

Utiliser un bloc de répétition pour générer des régularités numériques simples

Écrivons un code dans Scratch qui utilise un bloc de répétition pour générer des suites croissantes et décroissantes simples.

Dans la partie 1, vous écrirez le code de base de l'application Scratch.

Dans la partie 2, vous modifierez le code que vous venez d'écrire pour générer une variété de suites croissantes et décroissantes.

Partie 1 : Écrire le code de base pour une régularité numérique

La première régularité numérique que vous générez avec votre programme est la suivante : 0, 5, 10, 15, 20, 25

- Quelle est la règle de régularité pour cette suite de nombres ?

Un *pseudo-code* est une manière informelle de décrire un programme informatique — il se situe à mi-chemin entre le langage courant et le code de programmation. Les programmeurs informatiques, ou codeurs, utilisent souvent le pseudo-code lorsqu'ils planifient leurs programmes.

Le pseudo-code de l'application ressemble à ceci :

```
effacer maSuite[]  
nombre = 0  
répéter 6 fois  
    afficher nombre  
    ajouter nombre à maSuite[]  
    ajouter 5 à nombre  
fin répéter
```

Dans ce code, tout ce qui aurait pu être stocké dans la liste est supprimé.

La valeur initiale de notre variable **nombre** est mise à 0.

Dans le bloc de répétition, **nombre** est affiché, enregistré dans une liste appelée **maSuite** et puis est augmenté de 5.

Utiliser un bloc de répétition pour générer des régularités numériques simples (suite)

Puisque ceci est répété 6 fois, vous verrez 6 nombres dans votre suite.

Écrivons ce code dans Scratch. Le code complet est présenté ci-dessous. Si vous avez de l'expérience avec Scratch, vous pouvez le recréer maintenant et passer à la partie 2. Sinon, suivez les instructions ci-dessous.



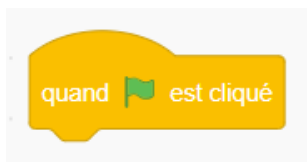
Remarque : Ce code utilise à la fois une variable et une liste. Les variables peuvent contenir une valeur à la fois, tandis qu'une liste peut contenir plusieurs valeurs de manière organisée. Dans ce code, **nombre** est une variable et **maSuite** est une liste.

Instructions détaillées*

*Passez à la partie 2 si vous avez déjà écrit le code Scratch basé sur l'image ci-dessus.

Allez sur <https://scratch.mit.edu/> et sélectionnez **Créer** pour commencer.

1. Dans **Événements**, faites glisser le bloc *quand le **drapeau vert** est cliqué* vers la zone d'édition du code.

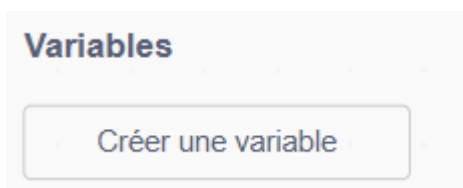


Utiliser un bloc de répétition pour générer des régularités numériques simples (suite)

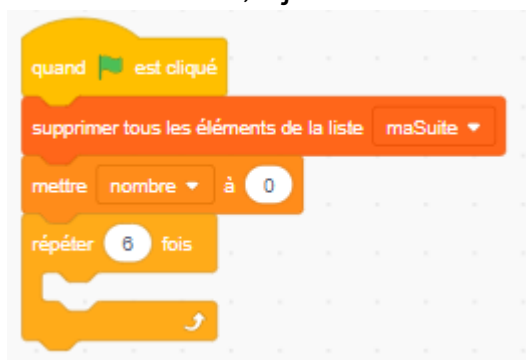
2. Créez une liste appelée **maSuite**. Pour ce faire, à partir de **Variables**, cliquez sur **Créer une liste**. Entrez le nom **maSuite** et cliquez sur **OK**. Pour que la liste soit vide à chaque fois que vous exécutez le code, ajoutez le bloc **supprimer tous les éléments de la liste maSuite** à votre code.



3. Créez une variable appelée **nombre**. Pour ce faire, à partir de **Variables**, sélectionnez **Créer une variable** et appelez-la **nombre**. Ajoutez le bloc **mettre nombre à 0** à votre code.



4. Dans **Contrôle**, ajoutez le bloc **répéter** à votre code et changez la valeur à **6**.



Utiliser un bloc de répétition pour générer des régularités numériques simples (suite)

Pour les étapes 5 à 7, vous ajouterez des blocs à l'intérieur du bloc de répétition.

5. Dans **Apparence**, ajoutez le bloc *dire Bonjour ! pendant 2 secondes* à votre code. Changez la valeur à 1 seconde. Dans **Variables**, glissez **nombre** à l'intérieur du bloc **dire** pour remplacer **Bonjour !**.

6. Dans **Variables**, ajoutez le bloc **ajouter chose à maSuite** à votre code. Glissez **nombre** à l'intérieur du bloc **ajouter** pour replacer **chose**.

7. Finalement, dans **Variables**, ajoutez le bloc **ajouter 1 à nombre** au code, et changez la valeur à 5.



Si vous avez une connexion Scratch, sauvegardez le projet dans votre compte en sélectionnant **Enregistrer maintenant** dans le menu **Fichier**. Une connexion n'est pas nécessaire pour travailler avec le code, mais vous ne pourrez pas sauvegarder votre projet sans elle.

Utiliser un bloc de répétition pour générer des régularités numériques simples (suite)

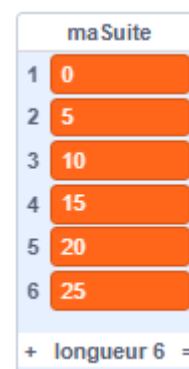
8. Cliquez sur le drapeau vert au-dessus de la scène pour exécuter le code.

L'image-objet devrait afficher les nombres suivants, un à la fois, dans la bulle de dialogue :

0, 5, 10, 15, 20, 25.

Chaque nombre devrait être ajouté à la liste **maSuite** après son apparition dans la bulle.

Si l'application ne fonctionne pas comme prévu, examinez attentivement le code et déboguez en vérifiant si chaque partie du code correspond aux blocs indiqués dans les étapes 1 à 7.



Partie 2 : Modifier le code pour générer d'autres suites croissantes et décroissantes

1. Modifiez le code pour générer une suite de nombres qui augmente de 3 à chaque fois. Produisez 10 nombres dans la suite, en commençant par 0. Les nombres produits devraient être les suivants : 0, 3, 6, 9, 12, 15, 18, 21, 24, 27.

Voici le pseudo-code pour générer cette nouvelle suite de nombres :

```
effacer maSuite[]
nombre = 0
répéter 10 fois
    afficher nombre
    ajouter nombre à maSuite[]
    ajouter 3 à nombre
fin répéter
```

Utiliser un bloc de répétition pour générer des régularités numériques simples (suite)

Le code Scratch ressemblera à ceci :



2. Modifiez un peu cette suite pour commencer à **9** au lieu de **0**.

Les nombres générés devraient être :

9, 12, 15, 18, 21, 24, 27, 30, 33, 36.

3. Modifiez le code pour générer chaque suite de nombres.

Quelle est la différence entre les suites des parties d) et e) et les suites des parties a), b) et c)?

a) 0, 4, 8

b) 10, 12, 14, 16, 18, 20, 22

c) 100, 200, 300, 400, 500

d) 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 0

e) 0, -6, -12, -18

Si vous êtes bloqué, demandez des conseils à un camarade de classe et/ou à votre enseignant.

Défi supplémentaire

Créez vos propres suites et mettez vos camarades au défi d'écrire le code correspondant.