Web Application Development using Python

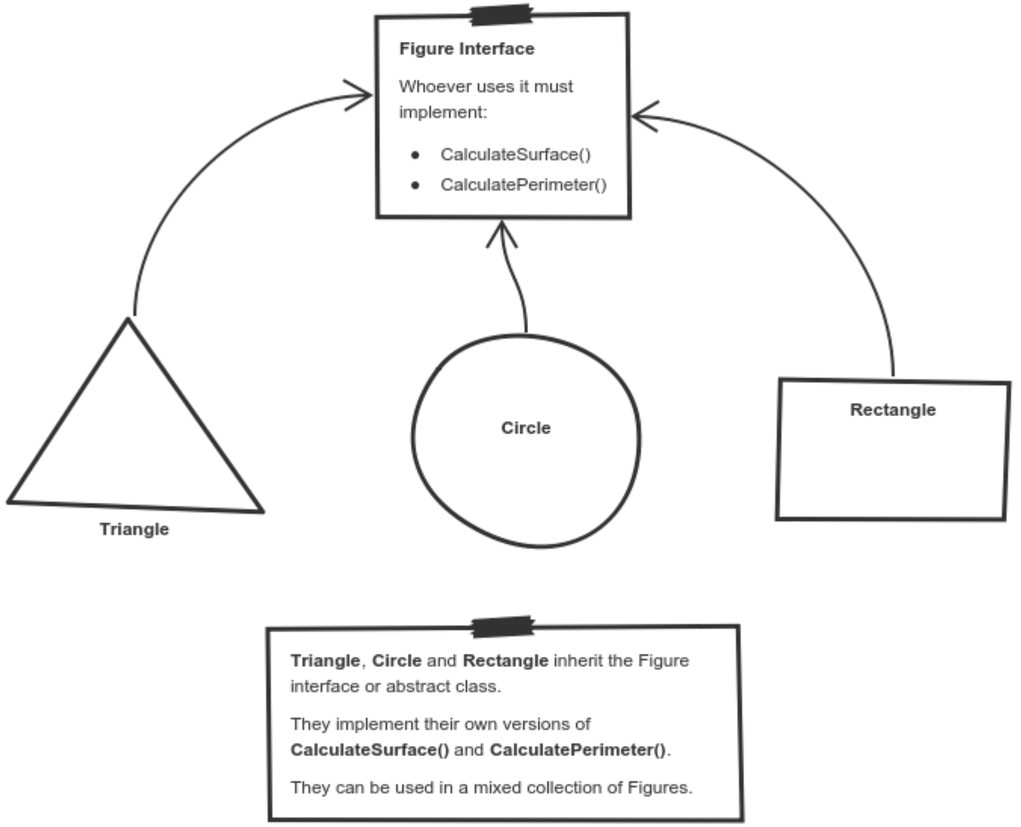
Introduction to Object Oriented Programming - Part 1

Prepared by George Khoury

Outline

- What is Object Oriented Programming?
- OOP vs. “functional programming”
- What is a `class`?
  - `__init__` method
  - What is `self`?
  - Attributes and methods
  - Instance / class object attribute
- How to instantiate objects?

What is Object Oriented Programming?

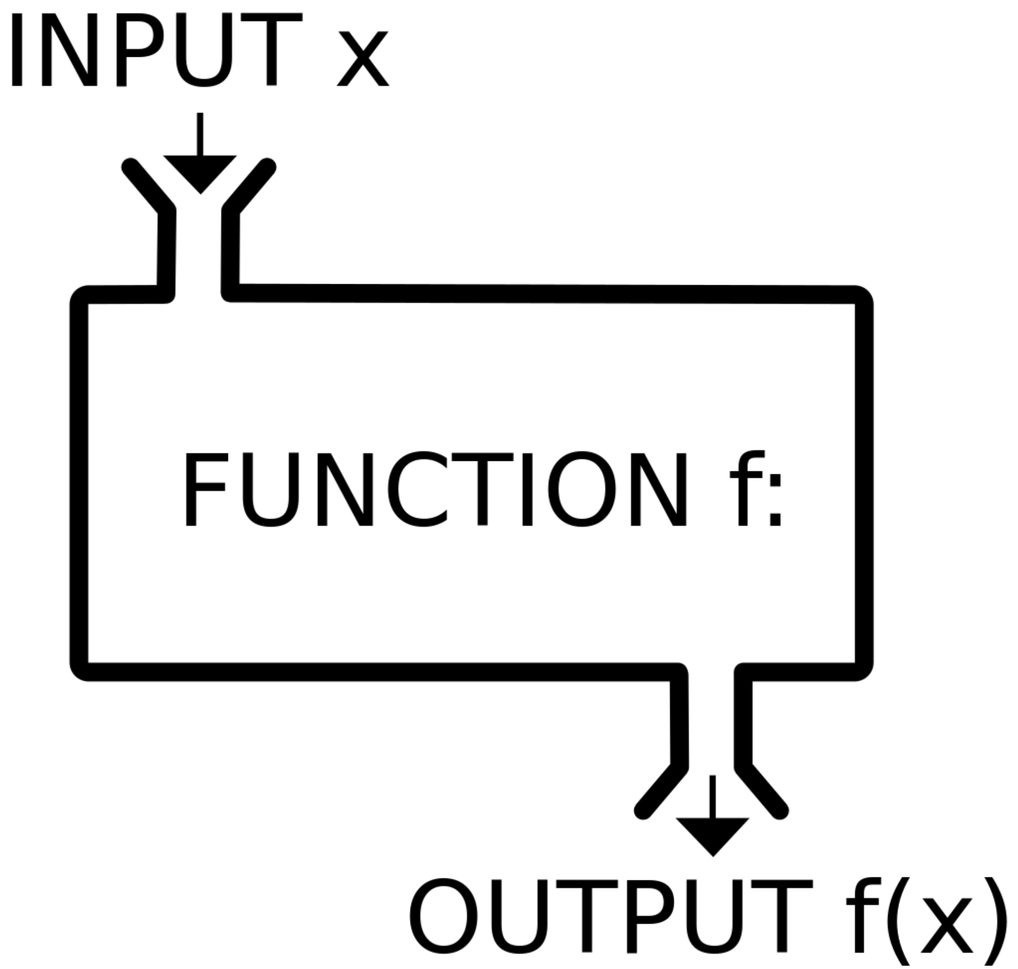


What is Object Oriented Programming?

- Object Oriented Programming (OOP) is a development approach where objects are used to represent the data model.
- Objects have **attributes** which hold the data. **Methods** which perform actions.
- OOP allows us to write neat and modular reusable code.
- Applications usually define more than one object type. Different objects work together to achieve a task.

OOP vs. “functional programming”

How do they compare?



OOP vs. “functional programming”

- The four major principles of object orientation are:
  - **Encapsulation** - a mechanism for restricting the access to some of an object's components, this means that the internal representation of an object can't be seen from outside of the objects definition.
  - Data Abstraction
  - Polymorphism
  - Inheritance

OOP	POP
Program is divided into objects	Program is divided into functions
Encapsulation using access modifiers	No encapsulation
More secure	Less secure
Objects can be used with other objects	Global data can be shared across functions
Supports an inheritance model	Does not support an inheritance model

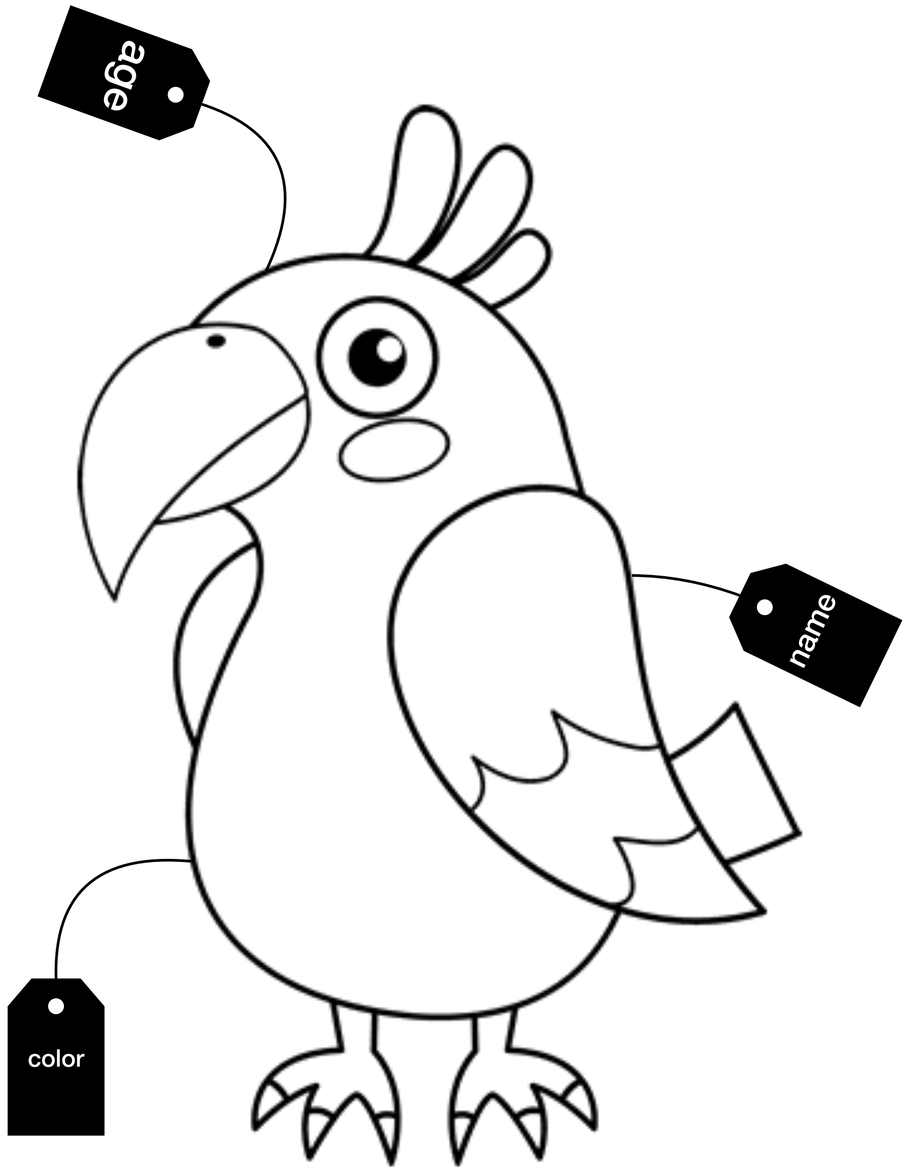
What is a `class`?

A “blueprint” or a definition of your object.

Classes

W3/S2/Birds/Parrot.py

- A class is a blueprint for the object.
- We can think of class as a sketch of a parrot with labels.
  - It contains some details about the parrot, the **name**, **age**, **color,** etc.
  - **Think of those details as attributes of the class.**
- We can also define actions as methods.



Objects

W3/S2/Birds/Parrot.py

- An object or **instance** is an instantiation of a class.
- When a class is defined, only the description for the object is created.
  - Therefore, no memory or storage is allocated.
- **We do not have an instance before we create it!**

Resources

- https://docs.python.org/3.8/tutorial/classes.html
- https://docs.python.org/3.8/tutorial/classes.html#class-definition-syntax
- https://www.python-course.eu/object_oriented_programming.php
- https://www.tutorialspoint.com/python/python_classes_objects.htm