Is it possible to receive a “positional argument follows keyword argument” SyntaxError while using the “Fix” function?

Authored by

stats writer

May 12, 2024

A "positional argument follows keyword argument" SyntaxError may occur while using the "Fix" function. This error indicates that a positional argument, which is defined by its position in the function call, has been placed after a keyword argument, which is defined by its name. This is a violation of the correct syntax for function calls and can be resolved by rearranging the arguments to match the correct order. The "Fix" function is designed to correct errors in code, but it is possible for this error to occur if the function is not used correctly.

Fix: SyntaxError: positional argument follows keyword argument

One error you may encounter in Python is:

SyntaxError: positional argument follows keyword argument

This error occurs when you use a positional argument in a function after using a keyword argument.

Here's the difference between the two:

Positional arguments are ones that have no "keyword" in front of them.

Example: my_function(2, 2)

Keyword arguments are ones that do have a "keyword" in front of them.

Example: my_function(a=2, b=2)

If you use a positional argument after a keyword argument then Python will throw an error.

Example: my_function(a=2, 2)

The following example shows how this error may occur in practice.

Example: Positional Argument Follows Keyword Argument

Suppose we have the following function in Python that multiplies two values and then divides by a third:

defdo_stuff(a, b): return a * b / c

The following examples show valid and invalid ways to use this function:

Valid Way #1: All Positional Arguments

The following code shows how to use our function with all positional arguments:

do_stuff(4, 10, 5)

8.0

No error is thrown because Python knows exactly which values to use for each argument in the function.

Valid Way #2: All Keyword Arguments

The following code shows how to use our function with all keyword arguments:

do_stuff(a=4, b=10, c=5)

8.0

Once again no error is thrown because Python knows exactly which values to use for each argument in the function.

Valid Way #3: Positional Arguments Before Keyword Arguments

The following code shows how to use our function with positional arguments used *before* keyword arguments:

do_stuff(4, b=10, c=5)

8.0

No error is thrown because Python knows that the value 4 must be assigned to the *a* argument.

Invalid Way: Positional Arguments After Keyword Arguments

The following code shows how we may attempt to use the function with positional arguments used after keyword arguments:

do_stuff(a=4, 10, 5)

SyntaxError: positional argument follows keyword argument

An error is thrown because we used positional arguments after keyword arguments.

Specifically, Python doesn't know if the 10 and 5 values should be assigned to arguments *b* or *c* so it's unable to execute the function.

The following tutorials explain how to fix other common

errors in Python: