Functions with Multiple Arguments

03 December 2024

18:37

You can define a function with multiple arguments

An example use case for this is an area function:

A screen shot of a computer program

Description automatically generated

This will use both arguments to execute the function

Key Word vs Non Key Word Arguments

03 December 2024

18:47

Non Key Word Argument (Positional Arguments):

A screenshot of a computer

Description automatically generated

Key Word Argument:

A screenshot of a computer

Description automatically generated

Key word arguments would allow us to do b=2 before a=2 (Non Positional)

Default vs Non-Default Parameters

03 December 2024

18:50

Default parameter is a and non-default parameter is B:

A black screen with colorful text

Description automatically generated

This shows b as a default parameter

However, you can still assign a different value to b through doing b=7 or by just doing print(area(3, 7)

**\*\*\*A default parameter cannot be before a non-default parameter**

Functions with Arbitrary Number of Non-Keyword Arguments

03 December 2024

18:56

Some Functions take different amounts of arguments (For example isinstance takes 2, and print takes infinite)

A computer screen with text and numbers

Description automatically generated

The above is a function with an arbitrary number of non-keyword arguments only using \*args

(x=3 would not work)

If we wanted to define a function that takes an indefinite number of strings:

(includes sorting alphabetically and capitalizing)

def foo(\*args):

args = [x.upper() for x in args]

return sorted(args)

Functions with Arbitrary Number of Keyword Arguments

09 December 2024

15:57

To do the same but for keyword arguments use \*\*kwargs

A black screen with white text and colorful text

Description automatically generated

This will return a dictionary

