

Exercices

JavaScript

Paul Bourgeois

Rappels sur les variables et opérations

Déclarations :

```
const myVariable = 2;    // variable initialisée, sa valeur ne peut être changée
let otherVariable = 2;    // pas forcément initialisée et valeur modifiable

var variable = 2;        // ancienne variable JS. Similaire à let mais portée différente
```

Types de données :

Types simples : Number, String, Boolean

Types composés : Array, Object

Opérations :

Sur les Numbers : + - * / %

Sur les Strings : "Hello " + "everyone" `Hello \${stringWorld}`

Sur les Booleans : && (and) ; || (or) ; ! (not)

Exercices 1

1) Inverser les valeurs de deux variables

2) Saisir des informations utilisateurs (nom, prénom, âge) et afficher les informations saisies

```
aide : let price = prompt("How much does it cost ?");  
      document.write("something");
```

3) Remplacer l'âge par l'année de naissance et calculer l'âge de l'utilisateur

4) Écrire un programme qui demande le Hors Taxes (HT) d'un article et le nombre d'articles. Fournir le prix Toutes Taxes Comprises (TTC) en y appliquant une Taxe sur la Valeur Ajoutée (TVA) de 20%

```
aide : calcule de 20% de 40 => (40 / 100) * 20 = 8
```

5) Écrire de plusieurs façons différentes une même phrase. Pensez aux concaténations de chaînes de caractères, appel de variables, ...

Rappels sur les boucles et conditions

Les Boucles

For :

- `for (let i = 0 ; i < 10 ; i++) {}`

While :

- `while(condition is true) {}`
- différentes conditions : `< ; <= ; == ; >= ; > ; != ; ===`
- opérateurs logiques : `&& || !`
- attention aux boucles infinies

Les instructions conditionnelles :

If, else if et else :

```
if (myBoolean est vrai) {  
    //réaction à la valeur vrai de myBoolean  
} else if (myBoolean2 est vrai) {  
    //réaction à la valeur vrai de myBoolean  
} else {  
    //réaction à la valeur fausse de myBoolean et myBoolean2  
}
```

Switch :

```
switch(trafficLightColor) {  
case 'green':  
    console.log("Let's go");  
    break;  
case 'red':  
    console.log("Stop right now !");  
    break;  
default:  
    console.log("Gotta be fast");  
}
```


La portée des variables

Une variable n'existe que dans le bloc d'instructions où elle est déclarée, sauf pour une déclaration avec **var** !!

```
if(true) {  
    var myVariable = 2;  
    let myVariable2 = 2;  
    const myVariable3 = 2;  
}  
alert(myVariable); //2  
alert(myVariable2); //erreur  
alert(myVariable3); //erreur
```

Exercices 2

- 1) En fonction de l'âge (exercices précédents), afficher si la personne est majeur ou non.
- 2) Écrire un algorithme qui demande 3 noms à l'utilisateur et informe ensuite s'ils sont rangés dans l'ordre alphabétique.
- 3) Écrire un algorithme qui demande une heure (heures minutes secondes)) un utilisateur et affiche l'heure qu'il sera 1 seconde plus tard. Vérifier la validité des données saisies.
- 4) Écrire un algorithme qui demande un nombre compris entre 1900 et 2021 jusqu'à ce que la réponse convienne.
- 5) Écrire un jeu qui fasse deviner un nombre.
- 6) Écrire un algorithme qui calcule la factorielle d'un nombre. (demander à Paul l'explication de la factorielle si besoin)

Structures de données

Tableau

C'est une structure de données ordonnées utilisant une association clé/valeur et dont l'indice de départ est 0.

```
let myArray = new Array();
let myArray2 = [];
let ohterArray = [23, "bonjour", true];
```

Propriétés et méthodes d'un tableau:

```
let length = array.length;
array.push("last element");      // ajoute un nouvel élément à la fin du tableau
array.unshift("first element");  // ajoute un nouvel élément au début du tableau
array.pop()                      // retire et renvoie le dernier élément du tableau
array.shift()                    // retire et renvoie le premier élément du tableau
```

Liste des propriétés et méthodes des tableaux

Parcourir un tableau

- A l'aide d'un [itérateur](#)
- Par clé : for (let i in array){}
- Par valeur : for(let value of array){}

Accéder à une valeur d'un tableau :

```
let value = array[2]; //value prendra la valeur de 3ème valeur du tableau
```

Objet

C'est une structure de données non ordonnées qui contient des propriétés et des méthodes.


```
let object = {  
  prop1: "test";  
  prop2: 23;  
  methode: myFunction  
}
```

Il existe des objets natifs en JS comme Maths et DOM

On déclare un objet de plusieurs manières :

```
let obj = new Object();  
let object = {};  
let myObject = {  
  prop1: true,  
  prop2: "awesome"  
}
```

On peut accéder aux propriétés d'un objet de 2 manières :

```
let value = obj.prop;  
let value2 = obj["prop2"];
```

On peut instancier des propriétés des mêmes façons :

```
myObject.prop = false;  
myObject["prop2"] = "test";
```

Exercices 3

- 1) Écrire un algorithme qui indique si un tableau contient une certaine valeur.
- 2) Écrire un algorithme qui détermine si un mot est un palindrome (qui se lit dans les deux sens : solos, été, kayak ...).
- 3) Écrire un algorithme qui trouve le plus petit nombre d'une liste.
- 4) Écrire un algorithme qui trie une liste de nombre par ordre croissant.

Les fonctions

Déclaration d'une fonction :

```
function square(number) {  
    return number * number;  
}
```

Appel d'une fonction :

```
let squaredNumber = square(4); //squaredNumber = 16
```

Les paramètres de type simple sont passés par valeur (number, string, boolean) alors que les objets et tableaux sont passés par références.

Les variables déclarées en dehors d'une fonction sont accessibles dans une fonction. Alors que les paramètres déclarés dans la fonction sont inaccessibles à l'extérieur de cette fonction.

Exercices 4

- 1) Mettre l'algorithme de trie et de recherche du minimum sous la forme de fonctions et tester ces fonctions.
- 2) Mettre sous la forme d'une fonction récursive la fonction qui teste les palindromes.
- 3) Ecrire une fonction récursive qui calcule une factorielle;
- 4) Écrire une fonction récursive qui calcule la suite de Fibonacci jusqu'à un élément n.
(demander à Paul pour des explications sur la suite de Fibonacci)