**\*Args and \*\*kwargs**

**\*Args:** it is a variable wrote in function definition

If you do not know how many arguments that will be passed into your function, add a \* before the parameter name in the function definition.

**\*\*kwargs:** it is the same \*args but the function can confirm keyword argument . Treats dictionary treatment

If you do not know how many keyword arguments that will be passed into your function, add two asterisk: \*\* before

the parameter name in the function definition.

**Dunder Methods**

Dunder methods in Python are the methods having two prefix and suffix underscores in the method name. Dunder here means “Double Under (Underscores)”. These are commonly used for operator overloading. Few examples for dunder methods are:

\_\_init\_\_, \_\_add\_\_, \_\_len\_\_, \_\_repr\_\_ etc.

The \_\_init\_\_ method for initialization is invoked without any call, when an instance of a class is created, like constructors in certain other programming languages such as C++, Java, C#, PHP etc. These methods are the reason we can add two strings with ‘+’ operator without any explicit typecasting.

**Abstraction and Encapsulation**

Data abstraction and encapsulation are synonymous as data abstraction is achieved through encapsulation.

**Abstraction**

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things, so that the name captures the basic idea of what a function or a whole program does.

**Encapsulation**

Encapsulation is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.