C5T15 – Algorithmique et programmation

Objectif 15-1 Pour prendre un bon départ

1. Quelques définitions (dictionnaire de l'académie française)

Information

Élément de connaissance, traduit par un ensemble de signaux selon un code déterminé, en vue d'être conservé, traité ou communiqué.
Sa représentation sous une forme adaptée à son exploitation est appelée **donnée**.

Par exemple, pour établir une fiche de paye, il faut disposer, entre autres, des données suivantes : nombres d'heures travaillées dans le mois, taux horaire, divers taux de cotisations, ...

Traitement de l'information (ou traitement informatique des données)

Emploi d'ordinateurs en vue d'effectuer des opérations logiques et mathématiques complexes à des fins scientifiques, administratives, etc.

Exemple : calculer et éditer une fiche de paye.

Informatique

Science du traitement automatique de l'information ; l'ensemble des applications de cette science.

Par exemple, connaître différentes façons de trier une liste, étudier leurs efficacités respectives.

Système informatique

Ensemble des moyens qui permettent de conserver, de traiter et de transmettre l'information.

Réseau

Ensemble de systèmes informatiques communiquant entre eux par voies locales, privées ou publiques.

Ordinateur

Terme générique qui désigne un équipement informatique permettant de traiter des informations.

Hardware ou matériel

Ensemble des éléments physiques (microprocesseur, mémoire, écran, clavier, disques durs. . .) utilisés pour traiter les données.

C5T15 – Algorithmique et programmation

Programme

Suite d'instructions rédigées dans un langage spécifique, qui commande à un dispositif, à un appareil ou à un système informatique d'exécuter une tâche donnée.

Software ou logiciel

Ensemble de programmes.

Système d'exploitation (Operating system , en abrégé O.S.)

C'est l'interface entre le hardware et les logiciels utilisés par les utilisateurs finaux.
Il assure la gestion du fonctionnement d'un l'ordinateur, d'une tablette, d'un smartphone, ...

Dans un ordinateur, le système d'exploitation gère tous les transferts d'informations, établit les communications avec l'extérieur, constitue des files d'attente de travaux et de résultats, assure l'enchaînement automatique des travaux, optimise la gestion des ressources, respecte les priorités relatives demandées entre tâches, analyse son propre fonctionnement pour son optimisation et son dépannage.

Exemples : Windows, Linux, Mac OS, Android, ...

Fichier de données

Ensemble organisé de données ayant trait à un même sujet, enregistré sur un support informatique, et facile à consulter ou à modifier par ordinateur.

Exemple : la liste des employés d'une entreprise ainsi que leurs coordonnées.

Algorithme

Suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat

Que produit l'exécution de l'algorithme ci-dessous ?

<table>
<tr><td rowspan="2">Algorithme :</td><td colspan="2">Procédure « côté » :
• tracer un segment de 5 cm
• tourner de 120° à droite</td></tr>
<tr><td>Programme :</td><td>Programme plus efficace :</td></tr>
<tr><td>tracer un segment de 5 cm
tourner de 120° à droite</td><td>« côté »</td><td rowspan="3">Répéter 3 fois « côté »</td></tr>
<tr><td>tracer un segment de 5 cm
tourner de 120° à droite</td><td>« côté »</td></tr>
<tr><td>tracer un segment de 5 cm
tourner de 120° à droite</td><td>« côté »</td></tr>
</table>

Réponse : Le tracé d'un triangle équilatéral de côté 5 cm.

Langages de programmation

Composés d'un alphabet, d'un vocabulaire, de règles de grammaire et de syntaxe, les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par l'ordinateur, et d'autre part d'indiquer comment seront traitées les données, selon quels algorithmes.

Dans ce thème nous verrons des exemples en langages Scratch et Python3.

2. Côté matériel

Informations détaillées ici **:** http://www.vulgarisation-informatique.com/architecture-pc.php

3. Quelques unités utilisées en informatique

bit

Un bit est un chiffre binaire (0 ou 1). C'est l'unité élémentaire d'information.

octet

Un octet est une unité d'information composée de 8 bits. Attention en anglais se dit « byte »

pixel

Le pixel est la plus petite unité adressable sur l'écran ou sur une image. En général composée de trois points de couleur rouge, vert et bleu, pour un écran couleur. Abréviation px.

Dpi et ppm

Pour les imprimantes on utilise :
dpi : dot per inch, c'est-à-dire points par pouce
ppm : page per minute, c'est-à-dire pages par minute.

Voir aussi https://fr.wikipedia.org/wiki/Unité_de_mesure_en_informatique

4. Algorithmes et organigrammes

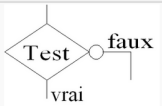
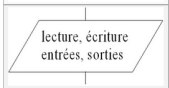
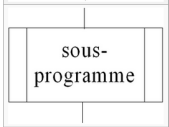
Les organigrammes permettent de visualiser les algorithmes et de décrire symboliquement toutes les opérations effectuées par l'ordinateur pour résoudre un problème.

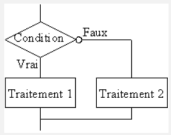
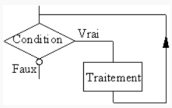
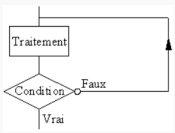
Un organigramme se présente sous la forme d'un graphe orienté, que l'on parcourt de haut en bas et en suivant les flèches.

Une norme ISO a été développée, elle porte le numéro ISO 5807.

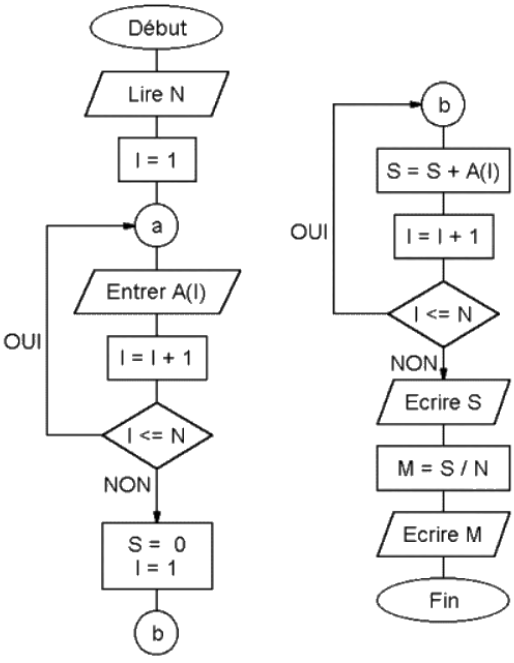
Elle décrit en détail les différents symboles à utiliser.

https://fr.wikipedia.org/wiki/Organigramme_de_programmation

Symboles normalisés	Commentaires:
	Les tests ou branchements conditionnels : • la pointe vers le haut est l'entrée du test ; • la pointe avec le rond est le résultat du test lorsqu'il est faux ; • la pointe vers le bas est le résultat du test lorsqu'il est vrai.
	Mise à disposition d'une information à traiter ou enregistrement d'une information traitée.
	Appel de sous-programme.

Séquence linéaire	Séquence alternative "si…alors…sinon"	Séquence répétitive "tant que…faire…"	Séquence répétitive "répéter…jusqu'à…"
			
Début • "Traitement 1" • "Traitement 2" Fin	Si "condition" • alors "Traitement 1" • sinon "Traitement 2" Fin si	Tant que "condition" • faire "traitement" Fin tant que	Répéter "traitement" jusqu'à "condition"

Exemple



Il s'agit du calcul de la somme et de la moyenne de N valeurs placées dans un tableau à une dimension A(I).

C5T15 – Algorithmique et programmation

Objectif 15-2 Algorithmes et programmation

Un algorithme exprime la structure logique d'un programme informatique et de ce fait est indépendant du langage de programmation utilisé.

1. Programmation

Avertissement : dans un premier temps pour une meilleure compréhension du paragraphe il est conseillé de sauter les remarques écrites en rouge.

La programmation d'un ordinateur consiste à lui «expliquer» en détail ce qu'il doit faire et sur quelles données doivent porter les actions, en sachant qu'il ne «comprend» pas le langage humain, mais qu'il peut seulement effectuer un traitement automatique sur des séquences de 0 et de 1.

* Un ordinateur n'est rien d'autre qu'une machine effectuant des opérations simples sur des séquences de signaux électriques, lesquels sont conditionnés de manière à ne pouvoir prendre que deux états seulement (par exemple un potentiel électrique maximum ou minimum). Ces séquences de signaux obéissent à une logique du type « tout ou rien » et peuvent donc être considérés conventionnellement comme des suites de nombres ne prenant jamais que les deux valeurs 0 et 1. Un système numérique ainsi limité à deux chiffres est appelé système binaire.

Programme source (ou code source)

Écrit dans un langage de programmation, un programme source permet :
- de décrire les structures des données qui seront manipulées par l'ordinateur
- d'indiquer comment seront traitées les données, selon quels algorithmes.

Exemples

Logiciel de géométrie	Avec un tableur	Avec Scratch
Tracer [AB]	Calculer le nombre A+2	Avancer de 50 pas
Choix de l'action « tracer un segment » Clic sur le point A, Clic sur le point B.	Écrire le nombre A en A1 Choisir la cellule A2 écrire : « =A1+2 »	Choisir la brique « avancer de 10 pas » Changer 10 par 50

Traduction en langage machine (code objet)

Pour « parler » à un ordinateur, il nous faudra donc utiliser des systèmes de traduction automatique.

* Un ordinateur est totalement incapable de traiter autre chose que des nombres binaires. Toute information d'un autre type doit être convertie, ou codée, en format binaire. Cela est vrai non seulement pour les données que l'on souhaite traiter (les textes, les images, les sons, les nombres, etc.), mais aussi pour les séquences d'instructions que l'on va fournir à la machine pour lui dire ce qu'elle doit faire avec ces données.

Ces traducteurs font passer du code source au langage machine compréhensible par l'ordinateur.

* Interpréteur ou compilateur ? Quand la traduction s'effectue au fur et à mesure on parle d'interpréteur, l'utilisateur ne la voit pas. Quand la traduction s'effectue d'un bloc on parle de compilateur. Un compilateur est un programme spécial qui, à partir d'un code source, génère un fichier exécutable qui contient le code objet. (Par exemple avec une extension .exe)

Programmeur → algorithme → langage de programmation → traduction→ langage machine → ordinateur

2. Scratch

Ce paragraphe résume la page web http://open-source.developpez.com/tutoriels/scratch/logiciel-a-tout-faire/

Le principe

Scratch est dynamique, il permet de modifier le code du programme en cours d'exécution. Il traite avec une grande facilité des concepts de base de la programmation comme les boucles, les tests, les affectations de variables, et surtout la manipulation des objets, comme les sons et les vidéos.

Scratch est visuel, tout le code est directement inscrit sous forme de briques en couleurs (par exemple les contrôles en jaune, les variables en rouge, les mouvements en bleu).

Scratch est libre. Le nom de Scratch fait référence à cet art de mélanger des sons grâce aux tables de mixage, comme à cette possibilité de réutiliser des objets.

Scratch peut être utilisé en ligne via un navigateur internet comme Chrome ou Firefox.

Vous pouvez aussi installer le logiciel pour l'utiliser hors ligne sur un ordinateur disposant de Linux, Windows ou Mac OS.

Utilisation en ligne

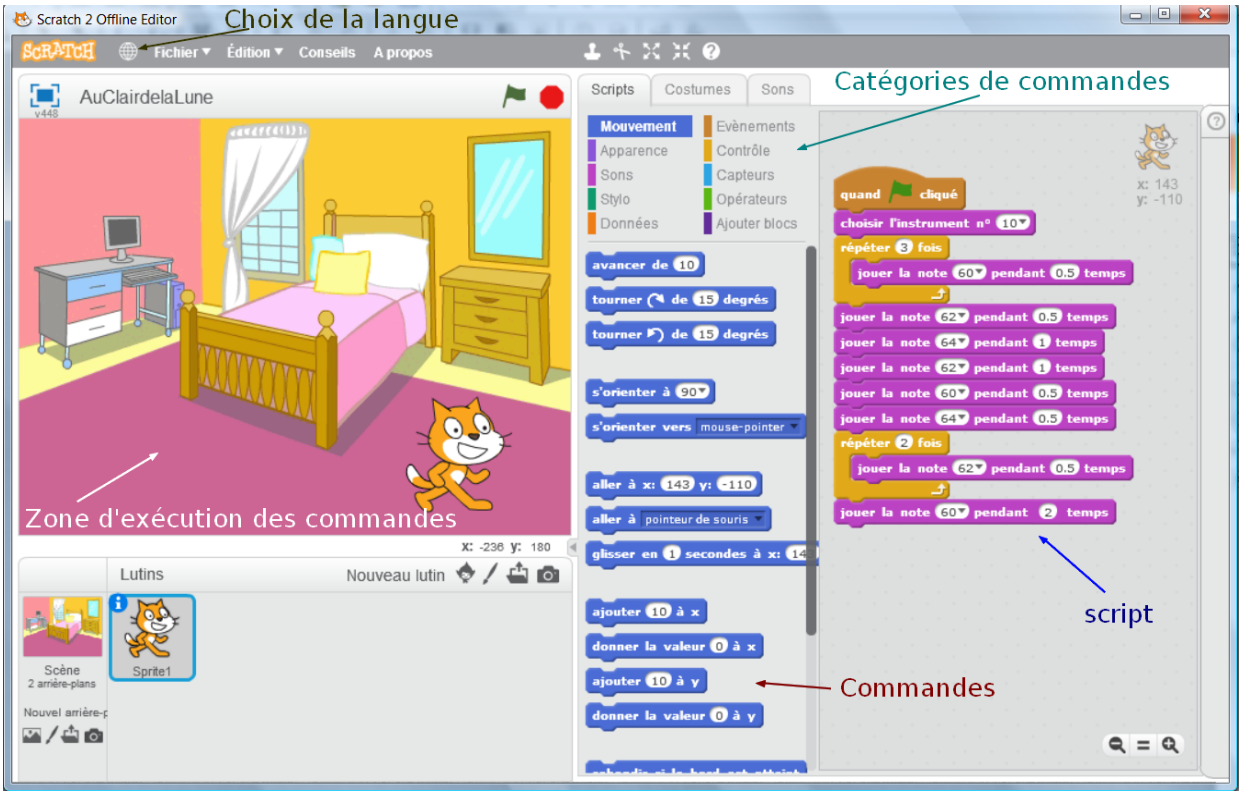
http://scratch.mit.edu/

Téléchargement

https://scratch.mit.edu/scratch2download/

Attention :pour l'utiliser en local nécessite l'installation d'Adobe Air

Interface



3. Python

Ce paragraphe résume la page web http://fsincere.free.fr/isn/python/cours_python.php

Préambule

En 1989, le hollandais **Guido van Rossum** commence le développement du langage de programmation Python.

Python est un langage **multiplateforme**, c'est-à-dire disponible sur plusieurs architectures (compatible PC, tablettes, smartphones,...) et systèmes d'exploitation (Windows, Linux, Mac, Android...).

Le langage Python est gratuit, sous **licence libre**. C'est un des langages informatiques les plus populaires avec C, C++, C#, Objective-C, Java, PHP, JavaScript, Delphi, Visual Basic, Ruby et Perl ...

Que peut-on faire avec Python ?

Beaucoup de choses !

- du calcul scientifique (librairie `NumPy`)
- des graphiques (librairie `matplotlib`)
- du traitement du son, de la synthèse vocale (librairie eSpeak)
- du traitement d'image (librairie `PIL`), de la vision artificielle par caméra (framework `SimpleCV`)
- de la bio-informatique (librairie `Biopython`)
- des applications avec interface graphique GUI (librairies `Tkinter`, `PyQt`, `wxPython`, `PyGTK`...)
- des jeux vidéo en 2D (librairie `Pygame`)
- des applications multi-touch (framework `kivy` pour tablette et smartphone à écran tactile)
- des applications Web (serveur Web `Zope` ; frameworks Web `Flask`, `Django`)
- interfacer des systèmes de gestion de base de données (librairie `MySQLdb`...)
- des applications réseau (framework `Twisted`)
- communiquer avec des ports série RS232 (librairie `PySerial`), en Bluetooth (librairie `pybluez`)...

Des dizaines de milliers de librairies sont disponibles sur le dépôt officiel `PyPI`.

IDLE

IDLE est un environnement de développement intégré pour Python.

(IDE en anglais : Integrated Development Environment)

IDLE propose un certain nombre d'outils :

- un éditeur de texte (pour écrire le programme)
- un interpréteur (pour exécuter le programme)
- un débogueur (pour tester le programme)

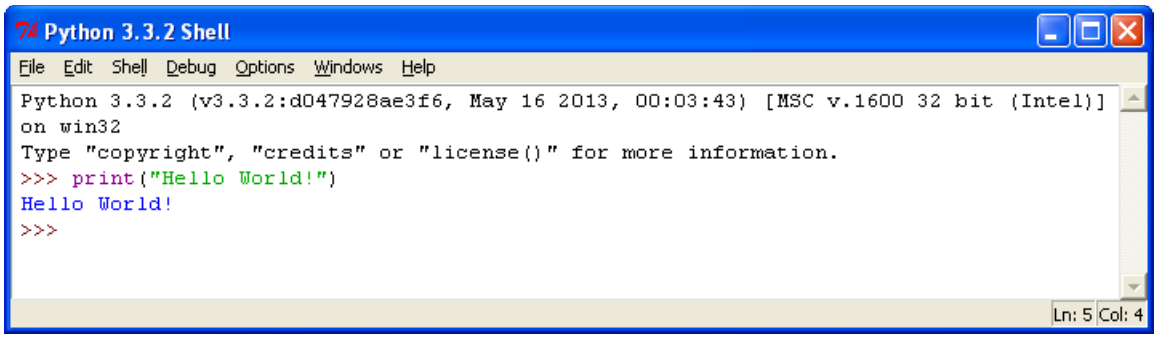
C5T15 – Algorithmique et programmation

Installation

Sous Windows, pour installer Python avec l'**environnement de développement IDLE**, il suffit de télécharger puis d'exécuter le fichier d'installation qui se trouve sur le site officiel : https://www.python.org/downloads/windows

Une fois installé, vous pouvez lancer IDLE en allant dans : Démarrer → Programmes → Python → IDLE (Python GUI)

Interface



Python est pré-installé sur la plupart des distributions Linux.

Scripts

Dans le cas d'un programme en langage Python, on parle souvent de **script Python**.
Un script se présente sous la forme d'un fichier texte avec l'extension **.py**

Exemples

Langage courant	Pseudo-code		Langage de programmation
	Scratch	Langage algorithmique	Python3

choisir le nombre x
 	
ajouter 3
multiplier le résultat par 5
le résultat est 5(x+3)
		lire le nombre x
		
		donner à x la valeur x+3
		donner à x la valeur 5*x
		écrire x
			`s = input('x= ')`
			`x=int(s)`
			`x=x+3`
			`x=5*x`
			`print(x)`

Algorithme qui calcule $5(x + 3)$ pour un nombre donné x et sa traduction en langages Scratch et Python.

C5T15 – Algorithmique et programmation

Objectif 15-3 Les variables

1. Généralités

Une variable est un élément que le programme va repérer par son nom. Lors de l'exécution du programme un espace mémoire lui est associé, espace dans lequel il est possible de stocker la valeur que prend la variable à un instant t.

On classe les variables en 3 catégories :

variables de **type numérique** : nombre entier, nombre réel
variables de **type texte** : caractère, chaîne de caractères
variables de **type booléen** : ne peut prendre que deux valeurs : vrai ou faux

Exemples dans GeoGebra :

Un curseur est une variable de type numérique,

une zone texte est une variable de type texte,

une boîte de sélection des objets à Afficher/Cacher est une variable de type booléen.

Remarques :

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

En langage Python, l'usage est de ne pas utiliser de lettres majuscules pour nommer les variables.

Le nom d'une variable s'écrit avec des lettres (non accentuées), des chiffres ou bien l'underscore "_"

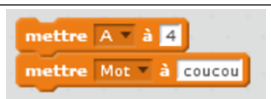
Le nom d'une variable ne doit pas commencer par un chiffre.

2. Affectation

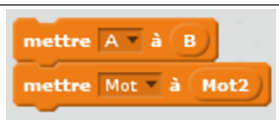
Affecter une valeur à une variable, c'est donner une valeur à cette variable.
Nous utiliserons le symbole ← pour indiquer une affectation en « langage algorithmique ».
Attention : Les langages de programmation utilisent plutôt le signe égal, par exemple lorsqu'on programme une cellule du tableur en écrivant « = suivi d'une expression ».

Exemples

L'instruction d'affectation consiste à « remplir » la « zone mémoire » repérée par le nom de la variable.

Langage algorithmique	Scratch	Python3
A ← 4 Mot ← « coucou »		a=4 mot="coucou"

On peut aussi affecter à une variable la valeur d'une autre variable.

Langage algorithmique	Scratch	Python3
A ← B Mot ← Mot2		a=b mot=mot2

Entrée (ou saisie) de variable. Quand on lit une variable, c'est l'utilisateur qui donne la valeur.

Le programme s'interrompt jusqu'à ce que l'utilisateur frappe la touche « entrée »

Lange algorithmique	Scratch	Python3
lire un nombre A ← nombre		`a=input("entrez un nombre : ")`

3. Agir sur les variables

Pour agir sur les variables, on utilise des opérateurs qui dépendent du type de variables.

numériques : + - * (multiplier) / (diviser) ^ (puissance)

texte : & + (met bout à bout deux chaînes)

logiques (booléen): « et » « ou » « non »

On utilise aussi de nombreuses fonctions prédéfinies comme :

ENT(nombre) renvoie la partie entière du nombre : ENT(2,5) = 2

MOD(entier1,entier2) renvoie le reste dans la division entière de entier1 par entier2 : MOD(11,2) = 1

Exemple

Langage algorithmique	Scratch	Python3

variable x : réel
variable y : réel
lire x
lire y
x ← x+y
x ← x²
écrire x
		`x=float(input("x="))`
		`y=float(input("y="))`
		`x=x+y`
		`x=x*x`
		`print("x=",x)`

Ce programme calcule $(x+y)^2$ pour deux nombres donnés x et y.

C5T15 – Algorithmique et programmation

Objectif 15-4 Les tests

Un **test** permet de choisir une action suivant une condition.

Syntaxes

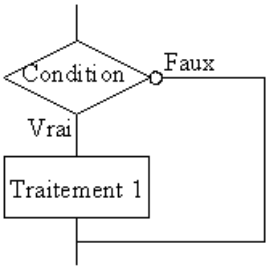
Si « condition est vraie » **alors** traitement1 fin de si

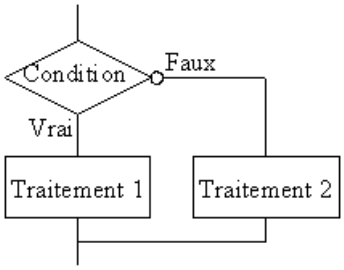
Si « condition est vraie » **alors** traitement1 **Sinon** traitement2 fin de si
(en anglais If ... then ... else ...)

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 » ....

Avec Python, il n'y a pas de « fin de... » mais on utilise l'**indentation** (retrait du texte qui permet de distinguer la partie de programme qui sera exécutée si la condition est réalisée).

Structures d'un test





Exemples

Dire si un nombre A est strictement négatif.

Langage algorithmique	Scratch	Python3

lire A
Si A<0 alors
écrire « A est strictement négatif »
fin de si
		`A=int(input("A = "))`
		`if A<0 :`
		`print(" A est strictement négatif")`

Programme qui demande ton âge en années et te répond si tu es mineur ou majeur.

Langage algorithmique	Scratch	Python3

lire age
Si age<18 alors
écrire « Tu es mineur »
Sinon
écrire« Tu es majeur »
fin de si
		`Age=int(input("Age = "))`
		`if Age<18 :`
		`print("Tu es mineur")`
		`else :`
		`print("Tu es majeur")`

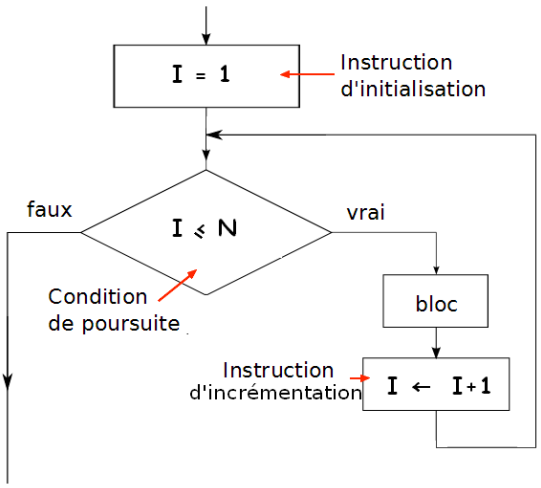
C5T15 – Algorithmique et programmation

Objectif 15-5 Les boucles « POUR » (en anglais boucle For)

Syntaxe d'une boucle « Pour »

Pour (variable variant de ... à ... par pas de ...) faire « bloc d'instructions »

Structure d'une boucle « Pour »



Organigramme de la boucle "Pour I variant de 1 à N par pas de 1 faire ... "

Itération : Définition du mot dans le Larousse :

- Action de répéter, de faire de nouveau, fait d'être répété.
- En informatique, procédé de calcul répétitif qui boucle jusqu'à ce qu'une condition particulière soit remplie.
- Répétition d'un calcul, d'une opération d'un raisonnement.

On connaît N, nombre de fois où le bloc d'instructions devra être exécuté. La boucle s'exécute 1 fois et est répétée (réitérée) N-1 fois.

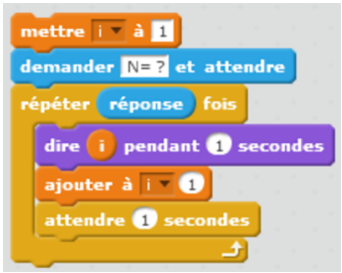
Une fois la ou les répétitions finies, le programme continue.

On doit décrire un «**compteur de boucle**», préciser la valeur de départ, la dernière valeur et le pas (on incrémente de ...).

Exemples

Afficher tous les entiers de 1 à N (donné) Avec Scratch, le lutin « compte ».

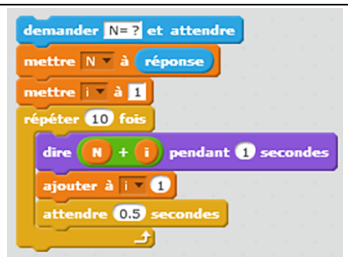
Langage algorithmique	Scratch	Python3

lire N entier
Répéter
pour i de 1 à N :
écrire i
fin de répéter
		`N=int(input("N= "))`
		`for i in range(1,N+1) :`
		`print(i," ",end='')`

Programme qui demande un nombre entier N et affiche tous les nombres de N+1 à N+10.

Avec Scratch, le lutin « compte ».

Langage algorithmique	Scratch	Python3

lire N
Pour i allant de 1 à 10 (pas de 1)
écrire N+i
fin de pour
		`N=int(input("N= "))`
		`for i in range(1,11) :`
		`print(N+i," ",end='')`

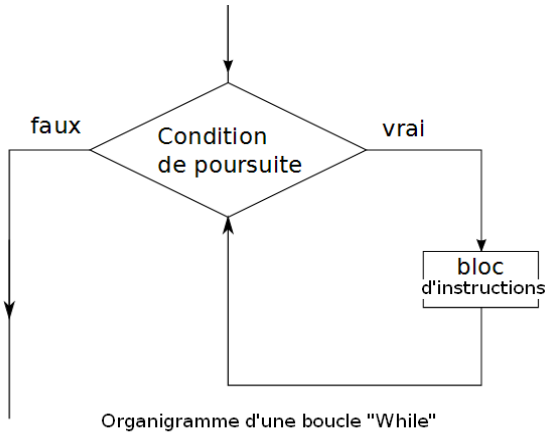
C5T15 – Algorithmique et programmation

Objectif 15-6 Les boucles « TANT QUE » (en anglais boucle While)

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

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

Structure d'une boucle « Tant que »

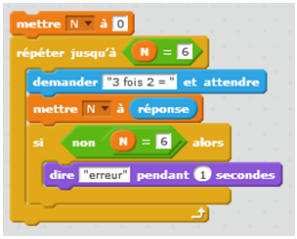


Organigramme d'une boucle "While"

Exemples

Demander « 3 fois 2 ». Lire la réponse N. Tant que N est différent de 6 dire « erreur » .

Langage algorithmique	Scratch	Python3

afficher « 3 fois 2 = »
N ← 0
Tant que N ≠ 6 :
  lire N
  Si N ≠6
  Afficher « erreur »
  fin de si
fin de tant que
		N=0
		while N!=6 :
		  N=float(input("3 fois 2 = "))
		  if (N!=6) :
		  print("erreur")

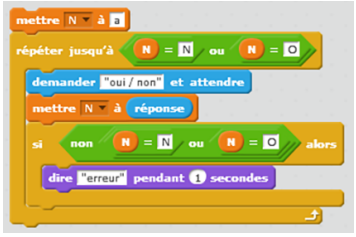
Lire un caractère A et l'afficher, jusqu'à ce que ce soit un « F ».

Langage algorithmique	Scratch	Python3

Répéter :
  lire A
  afficher A
jusqu'à ce que A soit « F »
		A=" "
		while A!="F" :
		  A=input("A= ")
		  print(A)

Programme qui demande une réponse à « oui/non » et n'accepte que « O » ou « N » comme réponse valable. Sinon, il affiche « erreur ».

Langage algorithmique	Scratch	Python3

N ← «A »
Tant que (N n'est pas
  «O » ou «N ») :
  afficher « oui / non »
  lire N
  Si (N n'est pas
  «O »ou «N ») :
  afficher « erreur »
  fin de si
fin du tant que
		N="a"
		while (N!="N") & (N!="O") :
		  N=input(" Oui / Non = ")
		  if (N!="N") & (N!="O") :
		  print("erreur")

C5T15 – Algorithmique et programmation

Objectif 15-7 Les tableaux, les listes

Ce sont des variables particulières. Elles sont utilisées pour stocker plusieurs variables de même type.

Un **tableau** est un ensemble de valeurs portant le même nom de variable et repérées par un nombre appelé **indice**.

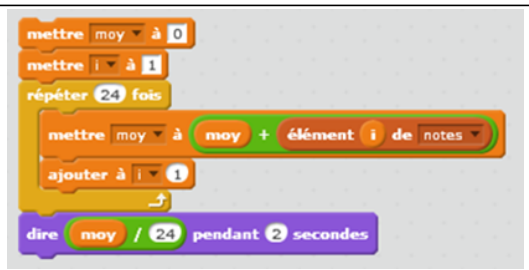
Pour désigner un élément du tableau, on fait figurer le nom du tableau, suivi de l'indice de l'élément, entre crochets.

Attention, dans la plupart des langages de programmation, les indices des tableaux commencent à 0, et non à 1. C'est le cas de Python3. Dans un tableau nommé T, le 1er élément est alors T[0]. Pour Scratch les indices des listes commencent à 1.

Exemple

Algorithme qui calcule la moyenne des notes de 24 élèves, données dans un tableau « notes ».

Langage algorithmique	Scratch	Python3

moy ← 0
Pour i de 1 à 24 :
  moy ← moy +notes[i]
fin de pour
 	
moy← moy/24
écrire moy
		`moy=0`
		`for i in range(24):`
		`  moy=moy+notes[i]`
		`print(notes)`
		`print("moyenne = ",moy/24)`