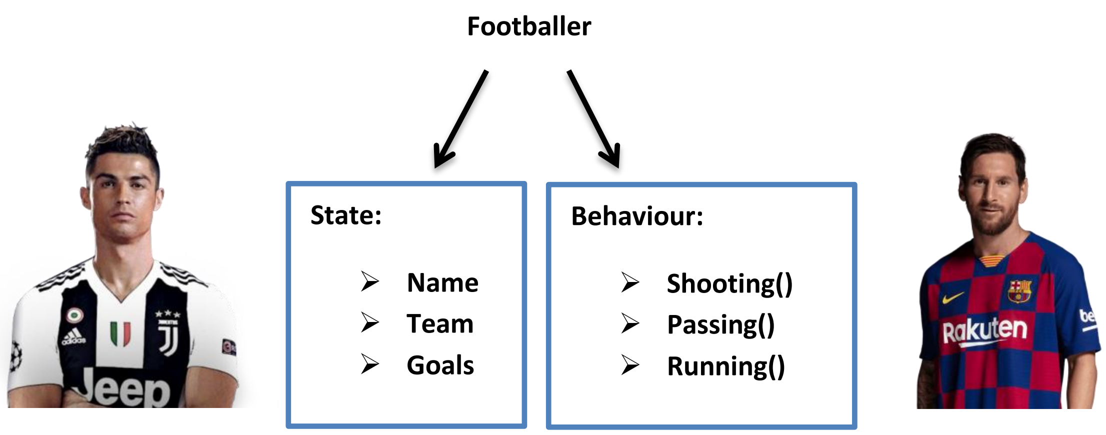
Python Fundamentals
day 45

Today's Agenda

- Object orientation contd…
- Creating instance variables

Object orientation contd…

In previous session we got to know multiple objects can be created. But we had created duplicate objects. So let us now consider an example where we shall have different objects of the same type.



Let us see how to write the python code for this particular example.

```python
class FootBaller:
    def __init__(self,name,team,goals):
        self.name=name
        self.team=team
        self.goals=goals

    def shooting(self):
        print(self.name,'is shooting')

    def passing(self):
        print(self.name,'is passing')

    def running(self):
        print(self.name,'is running')

def main():
    cr=FootBaller('Cristino','Juventus','746')
    print(cr.name)
    print(cr.team)
    print(cr.goals)
    cr.shooting()
    cr.passing()
    cr.running()

    messi=FootBaller('Messi','Barcelona','700')
    print(messi.name)
    print(messi.team)
    print(messi.goals)
    messi.shooting()
    messi.passing()
    messi.running()

if __name__ == '__main__':
    main()
```

Output:

```
In [2]: runfile('C:/Users/rooman/OneDrive/Desktop/python/
test.py', wdir='C:/Users/rooman/OneDrive/Desktop/python')
Cristino
Juventus
746
Cristino is shooting
Cristino is passing
Cristino is running
Messi
Barcelona
700
Messi is shooting
Messi is passing
Messi is running
```

Great!! But we Python is all above reducing lines of code and making it more efficient. So let us see how more efficient form of the above code.

```python
class FootBaller:
    def __init__(self,name,team,goals):
        self.name=name
        self.team=team
        self.goals=goals

    def shooting(self):
        print(self.name,'is shooting')

    def passing(self):
        print(self.name,'is passing')

    def running(self):
        print(self.name,'is running')

    def display(self):
        print(self.name)
        print(self.team)
        print(self.goals)

def main():
    cr=FootBaller('Cristino','Juventus','746')
    cr.display()
    cr.shooting()
    cr.passing()
    cr.running()

    messi=FootBaller('Messi','Barcelona','700')
    messi.display()
    messi.shooting()
    messi.passing()
    messi.running()

if __name__ == '__main__':
    main()
```

Output:

```
In [3]: runfile('C:/Users/rooman/OneDrive/Desktop/python/
test.py', wdir='C:/Users/rooman/OneDrive/Desktop/python')
Cristino
Juventus
746
Cristino is shooting
Cristino is passing
Cristino is running
Messi
Barcelona
700
Messi is shooting
Messi is passing
Messi is running
```

From the above example we got to know the following things

- ❖ For the given class we can create multiple objects. And objects can be initialised differently.
- ❖ There are two different types of functions
    1. __new__ : Is a constructor.
    2. __init__ : Is a initialiser.

Creating instance variables

There are two ways in which we can create instance variables.

- ❖ During object creation
- ❖ After object creation

```
class FootBaller:
    def __init__(self,name,team,goals): # before object creation
        self.name=name
        self.team=team
        self.goals=goals

    def shooting(self):
        print(self.name,'is shooting')

    def passing(self):
        print(self.name,'is passing')

    def running(self):
        print(self.name,'is running')

    def display(self):
        print(self.name)
        print(self.team)
        print(self.goals)
        print(self.age)
        print(self.jersey_no)

def main():
    cr=FootBaller('Cristino','Juventus','746')
    cr.age=35 #after object creation
    cr.jersey_no=7 #after object creation
    cr.display()
    cr.shooting()
    cr.passing()
    cr.running()

if __name__ == '__main__':
    main()
```

Output:

```
In [4]: runfile('C:/Users/rooman/OneDrive/Desktop/python/
test.py', wdir='C:/Users/rooman/OneDrive/Desktop/python')
Cristino
Juventus
746
35
7
Cristino is shooting
Cristino is passing
Cristino is running
```

There is also another way of creating instance variables after object creation, by using built-in method setattr(). Similarly we also have getattr() and hasattr(). Let us see how to use it

```python
class FootBaller:
    def __init__(self,name,team,goals): # before object creation
        self.name=name
        self.team=team
        self.goals=goals

    def shooting(self):
        print(self.name,'is shooting')

    def passing(self):
        print(self.name,'is passing')

    def running(self):
        print(self.name,'is running')

    def display(self):
        print(self.name)
        print(self.team)
        print(self.goals)
        print(self.age)
        print(self.jersey_no)

def main():
    cr=FootBaller('Cristino','Juventus','746')
    setattr(cr,'age',35) #after object creation
    setattr(cr,'jersey_no',7)#after object creation
    print(cr.name)
    print(getattr(cr,'name'))
    print(hasattr(cr,'name'))
    print(hasattr(cr,'gender'))

if __name__ == '__main__':
    main()
```

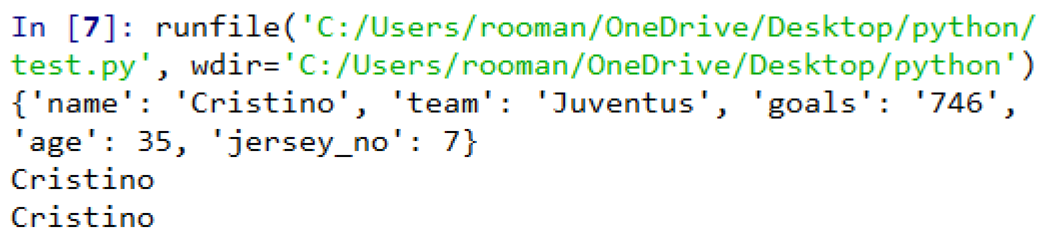
Output:

```
In [6]: runfile('C:/Users/rooman/OneDrive/Desktop/python/
test.py', wdir='C:/Users/rooman/OneDrive/Desktop/python')
Cristino
Cristino
True
False
```

Another interesting thing to know is all the instance variables are present in a dictionary. The name of this dictionary is __dict__. Let us see is this actually true.

```
class FootBaller:
    def __init__(self,name,team,goals): # befor
        self.name=name
        self.team=team
        self.goals=goals

    def shooting(self):
        print(self.name,'is shooting')

    def passing(self):
        print(self.name,'is passing')

    def running(self):
        print(self.name,'is running')

    def display(self):
        print(self.name)
        print(self.team)
        print(self.goals)
        print(self.age)
        print(self.jersey_no)

def main():
    cr=FootBaller('Cristino','Juventus','746')
    setattr(cr,'age',35) #after object creation
    setattr(cr,'jersey_no',7)#after object crea
    print(cr.__dict__) # printing the dictionar
    print(cr.name)
    print(cr.__dict__['name'])


if __name__ == '__main__':
    main()
```

Output:



```
In [7]: runfile('C:/Users/rooman/OneDrive/Desktop/python/
test.py', wdir='C:/Users/rooman/OneDrive/Desktop/python')
{'name': 'Cristino', 'team': 'Juventus', 'goals': '746',
'age': 35, 'jersey_no': 7}
Cristino
Cristino
```