléchées n'ont pas arguments

17. **Que retourne typeof [] ?**
A. "array"
B. "object"
C. "[]"
D. "undefined"

18. **Quelle méthode permet de vérifier si un tableau contient un élément ?**
A. Array.includes()
B. Array.has()
C. Array.find()
D. Array.match()

19. **Que renvoie false == 0 ?**
A. true
B. false
C. undefined
D. Lève une erreur

20. **Quelle méthode supprime le dernier élément d'un tableau et le renvoie ?**
A. push()
B. pop()
C. shift()
D. splice()

21. **Quelle syntaxe permet de déstructurer un objet ?**
A. let { nom } = objet;
B. let [nom] = objet;
C. let (nom) = objet;
D. let <nom> = objet;

22. **Comment déclarer une constante en JavaScript ?**
A. var CONSTANT
B. let constant
C. const constant
D. CONST constant

23. **Quel est le résultat de typeof NaN ?**
A. "number"
B. "NaN"
C. "undefined"
D. "object"

24. **Que fait la méthode String.prototype.slice(start, end) ?**
A. Supprime une partie de la chaîne
B. Renvoie une sous-chaîne entre start et end (exclu)
C. Convertit la chaîne en tableau
D. Découpe la chaîne en mots

25. **Qu'est-ce qui n'est pas une structure de données native en JavaScript ?**
A. Map
B. Set
C. WeakMap
D. List

26. **Quel est l'événement déclenché quand la page a fini de se charger en HTML ?**
A. onload
B. DOMContentLoaded
C. loadend
D. ready

27. **Quel opérateur compare et convertit éventuellement les types avant comparaison ?**
A. ===
B. !==
C. ==
D. =

28. **Quelle méthode convertit un objet JavaScript en JSON ?**
A. JSON.parse()
B. JSON.stringify()
C. obj.toString()
D. String()

29. **Dans quelle portée vit une variable déclarée avec let dans un bloc if ?**
A. Portée globale
B. Portée de fonction
C. Portée du bloc if
D. Portée de script

30. **Comment se nomme la syntaxe suivante : (...args) => {} ?**
A. Fonction classique
B. Fonction anonyme
C. Fonction fléchée (arrow function)
D. Méthode d'un objet

31. **Quelle méthode array renvoie le premier élément qui satisfait une condition ?**
A. filter()
B. find()
C. includes()
D. forEach()

32. **Quel objet global représente l'API de la console en JavaScript ?**
A. console
B. log
C. window.consoleAPI
D. debug

33. **Quelle est la sortie de Boolean([]) ?**
A. false
B. true
C. Lève une erreur
D. null

34. **Comment définit-on une propriété d'un objet en utilisant la notation littérale ?**
A. let obj = { nom: "Alice" };
B. let obj = Object(nom: "Alice");
C. let obj = (nom = "Alice");
D. obj.nom = "Alice";

35. **Que fait la méthode Array.prototype.forEach() ?**
A. Renvoie un nouveau tableau

B. Exécute une fonction sur chaque élément sans retourner de résultat
C. Filtre les éléments selon une condition
D. Casse la boucle dès qu'une condition est remplie

36. **Comment générer un entier aléatoire entre 0 et 9 inclus ?**
A. Math.random() * 10
B. Math.floor(Math.random() * 10)
C. Math.floor(Math.random() * 9)
D. Math.ceil(Math.random() * 9)

37. **Quelle propriété d'un objet Error en JavaScript contient le message d'erreur ?**
A. Error.text
B. Error.message
C. Error.info
D. Error.description

38. **Quelle est la bonne façon de déclarer une classe en JavaScript (ES6) ?**
A. function ClassName {}
B. class ClassName {}
C. var ClassName = class {}
D. create class ClassName {}

39. **Que renvoie isNaN("Hello") ?**
A. true
B. false
C. Lève une erreur
D. "Hello"

40. **Quel mot-clé permet de sortir d'une fonction et éventuellement de renvoyer une valeur ?**
A. break
B. stop
C. return
D. exit

41. **Quelle syntaxe pour déclarer un module ES ?**
A. <script type="text/javascript">
B. <script type="module">
C. <script module>
D. <module>

42. **Quelle est la sortie de console.log(typeof undefined) ?**
A. "object"
B. "null"
C. "undefined"
D. "string"

43. **Quelle méthode permet de chainer des promesses de façon séquentielle ?**
A. promise.stack()
B. promise.next()
C. promise.then()
D. promise.link()

44. **Comment nomme-t-on une fonction passée en paramètre à une autre fonction ?**
A. Fonction dynamique
B. Fonction callback
C. Fonction flechée
D. Fonction asynchrone

45. **Quelle méthode JSON convertit une chaîne JSON en objet JavaScript ?**
A. JSON.stringify()
B. JSON.parse()
C. JSON.format()
D. JSON.decode()

46. **Quelle syntaxe ES6 permet d'importer une exportation par défaut ?**
A. import { default } from "module";
B. import default from "module";
C. import * from "module";
D. import something from "module";

47. **Que renvoie typeof function(){} ?**
A. "function"
B. "object"
C. "function object"
D. "undefined"

48. **Lequel de ces opérateurs est utilisé pour la décomposition d'un tableau en arguments de fonction ?**
A. ...
B. ::
C. >>
D. ??

49. **Que fait l'opérateur ?? (nullish coalescing) ?**
A. Compare strictement deux valeurs
B. Retourne la valeur de droite si la valeur de gauche est null ou undefined
C. Effectue la fusion de deux objets
D. Vérifie si la valeur est strictement null

50. **Que se passe-t-il si on appelle une fonction avant sa déclaration avec function maFonction(){} ?**
A. Erreur de référence
B. La fonction est hoistée, donc accessible
C. La fonction est ignorée
D. Le code plante uniquement en mode strict

Partie 2 : QCM à Choix Multiples (50 questions)

Attention : Ici, **plusieurs réponses** peuvent être correctes.

1. **Quelles sont les méthodes qui modifient directement le tableau initial ?**
A. map()
B. splice()
C. push()
D. pop()

2. **Quelles valeurs sont considérées comme falsy en JavaScript ?**
A. false
B. 0

C. null
D. undefined

3. **Lesquels de ces opérateurs sont des opérateurs logiques ?**
   A. &&
   B. ||
   C. ===
   D. !

4. **Lesquels de ces objets sont des objets globaux du navigateur ?**
   A. window
   B. document
   C. console
   D. NodeList

5. **Lesquelles de ces syntaxes créent un tableau vide ?**
   A. const arr = [];
   B. const arr = new Array();
   C. const arr = {};
   D. const arr = [null];

6. **Quels mots-clés permettent de déclarer des variables en ES6 ?**
   A. var
   B. let
   C. const
   D. function

7. **Lesquels de ces types sont des types primitifs en JavaScript ?**
   A. string
   B. number
   C. boolean
   D. object

8. **Quels sont les deux paramètres de base de la fonction exécuteur d'une promesse ?**
   A. resolve
   B. reject
   C. success
   D. error

9. **Lesquelles de ces boucles existent en JavaScript ?**
   A. for
   B. for...in
   C. for...of
   D. foreach

10. **Quels événements sont liés aux clics de souris ?**
    A. click
    B. dblclick
    C. contextmenu
    D. mouseover

11. **Quels opérateurs comparent strictement sans conversion de type ?**
    A. ==
    B. ===
    C. !==
    D. !=

12. **Lesquelles de ces méthodes sont disponibles sur un objet Promise ?**
    A. .then()
    B. .catch()
    C. .all()
    D. .finally()

13. **Lesquels de ces mots-clés sont utilisés pour définir une classe et un constructeur ES6 ?**
    A. class
    B. function
    C. constructor
    D. extends

14. **Lesquelles de ces valeurs sont évaluées à true en contexte booléen ?**
    A. '0' (string)
    B. [] (tableau vide)
    C. null
    D. {} (objet vide)

15. **Lesquels de ces objets sont utilisables pour effectuer des requêtes HTTP en JavaScript (côté navigateur) ?**
    A. XMLHttpRequest
    B. fetch
    C. axios (librairie externe)
    D. Response

16. **Dans quels cas this fait référence à l'objet global en mode non strict ?**
    A. Dans une fonction classique appelée sans contexte
    B. Dans une fonction fléchée
    C. Dans un événement DOM sur un élément
    D. Dans une fonction de rappel (callback) attachée à un objet

17. **Lesquelles de ces méthodes peuvent être utilisées pour itérer sur un tableau sans modifier l'original ?**
    A. forEach()
    B. map()
    C. reduce()
    D. splice()

18. **Quels sont les éléments du modèle de programmation asynchrone en JavaScript ?**
    A. Event Loop
    B. Call Stack
    C. Promise Queue
    D. Heap

19. **Lesquels de ces opérateurs sont spécifiques à ES2020 et ES2021 ?**
    A. ** (exponentiation)
    B. ?? (nullish coalescing)
    C. ?. (optional chaining)
    D. :: (inexistant standard)

20. **Lesquels de ces objets sont des collections introduites en ES6 ?**
    A. Map
    B. Set
    C. WeakMap
    D. WeakSet

21. **Lesquels de ces mots-clés sont des instructions de saut ?**
    A. break
    B. continue
    C. exit
    D. return

22. **Lesquelles de ces lignes sont correctes pour convertir une chaîne en nombre ?**
    A. Number("42")
    B. parseInt("42")
    C. "42" * 1
    D. toInteger("42")

23. **Quels objets permettent de stocker des paires clé-valeur ?**
    A. Object
    B. Map
    C. WeakSet
    D. WeakMap

24. **Lesquelles de ces syntaxes sont valides pour une fonction fléchée ?**
    A. () => {}
    B. (x) => x * 2
    C. (x, y) => { return x + y }
    D. x => { x * 2 } (piège : manque le return)

25. **Lesquels de ces éléments sont utilisés pour la gestion des modules en JavaScript ?**
    A. import
    B. require (CommonJS côté Node.js)
    C. export
    D. module.exports (CommonJS)

26. **Lesquels de ces types de données sont immuables ?**
    A. string
    B. boolean
    C. object
    D. number

27. **Dans un try...catch, que se passe-t-il si le try ne lève aucune exception ?**
    A. Le catch est ignoré
    B. Le catch est exécuté quand même
    C. Le code se poursuit normalement
    D. Une erreur est levée si aucun finally

28. **Lesquels de ces événements JavaScript sont liés au clavier ?**
    A. keydown
    B. keypress
    C. keyup
    D. onload

29. **Lesquels de ces statements contrôlent le flux conditionnel ?**
    A. if / else
    B. switch / case
    C. for / while
    D. throw / catch

30. **Lesquelles de ces méthodes peuvent ajouter des éléments à un tableau ?**
    A. push()
    B. unshift()
    C. concat()
    D. slice()

31. **Lesquels de ces mots-clés JavaScript sont liés à la programmation orientée objet (ES6) ?**
    A. class
    B. extends
    C. super
    D. prototype

32. **Lesquels de ces objets sont des constructeurs natifs en JavaScript ?**
    A. Array
    B. Object
    C. Function
    D. React

33. **Lesquelles de ces méthodes peuvent transformer un tableau en chaîne de caractères ?**
    A. join()
    B. toString()
    C. pop()
    D. toLocaleString()

34. **Lesquels de ces mots-clés gèrent l'asynchronisme par promesses ?**
    A. async
    B. await
    C. then
    D. sync

35. **Lesquels de ces éléments sont nécessaires pour écouter un événement en DOM ?**
    A. Élément cible
    B. Type d'événement (ex: click)
    C. Gestionnaire de rappel (callback)
    D. return

36. **Lesquels de ces types de boucles permettent de parcourir un tableau directement ?**
    A. for
    B. for...in (plutôt pour objets)
    C. for...of
    D. while

37. **Lesquels de ces éléments sont spécifiques aux classes en ES6 ?**
    A. Constructeur constructor
    B. Méthodes statiques static
    C. Héritage via extends
    D. Héritage multiple natif

38. **Lesquels de ces frameworks ou bibliothèques sont basés sur JavaScript (hors Node.js) ?**
A. React
B. Angular
C. Vue.js
D. Django

39. **Lesquels de ces mots-clés sont valides pour la déclaration d'une variable en ES5 et ES6 ?**
A. var (ES5)
B. let (ES6)
C. const (ES6)
D. final

40. **Lesquelles de ces méthodes sont utilisées pour parcourir un tableau et peuvent renvoyer un nouveau tableau ?**
A. map()
B. filter()
C. reduce() (retourne une valeur unique)
D. forEach() (ne renvoie pas de nouveau tableau)

41. **Lesquels de ces opérateurs comparent en incluant une conversion de type ?**
A. ==
B. ===
C. !=
D. !==

42. **Lesquels de ces objets appartiennent à l'API Web Storage ?**
A. sessionStorage
B. localStorage
C. cookieStorage (n'existe pas)
D. indexedDB (API de stockage, mais différent)

43. **Quels éléments de syntaxe permettent d'écrire une chaîne multi-ligne ?**
A. Backticks (templates strings) ` ` B.\ndans une chaîne simple C."""triple-quote (inexistant en JS) D.+` pour concaténer

44. **Lesquels de ces objets sont natifs du moteur JavaScript (ECMAScript) et non spécifiques au DOM ?**
A. Math
B. Date
C. document
D. JSON

45. **Quelles syntaxes sont valides pour déclarer une fonction nommée ?**
A. function maFonction() {}
B. maFonction = () => {} (variable ou fonction fléchée)
C. function() => {} (invalide)
D. let maFonction = function(){}

46. **Lesquels de ces aspects concernent la "portée" (scope) en JavaScript ?**
A. Portée globale
B. Portée de bloc avec let
C. Portée de fonction avec var
D. Portée d'objet

47. **Lesquelles de ces syntaxes permettent de faire un commentaire en JavaScript ?**
A. // commentaire
B. /* commentaire */
C. <!-- commentaire --> (plutôt HTML)
D. # commentaire (style de commentaire en Python)

48. **Lesquels de ces éléments peuvent être passés comme paramètres à une fonction ?**
A. Des nombres
B. Des chaînes
C. Des objets
D. D'autres fonctions

49. **Lesquelles de ces méthodes peuvent trier un tableau ?**
A. sort()
B. reverse() (inverse l'ordre, pas un tri sémantique)
C. order() (inexistant)
D. localeCompare() (méthode de string, pas d'array)

50. **Quels mécanismes permettent la gestion asynchrone en JavaScript ?**
A. Callbacks
B. Promises
C. Async/await
D. Threads multiples par défaut

Partie 3 : 100 Questions à Choix Unique

1. **(CU) Dans une fonction fléchée, quelle est la valeur de this ?**
A. Dépend du contexte d'appel
B. Hérite du this englobant
C. Toujours l'objet global
D. Inexistant, lève une erreur

2. **(CU) Quelle syntaxe permet de définir une fonction auto-invoquée (IIFE) ?**
A. function() { ... }();
B. (function() { ... })();
C. (() => { ... })();
D. auto function() { ... }

3. **(CU) Lequel de ces mots-clés est utilisé pour marquer une fonction asynchrone ?**
A. await
B. async
C. defer
D. promise

4. **(CU) Comment appelle-t-on la fonction spéciale qui s'exécute avant toute chose lorsqu'on utilise CommonJS dans Node.js ?**
   A. La fonction main()
   B. Le module require()
   C. L'IIFE implicite du module
   D. La méthode init()

5. **(CU) Dans un générateur, quel mot-clé permet d'émettre une valeur et de mettre la fonction en pause ?**
   A. yield
   B. return
   C. break
   D. async

6. **(CU) Quelle propriété d'un objet Promise indique son état courant (pending, fulfilled, ou rejected) ?**
   A. status
   B. state
   C. Il n'existe pas de propriété publique pour ça
   D. result

7. **(CU) Dans un module ES (ESM), comment déclare-t-on un export par défaut ?**
   A. export = ...
   B. module.exports = ...
   C. export default ...
   D. exports.default = ...

8. **(CU) Quelle méthode de l'Array est utilisée pour vérifier si tous les éléments satisfont une condition ?**
   A. every()
   B. some()
   C. filter()
   D. all()

9. **(CU) Comment appelle-t-on la version minimaliste de JavaScript sans framework ?**
   A. ES6 JS
   B. Vanilla JS
   C. Pure JS
   D. Node JS

10. **(CU) Quelle méthode de l'Array est utilisée pour aplatir un tableau de tableaux en profondeur 1 par défaut ?**
    A. flat()
    B. reduce()
    C. flatten()
    D. spread()

11. **(CU) Quel mot-clé CommonJS permet d'importer un module dans un fichier Node.js ?**
    A. import
    B. include
    C. require
    D. use

12. **(CU) Comment appelle-t-on la fonctionnalité qui propage les arguments d'un tableau dans une fonction ?**
    A. L'opérateur spread ...
    B. La méthode apply()
    C. L'opérateur rest ...
    D. La déstructuration de tableau

13. **(CU) Dans quel cas peut-on utiliser le mot-clé await ?**
    A. Dans n'importe quelle fonction
    B. Uniquement dans une fonction marquée async
    C. Dans un bloc if
    D. Dans une fonction fléchée classique

14. **(CU) Quelle est la sortie de la fonction fléchée (() => 42)() ?**
    A. undefined
    B. 42
    C. Erreur de syntaxe
    D. Objet Promise

15. **(CU) Quel est le mot-clé propre au format ESM pour exporter plusieurs membres nommés ?**
    A. exports = ...
    B. module.exports = ...
    C. export { ... }
    D. default export

16. **(CU) Pour gérer des actions asynchrones successives, laquelle de ces syntaxes est la plus moderne ?**
    A. Les callbacks imbriqués
    B. Les promesses chaînées
    C. L'utilisation de async/await
    D. L'utilisation de setTimeout

17. **(CU) Dans une IIFE, les variables déclarées avec var sont :**
    A. Globales
    B. Accessibles uniquement dans la fonction IIFE
    C. Inexistantes, c'est interdit
    D. Converties en let

18. **(CU) Quel est l'effet de return dans une fonction génératrice ?**
    A. Émet une dernière valeur et continue
    B. Termine immédiatement le générateur
    C. Passe à l'instruction yield suivante
    D. Provoque une erreur

19. **(CU) Dans un module CommonJS, comment définir un export principal ?**
    A. exports = something
    B. module.exports = something
    C. export default something
    D. require.exports = something

20. **(CU) Comment appellerait-on un concept pour lier un module ESM dans un navigateur ?**
    A. <script type="module">
    B. <script module="esm">
    C. <script type="javascript-module">
    D. <script esmodule>

21. **(CU) Quelle méthode peut transformer un tableau en un seul résultat (accumulateur) ?**
    A. some()
    B. reduce()
    C. every()
    D. map()

22. **(CU) Une fonction auto-invoquée (IIFE) a pour but principal de :**
    A. Déclarer un module ESM
    B. Éviter la pollution de l'espace global
    C. Empêcher toute réutilisation de la fonction
    D. Remplacer une fonction fléchée

23. **(CU) Lequel n'est pas un mot-clé permettant la déstructuration ?**
    A. let { x } = obj;
    B. let [y] = arr;
    C. const { z } = obj;
    D. unpack { a } = obj;

24. **(CU) Dans un async function, que retourne par défaut un appel à cette fonction ?**
    A. Une promesse
    B. Un générateur
    C. Le résultat brut de la fonction
    D. Un tableau

25. **(CU) Dans un générateur, l'objet retourné par generator.next() contient :**
    A. Un champ value et un champ done
    B. Toujours un champ error
    C. Un champ result et un champ finished
    D. Un champ await

26. **(CU) Lequel de ces exports ESM est valide ?**
    A. export function maFonction() {}
    B. export var x = 10;
    C. exports.x = 10;
    D. export default class {}

27. **(CU) Que fait la méthode Array.prototype.some() ?**
    A. Teste si au moins un élément satisfait la condition
    B. Transforme chaque élément pour en faire un nouvel array
    C. Vérifie si tous les éléments satisfont une condition
    D. Filtre les éléments

28. **(CU) Quelle syntaxe de déclaration est spécifique à CommonJS ?**
    A. import ... from ...
    B. export default ...
    C. module.exports = ...
    D. export const x = ...

29. **(CU) Dans une IIFE, comment appelle-t-on la portion (function() { ... }) sans les () finaux ?**
    A. Expression de fonction anonyme
    B. Déclaration de fonction globale
    C. Fonction fléchée
    D. Fonction génératrice

30. **(CU) Dans un générateur, quel mot-clé permet de recevoir une valeur depuis l'extérieur ?**
    A. external
    B. yield
    C. next()
    D. Le paramètre value de la fonction génératrice

31. **(CU) Dans un module ES, comment importer la totalité des exports nommés dans un objet ?**
    A. import { * } from "module";
    B. import * as Mod from "module";
    C. require("module").all
    D. import all from "module";

32. **(CU) L'utilisation de return dans une méthode d'array comme forEach() est :**
    A. Possible pour sauter au prochain élément
    B. Ignorée par la méthode forEach()
    C. Termine la boucle
    D. Change la valeur de this

33. **(CU) Dans une fonction fléchée, si on ne met pas d'accolades et on écrit () => 42, que se passe-t-il ?**
    A. La fonction renvoie toujours undefined
    B. La fonction renvoie la valeur 42
    C. La fonction ne compile pas
    D. On obtient un objet Promise

34. **(CU) Quel est le nom du module standard de Node.js pour manipuler les chemins de fichier ?**
    A. file
    B. fs
    C. path
    D. url

35. **(CU) Dans un script ES Module côté navigateur, si on utilise import, à quel moment le module est chargé ?**
    A. Dès que le script est lu, de façon asynchrone
    B. Après l'événement DOMContentLoaded
    C. Uniquement quand la page est entièrement chargée
    D. Jamais sans attribut defer

36. **(CU) Quelle méthode de l'Array invoque une fonction pour chaque élément et accumule un seul résultat final ?**
    A. reduce()
    B. map()
    C. every()
    D. some()

37. **(CU) Le mot-clé yield* dans un générateur sert à :**
    A. Déclencher une erreur de syntaxe
    B. Déléguer à un autre générateur ou itérable
    C. Forcer la fermeture du générateur
    D. Émettre une valeur et stopper la fonction

38. **(CU) Dans un module CommonJS, __filename et __dirname représentent :**
A. Le nom et le dossier du fichier courant
B. Les variables d'environnement
C. Des alias pour fs
D. N'existent pas, c'est ESM seulement

39. **(CU) Quelle est la principale différence entre map() et forEach() ?**
A. map() modifie le tableau original
B. map() renvoie un nouveau tableau, forEach() ne retourne rien
C. forEach() est plus rapide
D. forEach() autorise le break

40. **(CU) Quel est le bon nom pour la fonction auto-invoquée suivante : +function(){ console.log("test"); }() ?**
A. IIFE
B. Arrow function
C. Promise chain
D. Classe anonyme

41. **(CU) Que se passe-t-il si l'on utilise await en dehors d'une fonction asynchrone (sans transpilation particulière) ?**
A. Ça fonctionne en mode strict
B. On obtient une erreur de syntaxe
C. On obtient undefined
D. L'exécution est bloquée

42. **(CU) Dans un générateur, yield peut :**
A. Renvoyer une valeur au next() appelant
B. Émettre des erreurs
C. Changer la valeur de this
D. Être omis si on ne veut rien retourner

43. **(CU) Quel fichier ou champ précise l'entrée de module principal dans un package Node.js ?**
A. .npmrc
B. index.json
C. La propriété "main" dans package.json
D. module.config

44. **(CU) Dans une fonction fléchée, le mot-clé arguments :**
A. Se réfère à la liste des arguments
B. Est inexistant (lèvera une erreur si utilisé directement)
C. Contient un objet de type array
D. Est un alias pour rest

45. **(CU) Lequel n'est pas un type de module reconnu nativement en JavaScript ?**
A. CommonJS
B. ESM
C. AMD (Asynchronous Module Definition) dans l'environnement Node.js sans loader spécifique
D. UMD

46. **(CU) Lors de l'utilisation de Array.prototype.every(), quel résultat indique que la condition est vraie pour tous les éléments ?**
A. true
B. false
C. Un nouveau tableau filtré
D. Une exception en cas de faux

47. **(CU) Dans un script JavaScript Vanilla, la variable globale est placée dans :**
A. globalThis
B. window (dans le navigateur)
C. global (dans Node.js)
D. document

48. **(CU) Qu'arrive-t-il si on appelle deux fois de suite generator.return("Fin") ?**
A. Le générateur reprend son exécution
B. La première fois termine le générateur, la seconde n'a plus d'effet
C. Le générateur renvoie deux valeurs
D. Ça lève une erreur

49. **(CU) Dans le format CommonJS, quel objet contient les exports nommés ?**
A. export
B. exports
C. module.export
D. module.defaults

50. **(CU) Quelle méthode d'Array renvoie le premier index correspondant à l'élément cherché ?**
A. findIndex()
B. indexOf()
C. search()
D. includes() (retourne vrai/faux, pas un index)

51. **(CU) Dans l'écosystème Node.js, l'usage de require (CommonJS) est remplacé au profit de :**
A. import (ESM)
B. importScripts
C. loadModule
D. module.import

52. **(CU) Qu'arrive-t-il quand on fait return dans un générateur ?**
A. On ajoute une valeur émise avant de continuer
B. Le générateur se termine immédiatement
C. On peut relancer le générateur par next()
D. On déclenche une exception

53. **(CU) Quelle option n'existe pas dans la syntaxe d'import ESM ?**
A. import * as mod from "module";
B. import defaultMember from "module";
C. import { namedMember } from "module";
D. import = require("module")

54. **(CU) Les fonctions fléchées :**
A. Ont leur propre this
B. Héritent du this lexical
C. Sont toujours asynchrones
D. Ne peuvent pas être nommées

55. **(CU) La fonction auto-invoquée (IIFE) suivante : (function(x){ console.log(x); })(5);**

A. Affiche 5
B. Provoque une erreur
C. Renvoie undefined en console
D. Est équivalente à une fonction fléchée

56. **(CU) Dans un async function, que renvoie la ligne return 10; ?**
A. 10 directement
B. Une promesse résolue avec la valeur 10
C. Une promesse rejetée
D. Une valeur undefined

57. **(CU) Pour itérer sur un générateur gen, on utilise :**
A. gen.run()
B. gen.call()
C. gen.next()
D. gen.continue()

58. **(CU) En CommonJS, si on veut exporter plusieurs choses, on peut faire :**
A. exports.obj = { ... }; exports.func = function(){};
B. module.exports = { obj: ..., func: ... };
C. export default { obj, func };
D. module.exportMany(...)

59. **(CU) La méthode Array.prototype.reduce() prend combien de paramètres pour la fonction de rappel ?**
A. 1 (l'élément courant)
B. 2 (l'accumulateur et l'élément courant)
C. 4 (accumulateur, valeur courante, index, tableau)
D. 3 (accumulateur, valeur courante, tableau)

60. **(CU) Comment s'appelle le "style" d'écriture JavaScript sans framework, ni librairie spécifique ?**
A. Node.js core
B. ES6 standard
C. Vanilla JS
D. Pure ECMAScript

61. **(CU) Quelle clause permet de capturer une erreur dans un bloc try ?**
A. catch
B. error
C. finally
D. rescue

62. **(CU) Dans une fonction fléchée à un seul paramètre, que peut-on omettre ?**
A. Les parenthèses autour du paramètre
B. Le mot-clé function
C. Le return si on met des accolades
D. Les accolades si on renvoie directement une valeur

63. **(CU) Dans un générateur, comment transmettre une valeur externe lors de l'appel next() ?**
A. generator.send(value)
B. generator.next(value)
C. generator.input = value
D. On ne peut pas transmettre de valeur

64. **(CU) Quelle syntaxe utilise-t-on pour déclarer un module ESM côté Node.js (dans un package.json) ?**
A. "type": "module"
B. "esm": true
C. "modules": "import"
D. "moduleType": "ES6"

65. **(CU) Sur quelle méthode d'Array se base-t-on pour vérifier si au moins un élément remplit une condition ?**
A. find()
B. some()
C. map()
D. every()

66. **(CU) Comment appelle-t-on la syntaxe (function(){ ... })() dans la documentation ?**
A. Arrow expression
B. IIFE (Immediately Invoked Function Expression)
C. Async closure
D. Module pattern

67. **(CU) Dans une fonction async, utiliser throw new Error("...") :**
A. Rejette la promesse retournée
B. Résout la promesse avec un objet Error
C. Interrompt l'exécution sans effet sur la promesse
D. Génère un warning sans stopper l'exécution

68. **(CU) Pour collecter tous les exports d'un module ESM dans un objet, on écrit :**
A. import * from "module"
B. import * as obj from "module"
C. import { ...all } from "module"
D. import { default as obj } from "module"

69. **(CU) Array.prototype.every() retourne true si :**
A. Au moins un élément satisfait la condition
B. Tous les éléments satisfont la condition
C. Aucun élément ne satisfait la condition
D. Le tableau est vide

70. **(CU) La forme (function myIIFE(){ ... }()) est :**
A. Une fonction nommée IIFE
B. Impossible, lève une erreur
C. Un module ESM
D. Un alias de la fonction fléchée

71. **(CU) Dans une fonction fléchée () => ({ x: 1 }), à quoi servent les parenthèses autour du bloc { x: 1 } ?**
A. À retourner un objet littéral
B. À créer un bloc de code vide
C. À ignorer la valeur
D. À activer le mode strict

72. **(CU) Quel est le cycle de vie d'un générateur ?**
A. start -> run -> complete -> restart
B. suspendedStart -> executing ->

suspendedYield -> completed
C. Il ne peut jamais compléter
D. pending -> fulfilled -> rejected

73. **(CU) Dans Node.js, comment activer l'ESM sans package.json "type": "module" ?**
A. Utiliser l'extension .mjs
B. Mettre "moduleMode": true
C. Passer un flag --commonjs
D. On ne peut pas

74. **(CU) Quelle méthode d'Array coupe un tableau en plusieurs morceaux qu'elle retourne comme sous-tableaux ?**
A. slice()
B. splice()
C. split() (non, c'est string)
D. Il n'y a pas de méthode native

75. **(CU) Dans une IIFE, pour exposer certaines fonctionnalités, on utilise souvent :**
A. Un return d'un objet public
B. Un window.myGlobal = ...
C. module.exports
D. export default

76. **(CU) Dans un async function, quand on fait return await X;, c'est équivalent à :**
A. return X; (sauf si on veut capturer l'erreur)
B. throw X;
C. return Promise.all([X]);
D. return new Promise(X);

77. **(CU) Dans un générateur, la forme yield* autreGenerateur() signifie :**
A. Qu'on concatène les valeurs des deux générateurs
B. Qu'on délègue l'itération à autreGenerateur()
C. Qu'on exécute en parallèle deux générateurs
D. Qu'on crée un objet Promise

78. **(CU) CommonJS est le format de modules historique de :**
A. Node.js
B. Les navigateurs
C. ECMAScript 2015
D. AMD

79. **(CU) array.reduce((acc, val) => acc + val, 0) fait :**
A. La somme des éléments du tableau
B. La concaténation des éléments
C. Filtre les éléments nuls
D. Crée un objet

80. **(CU) Quelle expression fléchée ne renvoie pas undefined par défaut ?**
A. () => { 42 }
B. () => 42
C. (x) => { x + 1 }
D. () => { return }

81. **(CU) Dans une IIFE, comment rendre la fonction disponible globalement ?**
A. var maIIFE = function(){...}()
B. window.maIIFE = (function(){...})(); (en navigateur)
C. export IIFE
D. module.exports = IIFE

82. **(CU) Dans un générateur, appeler deux fois next() de suite sans yield entre-temps :**
A. Fait avancer deux fois la machine d'état
B. Lève une exception
C. Met le générateur en pause
D. Donne toujours done: true

83. **(CU) Que fait la méthode Object.freeze(obj) ?**
A. Rend les propriétés de obj modifiables uniquement en lecture
B. Transforme obj en null
C. Rend obj immuable (pas de modifications possibles)
D. Copie obj dans un nouvel objet

84. **(CU) Dans un script ESM, import "./module.js"; sans accolades ni nom :**
A. Importe le module par défaut
B. N'importe rien, mais exécute le fichier module.js
C. Est invalide en ES
D. Importe tous les exports nommés

85. **(CU) La méthode Array.prototype.every() :**
A. Retourne true si tous les éléments passent le test
B. Retourne un nouveau tableau
C. Se termine dès qu'un élément échoue au test
D. Lance une exception si le test échoue

86. **(CU) Comment appelle-t-on l'opérateur ... dans (a, b, ...rest) => {} ?**
A. Spread operator
B. Rest parameter
C. Template operator
D. Destructuring operator

87. **(CU) Lorsqu'une fonction génératrice est "complétée", generator.next() renvoie :**
A. { value: undefined, done: true }
B. Un objet Promise
C. Une erreur StopIteration
D. { value: null, done: false }

88. **(CU) CommonJS supporte quel type d'import ?**
A. import ... from "..."
B. require("...")
C. import { ... }
D. module import

89. **(CU) Sur un tableau [10, 20, 30], myArray.some(x => x > 15) renvoie :**
A. true
B. false
C. undefined
D. Erreur

90. **(CU) L'IIFE (function namedIIFE() { ... }()) se différencie de (function() { ... })() par :**

A. Le fait qu'elle a un nom local qu'on peut appeler en récursif
B. Une syntaxe non valide en JS
C. Une exécution bloquante
D. Une portée globale

91. **(CU) Dans async function x() { return "ok"; }, x() renvoie :**
A. Une promesse résolue avec "ok"
B. La chaîne "ok" directement
C. Une erreur
D. Un générateur

92. **(CU) Que fait la méthode generator.throw(error) ?**
A. Lance une exception depuis le générateur à l'endroit du dernier yield
B. Redémarre le générateur depuis le début
C. Met le générateur en pause
D. Transforme l'erreur en warning

93. **(CU) Dans un module CommonJS, require.cache contient :**
A. Les données chargées depuis les modules
B. Les chemins de modules résolus
C. Les modules déjà chargés pour éviter de les recharger
D. Un tampon vide

94. **(CU) array.reduce((acc, val) => acc + val) sans second argument :**
A. Utilise 0 comme accumulateur initial
B. Utilise le premier élément comme accumulateur initial
C. Lève une erreur
D. Utilise null comme accumulateur

95. **(CU) Dans une fonction fléchée, comment renvoyer un objet littéral ?**
A. () => { key: "value" }
B. () => ({ key: "value" })
C. () => "key": "value"
D. function => { key: "value" }

96. **(CU) (() => { console.log("Hello"); })() est un exemple de :**
A. Fonction auto-invoquée fléchée
B. IIFE standard
C. Module ESM
D. Méthode asynchrone

97. **(CU) Une fonction async peut contenir :**
A. Des mots-clés await
B. Des boucles for, while
C. Des instructions return
D. Un objet Promise implicite

98. **(CU) Dans un générateur, comment récupérer la valeur finale après return ?**
A. Par generator.value
B. Par generator.returnValue
C. Via l'objet { value, done } de next()
D. On ne peut pas la récupérer

99. **(CU) Lequel n'est pas un export valide en ESM ?**
A. export const MY_VAR = 42;
B. export function doStuff() {}
C. export default function() {}
D. module.export = ...

100. **(CU) Dans l'écosystème JavaScript, "Vanilla JS" signifie :**
A. Un nouveau framework minimaliste
B. Du JavaScript natif sans librairie tierce
C. Du code Node.js
D. Une architecture React custom

Partie 4 : 20 Questions à Choix Unique (suite)

(Toujours 1 seule bonne réponse, niveau plus avancé/varié.)

1. **(CU) Quelle syntaxe de déstructuration permet de renommer une variable en l'extrayant ?**
A. const { prop: alias } = obj;
B. const [alias : prop] = arr;
C. const { rename = prop } = obj;
D. alias = obj.prop;

2. **(CU) Que signifie l'opérateur ?. (optional chaining) ?**
A. Permet d'enchaîner des appels de fonction de manière synchrone
B. Évite les erreurs si une propriété est null ou undefined
C. Convertit un nombre en entier
D. Compare deux objets

3. **(CU) Que fait Array.prototype.flatMap() ?**
A. Combine map() puis flat() à une profondeur de 1
B. Tri le tableau avant d'appliquer map()
C. Réalise un reduce() en combinant les éléments
D. Décompose un objet en plusieurs tableaux

4. **(CU) Dans une fonction fléchée, (() => this)() renvoie :**
A. L'objet global en mode non strict
B. undefined
C. Le this lexical englobant
D. Un objet Promise

5. **(CU) Dans un module ES, la directive export * from "./autre.js"; :**
A. Importe d'abord tout de autre.js puis l'exporte
B. Doit être suivie d'une liste de membres
C. N'est pas autorisée
D. Implémente un alias

6. **(CU) L'utilisation de await Promise.all([...]) dans une fonction async sert à :**
A. Exécuter les promesses en séquentiel

B. Exécuter les promesses en parallèle et attendre la fin de toutes
C. Cloner plusieurs objets
D. Lancer une unique promesse

7. **(CU) Dans le standard ECMAScript, les générateurs ont été introduits en :**
A. ES5
B. ES6 (ES2015)
C. ES7 (ES2016)
D. ES8 (ES2017)

8. **(CU) module.exports est spécifique à :**
A. ESM
B. AMD
C. CommonJS
D. UMD

9. **(CU) Array.prototype.map() :**
A. Transforme chaque élément et renvoie un nouveau tableau
B. Filtre les éléments nuls
C. Retourne un objet itérateur
D. Modifie le tableau initial

10. **(CU) Comment appeler une fonction auto-invoquée fléchée ?**
A. (=> console.log("test"))();
B. (() => { console.log("test"); })();
C. auto=>{...}();
D. invoke(()=>{...});

11. **(CU) Dans une fonction génératrice, function* gen() { yield 1; yield 2; }, que renvoie gen().next() ?**
A. { value: undefined, done: false }
B. { value: 1, done: false }
C. { value: 1, done: true }
D. { done: true }

12. **(CU) En ESM, comment importer un export par défaut et d'autres membres nommés ?**
A. import default, { named1, named2 } from "module";
B. import { default, named } from "module";
C. import default, named from "module";
D. import default * from "module";

13. **(CU) Array.prototype.some() renvoie true quand :**
A. Tous les éléments passent le test
B. Au moins un élément passe le test
C. Aucun élément ne passe le test
D. Le tableau est vide

14. **(CU) (() => 42)() retourne :**
A. 42
B. undefined
C. Une Promise
D. Un générateur

15. **(CU) Dans une IIFE, var x = 10;, que devient x après exécution ?**
A. Global si on n'est pas en mode strict
B. Locale au bloc IIFE
C. Écrase la variable x si elle existait en dehors
D. Propriété de window.x obligatoirement

16. **(CU) Dans un async function, comment gérer une erreur de promesse rejetée ?**
A. Avec un bloc try...catch
B. Avec promise.onError(...)
C. Impossible, la fonction se termine sans faute
D. En remplaçant return par throw

17. **(CU) Dans une fonction génératrice function* g() { yield 1; return 2; }, quel est le dernier value émis ?**
A. 1
B. 2
C. Aucun, return ne fait pas partie de la séquence des yield
D. Erreur

18. **(CU) CommonJS a historiquement été créé pour :**
A. Les navigateurs avant ES6
B. Node.js
C. Les modules AMD
D. Le Java côté serveur

19. **(CU) Array.prototype.every() retourne false si :**
A. Tous les éléments valident le test
B. Un seul élément ne valide pas le test
C. Le tableau est vide
D. On n'a pas de callback

20. (CU) Les fonctions fléchées sont particulièrement utiles pour :
A. Fournir un this dynamique
B. Simplifier l'écriture de callbacks
C. Générer des classes
D. Remplacer async/await

21. Dans le DOM, quelle méthode renvoie le premier élément correspondant à un sélecteur CSS ?
A. document.getElementById()
B. document.querySelector()
C. document.querySelectorAll()
D. document.getElementsByClassName()

22. Dans un document HTML, la propriété document.body fait référence :
A. À l'élément <head>
B. À l'élément <html>
C. À l'élément <body>
D. À tous les éléments

23. Quelle est la bonne syntaxe pour ajouter un écouteur d'événement "click" sur un bouton en JavaScript Vanilla ?
A. button.addListener("click", function() { ... });
B. button.on("click", function() { ... });
C. button.addEventListener("click", function() { ... });
D. addEvent(button, "click", fn);

24. (CU) Dans une page HTML, le DOM est entièrement chargé. Quel événement du document peut-on écouter

pour exécuter du code après le parsing initial du HTML ?
A. load
B. DOMContentLoaded
C. onready
D. beforeunload

25. (CU) Dans un script JavaScript, l'accès à document.querySelectorAll("p") renvoie :
A. Un tableau standard
B. Un objet NodeList
C. Une HTMLCollection
D. Une exception si le sélecteur est multiple

26. (CU) Quelle méthode du DOM insère un nouvel élément en tant qu'enfant à la fin d'un nœud parent ?
A. parent.replaceChild(newNode, oldNode)
B. parent.appendChild(newNode)
C. parent.insertAdjacentHTML("beforeend", newNode)
D. parent.addChild(newNode)

27. (CU) Dans un gestionnaire d'événements DOM, le paramètre généralement nommé event fait référence à :
A. Une chaîne contenant le type d'événement
B. L'objet global window
C. L'objet décrivant l'événement (position souris, touche pressée, etc.)
D. N'existe pas en JavaScript Vanilla

28. (CU) Dans le DOM, element.style.backgroundColor = "red"; :
A. Modifie le style inline du composant
B. Lance une exception car background-color est invalide
C. Modifie le fichier CSS externe
D. Ne fait rien, car les propriétés CSS sont en lecture seule

29. (CU)

Pour empêcher la propagation d'un événement vers les éléments parents, on utilise :
A. event.preventDefault()
B. event.stopPropagation()
C. event.cancel()
D. event.defaultPrevented = true

30. (CU)

Dans le DOM, element.classList.add("active") :
A. Ajoute la classe active à l'attribut class de l'élément
B. Remplace toutes les classes existantes
C. Ajoute un style en ligne "class: active;"
D. Efface d'abord toutes les classes, puis ajoute active

31. (CU)

En JavaScript Vanilla, quelle méthode du DOM supprime un nœud enfant d'un parent ?
A. parent.remove(child)
B. child.remove()
C. parent.removeChild(child)
D. child.delete()

32. (CU) Dans les événements DOM, l'objet event.target pointe vers :
A. L'élément sur lequel l'écouteur a été attaché
B. L'élément qui a réellement déclenché l'événement
C. L'objet global document
D. La fenêtre window

33. (CU) Quelle propriété du DOM permet de lire ou modifier le code HTML interne d'un élément ?
A. element.innerHTML
B. element.value
C. element.textContent
D. element.htmlContent

34. (CU) Dans un document, document.documentElement correspond généralement :
A. À l'élément <body>
B. À l'élément <head>
C. À l'élément <html>
D. À la racine du DOM, qui n'existe plus en HTML5

35. (CU) Pour enregistrer une fonction qui s'exécute **avant** que la fenêtre ne se ferme (ex. un avertissement), on utilise l'événement :
A. beforeunload
B. unload
C. onclose
D. window.exit

36. (CU) Dans un navigateur moderne, si on veut exécuter un code lorsque l'utilisateur clique **n'importe où** dans la fenêtre, on peut :
A. document.onGlobalClick = fn;
B. window.addEventListener("click", fn);
C. document.windowClick(fn);
D. document.event("click", fn);

37. (CU) Comment récupérer le contenu texte (sans HTML) d'un élément DOM ?
A. element.innerHTML
B. element.textContent
C. element.htmlText
D. element.value

38. (CU) Dans le DOM, querySelector("div > p") recherche :
A. Tous les paragraphes de la page
B. Le premier <p> descendant direct d'un
C. Le premier <p> dans un même profondément
D. Soulève une exception si plusieurs <p> sont trouvés

39. (CU) Laquelle de ces propriétés **DOM** n'existe pas ?
A. element.id
B. element.className
C. element.nodeName
D. element.motherNode

40. (CU) Quelle est la valeur de document.readyState lorsque la page est entièrement chargée ?
A. "ready"
B. "complete"

C. "loaded"
D. "interactive"

41. (CU)

Dans une fonction fléchée, si on souhaite retourner un objet littéral, on doit :
A. Mettre l'objet entre parentheses : () => ({ a: 1 })
B. Mettre l'objet après return : () => return { a: 1 }
C. On ne peut pas retourner d'objets littéraux en fléché
D. Mettre l'objet en string JSON

42. (CU)

Lequel de ces attributs script permet de charger un module ECMAScript ?
A. type="application/js-module"
B. type="module"
C. module="true"
D. defer="module"

43. (CU)

Dans un module CommonJS, require est :
A. Une fonction synchrone qui charge un module et renvoie ses exports
B. Une fonction asynchrone renvoyant une promesse
C. Un mot-clé ECMAScript officiel
D. Inexistant, on doit utiliser import

44. (CU)

Pour itérer sur les propriétés énumérables d'un objet en JavaScript, on utilise :
A. for...of
B. for...in
C. Object.forEach()
D. obj.traverse()

45. (CU) Dans un contexte Node.js, si on veut **exporter** plusieurs éléments dans un module CommonJS, on peut faire :
A. module.exports = { fonction1, fonction2 };
B. exports.fonction1 = ...; exports.fonction2 = ...;
C. export { fonction1, fonction2 }
D. module.importAll = { ... }

46. (CU) Une fonction auto-invoquée (IIFE) est souvent utilisée pour :
A. Déclarer des variables globales persistantes
B. Éviter la pollution de l'espace global et isoler un scope
C. Gérer l'asynchronisme avec await
D. Charger un module ESM

47. (CU) Comment définit-on une fonction génératrice ?
A. function gen() { yield ... }
B. function* gen() { ... }
C. generate function() { ... }
D. (function()*) { yield ... }

48. (CU) Dans le DOM, element.setAttribute("style", "color: red"); :
A. Ajoute ou modifie l'attribut style en définissant la couleur rouge
B. Ne fonctionne que si l'élément est un <style>
C. Lance une erreur si l'attribut n'existe pas
D. Enlève tous les styles sauf "color:red"

49. (CU)

Une fonction fléchée ne peut pas être utilisée comme méthode d'un objet si :
A. On a besoin de this pour faire référence à l'objet
B. L'objet est déclaré en const
C. Elle est déclarée en mode strict
D. On y passe des paramètres rest

50. (CU) Dans un script ESM, import x from "./module.js"; signifie :
A. On importe l'export par défaut nommé x depuis module.js
B. On importe un export nommé x
C. On importe tous les exports dans un objet x
D. On importe la propriété x du window.module

51. (CU) Dans une boucle for...of, on peut itérer sur :
A. Un NodeList
B. Un tableau ([])
C. Un objet littéral {}
D. Une chaîne de caractères

52. (CU)

Pour créer un nœud texte dans le DOM, on utilise :
A. document.createTextNode("du texte")
B. new Text("du texte")
C. document.innerText = "du texte"
D. document.makeText("du texte")

53. (CU) Comment annuler l'action par défaut (ex: suivi du lien) dans un gestionnaire d'événement DOM ?
A. event.stopPropagation()
B. event.preventDefault()
C. return false (fonctionne partiellement dans jQuery, pas standard)
D. event.cancelBubble = true

54. (CU) Dans un module CommonJS, la variable qui référence l'objet d'exports est :
A. module.exports
B. export default
C. this.exports
D. Common.exports

55. (CU) Une fonction fléchée "courte" s'écrit :
A. () -> 42
B. (=> 42)
C. () => 42
D. {} => { return 42 }

56. (CU) Dans un générateur, pour émettre des valeurs successives, on utilise :

A. yield à chaque fois
B. await à chaque fois
C. this.value = ...
D. generator.push(...)

57. (CU) Dans un contexte Node.js en CommonJS, la variable __dirname correspond :
A. Au répertoire dans lequel Node.js est installé
B. Au répertoire courant du fichier en exécution
C. À la racine du projet
D. Au répertoire node_modules

58. (CU) Dans le DOM, document.getElementById("monId") renvoie :
A. Une NodeList
B. Le premier élément avec l'attribut name="monId"
C. L'élément ayant l'ID "monId", ou null si introuvable
D. Toujours un objet non-null, même s'il n'existe pas

59. (CU) Lors d'un événement DOM, l'ordre de propagation est :
A. Ciblage -> capture -> bouillonnement
B. Capture -> ciblage -> bouillonnement
C. Bouillonnement -> ciblage -> capture
D. Capture -> bouillonnement -> ciblage

60. (CU) Pour éviter la fermeture d'un générateur après un return, on :
A. Ne peut rien faire, return termine forcément
B. Utilise yield return
C. Utilise throw
D. On place un yield après le return

61. (CU)import * as MyModule from "./fichier.js"; :
A. Importe tous les exports nommés dans un objet MyModule
B. Importe l'export par défaut sous le nom MyModule
C. Importe tout (y compris le Node.js core)
D. N'est pas une syntaxe valide

62. (CU) Dans le DOM, un objet Event possède la propriété type qui indique :
A. Le type de l'élément ciblé
B. Le type de l'événement (ex: "click")
C. La classe CSS en cours
D. La nature de la propagation

63. (CU) Pour transformer un NodeList en vrai tableau, on peut utiliser :
A. Array.from(NodeList)
B. [...NodeList] (opérateur spread)
C. NodeList.slice()
D. NodeList.toArray()

64. (CU) Dans une fonction fléchée, this est :
A. Lexical, c'est-à-dire hérité du scope parent
B. Dynamique selon l'appel
C. Toujours undefined
D. Équivalent à arguments.callee

65. (CU) Un module ESM dans le navigateur se déclare via :
A. <script type="module" src="fichier.js"></script>
B. <module src="fichier.js"></module>
C. <script module="true" src="fichier.js"></script>
D. <script src="fichier.mjs"></script> (possible, mais sans type="module", pas un ESM)

66. (CU)L'événement "submit" se déclenche sur :
A. Un bouton <button>
B. Un formulaire <form>
C. Tout élément pouvant être cliqué
D. Le document tout entier

67. (CU) Dans un IIFE, la variable déclarée avec var x = 10; n'est accessible qu'à l'intérieur de :
A. L'IIFE
B. Toutes les fonctions du même fichier
C. L'objet window.x
D. Aucune, c'est illégal en mode strict

68. (CU) Une fonction fléchée ne possède pas de propriété prototype. Quelle conséquence cela a-t-il ?
A. On ne peut pas l'utiliser avec new pour instancier un objet
B. Elle hérite des prototypes globaux
C. Elle doit être déclarée en mode strict
D. On doit la déclarer en var

69. (CU) Quelle méthode du DOM efface tout le contenu HTML d'un élément ?
A. element.remove()
B. element.innerHTML = "";
C. element.textContent = "";
D. element.removeChildAll()

70. (CU) En CommonJS, si on assigne directement exports = { ... }, cela a pour effet :
A. De modifier l'export par défaut
B. De redéfinir la référence exports, sans modifier module.exports
C. D'écraser module.exports
D. D'ajouter un nouveau champ

71. (CU)

Un générateur s'utilise en JavaScript principalement pour :
A. Gérer l'asynchronisme plus simplement qu'avec async/await
B. Produire des valeurs à la demande (itération paresseuse)
C. Forcer un code synchrone
D. Éviter les blocages d'exécution

72. (CU)

(function test(){ ... })() :
A. Est une IIFE nommée, on peut s'y référer récursivement dans le code interne
B. Crée une variable globale test
C. Lève une erreur si on veut l'appeler de l'extérieur
D. Transforme test en variable locale du fichier

73. (CU)

Dans le DOM, document.createElement("p") :
A. Crée un élément paragraphe, non encore inséré

dans la page
B. Ajoute directement l'élément <p> dans <body>
C. Renvoie null si <p> existe déjà
D. Lève une exception si <p> n'existe pas

74. (CU)

Une fonction fléchée avec (...args) => {} signifie :
A. On collecte tous les arguments dans un tableau args
B. On applique le spread operator
C. On destructure un tableau [args]
D. Erreur de syntaxe

75. (CU) Dans un module ES, la syntaxe import { something as alias } from "mod"; fait :
A. Importe la variable alias depuis something
B. Importe something sous un autre nom alias
C. Fait un rename local something -> alias
D. Lève une erreur s'il y a déjà un export par défaut

76. (CU)L'événement "keyup" se produit :
A. Quand on appuie sur une touche
B. Quand on relâche une touche
C. Quand la touche est maintenue
D. Au clic de la souris

77. (CU) Dans le DOM, element.children renvoie :
A. Une collection HTML des enfants éléments (pas les nœuds texte)
B. Un tableau complet de tous les nœuds
C. null si l'élément n'a pas de
D. Une NodeList statique

78. (CU) Pour empêcher l'exécution par défaut d'un lien <a> lors d'un click, on fait :
A. event.returnValue = false;
B. event.preventDefault();
C. window.location = "#"
D. event.stopPropagation();

79. (CU) Dans un générateur, yield* est appelé "yield delegate". Cela signifie :
A. On délègue l'itération à un autre itérable ou un autre générateur
B. On double la valeur du yield
C. On arrête la boucle
D. On effectue du recouvrement d'exception

80. (CU) En CommonJS, pour importer un module local, on écrit :
A. import "./localModule.js";
B. require("./localModule.js");
C. module.import("./localModule");
D. load "./localModule";

81. (CU) Comment sélectionner **tous** les éléments <p> dans le DOM avec querySelectorAll ?
A. document.querySelectorAll("p")
B. document.queryAll("p")
C. document.query("p", true)
D. document.getElementsBy("p")

82. (CU) Dans une fonction fléchée, si on omet les accolades et on écrit (a) => a * 2;, le return est :
A. Implícite
B. Impossible, la fonction renvoie undefined
C. Automatique seulement si a est un nombre
D. Nécessite return a * 2;

83. (CU) Lorsqu'on définit un module ESM dans un fichier .mjs côté Node.js :
A. Node le lit comme un module ES
B. Node le lit en CommonJS
C. Node le rejette automatiquement
D. On doit le transpiler avec Babel

84. (CU)L'événement "change" sur un champ texte <input type="text"> se déclenche :
A. À chaque frappe
B. Quand on perd le focus après avoir modifié la valeur
C. Jamais en JavaScript Vanilla
D. Seulement si on utilise un <select>

85. (CU) Un générateur en JavaScript renvoie un objet :
A. Iterator
B. AsyncIterator
C. Promise
D. Observable

86. (CU)(() => { console.log("Salut"); })() :
A. Affiche "Salut" immédiatement (IIFE fléchée)
B. Ne s'exécute pas, c'est une erreur de syntaxe
C. Retourne toujours un booléen
D. Se lance au chargement de la page

87. (CU) Dans un contexte Node.js, comment activer l'export ESM dans un package.json ?
A. "modules": true
B. "type": "module"
C. "esm": "on"
D. "ecmaVersion": 2020

88. (CU) Dans le DOM, la propriété nodeType d'un nœud permet de :
A. Savoir si c'est un élément, un texte, un commentaire, etc.
B. Modifier la classe CSS
C. Obtenir la valeur du type d'un <input>
D. Charger un module ESM

89. (CU) Lorsque event.stopPropagation() est appelé dans un listener :
A. L'action par défaut est annulée
B. La phase de bouillonnement (ou capture) ne se propage plus
C. L'événement est supprimé
D. Les autres événements identiques sur l'élément sont bloqués

90. (CU) Dans une fonction fléchée, (a, b) => a + b, si on invoque (2, 3), on obtient :
A. 5
B. undefined

C. Une promesse résolue
D. Un itérateur

91. (CU) En CommonJS, si vous faites const monModule = require("./module"), comment accéder à un export nommé truc ?
A. monModule.truc
B. truc
C. module.truc
D. exports.truc

92. (CU) Si une fonction génératrice a émis toutes ses valeurs, l'appel suivant à next() renvoie :
A. { value: undefined, done: true }
B. NaN
C. Une erreur StopIteration
D. { done: false }

93. (CU)

Dans le DOM, element.insertAdjacentHTML("afterbegin", "Hello") :
A. Insère le HTML juste avant l'élément
B. Insère le HTML juste après l'élément
C. Insère le HTML comme premier enfant de l'élément
D. Insère en dernier enfant de l'élément

94. (CU)À l'intérieur d'une fonction fléchée, le mot-clé arguments :
A. Fonctionne comme dans une fonction classique
B. Est non défini (inexistant)
C. Correspond au tableau des paramètres rest
D. Pointeur vers le global

95. (CU) Un module ESM peut avoir :
A. Plusieurs exports nommés
B. Un seul export par défaut
C. Zéro, un ou plusieurs exports nommés
D. Plusieurs export default

96. (CU) Dans le DOM, element.firstChild renvoie :
A. Le premier nœud enfant (texte ou élément)
B. Le premier élément enfant
C. Toujours null si ce n'est pas un
D. NodeList

97. (CU) Pour "consommer" un générateur jusqu'au bout, on peut utiliser :
A. Une boucle while(!gen.done)
B. Une boucle for...of sur le générateur
C. gen.consumeAll()
D. Promise.resolve(gen)

98. (CU) Une IIFE fléchée s'écrit :
A. () => ()
B. (=>{})()
C. (() => { /* code */ })()
D. (() => { /* code */ }[])()

99. (CU) En Node.js (CommonJS), __filename représente :
A. Le chemin absolu du fichier en cours d'exécution
B. Le dossier du fichier
C. Toujours "index.js"
D. Le chemin relatif à package.json

100. (CU) Dans le DOM, document.createElement("script") :
A. Ajoute un <script> dans la page directement
B. Crée un élément <script> qu'on doit encore attacher au DOM
C. Ne fonctionne qu'avec type="module"
D. Lance l'exécution du script immédiatement

Partie 3 : 100 Questions à Choix Multiples

(Plusieurs réponses correctes possibles)

1. **(CM) Quelles syntaxes sont valides pour créer une IIFE ?**
   A. (function() { ... })();
   B. (() => { ... })();
   C. function iife() { ... } iife();
   D. function() { ... }() (invalide seul)

2. **(CM) Dans un générateur, quelles instructions peuvent coexister ?**
   A. yield
   B. return
   C. break
   D. yield*

3. **(CM) Quelles sont les différences entre CommonJS et ESM ?**
   A. CommonJS utilise require, ESM utilise import
   B. CommonJS charge de façon synchrone, ESM peut charger de façon asynchrone
   C. CommonJS est standard ECMAScript, ESM non
   D. ESM supporte import/export, CommonJS utilise module.exports

4. **(CM) Les méthodes d'Array qui ne modifient pas le tableau original sont :**
   A. map()
   B. filter()
   C. slice()
   D. pop()

5. **(CM) Lesquelles de ces syntaxes concernent une fonction fléchée ?**
   A. const f = () => {};
   B. (x) => x * 2
   C. function => { ... } (invalide)
   D. () => ({ a: 1 })

6. **(CM) Lesquels de ces mots-clés sont liés à l'asynchronisme en JavaScript ?**
   A. async
   B. await
   C. promise (pas un mot-clé, mais un concept)
   D. defer (attribut HTML, pas JavaScript)

7. **(CM) Dans l'écosystème Node.js, quelles sont des solutions de module courantes ?**
   A. CommonJS (CJS)
   B. ESM
   C. AMD (pas nativement pris en charge)
   D. UMD (souvent pour navig. + Node combinés)

8. **(CM) Lesquelles de ces méthodes d'Array retournent un nouveau tableau ?**
   A. map()
   B. filter()
   C. concat()
   D. sort() (modifie l'existant)

9. **(CM) Quelles fonctionnalités sont apportées par ES6 (ES2015) ?**
   A. Classes
   B. Arrow functions
   C. Generators
   D. Async/await (introduit ES2017)

10. **(CM) Dans une fonction génératrice, on peut :**
    A. Utiliser yield plusieurs fois
    B. Utiliser return pour terminer
    C. Appeler next() en dehors pour avancer
    D. Définir super() (non, c'est dans une classe)

11. (CM) Quelles syntaxes du DOM permettent d'écouter un événement ?

A. element.addEventListener("click", fn)

B. element.onclick = fn

C. document.on("click", fn) (pas standard)

D. element.addListener("click", fn) (pas standard)

12. (CM) Dans un générateur, que peuvent faire les méthodes next(), throw(), et return() ?

A. next() fait progresser le générateur jusqu'au prochain yield

B. throw() lance une exception à l'intérieur du générateur

C. return() termine le générateur

D. return() émet une dernière valeur dans le flux

13. (CM) Les modules CommonJS ont les caractéristiques suivantes :

A. Ils utilisent require pour importer

B. Ils utilisent module.exports ou exports pour exporter

C. Ils sont nativement supportés par tous les navigateurs modernes sans build

D. Ils fonctionnent historiquement avec Node.js

14. (CM) Les méthodes d'array suivantes créent toujours un nouveau tableau :

A. map()

B. filter()

C. slice()

D. forEach()

15. (CM) Lesquels de ces événements du DOM sont liés à la souris ?

A. click

B. wheel

C. mousemove

D. keypress (lié au clavier)

16. (CM) Lesquelles de ces méthodes ne modifient pas le tableau original ?

A. map()

B. filter()

C. sort() (modifie sur place)

D. slice()

17. (CM) Dans une fonction fléchée, on peut utiliser :

A. Des paramètres rest (x, ...others) => ...

B. Un retour implicite quand on n'utilise pas d'accolades

C. Le mot-clé this qui fait référence à l'objet global

D. Aucun return explicite si on place le bloc entre { } (ce bloc nécessite return)

18. (CM) En JavaScript, l'asynchronisme peut être géré par :

A. Les callbacks

B. Les promesses

C. async/await

D. Les IIFE

19. (CM) Quelles de ces syntaxes importent des membres nommés en ESM ?

A. import { a, b } from "module";

B. import * as M from "module"; (on récupère M.a, M.b, etc.)

C. import defaultExport from "module"; (importe l'export par défaut)

D. import all from "module"; (invalide)

20. (CM) Lesquels de ces event.preventDefault() et event.stopPropagation() sont corrects ?

A. preventDefault() annule l'action par défaut (ex: suivre un lien)

B. stopPropagation() arrête la phase de bouillonnement ou de capture

C. stopPropagation() annule aussi l'action par défaut

D. preventDefault() arrête la propagation de l'événement

21. (CM) Lesquels de ces événements sont déclenchés sur un <input type="text"> en saisie ?

A. input (à chaque modification)

B. change (quand le champ perd le focus ou qu'on valide la modif)

C. keyup (lorsqu'on relâche la touche)

D. dblclick (lié à la souris, pas au clavier)

22. (CM) Lesquels de ces objets DOM sont des collections dynamiques (se mettent à jour si le DOM change) ?

A. document.getElementsByTagName("div") (HTMLCollection dynamique)

B. document.querySelectorAll("div") (NodeList statique)

C. element.children (HTMLCollection dynamique)

D. element.childNodes (NodeList souvent live selon l'implémentation, mais pas toujours)

23. (CM) Lesquelles de ces méthodes de tableau prennent un callback (element, index, array) ?

A. map()

B. filter()

C. reduce() (prend (acc, val, idx, arr))

D. forEach()

24. (CM) Dans un générateur function* gen() { ... }, on peut :

A. Utiliser yield plusieurs fois

B. Faire un return pour terminer

C. Gérer les erreurs avec try...catch dans le générateur

D. Repartir du début quand on appelle next() (non, on reprend où on s'était arrêté)

25. (CM) Lesquels de ces formats de module sont historiquement reconnus par le navigateur sans outil supplémentaire ?

A. CommonJS (non, nécessite un bundler ou Node)

B. ESM (<script type="module">)

C. AMD (RequireJS ou loader)

D. IIFE (pas vraiment un "format", mais le navigateur comprend le script classique)

26. (CM) Les fonctions fléchées ont la particularité de :

A. Ne pas avoir de this propre (héritage lexical)

B. Ne pas avoir de arguments

C. Pouvoir être appelées via new pour faire des instances

D. Être toujours asynchrones

27. (CM) Dans le DOM, pour créer et insérer un nouveau <li> dans une <ul>, on peut :

A. document.createElement("li")

B. ul.appendChild(li)

C. ul.innerHTML += "
" (possible, mais réécrit tout l'intérieur)

D. ul.addChild("li") (n'existe pas)

28. (CM) Lesquels de ces éléments sont typiquement "falsy" en JavaScript ?

A. 0

B. NaN

C. "" (chaîne vide)

D. null

29. (CM) Dans un événement DOM, event.target et event.currentTarget peuvent différer quand :

A. On a un mécanisme de bouillonnement (un enfant a déclenché l'événement, le parent l'a écouté)

B. L'élément courant n'est pas l'initiateur de l'événement

C. event.target est toujours identique à event.currentTarget

D. event.stopPropagation() a été appelé

30. (CM) Lesquels de ces prototypes sont natifs en JavaScript ?

A. Object.prototype

B. Array.prototype

C. String.prototype

D. Number.prototype

31. (CM) Lesquels de ces import ESM sont corrects ?

A. import defaultExport from "module";

B. import { namedExport } from "module";

C. import * as bundle from "module";

D. import default as named from "module"; (c'est import defaultExport, { named } from "module";)

32. (CM) Dans un générateur, yield* someIterable signifie :

A. On délègue l'itération à someIterable

B. On attend que someIterable émette une promesse

C. On renvoie un tableau

D. On fusionne deux générateurs en un

33. (CM) Dans un contexte Node.js (CommonJS) :

A. require.cache stocke les modules déjà chargés

B. require.resolve("module") donne le chemin résolu du module

C. module.exports = ... définit l'export principal

D. import { ... } from "module"; est la syntaxe par défaut (non, c'est ES Module)

34. (CM) Pour transformer une liste d'arguments en tableau dans une fonction classique, on peut :

A. Array.prototype.slice.call(arguments)

B. Array.from(arguments)

C. [...arguments]

D. arguments.toArray() (n'existe pas)

35. (CM) Lesquelles de ces syntaxes peuvent créer une fonction auto-invoquée ?

A. (function() { ... })();

B. (function named() { ... }());

C. +function() { ... }();

D. ()=>{ ... }(); (besoin de parenthèses (() => {...})();)

36. (CM) Lesquelles de ces opérations DOM sont possibles ?

A. parentNode.removeChild(childNode);

B. element.appendChild(newChild);

C. element.cloneNode(true);

D. element.unappendChild(child) (n'existe pas)

37. (CM) Lesquels de ces arguments la fonction callback de reduce() peut-elle recevoir ?

A. accumulateur

B. valeurCourante

C. index

D. tableau

38. (CM) Dans une fonction fléchée, on peut faire :

A. (a, b) => a + b

B. () => { const x = 1; return x; }

C. (a = 0) => a * 2

D. (a, b, ...rest) => rest.length

39. (CM) Lesquels de ces objets sont disponibles dans un script ESM côté navigateur ?

A. window

B. document

C. process (côté Node.js, pas navigateur)

D. console

40. (CM) Quels sont des exemples d'événements DOM liés au clavier ?

A. keydown

B. keyup

C. keypress (moins utilisé, mais existe)

D. focus (lié au focus, pas au clavier)

41. (CM) Dans le DOM, pour cibler un élément par son id, on peut :

A. document.getElementById("id")

B. document.querySelector("#id")

C. document.querySelectorAll("#id") (renvoie un NodeList, potentiellement d'un seul élément)

D. element.getId("id") (n'existe pas)

42. (CM) Lesquels de ces objets supportent la syntaxe d'itération for…of ?

A. Un Array

B. Un Map

C. Un NodeList (depuis ES6, NodeList est itérable dans la plupart des navigateurs modernes)

D. Un objet littéral { a: 1, b: 2 } (non, pas itérable par défaut)

43. (CM) Dans un générateur, on peut capturer une exception lancée par yield via :

A. Un try…catch englobant

B. Le paramètre next(error) (pas standard)

C. La méthode gen.throw(new Error()) depuis l'extérieur

D. L'utilisation d'un bloc finally

44. (CM) Lesquels de ces événements se produisent lors du cycle de vie d'une page Web côté navigateur ?

A. DOMContentLoaded (DOM prêt)

B. load (tout est chargé, images, etc.)

C. unload ou beforeunload (fermeture de la page)

D. ready (non standard, sauf jQuery)

45. (CM) Lesquels de ces opérateurs sont introduits dans des versions récentes d'ECMAScript ?

A. ?? (nullish coalescing, ES2020)

B. ?. (optional chaining, ES2020)

C. ** (exponentiation, ES2016)

D. === (existe depuis ES1)

46. (CM) Lesquels de ces modes de chargement de script existent côté navigateur ?

A. <script src="…" defer>

B. <script src="…" async>

C. <script type="module">

D. <script module="true"> (invalide)

47. (CM) Lesquels de ces éléments font partie de l'objet location dans le DOM ?

A. location.href

B. location.reload()

C. location.pathname

D. location.document (n'existe pas)

48. (CM) En CommonJS, on peut faire :

A. exports.maFonction = …;

B. module.exports = { x: 1 };

C. import { x } from "module"; (non, c'est ESM)

D. require("module").quelqueChose

49. (CM) Lesquels de ces événements sont liés à la gestion de formulaires ?

A. submit (quand on valide le formulaire)

B. reset (quand on réinitialise le formulaire)

C. change (quand un champ change de valeur et perd le focus)

D. hover (pas un événement standard, c'est mouseover / mouseout)

50. (CM) Lesquelles de ces méthodes d'un tableau JS retournent une valeur unique plutôt qu'un nouveau tableau ?

A. reduce()

B. find()

C. includes()

D. map() (retourne un tableau)

51. (CM) Dans un générateur, on peut utiliser :

A. yield x

B. yield* autreGenerateur

C. next(yield 5) (pas la syntaxe habituelle, mais on peut faire let val = yield 5;)

D. async yield (n'existe pas tel quel)

52. (CM) Lesquels de ces objets existent en JavaScript Vanille (natif) ?

A. Math

B. JSON

C. window (dans navigateur)

D. fs (Node.js)

53. (CM) Lesquels de ces attributs sont valides sur un <script> en HTML ?

A. src="fichier.js"

B. type="module"

C. async ou defer

D. commonjs="true" (n'existe pas)

54. (CM) Lesquels de ces "events" DOM sont liés à la souris ?

A. click

B. dblclick

C. mousedown, mouseup

D. pointerdown, pointerup (nouvelles API pointer)

55. (CM) Dans le DOM, element.innerHTML = "..." :

A. Remplace tout le contenu HTML de l'élément

B. Supporte le code HTML pour l'affichage

C. Est dangereux si on met du HTML non maîtrisé (risque XSS)

D. Ne modifie pas le DOM visuellement

56. (CM) Lesquels de ces appels renvoient un objet Promise en JavaScript moderne ?

A. fetch("url")

B. async function x() {} (appeler x() retourne une promise)

C. new Promise((resolve, reject) => {})

D. setTimeout(fn, 1000) (ne renvoie pas de promise nativement)

57. (CM) Dans une fonction fléchée, (a => a * 2)(5) :

A. Renvoie 10

B. Est une IIFE fléchée

C. Retourne undefined

D. Nécessite des accolades

58. (CM) Lesquels de ces objets font partie de l'API DOM standard ?

A. Event

B. Element

C. Node

D. FileSystem (API non standard, dépend d'implémentations)

59. (CM) Dans un générateur, generator.next(x) permet :

A. De passer la valeur x au dernier yield suspendu

B. De renvoyer une promise

C. De générer un événement de type x

D. D'avancer l'itération d'un pas

60. (CM) Lesquels de ces modules sont pris en charge par Node.js sans config ?

A. CommonJS (require())

B. ESM (avec "type": "module" ou .mjs)

C. AMD

D. UMD

61. (CM) Lesquelles de ces méthodes de tableau ne retournent aucun nouveau tableau ?

A. forEach() (retourne undefined)

B. sort() (modifie sur place et retourne le tableau modifié)

C. push() (retourne la nouvelle longueur)

D. splice() (retourne les éléments supprimés)

62. (CM) Dans le DOM, element.parentElement :

A. Renvoie l'élément parent direct (ou null)

B. Renvoie toujours document

C. Est différent de element.parentNode si le parent est un nœud non-élément

D. Retourne une HTMLCollection

63. (CM) Dans une fonction async, on peut :

A. Utiliser le mot-clé await

B. Retourner directement une valeur, qui sera englobée dans une promesse résolue

C. Gérer les erreurs avec try...catch

D. Définir async sur n'importe quel bloc de code

64. (CM) Lesquels de ces éléments font partie du cycle de vie des événements DOM ?

A. Phase de capture

B. Phase de ciblage (target)

C. Phase de bouillonnement

D. Phase d'arrêt complet (pas un terme officiel)

65. (CM) Lesquelles de ces méthodes sont disponibles pour parcourir un NodeList ?

A. forEach() (depuis les specs modernes)

B. map() (non, pas sur un NodeList standard)

C. [...NodeList] (spread possible si l'environnement est compatible)

D. for (const node of NodeList) (itération possible)

66. (CM) En CommonJS, pour récupérer des exports nommés on fait :

A. const mod = require("./mod"); let val = mod.val;

B. const { val, func } = require("./mod"); (déstructuration de l'objet exports)

C. import { val } from "./mod"; (non, c'est ESM)

D. module.importAll("./mod")

67. (CM) Lesquels de ces objets JavaScript sont nativement itérables ?

A. String (chaque caractère)

B. Array

C. Map

D. Object (non, pas itérable par défaut)

68. (CM) Lesquels de ces événements se déclenchent sur un élément input lors d'interactions utilisateur ?

A. focus / blur

B. input (tapez au clavier)

C. change (perte de focus ou validation)

D. mouseenter (lié à la souris, pas le clavier)

69. (CM) Dans un script <script type="module">, on peut :

A. Importer des modules via import

B. Utiliser await en haut niveau (top-level await, dépend de la version de la spécification)

C. Appeler require() (non, c'est CommonJS)

D. Accéder à import.meta (informations sur le module en cours)

70. (CM) Lesquels de ces appels renvoient un tableau différent du tableau d'origine ?

A. arr.filter(...)

B. arr.map(...)

C. arr.slice(...)

D. arr.sort(...) (modifie en place)

71. (CM) Dans le DOM, quelles méthodes insèrent réellement un nœud dans la page ?

A. appendChild()

B. insertBefore()

C. replaceChild()

D. newChild = document.createElement("div") (crée seulement l'élément, ne l'insère pas)

72. (CM) Lesquels de ces éléments sont nécessaires pour propager un événement DOM jusqu'à un parent ?

A. Que l'événement soit "bubblable"

B. Que stopPropagation() ne soit pas appelé

C. Que preventDefault() ne soit pas appelé (n'arrête pas la propagation, juste l'action)

D. Que le parent écoute l'événement dans la phase de bouillonnement

73. (CM) Lesquels de ces mots-clés/fonctionnalités sont disponibles dans un générateur ?

A. yield

B. yield*

C. next() (méthode sur l'itérateur, pas un mot-clé dans la fonction)

D. async yield (inexistant)

74. (CM) Lesquels de ces frameworks/bibliothèques utilisent massivement JavaScript Vanille + DOM ?

A. jQuery (manipule le DOM)

B. React (DOM virtuel)

C. Vue.js (DOM virtuel, templates)

D. Angular (framework complet, DOM virtuel aussi, binding)

75. (CM) Lesquels de ces formats de module sont reconnus par Node.js (sans loader tiers) ?

A. CommonJS (CJS)

B. ESM (via "type": "module" ou .mjs)

C. AMD (non, nécessite RequireJS ou équivalent)

D. UMD (pas en natif)

76. (CM) Lesquelles de ces méthodes de table peuvent retourner un booléen ?

A. some()

B. every()

C. includes()

D. map() (retourne un tableau)

77. (CM) Lors d'un click sur un <button>, quel est l'ordre d'appel (si on a 3 écouteurs) ?

A. Par défaut, tous les écouteurs sur le même élément sont appelés dans l'ordre où ils ont été ajoutés

B. L'ordre peut être modifié en passant { once: true }

C. Le bouillonnement arrive après avoir exécuté les écouteurs du bouton lui-même

D. L'événement s'arrête si un des écouteurs appelle stopPropagation()

78. (CM) Lesquels de ces scénarios illustrent une IIFE ?

A. (function() { ... })();

B. (function named(){ ... })();

C. (async function(){ ... })(); (IIFE asynchrone possible)

D. function(){ ... }() (invalide seul, il faut mettre (function(){...})())

79. (CM) Dans un générateur, qu'est-ce qui différencie yield et yield* ?

A. yield émet une valeur simple

B. yield* délègue l'itération à un autre générateur ou itérable

C. yield* fusionne deux tableaux

D. yield attend la fin de l'autre générateur

80. (CM) Lesquels de ces événements DOM sont déclenchés quand on survole un élément ?

A. mouseover

B. mouseenter

C. mouseout (quand on quitte l'élément)

D. hover (pas un événement standard, c'est CSS)

81. (CM) Lesquels de ces attributs peuvent être modifiés via element.setAttribute() ?

A. id

B. class

C. value (pour un <input>)

D. style (ajoutera un style inline)

82. (CM) Lesquelles de ces syntaxes import ESM sont valides ?

A. import defaultExport, { namedA, namedB } from "module";

B. import * as mod from "module";

C. import { default } from "module"; (mauvaise syntaxe, c'est import defaultExport from "module")

D. import "module"; (importe et exécute, sans récupérer d'exports)

83. (CM) Lesquelles de ces méthodes de tableau modifient le tableau existant ?

A. splice()

B. sort()

C. reverse()

D. map() (crée un nouveau tableau)

84. (CM) Dans un générateur, generator.return("fin") :

A. Termine le générateur avec { value: "fin", done: true }

B. Continue l'itération si on appelle next() après

C. Provoque une exception

D. On peut le relancer en appelant generator.next() (non, c'est terminé)

85. (CM) Dans le DOM, quels éléments peuvent déclencher un événement focus ?

A. Les champs de formulaire (input, textarea, select)

B. Les éléments avec tabindex

C. Les non focusables par défaut (sauf si tabindex)

D. Les <img> (non focusables par défaut)

86. (CM) Lesquels de ces scripts "auto-invoqués" sont corrects ?

A. (function named() { ... }());

B. +function() { ... }();

C. void function(){ ... }();

D. () => { ... }(); (il faut (() => {...})();)

87. (CM) Dans le DOM, pour récupérer tous les <li> d'un <ul>, on peut :

A. document.querySelectorAll("ul li")

B. ul.getElementsByTagName("li")

C. ul.querySelectorAll("li")

D. ul.find("li") (n'existe pas en vanilla JS)

88. (CM) Lesquelles de ces fonctions fléchées retournent explicitement une valeur sans accolades ?

A. (x) => x * 2

B. x => { return x * 2 } (retour explicite, mais avec accolades)

C. () => ({ a: 1 }) (parenthèses pour l'objet littéral)

D. (x, y) => x + y

89. (CM) Lesquels de ces éléments font partie du "bons usages" du DOM en Vanilla JS ?

A. Sélectionner les éléments (document.querySelector)

B. Ajouter des écouteurs d'événements (element.addEventListener)

C. Manipuler le style (element.style ou classes)

D. Utiliser document.injectHTML() (n'existe pas)

90. (CM) En CommonJS, module.exports = class {} :

A. Exporte une classe anonyme comme export principal

B. Est équivalent à export default class {} en ESM

C. Permet const X = require("./fichier"); let instance = new X();

D. Génère une erreur de syntaxe

91. (CM) Lesquels de ces appels sur un objet Promise sont corrects ?

A. promise.then(onFulfilled)

B. promise.catch(onRejected)

C. promise.finally(onFinally)

D. promise.next() (c'est pour les générateurs, pas pour les promesses)

92. (CM) Dans un générateur, yield :

A. Suspend l'exécution jusqu'au prochain next()

B. Peut recevoir une valeur via next(value)

C. Peut être utilisé sans valeur : yield;

D. Détruit le générateur si on l'appelle trop de fois

93. (CM) Lesquels de ces attributs HTML peuvent influencer l'exécution d'un <script> ?

A. async

B. defer

C. type="module"

D. reload="auto" (n'existe pas)

94. (CM) Lesquels de ces événements sont liés au DOM lorsque la page est en cours de navigation ?

A. beforeunload (l'utilisateur s'apprête à quitter la page)

B. unload (la page est déchargée)

C. hashchange (le hash de l'URL change)

D. popstate (historique, back/forward)

95. (CM) Lesquels de ces opérateurs/éléments sont spécifiques à ES2015 ou + ?

A. => (arrow functions, ES6)

B. class (ES6)

C. import / export (ES6)

D. var (existe depuis ES1)

96. (CM) Array.prototype.every() et Array.prototype.some() :

A. Renvoient un booléen

B. Acceptent un callback (element, index, array)

C. Parcourent tout le tableau sauf si la condition est déjà déterminée (court-circuit)

D. Modifient le tableau en place

97. (CM) Dans le DOM, quelles sont des méthodes pour insérer du texte ou du HTML ?

A. element.textContent = "..." (texte brut)

B. element.innerHTML = "..." (html interprété)

C. element.insertAdjacentHTML("beforeend", "...")

D. element.value = "..." (uniquement pour champs de formulaire)

98. (CM) Lesquels de ces scripts asynchrones existent côté navigateur ?

A. <script src="file.js" async>

B. <script src="file.js" defer>

C. <script type="module"> (charge de façon asynchrone par défaut)

D. <script src="file.js" concurrency> (n'existe pas)

99. (CM) Dans un IIFE, on peut :

A. Déclarer des variables locales qui n'affectent pas le scope global

B. Retourner un objet en fin d'exécution

C. Se passer totalement de () => {} si on n'a pas besoin de scope isolé

D. Lancer la fonction immédiatement par ()() (il faut (function(){})())

100. (CM) Lesquelles de ces lignes de code déclenchent un événement de manière programmatique dans le DOM ?

A. element.click() (simule un clic)

B. element.dispatchEvent(new Event("click"))

C. element.fireEvent("onclick") (ancien IE)

D. element.trigger("click") (jQuery, pas vanilla JS)