Chapitre 1 - Bases pour débuter avec Python

On voit dans ce chapitre comment créer des premiers codes Python.

1- UN PREMIER CODE PYTHON

Code python :

```
1   # Calcul de mon age en 2050
2   anneeNaissance = 2007
3   age = 2050 - anneeNaissance
4   print("\n En 2050 j'aurai : ",age,"ans \n")
```

Résultat de l'exécution :

```
En 2050 j'aurai :  43 ans
```

Analyse de ce code :

Ce code est écrit en langage Python et est composé de 4 lignes.

Chaque ligne **est exécutée l'une après l'autre** en allant du haut vers le bas :

```
# Calcul de mon age en 2050

anneeNaissance = 2007

age = 2050 - anneeNaissance

print("\n En 2050 j'aurai : ",age," ans \n")
```

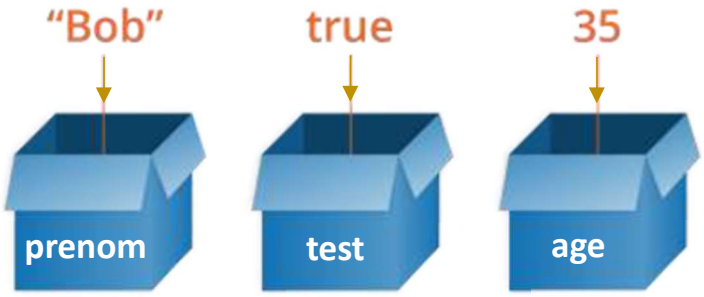
2- LES VARIABLES

Exemple de 3 variables :

Codes python qui les crée :

```
prenom = "Bob"
test = True
age = 35
```

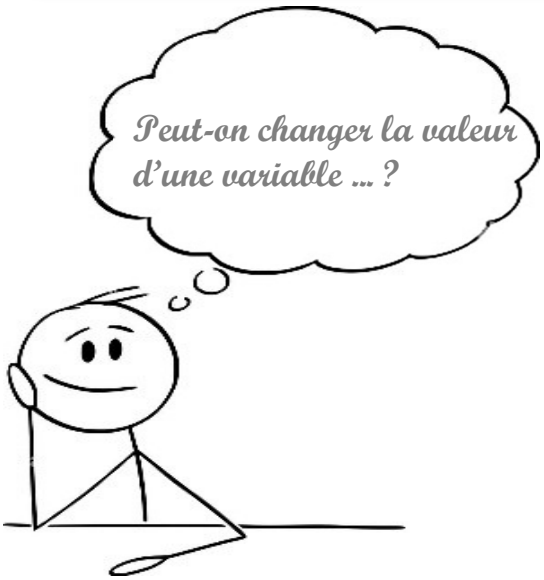
Représentation imagée :



VARIABLE : Le **nom** de la variable fait référence à un emplacement dans la mémoire de l'ordinateur. La **valeur** de la variable est égale au contenu que l'on met dans cet espace.

TYPE d'une variable :

Le **type** d'une variable correspond à la nature de celle-ci. Les quatre principaux types dont nous aurons besoin dans un premier temps sont les entiers (integer ou int), les nombres décimaux que nous appellerons floats, les chaînes de caractères (string ou str) et les booléen (true et false).



Codes python qui modifie 2 variables :

```
prenom = "Boby"
age = age + 50
```

Représentation imagée :

3- LES CHAINES DE CARACTERES (STRING EN ANGLAIS)

String : Une chaine de caractère est encadrée par des guillemets simples ou doubles

Code Python qui crée une variable	Commentaires
`a = "Kylian Mbappé"`	
`b = type("Kylian Mbappé")`	
`c = "Kylian Mbappé\n"`	
`d = 'Kylian Mbappé'`	
`e = "Dans 2 ans, j'aurai 18 ans"`	
`f = 'Dans 2 ans, j"aurai 18 ans'`	
`g = 'Dans 2 ans, j'aurai 18 ans'`	
`h = type("2023")`	
`i = "2023" + "1"`	
`j = "2023" + 1`	
`k = "2023" + str(1)`	

Concaténation :

Lorsque l'on écrit `"2023" + "1"` , on réalise une concaténation de 2 chaines de caractères. On obtient la chaine de caractère `"20231"`

Fonctions natives de Python utiles :

Fonction str() : `str(1)` renvoie `"1"`

Fonction type() : `type("Kylian Mbappé")` renvoie `<class 'str'>`

4- LES NOMBRES

Il existe 2 types de nombres. Les nombres **entiers relatifs** qui appartiennent à la famille des « *integer* » ou en abréviation « ***int*** » et **les autres** qui appartiennent à la famille des décimaux ou « ***float*** ». Les entiers occupent un espace mémoire plus petit que les décimaux. Autre avantage des entiers, leur conversion en binaire est exacte, contrairement aux décimaux pour lesquels la conversion est souvent approchée. On voit ça de suite :

<table>
<tr><th>Code Python</th><th>Code Python</th></tr>
<tr><td>

```
a = 2023
print(type(a))
```

</td><td>

```
print(a+b)
print(type(a+b))
```

</td></tr>
<tr><td>

```
b = 2023.0
print(type(b))
```

</td><td rowspan="2">

```
d = 3.99 + 0.11
print(d)
```

</td></tr>
<tr><td>

```
c = -2023
print(type(c))
```

</td></tr>
<tr><td>

```
e = int("2023")
print(type(e))
```

</td><td>

```
e = float("2023")
print(type(e))
```

</td></tr>
</table>

Fonctions natives de Python utiles pour convertir :

Fonction int() : `int("2023")` renvoie l'entier `2023`

Fonction type() : `float("2023")` renvoie le décimal `2023.0`

Des opérations mathématiques, propres aux entiers, sont disponibles :

<table>
<tr><th>Code Python</th><th>Code Python</th></tr>
<tr><td>

```
a = 11/2
```

</td><td>

```
c = 11%2
```

</td></tr>
<tr><td>

```
b = 11//2
```

</td><td>

```
d = 12%2
```

</td></tr>
</table>

Opérations mathématiques propres aux entiers :

- Quotient d'une division euclidienne : `9 // 4` est égal à `2`
- Reste d'une division euclidienne : `9 % 4` est égal à `1`

5- LES BOOLEENS

Les booléens sont un type de variables à 2 états *Vrai* ou *Faux* (en anglais ***True*** ou ***False***) . Ils sont souvent utilisés pour les tests. On voit cela de suite avec les variables booléennes nommées ci-dessous a, b, c, d, e, f :

Code Python	Code Python
`a = 9>4`	`d = 4!=(2+2)`
`b = 9==(1+8)`	`e = 9 > 4 and 2 == (1+1)`
`c = 9==(4+1)`	`f = 1!=7 or 1==7`

Opérateurs de test :

- Test d'égalité entre 2 variables : `==`
- Test de différence : `!=`
- Test d'inégalités : `>` ou `<`

6- FONCTION NATIVE DE PYTHON INPUT()

La fonction *input()* permet de gérer des saisies d'utilisateur au clavier. On voit cela tout de suite :

Code Python qui crée une variable	Ce que cela donne dans le terminal

`a = input("Saisis ton age : ")`
`print(a)`
`print(type(a))`
	`Saisis ton age : 16`
	`16`
	`<class 'str'>`

`a = input("Saisis ton age : ")`
`a = int(a)`
`print(a)`
`print(type(a))`
	`Saisis ton age : 16`
	`16`
	`<class 'int'>`

`a = float(input("Saisis ton age : "))`
`print(a)`
`print(type(a))`
	`Saisis ton age : 16`
	`16.0`
	`<class 'float'>`

La fonction *input()* renvoie une chaine de caractère

7- FONCTION NATIVE DE PYTHON PRINT()

La fonction *print()* permet de gérer les affichages dans le terminal. Il y a généralement 3 façons d'utiliser cette fonction native de python :

Par exemple, les 3 codes donnés ci-dessous, permettent d'afficher exactement la même phrase :

```
Dans 24 heures, il y a 86400 minutes
```

```
a = 24
b = 3600
print("Dans",a,"heures, il y a",a*b,"minutes")
```

```
a = 24
b = 3600
print(f"Dans {a} heures, il y a {a*b} minutes")
```

```
a = 24
b = 3600
message = "Dans "+ str(a) + " heures, il y a " + str(a*b) + " minute
print(message)
```

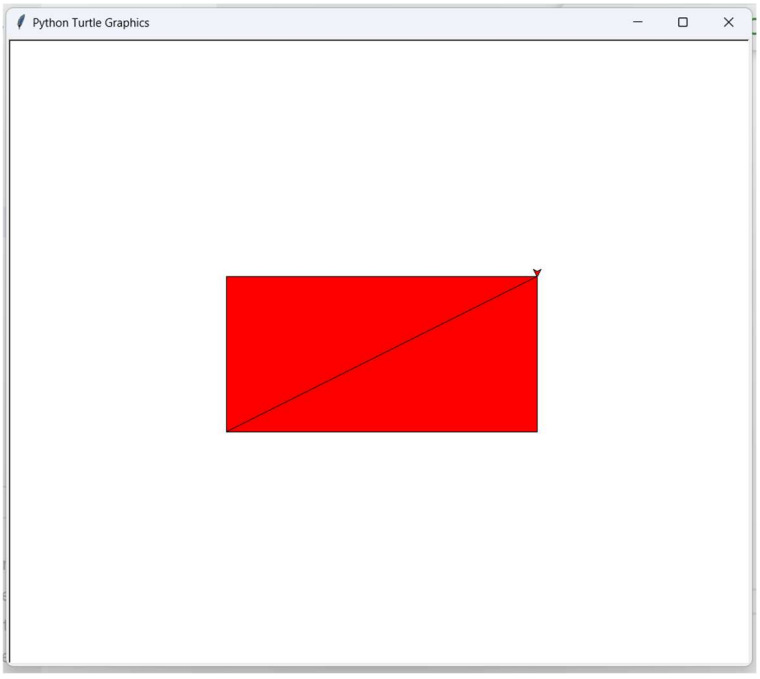
8- INTRODUCTION A LA BIBLIOTHEQUE TURTLE

Le code ci-dessous permet de créer une fenêtre graphique et d'y afficher une figure :


```
from turtle import *

fillcolor("red")
up()
goto(-200,-100)
down()
begin_fill()
forward(400)
left(90)
forward(200)
left(90)
forward(400)
left(90)
forward(200)
end_fill()
goto(200,100)

mainloop()
```



9- EXERCICES :

a. EXERCICE 1 :

Aux Etats-Unis, une température est mesurée en degrés Fahrenheit (°F). Dans le reste du monde, c'est le degré Celsius (°C) qui est utilisé. Si la température donnée en °F est notée *T*, si celle en °C est notée *temperatureCelsius*, on a : $temperatureCelsius = \frac{55}{100}T - 17$

Le code ci-dessous est incomplet. Il permet, après saisie d'une température en °C, d'afficher la même température exprimée en °F.

Son exécution donne dans la console :

```
>>> (executing file "calculTemperature.py")
Ce code convertit une température de °F à °C
Donner la température en °F : 80
80.0°F c'est pareil que 27.0°C
```

```
print("

print(f"{T}°F c'est pareil que {temperatureCelsius}°C")
```

b. EXERCICE 2 :

En plongeant sous l'eau, la pression p (en bars) augmente rapidement avec la profondeur d'immersion H (en mètres).

On a : $p = ro \times g \times H$ avec $ro = 1000$ et $g = 9{,}81$

Le code ci-dessous est incomplet. Il permet, après saisie d'une profondeur H, d'afficher la pression p.

Son exécution donne dans la console :

```
>>> (executing file "calculPression.py")
Ce code calcule la pression, à une profondeur H
Donner la profondeur en m : 55
Pour une profondeur 55.0 m, la pression est de 5.3955 bars
```

```
print(

print(message)
```

c. EXERCICE 3 :

L'aire d'un disque de diamètre D est donnée par la relation $A = \frac{\pi D^2}{4}$

Le code ci-dessous est incomplet. Il permet, après saisie d'un diamètre D, d'afficher l'aire. Son exécution donne dans la console :

```
>>> (executing file "calculSurface.py")
Ce code calcule l'aire d'un disque
Donner le diamètre du cercle : 1
Pour un diamètre de  1.0  m l'aire est de 0.785 m²
```

```
print(




aire = pi * D**2 / 4

print("Pour un diamètre de ",D,                                    )
```