**Python Built-in Modules**

**How to Use Built-in Modules**

To use a built-in module, you typically use the **import** statement. Once imported, you can call the functions or use the classes provided by the module.

import module\_name # Import the module

# Now you can use the functions or objects provided by the module

module\_name.function\_name()

Or, if you want to use a specific function or object directly from the module, you can use:

from module\_name import function\_name

**Popular Built-in Modules in Python**

1. **math** - Provides mathematical functions.
   * **Functions**: sin(), cos(), sqrt(), pow(), factorial(), and constants like pi, e.

**Example**:

import math

result = math.sqrt(16)

print(result) # Output: 4.0

1. **datetime** - Used for working with dates and times.
   * **Classes**: datetime, date, time, timedelta.

**Example**:

import datetime

now = datetime.datetime.now()

print(now) # Output: Current date and time

1. **random** - Generates random numbers and random selections.
   * **Functions**: randint(), choice(), shuffle(), random(), seed().

**Example**:

import random

random\_number = random.randint(1, 10)

print(random\_number) # Output: Random number between 1 and 10

1. **os** - Provides functions for interacting with the operating system, like working with files and directories.
   * **Functions**: os.path, os.mkdir(), os.remove(), os.rename(), os.getcwd(), os.environ.

**Example**:

import os

current\_directory = os.getcwd()

print(current\_directory) # Output: Current working directory

1. **sys** - Provides access to system-specific parameters and functions, like command-line arguments and Python runtime environment.
   * **Functions/Attributes**: sys.argv, sys.exit(), sys.version, sys.path.

**Example**:

import sys

print(sys.version) # Output: Python version being used

1. **re** - Provides functions for working with regular expressions.
   * **Functions**: re.match(), re.search(), re.findall(), re.sub().

**Example**:

import re

pattern = r'\d+'

text = "There are 12 apples"

result = re.findall(pattern, text)

print(result) # Output: ['12']

1. **json** - Used to work with JSON (JavaScript Object Notation), which is a popular data format.
   * **Functions**: json.dump(), json.dumps(), json.load(), json.loads().

**Example**:

import json

data = {"name": "Alice", "age": 30}

json\_str = json.dumps(data)

print(json\_str) # Output: {"name": "Alice", "age": 30}

1. **collections** - Implements specialized container datatypes like namedtuple(), deque(), Counter(), and defaultdict().
   * **Classes**: Counter, deque, namedtuple, defaultdict.

**Example**:

from collections import Counter

text = "hello world"

counter = Counter(text)

print(counter) # Output: Counter({'l': 3, 'o': 2, 'h': 1, 'e': 1, 'w': 1, 'r': 1, 'd': 1})

1. **itertools** - Provides a collection of tools for handling iterators (for creating iterators for efficient looping).
   * **Functions**: count(), cycle(), repeat(), chain(), combinations().

**Example**:

import itertools

numbers = [1, 2, 3]

result = itertools.permutations(numbers)

for perm in result:

print(perm)

1. **time** - Provides time-related functions, useful for measuring time or delaying execution.
   * **Functions**: time.sleep(), time.time(), time.localtime().

**Example**:

import time

start = time.time()

time.sleep(2) # Sleep for 2 seconds

end = time.time()

print("Elapsed time:", end - start) # Output: Elapsed time: 2.0 (approximately)

1. **socket** - Provides access to low-level networking interfaces and allows you to communicate over a network.
   * **Functions**: socket.connect(), socket.bind(), socket.send(), socket.recv().

**Example**:

import socket

s = socket.socket()

s.connect(('localhost', 8080)) # Example connection to a server

1. **csv** - Handles CSV (Comma Separated Values) files.
   * **Functions**: csv.reader(), csv.writer().

**Example**:

import csv

with open('data.csv', mode='w', newline='') as file:

writer = csv.writer(file)

writer.writerow(['Name', 'Age', 'Country'])

writer.writerow(['Alice', 25, 'USA'])

writer.writerow(['Bob', 30, 'Canada'])

1. **urllib** - A module for working with URLs (making HTTP requests, parsing URLs, etc.).
   * **Functions**: urllib.request, urllib.parse.

**Example**:

import urllib.request

response = urllib.request.urlopen('https://www.example.com')

html = response.read()

print(html)

1. **pdb** - Provides a debugger for interactive debugging.
   * **Functions**: pdb.set\_trace() to set a breakpoint.

**Example**:

import pdb

def my\_function():

x = 10

y = 20

pdb.set\_trace() # Set breakpoint

result = x + y

print(result)

my\_function()

**How to Find More Built-in Modules**

If you're ever unsure about which modules are available in the Python Standard Library, you can check the [official Pythondocumentation](https://docs.python.org/3/library/). It provides a comprehensive list of all built-in modules, along with examples and explanations for how to use them.

You can also use the **help() function** in Python to get information about a particular module or function.

**Example:**

import math

help(math) # Displays help for the math module