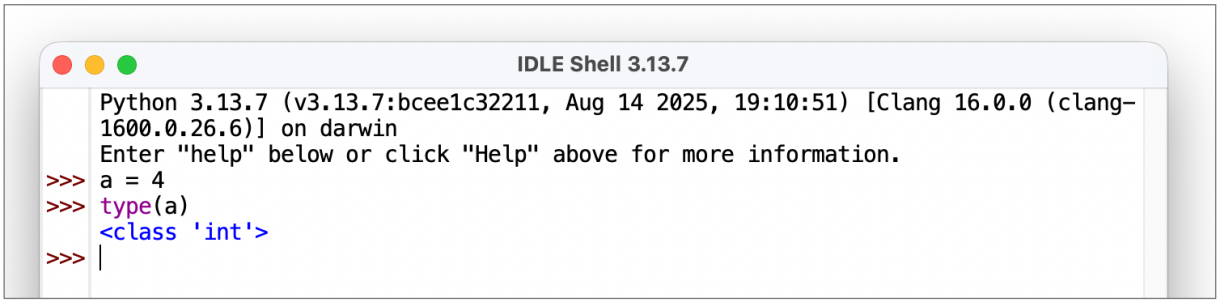
Premiers pas en langage Python

Lorsqu'on souhaite programmer en langage Python, il est important de disposer d'un environnement adapté. **IDLE Python** est précisément ce que l'on appelle un environnement de développement en langage Python.

Lorsqu'on exécute **IDLE**, une fenêtre vierge s'ouvre : il s'agit de la fenêtre de l'**Interpréteur** ou **Console** Python, laquelle permet d'entrer, derrière trois chevrons `>>>` et une invite ou prompt `|` qui clignote, des instructions ou commandes en langage Python.

Exemple 1

La fenêtre ci-dessous est la fenêtre de l'Interpréteur Python dans laquelle ont été écrites et exécutées deux instructions :



L'instruction `a = 4` une instruction d'affectation. Elle affecte la valeur `4` à la variable `a`. En réalité, les choses sont légèrement plus subtiles.

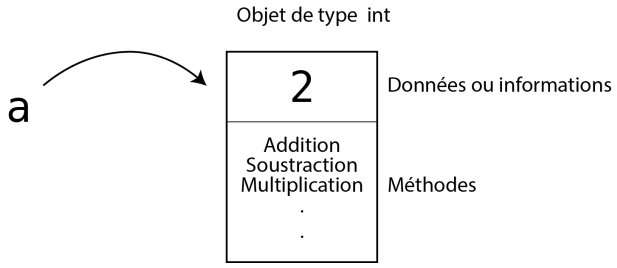
L'instruction `a = 4` crée, par typage dynamique, un **objet** d'une classe d'objet. Ici, l'objet créé est référé par la référence `a`.

L'instruction `type(a)` permet de savoir quel est le type de l'objet `a`.

L'objet créé appartient à la **classe d'objets int** (integer), comme l'indique `<class 'int'>`.

Qu'est-ce qu'un objet ?

Il s'agit d'une structure définie par (1) des données et (2) des méthodes.



Lorsqu'on crée des objets, il n'est possible d'exécuter sur ces objets que les **méthodes** associées à ces objets. Ainsi, sur des nombres entiers, il est possible de réaliser des additions, méthodes qui sont associées aux entiers (integers), comme on le voit sur le schéma ci-dessus.

En résumé, le langage Python est un **langage de programmation Objet**, les objets étant des structures composées de données et de méthodes.

Exemple 2

```
    Python 3.13.7 (v3.13.7:bcee1c32211, Aug 14 2025, 19:10:51) [Clang 16.0.0 (clang-
    1600.0.26.6)] on darwin
    Enter "help" below or click "Help" above for more information.
>>> a = 4
>>> type(a)
    <class 'int'>
>>> b = 3.14
>>> type(b)
    <class 'float'>
>>> name = "Python"
>>> type(name)
    <class 'str'>
```

Dans la fenêtre ci-dessus, on observe que :

- quand on écrit `a = 4`, un objet de type **int** (nombre entier) est créé,
- quand on écrit `b = 3.14`, un objet de type **float** (nombre à virgule) est créé, et
- quand on écrit `name = 'Python'`, un objet de type **str** (chaîne de caractères) est créé.

On dit que le langage Python est un **langage à typage dynamique**.

A chaque type d'objet sont associées des méthodes. Ainsi, on peut multiplier des nombres entiers **int** ou des nombres à virgules **float**. Mais peut-on multiplier entre elles des chaînes de caractères **str** ?

```
>>> a * b
    12.56
>>> name * name
    Traceback (most recent call last):
      File "<pyshell#7>", line 1, in <module>
        name * name
    TypeError: can't multiply sequence by non-int of type 'str'
```

On constate qu'il n'est pas possible de multiplier une chaîne de caractère par une chaîne de caractères. Mais peut-on multiplier une chaîne de caractère par un nombre ?

```
>>> 3*name
    'PythonPythonPython'
```

Multiplier par 3 la chaîne de caractères 'Python' crée une nouvelle chaîne de caractères répétant trois fois la chaîne multipliée.

synthèse

Le langage Python est un **langage de programmation Objet** à **typage dynamique** dans lequel des **objets** issus de **classes d'objets** de type **int**, **float**, **str**, par exemple, sont manipulés à l'aide de **méthodes**.