

Lab Assignment & Solution



Cybersecurity Professional Program
Introduction to Python
for Security

Functions

PY-05-LS2

Returning Lists

Copyright © 1996–2021 HackerU Ltd.
All Rights Reserved.

Note: Solutions for the instructor are shown inside the green box.

Lab Objective

Understand how functions are able to return different types of lists and identify each type.

Lab Mission

Create multiple functions that return different list types and check their types.

Lab Duration

15–25 minutes

Requirements

- Basic knowledge of Python

Resources

- Environment & Tools
 - Windows, Linux, macOS
 - PyCharm
 - Python 3

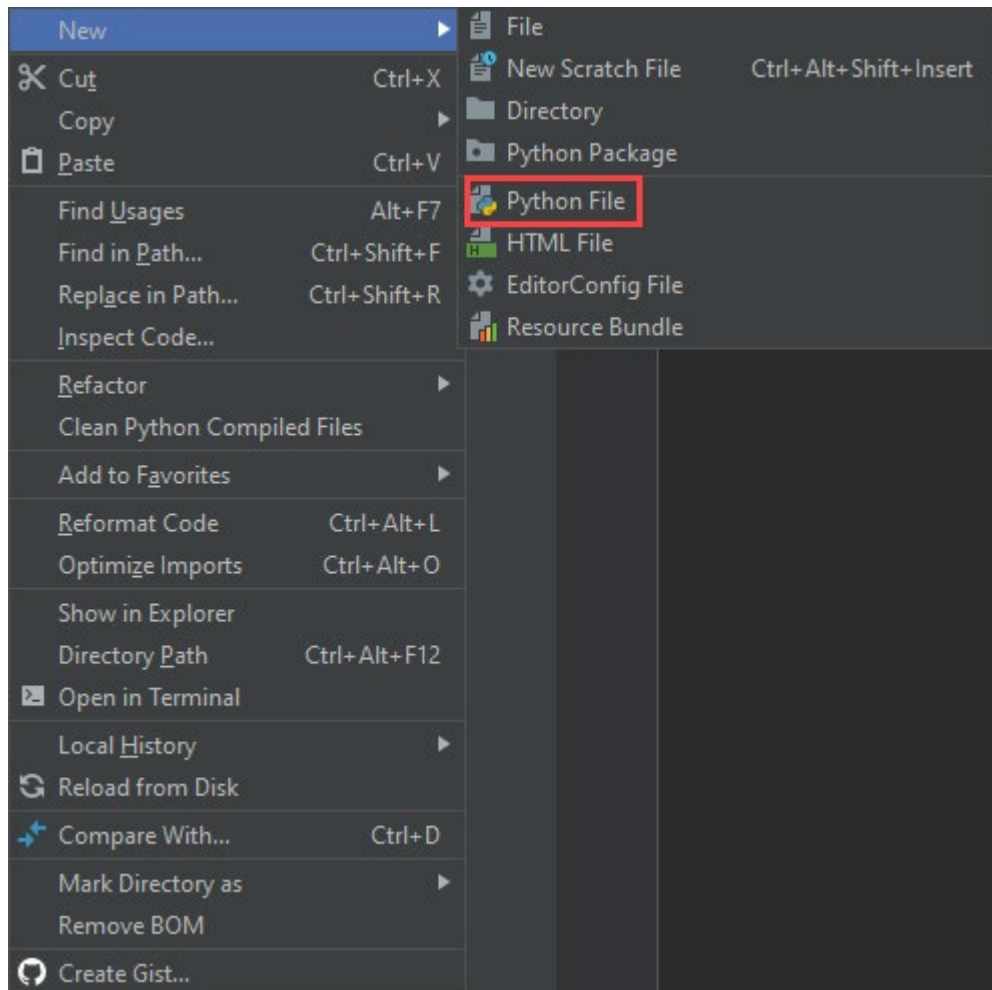
Textbook References

- Chapter 5: Functions
 - Section 1: Introduction to Functions

Lab Task: Returning Lists with Python

In this task, you will create several functions that return a type of value and identify it.

- 1 Create a new Python file in PyCharm by right clicking the project you created and selecting **New > Python File**.



- 2 Declare a function that accepts two parameters.

```
def returned_list(first, second):
```

- 3 Make the function return a list of the parameters.

```
def returned_list(first, second):  
    return [first, second]
```

- 4 Declare a second function that accepts two parameters.

```
def returned_dict(first, second):
```

- 5 Make the function return a dictionary of the parameters.

```
def returned_dict(first, second):  
    return {first: second}
```

- 6 Declare a third function that accepts two parameters.

```
def returned_tup(first, second):
```

- 7 Make the function return a tuple of the parameters.

```
def returned_tup(first, second):  
    return first, second
```

- 8 Declare a fourth function that accepts two parameters.

```
def returned_none(first, second):
```

- 9 Make the function perform an addition operation on the parameters.

```
def returned_none(first, second):  
    res = first + second
```

10 Create the main function to handle the program.

```
def main():
```

11 Create two variables that request inputs of values from a user.

```
def main():  
    first_val = input("Please enter a value: ")  
    second_val = input("Please enter another value: ")
```

12 Save the first function's result to a variable.

```
def main():  
    first_val = input("Please enter a value: ")  
    second_val = input("Please enter another value: ")  
    list_res = returned_list(first_val, second_val)
```

13 Save the second function's result to a variable.

```
def main():  
    first_val = input("Please enter a value: ")  
    second_val = input("Please enter another value: ")  
    list_res = returned_list(first_val, second_val)  
    dict_res = returned_dict(first_val, second_val)
```

14 Save the third function's result to a variable.

```
def main():  
    first_val = input("Please enter a value: ")  
    second_val = input("Please enter another value: ")  
    list_res = returned_list(first_val, second_val)  
    dict_res = returned_dict(first_val, second_val)  
    tup_res = returned_tup(first_val, second_val)
```

15 Save the fourth function's result to a variable.

```
def main():
    first_val = input("Please enter a value: ")
    second_val = input("Please enter another value: ")
    list_res = returned_list(first_val, second_val)
    dict_res = returned_dict(first_val, second_val)
    tup_res = returned_tup(first_val, second_val)
    none_res = returned_none(first_val, second_val)
```

16 Print the results and the types saved in the variables.

```
def main():
    first_val = input("Please enter a value: ")
    second_val = input("Please enter another value: ")
    list_res = returned_list(first_val, second_val)
    dict_res = returned_dict(first_val, second_val)
    tup_res = returned_tup(first_val, second_val)
    none_res = returned_none(first_val, second_val)
    print("{} and its type is: {}".format(list_res,
type(list_res)))
    print("{} and its type is: {}".format(dict_res,
type(dict_res)))
    print("{} and its type is: {}".format(tup_res,
type(tup_res)))
    print("{} and its type is: {}".format(none_res,
type(none_res)))
```

17 Invoke the main function to run the program.

```
main()
```