PYTHON

CLASSES

Overview:

- learn the advantages of object-oriented programming
- explain how to define a class

Introduction

- class: user defined data type (data and methods)
- object: instance of a class
- Python programs are built with objects
- classes define objects
- defined using *class* keyword
- new classes can be derived and methods inherited

Example: Class *Sphere*

- *sphere* - class
- green_ball - an instance with radius 2
- red_ball - an instance with radius 3
- instances are distinct

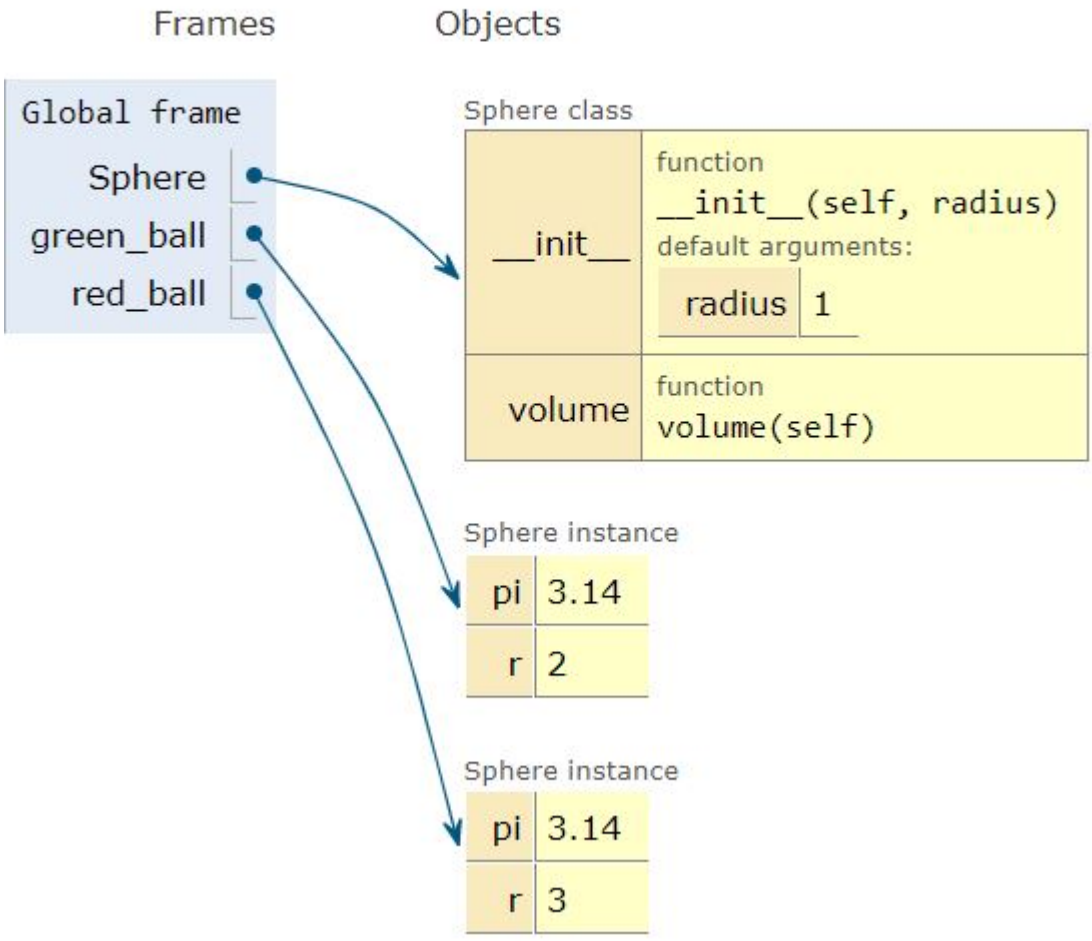
Example: Class *Sphere*

```
class Sphere():
    def __init__(self, radius = 1):
        self.pi = 3.14
        self.r = radius

    def volume(self):
        return 4 * self.pi * self.r**3 / 3

green_ball  = Sphere(2)
red_ball    = Sphere(3)
print('green_ball volume:', green_ball.volume())
print('red_ball   volume:', red_ball.volume())
```

- *__init__*() is a constructor
- *volume*() is a method
- r and pi are object variables
- volume for sphere: $4\pi r^3/3$

Details for Class *Sphere*

```
green_ball volume: 33.49
red_ball   volume: 113.04
```



Exercise(s):

- define a class *Circle*
- takes *radius* as constructor
- default radius is 3
- has a single method *area*()

```
small_circle = Circle()      # radius 2
large_circle = Circle(10)    # radius 10
small_area   = small_circle.area()
large_area   = large_circle.area()
```