

Chapitre 1 : Introduction à la programmation Python

Parcours 1 : Langage Python

I - Le langage Python : introduction

Python est un langage de programmation qui est placé sous licence libre et fonctionne sous la plupart des systèmes d'exploitations (Windows, MacOS, GNU/Linux) mais aussi sur smartphones ou sur des pages web.

C'est un langage haut-niveau, c'est-à-dire qu'il est proche du langage courant (mais en anglais) et utilise les symboles et les opérations mathématiques courants.

Il a été créé en 1991, écrit en C (un autre langage de programmation) et il est toujours mis à jour régulièrement (on est à la version 3.10 actuellement).



I - Le langage Python : l'environnement de programmation

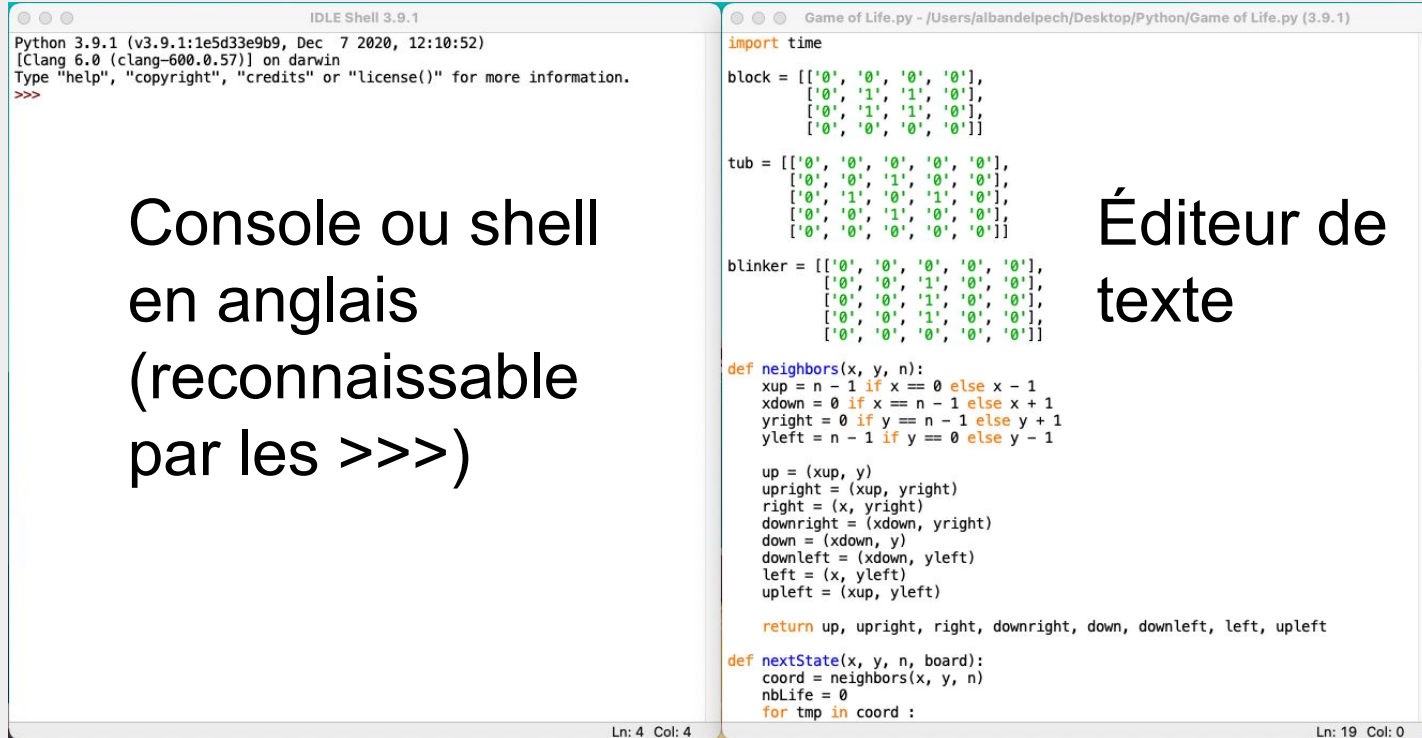
Pour pouvoir exécuter du code Python, on a besoin d'un programme spécial appelé **interprète Python**. Il existe pleins d'interprètes Python différents. Dans ce cours, j'utiliserai IDLE mais la plupart des interprètes se ressemblent et il est très facile de passer d'un à l'autre.

Un interprète Python est toujours composé de deux parties :

- un **éditeur de texte** qui permet d'écrire plusieurs lignes de code qui seront ensuite exécutées ;
- une **console** qui permet d'afficher les résultats d'un code et qui permet aussi d'écrire et d'exécuter des instructions Python.



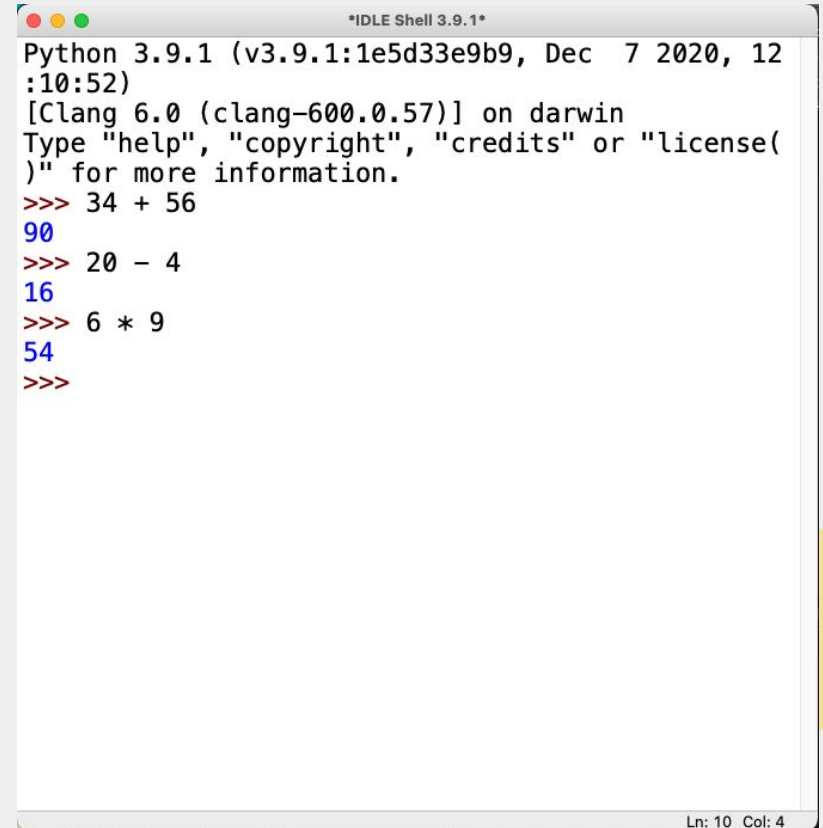
I - Le langage Python : l'environnement de programmation



I - Le langage Python : l'environnement de programmation

Dans la console, vous pouvez dialoguer directement avec l'interpréteur mais uniquement ligne par ligne : elle n'est donc pas adaptée pour écrire du code, voyez-la plutôt comme une calculatrice.

Pour exécuter une instruction Python avec la console, il suffit d'écrire l'instruction après les trois chevrons `>>>` et d'appuyer sur entrée. Le résultat sera écrit sur la ligne juste en dessous.



```
*IDLE Shell 3.9.1*
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec  7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> 34 + 56
90
>>> 20 - 4
16
>>> 6 * 9
54
>>>
```

Ln: 10 Col: 4

I - Le langage Python : calculer avec Python

Comme dit précédemment, Python permet d'effectuer des calculs. Voici quelques opérateurs qui vous seront utiles.

Opérateur	Résultat	Exemple
+	Addition	5 + 3 renvoie 8
-	Soustraction	5 - 3 renvoie 2
*	Multiplication	4 * 3 renvoie 12
/	Division	13 / 2 renvoie 6.5
//	Division entière	13 // 2 renvoie 6
%	Reste de la division entière	13 % 2 renvoie 1
**	Puissance	4 ** 3 renvoie 64

I - Le langage Python : calculer avec Python

Remarques :

- Pour écrire un nombre à virgule, on utilise un point et pas une virgule (8,45 s'écrit 8.45 en Python).
- Python utilise les mêmes règles de priorité d'opération que vous connaissez et vous pouvez utiliser des parenthèses.
- Pour écrire un nombre négatif, il suffit de mettre un - devant.

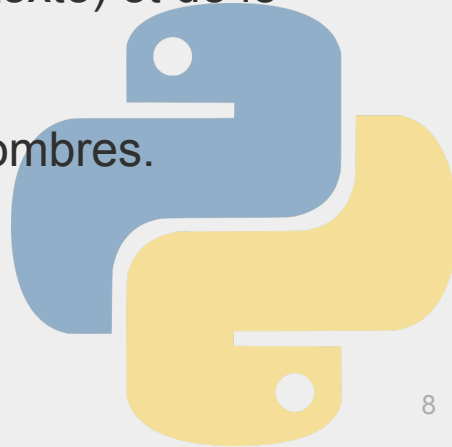


II - Les variables : principe

Le but d'un programme est de traiter des données. Pour pouvoir être utilisées, ces données sont représentées sous forme de **variable**.

Une variable permet de mémoriser une donnée (nombre ou du texte) et de le stocker dans la mémoire de l'ordinateur, prêt à être utilisé.

Pour le moment nous verrons des variables représentant des nombres.



II - Les variables : définir une variable

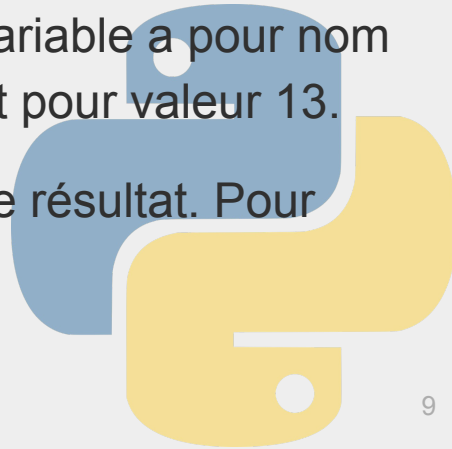
Une variable est créée en lui donnant un nom et une valeur et en utilisant le symbole =.

Exemple :

```
>>> maVariable = 4  
>>> var2 = 5 + 8
```

Dans l'exemple, nous avons créé deux variables : la première variable a pour nom "maVariable" et pour valeur 4, la deuxième a pour nom "var2" et pour valeur 13.

Contrairement à un calcul, définir une variable ne renvoie pas de résultat. Pour connaître la valeur d'une variable, il suffit d'écrire son nom.



II - Les variables : définir une variable

Le symbole = est une opération d'affectation : il attend à sa gauche un nom valide et à sa droite une expression (calcul, texte,...). Une fois une variable définie, on peut la réutiliser en utilisant son nom.

```
>>> var = 7  
>>> var * (1 + var)  
56  
>>>
```

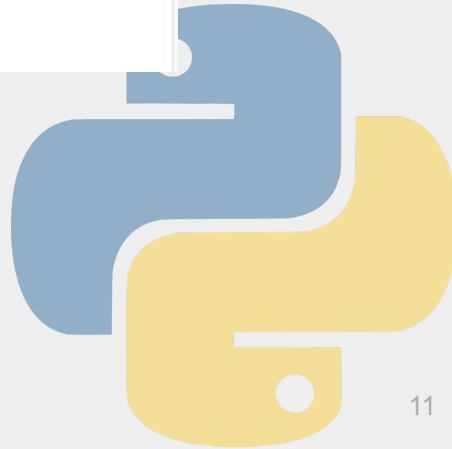
Attention : Un nom de variable doit respecter certaines règles : il doit être formé uniquement de lettres, chiffres et du caractère underscore _, il ne doit pas commencer par un chiffre et certains nom sont interdits car ils sont réservés par le langage Python.

II - Les variables : modifier une variable

Comme son nom l'indique, une variable peut avoir sa valeur modifiée pendant un programme. Une nouvelle affectation remplacera l'ancienne valeur de la variable. Une variable a comme valeur la dernière valeur affectée.

```
>>> var = 8  
>>> var = var + 5  
>>> var  
13
```

On a redéfini la valeur de var en utilisant son ancienne valeur.

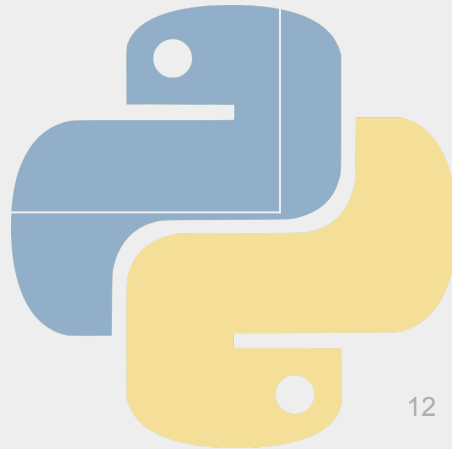


II - Les variables : activité

Quelles sont les valeurs affichées par la console après les séquences d'instructions suivantes ?

```
>>> a = 3
>>> a = 4
>>> a = a + 1
>>> a
```

```
>>> a = 2
>>> b = a * a
>>> b = a * b
>>> b = b * b
>>> b
```



III - Les instructions de base : affichage avec print()

A partir de maintenant, nous n'utiliserons plus la console, tous les programmes se feront dans le mode éditeur de texte. Mais dans ce mode, les résultats des expressions ne sont plus affichés à l'écran.

On va utiliser une fonction qui nous permettra de faire apparaître dans la console ce que l'on souhaite. Cette fonction c'est ***print***.

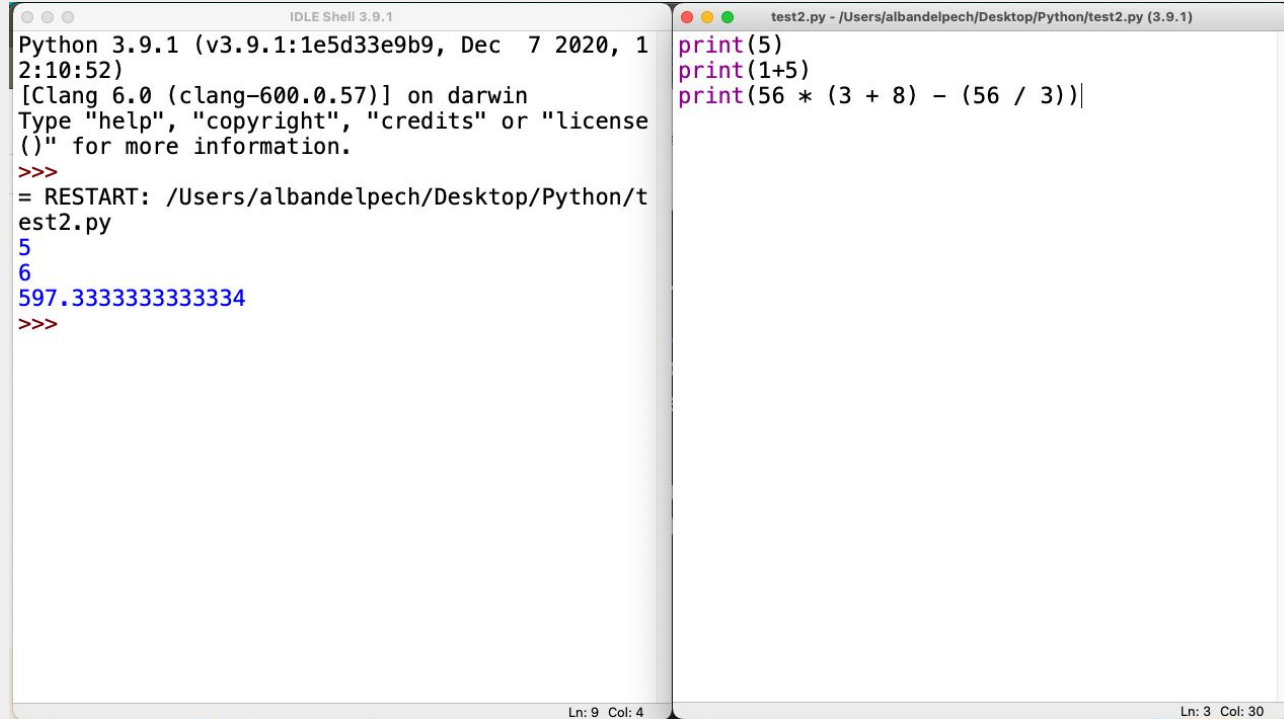


III - Les instructions de base : affichage avec print()

Dans cet exemple, on peut voir que la fonction print va afficher successivement chacun des calculs qui lui sont donnés.

Attention :

L'expression à afficher doit être entre parenthèses.



The image shows two side-by-side screenshots of a Python IDE. The left window, titled 'IDLE Shell 3.9.1', displays the Python 3.9.1 startup banner and a prompt where the user has entered '5', '6', and a large number '597.33333333333334'. The right window, titled 'test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)', shows a Python script with three lines of code: `print(5)`, `print(1+5)`, and `print(56 * (3 + 8) - (56 / 3))`. The status bars at the bottom of the windows indicate 'Ln: 9 Col: 4' for the shell and 'Ln: 3 Col: 30' for the script.

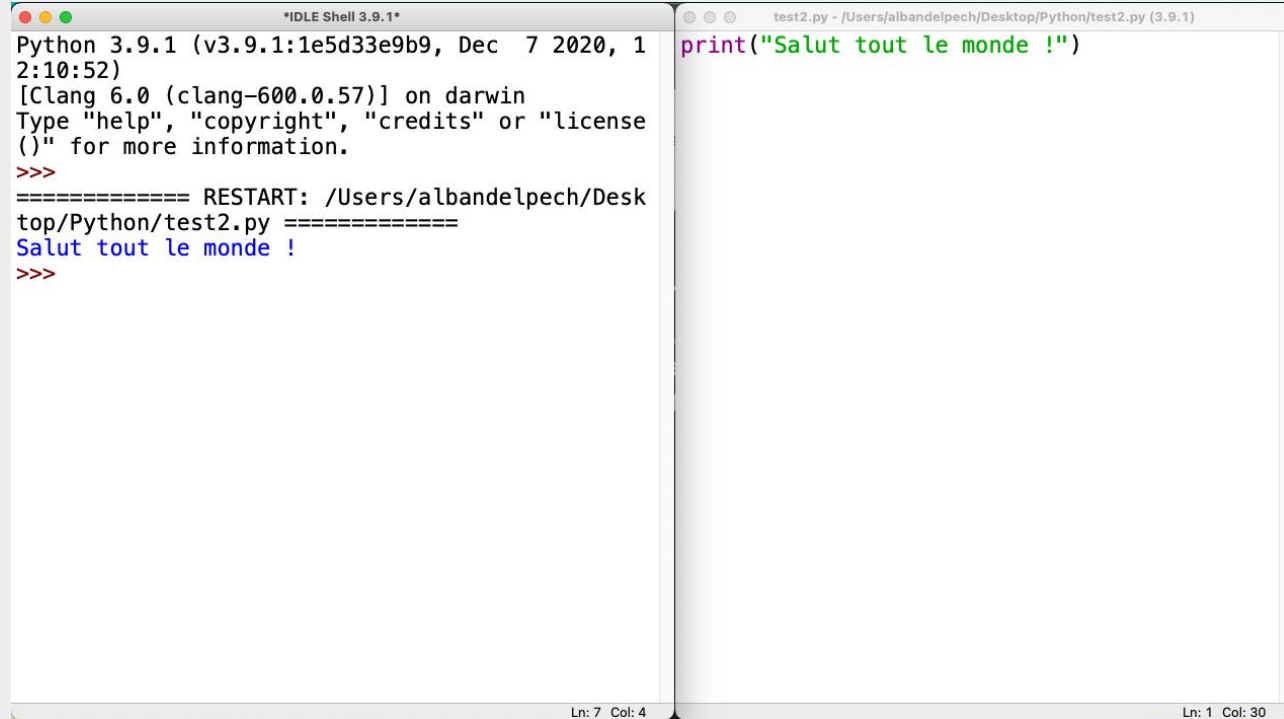
```
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec 7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /Users/albandelpech/Desktop/Python/test2.py
5
6
597.33333333333334
>>>
```

```
print(5)
print(1+5)
print(56 * (3 + 8) - (56 / 3))
```

III - Les instructions de base : affichage avec print()

L'instruction print n'est pas limitée à l'affichage de nombres. On peut aussi afficher du texte en le mettant entre guillemet.

Le texte entre guillemets est une **chaîne de caractères**.



```
*IDLE Shell 3.9.1*
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec 7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/albandelpech/Desktop/Python/test2.py =====
Salut tout le monde !
>>>
```

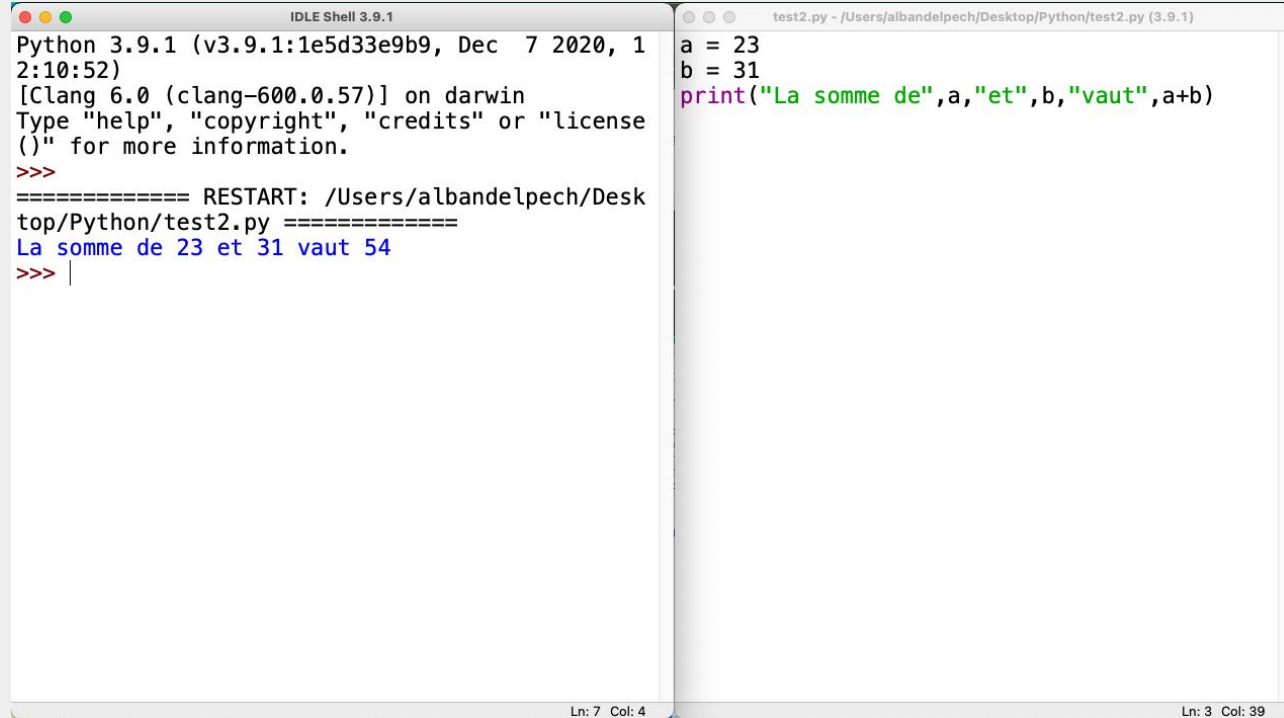
```
test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)
print("Salut tout le monde !")
```

Ln: 7 Col: 4

Ln: 1 Col: 30

III - Les instructions de base : affichage avec print()

On peut aussi afficher du texte, des calculs et la valeur d'une variable en les séparant par des virgules.



The image shows two side-by-side screenshots of a Python IDE. The left window, titled 'IDLE Shell 3.9.1', displays the Python 3.9.1 startup banner and a prompt where the user has entered a restart command. The output shows the file path and the result of the calculation. The right window, titled 'test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)', shows a script with two variables and a print statement that formats the output with French text and color coding.

```
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec 7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/albandelpech/Desktop/Python/test2.py =====
La somme de 23 et 31 vaut 54
>>> |
```

```
a = 23
b = 31
print("La somme de",a,"et",b,"vaut",a+b)
```

Ln: 7 Col: 4

Ln: 3 Col: 39

III - Les instructions de base : interagir avec l'utilisateur

Pour permettre une interaction entre programme et utilisateur, par exemple la saisie d'une valeur, on doit procéder en deux temps :

1. on utilise d'abord la fonction ***input*** pour récupérer les caractères entrés ;
2. puis on utilise la fonction ***int*** pour transformer la chaîne de caractère en entier.

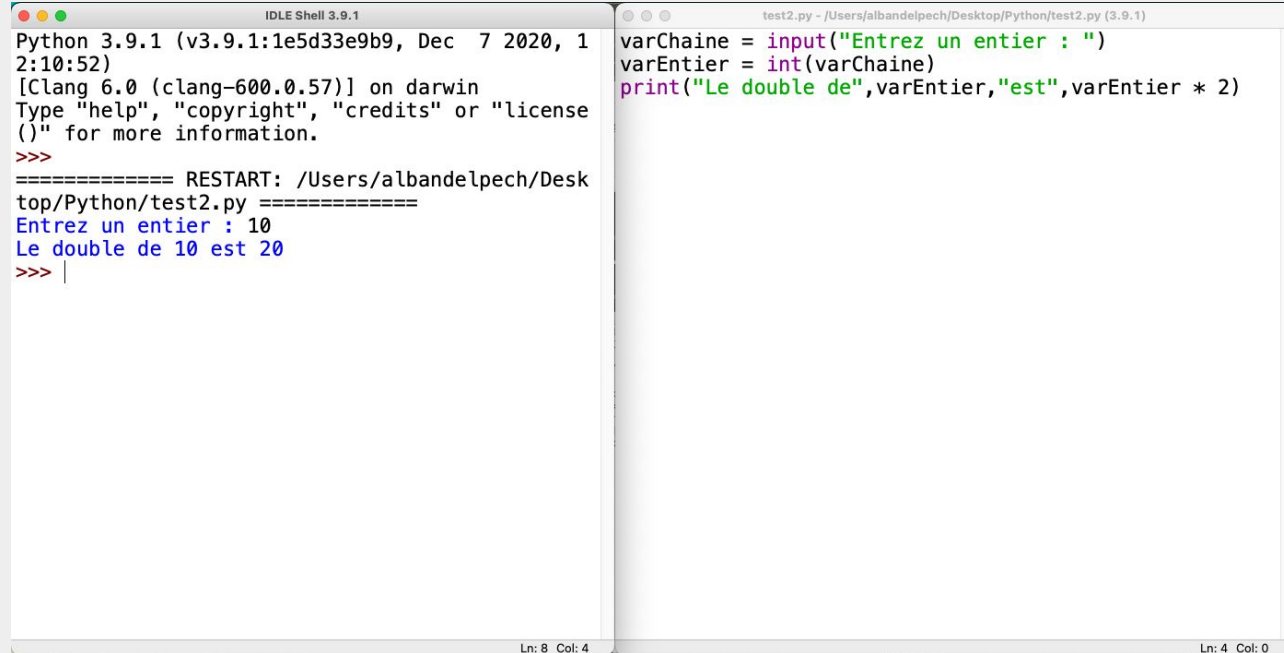
Attention : On effectue l'étape 2 uniquement si on attend que l'utilisateur rentre un entier. Si on attend une chaîne de caractère (par exemple, le nom de l'utilisateur), on ne fait que l'étape 1.



III - Les instructions de base : interagir avec l'utilisateur

La fonction va interrompre le programme jusqu'à que l'utilisateur rentre un nombre dans la console.

Le texte dans input est optionnel et sert uniquement pour donner des indications à l'utilisateur.



The image shows a screenshot of a Python IDE with two windows. The left window, titled 'IDLE Shell 3.9.1', displays the output of a Python script execution. It shows the Python version (3.9.1), the compiler (Clang 6.0), and the operating system (darwin). It then prompts the user to enter a number, and the user enters '10'. The script then prints 'Le double de 10 est 20'. The right window, titled 'test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)', shows the source code of the script. The code defines a variable 'varChaine' using the 'input' function with the prompt 'Entrez un entier : ', converts it to an integer using 'int', and then prints the result of multiplying it by 2 using the 'print' function.

```
Python 3.9.1 (v3.9.1:1e5d33e9b9, Dec 7 2020, 12:10:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/albandelpech/Desktop/Python/test2.py =====
Entrez un entier : 10
Le double de 10 est 20
>>> |
```

```
varChaine = input("Entrez un entier : ")
varEntier = int(varChaine)
print("Le double de", varEntier, "est", varEntier * 2)
```

III - Les instructions de base : interagir avec l'utilisateur

Remarques :

- Si on utilise la fonction `int` alors que la chaîne ne représente clairement pas un entier (ex : "toto"), on obtient une erreur et le programme ne continue pas.
- On peut associer les deux fonctions `int` et `input` en une seule ligne pour gagner en place. Attention à ne pas oublier les parenthèses.



```
*test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)*  
varChaine = input("Entrez un entier : ")  
varEntier = int(varChaine)
```

```
*test2.py - /Users/albandelpech/Desktop/Python/test2.py (3.9.1)*  
varEntier = int(input("Entrez un entier : "))
```