TP Informatique – le langage Javascript ES6

Exercice 0 : mise en route et révisions

- ✓ Utiliser « Visual Basic Editor »
- ✓ Créer un squelette de document HTML *[raccourci html:5]* dans le fichier exo0.html
- ✓ Créer un fichier exo0.js et le lier au fichier exo0.html
- ✓ Déclarer un tableau contenant plusieurs nombres entiers
- ✓ Écrire 3 fonctions :
  - sommeT : Calcul de la somme des nombres
  - moyenneT : Calcul de la moyenne
  - minimumT : Recherche de la plus petite valeur

  → chaque fonction doit avoir 3 commentaires au format JSDoc *[raccourci /**]* (proche de la référence JavaDoc)

```
/**
 * role de la fonction
 * @param t signification de chaque paramètre
 * @return signification valeur de retour
 */
```

- ✓ Ecrire une deuxième version en utilisant la boucle *for…of*

Exercice 1 : les fonctions et fonctions flèches : les nombres particuliers

- ✓ Écrire la fonction « somDivPropre » qui calcule la somme des diviseurs propres d'un nombre entier n (les diviseurs sauf lui-même) : *attention le résultat n'est défini que pour n >=1*
- ✓ Écrire la fonction « premier » *(sous la forme d'une fonction flèche)* qui indique si un nombre n est premier -> il n'a que deux diviseurs 1 et lui-même
- ✓ Écrire la fonction « parfait » *(sous la forme d'une fonction flèche)* qui indique si un nombre n est parfait -> n est égal à la somme de tous ses diviseurs propres
- ✓ Vérifier que dans l'intervalle [2,1000] seuls 6, 28 et 496 sont parfaits
- ✓ Écrire la fonction « sublime » *(sous la forme d'une fonction flèche)* qui si un nombre n est sublime -> le nbre de diviseurs de n et la somme des diviseurs de n sont 2 nombres parfaits.
- ✓ Vérifier que dans l'intervalle [2,1000] seul 12 est sublime

Exercice 2 : les tableaux : les pays

!! attention : vous ne devez utiliser aucune boucle !!

- ✓ Déclarer ces 3 tableaux :
  let liste1 = ['Suisse', 'Italie', 'Espagne', 'Portugal'];
  let liste2 = ['Suède', 'Danemark', 'Pays-Bas' ,'Belgique'];
  let tPays = [] ;
- ✓ Initialiser tPays avec la liste1
  puis afficher tPays et le nb d'éléments de tPays (en utilisant l'interpolation de chaines)
- ✓ Ajouter la « Grèce » à la fin de tPays et afficher tPays et le nb d'éléments
- ✓ Supprimer le dernier pays et afficher tPays et le nb d'éléments
- ✓ Ajouter la « Grèce » et la « France » en début de tableau ; afficher tPays et le nb d'éléments
- ✓ Ajouter les pays de liste2 ; afficher tPays et le nb d'éléments

- ✓ Rechercher un pays dans tPays *(d'abord la France puis l'Espagne)*
  Utiliser l'opérateur ternaire pour afficher le message 'trouvé' ou 'pas trouvé'
- ✓ Trier la liste tPays dans l'ordre alphabétique
- ✓ Trier la liste tPays dans l'ordre alpha inverse *en utilisant une fonction flèche*
- ✓ Trier sur la longueur des noms des Pays *en utilisant une fonction flèche*
- ✓ Créer un nouveau tableau contenant les longueurs de chacun des pays *en utilisant une fonction flèche*
- ✓ Construire un nouveau tableau contenant uniquement les pays commençant par P ou F *en utilisant une fonction flèche*
- ✓ Calculer la somme des longueurs des chaines de la liste *en utilisant une fonction flèche*
- ✓ Mettre tous les pays en majuscules *en utilisant une fonction flèche*

- ✓ Copie de tableaux ? comprendre la différence entre ces 2 instructions :

```
let liste3 = liste1
let liste4 = [...liste1]
```

Afficher le nb d'éléments et la liste des éléments de liste1, liste3 et liste4 (tjs sans boucle et avec interpolation de textes)
Ajouter 'Grèce' dans liste1
Afficher à nouveau le nb d'éléments et la liste des éléments de liste1, liste3 et liste4
Quelle différence entre les 3 listes ?
Et donc que fait l'instruction « liste3 = liste1 »
Et aussi l'instruction « liste4 = [...liste1]
→conclusion : comment faut-il faire pour faire une copie d'un tableau ?

Exercice 3 : les classes : liste de choses à faire

!! à faire plus tard dans l'année après les enseignements de POO !!

- ✓ Créer une classe correspondant à une chose à faire caractérisée par un texte et un indicateur permettant de savoir si la chose est à faire ou si elle a déjà été faite.
- ✓ Le constructeur doit permettre d'initialiser le texte et l'indicateur ; si celui-ci n'est pas fourni il sera initialisé à faux par défaut
- ✓ La classe doit comporter un getter pour le texte et un setter pour le texte et l'indicateur
- ✓ La classe doit comporter une méthode pour afficher la chose à faire sous la forme
  ==> chose (à faire) ou ==> chose (faite)
  →utiliser l'interpolation de chaines et l'opérateur ternaire
- ✓ Créer une classe permettant de gérer une liste de choses à faire
  Le constructeur doit permettre d'initialiser la liste à partir de plusieurs chaines de caractères (les textes des choses à faire).
  Il doit y avoir des méthodes :
  - ajouter une chose à partir de son texte
  - afficher la liste des choses à faire
  - supprimer une chose à faire à partir de son texte
  - faire une chose
- ✓ Exemple de pgm de test :

```
let maliste = new Liste("menage","courses");
maliste.afficher();
maliste.ajouterChose("velo");
maliste.afficher();
maliste.faire("menage");
maliste.afficher();
maliste.supprimerChose("courses");
maliste.afficher();
```