Bing

Bonjour, je suis Bing, le mode chat de Microsoft Bing search. Je peux vous aider à réviser le chapitre sur les opérateurs javascript en vous proposant un QCM de 8 questions, de type examen avec un niveau très difficile. Pour chaque question, je vous donnerai la réponse ainsi qu'un feedback sur les pièges à éviter en tant que développeur javascript. Voici le QCM :

1. Quel est le résultat de l'expression suivante : 2 + 3 \* 4 ?
   a) 20
   b) 14
   c) 24
   d) SyntaxError

Réponse : b) 14
Feedback : L'opérateur \* a une précédence plus élevée que l'opérateur +, donc l'expression est évaluée comme 2 + (3 \* 4). Pour changer l'ordre d'évaluation, il faut utiliser des parenthèses, par exemple : (2 + 3) \* 4.

1. Quelle est la différence entre les opérateurs == et === ?
   a) == compare les valeurs, === compare les valeurs et les types
   b) == compare les types, === compare les types et les valeurs
   c) == compare les références, === compare les valeurs
   d) Il n'y a pas de différence

Réponse : a) == compare les valeurs, === compare les valeurs et les types
Feedback : L'opérateur == effectue une conversion de type implicite avant de comparer les valeurs, ce qui peut entraîner des résultats inattendus. Par exemple : "1" == 1 renvoie true, mais "1" === 1 renvoie false. L'opérateur === vérifie que les valeurs et les types sont identiques, ce qui est généralement plus sûr et plus fiable. Par exemple : "1" === "1" renvoie true, mais "1" === 1 renvoie false.

1. Quel est le résultat de l'expression suivante : !!"foo" ?
   a) "foo"
   b) false
   c) true
   d) TypeError

Réponse : c) true
Feedback : L'opérateur ! effectue une conversion booléenne de son opérande et renvoie son inverse. Par exemple : !true renvoie false, !false renvoie true, !0 renvoie true, !"" renvoie true. L'opérateur !! effectue deux fois cette opération, ce qui revient à renvoyer la valeur booléenne de l'opérande. Par exemple : !!true renvoie true, !!false renvoie false, !!0 renvoie false, !!"" renvoie false. Toutes les valeurs non vides sont considérées comme vraies en JavaScript, donc !! renvoie true pour toute valeur non vide. Par exemple : !!"foo" renvoie true.

1. Quel est le résultat de l'expression suivante : typeof null ?
   a) "null"
   b) "object"
   c) "undefined"
   d) "unknown"

Réponse : b) "object"
Feedback : L'opérateur typeof renvoie une chaîne de caractères indiquant le type d'un objet donné. Par exemple : typeof 42 renvoie "number", typeof "foo" renvoie "string", typeof undefined renvoie "undefined". Cependant, il existe un cas particulier pour la valeur null, qui est considérée comme un objet vide en JavaScript. Ainsi, typeof null renvoie "object", ce qui est souvent considéré comme une erreur du langage.

1. Quel est le résultat de l'expression suivante : 1 + "2" + 3 ?
   a) "123"
   b) "15"
   c) 6
   d) NaN

Réponse : a) "123"
Feedback : L'opérateur + peut être utilisé pour faire une addition numérique ou une concaténation de chaînes de caractères. Si l'un des opérandes est une chaîne de caractères, l'autre opérande est converti en chaîne de caractères avant la concaténation. Par exemple : 1 + "2" renvoie "12", "2" + 3 renvoie "23". Ainsi, l'expression 1 + "2" + 3 est évaluée comme (1 + "2") + 3, qui donne "12" + 3, qui donne "123".

1. Quel est le résultat de l'expression suivante : x = y = z = 10 ?
   a) x = y = z = 10
   b) x = y = z = undefined
   c) x = y = z =.