**What is Python \*args ?**

The special syntax *\*args* in function definitions in python is used to pass a variable number of arguments to a function. It is used to pass a non-key worded, variable-length argument list.

* The syntax is to use the symbol \* to take in a variable number of arguments; by convention, it is often used with the word args.
* What *\*args* allows you to do is take in more arguments than the number of formal arguments that you previously defined. With *\*args*, any number of extra arguments can be tacked on to your current formal parameters (including zero extra arguments).
* 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.
* 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).

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