PYTHON MODULES & PACKAGES

By:
Zappcode Academy

WHAT IS MODULE

❑ In Python, a module is a single file containing Python definitions and statements. These definitions and statements can include variables, functions, and classes and can be used to organize related functionality into a single, reusable package. Module organizes and reuses code in Python by grouping related code into a single file.

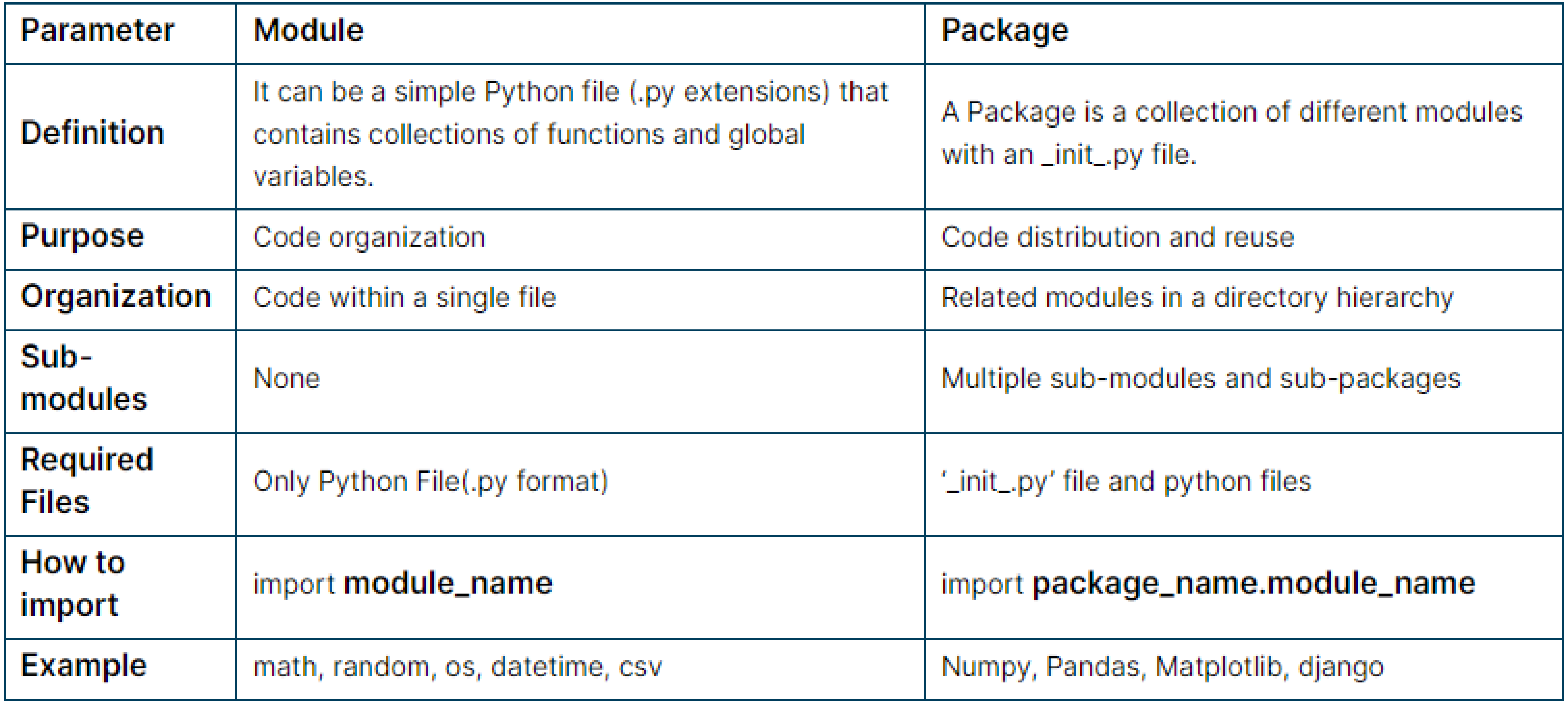
❑ Modules can be imported and used in other Python files using the **import** statement.

❑ Some popular modules in Python are math, random, csv, and datetime.

MODULE VS PACKAGE

A module is a single file containing python code, whereas a package is a collection of modules that are organized in a directory hierarchy.

CONTINUED….



Parameter	Module	Package
Definition	It can be a simple Python file (.py extensions) that contains collections of functions and global variables.	A Package is a collection of different modules with an _init_.py file.
Purpose	Code organization	Code distribution and reuse
Organization	Code within a single file	Related modules in a directory hierarchy
Sub-modules	None	Multiple sub-modules and sub-packages
Required Files	Only Python File(.py format)	‘_init_.py’ file and python files
How to import	import **module_name**	import **package_name.module_name**
Example	math, random, os, datetime, csv	Numpy, Pandas, Matplotlib, django

LISTING PYTHON MODULE DIRECTORIES

❑You can use the ***sys.path*** attribute to get a list of the directories that Python searches when looking for modules. The list of strings stored in the ***sys.path*** variable determines the locations where Python will look for modules when they are imported.

❑For example:

```
import sys
print(sys.path)
```

RENAMING PYTHON MODULES

- To rename a Python module, you can use the ***import*** statement with the ***as*** a keyword. It would have a syntax as shown below:

Syntax:

import Module_name as Alias_name

PYTHON BUILT IN MODULES

- **sys**: The built-in sys module in Python allows interaction with certain aspects of the interpreter, such as variables and functions, that have a strong relationship with it.

- **os**: Provides functions for interacting with the operating system, such as reading and writing files, starting processes, etc.

- **math**: Provides mathematical functions and constants, such as trigonometric functions, logarithms, and pi.

- **time**: Provides functions for working with time, such as getting the current time and sleeping for a certain number of seconds.

CONTINUED...

- **random**: Provides functions for generating random numbers and selecting random elements from a sequence.
- **string**: Provides functions for working with strings, such as formatting strings, parsing strings, and creating common string operations.
- **re**: Provides functions for working with regular expressions.
- **datetime**: Provides classes for working with dates and times, such as calculating the difference between two dates or formatting a date as a string.

PRATICALS

Question: Write a Python function that generates a random password of a specified length. The password should include a mix of uppercase letters, lowercase letters, digits, and special characters. Use the **random** module to achieve this.

```
#generate random password
import random
import string

def generate_random_password(length):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

#calling function:
password = generate_random_password(12)
print("Random Password:", password)
```

Note: string.ascii_letters (for letters), string.digits (for digits), and string.punctuation (for special characters

```
#using os module
import os
os.mkdir("myfolder")
with open("myfolder/myfile.txt","w+") as f:
    f.write("Hello World")
    print(f.read())
    f.close()
old_name=r"myfolder/myfile.txt"
new_name=r"myfolder/myfiles.text"
os.rename(old_name,new_name)
```

Question: Write a Python program that calculates the area of a circle. Use the **math** module to access the value of π and perform the calculation.

```
#using math module
import math
def calculate_circle_area(radius):
    area = math.pi * (radius ** 2)
    return area

# Get user input for the radius of the circle
radius = float(input("Enter the radius of the circle: "))

# Calculate and display the area of the circle
circle_area = calculate_circle_area(radius)
print(f"The area of a circle with radius {radius} is: {circle_area:.2f}")
```

```
#another example using math module
import math

def calculate_factorial(number):
    factorial = math.factorial(number)
    return factorial

# Input the number for which factorial needs to be calculated
num = int(input("Enter a number to calculate its factorial: "))

# Calculate and display the factorial of the number
result = calculate_factorial(num)
print(f"The factorial of {num} is {result}")
```

```
#using time module
import time
current_time = time.localtime()  # Get the current time in struct_time format
formatted_time = time.strftime("%H:%M:%S", current_time)  # Format the time as HH:MM:SS
print(f"The current time is: {formatted_time}")
```

```
#using datetime module to get current time in New Work and India
from datetime import datetime
from pytz import timezone
# Define time zones for New York and India
new_york_timezone = timezone('America/New_York')
india_timezone = timezone('Asia/Kolkata')
# Get the current time in New York and India
ny_time = datetime.now(new_york_timezone)
india_time = datetime.now(india_timezone)
# Format the time in a human-readable way
ny_time_str = ny_time.strftime('%Y-%m-%d %H:%M:%S %Z%z')
india_time_str = india_time.strftime('%Y-%m-%d %H:%M:%S %Z%z')
# Display the times
print(f'Current time in New York: {ny_time_str}')
print(f'Current time in India: {india_time_str}')
```

```
#using random module
import random

def roll_dice():
    return random.randint(1, 6)  # Simulating a standard 6-sided dice

# Roll the dice and display the result
print("Let's roll the dice!")
result = roll_dice()
print("The dice shows:", result)
```

```
#another example of using random
import random
items = ['Apple', 'Orange', 'Banana', 'Grape', 'Cherry']
random_item = random.choice(items)
print(f"Randomly selected item: {random_item}")
```

```
#using string module
import string
def has_alphabetic_characters(input_string):
    for char in input_string:
        if char in string.ascii_letters:
            return True
    return False
# Check if the input string has alphabetic characters
user_input = input("Enter a string to check for alphabetic characters: ")
if has_alphabetic_characters(user_input):
    print("Contains alphabetic characters.")
else:
    print("Does Not contain any alphabetic characters.")
```

```
#using sys module to exit the program
import sys
def main():
    answer = input("Do you want to exit the program? (yes/no): ")
    if answer.lower() == 'yes':
        print("Exiting the program...")
        sys.exit(0)
    else:
        print("Continuing the program.")
main()
```

```
#using re module
import re

# Text containing a specific word to replace
text = "The color of the sky is blue. The color of the sea is also blue."

# Define a word to be replaced and its replacement
a = 'blue'    #to replace
b = 'green'   #replace with

# Using re.sub to replace the word 'blue' with 'green'
new_text = re.sub(a, b, text)
print("Original Text:", text)
print("Modified Text:", new_text)
```

TO GET SQUARE ROOT OF A USER INPUT NUMBER

```
import math

# Taking user input for a number
number = float(input("Enter a number: "))

# Calculating the square root using math.sqrt() function
square_root = math.sqrt(number)

print(f"The square root of {number} is {square_root}")
```

GENERATING A RANDOM INTEGER NUMBER

```
import random

# Generating a random number between 1 and 10 using random.randint()
random_number = random.randint(1, 10)

print(f"Random number: {random_number}")
```

GENERATING A RANDOM FLOAT NUMBER

```
import random

random_float = random.uniform(1, 100)
print(random_float)
```

RANDOM PASSWORD

```
import random

def generate_password(length):
    characters = ['1', '6', ‘4', '9']
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

random_password = generate_password(3)
print("Randomly generated password:", random_password)
```

ROUNDING A NUMBER

```
import math

value = 3.78
rounded_value = math.floor(value)  # or math.ceil(value) for ceiling rounding
print(f"The rounded value of {value} is {rounded_value}")
```