Introduction to Python

Python Journey

Pedro Gasparinho

April 15, 2025

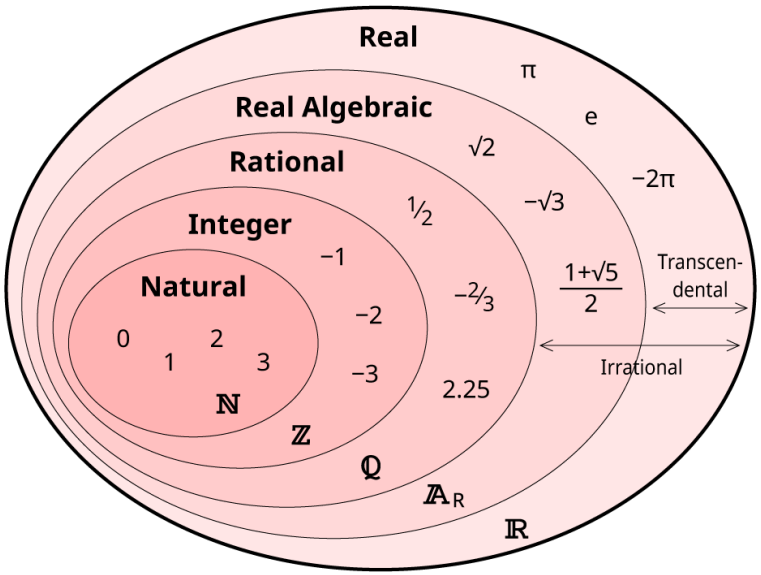
First Program

```python
print('Hello World!')
```

Python

- *print* - Function
- *'Hello World!'* - Value of type string

Number Sets



- There is a need to group numbers with certain characteristics in different sets.
- These sets are usually built on top of each other.
- Although, it is possible to restrict which set or subset will be used to fit our needs.

Data Types in Python

- Similar to Maths, Python and other programming languages have sets of values, known as data types.
- However, these sets are not limited to numbers, it is also possible to represent text, true and false, and more.
- A value can have multiple representations, but each representation is only associated with a single data type. For instance, 1 is an integer value, while 1.0 is considered as a real number (float).

Data Types in Python

- Integers
  - Represented as *int*
  - Examples: *1*, *-4*, *0*
- Real Numbers (Floats)
  - Represented as *float*
  - Examples: *1.0*, *-6.2*, *3.1415926*
- Logical Values (Booleans)
  - Represented as *bool*
  - Exemples: *True*, *False*

Data Types in Python

- Text (Strings)
  - Represented as *str*
  - Exemples: *'1'*, *"Hello, world"*, *"""Olá"""*
- Lists
  - Represented as *list*
  - Exemples: *[1, 2, 3]*, *[]*, *["Olá", "Olé"]*, *["Olá", 1, ["ok"], true]*
- And much more...

Some notes about Floats

- Computers are not able to fully represent real numbers, for instance, numbers with infinite decimal places, since computers have finite storage space to store data.
- Another classical example is the sum of *0.1* with *0.2*, which results in *0.30000000000000004*.
- This occurs due the to technical standard IEEE 754, used to represent real numbers, but will not be discussed in these materials.

Some notes about Strings

- Must be delimited with either apostrophes ('...') or quotation marks ("...").
- Choosing apostrophes as a delimiter allows the use of quotation marks inside the string, and vice versa.
- Moreover, it is possible to repeat the same delimiter 3 times in sequence to allow strings to span multiple lines. This is commonly used in comments.

Some notes about Lists

- The purpose of lists is to store multiple values in a specific order and allows for repetitions.
- In Python, unlike many other programming languages, lists can values of different data types.

One of the most simple uses of programming languages is as calculator, through the use of basic arithmetic operators:

- $+$ represents sum. Ex:
  - $2 + 2$, is equal to 4
  - $-1.2 + 3.6$, is equal to 2.4
- $-$ represents subtraction. Ex:
  - $3 - 1$, is equal to 2
  - $5.3 - 1.8$, is equal to 3.5
- $*$ represents multiplication. Ex:
  - $5 * 2$, is equal to 10
  - $2.9 * 3$, is equal to 8.7

Arithmetic expressions

- / represents real division. Ex:
  - 4/2, is equal to 2.0
  - 5.4/0.25, is equal to 21.6
- // represents integer division. Ex:
  - 4//2, is equal to 2
  - 5//2, is equal to 2
- % represents integer division remainder. Ex:
  - 4//2, is equal to 0
  - 5//2, is equal to 1
- ** represents exponentiation. Ex:
  - 2 ** 4, is equal to 16
  - 9 ** 0.5, is equal to 3.0

Notes about Arithmetic expressions

- Similar to Maths, division by 0 is not defined in Python, and if performed it will produce an **error**.
- Real division always produces a value of the **float** type.
- Both the integer division and its remainder always produce values of the **integer** type.
- The data type of the result of the exponentiation operation depends on its operands, these being the base and exponent. If at least one is float then the result will be float, if both are integers it will be integer.

Python by itself offers the previously mentioned operations, but many more mathematical operations and constants can be accessed through the Math library, for instance pi, neper number, the logarithm function, the square root (alternatively to the exponentiation with exponent 1/2), and much more.

Write a program in Python that calculates the Euclidean distance between points (3, 4) e (7, 1). The formula for points (x_1, y_1) and (x_2, y_2) is:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Maybe the most intuitive solution is:

Python

```
import math

print(math.sqrt((3 - 7)**2 + (4 - 1)**2))
```

What if the exercise also asked us to calculate the distance between two other points? Easy, just copy and paste! However,

this introduces a few problems:

- The formula of the Euclidean distance is relatively simple, however copying complex code excerpts with multiple lines can make the code hard to read.
- If we detect an error in the formula, then it would be necessary to correct it in multiple places, with the cost of possibly forgetting to change in some place.

What is the best solution?

A more adequate solution would be:

Python

```
import math

def dist(x1, y1, x2, y2):
    return math.sqrt((3 - 7)**2 + (4 - 1)**2)

print(dist(3, 4, 7, 1))
print(dist(0, 0, 0, 1))
```

However, the best solution would be:

Python

```python
import math

def dist(x1, y1, x2, y2: float) -> float:
    """ Calculates the Euclidean distance between two points
        x1, y1 - Coordinates of point 1
        x2, y2 - Coordinates of point 2
        Returns the distance as a real number """
    return math.sqrt((3 - 7)**2 + (4 - 1)**2)

print(dist(3, 4, 7, 1))
print(dist(0, 0, 0, 1))
```

- The order of the parameters matters (Although there are techniques that contradict this, but will not be presented here)
- With a few exceptions names do not matter, but for good practice should be related to the expected behaviour.
- Why use functions? Would you need a button in a calculator that specifically calculates $ln(1 + 5/2)$? Instead, you need buttons for the numbers, the sum, the division, the logarithm function, etc.