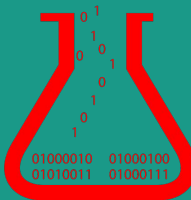

CLASSES AND OOP



What is OOP? (Object-Oriented Programming)

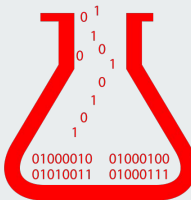
Objects have two parts

- Attributes
- Methods (Behavior)

Example: Cow is an object,

Attributes of a cow: name, breed, size

Behavior of a cow: eating, mooing, sleeping



What is OOP? (another viewpoint)

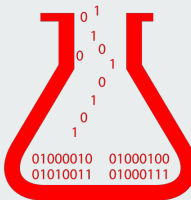
Objects have two parts ← NOUNS

- Attributes ← ADJECTIVES
- Methods (Behavior) ← VERBS

Example: Cow is an object,

Attributes of a cow: name, breed, size

Behavior of a cow: eating, mooing, sleeping



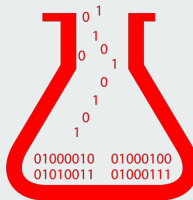
What is OOP? (Object-Oriented Programming)

Classes are used to define the object and create a template for the object you're creating

Class definition is like a cookie cutter

The object is the cookie

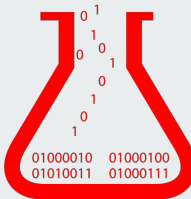
Each cookie cutter can generate many distinct cookies



Methods

Methods are functions defined inside the body of a class. They are used to define the behaviors of an object.

These are like functions that belong to (or “are bound to”) a class



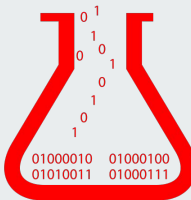
Attributes



Attributes are values (data) that are defined inside the body of a class.

Attributes are bound to each individual object.

The `__init__()` statement defines required attributes



The structure of a class definition

```
class Cow:
    species = "moo cow"
    def __init__(self, name):
        self.name = name
    def get_species(self):
        return self.species
```

The class definition

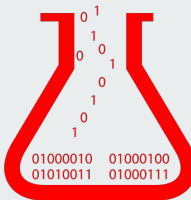
Class level attribute

Initialization statement

Instance level attribute

Class method

This is returned when the method is run

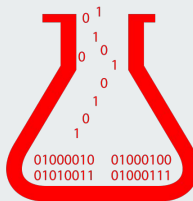


The structure of a class definition

```
class Cow:
    species = "moo cow"
    def __init__(self, name):
        self.name = name
    def get_species(self):
        return self.species
```

"self" brings the current object into
the method

A required argument

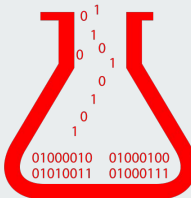


Instantiation and use of the class

We make the cow object “Betsy” and get her species

```
10 Betsy = Cow("Betsy")
11 #note that the name argument is now required when creating a Cow()
12 print("{} is a {}".format(Betsy.name, Betsy.get_species()))
```

Betsy is a moo cow

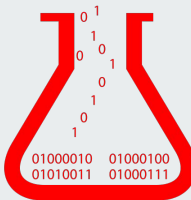


OOP Principles



The basic principles of OOP in python

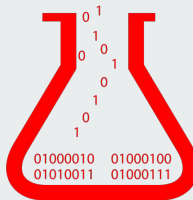
- Encapsulation
- Abstraction
- Inheritance
- Polymorphism



Encapsulation



Encapsulation is grouping similar data and functions into a group to make them easier to access and use.

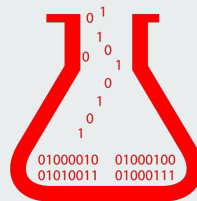


Abstraction



Abstraction is the concept of hiding irrelevant details. In other words, make complex system simple by hiding the unnecessary detail from the user.

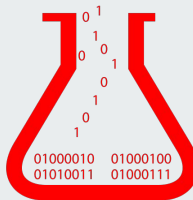
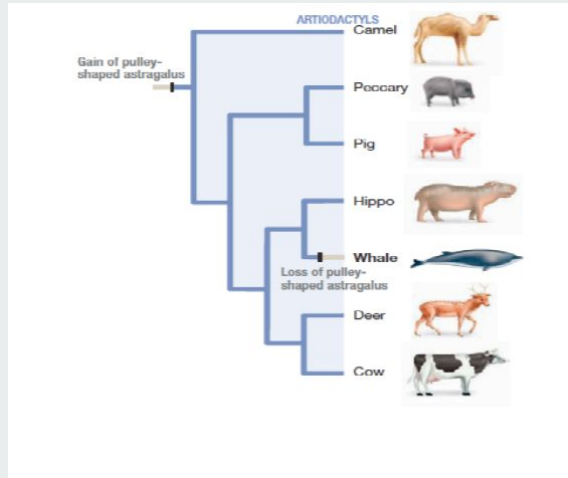
In Python, you can denoted a private attribute or method by using “_” or “__”.



Inheritance

Inheritance is the ability to create a new class using the existing details of an existing class without changing it.

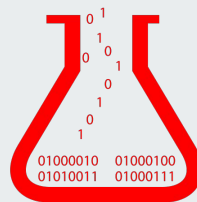
Think of this as a (base class)parent and (derived class)child class.



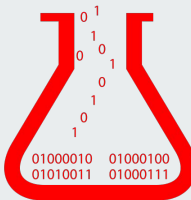
Polymorphism



This is an ability in OOP to use a common interface for multiple data types by defining how data types will interact with each other you can use a combination of this and inheritance to extract away the interaction.



Go to Jupyter Notebooks Examples



Practice Problem I

Write a program that computes the value of $a+aa+aaa+aaaa$ with a given digit as the value of a .

Suppose the following input is supplied to the program:

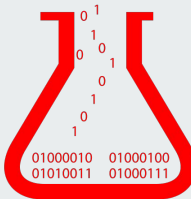
9

Then, the output should be:

11106

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.



Practice Problem II

Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:

D 100

W 200

D means deposit while W means withdrawal.

Suppose the following input is supplied to the program:

D 300

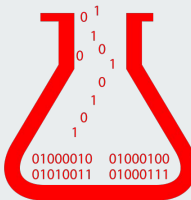
D 300

W 200

D 100

Then, the output should be:

500



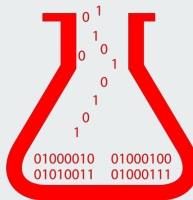
Practice Problem III

A website requires the users to input username and password to register. Write a program to check the validity of password input by users.

Following are the criteria for checking the password:

- At least 1 letter between [a-z]
- At least 1 number between [0-9]
- At least 1 letter between [A-Z]
- At least 1 character from [\$#@]
- Minimum length of transaction password: 6
- Maximum length of transaction password: 12

Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.



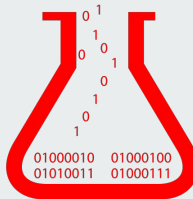
Practice Problem III Example

If the following passwords are given as input to the program:

ABd1234@1,a F1#,2w3E*,2We3345

Then, the output of the program should be:

ABd1234@1



Practice Problem IV

You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is:

- 1: Sort based on name;
- 2: Then sort based on age;
- 3: Then sort by score.

The priority is that name > age > score.

If the following tuples are given as input to the program:

Tom,19,80

John,20,90

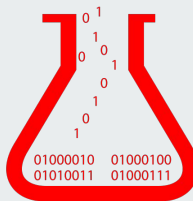
Jony,17,91

Jony,17,93

Json,21,85

Then, the output of the program should be:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]



Practice Problem V

Define a class which has at least two methods:

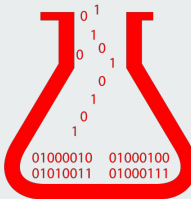
getString: to get a string from console input

printString: to print the string in upper case.

Also please include simple test function to test the class methods.

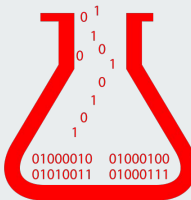
Hints:

Use `__init__` method to construct some parameters



Practice Problem VI

- Write a Pig class. Then, create two child classes (inherit from the Pig class) for WildHog and FarmPig. WildHog should be able to charge/attack. FarmPig should be able to sleep. Make sure all classes have a species, a name, and at least 2 methods. (To keep things simple, do not use hidden attributes.)
- All pigs can Oink and Eat (so these should be part of the parent class). It's up to you how you implement these methods, and what they do.



Questions?



Contact:

Denis Vrdoljak
denis@bds.group
