Programiz

TUTORIALS EXAMPLES GET APP

Python exec()

The exec() method executes the dynamically created program, which is either a string or a code object.

The syntax of exec();

```
exec(object, globals, locals)
```

exec() Parameters

The exec() takes three parameters:

- **object** - Either a string or a code object
- **globals** (optional) - a dictionary
- **locals** (optional)- a mapping object. Dictionary is the standard and commonly used mapping type in Python.

The use of `globals` and `locals` will be discussed later in the article.

Return Value from exec()

The exec() doesn't return any value, it returns `None` .

Example 1: How exec() works?

script.py IPython Shell

```python
program = 'a = 5\nb=10\nprint("Sum =", a+b)'
exec(program)
```

Run

Powered by DataCamp

When you run the program, the output will be:

```
Sum = 15
```

Here, the string object program is passed to exec() which executes the program. globals and locals are omitted in this case.

Example 2: Allow user to provide input

```python
program = input('Enter a program:')
exec(program)
```

When you run the program, the output will be:

```
Enter a program: [print(item) for item in [1, 2, 3]]
1
2
```

```
3
```

If you want to take Python code from the user which allows multiline code (using `'\n'`), you can use `compile()` method before using exec().

Learn more about compile() method in Python.

Be careful while using exec()

Consider a situation, you are using a Unix system (macOS, Linux etc) and you have imported `os` module. The os module provides portable way to use operating system functionalities like: read or write a file.

If you allow users to input a value using `exec(input())`, the user may issue commands to change file or even delete all the files using command `os.system('rm -rf *')`.

If you are using `exec(input())` in your code, it's a good idea to check which variables and methods the user can use. You can see which variables and methods are available using dir() method.

script.py IPython Shell

```python
from math import *
exec('print(dir())')
```

Run

Powered by DataCamp

When you run the program, the output will be:

```
['In', 'Out', '_', '__', '___', '__builtin__', '__builtins__', '__name__'
```

Restricting the Use of Available Methods and Variables in exec()

More often than not, all the available methods and variables used in exec() may not be needed, or even may have a security hole. You can restrict the use of these variables and methods by passing optional globals and locals parameters (dictionaries) to the exec() method.

1. Both globals and locals parameters are omitted

If both parameters are omitted (as in our earlier examples), the code expected to be executed by exec() is executed in the current scope. You can check the available variables and methods using the following code:

```
exec('print(dir())')
```

Passing globals parameter; locals parameter is omitted

The globals and locals parameters (dictionaries) are used for global and local variables respectively. If the locals dictionary is omitted, it defaults to globals dictionary. Meaning, globals will be used for both global and local variables.

Note: You can check the current global and local dictionary in Python using globals() and locals() built-in methods respectively.

Passing empty dictionary as globals parameter

script.py IPython Shell

```python
from math import *
exec('print(dir())', {})

# This code will raise an exception
# exec('print(sqrt(9))', {})
```

Run

Powered by DataCamp

If you pass an empty dictionary as globals , only the __builtins__ are available to the object (first parameter to the exec()). Even though we have imported math module in the above program, trying to access any of the functions provided by the math module will raise an exception.

When you run the program, the output will be:

```
['__builtins__']
```

Making Certain Methods available

script.py IPython Shell

```python
from math import *
exec('print(dir())', {'sqrt': sqrt, 'pow': pow})

# object can have sqrt() module
exec('print(sqrt(9))', {'sqrt': sqrt, 'pow': pow})
```

Run

Powered by DataCamp

Here, the code that is executed by exec() can also have sqrt() and pow() methods along with __builtins__ .

It's possible to change the name of the method according to your wish.

script.py IPython Shell

```python
from math import *
exec('print(dir())', {'squareRoot': sqrt, 'pow': pow})

# object can have squareRoot() module
exec('print(squareRoot(9))', {'squareRoot': sqrt, 'pow': pow})
```

Run

Powered by DataCamp

In the above program, squareRoot() calculates the square root (similar functionality like sqrt()). However, trying to use sqrt() will raise an exception.

Restricting the Use of built-ins

You can restrict the use of __builtins__ by giving value None to the '__builtins__' in the globals dictionary.

```python
exec(object, {'__builtins__': None})
```

3. Passing both globals and locals dictionary

You can make needed functions and variables available for use by passing `locals` dictionary. For example:

script.py IPython Shell

```python
from math import *

globalsParameter = {'__builtins__' : None}
localsParameter = {'print': print, 'dir': dir}
exec('print(dir())', globalsParameter, localsParameter)
```

Run

Powered by DataCamp

When you run the program, the output will be:

```
['dir', 'print']
```

Here, only two built-in methods print() and dir() can be executed by the exec() method.

It's important to note that, exec() executes the code and doesn't return any value (returns `None`). Hence, you cannot use return and yield statements outside of the function definitions.

Built-in Methods

Python abs()

Python any()

Python all()

Python ascii()

Python bin()

Python bool()

Python bytearray()

Python callable()

Python bytes()

Python chr()

Python compile()

Receive the latest tutorial to improve your programming skills.

Enter Email Address*

Join

TUTORIALS

Python Tutorials

C Tutorials

Java Tutorials

Kotlin Tutorials

C++ Tutorials

Swift Tutorials

R Tutorials

DSA

EXAMPLES

Python Examples

C Examples

Java Examples

Kotlin Examples

C++ Examples

R Examples

COMPANY

About

Advertising

Contact

LEGAL

Privacy Policy

Terms And Conditions

App's Privacy Policy

App's Terms And Conditions