Comment faire ses propres boucles de façon simple :

Les fonctions récursives :

On ne va pas utiliser de for et while pour qu’elle fasse appel à elle-même

Les for et while ne peuvent pas se rappeler elles-mêmes. Si on veut les rappeler, je dois les copier à la main x fois

La fonction récursive fait appel à elle-même et peut se rappeler à l’infini (si on ne met pas de conditions) en quelques lignes de code.

Utiles si on a des tableaux dans des tableaux dans des tableaux etc.

Les fonctions générales (non-récursives) :

* fonction qui existent déjà (build-in) – pas besoin de recréer, ça existe déjà et je peux
* fonction que je crée moi-même ( – elles ne peuvent rien inventer de plus que ce que moi j’ai indiqué).

Pour une concaténation en JS, on utilise le +

Argument : les infos que je donne à ma fonction quand je l’appelle (qu’est-ce que mon console.log va indiquer ?). Si plusieurs arguments, il faut les séparer par une virgule.

Pour une concaténation en JS, on utilise le +.

Paramètre : les variables où on va enregistrer les valeurs dedans. C’est ce qu‘on retrouve quand on définit, crée une fonction.

Différents types de paramètres : 2 types ceux à qui on donne une valeur et ceux par défaut.

Mon argument va prendre une valeur par défaut quand il est undifined ou quand je ne lui donne pas de valeur.

Paramètre de …rest :

personnes = ["Georges", "Arielle", "Elio", "Bart"];

nationalites = ["Français", "Allemand", "Belge", "Flamand"];

for (let echec=0 ; croa<personnes.length ; croa++){

console.log(personnes[croa]) + « est » + nationalites[croa] ;

}