

# Les boucles : boucle finie ou infinie

## I) Règle

### 1) Algorithme

- Lorsque le programme nécessite d'effectuer **des répétitions jusqu'à ce qu'une condition soit vérifiée**, on peut utiliser une boucle avec une condition « **répéter jusqu'à ce que .....** ».
- Lorsque le programme nécessite d'effectuer **des répétitions un nombre de fois donné** on peut utiliser une boucle avec une condition « **répéter .....fois** ».
- Lorsque le programme nécessite d'effectuer des répétitions sans jamais s'arrêter, on peut utiliser une boucle infinie : « **répéter indéfiniment** »

### 2) Utilisation de la structure conditionnelle sur scratch

Dans Scratch, pour utiliser les boucles finies ou infinies on va dans la catégorie « **Contrôle** » et les blocs de commande utilisés sont :

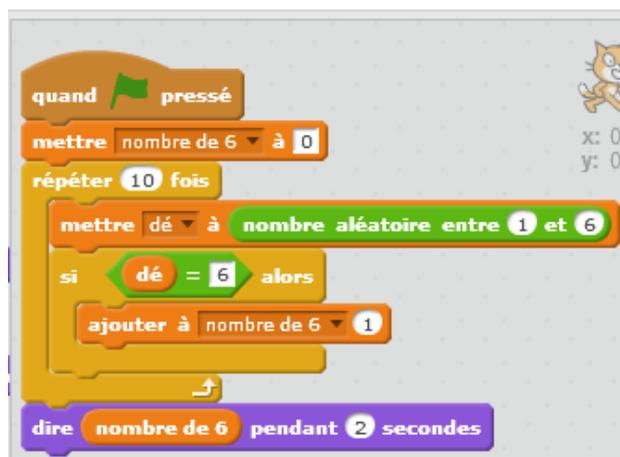


### Exemple :

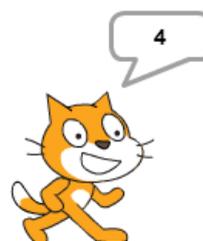
#### Exemple 1 : Répéter un nombre de fois donné :

L'ordinateur va simuler un lancer de dé 10 fois de suite. On note à la fin le nombre de 6 obtenus.

Voici un programme possible :



Une des réponses est :



Ce qui veut dire que sur ces 10 lancers, le chiffre 6 est apparu 4 fois.

- On pourrait aussi rajouter la fréquence du nombre de 6 obtenus il suffit de rajouter :

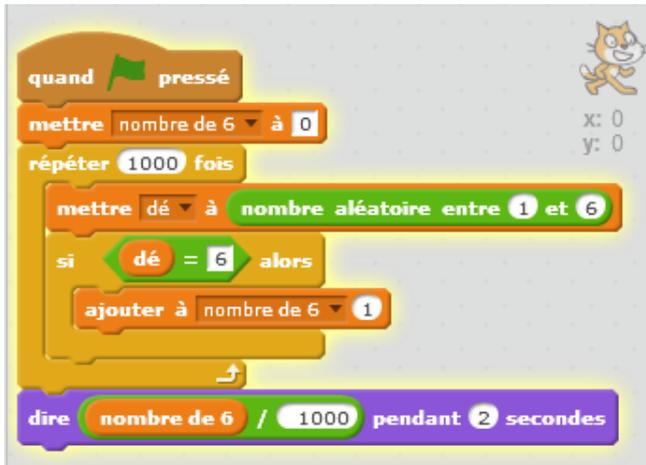


La fréquence du nombre 6 est :

$$\frac{\text{nombre de 6 obtenus}}{\text{nombre de tirages}}$$

- On peut aussi changer le lutin et mettre comme « costume » un carré noir et tracer les 6 faces du dé
- Lorsqu'on introduit la notion de probabilité on peut simuler 100 lancers puis 1000 etc ..... Il suffit de remplacer 10 par 100 ou 1000....

Comme dans l'exemple ci-dessous où on lance 1000 fois un dé , on peut noter que la fréquence du nombre 6 est proche de sa probabilité qui est  $\frac{1}{6} \approx : 0,166...$



Sur plusieurs essais, les résultats ont varié entre 0,16 et 0,18 : ce qui reste très proche de la probabilité d'apparition du chiffre 6

### Exemple 2 : Répéter jusqu'à ce qu'une condition soit vérifiée

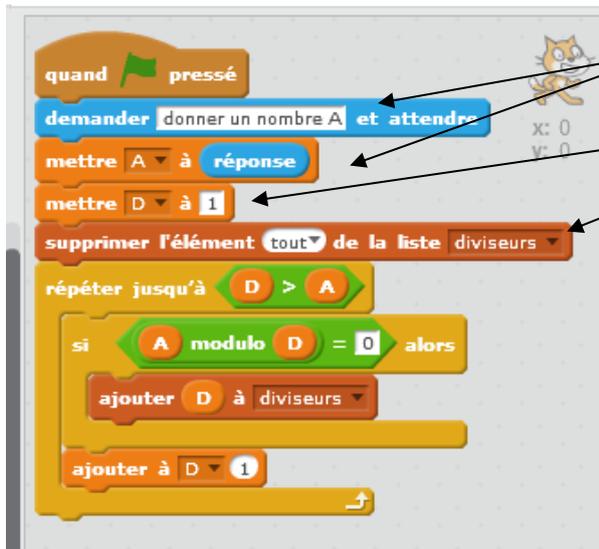
Dans ce programme on cherche la liste de tous les diviseurs d'un nombre donné. Le programme s'arrête tant que D (un diviseur cherché) est inférieur à A (le nombre choisi)

Avant cela il faut créer une liste que l'on nommera « Diviseurs » dans le bloc données :

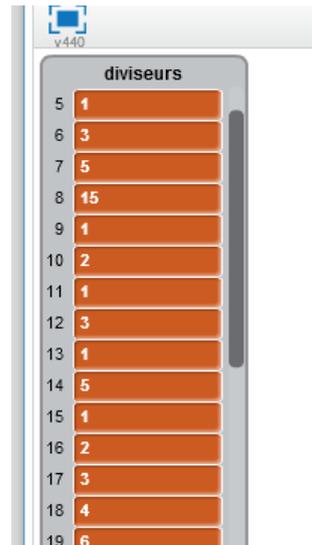


Dès que la liste « Diviseurs » est créée il apparait ce menu :

Le programme est :



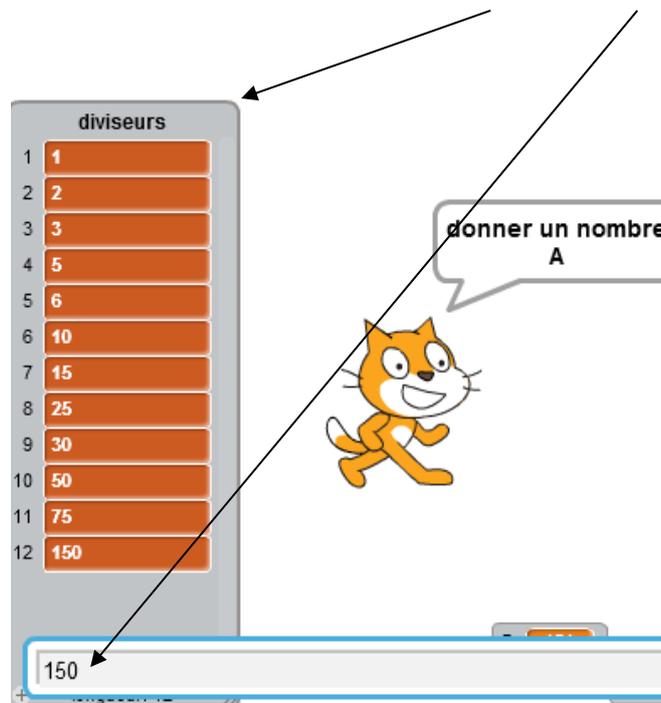
- A est un nombre choisi (On donne à A la valeur choisie)
- D = 1
- On supprime la liste des diviseurs (cela efface la liste après chaque réponse) on évite d'avoir une liste qui enchaîne toutes les réponses précédentes : du type :



**L'algorithme est :**

- Choisir un nombre A
- D = 1
- Répéter jusqu'à D > A (c'est-à-dire tant que  $D \leq A$ )
  - Si Le reste de la division euclidienne de A par D est 0 alors D est un diviseur de A
  - L'ajouter à la liste « Diviseurs »
  - D prend la valeur D+1

**Le lutin demande de donner un nombre A :** On écrit en bas le nombre choisi on clique sur « entrer » et on obtient la liste de tous les diviseurs :



### Exemple 3 : Répéter indéfiniment une action :

Ce programme est tout simple, notre lutin suit indéfiniment le mouvement de la souris

Ce type de boucle est très pratique lors de la création de certains jeux

