Object Oriented Programming

May 11, 2022

Basic data types

* type(5) ⟶ int

  print (type(5)) ⟶ class 'int' — data type

* type("hello") ⟶ string / str — data type
* type(True) ⟶ bool — data type
* type(hello) ⟶ X

  hello is a variable.

Each variable / data has its own data type;

Composite data types

list, dict, set } composite

l = [1, 2, 3, 4]

print (type(l)) ⟵ class 'list' — data type

Complex numbers: (2+3i)

(2+3i) + (4+6i)

```
def complex-sum (a, b)
    return a+b
```

a+ib, c+id

```
def complex-sum (a,b, c,d)
    return a+c, b+d
```

a+c + i(b+d)

```
l = list()
l = [1,2,3]
```

```
c1 = complex(1, 2)
c2 = complex(3, 4)
```

```
def add (c1, c2):
    return c1+c2
```

complex

add
subtract

** Object-Oriented Programming **

** Procedural programming

↓

Using functions and logic as the crux of your program.

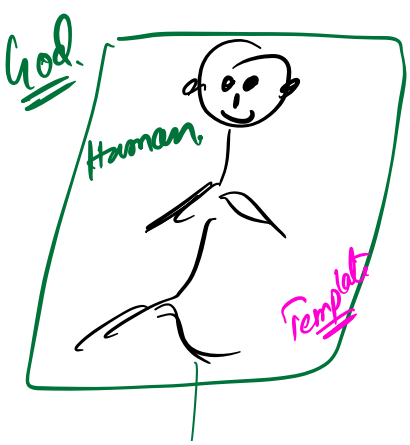
* OOP. → approach to write your programs in such a way so as to arrange your code around data and objects.

Class / Objects

A class is a blueprint -

Where object is an instance of the class.

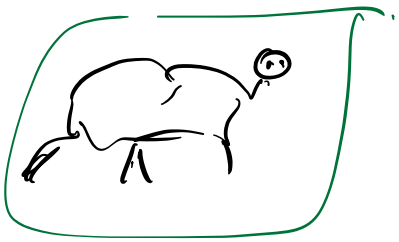
Humans



template

↓

define the rules of how the object should look like



Human 1 ← 2 eyes, 2 legs.
Human 2 ←
Human 3. ←
⋮
Human 10 ←

object Dog 1 =
object Dog 2 =
object Dog 3 =

4 legs
0 hands.
2 eyes
1 tail

← Template

accident.
1 leg was damaged.

Memory

object1: 4 1 1
object2: 4 3 1

Class & object

Blueprint

Several instances of this class are possible.
Each instance follows the same behaviour as defined in the template, but all are independent.

Google Colab
(Python Notebook) Link.

https://colab.research.google.com/drive/1qUlMUm9Bg7Tsi5-iEk0FSWooIeH1cvIe?usp=sharing