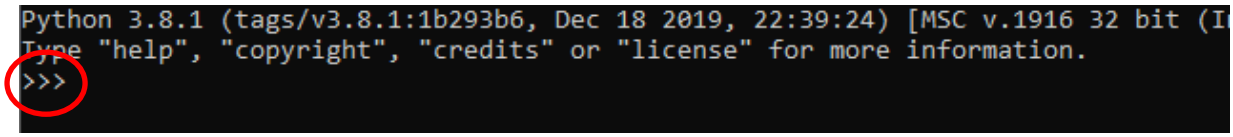
Python : Chaine de caractères, calculs, variables et listes

Dans le chapitre précédent : 2de-Python-introduction, nous avons expliqué comment installer Python (la version utilisée est la version 3.8)

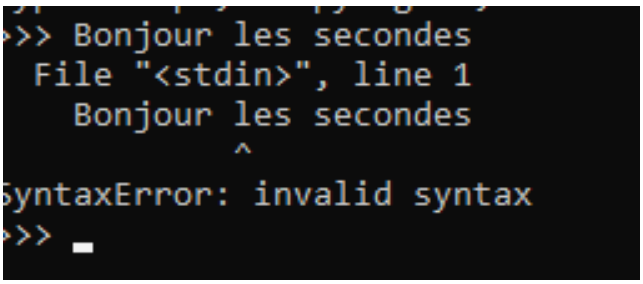
I) Chaine de caractères et commencement d' un programme

Lorsque la console Python est active, il apparait à l'écran :



Le symbole >>> signifie que le logiciel Python attend une commande, une instruction.

Si on écrit directement un texte comme par exemple : Bonjour les secondes, écrit ainsi, il apparait une erreur « invalid sythax »



Python ne comprend pas ce qui lui est demandé. La syntaxe est très importante.

Une chaine de caractère est composée d'un ou plusieurs caractères. Pour écrire un texte dans le langage Python on écrit le texte entouré de guillemets simples ou doubles.

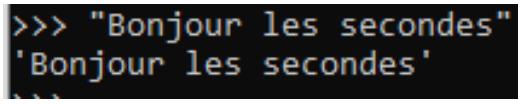
Si nous voulons écrire par exemple : *Bonjour les secondes*, on l'écrit soit :

'Bonjour les secondes' ou

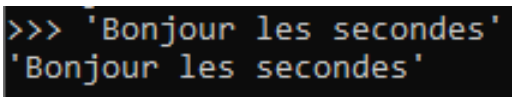
« Bonjour les secondes »

On peut aussi utiliser la commande print(« texte »)

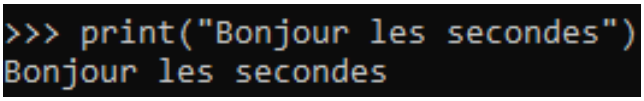
Exemple : print(« Bonjour les secondes »)



ou



ou



II) Calculs mathématiques

Python sait calculer :

On peut écrire un calcul directement, Python calcule le résultat

Si on tape : >>> 2+3 la ligne suivante sera :

5

Les symboles + ; - ; × et ÷ s'écrivent respectivement en langage Python :

+; - ; * ; /

La puissance a pour symbole dans python **

Exemple 25 s'écrira 25**

```
>>> 2+3
5
>>> 7*5+2-4
33
```

Dans une division euclidienne de a par b

a%b est le reste

a//b est le quotient

Exemple :

>>> 5/2 *(Python fait la division décimale de 5 par 2)*
2.5

>>>5//2 *(Python calcule le quotient de la division euclidienne de 5 par 2)*
2

>>> 5%2 *(Python calcule le reste de la division euclidienne de 5 par 2*
1

>>> 53** *(Python calcule 5^3)*

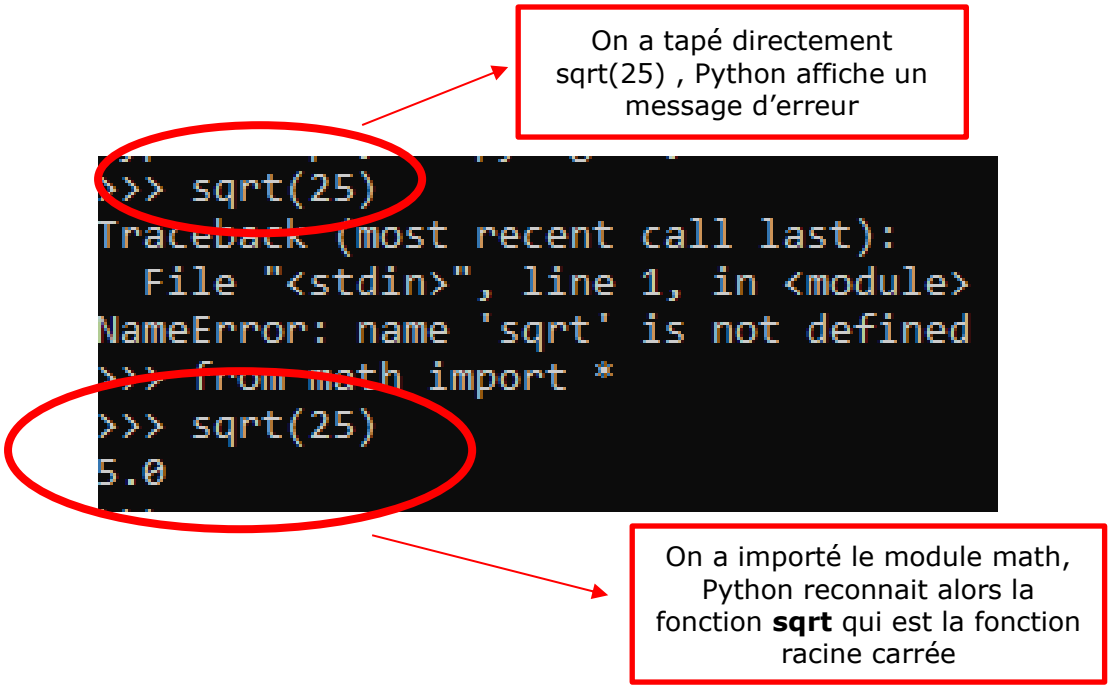
125

Lorsque l'on veut accéder à certaines fonctions mathématiques on importe le module maths : On tape alors dans la console Python :

from math import *

Ensuite on peut utiliser la fonction mathématique voulue comme par exemple la racine carrée qui se note sqrt

Attention si on n'importe pas le module maths, Python ne reconnaitra pas la fonction demandée. Par exemple si on tape sqrt(25) sans avoir au préalable taper « from math import * » , Python ne comprendra pas ce qui lui est demandé.



III) Variable et affectation

1) Affectation de variables

Dans les programmes on doit stocker des données que l'on utilisera par la suite.

Prenons l'exemple où l'on veut que la variable a prenne la valeur 20 :

En langage algorithmique on dit : on affecte à a la valeur 20. On le note **a ← 20**

En langage Scratch on le programme :

En langage Python on le programme : a = 20

Attention le signe = n'a pas la même signification qu'en langage mathématiques , c'est une affectation.

a) Différents type de variables

Une variable peut contenir des données de différentes natures que l'on appelle type de variable qui peuvent être :

Des nombres entiers (entiers relatifs)

Des flottants (les nombres réels, en python la virgule décimale est représentée par un point.

Des chaines de caractères (des mots)

Des booléens (ce sont des variables qui peuvent prendre deux valeurs : true ou false c'est-à-dire vrai ou faux. On les retrouve souvent avec les opérateurs de comparaison. Est-ce plus grand que ? plus petit que ? est-ce égal à ? etc ….

L'affectation est l'instruction qui affecte une valeur à une variable qui est stocké, qui est noté par une flèche : ↤

Exemples

a ↤ bonjour : veut dire que la chaine de caractère « bonjour » est stocké dans la variable a

b ↤ 2 : veut dire que la nombre entier 2 est stocké dans la variable b

Pour arrondir des flottants

Si on veut arrondir un flottant au nombre donné de chiffres après le point décimal on utilise la fonction round()

Exemples :

round(3.1415926,2) donne comme réponse 3.14

round(2.14874,3) donne comme réponse 2.149

b) Variables en langage Python

Nommer une variable

En langage Python, les deux règles principales pour nommer une variable sont :

- Le nom commence par une lettre ou des underscores (tirets bas (_)

- la variable doit contenir uniquement des lettres, des chiffres ou/ et des underscores (tiret bas (_))mais attention sans espace, sans caractères spéciaux, ni accentués (sans accent)

Exemples : la2r ; m348z ; _banane_12 peuvent être des noms de variables. 2la ne peut pas être un nom de variable car le nom commence par un chiffre.

Des bons nombres de variables doivent être courts mais descriptifs :

Exemples :

Prenom = 'Clara'

Nom_famille = 'Parfenoff'

age=48

Evitez les accents é ; è ; ï etc.. , les espaces et les caractères spéciaux.

Affecter un type de variable à une variable

-Pour afficher la valeur contenue dans une variable, on tape le nom de la variable puis tapez sur la touche entrer.

```
>>> m3=12
>>> n="Bonjour"
>>> n
'Bonjour'
>>> m3
12
```

- **Pour affecter un type de variable à une variable on utilise le signe =**

L'instruction m3 = 12 signifie que la valeur 12 est affectée à la variable m3

- Le type de variable, peut être vérifié sous Python en utilisant la fonction type().

```
>>> type(m3)
<class 'int'>
>>> type(n)
<class 'str'>
>>>
```

- **En langage Python :**

Les nombres entiers se notent « int »

Les nombres réels se notent « float »

Les chaines de caractères se notent ; « str »

Les booléens se notent « bool »

Comment insérer la valeur d'une variable dans une chaine de caractère :

Tout d'abord on affecte à la variable un type de variable.
Ensuite on utilise la fonction print (on met entre guillemets le texte que l'on veut écrire suivi d'une virgule suivie du nom de la variable)

Exemple 1:

nom = « Enstein »
print(« Bonjour »,nom)
il s'affichera : Bonjour Enstein

```
>>> nom="Enstein"
>>> print("Bonjour",nom)
Bonjour Enstein
```

Exemple 2:

nombre_choisi=18

print (« le nombre que j'ai choisi est : »,nombre_choisi)

Il s'affichera : le nombre que j'ai choisi est 18

```
>>> nombre_choisi=18
>>> print("le nombre que j'ai choisi est:",nombre_choisi)
le nombre que j'ai choisi est: 18
```

Exemple 3 : Programmer en langage Python, l'algorithme suivant :

Affecter à n la valeur 5

Affecter à m la valeur n-1

Affecter à p la valeur n+5

Affecter à q la valeur 2p

Affecter à r la valeur p-q

Programme :

```
>>> n = 5

>>> m = n-1

>>> p = n+5

>>> q =2*p

>>> r = p-q
```

```
>>> n=5
>>> p=n+5
>>> q=2*p
>>> r=p-q
>>>
```

Si on arrête le programme là , les valeurs de m ; p ; q et r sont calculés mais lors de l'exécution du programme **les valeurs des variables ne seront pas affcihées,** comme nous pouvons le voir ci-dessus.

Pour qu'elles le soient il faut taper le nom de la variable suivi d'entrer :

```
>>> n=5
>>> m=n-1
>>> p=n+5
>>> q=2*p
>>> r=p-q
>>> n
5
>>> m
4
>>> p
10
>>> q
20
>>> r
-10
>>>
```

On peut aussi rajouter au programme :

print(n,m,p,q,r)

la réponse s'affiche :

5 4 10 20 -10

```
>>> n=5
>>> m=n-1
>>> p=n+5
>>> q=2*p
>>> r=p-q
>>> print(n,m,p,q,r)
5 4 10 20 -10
```

Si on veut une réponse plus claire :

n = 5 (la valeur de n)

m = 4 (la valeur de m) etc … dans ce cas on affiche :

print(« m= »,m, « n= »,n, « p= »,p,« q= »,q, «r= »,r)

« m= » pour qu'il affiche explicitement m= suivi de sa valeur m

```
>>> n=5
>>> m=n-1
>>> p=n+5
>>> q=2*p
>>> r=p-q
>>> print(n,m,p,q,r)
5 4 10 20 -10
>>> print("n=",n,"m=",m,"p=",p,"q=",q,"r=",r)
n= 5 m= 4 p= 10 q= 20 r= -10
```

Vous venez de voir toutes les façons de programmer pour que les valeurs des variables 3 ; n ; p ; q et r soient calculées et affichées.

IV) Commentaires

On peut inclure un commentaire dans son programme.

Un commentaire est une ligne de texte ignorée par l'interpréteur Python.

Les commentaires nous permettent d'écrire des notes à l'intérieur du programme. Ils doivent être précédés du signe #. Ces notes ne seront donc pas interprétées.

Exemple : # Ce programme permet de calculer une racine carrée

```
>>> # ce programme permet de calculer une racine carrée
```

V) Les listes

1) Définitions

Une liste contient des données de différents types.

Une liste se note entre crochets avec une virgule comme séparateur.

Exemple : si on veut affecter à a la liste : 3 ; 5 ; 7 ; 9

En langage Python , on l'écrit :

a = [3,5,7,9]

```
>>> a=[3,5,7,9]
>>>
>>> a
[3, 5, 7, 9]
>>>
```

Le premier élément d'une liste est 0, si on prend l'exemple précédent, la liste de a est : a= [3,5,7,9]

Au rang 0 le nombre est 3, pour avoir ce nombre on tape en langage Python : a[0]

Au rang 1 le nombre est 5, pour avoir ce nombre on tape en langage Python : a[1]

Au rang 2 le nombre est 7, pour avoir ce nombre on tape en langage Python : a[2]

Au rang 3 le nombre est 9, pour avoir ce nombre on tape en langage Python : a[3]

```
>>> a=[3,5,7,9]
>>> a[0]
3
>>> a[1]
5
>>> a[2]
7
>>> a[3]
9
```

2) Longueur d'une liste

La longueur d'une liste est l'instruction : len

Reprenons l'exemple précédent : a=[3,5,7,9]

Sa longueur est 4 car cette liste contient 4 éléments

On écrit : len(a) pour que le programme Python détermine sa longueur.

```
>>> a=[3,5,7,9,]
>>> len(a)
4
```

3) Ordonner une liste dans l'ordre croissant

Pour ordonner une liste dans l'ordre croissant on utilise : nom. Sort()

Exemple : b=[9,5,1,8]

Il faut taper l'instruction : b.sort() puis tapez la touche entrée puis on écrit le nom de la liste c'est-à-dire b dans notre exemple .Le résultat sera alors :

1,5,8,9

```
>> b=[9,5,1,8]
>> b.sort()
>> b
1, 5, 8, 9]
```

4) Fonction range

La fonction range est la création d'une liste où l'on donne le premier nombre inclus, dernier nombre non inclus, ainsi que le pas) écris de la façon suivante :

range (premier nombre inclus, dernier nombre non inclus, la pas).

Exemples :

a=range(2,10,3)

Le pas est 3 donc on va de 3 en 3 en commençant par le premier nombre 2 en s'arrêtant à 10 non compris.

a = range(2,10,3)

a=[2 ,5,8] on s'arrête à 8 car 8+3= 11 on dépasse 10

b= range(0,10,2)

Le pas est 2, on va de 2 en 2 en partant du premier nombre 0 et on s'arrête à 10 non compris :

b=range(0,10,2)

b=[0,2,4,6,8]

On peut écrire aussi range (premier nombre, deuxième nombre non inclus) dans ce cas le premier nombre est le premier de la liste inclus, le dernier est le dernier de la liste non inclus, le pas n'étant pas précisé il sera automatiquement de 1.

Exemple :

c= range (3,6) alors c= [3, 4, 5] le pas sera de 1automatiquement

d= range(7,12) alors d=[7,8,9,10,11]

On peut écrire aussi range (nombre), ce nombre est alors le dernier nombre non inclus, le premier étant automatiquement 0 et le pas est 1.

e=range (6) alors e= [0,1,2,3,4,5], s'il n'y a qu'un seul nombre alors par défaut le premier nombre est , la pas est et on va jusqu'à non compris.

f=range(9) alors f=[0,1,2,3,4,5,6,7,8]

Cette fonction est beaucoup utilisée dans les boucles un des chapitres suivants.

Exemples :

a=range(4) , il n'y a qu'un seul nombre donc c'est forcément le dernier nombre non inclus le pas est 1 et le premier nombre 0 donc **a = [0,1,2]**

b = range(6,9) , il y a deux nombres donc le pas est 1 le premier nombre 6 et on s'arrête à 9 non inclus donc **b = [6,7,8]**

c = range(6,12,3) dans ce cas le premier nombre est 6 le dernier non inclus est 12 et le pas est 3 donc **c=[6,9]** **12 ne peut être pris.**