Algorithme et programmation PYTHON

Types de variables

Dans un algorithme, les variables considérées ont des types qui définissent la nature des valeurs qu'elle peuvent prendre.

▶ les entiers : "int"	$a = 2$; $a = -3$
▶ les flottants : "float"	$a = 2.0$; $a = -1.86$
▶ les chaines de caractères : "str"	$a =$"2" ; $a =$"mot"
▶ les booléens : "bool"	$a = True$; $a = False$

Affectation

L'instruction d'affectation permet d'attribuer une valeur à une variable, ou de la modifier :

En langage naturel	En python
$a \leftarrow 2$	a=2
$a \leftarrow a + 1$	a=$a + 1$

Notion de fonction algorithmique

Une fonction est un bloc d'instructions qui ne sera exécuté que s'il est appelé (éventuellement plusieurs fois). Une fonction possède éventuellement des paramètres et/ou renvoie une valeur de retour (mais pas systématiquement).

En langage naturel	En python
fonction **nom**(p1,p2,...) : *[instructions]* renvoie *valeur*	def **nom** (p1,p2,...) : *[instructions]* return *valeur*

Instructions conditionnelles

Pour exécuter une ou plusieurs instructions uniquement si une certaine condition est vérifiée. Si la condition n'est pas vérifiée, on peut soit ne rien faire, soit exécuter un autre bloc.

En langage naturel	En python
Si *condition* : Alors *[instructions]* Sinon *[instructions]* FinSi	if *condition* : *[instructions]* else : *[instructions]*

Boucles bornées

Lorsque l'on veut répéter un nombre de fois connu un même bloc d'instructions, on utilise une boucle bornée aussi appelée boucle Pour. Ces boucles sont munies d'une variable compteur.

En langage naturel	En python
Pour i allant de 1 à n : *[instructions]* FinPour	for i in range(1 , n+1) : *[instructions]*

Boucles non bornées

Lorsqu'on veut répéter un même bloc d'instructions tant qu'une certaine condition est vérifiée, on utilise une boucle non bornée, aussi appelée boucle Tant que.

En langage naturel	En python
Tant que *[condition]* : *[instructions]* FinTantQue	while *[condition]* : *[instructions]*

Entrée

Pour demander à l'utilisateur d'affecter une valeur à la variable A, on utilise une entrée en tenant compte du type de la variable.

En langage naturel	En python
Saisir A (type string)	A = input() A = input("A=?")
Saisir A (type int)	A = int(input())
Saisir A (type float)	A = float(input())

Sortie

Pour afficher un résultat ou un texte, on utilise une sortie.

En langage naturel	En python
Afficher A	print(A)
Afficher("Un texte")	print ("Un texte")
Afficher("Un texte", A)	print ("Un texte", A)

Modules RANDOM

Si on veut utiliser des tirages aléatoire, il faut importer le module RANDOM : from random import*

En langage naturel	En python
Nombre réel aléatoire dans [0 ; 1[	random()
Nombre entier aléatoire entre a et b	randint(a,b)
Nombre réel aléatoire entre a et b	uniform(a,b)

Modules MATH

Si on veut utiliser des fonctions mathématiques, il faut importer le module MATH : from math import*

En langage naturel	En python
$\sqrt{x}$	$sqrt(x)$
Partie entière de x (arrondi à l'entier inférieur)	$floor(x)$

Tests conditionnels

En langage naturel	En python
$x \overset{?}{=} y$	x==y
$x \overset{?}{\neq} y$	x!=y
$x \overset{?}{\geq} y$	x>=y
et ; *ou* ; *non*	*and* ; *or* ; *not*

Calculs (en Python)

$x + y$; $x - y$; $x * y$; x/y (division décimale)

x**y (calcul x puissance y : x^y)

$abs(x)$ (Valeur absolue de x)

$x//y$ (Quotient entier de x par y)

$x\%y$ (Reste dans la division de x par y)

Région académique Auvergne-Rhône-Alpes
académie Grenoble