



Chouette
Je réussis !

5^e
4^e
3^e

12-15 ans

CONFORME
AU PROGRAMME

Programmer avec Scratch



Pour s'initier
à l'algorithmique
et à la programmation

Ton parcours en 10 étapes



Dans ce cahier, un entraînement très progressif :

FORCE 1

Pour commencer à t'entraîner...

FORCE 2

Tu vas y parvenir !

FORCE 3

Prêt(e) pour l'année prochaine ?

1

Créer
des variables
p. 2

2

Utiliser
des boucles
p. 4



3

Utiliser
des tests
p. 6

4

Programmer
en parallèle
p. 8



5

Se coordonner
à l'aide de messages
p. 10

6

Dessiner
des polygones
p. 12



7

Dessiner
des figures décoratives
p. 14

8

Découvrir
les listes
p. 16

9

Réaliser un projet
complexe (1)
p. 18



10

Réaliser un projet
complexe (2)
p. 20

Fais le bilan

Programmer
avec Scratch
p. 22

En complément

- Les **CORRIGÉS** à détacher au centre du cahier
- L'art de programmer en fin de cahier

Maquette : Pictorus, Primo&Primo

Mise en pages et schémas : Facompo, Rouen

Dessins de la chouette : Patrick Morize

Édition : Jean-Marc Cheminée



Chouette
Je réussis !

5^e
4^e
3^e
12-15 ans

Programmer avec Scratch

Daniel Daviaud
Bernard Revranche

© Éditions Hatier, Paris, 2023

Tous droits de reproduction, de traduction et d'adaptation réservés pour tous pays.
Articles L. 122-5 et L. 122-10 du Code de la propriété intellectuelle.



1 Créer des variables

FORCE 1



Observe et retiens

► L'instruction **mettre** **A** à sert à **mettre une valeur dans la variable A**.

Le **nom de la variable** peut être :

- une lettre : A, n...
- un mot : largeur, score, A1...
- plusieurs mots : petite base, nombre de pommes...

variable

Dans la variable, **on peut mettre** :

- un nombre : 27, -3...
- un mot : ciel, soleil...
- le contenu d'une autre variable : **réponse**, **A** ...
- le résultat d'une expression mathématique : **B + C** ...

► Lorsqu'on met une nouvelle valeur dans une variable, la valeur précédente est **effacée**.

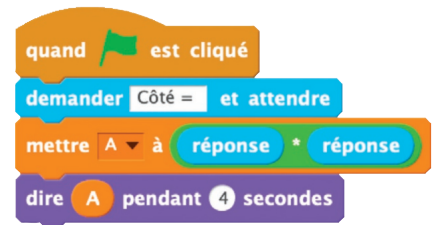
1 Avec un carré

a) Le programme ci-contre demande le côté d'un carré et donne un certain résultat.

Assembler ce programme sans oublier de créer la variable A.

Faire exécuter plusieurs fois avec des côtés différents.

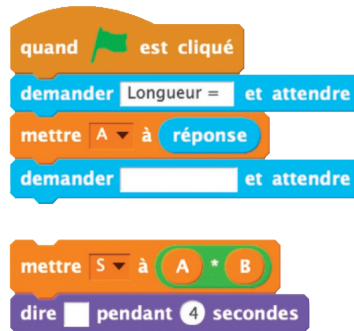
Que donne ce programme ?



b) Modifier le programme pour qu'il donne le périmètre du carré. (Utiliser la variable P.)

2 Avec un rectangle

a) Compléter ce programme qui demande la longueur et la largeur d'un rectangle, et qui donne l'aire du rectangle.



La variable **réponse** est une variable prédéfinie qui reçoit le nombre tapé au clavier.

b) Remplacer la dernière ligne du programme par la ligne suivante :

dire regroupe aire = S pendant 4 secondes .

Exécuter ce nouveau programme.
Observer l'exécution.



La fonction regroupe rassemble deux objets, ici du texte et une variable.

3 Avec des dés

a) Assembler et exécuter le programme suivant qui simule le jet d'un dé à 6 faces.

quand est cliqué
mettre N à nombre aléatoire entre 1 et 6
dire N

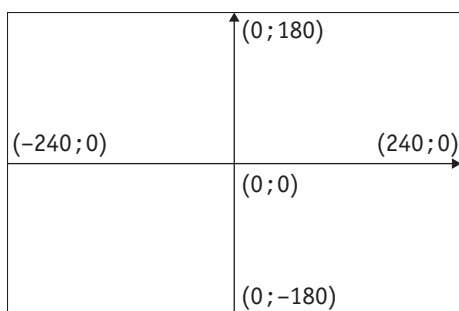


La fonction nombre aléatoire entre 1 et 6 donne un nombre entier au hasard entre 1 et 6.

b) Le modifier afin qu'il simule le jet de deux dés. (Utiliser la fonction regroupe .)

4 À la manière de Pollock

La scène où évolue le lutin est munie d'un repère comme l'indique la figure suivante.



Assembler et exécuter plusieurs fois le programme ci-contre. (Appuyer sur la touche espace pour commencer.)
Observer le résultat.

Remarque : Relancer plusieurs fois de suite le même programme est parfois fastidieux. Dans le chapitre suivant, nous apprendrons à remplacer cette succession de relances par une « boucle ».

quand espace est pressé
cacher
cacher la variable A
cacher la variable B
stylo en position d'écriture
ajouter 10 à la couleur du stylo
mettre A à nombre aléatoire entre -200 et 200
mettre B à nombre aléatoire entre -160 et 160
glisser en 0.5 secondes à x: A y: B



L'instruction cacher efface le lutin de l'écran.
L'instruction cacher la variable A évite l'affichage de cette variable sur l'écran.

5 EXERCICE DÉFI Une bonne année

Écrire un programme qui demande : « Quel est ton âge ? » et qui donne l'année de naissance. (Utiliser la fonction actuel année .)

2 Utiliser des boucles

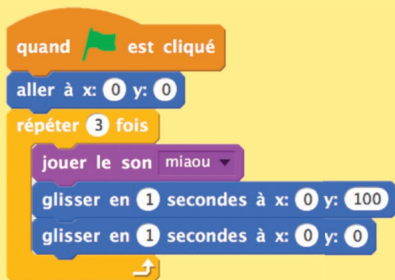
FORCE 2



Observe et retiens



Programme P1



Programme P2

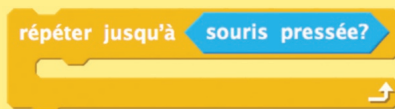
- Le programme P2 effectue la même action que le programme P1, mais il utilise une **boucle** pour répéter trois fois la suite des instructions incluses dans le bloc jaune.

Le programme P2 est ainsi **plus concis** que le programme P1.

- Avec la boucle « répéter indéfiniment » ci-dessous, le programme fonctionnerait **sans s'arrêter**. On doit cliquer sur le panneau « stop » pour l'interrompre.



- La boucle ci-dessous teste si la souris est pressée juste avant de commencer une nouvelle répétition. Dès que la souris est pressée, le programme s'arrête.



1 Des comptes à minet

- a) Assembler et exécuter le programme ci-contre (penser à créer la variable n). Que fait ce programme ?

- b) Modifier le programme pour qu'il compte de 2 en 2.

- c) Modifier ce programme pour qu'il compte à rebours, de 5 jusqu'à 0, comme pour le départ d'une fusée.



2 Chat rit et varie

- Assembler et exécuter le programme ci-contre.
- Modifier le programme pour que le chat se déplace plus vite.
- Modifier le programme pour que le chat rapetisse.

 Utiliser les nombres relatifs et faire démarrer la taille initiale du chat à 300 % au lieu de 25 %.

```

quand [drapeau] est cliqué
  aller à x: 0 y: 0
  mettre à 25 % de la taille initiale
  s'orienter à 40°
  répéter indéfiniment
    avancer de 10
    rebondir si le bord est atteint
    ajouter 0.5 à la taille
  
```

3 Le chronomètre

L'objectif de ce programme est de mesurer une durée assez courte : le temps mis pour faire cuire un œuf à la coque ou pour courir un 5 000 m, par exemple. Au départ, il affiche 0 min 0 sec. Puis, grâce à deux boucles emboîtées, les secondes et les minutes défilent jusqu'à ce qu'on arrête l'exécution (bouton stop). Mais le programmeur, distrait, a commis deux erreurs.

- Assembler ce programme en créant les deux variables et en corrigeant les deux erreurs. Tester son fonctionnement.
- Combien de temps fonctionne ce programme si on attend qu'il s'arrête tout seul ?

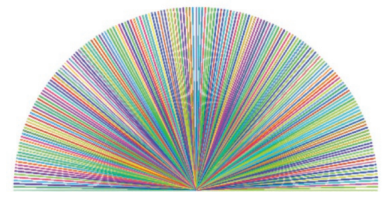
 Les instructions   font afficher en temps réel les deux variables sur la scène.

```

quand [drapeau] est cliqué
  mettre min à 0
  mettre sec à 0
  montrer la variable min
  montrer la variable sec
  répéter 60 fois
    répéter 100 fois
      attendre 1 secondes
      ajouter à sec 1
    mettre sec à 0
    ajouter à min 60
  
```

4 EXERCICE DÉFI L'éventail

Toutes les instructions du programme qui dessine ce bel éventail sont données ci-dessous, mais dans le désordre. Assembler et exécuter le programme.



quand [drapeau] est cliqué

avancer de 230

aller à x: 0 y: -140

cache

s'orienter à nombre aléatoire entre -90 et 90

relever le stylo

mettre la couleur du stylo à nombre aléatoire entre 1 et 200

stylo en position d'écriture

répéter indéfiniment

3 Utiliser des tests

FORCE 3



Observe et retiens

- Un automobiliste sait ce qu'il doit faire en présence d'un feu tricolore :
si le feu est rouge, alors il s'arrête.

Des situations semblables sont fréquentes en programmation.

L'instruction « **si condition alors instruction** » permet de les traiter.

EXEMPLE : En assemblant et en exécutant le programme ci-contre, on observe que :

- si la condition « réponse = oui » est réalisée, alors l'instruction dans le bloc orange est exécutée ;
- sinon le programme ignore cette instruction.

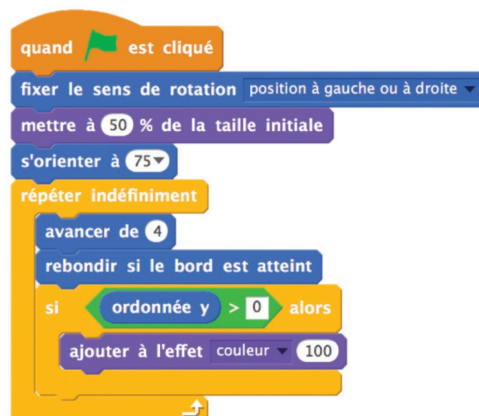


1 L'instruction « si... alors... »

- a) Dans le programme ci-dessus, que dit le lutin si la réponse est « oui » ?
- b) Que dit-il si la réponse n'est pas « oui » ?

2 « Si... alors... » dans une boucle

- a) Choisir le lutin *butterfly3*, puis faire apparaître le repère xy en choisissant *xy-grid* dans la bibliothèque des arrière-plans.
- b) Assembler et exécuter le programme ci-contre. Observer que l'axe des abscisses joue un rôle dans ce programme.
- c) Modifier le programme pour que le rôle de l'axe des abscisses soit joué par l'axe des ordonnées.



3 Les diviseurs d'un nombre entier

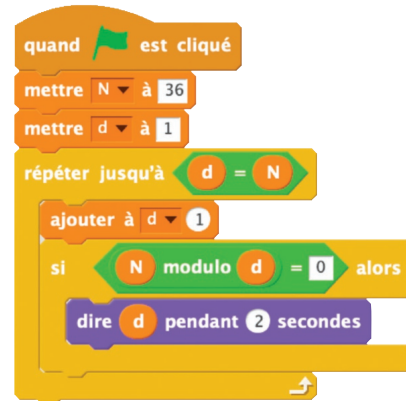
- a) Le reste de la division euclidienne de 42 par 5 est :, celui de 42 par 6 est :
- b) Dans Scratch, les restes de ces divisions s'expriment à l'aide de l'opérateur « modulo ». Compléter.
 42 modulo 5 =; 42 modulo 6 =; 27 modulo 7 =; 12 modulo 3 =

c) Que peut-on dire de deux nombres d et N lorsque $(N \text{ modulo } d) = 0$?

d) Assembler et exécuter le programme ci-contre.
Noter les diviseurs de 36 trouvés par le programme.

e) Ce programme oublie le nombre 1 qui est pourtant un diviseur de 36. Apporter la correction nécessaire.

f) Modifier encore le programme afin qu'il demande un nombre N et donne successivement tous les diviseurs de N .
Application : quels sont les 9 diviseurs de 441 ?



4 Danse de Saint-Guy

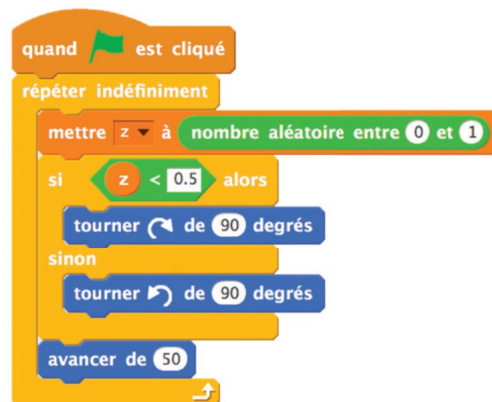
Dans ce programme, on tire au hasard des 0 et des 1.
Si on obtient 0, le lutin tourne vers sa droite.
Si on obtient 1, le lutin tourne vers sa gauche.

a) Assembler et exécuter ce programme.

b) Le lutin, au cours de sa danse, gigote trop vite et risque de disparaître de l'écran.

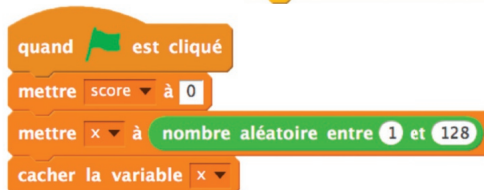
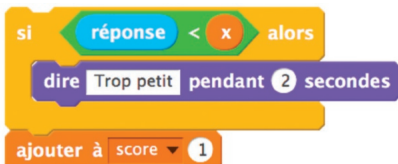
Pour pallier ces deux inconvénients, insérer les instructions suivantes :

attendre 0.1 secondes et rebondir si le bord est atteint .



5 EXERCICE DÉFI Le jeu du nombre mystérieux

Le jeu consiste à deviner, en un minimum d'essais, un nombre entier compris entre 1 et 128.
Pour chaque essai, le programme indique si le nombre proposé est trop grand, trop petit ou bien si le joueur a gagné. Assembler les 4 morceaux du programme dans le bon ordre, puis jouer. Ne pas oublier de créer 2 variables.



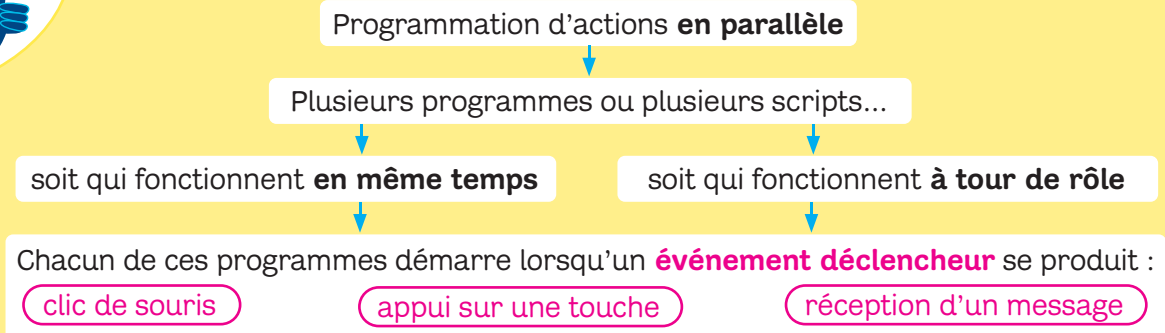
4

FORCE 2

Programmer en parallèle



Observe et retiens



1 Galopades sur l'herbe



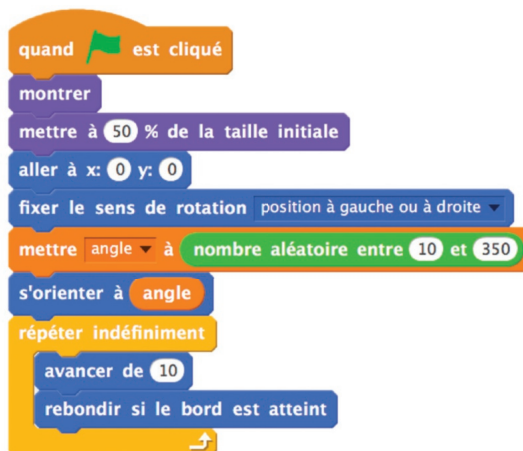
Programme P1

Pendant que Dame Grenouille gambade en parcourant un cercle, Maître Renard traverse la prairie en zigzaguant.

a) Activer les lutins *frog* et *fox*. Cliquer sur *frog* et assembler le programme P1.

Assembler pareillement le programme P2 dans la zone dédiée à *fox*.

b) Cliquer sur le drapeau vert. Qu'observe-t-on ?



Programme P2



L'instruction :

fixer le sens de rotation position à gauche ou à droite
impose au lutin de garder la tête en l'air lorsqu'il rebondit sur un bord.

c) Dans le programme P2, à quoi sert de donner une valeur aléatoire à la variable **angle** ?