1)*args (Non-KeywordArguments)

What is Python *args?

To pass a variable number of arguments to a function in Python, use the special syntax *args in the function specification. It is used to pass a variable-length, keyword-free argument list.

- By convention, the sign * is frequently used with the word args in the syntax for taking in a variable number of arguments.
- You can accept additional arguments using *args than the number of formal arguments you previously defined. Any number of additional arguments can be added to your current formal parameters using *args (including zero extra arguments).
- For example, we want to create a multiply function that can multiple any number of inputs continuously. You can carry it out using *args.
- By using the *, the variable we associate with the * is made Expected, allowing you to run higher-order functions like map and filter as well as other operations on it.
- A function can accept a range of parameters in a few different ways. For those who are used with collecting, the first method is commonly the most natural. You simply provide your function with a list or set of all the arguments. You may therefore supply a list of all the numbers you need to add to my sum().

2)**kwargs (Keyword Arguments)

What is Python **kwargs?

**kwargs works just like *args, but instead of accepting positional arguments it accepts keyword (or named) arguments.

Python function definitions can accept a keyworded, variable-length argument list by using the different syntax **kwargs. With the double star, we use the name kwargs. The double star's ability to pass through keyword arguments is the cause for this (and any number of them).

- A keyword argument is used to give the variable a name before passing it to the function.
- The kwargs can be viewed as a dictionary that matches each term with the value that is passed along with it. Because of this, there doesn't appear to be any sequence in which the kwargs were printed out when we iterate through them.

3) Encapsulation

What is the Encapsulation?

Giving every variable in the project global access is not a good decision when working with classes and managing sensitive data. Without giving the programmer complete access to any of those variables, encapsulation offers a system for us to obtain the necessary variables.

Methods that are defined explicitly for the purpose can be used to update, edit, or delete data from variables. This method of programming has the advantages of better security and control over the input data.

What does Python's encapsulation mean?

Each object-oriented programming language uses the same encapsulation idea. When the ideas are applied to specific languages, the similarity is clear.

Python offers global access to all variables and methods in contrast to languages like Java that only support protected or private access modifiers for variables and methods as:

protected members

The members of a class that can only be accessed from within the class and its subclasses are known as protected members (in C++ and Java). Simply keep to convention and add a single underscore ("_") to the beginning of the member name in Python to achieve this.

Private members

The difference between private and protected members is that a class member that has been declared private should not be used by any base class or by anyone outside of the class. Private example variables that can only be accessed within a class do not exist in Python.
However, add a double underscore ("__") before the member name to define it as a private member.

4) Abstraction

Abstraction is used to hide from users the internal workings of a function. Users only connect with the function's basic implementation; the inner workings are kept secret. The user is know "what function does," but they are ignorant of "how it does."

Simple terms, we all use smartphones and are extremely familiar with its abilities, such as the camera, voice recorder, call-dialing, etc., but we don't understand how these processes work behind the scenes. Let's look at another illustration: showing up the TV's volume using the remote control. We have no idea how key presses change the TV's volume. We only are aware of how to raise the volume by pressing the "+" button.