PES UNIVERSITY

Strings in Python

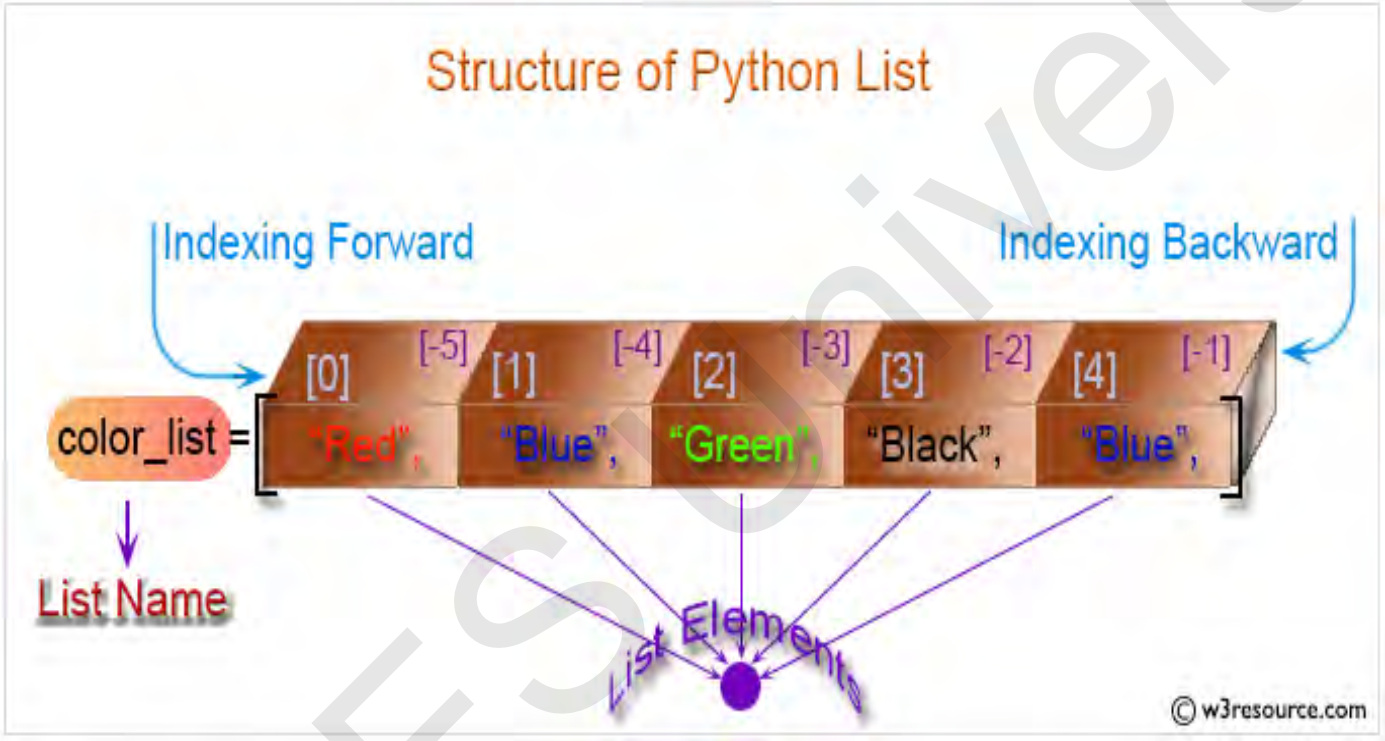
str="BANGALORE”

B	A	N	G	A	L	O	R	E
0	1	2	3	4	5	6	7	8

Accessing string elements
Str[0] returns B
Str[1] returns A
---

count()
format()
index()
islower()
join()
lower()
upper()
strip()
replace()
split()

Lists in Python



Classes in Python: User defined data types

```
class class_name :
        data members
        methods
```

Example:

```
class Circle
        radius
        findArea()
        findPerimeter()
```

- Classes provide a means of bundling data and functions together.
- it is a collection of variables and functions
- variables are called as data members and
- Function are called as member functions or **methods**
- Classes are used to represent real world entities

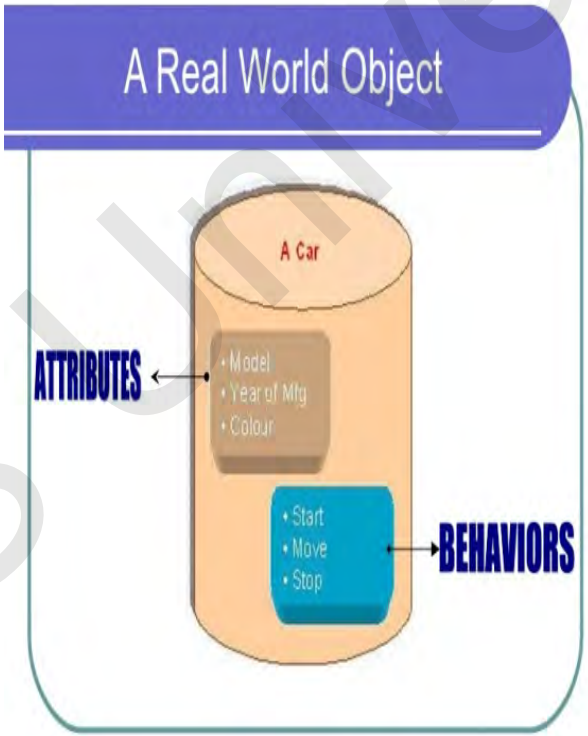
Real world objects/entities have two major things
1)state/attributes(what it is)
2)Behaviour/Actions(what it does)
Pyth0n classes can be used to simulate real world entities

```
class car :
        year; make;
        speed;

        start()
        accelerate();
        brake();
```

```
class flower :
        name;
        color;

        makegarlend()
```



Object-Oriented Programming

Python is an object-oriented programming language

- Encapsulation

  (security to data)

- Inheritance

  (reusability)

- Polymorphism

  (Having many forms)

These are also called as

pillars of object-oriented development

Classes: user defined data types

```
class person:
    def __init__(self,x,y):
        self.name=x
        self.age=y
    def display(self):
        print(self.name,self.age)

p1=person("john",30)
p2=person("Ram",32)

print(p1.name,p1.age)
print(p2.name,p2.age)

p1.display()
```

Classes in Python

Class is a blueprint of a house

Objects

Person class

```
class person:
    def __init__(self,x,y):
        self.name=x
        self.age=y
    def display(self):
        print(self.name,self.age)
```

p1=person("john",30)

p2=person("Ram",32)

WAP to find area and perimeter of a circle using classes

```
class Circle:
    def __init__(self,r):
        self.radius=r
    def findarea(self):
        print(3.14*self.radius*self.radius)
    def findperimeter(self):
        print(2*3.14*self.radius)


c=Circle(1)
c.findarea()
c.findperimeter()
```

Using classes, write a program to find distance between two points

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def disp(self):
        print ("x : ", self.x)
        print ("y : ", self.y)
    def findDistance(p1,p2):
        res=pow(p2.x-p1.x,2)+pow(p2.y-p1.y,2)
        print(math.sqrt(res))

p1 = Point(3, 4)
p2= Point(4, 3)
p1.findDistance(p2)
```