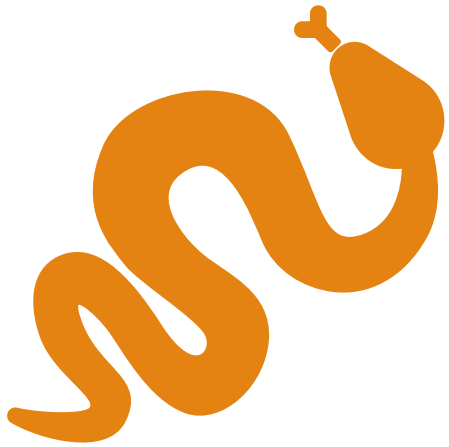

Introduction to Python

ICDF 2021 PYTHON CLASS

2021-11-06

Content

- Compiler vs. Interpreter
- Basic types.
- Variables and Expressions.
- Classes and Objects.
- Methods and Functions.
- String operations.
- CoLab.

Why Python?

- **Compiler:**
  - Translates the whole code or program into computer language.
  - Execution time is far faster than interpreters.
  - Relatively hard to understand and debug.
  - Rigid structure.
  - C, Java.
- **Interpreter:**
  - Translates code line by line.
  - Easier to understand and debug.
  - Execution time is longer.
  - More flexible.
  - Python, JavaScript.

Basic Types and Casting

- Integer:
  - *class* **int**
  - -10, 0, 10…
- Float:
  - *class* **float**
  - 1.3, -24.05, 0.0
- Bool:
  - *class* **bool**
  - True, False
- String:
  - *class* **str**
  - ‘A’, ‘fjdakA54’, ’45’

- Integer -> Float:
  - float(10) -> 10.0
  - 10 * 1. -> 10.0
- Float -> Integer:
  - int(10.9) -> 10
- Bool -> Integer:
  - int(True) -> 1
  - int(False) -> 0
- Integer -> Bool:
  - bool(0) -> False
  - bool(2) -> True

Variables and Expressions

Variable:

- Location in memory (RAM).
- Can be of any type (int, bool, float, etc.).

```
In [10]: a = 10
         a

Out[10]: 10
```

Expressions:

- Operations to be executed.
- Can include variables and/or constants.

```
In [11]: a + 20 - 3

Out[11]: 27
```

Classes and Objects

- Class:
  - Collection of data (variables) and methods.
  - Unique identifier.
- Object:
  - Instance of a class.
  - Multiple instances of a class.
  - *Everything is an object!*

```
p1 = Particle()
p2 = Particle()
p3 = Particle()
```

```
# Particle Class
class Particle(object):
    def __init__(self, dimensions: int = 2, x_min: float = -2.,
                 x_max: float = 2., v_ratio: float = 3.):
        super(Particle, self).__init__()
        self.dimensions = dimensions
        self.position = np.random.uniform(x_min, x_max, dimensions)
        self.velocity = np.random.uniform(x_min / v_ratio, x_max / v_ratio, dimensions)
        self.x_max = x_max
        self.x_min = x_min
        self.v_ratio = v_ratio

    def move(self, velocity):
        self.velocity = velocity
        self.position = np.clip(self.position + velocity, self.x_min, self.x_max)
        return self.get_position()

    def get_position(self):
        return self.position

    def get_velocity(self):
        return self.velocity

    def get_attributes(self):
        return (self.get_position(), self.get_velocity())
```

```
In [14]: class Math():
             def add(self, a, b):
                 return a + b

In [15]: m = Math()
         m.add(5, 6)

Out[15]: 11

In [12]: def add(a, b):
             return a + b

In [13]: add(5, 6)

Out[13]: 11
```

Methods and Functions

- Method:
  - It's called on an object or instance.
  - Cannot be accessed outside the object (encapsulation).
  - It may modify the object's state.
- Function:
  - Independent, not associated with objects.
  - Can be called anywhere.

```
In [16]: help('keywords')

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

Reserved Keywords

PYTHON IS CASE SENSITIVE!

String Operations

Concatenation:

‘A’ + ‘B’ -> ‘AB’

Indexing:

Zero-index.

Each character in the sequence (string) has a zero-based index.

Negative indexing.

Slice:

Gets a piece (slice) of the sequence (string).

Google CoLab

- Based on Jupyter Notebook.
- Python ready, no need to install anything.
- Accessed from the web.
- https://colab.research.google.com/