Bing

Oui, voici un autre QCM de 5 questions sur les structures conditionnelles en JavaScript. Bonne chance!

1. Quel est le résultat de ce code?

let x = 10;

let y = 20;

let z = x > y ? "x est plus grand que y" : "x est plus petit que y";

console.log(z);

A) x est plus grand que y
B) x est plus petit que y
C) Erreur de type
D) Erreur de syntaxe

Réponse: B) x est plus petit que y

Feedback: Cette question teste votre compréhension de l'opérateur ternaire (?:), qui permet d'assigner une valeur à une variable selon une condition. Il faut bien connaître la syntaxe de cet opérateur, qui prend trois opérandes: une condition, une valeur si la condition est vraie, et une valeur si la condition est fausse. Ici, comme x est inférieur à y, l'opérateur ternaire assigne la valeur "x est plus petit que y" à la variable z, donc le code affiche "x est plus petit que y".

1. Quel est le résultat de ce code?

let a = 5;

let b = 10;

let c = a + b;

if (c === 10) {

console.log("c vaut 10");

} else if (c === 15) {

console.log("c vaut 15");

} else {

console.log("c vaut autre chose");

}

A) c vaut 10
B) c vaut 15
C) c vaut autre chose
D) Erreur de syntaxe

Réponse: B) c vaut 15

Feedback: Cette question teste votre compréhension de l'opérateur d'égalité stricte (===), qui compare la valeur et le type de deux expressions. Il faut bien distinguer cet opérateur de l'opérateur d'égalité (==), qui compare seulement la valeur des expressions. Ici, comme c vaut 15 et que c et 15 sont tous les deux de type nombre, l'opérateur === renvoie vrai, donc le code affiche "c vaut 15".

1. Quel est le résultat de ce code?

let i = 0;

for (let i = 0; i < 3; i++) {

console.log(i);

}

console.log(i);

A)
0
1
2
3

B)
0
1
2
0

C)
0
1
2
Erreur de référence

D) Erreur de syntaxe

Réponse: B)
0
1
2
0

Feedback: Cette question teste votre compréhension de la portée des variables déclarées avec le mot-clé let, qui sont limitées au bloc dans lequel elles sont définies. Il faut bien distinguer les variables locales et globales, et faire attention aux conflits de noms. Ici, il y a deux variables i, une globale et une locale à la boucle for. La variable locale s'incrémente jusqu'à être inférieure à 3, donc la boucle affiche les nombres de 0 à 2. La variable globale reste inchangée, donc le code affiche aussi 0 après la boucle.

1. Quel est le résultat de ce code?

let j = 0;

while (j < 3) {

console.log(j);

j++;

}

console.log(j);

A)
0
1
2

B)
0
1
2
3

C)
1
2
3

D) Erreur de boucle infinie

Réponse: B)
0
1
2
3

Feedback: Cette question teste votre compréhension de la boucle while, qui permet de répéter un bloc de code tant qu'une condition est vraie. Il faut bien suivre l'évolution de la variable j et le nombre d'itérations de la boucle. Ici, comme j commence à 0 et s'incrémente à chaque tour, la boucle s'arrête quand j vaut 3, donc le code affiche les nombres de 0 à 2 dans la boucle, puis 3 après la boucle.

1. Quel est le résultat de ce code?

let k = 0;

do {

console.log(k);

} while (k++ < 3);

A)
0
1
2

B)
0
1
2
3

C)
1
2
3

D)
1
2
3
4

Réponse: B)
0
1
2
3

Feedback: Cette question teste votre compréhension de la boucle do...while, qui permet de répéter un bloc de code au moins une fois, puis tant qu'une condition est vraie. Il faut aussi connaître la différence entre l'opérateur d'incrémentation préfixé (++k) et postfixé (k++), qui modifie la valeur de la variable avant ou après son évaluation. Ici, comme k commence à 0 et s'incrémente après d'être comparé à 3, la boucle s'arrête quand k vaut 4, donc le code affiche les nombres de 0 à 3.