# Modules

## 🎯 Session Objectives:

- ✅ Understand what a Python module is
- ✅ Create and use your own module
- ✅ Import specific functions or all functions from a module
- ✅ Rename a module or function using an alias
- ✅ Understand and use built-in modules

## 1 Module

A module is just a single Python file (.py) that contains code – functions, classes, or variables – that you can import into another Python file.

Purpose: Organize code into reusable pieces.

## 2 Package

A package is a folder containing multiple Python modules and a special `__init__.py` file (even if empty).

It helps organize modules into a hierarchical structure.

## 3 Library

A library is a collection of related modules and packages that provide a set of functionalities.

Example: NumPy, Pandas

## 4 Framework

A framework is a bigger structure that provides not only modules and packages but also a set of rules and architecture for building applications.

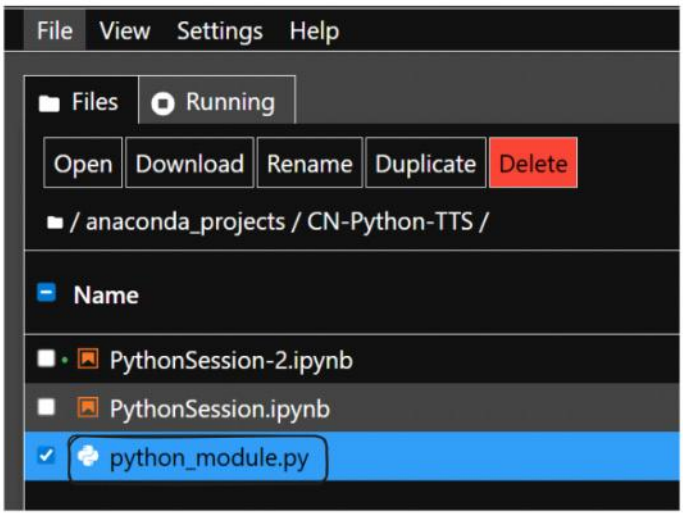
Example: Django, Flask (web frameworks).

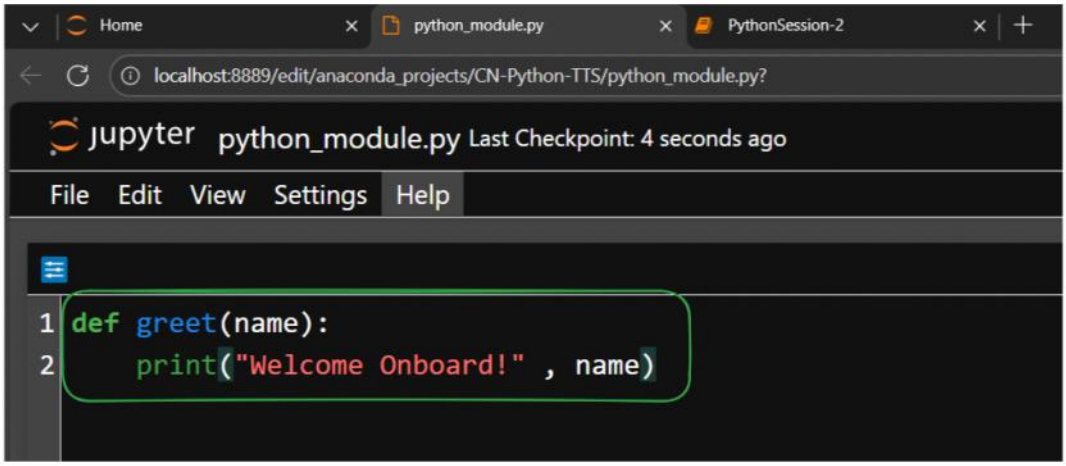
## What is a Module?

A module is just a .py file that contains reusable Python code—like functions, variables, or classes.

### Why Use Modules?

- Avoid writing the same code again and again.
- Break large programs into smaller, manageable pieces.
- Share functionality across different files
- Make code cleaner, organized, and reusable



# What is a Module?

**A module is just a .py file that contains reusable Python code—like functions, variables, or classes.**

## Why use Modules?

- Avoid writing the same code again and again.
- Break large programs into smaller, manageable pieces.
- Share functionality across different files
- Make code cleaner, organized, and reusable

## Analogy :

- Module - A single Chapter in a book
- Package - A book made of multiple Chapters
- Library - A bookshelf containing multiple books on a subject.
- Framework - A whole Library Building with a set of rules for how books are organized and used.

```python
import python_module
python_module.greet("Aditya Verma!")
```

```
Welcome Onboard! Aditya Verma!
```

```python
python_module.greet("Nihal Mehra")
```

```
Welcome Onboard! Nihal Mehra
```

```python
python_module.greet("Arpit Gupta")
```

```
Welcome Onboard! Arpit Gupta
```

User Defined

Home | math_utils.py

localhost:8889/edit/anaconda_projects/CN-Python-TTS/math_utils.

jupyter math_utils.py Last Checkpoint: 8 seconds ag

File Edit View Settings Help

```python
def add(x,y):
    return x + y

def sub(x,y):
    return x - y

def multiply(x,y):
    return x * y

def divide(x,y):
    return x / y

def square(val):
    return val ** 2

def cube(val):
    return val ** 3
```

math_utils.py

python_module.py

```python
def add(x,y):
    return x + y

def sub(x,y):
    return x - y

def multiply(x,y):
    return x * y

def divide(x,y):
    return x / y

def square(val):
    return val ** 2

def cube(val):
    return val ** 3

def fact(n):
    if n < 0:
        raise ValueError("n in factorical can't be negative")
    elif n == 1:
        return 1
    else:
        return n * fact(n-1) # Recursion.
```

```python
import math_utils # Importing .py file
result = math_utils.add(10,11)
print(result)
```

```
21
```

```python
result = math_utils.sub(21,11)
print(result)
```

```
10
```

```python
result = math_utils.multiply(10,11)
print(result)
```

```
110
```

```python
result = math_utils.divide(11,10)
print(result)
```

```
1.1
```

```python
result = math_utils.cube(11)
print(result)
```

```
1331
```

```python
result = math_utils.square(11)
print(result)
```

```
121
```

```python
result = math_utils.fact(-11)
print(result)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[10], line 1
----> 1 result = math_utils.fact(-11)
      2 print(result)

File ~\anaconda_projects\CN-Python-TTS\math_utils.py:21, in fact(n)
     19 def fact(n):
     20     if n < 0:
---> 21         raise ValueError("n in factorical can't be negative")
     22     elif n == 1:
     23         return 1

ValueError: n in factorical can't be negative
```

userdefined

Fix Code

```python
result = math_utils.fact(11)
print(result)
```

```
39916800
```

Home | string_utils.py | PythonSession-2

localhost:8889/edit/anaconda_projects/CN-Python-TTS/string_utils.py?

jupyter string_utils.py Last Checkpoint: 5 seconds ago

File Edit View Settings Help

```python
def reverse_string(_str):
    return _str[::-1]

def counts_vowels(_str):
    vowels = "aeiouAEIOU"
    count = 0
    for char in _str:
        if char in vowels:
            count+=1
    return count
```

```python
import string_utils
rev_str = string_utils.reverse_string("racecar")
rev_str
```

```
'racecar'
```

```python
rev_str = string_utils.reverse_string("ninja")
rev_str
```

```
'ajnin'
```

```python
rev_str = string_utils.reverse_string("hello")
rev_str
```

```
'olleh'
```

```python
count_vowels = string_utils.counts_vowels("Coding Ninjas")
count_vowels
```

```
4
```

```python
count_vowels = string_utils.counts_vowels("Python Programmings")
count_vowels
```

```
4
```

```python
count_vowels = string_utils.counts_vowels("Today is a very Awesome day!")
count_vowels
```

```
10
```

```python
from string_utils import reverse_string , counts_vowels
rev_str = reverse_string("Python")
rev_str
```

```
'nohtyP'
```

```python
vowels_count = counts_vowels("Aditya Verma")
vowels_count
```

```
5
```

```python
from math_utils import *
_sum = add(20,51)
print(_sum)
```

```
71
```

```python
_mul = multiply(10,5)
print(_mul)
```

```
50
```

```python
# Aliases 'AS'
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```python
# Aliases 'as'
import math_utils as math
math.add(10,11)
```

```
21
```

```python
math.square(10)
```

```
100
```

```python
math.cube(9)
```

```
729
```

## What Are Built-in Modules?

Python includes standard modules that come pre-installed.
You don't have to download them—you just import and use them.

They help with:

- Getting system or platform information
- Performing mathematical operations
- Handling dates and times
- Generating random values
- Working with files and directories

```python
import platform as p
dir(platform)
```

```
['_Processor',
 '_WIN32_CLIENT_RELEASES',
 '_WIN32_SERVER_RELEASES',
 '__builtins__',
 '__cached__',
 '__copyright__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '__version__',
 '_comparable_version',
 '_default_architecture',
 '_follow_symlinks',
 '_get_machine_win32',
```

```python
p.system()
```

```
'Windows'
```

```python
p.version()
```

```
'10.0.26100'
```

```python
p.release()
```

```
'11'
```

```python
p.machine()
```

```
'AMD64'
```

```python
p.architecture()
```

```
('64bit', 'WindowsPE')
```

```python
p.platform()
```

```
'Windows-11-10.0.26100-SP0'
```

```python
p.processor()
```

```
'Intel64 Family 6 Model 186 Stepping 3, GenuineIntel'
```

```python
p.uname()
```

```
uname_result(system='Windows', node='Priya', release='11', version='10.0.26100', machine='AMD64')
```

```python
p.python_version()
```

```
'3.12.7'
```

```python
p.python_compiler()
```

```
'MSC v.1929 64 bit (AMD64)'
```

```python
p.python_implementation()
```

```
'CPython'
```

```python
p.mac_ver()
```

```
('', ('', '', ''), '')
```

```python
import math
dir(math) # provide an info all the existing functions
```

```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
```

```python
math.ceil(199.79) # Rounding Up
```

```
200
```

```python
math.floor(199.65) # Rounding Down
```

```
199
```

```python
math.fabs(-199.99) # Return float with abs
```

```
199.99
```

```python
math.factorial(5)
```

```
120
```

```python
math.sqrt(81) # 9
```

```
9.0
```

```
math.sqrt(121) # 11
```

```
11.0
```

```
math.gcd(15,36)
```

```
3
```

```
math.lcm(15,25)
```

```
75
```

```
math.pi
```

```
3.141592653589793
```

```
# Area of a Circle = pi * r^2
pi = math.pi
radius = int(input("Enter the value of radius: "))
area_of_circle = math.pi * math.pow(radius , 2)
print(area_of_circle)
```

```
Enter the value of radius:  11
380.132711084365
```

```
# pi ~ 180 degree
math.degrees(math.pi)
```

```
180.0
```

```
math.radians(180)
```

```
3.141592653589793
```

```
math.sin(math.pi/2)
```

```
1.0
```

```
math.tan(math.pi/4)
```

```
0.9999999999999999
```

```
math.ceil(math.tan(math.pi/4))
```

```
1
```

```
math.log(100,10)
```

```
2.0
```

```
math.log10(100000) # 10^5
```

```
5.0
```

```
import datetime
dir(datetime)
```

```
['MAXYEAR',
 'MINYEAR',
 'UTC',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'date',
 'datetime',
```

```
datetime.MAXYEAR
```

```
9999
```

```
datetime.MINYEAR
```

```
1
```

```
now = datetime.datetime.now()
now
```

```
datetime.datetime(2025, 9, 23, 23, 1, 7, 29845)
```

```
print(now)
```

```
2025-09-23 23:01:07.029845
```

```
today = datetime.datetime.today()
today
```

```
datetime.datetime(2025, 9, 23, 23, 1, 35, 186374)
```

```
print(today)
```

```
2025-09-23 23:01:35.186374
```