**Les boucles**

Je ne vous surprendrai sûrement pas en vous disant que les boucles servent à répéter des suites d’instructions. Il y a deux types de boucles généralement rencontrées, la première étant la boucle « while » (« tant que » en Français). Similairement aux conditions que nous avons vues, la boucle « while » est suivie entre parenthèses d’une condition. Avant chaque tour de boucle, la condition est vérifiée et tant qu’elle est vraie les instructions sont répétées.

Ex : **var counter = 0**

**while (counter < 5)** *ça c’est ma boucle* **{**

**console.log(counter)** // Va afficher successivement 0 puis 1, 2, 3, 4 **counter = counter + 1**

**}**

Remarquez ici deux choses un peu spéciales, d’abord j’affecte la valeur 0 à une variable que j’appelle « counter » (« compteur » en Français), après à chaque tour de boucle je lui donne une nouvelle valeur qui est son ancienne valeur plus un. C’est ce qu’on appelle *l’incrémentation*. Cette dernière étape est tout particulièrement importante car dans le cas contraire notre compteur aurait une valeur fixe et donc notre condition *serait toujours vraie*. Le résultat serait une boucle infinie est donc un ralentissement important de votre ordinateur, voir un crash.

Notre première boucle est parfaitement fonctionnelle mais dans les cas où on sait combien de fois on doit boucler on préférera utiliser une boucle « for » (« pour » en Français) et on gardera les boucles « while » dans les cas contraires, question de praticité et lisibilité.

Une boucle « for » comprend les mêmes parties que notre « while » de l’exemple précédent mais en une version compacte et avec une meilleure lisibilité. L’exemple repris en version « for » donne ceci :

**for (var counter = 0 ; counter < 5 ; counter = counter + 1) {**

**console.log(counter)** // Va afficher successivement 0 puis 1, 2, 3, 4

**}**

Beaucoup mieux non ? Mais reprenons la syntaxe étape par étape. Une boucle « for » se compose du mot-clef « for » suivi par entre parenthèses d’une initialisation, d’une condition puis d’une chose à faire à la fin de chaque tour de boucle (généralement une incrémentation). Chaque partie est séparée par ce qu’on appelle en Anglais un « semicolon » c’est-à-dire le « ; ».

* L’initialisation ne passe qu’une seule fois avant le premier tour de boucle.
* La condition est évaluée à chaque tour de boucle et la boucle s’arrête dès que la condition est invalide.
* La dernière partie contient les choses à effectuer à chaque fin de tour de boucle juste avant de repasser à la vérification de la condition.

*var counter = 0 :**INITIALISATION ne s’exécutera qu’une et une seule fois au début de la boucle For – sinon on tomberait en boucle infinie*

***counter < 5 :*** *CONDITION*

*counter = counter + 1 :**CE QUI DOIT S’EFFECTUER A LA FIN DE LA BOUCLE (ici une incrémentation)*

Même avertissement que pour la boucle « while », il faut ***absolument*** que la condition devienne fausse à un moment sinon on aura affaire à une boucle infinie.

# Quizz

1. Dans quel cas utilise-t-on une boucle « while » et une boucle « for » ?

For quand le nombre d’interactions est connu

1. Comment écrire une boucle « for » qui compte de 5 jusqu’à 10 inclus ? Comment écrire la même boucle avec un « while » ?

for (var counter=5; counter<11; counter=counter+1){console.log(counter);}!

var counter=5

while (counter<11){

console.log(counter)

counter=counter+1

}

1. Comment faire pour afficher à partir d’un nombre stocké dans une variable du nom de multiple sa table de multiplication jusqu’à 10 ?

var multiple pour la table à afficher

var loop pour poser ma boucle

~~for (var multiple=1 ; multiple<=10 ; multiple=multiple+1){console.log(multiple x « le chiffre que je désire »)}~~

for (var loop=1 si je commence à 1 ; loop<=10 ; loop=loop+1) maintenant ma boucle est définie du coup je calcule {console.log(multiple x loop)}

1. Comment écrire une boucle « while » qui compte les chiffres en partant de 1 jusqu’à 20 en incrémentant de 2 à chaque tour ?

var piou=1

while (piou<20){

console.log(piou)

piou=piou+2}

et si je veux qu’à la fin il affiche autre chose, voici ce que je dois préciser

*var piou=1*

*while (piou<20){*

*console.log(piou)*

*piou=piou+2*

*}*

*console.log("Fin !")*

A PARTIR DE LA JE N’AI PAS FAIT MOI-MEME

1. Comment écrire une boucle « for » qui compte les chiffres de 10 à 1 en décrémentant de 1 à chaque tour de boucle ?

for (var counter=10 ; counter>0 ;counter=counter-1){console.log(counter) ;}

for (var piou=10; piou>0; piou=piou-1){console.log(piou);}

1. Quel est le résultat du code ci-dessous ? comment l’écrire correctement

**for (var counter1 = 0 ; counter1 < 3 ; counter = counter + 1) {**

**for (var counter2 = 0 ; counter2 < 5 ; counter = counter + 1) { console.log(counter1)**

**}**

**}**

Déjà mes variables ne s’appellent pas counter mais counter1 et counter2

for (var counter1=0 ; counter1<3 ; counter1=counter1+1){

for (var counter2=0; counter2<5; counter2=counter2+1){

console.log(counter1)

}

}

Je fais une deuxième boucle pour faire tourner la première boucle plusieurs fois. Par exemple si je veux qu’il compte 3x jusqu’à 10, la première boucle le fera compter jusqu’ 10 et la deuxième boucle le fera répéter encore 2x

1. Comment écririez-vous à base de boucles un programme qui affiche la suite de nombre de 0 à 5 trois fois de suite ?

For (var counter1=0; counter1<3 ; counter1=counter1+1){

for (var counter2=0; counter2<6; counter2=counter2+1){

console.log(counter2) ;

} }

for (var cuicui=1; cuicui<4; cuicui=cuicui+1){

for (var piou=0; piou<6; piou=piou+1){

console.log(piou);

}

}