

Modules & Packages

Textbook

Modules & Packages



Some common modules and packages people use in Python are random, Datetime, and Python Math. You can also create your own modules and packages that you could use!

This can be very useful. If there's a kind of function that you use often, instead of coding it every time, you can just put it in a module and use it in other programs across your project.

1. Create a Python File for your Module

Create a new Python file within the same project.

2. Add a Function

Within the new file, add the function that you want to use throughout your programs.

```
1 def greeting():
```

```
2 print("Good Morning!")
```

Note: You do not need to include the function call when creating the function in the module file.

3. Import the Module File

In the Python file that you want to work in, you can import the other file containing the function.

In this example, the file containing the module is named `mymodules`.

```
1 import mymodules
2
```

Now that you've imported that file, you can call the function within the file.

4. Call the Function

```
1 import mymodules
2 mymodules.greeting()
```

This will print out `Good Morning!`

Note: This is where the function call appears.

In these examples, the function named `greeting()` is the module you are using from the imported file.

Parameters

You can use parameters in your module function just as you would in any other function.

In the module file:

```
1 def greeting(name):
2     print("Good Morning " + name)
```

In your Python project:

```
1 import mymodules
2 mymodules.greeting("Shawn")
```

This will print out `Good Morning Shawn`.

Multiple Modules

You can use multiple modules (or functions) from the same file.

Let's say the module file has several functions:

```
1 def greeting():
2     print("Good Morning!")
```

```

3
4 def question():
5     print("How are you?")
6
7 def answer():
8     print("I'm fine, thank you!")
9

```

In your Python file, you can use as many or as few of these functions as you want!

```

1 import mymodules
2 mymodules.greeting()
3 mymodules.answer()
4

```

Module vs Package?

A [module](#) is a specific chunk of code being used. In these examples, the module is the function.

A [package](#) is a grouping of modules. In these examples, the entire file you are importing is the package.

Document Modules

Sometimes modules can contain many chunks of code. Documenting the code with comments is important.

```

1 def greeting():
2     print("Good Morning!")
3
4 #conversation print statements below
5
6 def question():
7     print("How are you?")
8
9 def answer():
10    print("I'm fine, thank you!")
11

```

Try it Out!

Now try to make your own module! Create a new python file and add a function. Import that file into your Python project and reference your function.

Questions (3)

1. True or False: Functions that are used as a module cannot contain parameters.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

2. Which of the following is the correct way to import a package? Assume the file is named practice.

MULTIPLE CHOICE

Choose the correct answer:

- A. import practice
- B. download practice
- C. use practice
- D. practice import

3. Where do you call a function that was declared in a module file?

MULTIPLE CHOICE

Choose the correct answer:

- A. In the module file
- B. In the Python project file.
- C. No function call is needed.
- D. Both in the module and Python project file.

Answer Keys & Solutions

Questions

1. True or False: Functions that are used as a module cannot contain parameters.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Functions used as a module can also have parameters just like any function.

2. Which of the following is the correct way to import a package? Assume the file is named practice.

MULTIPLE CHOICE

Correct Answer:

A. import practice

✓ Correct

B. download practice

✗ Incorrect

C. use practice

✗ Incorrect

D. practice import

✗ Incorrect

Explanation:

This is the same way to import the Random library.

3. Where do you call a function that was declared in a module file?

MULTIPLE CHOICE

Correct Answer:

A. In the module file

✗ Incorrect

B. In the Python project file.

✓ Correct

C. No function call is needed.

✗ Incorrect

D. Both in the module and Python project file.

✗ Incorrect

Explanation:

You call the function when you choose which function to use in your project.