Python Tutorial on Class

Leo Zipeng Lin

March 13, 2020

Friendly Alert

Now we talk about python class.

1 Why do we need class?

Friendly Alert

If you did AP Computer Science A, you can probably skip this part (If you remember :))

Sometimes we need to operate through a same type of *objects*. [2] For instance, if the school wants to ’take care” of the students’, then it has to know the **Name**, **Age**, **Grade**, and other characters. Of course, we are able to assign those ’character’ one by one, like

```python
Student1_name="Eric"
Student1_Age=12
Student2_name="Mary"
Student2_Age=14
# Start a report
print(Student1_name+" "+str(Student1_Age))
print(Student2_name+" "+str(Student2_Age))
```

Apparently, this is **efficient** if there is 200 students. We can think of a way like this: what if we just treat the people as **Objects**? For instance, we treat all the students as ”human beings with age number, gender, name, and other information”, and we just create one ’class’ for all of the students. That’s basically why class is so useful

To say this in another way, you can create your own type!

2 Example

```
# Use object, not Object
class student(object):
    def __init__ (self, age, name, grade, gender):
        self.name  = name
        self.age   = age
        self.grade =grade
        self.gender=gender
    def announcement(self):
        if (self.grade==11):
            print("Junior")
        elif (self.grade==12):
            print("Senior")
        elif (self.grade==10):
            print("Sophomore")
        elif (self.grade==9):
            print("Freshmen")
        elif (self.grade>6):
            print("Middle Schooler")
        else:
            print("Young kid")

leolin=student(12,"Leo Lin", 11, "Preferred not to tell")
leolin.announcement()
```

Right now, we are able to assign each student with their characteristics. Consider the effect of creating a class when we need to write a system to sort students in their grade. Since the grade is assigned, we are able to access to the data easily. [1]

3 Explanation for the code

Reader's Complain

Hey, we don't even understand the code!

Don't worry, now we are going to go through the code.

```
        class student(object):
```

In this line, we create a new **Class** called student.
Pay attention to **object**, now we are going to explain what this is.

```
        def __init__ (self , age , name , grade , gender ):
```

In this line, we create a **function** (read the chapter before) called **init**. It is used to **initiate** the object. All the classes have a function called 'init', which is always executed when the class is being initiated. When we execute this function, we do the operations to assign **basic values**. Compared to other functions, this one is executed *automatically*.

```
  self.name=name
        self.age=age
        self.grade=grade
        self.gender=gender
```

Here, the **self** is a parameter that refers to the current object of the class, here it is the **student**. It does not have to be the word called 'self', we can also use **any word you want**.

4 Exercises

1. Try to write a **object** for all the countried
   - Try to write a function in case to create a country.
   - Write a function to update the population (like after the cencus)

References

[1] Python Software Foundation. *Python Tutorial Classes*. 2020. URL: https://docs.python.org/3/tutorial/classes.html.

[2] Wikipedia contributors. *Object-oriented programming — Wikipedia, The Free Encyclopedia*. [Online; accessed 4-March-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?title=Object-oriented_programming&oldid=940209067.