

Name: Mariam Hussien Ahmed

Dunder methods

1) Introduction:

Dunder methods are names that are preceded and succeeded by double underscores, hence the name dunder. They are also called magic methods and can help override functionality for built-in functions for custom classes. Implementing dunder methods for classes is a good form of Polymorphism. If you have ever created a class in Python and used the init function, then you have already been using dunder methods.

2) Dunder methods for our class:

Before move forward, please make sure that your indentation is correct.

This methods will be discussed below are methods that belong to the class we created and should be indented better.

2.1) init:

This is a method you should use it if you have worked with classes. The init method is used to create an instance of the class.

```
def __init__(self, names):  
    if names:  
        self.names = names.copy()  
        for name in names:  
            self.versions[name] = 1  
    else:  
        raise Exception("Please Enter the names")
```

The init method defined above accepts a list of names as parameters and stores them in a list of class names. In addition, it also fills the dictionary of editions. We have also checked the list of names.

If the list is empty, an exception is raised. Which is how to use the init method.

2.2) str:

The str method is benefit when we want to use instances of our class in a print statement. As discussed before, it usually returns a memory object. But we can exceed the str method to implement our desire.

```
def __str__(self):  
    s = "The current softwares and their versions are listed  
below: \n"  
    for key,value in self.versions.items():  
        s+= f"{key} : v{value} \n"  
    return s
```

The str method above returns programs and their versions. Also how we can call the method

```
print(p)
```

2.3) len:

The len method in a dictionary returns the number of key-value pairs in a dictionary or the number of elements in a list.

For our class, we may define a len method as well.

```
def __len__(self):  
    return len(self.names)
```

To return the number of software, we are using the list's built-in len method.

The following example show how we use our class' len method.

```
print(len(p))
```

2.4) new:

The new operator is used to generate a new instance of a class in languages like Java and C#. Python automatically calls the magic `__new__()` method before the `__init__()` method. The `__init__` method initializes the new object that the `__new__()` method returns ().

2.5) add:

The magic method `__add__` is used to add attributes to the class instance. Pretend object1 and object2 are members of classes A and B, respectively, and that both of these classes have an attribute with the name "a" that contains an integer for each class. The `__add__` function adds the characteristics of these objects, such as `object1.a + object2.a`, when the operation `object1 + object2` is complete. As you can see in the first line of code, we have to re-initialize the string class. Then , we used the magic method to start the object. The `__add__` method was then used to print the string object. After the driver code, which is the following line of code, the object is constructed. In the final line of code, a String was concatenated.

3) Conclusion:

Dunder methods are extremely strong Python intrinsic methods that can be used to develop complex classes. This lesson provides a step-by-step guide of Python methods and how to use them, complete with examples.

