

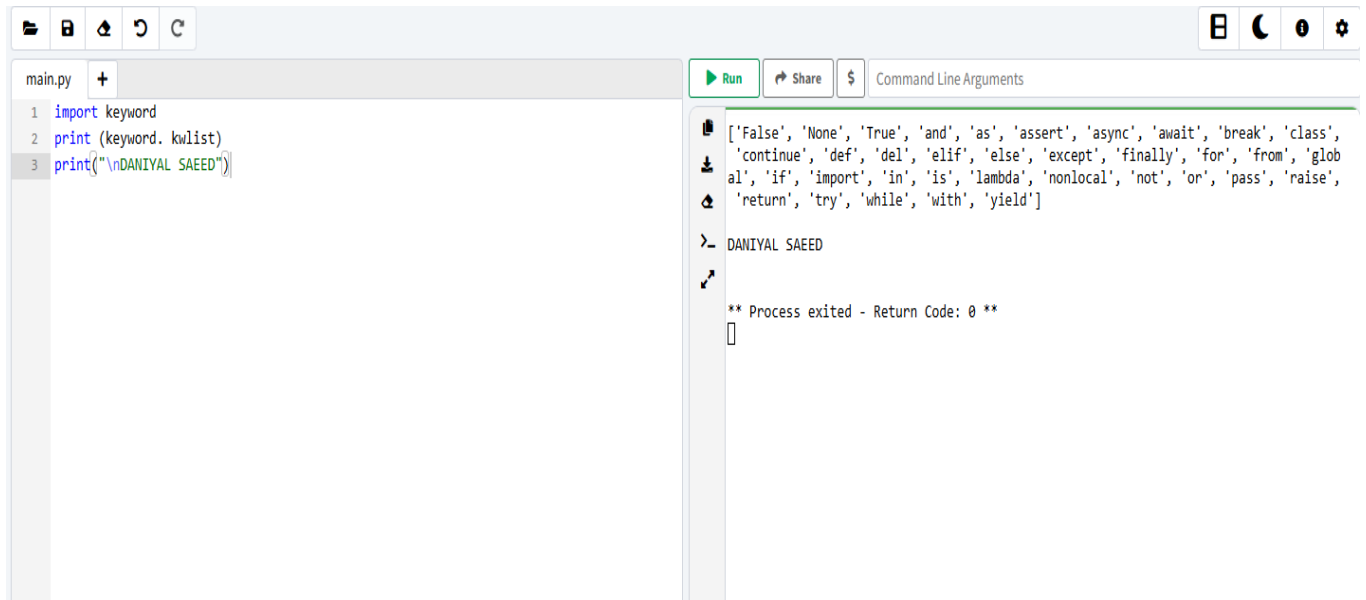
NAME: DANIYAL SAEED

SAP ID: 53937

ASSIGNMENT NO 1

Keywords Exploration:

- Write a Python program to display all reserved keywords using keyword.kwlist.



The screenshot shows a Python IDE interface. On the left, a file named 'main.py' is open, containing the following code:

```
1 import keyword
2 print(keyword.kwlist)
3 print("\nDANIYAL SAEED")
```

On the right, the 'Run' button is highlighted. Below it, the output of the program is displayed in a terminal window. The output shows the list of reserved keywords in Python, followed by the name 'DANIYAL SAEED' on a new line. The list of keywords is: ['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'].

```
[False, 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'glob
al', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
DANIYAL SAEED
** Process exited - Return Code: 0 **
```

- Try to use one keyword as a variable name and observe what happens. Explain the error.

```
main.py
1 def = 10
2 print(def)
3
```

```
File: main.py...1108.1
def = 10
^
SyntaxError: invalid syntax

>
** Process exited - Return Code: 1 **
```

- **Keywords** in Python are reserved words that have special meaning in the language (like `def`, `if`, `else`, `for`, `while`, etc.).
- You **cannot** use these keywords as variable names because Python uses them to understand the structure and flow of the code.
- Using a keyword as a variable name confuses the Python interpreter, so it raises a syntax error.

2. Working with Data Types:

- Create three variables: an integer, a float, and a string.
- Print their values and use the `type()` function to show their data types.
- Convert the float into an integer and print the result.

```
main.py + Run Share $ Command Line Arguments

1 integer = 30
2 floats = 35.3
3 string = 'Rawalpindi'
4 print("\t\t CREATING 3 VARIABLE")
5
6
7 print (integer)
8 print (floats)
9 print (string)
10
11 print("\n")
12 print("\t\t SHOWING VARIABLE TYPE")
13
14 print (type(integer))
15 print (type(floats))
16 print (type(string))
17
18 print("\n")
19 print("\t\t SHOWING INTO INTEGER VAL")
20
21
22 a = int(floats)
23 print(a)
24 print("\n")
25 print("DANIYAL SAEED")
```

CREATING 3 VARIABLE

30
35.3
Rawalpindi

SHOWING VARIABLE TYPE

<class 'int'>
<class 'float'>
<class 'str'>

SHOWING INTO INTEGER VAL

35

DANIYAL SAEED

** Process exited - Return Code: 0 **

3. Escape Sequences in Strings

- Write a program that demonstrates at least four escape sequences:
- Newline (\n)
- Tab (\t)
- Double quotes inside a string (\")
- Backslash (\\)
- Print the results and explain the formatting changes.

```
main.py + Run Share $ Command Line Arguments

1 integer = 30
2 floats = 35.3
3 string = 'Rawalpindi'
4 print("\t\t CREATING 3 VARIABLE")
5
6
7 print (integer)
8 print (floats)
9 print (string)
10
11 print("\n")
12 print("\t\t SHOWING VARIABLE TYPE")
13
14 print (type(integer))
15 print (type(floats))
16 print (type(string))
17
18 print("\n")
19 print("\t\t SHOWING INTO INTEGER VAL")
20
21
22 a = int(floats)
23 print(a)
24 print("\n")
25 print("DANIYAL SAEED")
```

CREATING 3 VARIABLE

30
35.3
Rawalpindi

SHOWING VARIABLE TYPE

<class 'int'>
<class 'float'>
<class 'str'>

SHOWING INTO INTEGER VAL

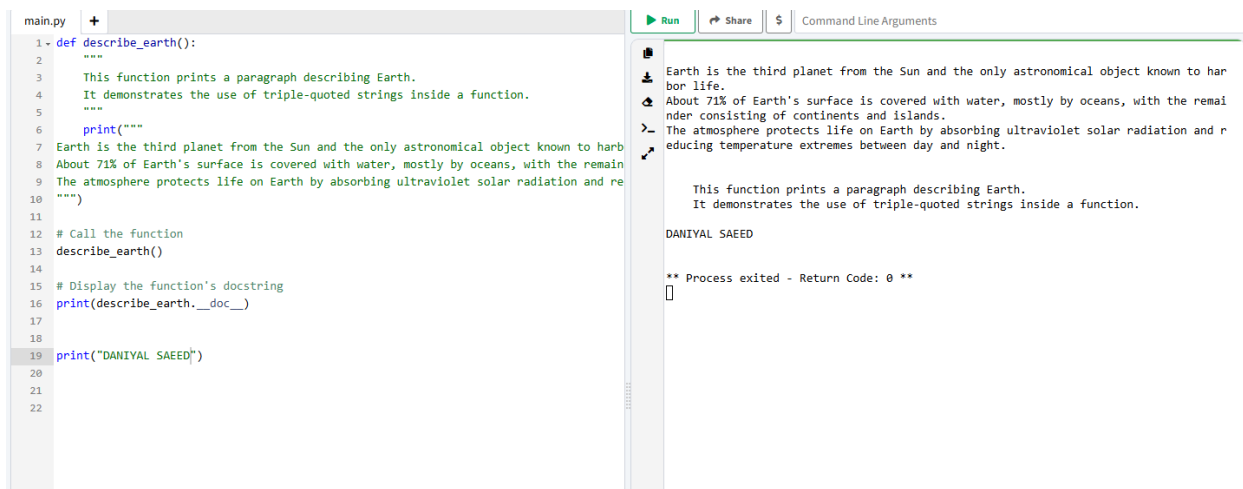
35

DANIYAL SAEED

** Process exited - Return Code: 0 **

4. Triple-Quoted Strings & Docstrings

- Write a function with a docstring that explains what the function does.
- Inside the function, use a triple-quoted string to print a paragraph spanning multiple lines.
- Call the function and display both its output and its `__doc__`.



```
main.py +
1 - def describe_earth():
2     """
3     This function prints a paragraph describing Earth.
4     It demonstrates the use of triple-quoted strings inside a function.
5     """
6     print("""
7     Earth is the third planet from the Sun and the only astronomical object known to harbor
8     life. About 71% of Earth's surface is covered with water, mostly by oceans, with the remainder
9     consisting of continents and islands. The atmosphere protects life on Earth by absorbing ultraviolet solar radiation and
10    reducing temperature extremes between day and night.
11    """)
12    # Call the function
13    describe_earth()
14
15    # Display the function's docstring
16    print(describe_earth.__doc__)
17
18
19    print("DANIYAL SAEED")
20
21
22
```

Run | Share | Command Line Arguments

Earth is the third planet from the Sun and the only astronomical object known to harbor life.
About 71% of Earth's surface is covered with water, mostly by oceans, with the remainder consisting of continents and islands.
The atmosphere protects life on Earth by absorbing ultraviolet solar radiation and reducing temperature extremes between day and night.

This function prints a paragraph describing Earth.
It demonstrates the use of triple-quoted strings inside a function.

DANIYAL SAEED

** Process exited - Return Code: 0 **

5. Perform DMAS Operations

- Create two integer variables, a 15 and b = 4
- Perform and print the results of:
 - D (Division) → a / b (floating-point division) and $a // b$ (integer division)
 - M (Multiplication) $a * b$
 - A (Addition) $a + b$
 - S (Subtraction) $a - b$

main.py

+

Run

Share

\$

Command Line Arguments

```
1 a = 15
2 b = 4
3
4 # Division
5 float_division = a / b      # floating-point division
6 int_division = a // b      # integer division
7
8 # Multiplication
9 multiplication = a * b
10
11 # Addition
12 addition = a + b
13
14 # Subtraction
15 subtraction = a - b
16
17 # Print results
18 print("Division (float):", float_division)
19 print("Division (int):", int_division)
20 print("Multiplication:", multiplication)
21 print("Addition:", addition)
22 print("Subtraction:", subtraction)
23
24
25
26 print("DANIYAL SAEED")
```

Division (float): 3.75

Division (int): 3

Multiplication: 60

Addition: 19

Subtraction: 11

DANIYAL SAEED

** Process exited - Return Code: 0 **