Understanding

1 ]

On définit une valeur max pour l’incrémentation, cette valeur est de 10000.

Ensuite on initialise notre nombre à la valeur voulue (ici 1), puis dans une boucle for on l’incrémente de 1 en 1.

Puis on effectue différents test. Tout d’abord on définit des variables locales via la commande (let « t1 = nr%5 » , par exemple). Respectivement les variables t1, t2, t3 récupèrent les modulos 5,7,9 de la variable que l’on incrémente.

Une fois les variables initialisée, on effectue un test dans un if pour chaque variables , respectivement différent de 3,4,5.

Si un des test est vrai, on continue le programme et on passe à la prochaine itération de la boucle for.

Si tous les tests sont faux, on arrive alors à l’instruction break, qui va nous faire sortir de la boucle for, et nous affichons alors la variable de cette boucle à la valeur correspondant au nombre d’incrémentation qu’elle a subi.

L’exit 0 permet de savoir si l’on a bien rentrer le programme et que tout s’est bien passé.

2 ]

Si aucune des conditions if du programme n’est vérifiée, on ne passe alors pas par l’instruction continue, et l’on arrive au break. Cette instruction va arrêter le déroulement de la boucle for et passer aux lignes de commandes suivant les accolades du for.

Pour résumer le break nous fais sortir de la boucle sans que celle-ci ai atteint sa condition de fin d’execution.

3 ]

Le let initialise les variables t1,t2,t3 en leur affectant le résultat du calcul par modulo.

Donc let permet d’initialiser des variables. Il est aussi possible avec cette commande d’assigner des valeurs à plusieurs variables. On peut stocker dans ces variables une simple valeur ou le résultat d’un calcul arithmétique.

Coding