Algorithmique

et Python

Plan

Algorithme et Python

Définition

Un algorithme est une succession d'un nombre fini d'étapes réalisées dans un ordre précis et qui a pour but de résoudre un problème.

instructions dans un langage compréhensible par l'exécutant :

- humain $\Rightarrow$ langage courant.
- ordinateur $\Rightarrow$ langage de programmation (Python).
- algorithmique $\Rightarrow$ pseudo-code.

Python

Langage de programmation interprété.
Un programme en Python :

1. écrire texte contenant les instructions
2. demander à Python d'exécuter le programme.
3. Python lit le texte et suit les instructions.
4. Si tout va bien, il va au bout. Si il rencontre une erreur, il s'arrête et renvoi un message d'erreur.

Il faut une distribution Python et un éditeur de code pour l'utiliser. On va privilégier les tout-en-un.

- Sur ordinateur : Installer un logiciel (Spyder ou Idle)
- Sur internet, dans un navigateur :
  https://www.programiz.com/python-programming/online-compiler/
  https://repl.it/languages/python3
- Sur un téléphone ou une tablette : installer une application (Pythonista 3)

Ecrire un programme en Python

Deux fenêtres. Un Editeur (pour taper du code) et une Console (donne le résultat)

1) Editeur. le code = texte contenant les instructions .
Run exécuter le programme. Python lit le texte et suit les instructions.

2) Console. Les résultats ou les messages d'erreurs s'affichent.

En Pyhthon :

- Les commentaires commencent par le symbole #
- Les blocs d'instruction sont identifiés par l'indentation
- Des mots sont réservés à Pyhton :
  if, then, else, while, def, int, string, bool, print, input

Exemple :

Python

```
def Calcul(n):
  if n<2 :
    return 1
  else:
    return (n*9 -2)
a= 5
b=2+a
c =Calcul(b)
print(c)
```

Plan

Définition de variable

Définition

Une variable est un espace de stockage dans un emplacement de la mémoire (de l'ordinateur).

Caractéristiques :

- nom : permet d'identifier la variable
- type : définit quelle donnée peut contenir la variable
- valeur : utilisée par le programme, peut éventuellement changer
- (adresse : l'endroit de l'ordinateur où est stockée la variable.)

Remarque : Python gère souvent tout seul les types de variables

Définition

Déclarer une variable : attribuer un nom et un type à une boite mémoire

Affecter une variable : associer une valeur à la boite

En Python, on fait les deux à la fois.

Exemple : A, b, prix et Total ci-dessous sont des variables. Elles ont des types différents.

Python

```
A= 50
prix = 1.52
Total = A * prix
b= "prix de l'essence"
```

Propriété.

Bonnes pratiques pour nommer une variable :

- Commencer par une lettre minuscule
- choisir un nom explicite pour bien comprendre le programme
- Utiliser des majuscules au milieu du nom _ pour séparer des parties du nom
- éviter les lettres accentuées mais chiffres autorisés

Remarque : majuscules $\neq$ minuscules.

prix $\neq$ Prix

Fonctions utiles pour les variables

- type(.....) donne le type de la variable qu'on met dedans
- print(....) affiche le contenu de la variable.
- print(..., ...,....) imprime plusieurs éléments.
- variable = input("Entrer la valeur de la variable")
  permet de faire entrer au clavier une valeur et la stocke dans la variable.
  Attention, la variable sera de type *string*. Penser à convertir
  n=int(variable) ou n=float(variable)

Exercice

Jean veut faire un programme inversant le contenu de deux variables. Il propose le programme suivant :

Python

```
a=7
b=2
a=b
b=a
```

1. Expliquer pourquoi le programme ne marche pas.
2. modifier le programme pour que la valeur des variables a et b soient inversées.

Plan

Types de nombres

Les nombres peuvent être de différents types :

- int entiers relatifs
- float flottant (nombres décimaux).
  Ils s'écrivent avec un point à la place de la virgule.
- complex Nombres complexes de la forme $a + bj$.
  La partie imaginaire se note avec un j, a et b sont des float.
  obligatoire de mettre un nombre devant le j, même 0 ou 1.

Exemple :

- -6.9 float
- -69 int
- 3 int
- 3.0 float
- 2+5.1j complex
- -9.7 + 0j aussi

Conversion de type

Il est possible de convertir quelque chose en un type de nombre.

- `int(x)` converti x en entier
- `float(x)` converti x en nombre flottant.
- `complex(a,b)` crée le nombre complexe $a + ib$

Exemples :

- int(3.6) $\rightarrow$ 3
- float(5) $\rightarrow$ 5.0
- complex(2.3)$\rightarrow$ 2.3 + 0j
- complex(2.3, 6)$\rightarrow$ 2.3 + 6.0j

Remarque : Les trop petits nombres ($< 10^{-323}$) deviennent 0.
Les nombres trop grands ($> 10^{308}$) renvoient une erreur.

Opérations

- $+$ addition : $3 + 4 \rightarrow 7$
- $-$ soustraction : $5 - 6.7 \rightarrow -1.7$
- $*$ multiplication : $12 * (-4)$
- $/$ division avec résultat à virgule : $81/10 \rightarrow 8.1$
- $//$ quotient de la division euclidienne $81//10 \rightarrow 8$
- $\%$ reste de la division euclidienne $81\%10 \rightarrow 1$
- $**$ Puissance : $2 * *3 \rightarrow 2^3$, on peut aussi utiliser pow(2,3)

Remarque : entiers et flottants compatibles pour les opérations arithmétiques, mais résultat flottant.

Pour faire d'autres fonctions, il faut charger une library mathématique

from math import *

- racine carrée sqrt(...)
- π pi
- e e
- fonction exponentielle exp(...)
- fonction ln log(...)
- fonctions trigonométriques cos(...), sin(...), tan(...)

Raccourci d'opérations

x += a correspond à $x = x + a$

x a été remplacée par $x + a$

ça marche aussi avec

$- =$

$* =$

$/ =$

$** =$

$\% =$

imprécision avec les float

```
x=0.1
y=x+x
print("y est-il égal à
0.2 ?" , y==0.2)
z=x+x+x
print("z est-il égal à
0.3 ?" , z==0.3)
print("z=", z)
```

y est-il égal à 0.2 ? True
z est-il égal à 0.3 ? False
z= 0.30000000000000004

Les float en Python sont stockés d'une manière "non exacte" en mémoire, et Python affiche une valeur approchée.

Exemple : 0.1 est stockée en python avec une valeur de

0.1000000000000000055511151231257827021181583404541015625

Attention test d'égalité avec des float.

Mode fraction

```
from fractions import Fraction
```

- `Fraction(n,d)` $= \frac{n}{d}$ exactement.
- pour convertir un float f en fraction `Fraction('f')`

Mode décimal (plus précis mais plus gourmand en mémoire)

```
from decimal import Decimal
```

Pour écrire un nombre en décimal, `Décimal('nombre')`

Chaine de caractères

Définition

Une chaine de caractère, type string (str), s'écrit entre des "guillemets " (ou des 'guillemets').

Exemple : "Bonjour" est un string
"landfql r gfmk " aussi
"3" aussi

Opérations

- `+` concatène deux chaines de caractère : "chaine1" + "chaine2" donne "chaine1chaine2"
- `"une chaine de caractère"[n]` donne la lettre numéro n de la chaine de caractère, en comptant à partir de 0.
- `\n` dans une chaine de caractère insère un retour à la ligne
- `print("une chaine")` affiche la chaine de caractère
- `len("une chaine")` donne la longueur de la chaine de caractère
- `str(un objet)` convertit un objet en chaine de caractère.

Exemple : "Bonjour" + "le monde" donne "Bonjourle monde"
"Bonjourle monde"[4] donne la lettre o.

Insérer des résultats dans une phrase

Les f-string sont des chaines de caractère avec des places libres prévues pour des variables ou des calculs.

f ” ..... { variable } ..... { fonction(truc) } ..... ”

Booléen

Un booleen `bool` ne peux prendre que deux valeurs

vrai `True` ou faux `False`

Par défaut, la majorité des objets Python sont considérés comme Vrai.

- Les chaines de caractères non vides sont True. Une chaine de caractère vide est False.
- Un nombre non nul est True. Le nombre 0 est False.
- Une liste non vide est True. Une liste vide est False.
- La valeur `None` est False

Les booléens servent à écrire des conditions et des tests.

Exercice

Donner le nom, la valeur et le type des variables ci-dessous

Python

```
A = 5.2
B = \Bonjour"
C =6
D = \Choisir un nombre entier"
E =False
F =A+C
G= C - 10
H =(26 + 4*2 - (12 + 6*3))/2
```