



Région académique
PAYS DE LA LOIRE



Comment déboguer ? (avec Scratch)

Ce document a pour seul objectif de donner quelques pistes lorsqu'un programme ne fonctionne pas. Il n'y a aucune recherche d'exhaustivité.

Quelques précautions élémentaires

Précaution n°1

Toujours démarrer un programme par le drapeau vert :



Les autres possibilités correspondent à des interactions clavier une fois le programme démarré. La différence, minime, peut générer un bug :

La variable **réponse** est réinitialisée à « vide » en démarrant au drapeau vert mais n'est pas réinitialisée sinon.

Cela permet aussi de repérer rapidement où est le début du programme !

Précaution n°2

Ne mettre qu'un seul drapeau vert par lutin : lorsqu'il y a plusieurs drapeaux verts, le démarrage ne se fera pas exactement en même temps, il y en aura nécessairement un qui commencera son exécution avant l'autre. Et cela peut créer des conflits, par exemple sur des variables.

Précaution n°3

En lien avec la précédente, l'initialisation des variables peut s'avérer essentielle !



Ce programme va compter de 0 à 10.

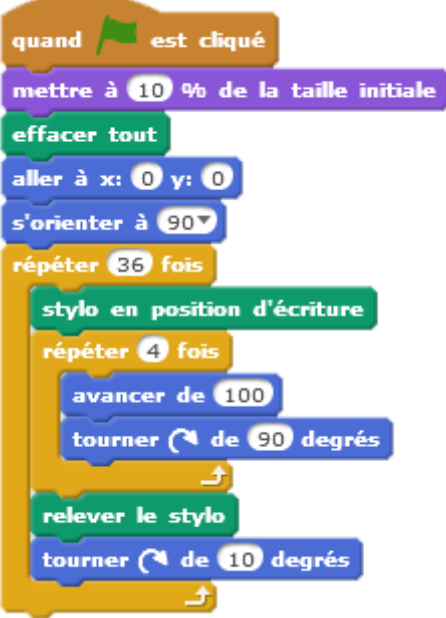
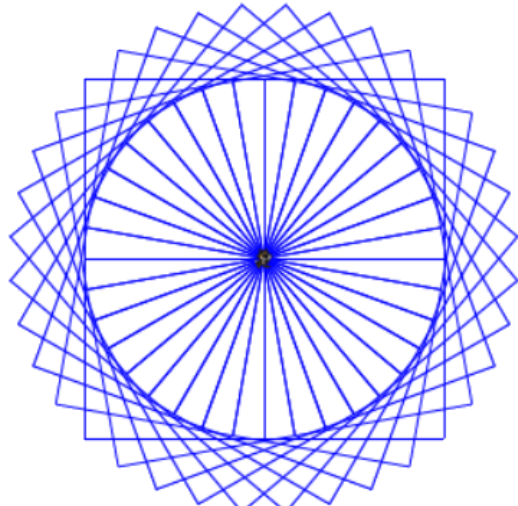


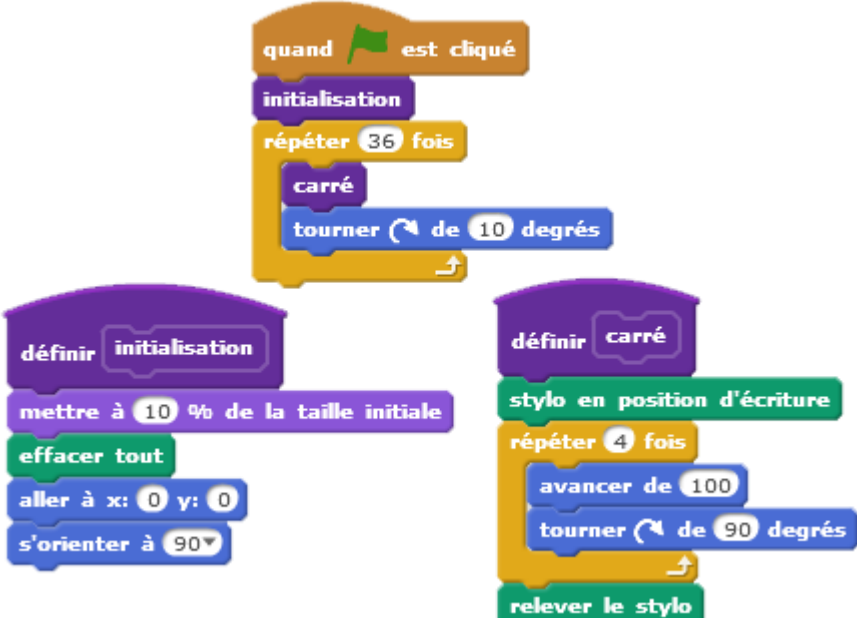
Ce programme va compter de 1 à 10.

Pour aller un peu plus loin...

Structurer : les sous-programmes (ou blocs)

Structurer un programme permet de mieux le comprendre, de l'analyser plus rapidement lorsqu'il y a une erreur.

 <pre>quand [drapeau] est cliqué mettre à 10 % de la taille initiale effacer tout aller à x: 0 y: 0 s'orienter à 90 répéter 36 fois stylo en position d'écriture répéter 4 fois avancer de 100 tourner de 90 degrés relever le stylo tourner de 10 degrés</pre>	<p>Ce premier programme, quoique simple, est plus difficile à appréhender que le suivant alors qu'ils réalisent tous les deux la même chose.</p> 
--	--



```
quand [drapeau] est cliqué
  initialisation
  répéter 36 fois
    carré
    tourner de 10 degrés

définir initialisation
mettre à 10 % de la taille initiale
effacer tout
aller à x: 0 y: 0
s'orienter à 90

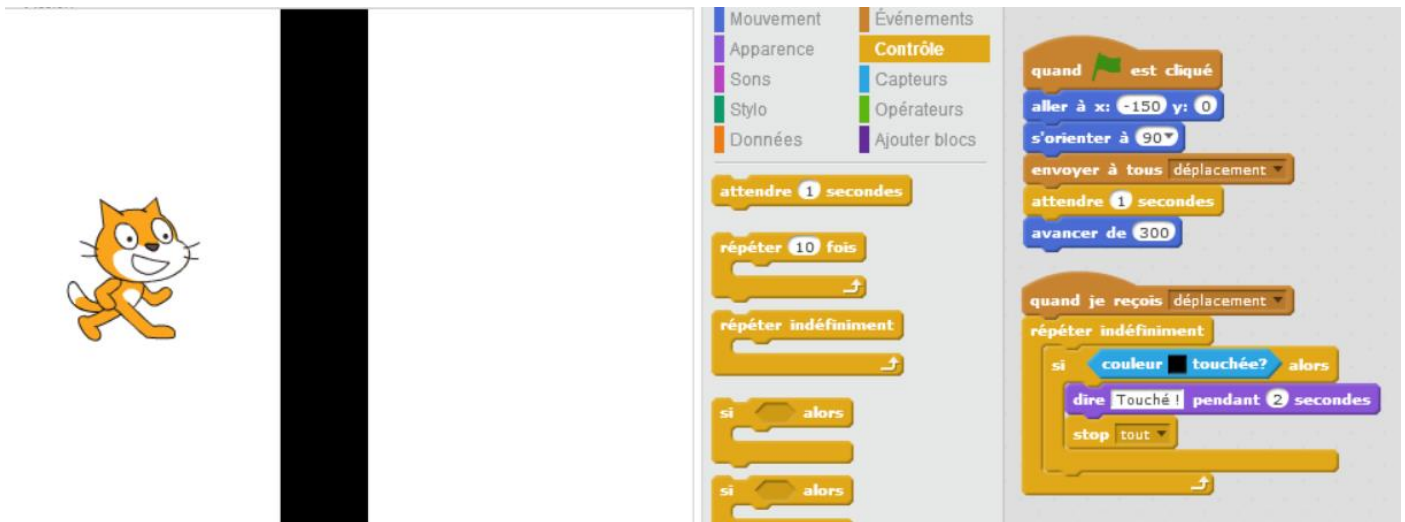
définir carré
stylo en position d'écriture
répéter 4 fois
  avancer de 100
  tourner de 90 degrés
relever le stylo
```

La structuration en sous-programmes aide à la lecture du programme principal (sous le drapeau vert) : on initialise puis on trace un carré en le faisant tourner 36 fois.

On va à l'essentiel d'abord, puis dans les sous-programmes, on peut aller étudier les différents éléments.

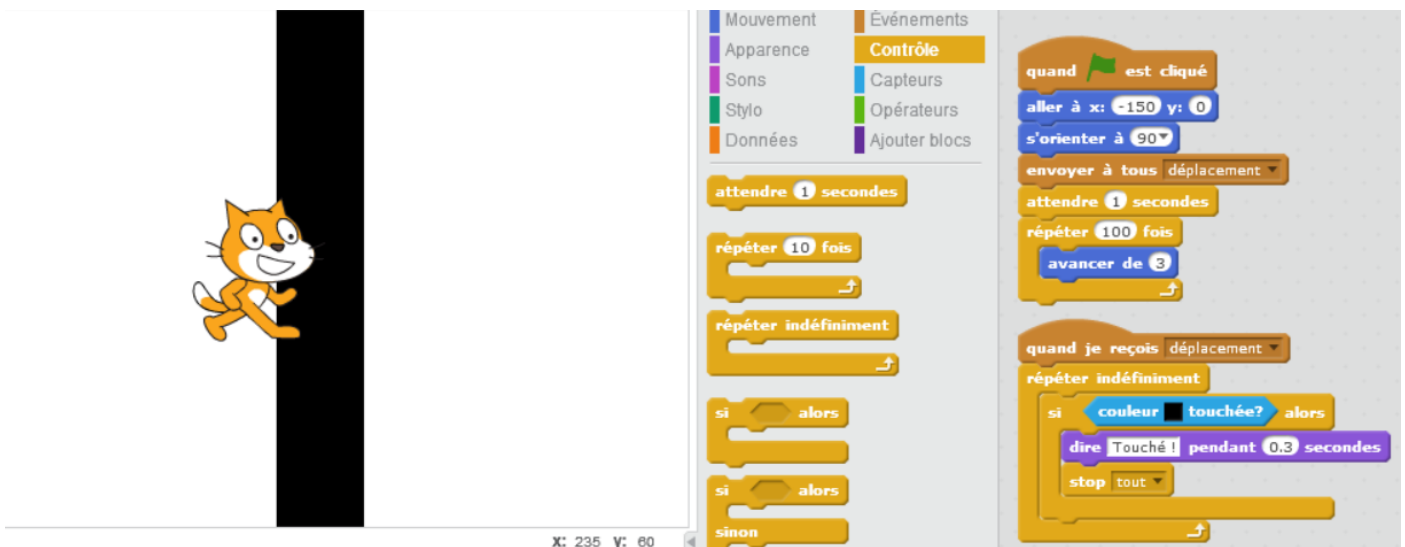
Faire attention aux nuances de certains blocs...

Examinons ce programme, avec la scène associée :



À la lecture de ce programme, dès que le lutin va avancer, il rencontrera rapidement la barre de couleur noire : il devra donc dire « touché ! » et arrêter le programme. Pourtant si on lance avec le drapeau vert, rien de tout cela ne se passe.

Cela signifie que l'instruction **avancer de 300** n'est pas conçue comme un déplacement continu comme notre persistance rétinienne pourrait le croire mais comme 'un saut' de la position initiale, ici (-150 ;0), à la position finale, ici (150 ;0).



Cette nouvelle version fera le lutin s'arrêter et le programme aussi. Modifier le temps du message « dire... » ou le nombre de répétitions de la boucle



est très instructif pour mieux appréhender le fonctionnement de ces instructions avec Scratch.

Voir aussi le document : [gérer la temporalité sur Scratch](#)

Une autre variante :



L'ordre des instructions

Cela peut apparaître comme évident avec ces deux programmes mis côte à côte :



L'un va compter jusqu'à 100, l'autre va compter jusqu'à 1.

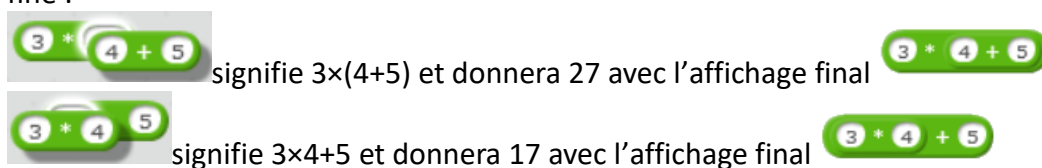
Ce principe de mauvais positionnement d'une instruction peut générer pas mal d'erreurs dans des programmes.

Voir aussi le document : [la boucle répéter](#), et le document [importance de bien lire un programme](#).

Une mauvaise utilisation de blocs ou d'éléments

1) Les opérateurs

L'usage des opérateurs dans le cadre des priorités de calculs nécessite une manipulation fine, voire très fine :



La nuance sur l'affichage final n'est pas évidente à repérer : il faut souvent déplacer un peu un élément pour s'assurer de sa bonne disposition.

2) Ajouter ou regrouper

Il y a là une difficulté liée au fait que Scratch ne prévoit pas de typage des variables. De ce fait, un programme comme celui-ci :



ne renvoie pas de message d'erreur : il laisse la variable « prénom+nom » à zéro.

Ce programme en revanche respectera bien la demande (à un espace près) :



Voir le document : [conjuguer des verbes en ER](#).