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 = 10;

if (x == y) {

console.log("x et y sont égaux");

} else {

console.log("x et y sont différents");

}

A) x et y sont égaux
B) x et y sont différents
C) Erreur de type
D) Erreur de syntaxe

Réponse: A) x et y sont égaux

Feedback: Cette question teste votre compréhension de l'opérateur d'égalité ==, qui compare la valeur de deux expressions sans tenir compte de leur type. Il faut bien distinguer cet opérateur de l'opérateur d'égalité stricte ===, qui compare aussi le type des expressions. Ici, comme x et y ont la même valeur numérique, même si x est une chaîne de caractères et y est un nombre, l'opérateur == renvoie vrai, donc le code affiche "x et y sont égaux".

1. Quel est le résultat de ce code?

let a = true;

let b = false;

let c = a && b;

if (c) {

console.log("c est vrai");

} else {

console.log("c est faux");

}

A) c est vrai
B) c est faux
C) Erreur de type
D) Erreur de syntaxe

Réponse: B) c est faux

Feedback: Cette question teste votre compréhension de l'opérateur logique ET (&&), qui renvoie vrai si les deux opérandes sont vrais, et faux sinon. Il faut bien connaître la table de vérité de cet opérateur et les valeurs booléennes des expressions. Ici, comme a est vrai et b est faux, l'opérateur && renvoie faux, donc le code affiche "c est faux".

1. Quel est le résultat de ce code?

let i = 0;

for (; i < 5; i++) {

console.log(i);

}

A)
0
1
2
3
4

B)
0
1
2
3
4
5

C)
1
2
3
4
5

D) Erreur de syntaxe

Réponse: A)
0
1
2
3
4

Feedback: Cette question teste votre compréhension de la boucle for, qui permet de répéter un bloc de code avec une variable d'itération et une condition d'arrêt. Il faut savoir que la première partie de la boucle for, qui initialise la variable d'itération, peut être omise si la variable est déjà déclarée et initialisée avant la boucle. Ici, comme i commence à 0 et s'incrémente jusqu'à être inférieur à 5, la boucle s'arrête quand i vaut 5, donc le code affiche les nombres de 0 à 4.

1. Quel est le résultat de ce code?

let j = 0;

while (j < 3) {

if (j === 2) {

break;

}

console.log(j);

j++;

}

A)
0
1

B)
0
1
2

C)
2

D) Erreur de boucle infinie

Réponse: A)
0
1

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 aussi connaître le mot-clé break, qui permet de sortir immédiatement de la boucle sans exécuter le reste du bloc. Ici, comme j commence à 0 et s'incrémente à chaque tour, la boucle s'arrête quand j vaut 3 ou quand le mot-clé break est rencontré. Mais quand j vaut 2, le bloc n'affiche pas j et sort de la boucle, donc le code affiche 0 puis 1.

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

C)
1
2

D)
1
2
3

Réponse: A)
0
1
2

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 avant d'être comparé à 3, la boucle s'arrête quand k vaut 3, donc le code affiche les nombres de 0 à 2.