**\*args and \*\*kwargs in Python**

**Special Symbols Used for passing arguments: -**

* \*args (Non-Keyword Arguments)
* \*\*kwargs (Keyword Arguments)

## **What is Python \*args?**

The special syntax in function definitions in Python allows a function to pass a number of arguments to a variable. It does so by passing a non-key-worded list of arguments.

1-The symbol \* is used to represent the number of arguments that a function takes in. It is typically used with the word args.

2- With \*args, a function can take in more arguments than it previously defined. This allows it to add any number of extra arguments to its formal parameters.

3-For example, we want to make a multiply function that takes any number of arguments and is able to multiply them all together. It can be done using \*args.

4-Using the \*, the variable that we associate with the \* becomes an iterable meaning you can do things like iterate over it, run some higher-order functions such as map and filter, etc.

## **What is Python \*\*kwargs?**

The special syntax *\*\*kwargs* in function definitions in python is used to pass a keyworded, variable-length argument list. We use the name *kwargs* with the double star. The reason is that the double star allows us to pass through keyword arguments (and any number of them).

1-A keyword argument is where you provide a name to the variable as you pass it into the function.

2-One can think of the *kwargs* as being a dictionary that maps each keyword to the value that we pass alongside it. That is why when we iterate over the *kwargs* there doesn’t seem to be any order in which they were printed out.

# Encapsulation in Python

Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To prevent accidental change, an object’s variable can only be changed by an object’s method. Those types of variables are known as **private variables.**

A class is an example of encapsulation as it encapsulates all the data that is member functions, variables, etc.

Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of the sales section and the employees that can manipulate them are wrapped under a single name “sales section”. Using encapsulation also hides the data. In this example, the data of the sections like sales, finance, or accounts are hidden from any other section.

# Abstract Classes in Python

An abstract class can be considered as a blueprint for other classes. It allows you to create a set of methods that must be created within any child classes built from the abstract class. A class which contains one or more abstract methods is called an abstract class. An abstract method is a method that has a declaration but does not have an implementation. While we are designing large functional units we use an abstract class. When we want to provide a common interface for different implementations of a component, we use an abstract class.   
    
**Why use Abstract Base Classes :**   
By defining an abstract base class, you can define a common Application Program Interface(API) for a set of subclasses. This capability is especially useful in situations where a third-party is going to provide implementations, such as with plugins, but can also help you when working in a large team or with a large code-base where keeping all classes in your mind is difficult or not possible.   
    
**How Abstract Base classes work :**   
By default, Python does not provide abstract classes. Python comes with a module that provides the base for defining Abstract Base classes(ABC) and that module name is ABC. ***ABC*** works by decorating methods of the base class as abstract and then registering concrete classes as implementations of the abstract base. A method becomes abstract when decorated with the keyword @abstractmethod. For Example