



Programmer avec Scratch 2

Initiation à la programmation

Ndeye Massata NDIAYE

Objectifs spécifiques

A l'issue de cette séquence, l'étudiant sera capable de:

- Comprendre les algorithmes
- Coder avec le langage Scratch 2
- Manipuler les boucles

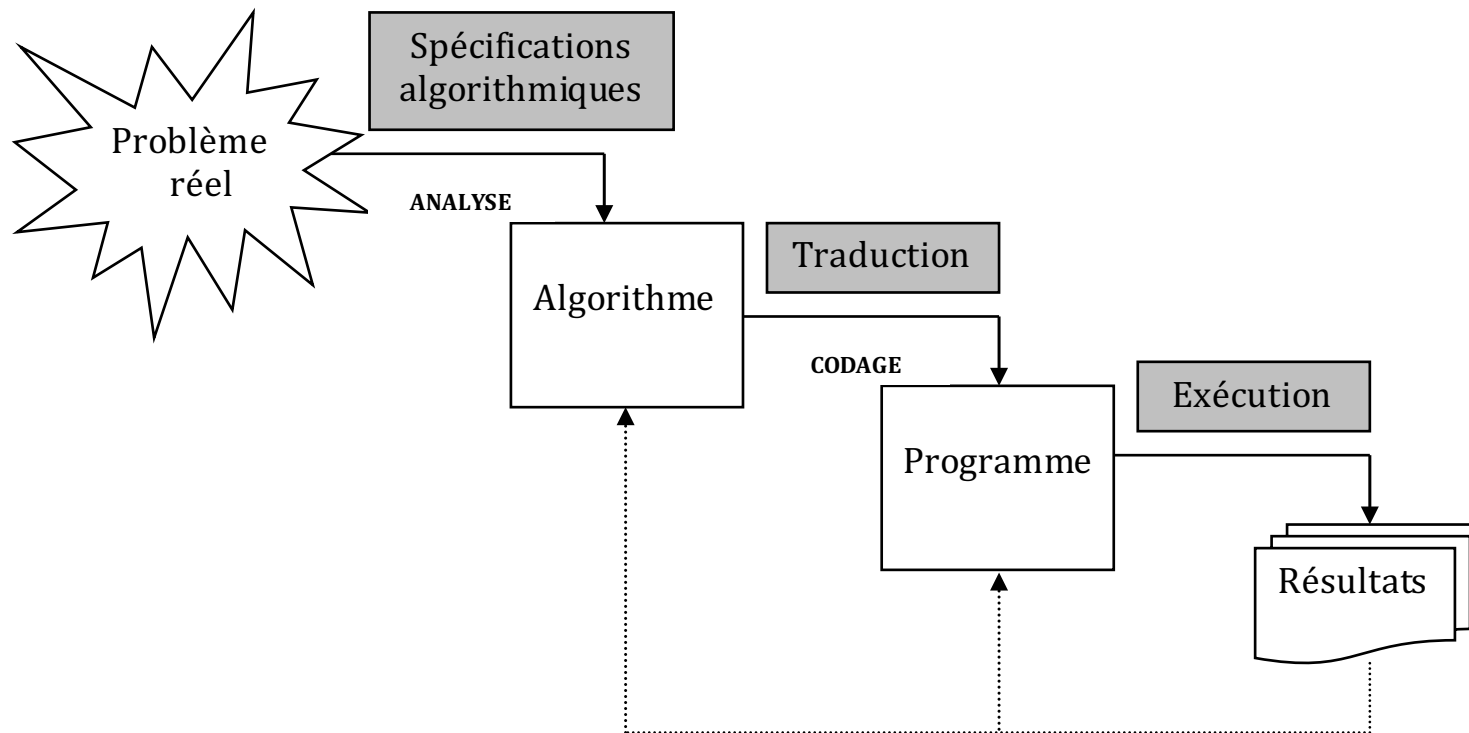


Les algorithmes

- A travers la première séquence, vous avez créé une animation avec le langage Scratch, en d'autres termes vous avez **codé** en utilisant Scratch 2.
- **Coder**, c'est quoi ?
C'est le fait de donner des **instructions** à l'ordinateur qu'il doit exécuter pour donner les résultats escomptés.
Exemple: avec les différents blocs de Scratch, vous avez donné des instructions à l'ordinateur pour qu'il exécute des actions que vous souhaitez obtenir avec le chat.
- Un **algorithme**, c'est quoi alors ?
Une série d'actions ordonnées en séquence.
Une liste ordonnée et logique d'instructions permettant de résoudre un problème.



Etapes du processus de programmation



Les parties d'un algorithme

- Un algorithme se compose de trois grandes parties:
 - Les informations dont on a besoin au départ;
 - La succession d'instructions à appliquer;
 - La réponse que l'on obtient à l'arrivée.

Exemple: addition de deux nombres

- Informations de départ: **nombres A et B**
- Succession d'instructions: **calculer A+B**
- Réponse à l'arrivée: **la somme obtenue**



Quelques règles d'un algorithme

- Par souci de clarté, un algorithme doit éviter de comporter plusieurs fois la même série d'instructions
- Solutions
 - utiliser les **boucles**
 - Utiliser les **fonctions/procédures**



La programmation

- Pour qu'un algorithme soit exécuté par une machine, il faut réunir au moins deux conditions:
 1. Écrire un algorithme qui permet d'atteindre le résultat visé
 2. Traduire l'algorithme en **programme** compréhensible par la machine
- Un **programme**: c'est un algorithme traduit en instructions écrites dans un langage de programmation (Scratch2) pour que la machine puisse l'exécuter.



Cycle de programmation

- Pour pouvoir communiquer avec les machines, on utilise un langage de programmation (C, Java, Scratch, PHP,)

Utilisateur → algorithme → langage → ordinateur



Les parties d'un programme

- En programmation, les trois grandes parties d'un algorithme sont:
 - Les informations dont on a besoin au départ → les entrées (lecture)
 - La succession d'instructions à appliquer → le traitement
 - La réponse que l'on obtient à l'arrivée → les sorties (sortie)



Remarques

- Quand on dit « **lire** » ou « **écrire** » on se place du point de vue du programme ou de la machine.
- On **lit** les entrées : au clavier, à la souris (position ou clic) , à partir d'un fichier (image, texte, son), à partir d 'un capteur ... Cela arrête le programme et attend qu'une action soit réalisée par l'usager ou un autre programme afin d'obtenir une valeur.
- On « **écrit** » les sorties : affichage à l'écran, sur une imprimante, dans un fichier...



Exemple: ajouter 2 à un nombre A qui vaut 4

Nombre de départ: A

Attribuer à A la valeur 4

Ajouter 2

Le résultat est 6

Algorithme

donner à A la valeur 4

ajouter 2 à A

écrire A

Scratch



Les variables

- Une **variable** est une information contenue dans une « boîte », que le programme va repérer par son nom. Pour avoir accès au contenu de la boîte, il suffit de la désigner par son nom. Le contenu de cette « boîte » dépend du type de variable.
- Il y a plusieurs types de variables:
 - numérique : entier, réel
 - texte : caractère, chaîne
 - Booléen

NB: Certains langages n'utilisent pas la déclaration de type. C'est le cas de Scratch.



Affectation

- Définition: **Affecter** une valeur à une variable, c'est donner une valeur à cette variable.

Exemple 1:

$A \leftarrow 4$

Mot \leftarrow « coucou »

Scratch



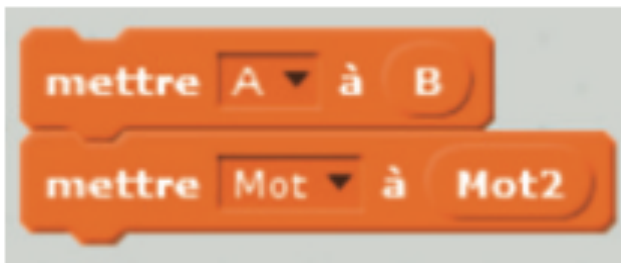
Exemples

Exemple 2

$A \leftarrow B$

$\text{mot} \leftarrow \text{mot2}$

Scratch

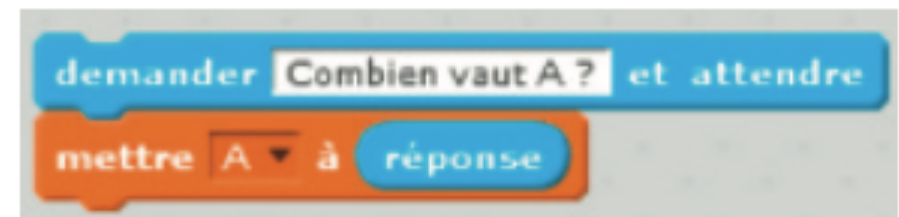


Exemple 3

Lire un nombre

$A \leftarrow \text{nombre}$

Scratch



Opérations sur les variables

Types de la variable	Opérateurs
numérique	+, -, *, /, ^
texte	& , + (concaténation de deux chaînes)
booléen	et, ou, non

Exemple

Langage algorithmique

$A \leftarrow 2(B+5)$

mot \leftarrow mot2 & « oui »

Scratch



Les tests conditionnels

Définition

Un test permet de choisir une action suivant une condition.

Structure d'un test

Si **condition**

alors

action1

FinSi

Si **condition**

alors

action1

Sinon

action2

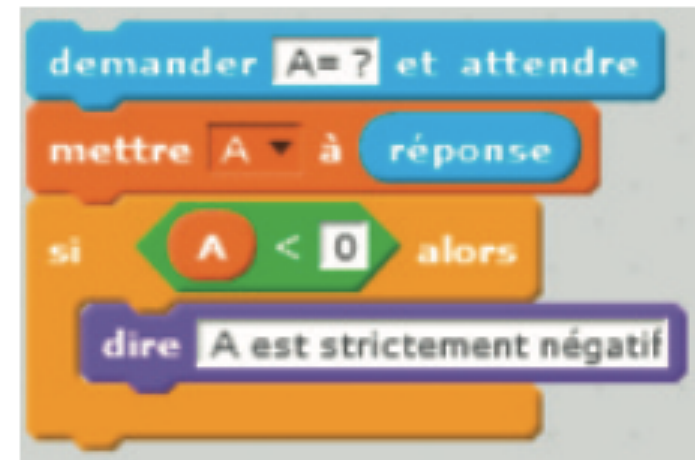
FinSi



La condition ?

En général une condition est une comparaison, elle est **vraie** ou **fausse**. La condition peut aussi être une variable de type booléen. On peut utiliser des opérateurs : « égal à » « différent de » « plus petit que »

```
lire A
Si A<0
alors
    écrire « A est strictement négatif »
finsi
```



Exemple

Langage algorithmique

lire A et B

Si A n'est pas nul

alors

diviser B par A

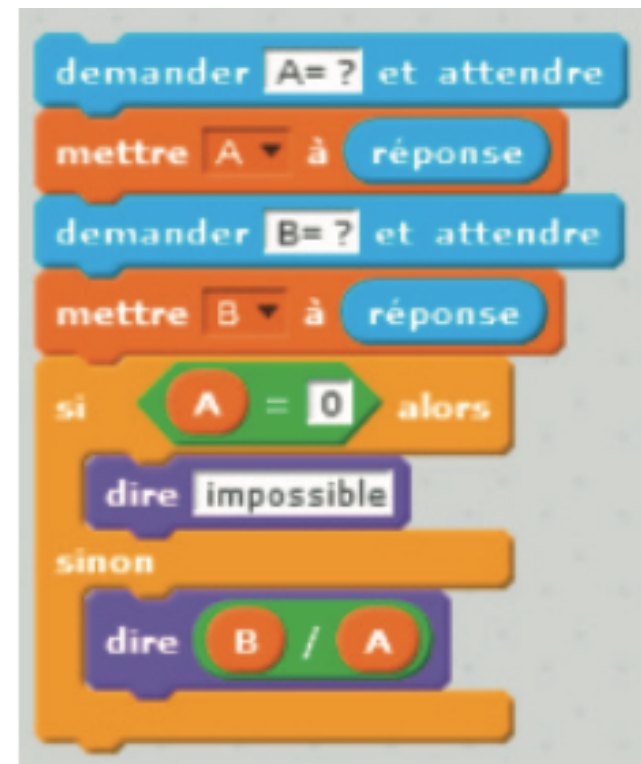
écrire B

Sinon

écrire « impossible »

Finsi

Scratch



LES BOUCLES

La boucle POUR

Une itération sert à répéter une même action. Une fois la répétition finie, le programme continue.

Avec la boucle Pour, on connaît le nombre de fois où l'action doit être répétée.

Exemple:

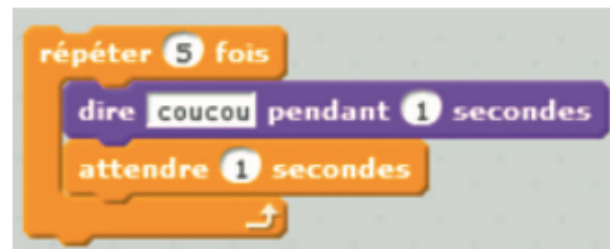
Langage algorithme

Répéter 5 fois:

 écrire « coucou »

Fin répéter

Scratch



Exemple: afficher tous les entiers de 1 à N (donné)

Langage algorithmique

lire N entier
Répéter
 pour i de 1 à N :
 écrire i
Fin répéter

Scratch



Les boucles « Tant que »

Une boucle « **Tant que** » sert à répéter une même action, jusqu'à ce qu'une condition se réalise.

Remarque: on ne sait pas à l'avance le nombre de fois que la boucle sera répétée.

Exemple: Demander « 3 fois 2 ». Lire la réponse N.

afficher « 3 fois 2 = »

$N \leftarrow 0$

Tant que $N \neq 6$:

lire N

Si $N \neq 6$

Afficher « erreur »

fin de si

Fin tant que

