

# TP Python

Exercice 1 :

Écriture hello python .

The screenshot shows the Spyder IDE interface. On the left, the code editor displays a single file named '1.py' with the following content:

```
1 print("Hello, Python")
```

On the right, the 'Console' tab is active, showing the following session:

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 08:28:27) [Clang 14.0.6]
Type "copyright", "credits" or "license" for more information.
IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/gbaguidi/td python 1.py', wdir='/Users/gbaguidi')
File <ipython-input-1>
      ^
      |
      print('Hello, Python'
SyntaxError: unterminated string literal (detected at line 9)

In [2]: runfile('/Users/gbaguidi/td python 1.py', wdir='/Users/gbaguidi')
Hello, Python

In [3]: runfile('/Users/gbaguidi/td python 1.py', wdir='/Users/gbaguidi')
Hello, Python

In [4]:
```

ci-joint écriture des variables avec leur affichage .

The screenshot shows the Spyder IDE interface. On the left, the code editor displays a file named 'td python 1.py' with the following content:

```
1 print("Hello, Python")
2 #Déclaration des variables
3 Année = 2025
4 Ville = "Paris"
5 température = 18.5
6
7 print("Année : ", Année)
8 print("Ville : ", Ville)
9 print("température : ", température)
```

On the right, the 'Console' tab shows the execution of the script and its output:

```
In [1]: runfile('/Users/gbaguidi/td python 1.py', wdir='/Users/gbaguidi')
Hello, Python
Année : 2025
Ville : Paris
température : 18.5
```

A tooltip window titled 'Usage' provides information on how to get help for objects.

## Exercices 2 :

ci-joint un ensemble d'opération d'arithmétiques avec deux variables .

The screenshot shows the Spyder IDE interface. On the left, the code editor displays a file named 'td python 1.py' with the following content:

```
1
2
3
4
5
6
7
8
9
10 a = 10
11 b = 3
12
13 print("Addition : ", a+b)
14 print("Soustraction : ", a - b)
15 print("Multiplication : ", a * b)
16 print("Division entière : ", a // b)
17 print("Reste de la division : ", a % b)
```

On the right, the 'Console' tab shows the execution of the script and its output:

```
In [1]: runfile('/Users/gbaguidi/td python 1.py', wdir='/Users/gbaguidi')
Addition : 13
Soustraction : 7
Multiplication : 30
Division entière : 3
Reste de la division : 1
```

At the bottom, the status bar indicates 'conda: base (Python 3.12.7)' and memory usage 'Mem 73%'.

## Exercices 3 :

Ci-joint la suite d'opération logique avec deux variables booléennes .

The screenshot shows a Jupyter Notebook interface. On the left, a code cell contains Python code for logical operations:

```
27
28
29
30
31
32
33     a = False
34     b = True
35     print("a AND b : ", a and b)
36     print("a OR b : ", a or b)
37     print("NOT a : ", not a)
38     print("NOT b : ", not b)
```

The right side shows the output of the code in the IPython Console:

```
a AND b : False
a OR b : True
NOT a : True
NOT b : False
```

## Exercices 4 :

ci-joint des entrées et affichages pour demander des données à un utilisateur .

The screenshot shows the Spyder IDE environment. On the left, a script file named `td python 1.py` contains the following code:

```
1 nom = input("Entrez votre nom : ")
2 profession = input("Entrez votre profession : ")
3
4 print(f"Bienvenue {nom}, vous êtes {profession} !")
5
6 karim
```

In the center, a help dialog titled "Usage" provides information on how to get help for objects:

```
Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
```

At the bottom, the IPython Console shows the execution of the script and the resulting output:

```
Entrez votre nom : karim
Entrez votre profession : ingénieur
Bienvenue karim, vous êtes ingénieur !
```