Object Oriented Programming (OOP)

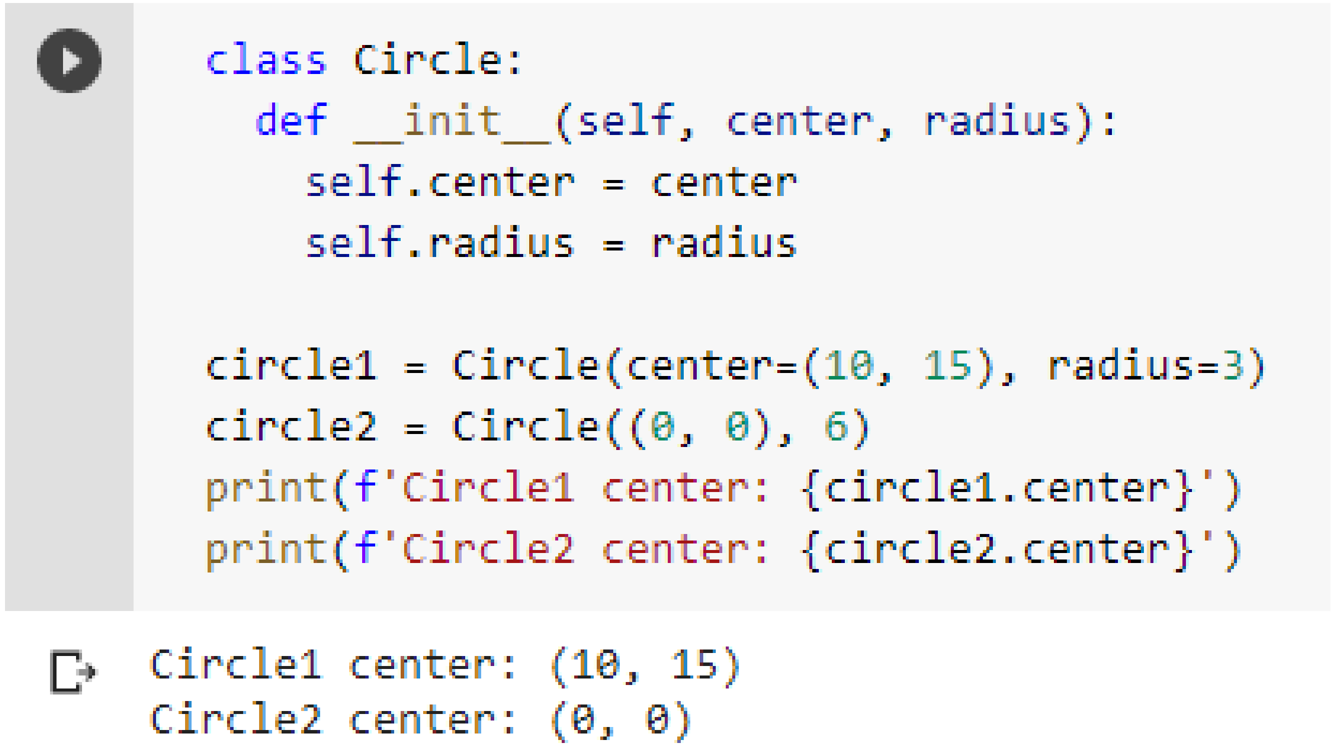
Part 2

Agenda

1. Review on class
2. Constructor __init__
3. Info about classes and objects
4. Private__

Constructor

- Special method for the initiation of the object:



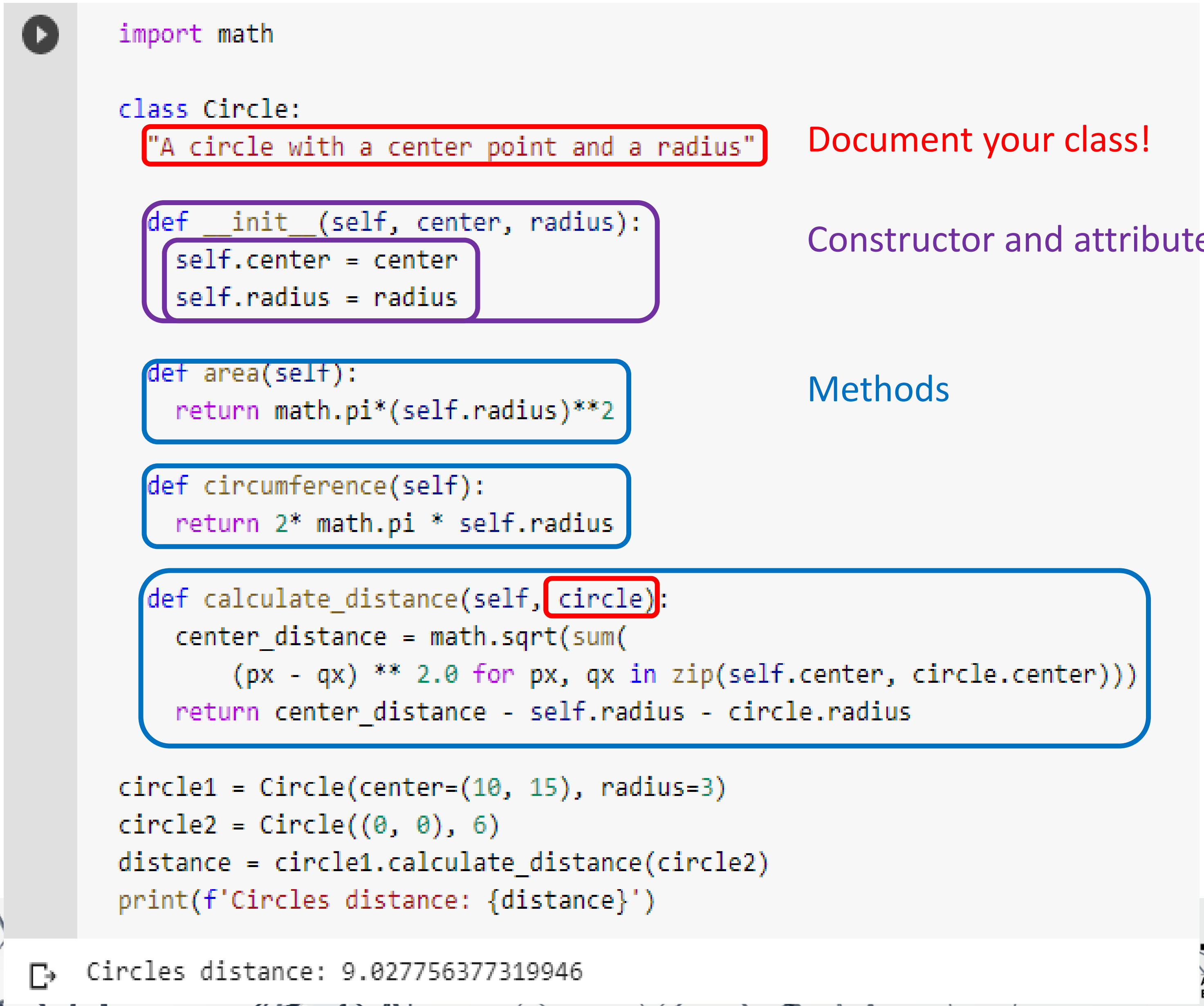
```
class Circle:
  def __init__(self, center, radius):
    self.center = center
    self.radius = radius

circle1 = Circle(center=(10, 15), radius=3)
circle2 = Circle((0, 0), 6)
print(f'Circle1 center: {circle1.center}')
print(f'Circle2 center: {circle2.center}')
```

```
Circle1 center: (10, 15)
Circle2 center: (0, 0)
```

```
def __init__(self)
```

Circle example

- Full class:



```python
import math

class Circle:
  "A circle with a center point and a radius"

  def __init__(self, center, radius):
    self.center = center
    self.radius = radius

  def area(self):
    return math.pi*(self.radius)**2

  def circumference(self):
    return 2* math.pi * self.radius

  def calculate_distance(self, circle):
    center_distance = math.sqrt(sum(
        (px - qx) ** 2.0 for px, qx in zip(self.center, circle.center)))
    return center_distance - self.radius - circle.radius

circle1 = Circle(center=(10, 15), radius=3)
circle2 = Circle((0, 0), 6)
distance = circle1.calculate_distance(circle2)
print(f'Circles distance: {distance}')
```

```
Circles distance: 9.027756377319946
```

Document your class!

Constructor and attributes

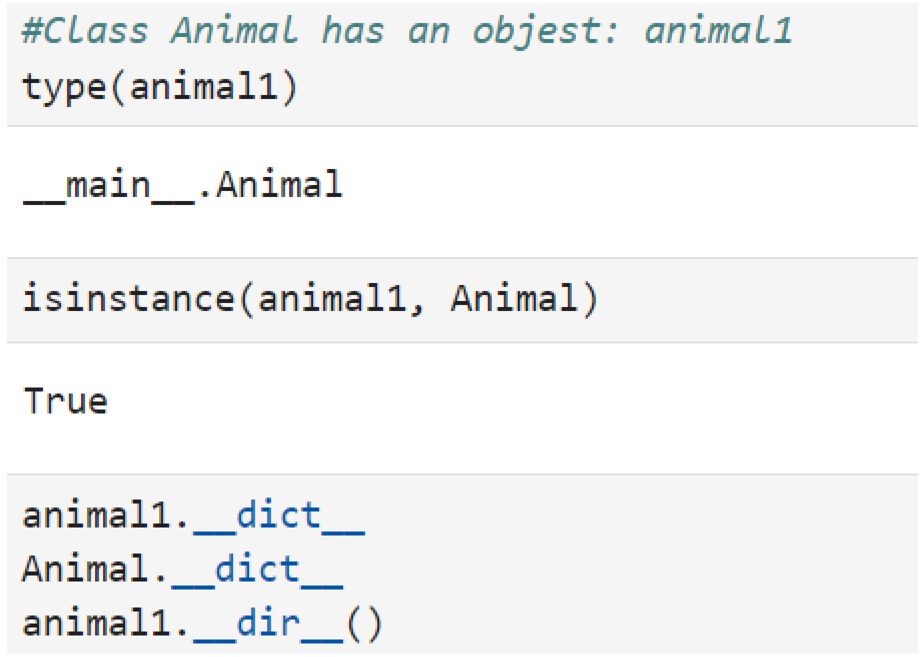
Methods

Naming convention

- Variables:
  - All small letters.
  - Separate words with _.
  - Don’t start with a number.
- Functions:
  - All small letters.
  - Separate words with _.
  - Don’t start with a number.
- Classes:
  - Class name with CamelCase.
  - Attributes – like regular variables.
  - Methods – like regular functions.
  - One class in one file.
  - File name as the class name.

Info about Classes and Attributes

Type()
Isinstance(*object name, class name*)
name.__dict__
object name.__dir__()



```
#Class Animal has an objest: animal1
type(animal1)
```

```
__main__.Animal
```

```
isinstance(animal1, Animal)
```

```
True
```

```
animal1.__dict__
Animal.__dict__
animal1.__dir__()
```

Private attributes and methods

No such thing
but it is insinuated by underscores:

- *_name*

- *__name*