

PROGRAMMATION avec le Langage Python

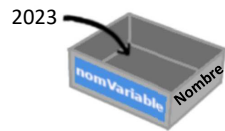
I- Le langage informatique :

Un programme informatique est une suite d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat. Nous utiliserons le langage Python inventé en 1991 par Guido Van Rossum (hollandais). Python est gratuit, sous licence libre, et fonctionne sur plusieurs architectures (PC, tablettes, smartphones, ...)

II- Le b.a.-ba du Python

Les variables

Une variable est un espace mémoire dans lequel il est possible de stocker une valeur. C'est un peu comme une « boîte » sur laquelle il y aurait une étiquette désignant le nom de son contenu. Les algorithmes manipuleront le nom de la variables (nom des boîtes) plutôt que leur contenu.

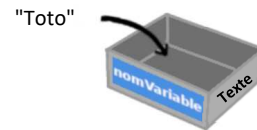


- Le **nom** de la variable correspond à l'étiquette collée sur la boîte.
- La **valeur** de la variable est l'information stockée. Cette valeur peut changer au cours de l'exécution d'un programme.

Les variables peuvent être de différents types. Il faut prendre soin d'adapter le contenu au contenant et vice-versa.

Dans cet exemple le contenu de la variable C sera 25 après l'exécution du programme

```
a = 10
b = 15
c = a + b
```



Les types de bases en Python

Nom français	Nom Python	Désignation	Exemple
Entier	int ou long	Nombre entier (sans virgule)	1 10 20254
Réel	float	Nombre à virgule (nombre réel)	1,1 5425,87458 12,0
Chaîne de caractères	str	Suite quelconque de caractères	"Bonjour" "Ok"
booléen	bool	Peut prendre les valeurs True ou False (vrai ou faux)	True ou False

Il existe plusieurs fonctions en Python qui permettent de forcer le type d'une variable en un autre type :

- int() : permet de modifier une variable en entier.
- float() : permet la transformation en flottant.
- str() : permet de transformer la plupart des variables d'un autre type en chaînes de caractères.

Exemple : Soit x une variable de type chaîne de caractères : x="50"

L'opération x+10 générera une erreur. Pour réaliser une opération arithmétique sur la variable x nous devons la convertir en type entier avec la commande suivante : x=int(x)

→ Réaliser ces tests dans l'ordre avec votre Numworks en mode console Python :

```
>>> x= "50"
>>> x
>>> x+10
>>> x*10
>>> x=int(x)
>>> x
>>> x+10
>>> x*10
```

Les opérateurs d'entrées/sorties en Python

a) Les sorties

Pour permettre au programme en cours d'exécution d'afficher un texte ou un nombre on utilise la commande print().

```
print("Bonjour !") # c'est la chaîne de caractère "Bonjour" qui sera affichée.
a = 3
print(a) # c'est le contenu de la variable a qui sera affiché.
print("Le carrée de", a,"est", a * a) # affichera « Le carrée de 3 est 9 »
```

b) Les entrées

Il est souvent nécessaire de donner une valeur en utilisant le clavier. On utilise alors la commande Input().

```
nom = input("Quel est votre nom ?")
#nom contiendra la chaîne de caractère saisie au clavier
```

Input() est une fonction **qui renvoie toujours une chaîne de caractères**. Il est donc parfois nécessaire de changer le type de la variable rentrée.

```
n = input("Entrer un nombre") # n est une variable de type String
n = int(n) # n devient une variable de type entier
print("Le carrée de", n,"est", n * n)
```

Les opérateurs de base en python

+ addition
- soustraction
* multiplication
/ division 5/2 donne 2.5
// division entière 5//2 donne 2
% reste de la division entière 5%2 donne 1 -> Numworks : fmod(5,2)
** puissance 2**10 donne 1024

== égalité (à ne pas confondre avec l'affectation)
!= différent
<, >, <=, >= inférieur, supérieur, inférieur ou égal, supérieur ou égal
and opérateur booléen ET
or opérateur booléen OU
not opérateur booléen NON

→ Tester (en mode console sous Python) la valeur d'une variable contenant un nombre.

Soit n une variable contenant le nombre 12 (**faire n=12** avant de les tests).

Test en français	Écrit en Python 3	Renvoie quoi ?
n est-il égal à 12 ?	n==12	
n est-il égal à 10 ?	n==10	
n est-il positif ?	n>0	
n est-il différent de 10 ?	n!=10	
Si n est compris strictement entre 0 et 20	(n>0) and (n<20) ou alors 0<n<20	

Les structures algorithmiques fondamentales

a) La structure "SI ALORS SINON"

Nous utilisons cette structure dès que nous voulons exprimer une possibilité de choix simple à deux alternatives. C'est une des opérations les plus utilisées, nous pourrions tout décrire avec des opérations simples et des « SI ALORS SINON ».

Exemple :

```
if note_ds >= 10:
    print("Vous avez la moyenne")
else:
    print("Vous n'avez pas la moyenne")
```

Le bloc else: est optionnel

On peut aussi « imbriquer » les if avec l'alternative elif : (la contraction de alors si)

b) La structure de boucle "REPETER TANT QUE" (WHILE)

Dans cette structure on commence par tester une condition. Si elle est vérifiée, le traitement est exécuté.

Exemples :

Test d'un mot de passe

```
mot_de_passe=""

while mot_de_passe!="bidule":
    mot_de_passe=input("mot de passe svp : ")

print("le mot de passe est validé")
```

Écrit la table de multiplication de 7

```
i=1
while i<11:
    print(i, "x 7 =", i*7)
    i=i+1
```

c) La structure de boucle "POUR ... DE ... A ..." (FOR)

Lorsque l'on souhaite répéter un nombre donné de fois la même instruction ou le même bloc d'instructions, la commande for est la plus appropriée.

Exemples :

Écrit tous les nombres de 0 à 99 inclus

```
for i in range(0,100):
    print(i)
```

Écrit la table de multiplication de 7

```
for i in range(1,11):
    print(i, "x 7 =", i*7)
```

Applications

→ Programmer en python une application demandant le prénom d'un utilisateur puis son âge afin d'écrire le compte-rendu suivant :

« Bonjour Jojo, tu es donc né(e) en 2007 » *(si la personne s'appelle Jojo et qu'elle a 16 ans).*

→ Ajouter au programme précédent un affichage mentionnant si l'utilisateur est majeur ou mineur.
« Bonjour Jojo, tu es donc né(e) en 2007, tu es mineur(e) » *(si la personne s'appelle Jojo et qu'elle a 16 ans).*

→ Donner un programme capable de saisir la taille puis le poids d'une personne afin d'afficher son IMC accompagné d'un bilan.

Pour information :

L'IMC permet de savoir si votre poids est idéal, autrement dit, s'il est adapté à votre taille. Il correspond au poids divisé par le carré de la taille (IMC = poids en kg/taille² en m).

Voici le bilan :

- *IMC < 18,5 kg/m² : insuffisance pondérale*
- *18,5 < IMC < 24,9 : poids normal*
- *25 < IMC < 29,9 : surpoids*
- *IMC > 30 : obésité*