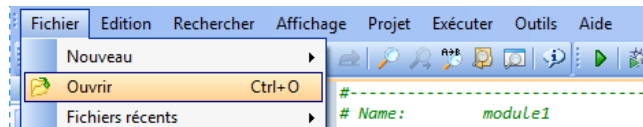


LE LANGAGE PYTHON

Utiliser « PyScripter-Portable » : l'utilisation est très simple :

1. On ouvre un fichier (ouverture ou création d'un nouveau fichier avec le menu déroulant)



2. On écrit son script
3. On le sauvegarde (« CTRL S » ou « enregistrer » ou « enregistrer sous » dans le menu déroulant)

Remarques :

1. Le caractère # ouvre une ligne de commentaire qui ne sera pas lue par le programme
2. Chaque appel de « run » remet la mémoire à zéro (affichage de RESTART)
3. S'il y a un problème de syntaxe, alors on aura un message d'erreur

L'affectation de variables :

1. Règles du choix des noms de variables :
 - 33 mots réservés interdits
 - Pas de caractères spéciaux sauf « _ » (underscore)
 - Doivent commencer par une lettre
2. Affectation
 - Se fait avec le symbole = (pour une définition ou une modification)

```
>>> x=5
```

```
>>> phrase= " bonjour tout le monde "
```
 - Le nom est toujours en premier
 - Une affectation multiple est possible

```
>>> x,y=2.4,8
```

Les types de variables :

En python, pas besoin de déclaration de type pour les valeurs, le typage se fait automatiquement (typage dynamique)

Exemple :

```
>>> x,y,ph=4,15.3,'bonjour'
>>> print(type(x))
>>> print(type(y))
>>> print(type(ph))
```

Pour les nombres on utilisera :

Integer : pour les entiers
Float : pour les réels

Quelques fonctionnalités :

Puissance :

$x^{**}n$ ou `pow(x,n)`

```
>>> 4**3
```

```
64
```

```
>>> 4**0.5
```

```
2.0
```

Reste de la division euclidienne :

% renvoie le reste de la division euclidienne : Si $a=bq+r$ alors `a%b` renvoie `r`

```
>>> 24%5
```

```
4
```

Division entière :

// effectue une division "entière"

C'est-à-dire un quotient : si $a=bq+r$ alors `a//b` renvoie `q`

```
>>> 9//2
```

```
4
```

```
>>> 9/2
```

```
4.5
```

« Dialoguer » avec la machine : entrée/sortie

- ✓ Pour les « sorties » : on utilise **print**

print permet l'affichage de données (n'importe lesquelles). C'est une fonction donc il y aura des parenthèses.

print n'affiche pas les guillemets ou apostrophes "externes" d'une chaîne de caractères

```
>>> x='bonjour'
```

```
>>> print(x)
```

```
Bonjour
```

- ✓ Les « entrées » : il faut pouvoir saisir une réponse donnée par l'utilisateur. Pour cela on utilise **input**

Input(chaîne) affiche chaîne et attend une entrée clavier. Cette fonction renvoie une chaîne de caractères contenant la donnée saisie par l'utilisateur.

Exercice n°1 :

Écrire un petit programme qui affichera ce « dialogue »

Bonjour

Quel est votre prénom ? Françoise

Comment allez-vous françoise ?

On va faire un calcul

Entrez un nombre : 6

Entrez un autre nombre : 7

La somme de 6 et 7 est égale à 13

Syntaxe de la structure de contrôle : si ... alors

```
If <test1> :  
    <bloc instructions1>  
else :  
    <bloc instruction2>
```

OU

```
If <test1> :  
    <bloc instructions1>  
elif :  
    <bloc instruction2>  
else :  
    <bloc instruction3>
```

À remarquer :

- ✓ Les : en fin de ligne
- ✓ L'indentation : else et elif doivent être alignés avec if.
- ✓ Les « blocs » d'actions sont aussi alignés.
- ✓ Le test « a égale b » se note : $a==b$ et « a différent de b » se note $a!=b$

Les fonctions :

On peut créer des fonctions (avec ou sans paramètres) pour appeler des programmes

Syntaxe :

```
def nom_fonction(arg1,arg2,...) :  
    bloc d'instructions
```

arg1, arg2, ... sont appelés les paramètres (ou arguments) de la fonction

Exemple :

```
def aire_rect(long,larg) :  
    print (long*larg)
```

```
>>>aire_rect(10,4)  
40
```

Exercice n°2 : Pythagore

Le but est de créer un programme qui affiche si un triangle est rectangle ou non après avoir rentré la longueur de ses trois côtés.

1. Écrire la fonction maxi(a,b,c) qui renvoie le plus grand de 3 nombres.
2. Écrire la fonction mini(a,b,c) qui renvoie le plus petit de 3 nombres.
3. Écrire la fonction moyen(a,b,c) qui renvoie le nombre médian parmi 3 nombres.
4. Écrire la fonction pythagore() qui demande à l'utilisateur de rentrer les longueurs des trois côtés d'un triangle et qui affiche si le triangle est rectangle ou pas.

Les boucles :

En python, nous verrons 2 manières d'écrire des boucles :

for : C'est un parcours d'une séquence (chaîne, liste, séquence d'entiers, ...)

while : C'est la boucle « tant que » : on peut tout écrire avec mais le for peut être plus rapide à mettre en place.

<pre>for x in range(1,5) : print(x)</pre> <p>va afficher :</p> <p>1 2 3 4</p>	<pre>for x in range(2,13,3) : print(x)</pre> <p>va afficher :</p> <p>2 5 8 11</p>
---	---

Exercice n°3 : Moyenne de notes

1. Version 1

Écrire un programme qui demande à l'utilisateur de saisir un entier qui correspond au nombre de valeurs qu'il saisira, puis de saisir les valeurs en question. A la fin, le programme affiche la moyenne des valeurs saisies.

Exemple :

Bonjour, combien de valeurs voulez-vous saisir? 5

valeur 1 : 5

valeur 2 : 15

valeur 3 : 10

valeur 4 : 12

valeur 5 : 8

la moyenne des valeurs saisies est 10.000000

2. Version2

On reprend la saisie de valeurs mais sans demander au préalable le nombre de valeurs à saisir. Le programme demande des valeurs à l'utilisateur jusqu'à ce que l'utilisateur tape la valeur "-1". Le programme affiche alors le nombre de valeurs saisies et leur moyenne.

Exemple :

première valeur ? 5

valeur suivante? -1 pour stopper 10

valeur suivante? -1 pour stopper 15

valeur suivante? -1 pour stopper 8

valeur suivante? -1 pour stopper 12

valeur suivante? -1 pour stopper -1

la moyenne des 4 valeurs saisies est 11.250000