Introduction to Data Structures

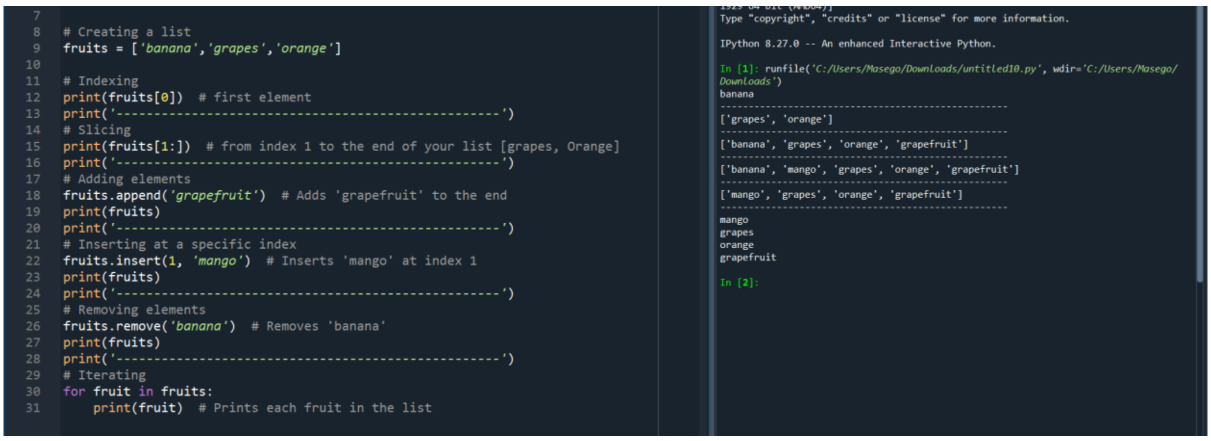
- **Definition**: A data structure is a way of organizing, managing, and storing data for efficient access and modification.

Key Built-in Python Data Structures:

- Lists
- Tuples
- Sets
- Dictionaries

- **List:**

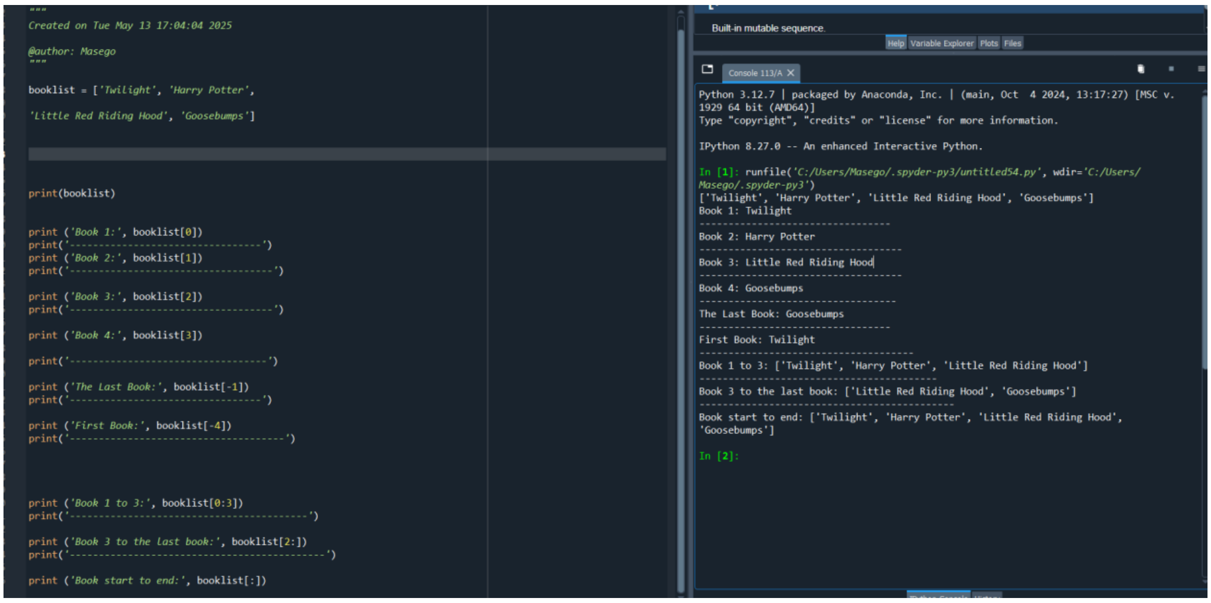
This container can hold data of any type, with a dynamic number of objects. Lists use **[]** (square brackets) to define their elements.



```python

# Creating a list
fruits = ['banana','grapes','orange']

# Indexing
print(fruits[0])  # first element
print('--------------------------------------------------')
# Slicing
print(fruits[1:])  # from index 1 to the end of your list [grapes, Orange]
print('--------------------------------------------------')
# Adding elements
fruits.append('grapefruit')  # Adds 'grapefruit' to the end
print(fruits)
print('--------------------------------------------------')
# Inserting at a specific index
fruits.insert(1, 'mango')  # Inserts 'mango' at index 1
print(fruits)
print('--------------------------------------------------')
# Removing elements
fruits.remove('banana')  # Removes 'banana'
print(fruits)
print('--------------------------------------------------')
# Iterating
for fruit in fruits:
    print(fruit)  # Prints each fruit in the list
```

```
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/Downloads/untitled10.py', wdir='C:/Users/Masego/
Downloads')
banana
--------------------------------------------------
['grapes', 'orange']
--------------------------------------------------
['banana', 'grapes', 'orange', 'grapefruit']
--------------------------------------------------
['banana', 'mango', 'grapes', 'orange', 'grapefruit']
--------------------------------------------------
['mango', 'grapes', 'orange', 'grapefruit']
--------------------------------------------------
mango
grapes
orange
grapefruit

In [2]:
```

Another Example



```python
"""
Created on Tue May 13 17:04:04 2025

@author: Masego
"""

booklist = ['Twilight', 'Harry Potter',

'Little Red Riding Hood', 'Goosebumps']



print(booklist)


print ('Book 1:', booklist[0])
print('--------------------------------')
print ('Book 2:', booklist[1])
print('----------------------------------')

print ('Book 3:', booklist[2])
print('----------------------------------')

print ('Book 4:', booklist[3])

print('---------------------------------')

print ('The Last Book:', booklist[-1])
print('--------------------------------')

print ('First Book:', booklist[-4])
print('------------------------------------')




print ('Book 1 to 3:', booklist[0:3])
print('----------------------------------------')

print ('Book 3 to the last book:', booklist[2:])
print('-------------------------------------------')

print ('Book start to end:', booklist[:])
```

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/.spyder-py3/untitled54.py', wdir='C:/Users/
Masego/.spyder-py3')
['Twilight', 'Harry Potter', 'Little Red Riding Hood', 'Goosebumps']
Book 1: Twilight
--------------------------------
Book 2: Harry Potter
----------------------------------
Book 3: Little Red Riding Hood
----------------------------------
Book 4: Goosebumps
---------------------------------
The Last Book: Goosebumps
--------------------------------
First Book: Twilight
------------------------------------
Book 1 to 3: ['Twilight', 'Harry Potter', 'Little Red Riding Hood']
----------------------------------------
Book 3 to the last book: ['Little Red Riding Hood', 'Goosebumps']
-------------------------------------------
Book start to end: ['Twilight', 'Harry Potter', 'Little Red Riding Hood',
'Goosebumps']

In [2]:
```

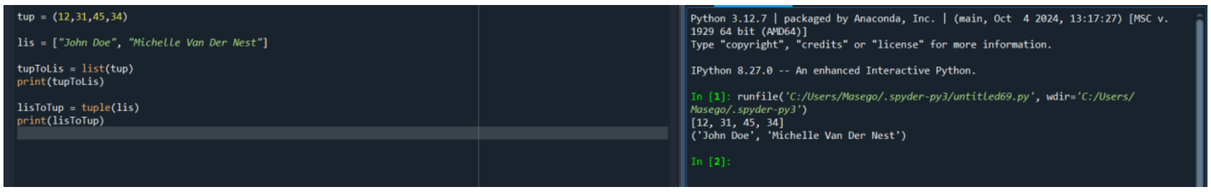
Class activity

Write a program that will prompt the user to enter 4 assessment marks and store the marks in a list.

Tuple:

This container can hold data of any type, but only a fixed number of objects. Tuples use **()** (Normal brackets) to index their elements.

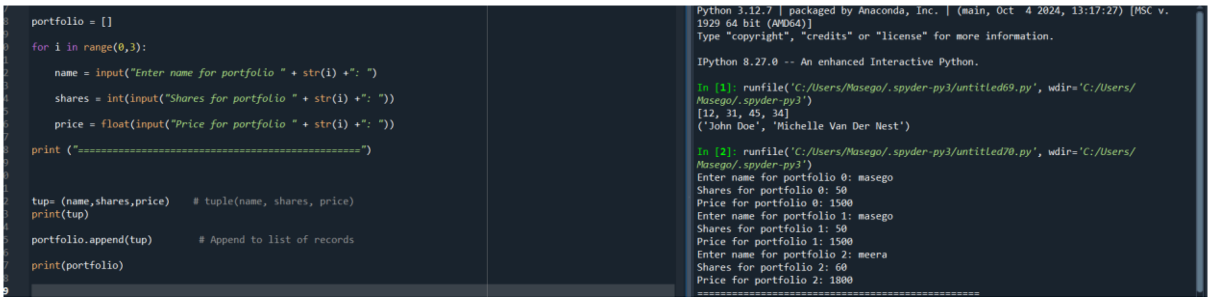
Tuples are Ordered, immutable, allows duplicates.



```python
tup = (12,31,45,34)

lis = ["John Doe", "Michelle Van Der Nest"]

tupToLis = list(tup)
print(tupToLis)

lisToTup = tuple(lis)
print(lisToTup)
```

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/.spyder-py3/untitled69.py', wdir='C:/Users/
Masego/.spyder-py3')
[12, 31, 45, 34]
('John Doe', 'Michelle Van Der Nest')

In [2]:
```

Example 2



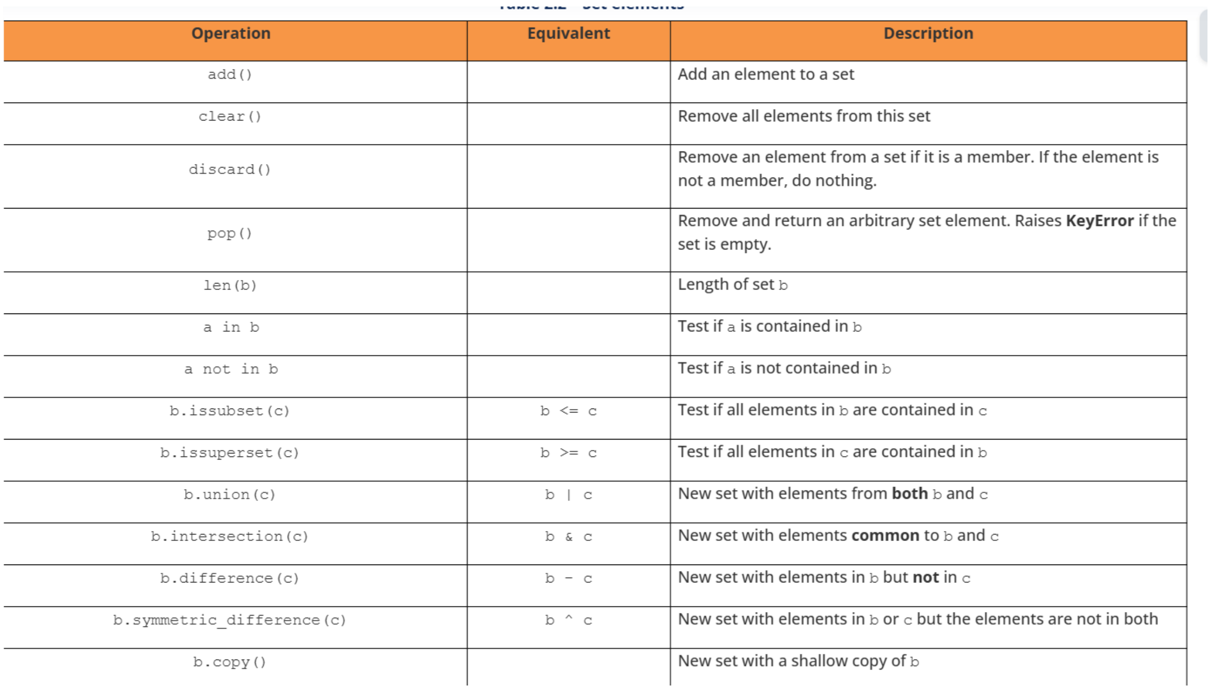
```python
portfolio = []

for i in range(0,3):

    name = input("Enter name for portfolio " + str(i) +": ")

    shares = int(input("Shares for portfolio " + str(i) +": "))

    price = float(input("Price for portfolio " + str(i) +": "))

print ("================================================")


tup= (name,shares,price)    # tuple(name, shares, price)
print(tup)

portfolio.append(tup)        # Append to list of records

print(portfolio)
```

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/.spyder-py3/untitled69.py', wdir='C:/Users/
Masego/.spyder-py3')
[12, 31, 45, 34]
('John Doe', 'Michelle Van Der Nest')

In [2]: runfile('C:/Users/Masego/.spyder-py3/untitled70.py', wdir='C:/Users/
Masego/.spyder-py3')
Enter name for portfolio 0: masego
Shares for portfolio 0: 50
Price for portfolio 0: 1500
Enter name for portfolio 1: masego
Shares for portfolio 1: 50
Price for portfolio 1: 1500
Enter name for portfolio 2: meera
Shares for portfolio 2: 60
Price for portfolio 2: 1800
================================================
```

Sets

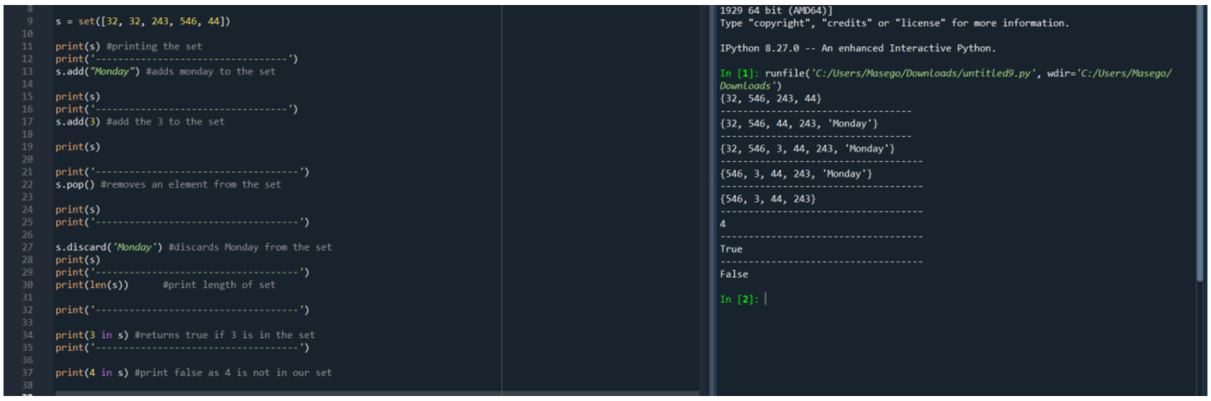
set is an unordered collection of objects that can be contained in a hashtable (also known as directories in Python).

Set Elements



Operation	Equivalent	Description
`add()`		Add an element to a set
`clear()`		Remove all elements from this set
`discard()`		Remove an element from a set if it is a member. If the element is not a member, do nothing.
`pop()`		Remove and return an arbitrary set element. Raises **KeyError** if the set is empty.
`len(b)`		Length of set `b`
`a in b`		Test if `a` is contained in `b`
`a not in b`		Test if `a` is not contained in `b`
`b.issubset(c)`	`b <= c`	Test if all elements in `b` are contained in `c`
`b.issuperset(c)`	`b >= c`	Test if all elements in `c` are contained in `b`
`b.union(c)`	`b | c`	New set with elements from **both** `b` and `c`
`b.intersection(c)`	`b & c`	New set with elements **common** to `b` and `c`
`b.difference(c)`	`b - c`	New set with elements in `b` but **not** in `c`
`b.symmetric_difference(c)`	`b ^ c`	New set with elements in `b` or `c` but the elements are not in both
`b.copy()`		New set with a shallow copy of `b`

Example



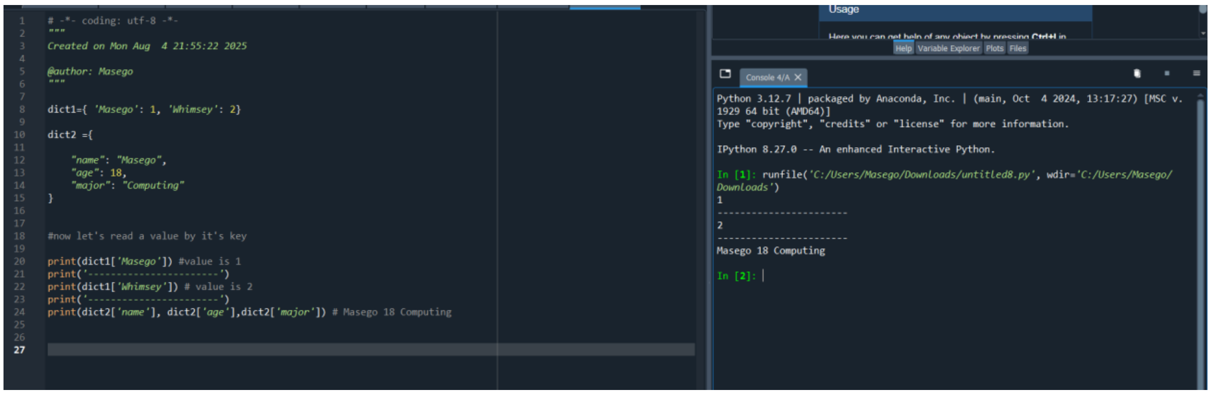
```python
s = set([32, 32, 243, 546, 44])

print(s) #printing the set
print('----------------------------------')
s.add("Monday") #adds monday to the set

print(s)
print('----------------------------------')
s.add(3) #add the 3 to the set

print(s)

print('------------------------------------')
s.pop() #removes an element from the set

print(s)
print('------------------------------------')

s.discard('Monday') #discards Monday from the set
print(s)
print('------------------------------------')
print(len(s))      #print length of set

print('------------------------------------')

print(3 in s) #returns true if 3 is in the set
print('------------------------------------')

print(4 in s) #print false as 4 is not in our set
```

```
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/Downloads/untitled9.py', wdir='C:/Users/Masego/
Downloads')
{32, 546, 243, 44}
----------------------------------
{32, 546, 44, 243, 'Monday'}
----------------------------------
{32, 546, 3, 44, 243, 'Monday'}
------------------------------------
{546, 3, 44, 243, 'Monday'}
------------------------------------
{546, 3, 44, 243}
------------------------------------
4
------------------------------------
True
------------------------------------
False

In [2]: 
```

Dictionary

A dict could also be called an associative container. Other containers, normally sequences, use a numeric index, but a dict‘s index is made up of the key objects. Each key is mapped to the appropriate value. Dictionaries are created by placing a comma between a list of keys and value pairs within braces.

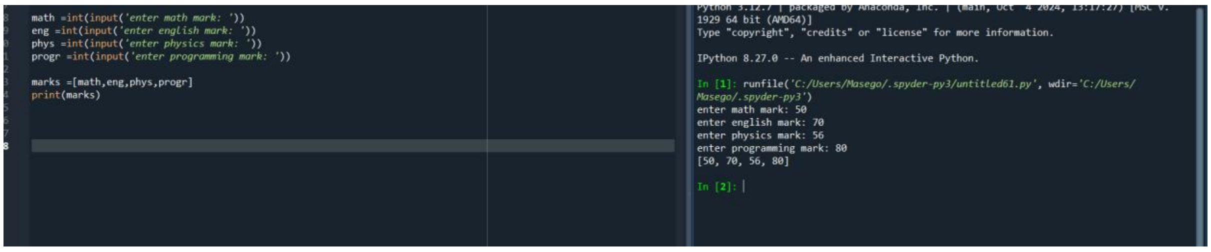
A dict literal is created by surrounding the key and value list with '{}'. A ':' separates the key and value list from each other. The 'key : value' pairs are separated by commas (','). An empty dict is simply '{}'.



```python
# -*- coding: utf-8 -*-
"""
Created on Mon Aug  4 21:55:22 2025

@author: Masego
"""

dict1={ 'Masego': 1, 'Whimsey': 2}

dict2 ={

    "name": "Masego",
    "age": 18,
    "major": "Computing"
}


#now let's read a value by it's key

print(dict1['Masego']) #value is 1
print('----------------------')
print(dict1['Whimsey']) # value is 2
print('----------------------')
print(dict2['name'], dict2['age'],dict2['major']) # Masego 18 Computing
```

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/Downloads/untitled8.py', wdir='C:/Users/Masego/
Downloads')
1
----------------------
2
----------------------
Masego 18 Computing

In [2]:
```

Activity Answer



```python
math =int(input('enter math mark: '))
eng =int(input('enter english mark: '))
phys =int(input('enter physics mark: '))
progr =int(input('enter programming mark: '))

marks =[math,eng,phys,progr]
print(marks)
```

```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.
1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Masego/.spyder-py3/untitled61.py', wdir='C:/Users/
Masego/.spyder-py3')
enter math mark: 50
enter english mark: 70
enter physics mark: 56
enter programming mark: 80
[50, 70, 56, 80]

In [2]:
```

Summary

Structure	Description	Mutable	Ordered	Duplicates	Example
List	Ordered collection	Yes	Yes	Yes	[1, 2, 3]
Tuple	Immutable ordered collection	No	Yes	Yes	(1, 2, 3)

Set	Unordered unique elements	Yes	No	No	{1, 2, 3}
					Set([1,2,3])

Dictionary	Key-value pairs	Yes	Yes	Keys: No	{"age":15, "name":"Masego"}

END