**Important!** Pour que vos fonctions soit appelées, vous devez modifier le fichier « Lab3\_boucles\_&\_fonctions.html » afin de mettre le nom de votre fonction dans le « onclick » des buttons. Ex : <button onclick="maFonction()">Exercice 1</button>

Exercice 1

Grâce à une boucle, écrivez la fonction JavaScript qui permet d’afficher la table de multiplications d’une valeur saisie par l’utilisateur.

Conseil : Pensez à la matière vue lors du dernier laboratoire pour avoir la saisie de l’usager !

Conseil : Utilisez les fonctions **ajouterContenu** et **viderContenu** déjà présente dans le fichier **.js** afin d’écrire dans ou de vider le div « **resultat** ».

Exercice 2

Créer les fonctions JavaScript nécessaires afin de réaliser la calculatrice suivante. Il doit y avoir une fonction principale et une par opérateur. La fonction principale doit :

* Demander trois entrées à l’utilisateur, soit **2** chiffres et **1** opérateur ;
* Les opérateurs possibles sont : +, -, \* et / ;
* Afficher le résultat obtenu ;
* Demander si l’utilisateur veut continuer. Si oui, la fonction doit continuer et pouvoir prendre d’autres entrées ; Faire 3 choix. Recommencer un autre calcul, Continuer le calcul actuel, Terminer. Au choix de la personne.
* S’exécuter **AU MOINS** une fois. Si la personne choisit l’option recommencer un autre calcul, appeler la fonction principale à nouveau.

Autres fonctions sont appelées à partir d’un switch sur l’opérateur.

Les autres fonctions doivent :

* Prendre en paramètre deux variables ;
* Valider que les deux variables sont des nombres ;
* Retourner le résultat de l’opération.

Exercice 3

Lorsqu’on utilise un tableau, il est possible d’utiliser les fonctions **map** et **filter** qui prennent chacune une fonction anonyme en paramètre. Écrivez une fonction JavaScript utilisant **correctement** map et filter.

Conseil : Faite des recherches sur internet pour voir comment s’utilise les fonctions **map** et **filter**.

Exercice 4

En utilisant la valeur entrée par l’usager pour déterminer le nombre de lignes, générez un triangle d’étoiles comme suit :

\* 1ère ligne

\*\* 2ième ligne

\*\*\* 3ième ligne

\*\*\*\* 4ième ligne

Etc.

Conseil : Utilisez les fonctions **ajouterContenu** et **viderContenu** déjà présente dans le fichier **.js** afin d’écrire dans ou de vider le div « **resultat** ».

Défi

En vous basant sur l’exercice précédent, êtes-vous capable de générer d’autres triangles ?

\* \*\*\*\* \*\*\*\*\*\*\*

\*\*\* \*\*\* \*\*\*\*\*

\*\*\*\*\* \*\* \*\*\*

\*\*\*\*\*\*\* \* \*

**Le laboratoire doit être remis dans votre dépôt Github.**