Types of Variables and Methods

In python there are different types of variables and methods

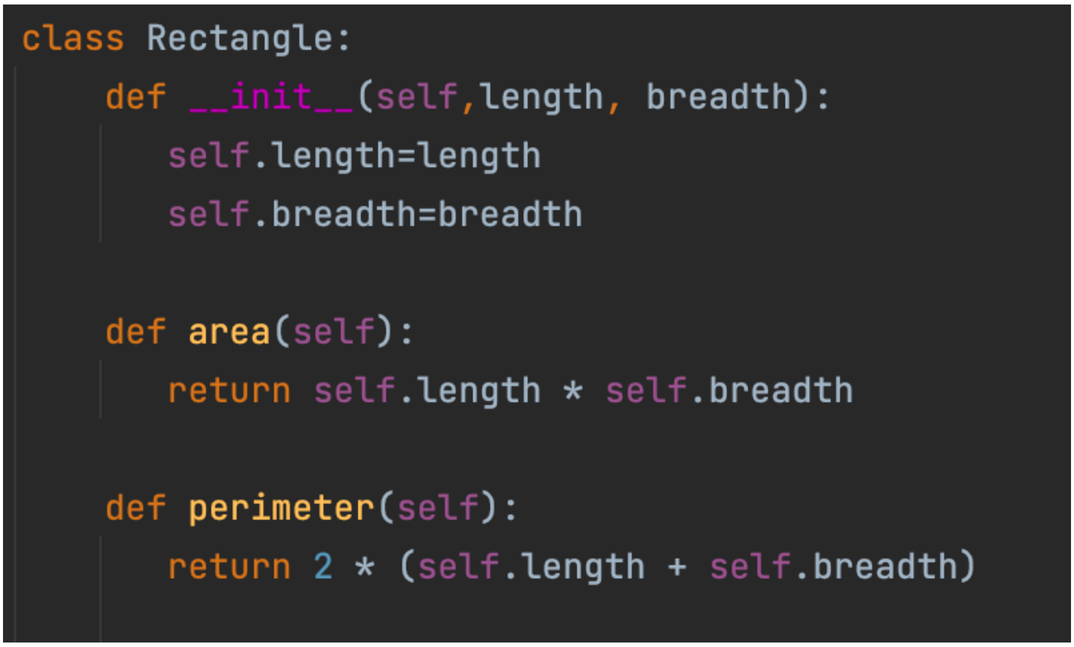
Variables:

- Instance variable
- Static variable

Methods:

- Instance method
- Class method
- Static method

Instance :

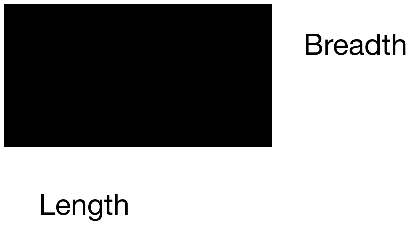


```
class Rectangle:
    def __init__(self,length, breadth):
        self.length=length
        self.breadth=breadth

    def area(self):
        return self.length * self.breadth

    def perimeter(self):
        return 2 * (self.length + self.breadth)
```

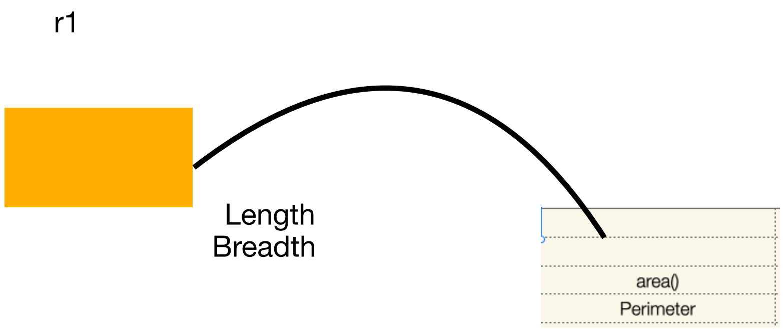
Example: Rectangle



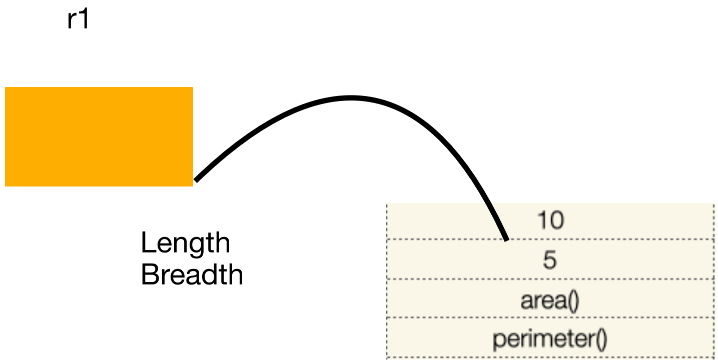
area= length*breadth

Creating a variable for rectangle
r1= rectangle()# automatically calls the constructor of the class

In the memory it looks like



It will form like this



r2= Rectangle(15,7) # another object

So again one more object screamed in memory

These are called instance of class

Each instance will have its own members

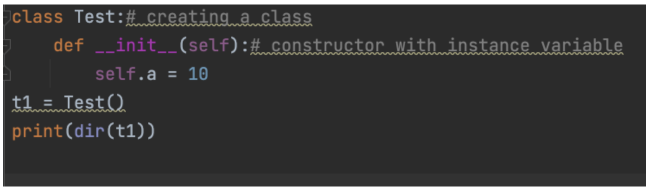
The parameter that is always taken by instance method is self and using self they access instance variable and the methods which are accessing them are called as instance methods.

Ways of creating instance variables.

So, there are multiple ways we can declare instance variable.

Program:

Input:



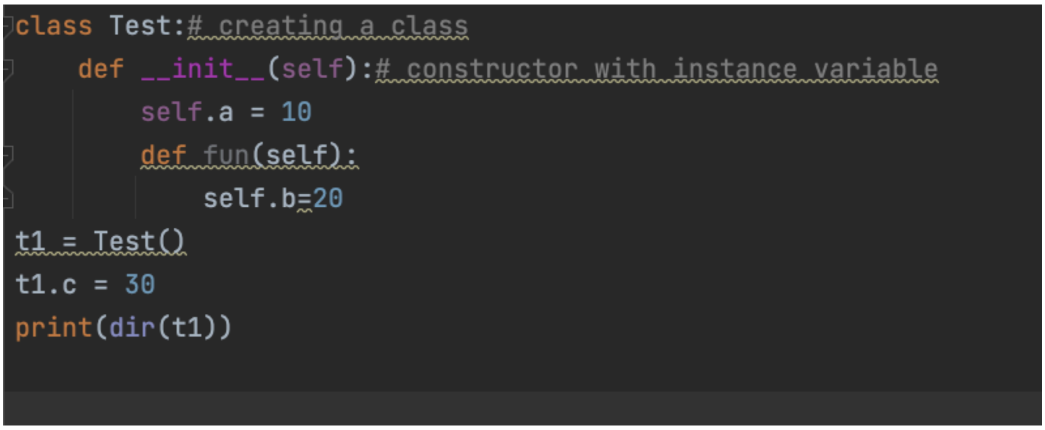
```
class Test:# creating a class
    def __init__(self):# constructor with instance variable
        self.a = 10
t1 = Test()
print(dir(t1))
```

Output:
['__doc__', '__init__', '__module__', ‘a']

So it is having one variable that is a

Program

Input:



```
class Test:# creating a class
    def __init__(self):# constructor with instance variable
        self.a = 10
        def fun(self):
            self.b=20
t1 = Test()
t1.c = 30
print(dir(t1))
```

Output:
['__doc__', '__init__', '__module__', 'a', 'c']

It is still having a and function but b is not there for that function should be called
We have added one more variable to it so, it means we can declare and initialise
Instance variable in init method also and other method also.

Instead of self we can use the variable name and.
Self is not a keyword it can be anything. The first we are writing is constructor is called as (instance variable)

We can give anytime instead of self

If a method is accessing instance variable then that method becomes instance method.