

The compile () function in Python

The compile () function returns a code object in Python from the specified source. So how to use compile () function? Please find out in this article.

The compile () function returns a code object in Python from the specified source, be it a normal string, byte string or AST object. How is the syntax of *compile ()* function, what parameters does it have and how to use it? Invites you to read the track.



The compile () function syntax in Python

```
compile(source, filename, mode, flags=0, dont_inherit=False, optimize=-1)
```

The *compile ()* function uses when your Python code is in the form of strings or AST and you want to convert it to an object code.

The object code converted with *compile ()* will be invoked by the program by *exec ()* and *eval ()*.

The parameters of compile function ()

1. *source*: the source you want to convert, usually in the form of a string, byte object or AST object.
2. *filename*: file name contains the source. If the source does not belong to any file, you can name it here.
3. *mode*: includes the following values:
 1. *eval* - if the source is an expression.
 2. *exec* - if the source is a block of statements, classes, and functions.
 3. *single* - if the source is an interactive statement.

4. *flags*: control commands that affect source compilation, default to 0. This parameter is optional.
5. *dont_inherit*: control the commands that affect the source compilation, the default is False. This parameter is not required.
6. *optimize*: specify the optimization level of the compiler, the default is -1. This parameter is not required.

Return value from compile ()

The *compile ()* function returns a code object in Python.

Example: How does the compile () function work?

Follow the following example:

```
codeInString = 'a = 5\nb=6\nsum=a+b\nprint("sum =",sum)'\ncodeObject = compile(codeInString, 'sumstring', 'exec')\n\nexec(codeObject)
```

Run the program, the result is:

```
sum = 11
```

In the above example, the *source* here is a normal string, filename is *sumstring*, mode is *exec* after conversion, you can use the *exec ()* function to call the resulting code object.

You can see a list of built-in Python functions and don't forget to do Python exercises to reinforce your knowledge.

Previous lesson: *complex ()* function in Python

Next article: *Function setattr ()* in Python

You finished reading the article "**The compile () function in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.