

Exercices 1 : Javascript – Correction

1°) Echange valeurs de 2 variables

```
exercice1.js > ...
1  let A = 5;
2  let B = 10;
3  let temp;
4  console.log("-----Avant inversion-----");
5  console.log("A : " + A);
6  console.log("B : " + B);
7  temp = A;
8  A = B;
9  B = temp;
10 console.log("-----Après inversion-----");
11 console.log("A : " + A);
12 console.log("B : " + B);
13
14
15
```

2°) Réaliser l’affichage : A partir des variables

```
JS exercice2.js > ...
1  let nom = "Toto";
2  let age = 30;
3  let homme = true;
4
5  console.log("Nom : " + nom);
6  console.log(`Age : ${age}`);
7  if(homme){
8      console.log("Sexe : Homme");
9  }else{
10     console.log("Sexe : Femme");
11 }
```

3°) Réaliser un programme permettant à l'utilisateur de saisir la puissance de 2 qu'il souhaite afficher

```
Exercice1 > JS exercice3.js > ...
1 let readline = require("readline-sync");
2 const PUISSANCE = 2;
3 let saisie = parseInt(readline.question("Veuillez saisir la puissance "));
4 console.log(PUISSANCE + " à la puissance " + saisie + " = " + Math.pow(PUISSANCE, saisie));
```

4°) Vous devez réaliser un programme permettant de saisir le nom, le prénom et l'âge d'un utilisateur

```
Exercice1 > JS exercice4.js > ...
1 let readline = require("readline-sync");
2 let nom = readline.question("Quel est votre nom : ");
3 let prenom = readline.question("Quel est votre prenom : ");
4 let age = parseInt(readline.question("Quel est votre age : "));
5
6 let txt = "-----\n";
7 txt += "-----BIENVENUE-----\n";
8 txt += "-----\n";
9 txt += "Votre nom est : " + nom + "\n";
10 txt += "Votre prenom est : " + prenom + "\n";
11 txt += "Votre age est : " + age + "\n";
12 console.log(txt);
```

5°) Afficher une table de multiplication

```
Exercice1 > JS exercice5.js > ...
1 let readline = require("readline-sync");
2 let saisie = parseInt(readline.question("Quelle table de multiplication voulez-vous afficher : "));
3
4 for(let i=1; i<=10;i++){
5     let resultat = saisie * i;
6     console.log(saisie + " * " + i + " = " + resultat);
7 }
```

6°) Saisie d'un chiffre et vérification de sa valeur

```
Exercice1 > JS exercice6.js > ...
1  let readline = require("readline-sync");
2  // let saisie = parseInt(readline.question("Quel nombre voulez-vous saisir : "));
3  let saisie = readline.question("Quel nombre voulez-vous saisir : ");
4
5  let chiffreSaisie = false;
6
7  while(!chiffreSaisie){
8      if(isNaN(saisie)){
9          console.log("Veuillez recommencer la saisie, vous devez saisir un nombre :");
10         let saisie = readline.question("Quel nombre voulez-vous saisir : ");
11     }else{
12         console.log("C'est un chiffre");
13         chiffreSaisie = true;
14     }
15 }
16
17 console.log("Le chiffre saisi est : " + saisie);
18
19
```

7°) Fonction qui vérifie si un chiffre est divisible par 3

Pour savoir si un chiffre est divisible par 3, il faut regarder le reste de la division euclidienne. Si celui est égal à 0, alors le chiffre est bien divisible par 3. Pour cela, il existe l'opérateur modulo que l'on peut utiliser grâce au signe %.

```
1  let nb1 = 123;
2  let nb2 = 20;
3
4  afficherResultatDivision(nb1);
5  afficherResultatDivision(nb2);
6
7
8  function afficherResultatDivision(nombre){
9      if(verificationDivision){
10         console.log("Le résultat de " + nombre + " / 3 = " + (nombre/3));
11     }else{
12         console.log("Le nombre n'est pas divisible par 3");
13     }
14 }
15
16 function verificationDivision(nombre){
17     return nombre % 3 === 0;
18 }
```

8°) Vérifier si un mot contient moins de 8 caractères

Exercice1 > JS exercice8.js > ...

```
1 //fonction qui permet d'afficher la taille d'un mot
2 function afficherTaille(mot){
3
4     if(mot.length < 8){
5         console.log(`Le mot contient ${mot.length} caractères`);
6         // console.log("Le mot contient " + mot.length + " caractères");
7     }else{
8         console.log("Le mot fait plus de 8 caractères");
9     }
10 }
11
12 let mot1 = "toto";
13 let mot2 = "heureusement";
14
15 //Appel de la fonction
16 afficherTaille(mot1);
17 afficherTaille(mot2);
18 |
```