**Exercice JAVASCRIPT 3 les boucles**

**Algorithme à implémenter**

À implémenter étape par étape:

1. Écrire un programme qui affiche les nombres de 1 à 199 (compris) dans la console.
2. Pour les multiples de 3, afficher Fizz au lieu du nombre.
3. Pour les multiples de 5, afficher Buzz au lieu du nombre.
4. Pour les nombres multiples de 3 et 5, afficher uniquement FizzBuzz.

Pour ne pas s’embrouiller, il est recommandé de:

* commencer par écrire la logique d’une seule itération, avant de la faire se répéter à l’aide d’une boucle;
* écrire sous forme de pseudo-code (en langue française), avant de l’implémenter en JavaScript.

**Trace d’exécution de l’algorithme**

On devrait obtenir les lignes suivantes dans la console :

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

… et ainsi de suite, jusqu’à 199.

**Comment savoir si un nombre est multiple d’un autre ?**

Pour savoir si un nombre est multiple de 3 et/ou de 5, nous allons utiliser deux fonctions fournies ci-dessous :

function estMultipleDeTrois(nombre) {

return nombre % 3 === 0;

}

function estMultipleDeCinq(nombre) {

return nombre % 5 === 0;

}

Après avoir copié-collé la définition de ces deux fonctions dans la console JavaScript, vous pouvez les utiliser de la manière suivante :

estMultipleDeTrois(2); // => retourne false, car 2 n'est pas multiple de 3

estMultipleDeTrois(6); // => retourne true, car 6 est un multiple de 3

estMultipleDeCinq(6); // => retourne false, car 6 n'est pas un multiple de 5

estMultipleDeCinq(15); // => retourne true, car 15 est un multiple de 5

Vous pouvez alors appeler ces fonctions dans les parenthèses de vos conditions if, car, comme une expression de comparaison de valeurs, un appel de fonction retourne une valeur qui est vraie (true) ou fausse (false).

Exemple :

var monNombre = 5; // valeur fournie en guise d'exemple

if (estMultipleDeCinq(monNombre)) {

console.log('monNombre est multiple de 5');

} else {

console.log('monNombre n\'est pas multiple de 5');

}

Pour les plus téméraire voici un exercice supplémentaire :

**Fonctionnement du jeu à implémenter**

* En début de partie, l’ordinateur va choisir un nombre aléatoire entre 0 et 100.
* Le joueur a droit à 10 tentatives pour deviner ce nombre.
* À chaque tentative :
  + Si le nombre saisi est inférieur à celui de l’ordinateur, afficher Plus grand.
  + Si le nombre saisi est supérieur à celui de l’ordinateur, afficher Plus petit.
  + Si le joueur a réussi à deviner le bon nombre, afficher Bravo !.
* La partie continue jusqu’à ce que le joueur gagne ou épuise ses 10 tentatives.

Pour implémenter ce jeu :

* vous allez avoir besoin d’interagir avec l’utilisateur, et donc d’utiliser les mots-clés prompt et alert;
* il est recommandé de commencer par écrire le code d’une seule itération (sans utiliser de boucle), et de le tester en donnant des valeurs arbitraires à vos variables, pour simuler chaque cas.

**Exemple de déroulement d’une partie**

Nombre saisi par le joueur : 10

Réponse de l'ordinateur : Plus grand

Nombre saisi par le joueur : 20

Réponse de l'ordinateur : Plus petit

Nombre saisi par le joueur : 16

Réponse de l'ordinateur : Bravo !

**Comment obtenir un nombre aléatoire**

Pour obtenir un nombre aléatoire entier entre 0 et 100 :

Math.round(Math.random() \* 100)

Libre à vous de stocker cette valeur dans une variable, si vous avez besoin de la comparer à d’autres valeurs, par exemple.