Feuille3

Algorithmes et Programmation – Scratch

Nous allons utiliser `Scratch` soit en ligne, soit directement s'il est installé sur vos machines portables (il ne l'est pas dans les salles machines du département).

Dans cette séance nous allons nous intéresser plus particulièrement au tracé avec SCRATCH de figures géométriques et aux déplacements des lutins. Pour les tracés de figures géométriques, on vous conseille de travailler avec le lutin « crayon »dont vous pouvez choisir la couleur et la taille. Un crayon trop grand cache en partie les figures.

Les commandes de déplacement sont dans le menu `Mouvement` ; les commandes de tracé sont dans le menu `Stylo`.

N.B. : Les exercices sont faits pour vous, pour vous faire découvrir `Python`*, ils ne sont pas rédigés pour des élèves de lycée !*

N.B. : Allez à votre vitesse, sautez les exercices qui ne vous apportent rien (certains sont très faciles), approfondissez ceux qui vous paraissent intéressants...

Exercice 1 : *Un peu d'arithmétique*

Q1. Proposez un script où le lutin demande à l'utilisateur un nombre entier positif (il pourra renouveler sa demande tant que l'entier n'est pas strictement positif...) puis calcule la liste des diviseurs de ce nombre

Q2. Créez un bloc à un paramètre qui calcule et remplit la liste des diviseurs du paramètre.

Q3. Proposez un script où le lutin demande à l'utilisateur un nombre entier positif puis détermine si ce nombre est premier ou pas.

Q4. Proposez un script qui donne la liste de tous les nombres compris entre 1 et 100 qui ont exactement 3 diviseurs. Justifiez que le résultat proposé par le lutin est le bon.

Exercice 2 : *Calcul de PGCD en utilisant les listes de diviseurs*

Dans cet exercice, nous allons faire dialoguer plusieurs lutins

Q1. Proposez un script exécuté par un premier `sprit` qui se lance lorsque la touche `espace` est tapée. Ce script devra demander à l'utilisateur deux nombres, vérifier que ces nombres sont strictement positifs et les stocker dans des variables p et q. A la fin de ce script, envoyez un message aux autres `sprite` pour les prévenir que les variables p et q sont entrées : ils peuvent commencer à travailler.

Q2. Proposez un deuxième script exécuté par un deuxième `sprite` qui se lance lorsqu'il reçoit le message du premier script. Ce script devra calculer la liste des diviseurs de p et de q et les stocker dans deux listes `div_p` et `div_q`. A la fin de ce script, envoyez un message aux autres `sprite` pour les prévenir que les listes de diviseurs sont pleines.

Q3. Proposez un troisième script exécuté par un troisième `sprite` qui se lance lorsqu'il reçoit le message du deuxième script. Ce script devra déterminer le plus grand élément commun aux deux listes des diviseurs, c'est-à-dire le PGCD de p et de q. A la fin de ce script, envoyez un message aux autres `sprite` pour les prévenir que le PGCD est calculé.

Q4. Proposez un nouveau script exécuté par le premier `sprite` qui affiche le PGCD de a et b.

Exercice 3 : *Calcul de PGCD avec l'algorithme d'Euclide*

Proposez un script où le lutin demande à l'utilisateur deux nombres entiers positifs puis calcule leur PGCD par l'algorithme d'Euclide.

Exercice 4 : *Un tunnel*

Nous voulons créer un jeu pour piloter à l'aide des flèches du clavier un lutin à l'intérieur d'un tunnel.

Q1. Dessinez un arrière-plan : sur un fond marron, tracez avec un pinceau large un tunnel jaune et positionnez un rond rouge à la sortie.

Q2. Choisissez un lutin, réglez sa taille et positionnez à l'entrée du tunnel.

Q3. Gérez le déplacement du lutin commandé par les 4 flèches du clavier.

Q4. Dans le jeu, lorsque le lutin touche le bord du tunnel (couleur marron), le jeu s'arrête après un message du lutin « j'ai perdu ».

Q5. Dans le jeu, lorsque le lutin touche la sortie rouge, le jeu s'arrête après un message du lutin « j'ai gagné ».

Défi : faites des scripts en parallèle, créez un chronomètre, limitez le temps de jeu et affichez le temps de sortie lorsque le lutin gagne.

Exercice 5 : *Un exercice utilisant des mots*

Le code EAN13 est un code-barres utilisé par le commerce et l'industrie permettant d'identifier des objets de façon unique et d'être lu par un scanner. Ce code-barres est composé de 13 chiffres, le dernier étant une clé de contrôle obtenue de la façon suivante :

- on lit chacun des chiffres de gauche à droite (le premier est dit de rang 1 par convention) ;
- on multiplie par 3 tous les chiffres de rang pair et on additionne tous ces nombres ;
- on ajoute au résultat la somme des chiffres de rang impair ;
- on calcule ensuite le reste de la division euclidienne par 10 ;
- si ce reste vaut 0, la clé est 0 ;
- sinon, la clé est obtenue en retranchant ce reste à 10.

Voici un exemple d'un tel code 978201395357 de clé 3. Vous pourrez utiliser cet exemple pour tester votre code.

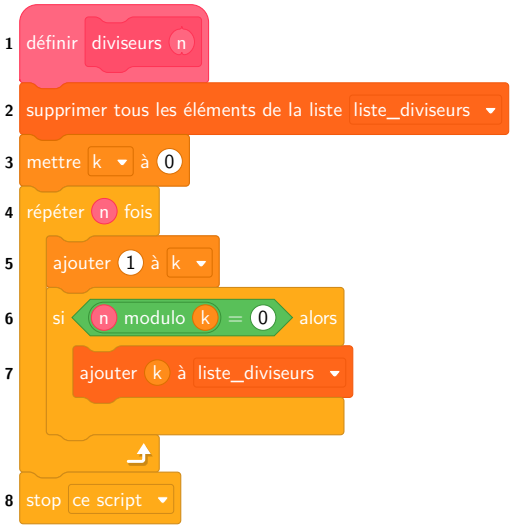
Proposez un script qui demande à l'utilisateur un code-barres à 13 chiffres et vérifie que le code est le bon.

Indication : les chiffres d'un code peuvent être vu par scratch comme un mot et vous pourrez avec accès à chacun de ces chiffres en utilisant l'instruction 

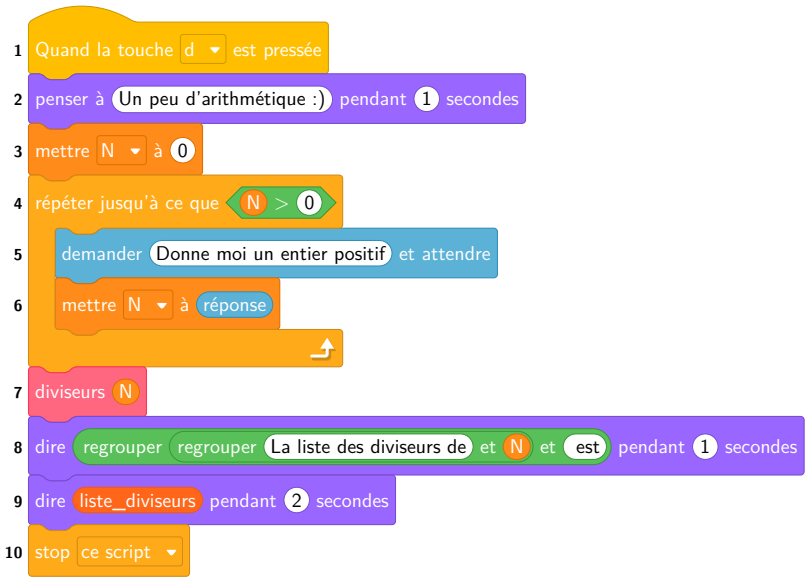
.

Solution de l'exercice 1 : *Un peu d'arithmétique*

Voici un premier bloc qui fabrique la liste des diviseurs d'un entier passé en paramètre.



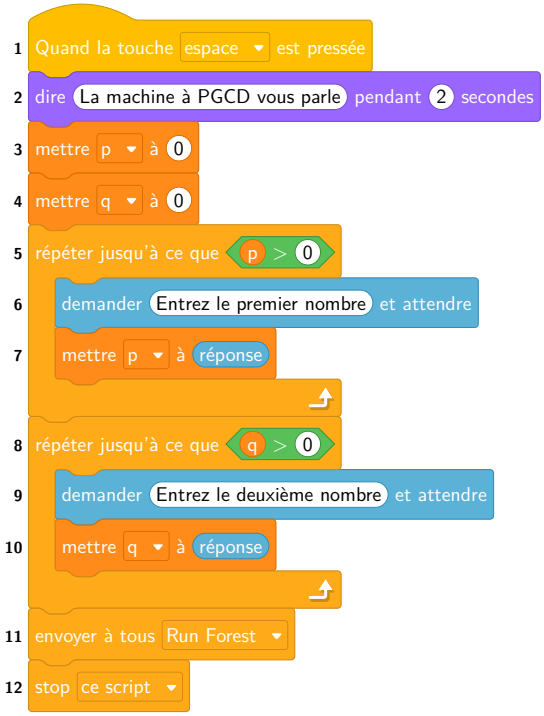
Voici deux scripts qui demandent à l'utilisateur un nombre entier et qui 1-affiche la liste des diviseurs de cet entier, 2-affiche si l'entier est premier.

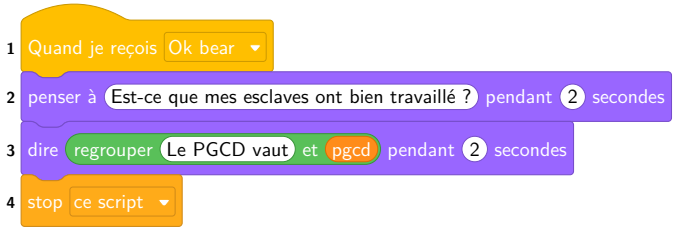


Voici finalement un script qui calcule tous les entiers compris en 1 et 100 qui possèdent exactement 3 diviseurs.

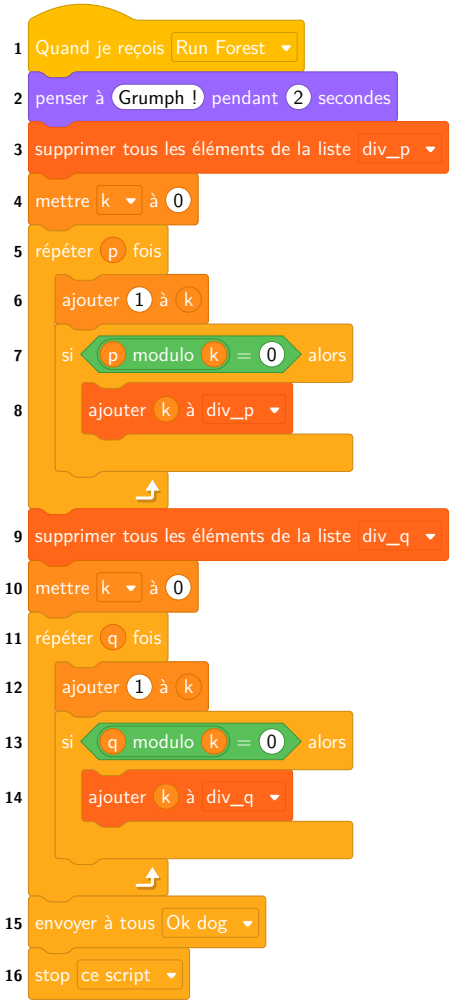
Solution de l'exercice 2 : *Calcul de PGCD en utilisant les listes de diviseurs*

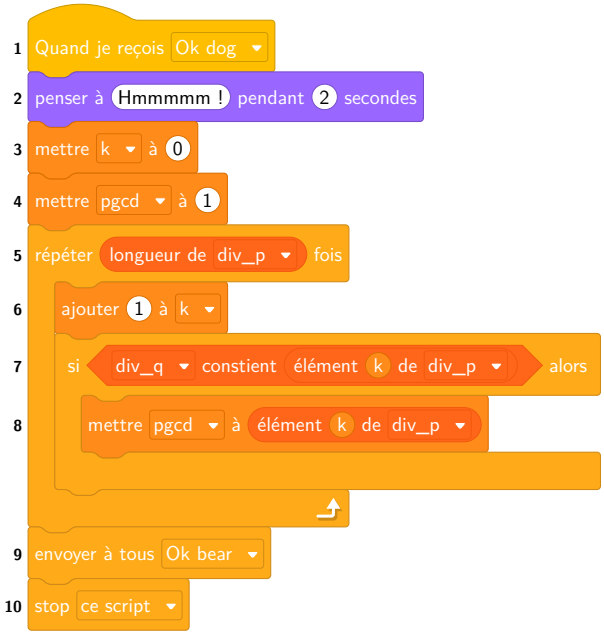
Voici les 2 scripts pour le premier `sprite`





Et voici les scripts pour les deux autres `sprites` (chien et ours)


1 Quand je reçois Run Forest
2 penser à Grumph ! pendant 2 secondes
3 supprimer tous les éléments de la liste div_p
4 mettre k à 0
5 répéter p fois
6 ajouter 1 à k
7 si p modulo k = 0 alors
8 ajouter k à div_p
9 supprimer tous les éléments de la liste div_q
10 mettre k à 0
11 répéter q fois
12 ajouter 1 à k
13 si q modulo k = 0 alors
14 ajouter k à div_q
15 envoyer à tous Ok dog
16 stop ce script

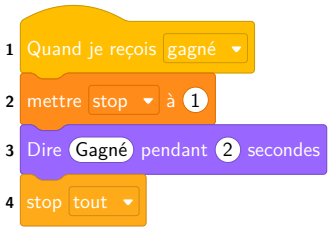

1 Quand je reçois Ok dog
2 penser à Hmmmm ! pendant 2 secondes
3 mettre k à 0
4 mettre pgcd à 1
5 répéter longueur de div_p fois
6 ajouter 1 à k
7 si div_q constient élément k de div_p alors
8 mettre pgcd à élément k de div_p
9 envoyer à tous Ok bear
10 stop ce script

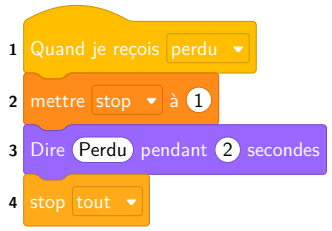
Solution de l'exercice 3 : *Calcul de PGCD avec l'algorithme d'Euclide*

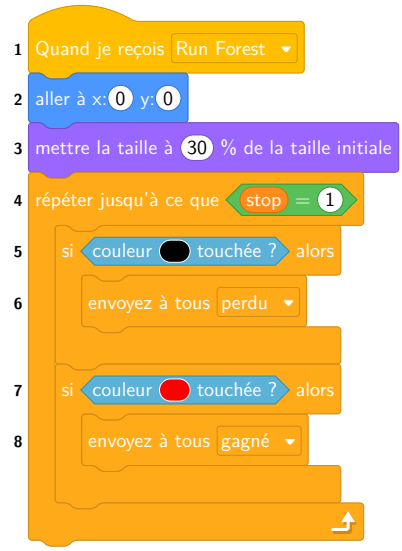
Voici le script pour l'algorithme d'Euclide

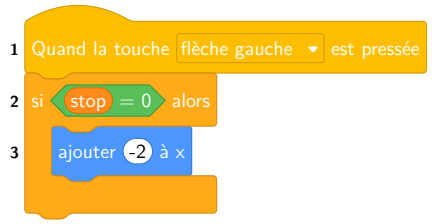
Solution de l'exercice 4 : *Un tunnel*

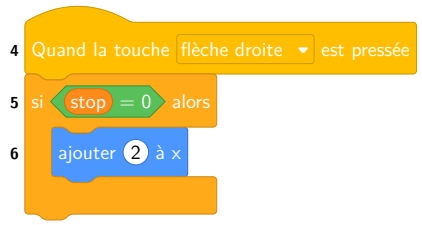
Voici tous les petits scripts en parallèle :

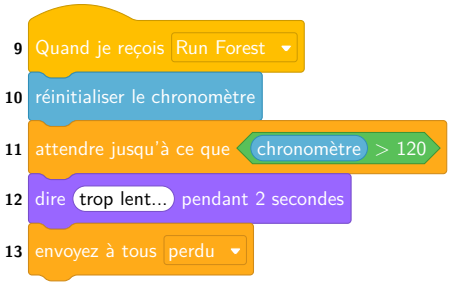


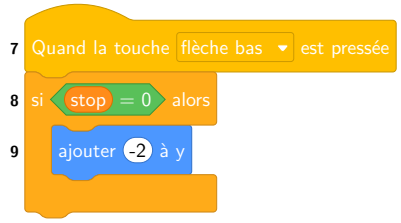


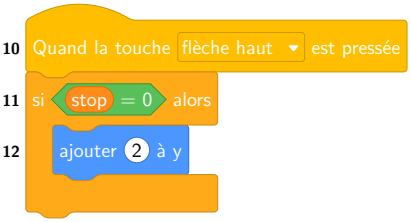












Solution de l'exercice 5 : *Un exercice utilisant des mots*

script pour vérifier le code-barres