# MODULE 1.2

Python Libraries

TECHNION
Azrieli School of Continuing Studies of the Technion

# Program

- Introduction
- Importing libraries
- Python Standard Library
- Researching Libraries (Group Lab)

# Learning Objectives

- You will be able **to import** modules and libraries into Python
- You will be able **to research** libraries and their contents with tools provided in Python

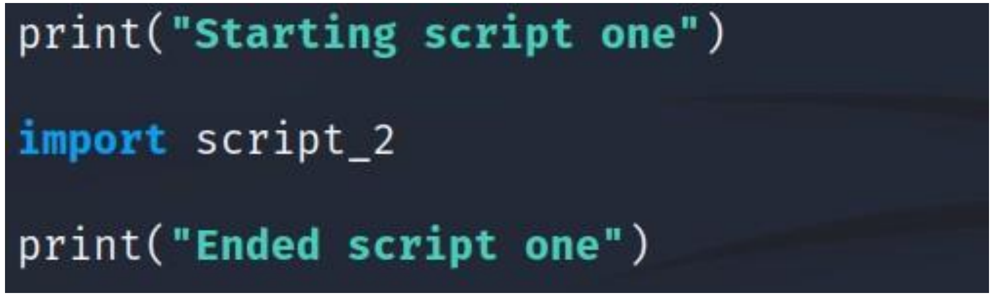
# Importing Libraries

# Python Scripts

- Python files are text files, filled with Python code.
- Let’s say we want to use (execute) the code that is in a different Python script.
- We can do this by using the import statement. When using import, the whole file is executed, line-by-line.
  - This means that the current script is paused until all lines of the other script have finished executing.
  - Only then, the current script continues to run.
- You can only import scripts that are in the same directory (folder) as your script.

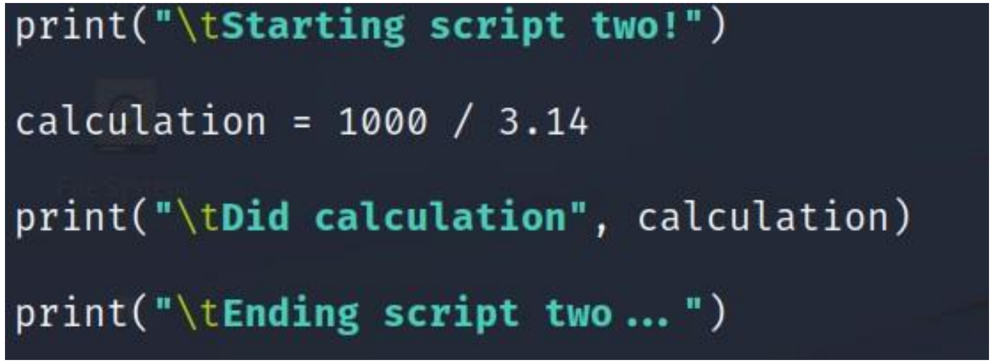
# 10 minutes exercise…

- kali@kali:~$ nano script_1.py



```
print("Starting script one")

import script_2

print("Ended script one")
```

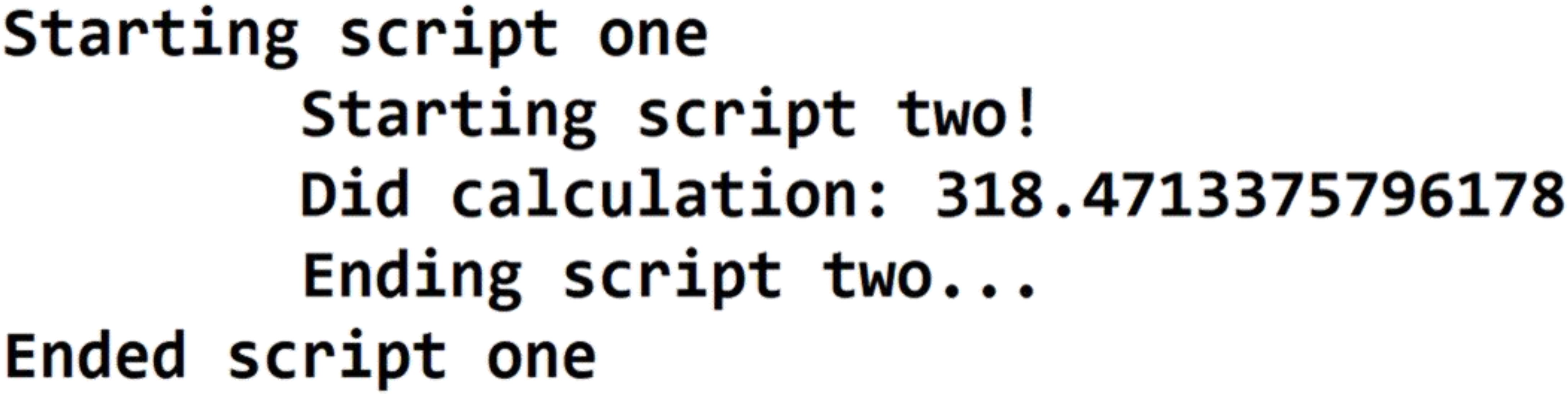
- kali@kali:~$ nano script_2.py



```
print("\tStarting script two!")

calculation = 1000 / 3.14

print("\tDid calculation", calculation)

print("\tEnding script two ... ")
```

- kali@kali:~$ python3 script_1.py

# Output



```
Starting script one
        Starting script two!
        Did calculation: 318.4713375796178
        Ending script two...
Ended script one
```

# Using Imported Functions

- After importing, we can now use variables and functions (or properties and methods) defined in our imported script!
- We do this just the same as we use methods on an object – by using a dot.
- Examples:
  - math.cos()
  - re.compile()
  - sys.argv

# 10 min exercise….

- Add a simple user defined function to script_2.py from the previous exercise
- Call this function from script one
  - script_2.your_function()

# The as And from Keywords

- We can use the as keyword to give the imported script a nickname:
  - import script_2 as s2
  - s2.your_function()

- We can use the from keyword to import a specific object from our script.
  - from script_2 import your_function
  - from script import function_1, function_2

- Combining as and from
  - from script import function_1 as f1, function_2 as f2

*Note: once imported, you cannot import a script again.*

# from module_name import *

- Using* wildcard instead of specific function names imports all the functions in the other script.
- *Discussion:* Why is this not recommended to do?

- In the remainder of the course, please do import the full libraries when doing demo’s / presentations without renaming. *Why?*

# Python Standard Library

Azrieli School of Continuing Studies of the Technion

TECHNION
Azrieli School of Continuing Studies of the Technion

# The Python Standard Library

- A collection of modules that come bundled with the Python language.
- These modules provide a wide range of functionalities, including file I/O, network programming, regular expressions, data handling, and many more.
  1. **os -** provides a way to interact with the operating system
  2. **shutil -** provides a higher-level interface for various file operations that are not available with the built-in os module
  3. **sys -** provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter
  4. **re -** provides regular expression matching operations
  5. **time -** time - provides time-related functions
  6. **datetime -** provides classes for working with dates and times
  7. **socket -** provides low-level network programming functionality

# Researching Libraries

# Researching Libraries

- https://docs.python.org/3/library/index.html
- https://chat.openai.com/ (directed queries!)
- The REPL / interactive interpreter:
  - dir() and vars() function in combination with pprint.pprint()
  - the inspect module (getmembers())
  - the type() command
  - the built-in help() function

# Presentations

- Research the library assigned to your group:
    1. os - provides a way to interact with the operating system
    2. shutil - provides a higher-level interface for various file operations that are not available with the built-in os module
    3. sys - provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter
    4. re - provides regular expression matching operations
    5. time - time - provides time-related functions
    6. datetime - provides classes for working with dates and times

- Make a short presentation about the libraries methods and properties (max 5 slides – 10 minutes)
- Write a very simple script, demonstrating the possibilities of this library

# Learning Objectives

- You will be able **to import** modules and libraries into Python
- You will be able **to research** libraries and their contents with tools provided in Python