Basics of python

Section one

What python is?

Python is a programming language that is widely used in web applications, software development, data science, and machine learning (ML).

Python is a object-oriented, open-source, high-level programming language with dynamic semantics.

Python handles complex tasks like memory management, allowing developers to focus on coding rather than low-level details.

Python code is executed line by an interpreter, which simplifies the development process and makes debugging easier.

How can declaring a variable in Python?

example:

x = 5 => the variable type is integer

y = " Computer Science“=> variable type is string

Comments in Python:

1) the # used to make a comments in python

2) To add a multiline comment you could insert a # for each line,

=> we can add a multiline string (triple quotes) in your code, and place your comment

inside it for example:

```
"""
MET
Dep. of CS
Level 4
"""
```

Print command in Python?

To print anything in python we use print() function

example:

```
print("MET")
```

How declaring many values to multiple variables in Python?

```
x, y, z = "CS","IS “,"IT"

print(x) CS

print(y) IS

print(z) IT
```

How to assign the same value to multiple variables?

```
x = y = z = "MET"

print(x) MET

print(y) MET

print(z) MET
```

If—else command in Python?

```
degree = 90

if degree >= 85:

 print ("excellent “)

elif degree >= 75 and degree < 85:

 print ("very good “)

elif degree >= 65 and degree < 75:

 print ("good “)

else:

 print ("failed “)
```

Looping in Python:

For loop:

```
sum = 0
for i in range (1, 10):
sum +=i
print (i, sum)
```

While loop:

```
i = 1
while i < 5:
 print(i)
 i += 1
else:
 print ("no result “)
```

How to create a function in python?

```
def functionName ():
 function body
#To call function:
functionName ()
```

Example:

```
x = 5
y = 6
def sumfun ():
 print (x + y)
```

```
#calling function:
sumfun ()
```

Python Libraries?

- **Scikit-learn:**

  A comprehensive library for traditional machine learning algorithms, including classification, regression, clustering, and more.

- **TensorFlow:**

  Popular frameworks for deep learning, offering tools to build and train neural networks.

- **PyTorch:**

  Another widely used open-source deep learning framework known for its flexibility and dynamic computational graph.

- **Pandas:**

  Used for data manipulation and analysis, providing data structures like DataFrames for efficient handling of structured data.

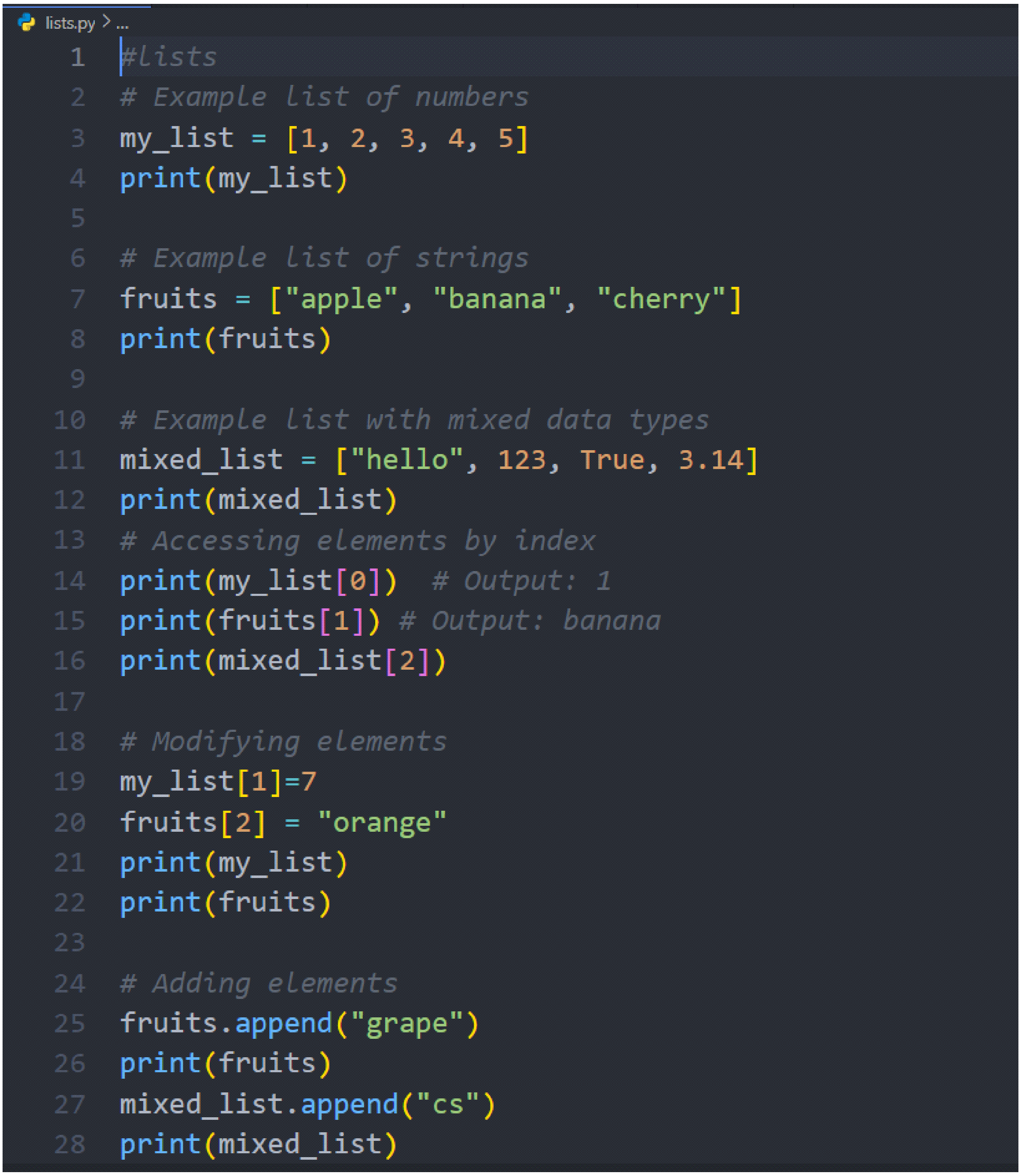
What is Data structure in Python?

- **A data structure in Python** is a way of organizing and storing data in a computer's memory so that it can be accessed and manipulated efficiently.

- **A data structure in Python:**
  - **Mutable:**

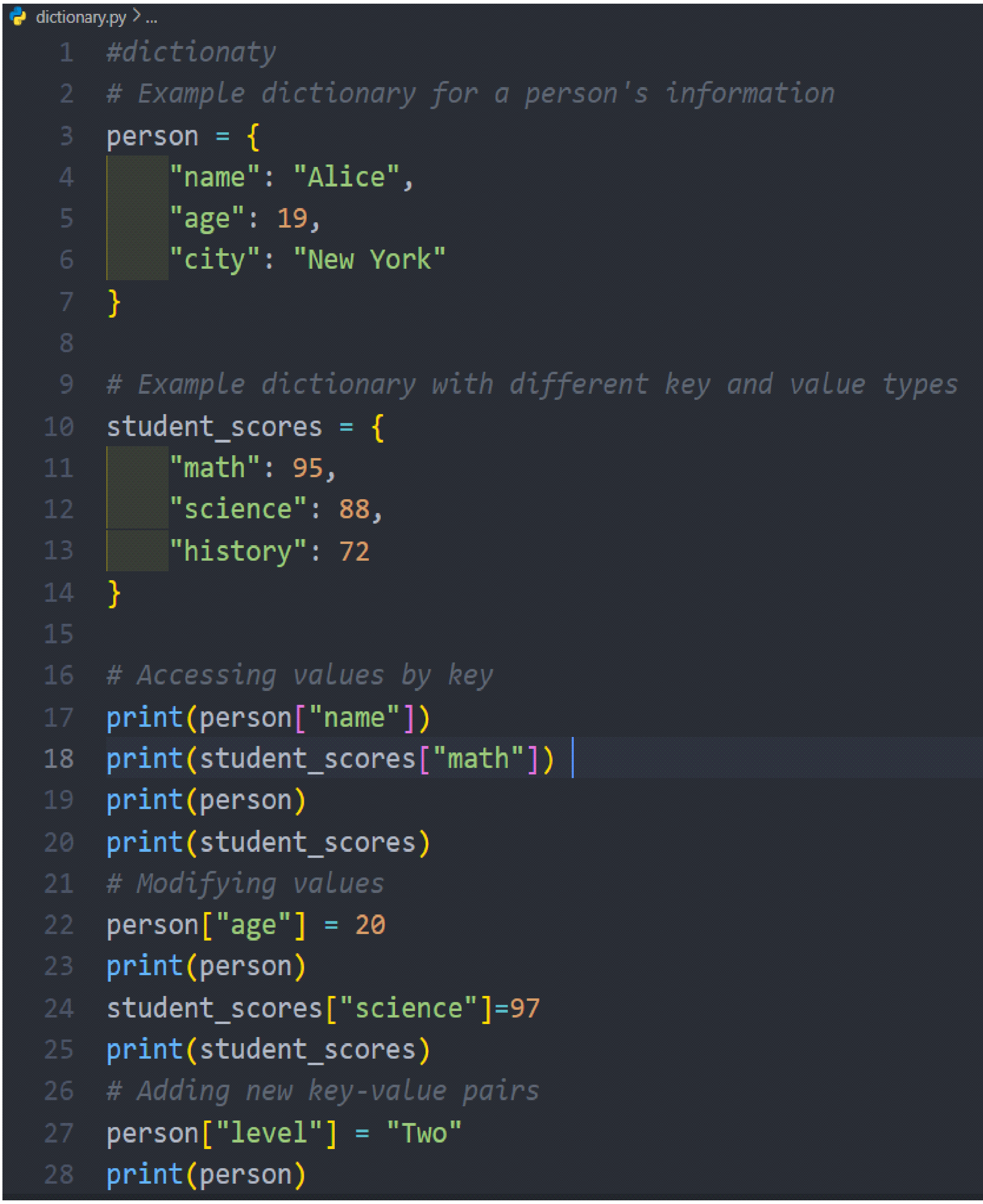
    Python has three mutable data structure (**lists, dictionaries, sets**).

    - **Lists**: Ordered, mutable collections of items. They can contain elements of different data types and are defined using square brackets [].



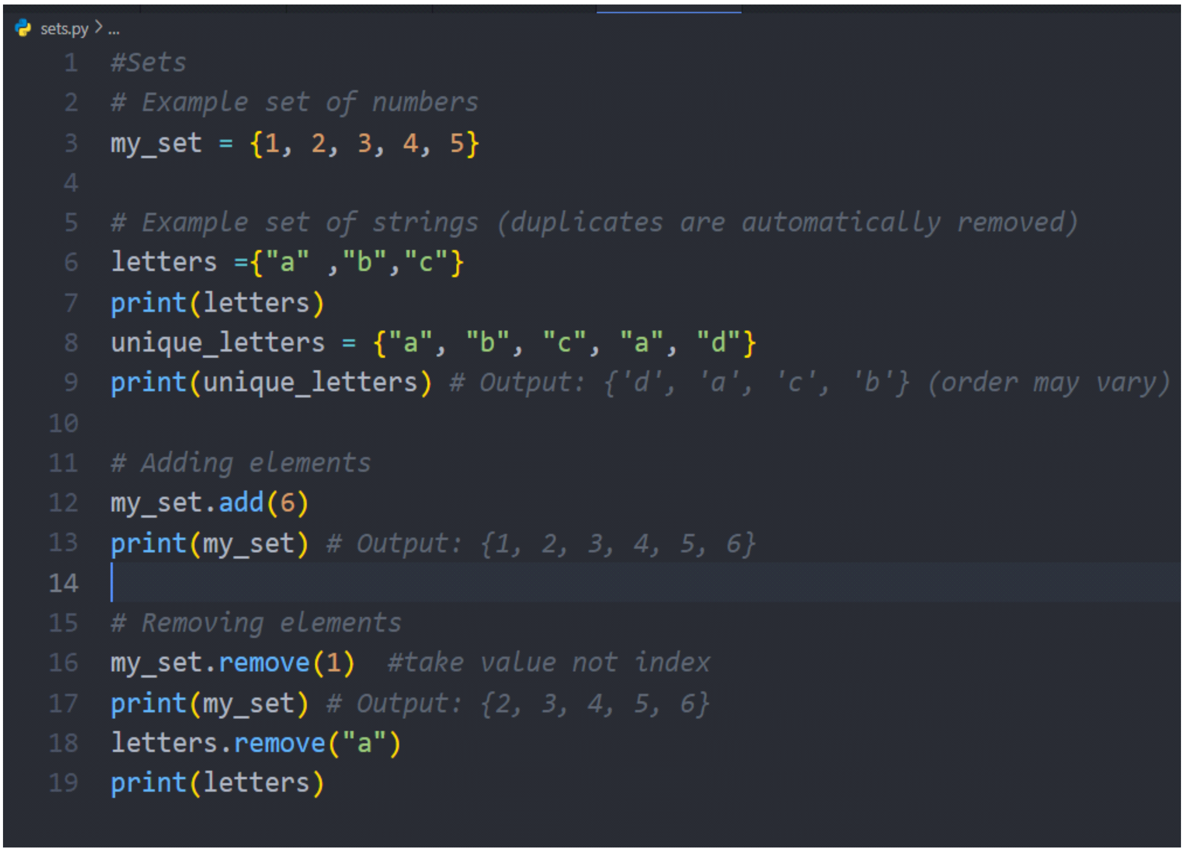
```python
#lists
# Example list of numbers
my_list = [1, 2, 3, 4, 5]
print(my_list)

# Example list of strings
fruits = ["apple", "banana", "cherry"]
print(fruits)

# Example list with mixed data types
mixed_list = ["hello", 123, True, 3.14]
print(mixed_list)
# Accessing elements by index
print(my_list[0])  # Output: 1
print(fruits[1]) # Output: banana
print(mixed_list[2])

# Modifying elements
my_list[1]=7
fruits[2] = "orange"
print(my_list)
print(fruits)

# Adding elements
fruits.append("grape")
print(fruits)
mixed_list.append("cs")
print(mixed_list)
```

- **Dictionaries:** Unordered collections of key-value pairs. Each key must be unique and immutable, while values can be of any data type. They are defined using curly braces {} with key-value pairs separated by colons.



```python
#dictionaty
# Example dictionary for a person's information
person = {
    "name": "Alice",
    "age": 19,
    "city": "New York"
}

# Example dictionary with different key and value types
student_scores = {
    "math": 95,
    "science": 88,
    "history": 72
}

# Accessing values by key
print(person["name"])
print(student_scores["math"])
print(person)
print(student_scores)
# Modifying values
person["age"] = 20
print(person)
student_scores["science"]=97
print(student_scores)
# Adding new key-value pairs
person["level"] = "Two"
print(person)
```

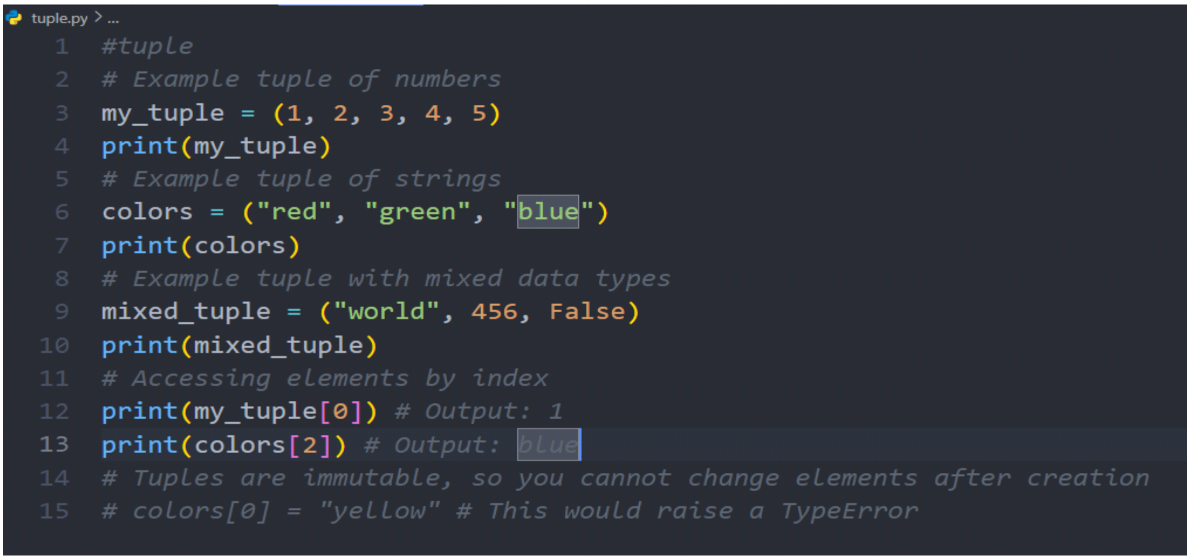
- **Sets:** Unordered collections of unique items. They do not allow duplicate elements and are defined using curly braces{}.



```python
#Sets
# Example set of numbers
my_set = {1, 2, 3, 4, 5}

# Example set of strings (duplicates are automatically removed)
letters ={"a" ,"b","c"}
print(letters)
unique_letters = {"a", "b", "c", "a", "d"}
print(unique_letters) # Output: {'d', 'a', 'c', 'b'} (order may vary)

# Adding elements
my_set.add(6)
print(my_set) # Output: {1, 2, 3, 4, 5, 6}

# Removing elements
my_set.remove(1)  #take value not index
print(my_set) # Output: {2, 3, 4, 5, 6}
letters.remove("a")
print(letters)
```

- **Immutable: the** only basic built-in immutable data structure in python is **(tuple)**.

- **Tuple:** is ordered, immutable sequences of elements.



```python
#tuple
# Example tuple of numbers
my_tuple = (1, 2, 3, 4, 5)
print(my_tuple)
# Example tuple of strings
colors = ("red", "green", "blue")
print(colors)
# Example tuple with mixed data types
mixed_tuple = ("world", 456, False)
print(mixed_tuple)
# Accessing elements by index
print(my_tuple[0]) # Output: 1
print(colors[2]) # Output: blue
# Tuples are immutable, so you cannot change elements after creation
# colors[0] = "yellow" # This would raise a TypeError
```