

CECC421: Numerical Analysis

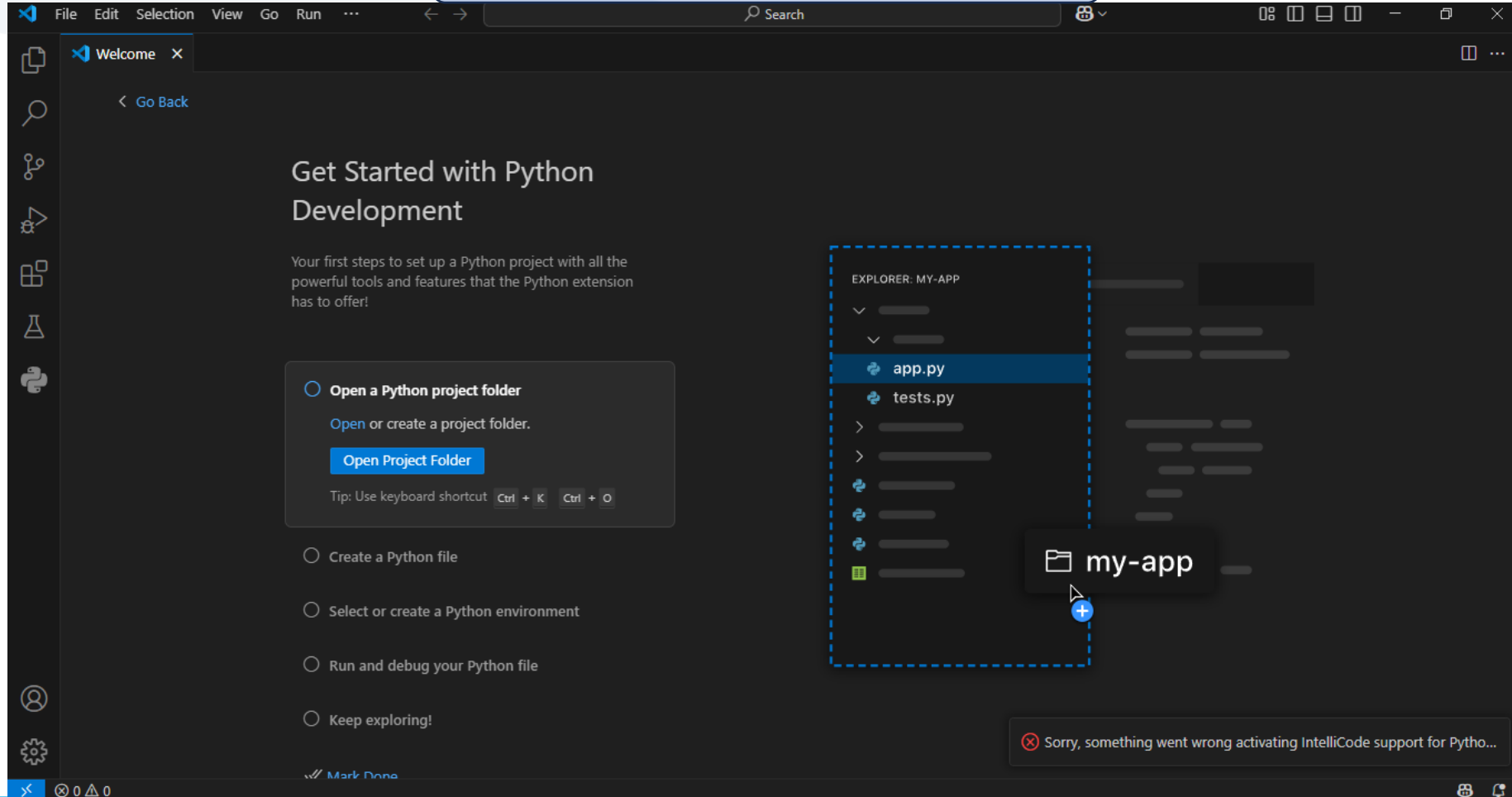
Lecture 1: Python & Visual Studio Code

Eng. Aya Kherbek
Eng. Baher Kherbek
Faculty of Engineering
Department of Informatics
Manara University

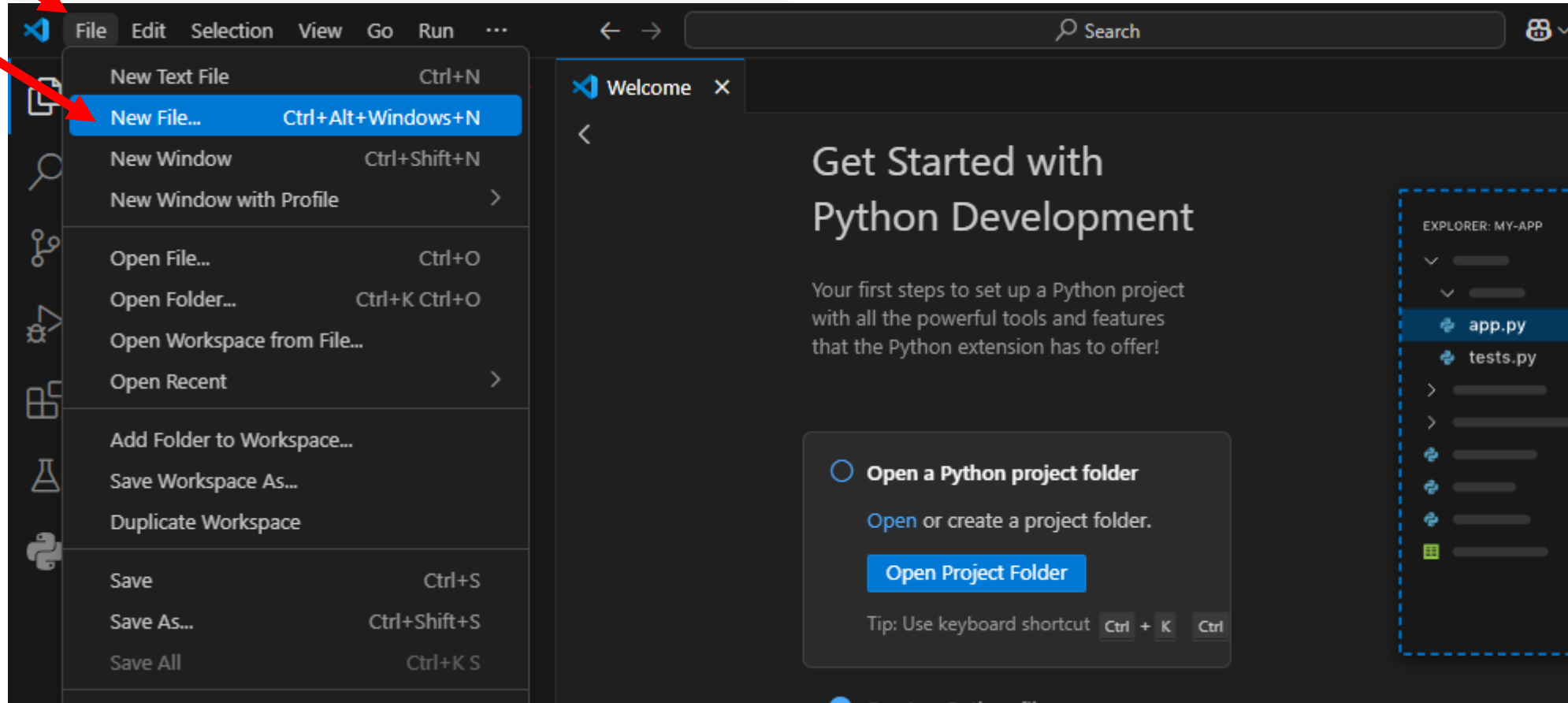
Links to start

1. Download Visual Studio Code to Start Coding, using the following link: <https://code.visualstudio.com/>
2. Insall Python Package using the following link: <https://www.python.org/downloads/>

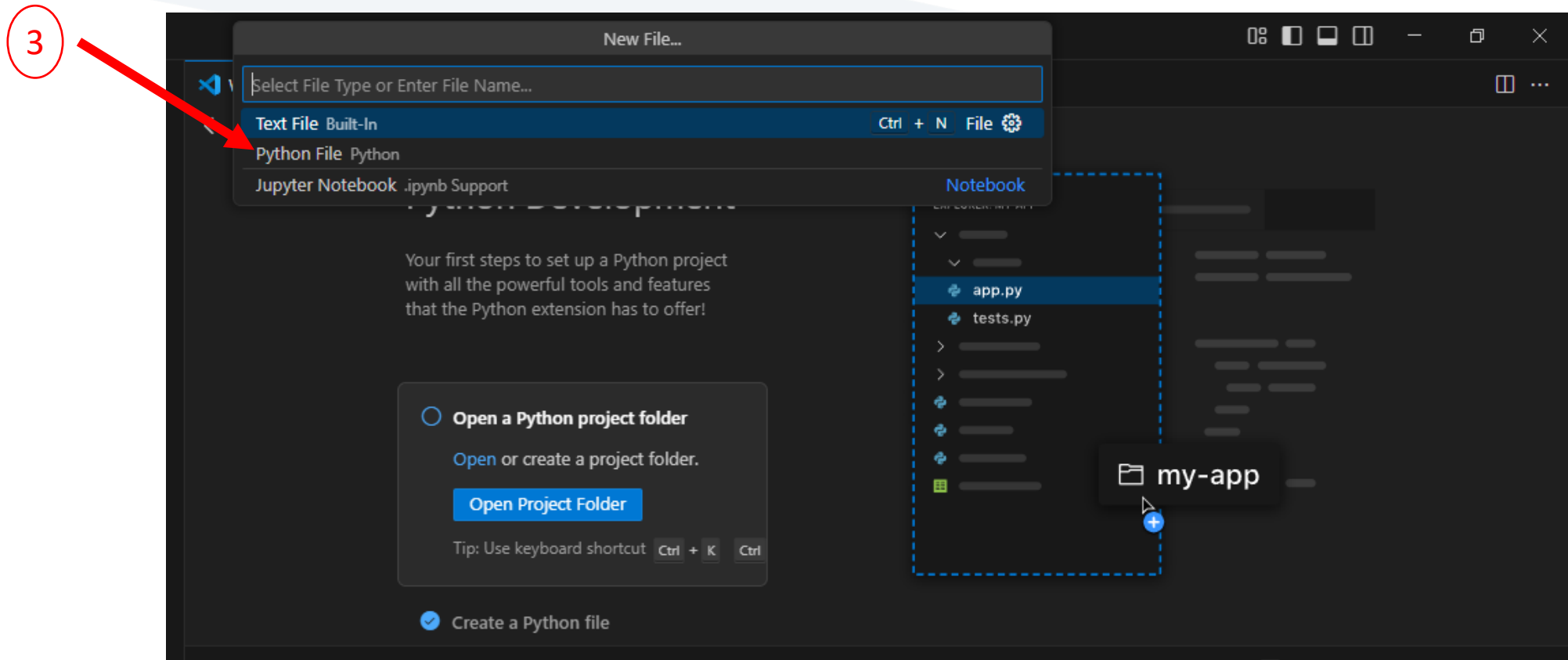
Visual studio Code interface



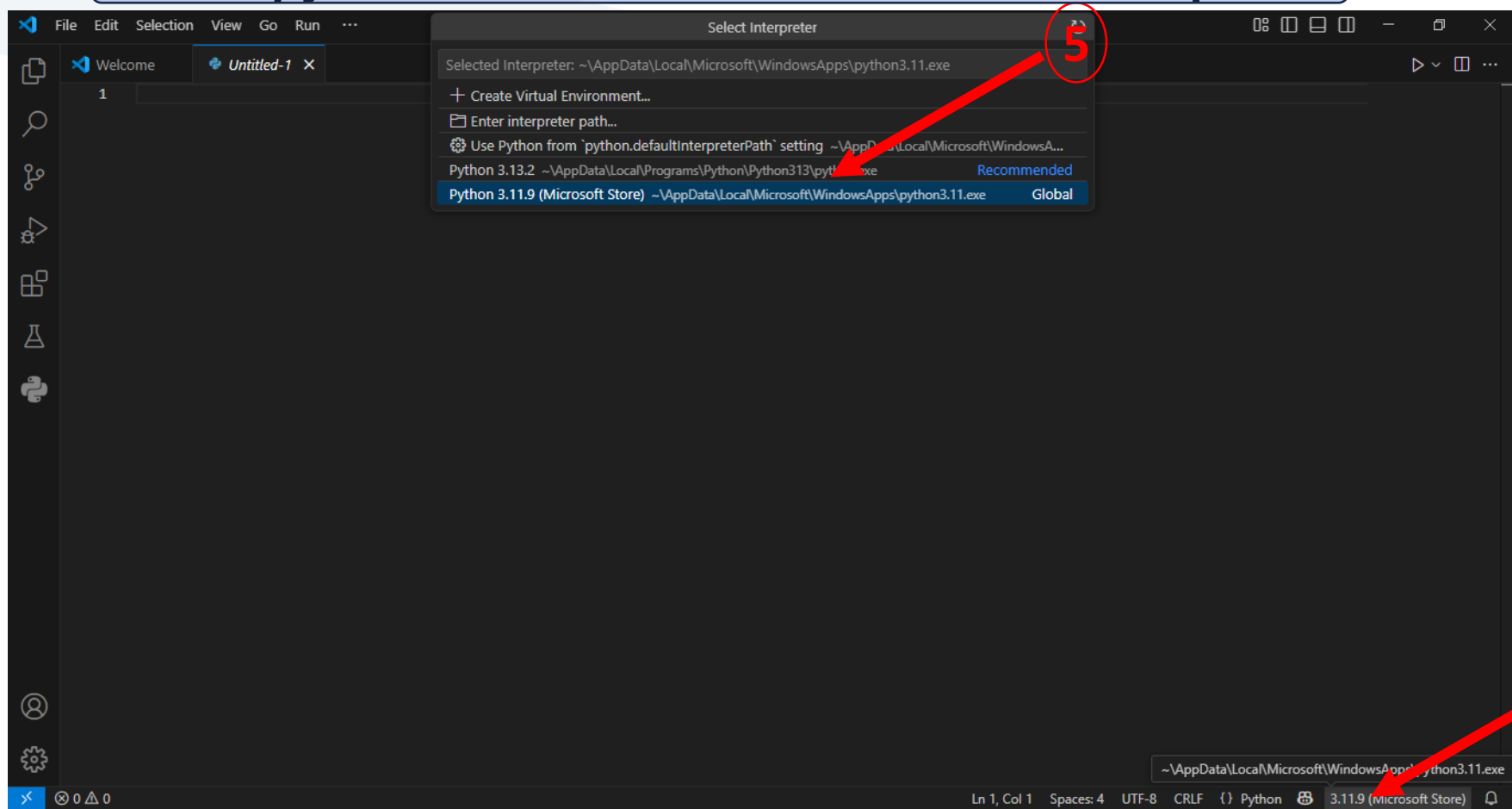
Create python file and choose the correct interpreter



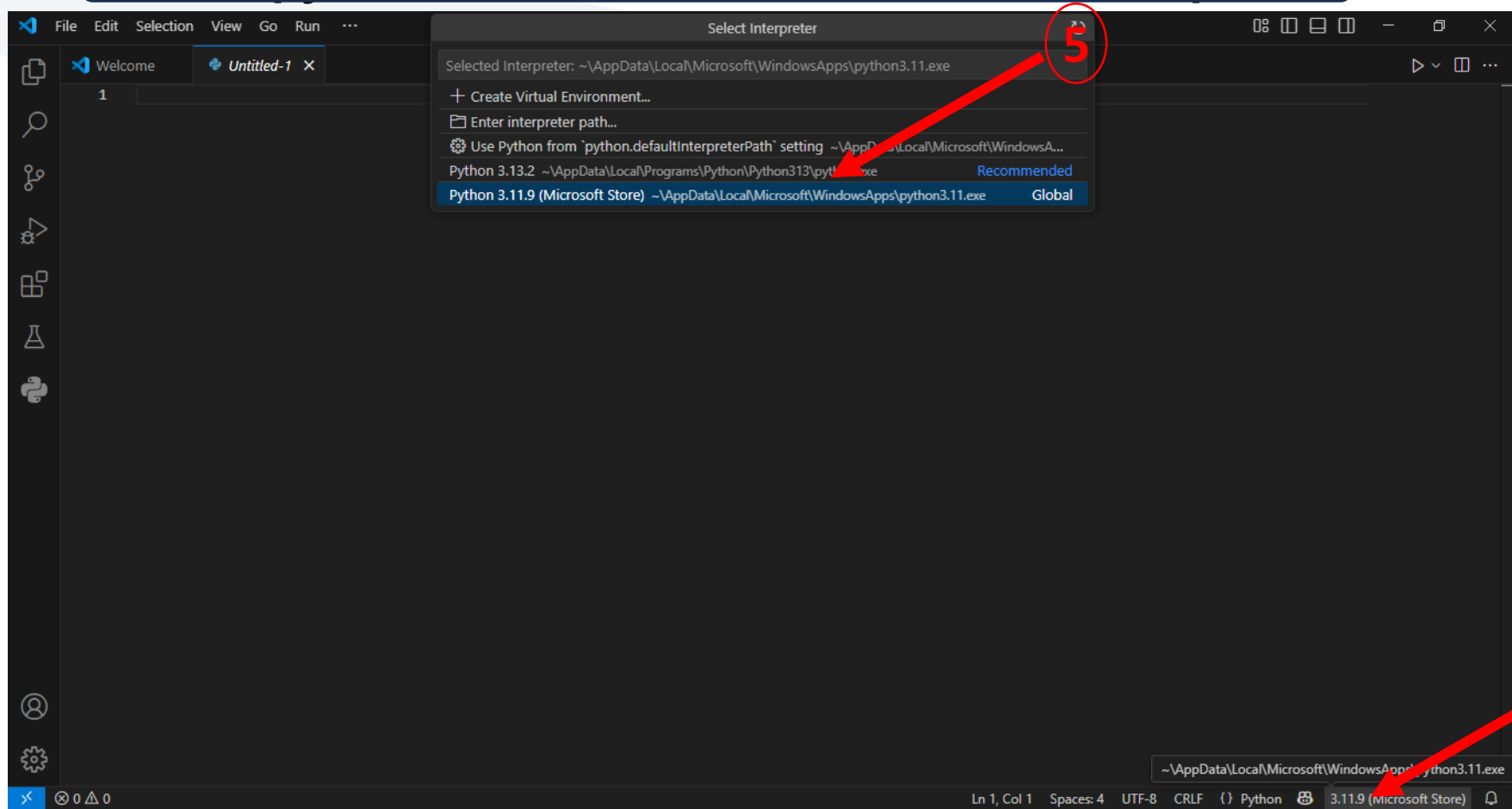
Create python file and choose the correct interpreter



Create python file and choose the correct interpreter

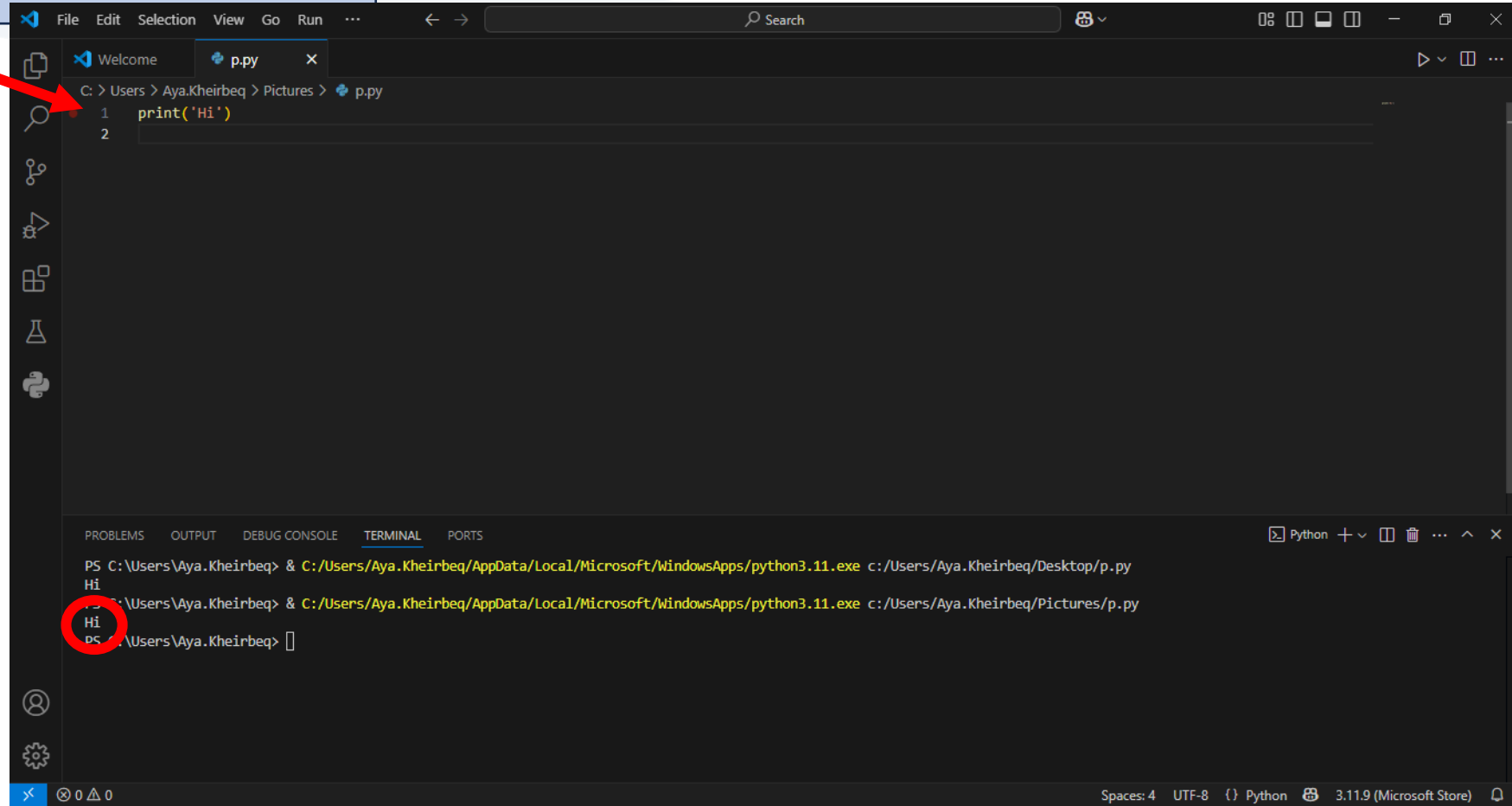


Create python file and choose the correct interpreter



Starting New Program

6



```
File Edit Selection View Go Run ... Search
Welcome p.py x
C: > Users > Aya.Kheirbeq > Pictures > p.py
1 print('Hi')
2

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Desktop/p.py
Hi
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
Hi
PS C:\Users\Aya.Kheirbeq>
```

Print()

>> print()

print() without arguments results in a blank line

⇔ '\n' # Blank line

```
print("Python is powerful")  
print('Python is powerful')
```

With Variable

```
message='Please wait while the program is  
loading...'  
Print(message)
```

With end specifications

```
Print('Good Morning!', end=' ') #print with end whitespace
```

'\n' # Black line

' ' # Empty line

Inline Comment

Different Cases

Print('Good Morning!', end=' ') #print with end whitespace

'\n' # Black line

' ' # Empty line

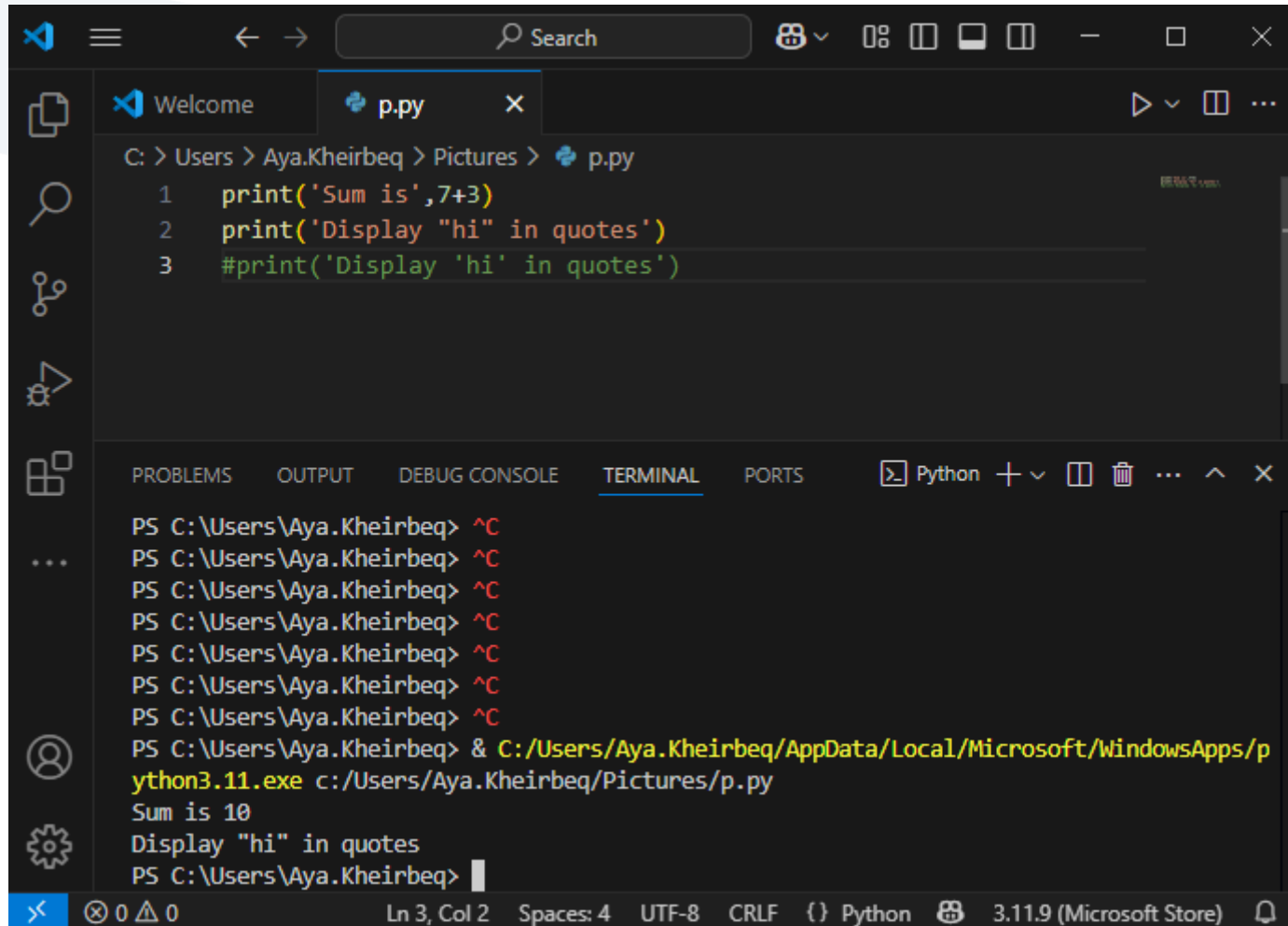
Different Cases

```
print(What do you call bear with no teeth ? A gummy bear\n')
```

What do you call bear with no teeth ? A gummy bear

```
print(What do you call bear with no teeth ?\n A gummy bear')
```

What do you call bear with no teeth ?
A gummy bear



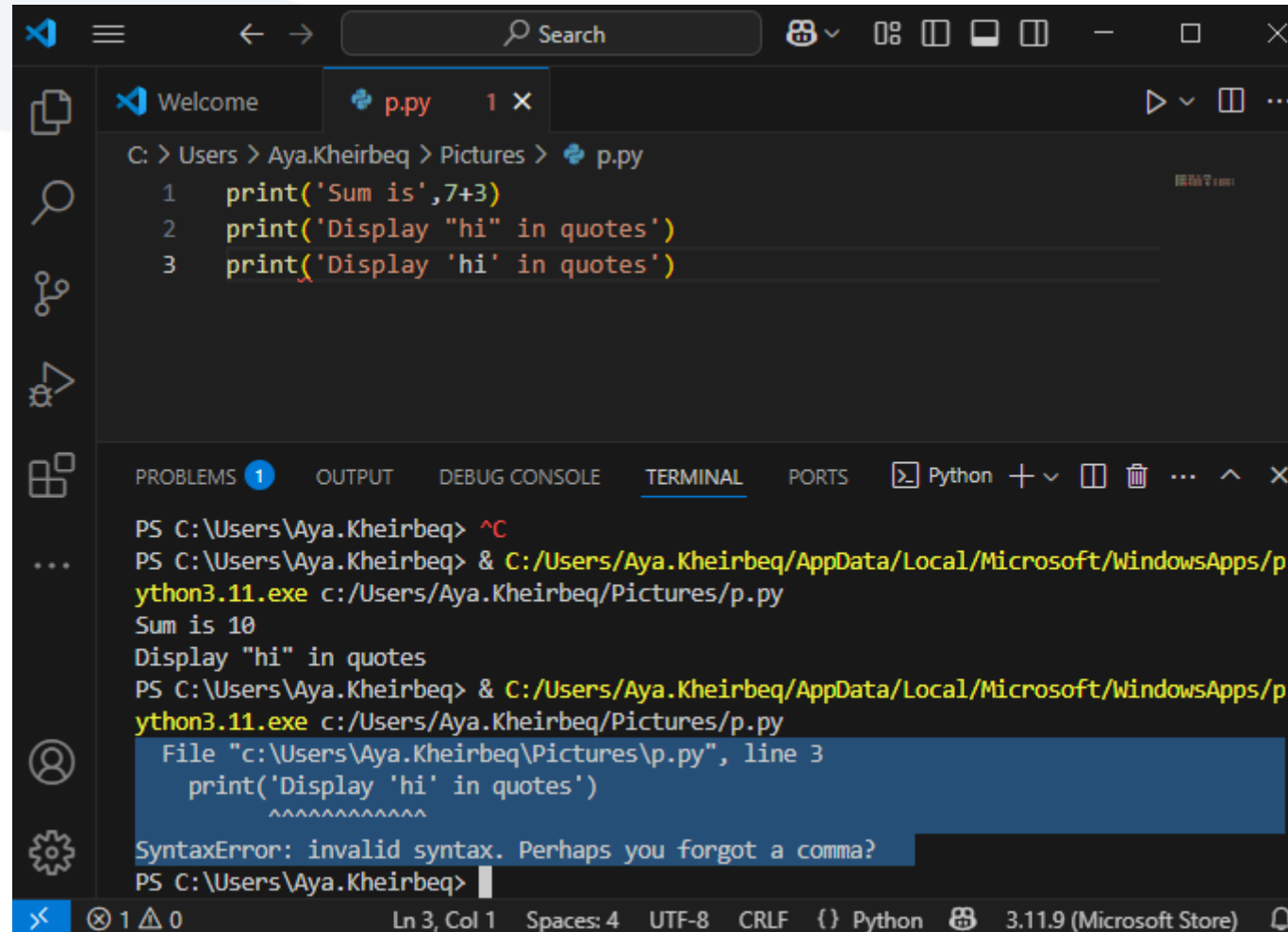
The screenshot shows the Visual Studio Code interface. The editor window displays a file named `p.py` with the following code:

```
1 print('Sum is',7+3)
2 print('Display "hi" in quotes')
3 #print('Display 'hi' in quotes')
```

The terminal window at the bottom shows the execution of the script using Python 3.11.9. The output is:

```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
Sum is 10
Display "hi" in quotes
PS C:\Users\Aya.Kheirbeq>
```

The status bar at the bottom indicates the file is a Python script (3.11.9 (Microsoft Store)) and shows the current cursor position (Ln 3, Col 2).



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `p.py` with the following code:

```
1 print('Sum is',7+3)
2 print('Display "hi" in quotes')
3 print('Display 'hi' in quotes')
```

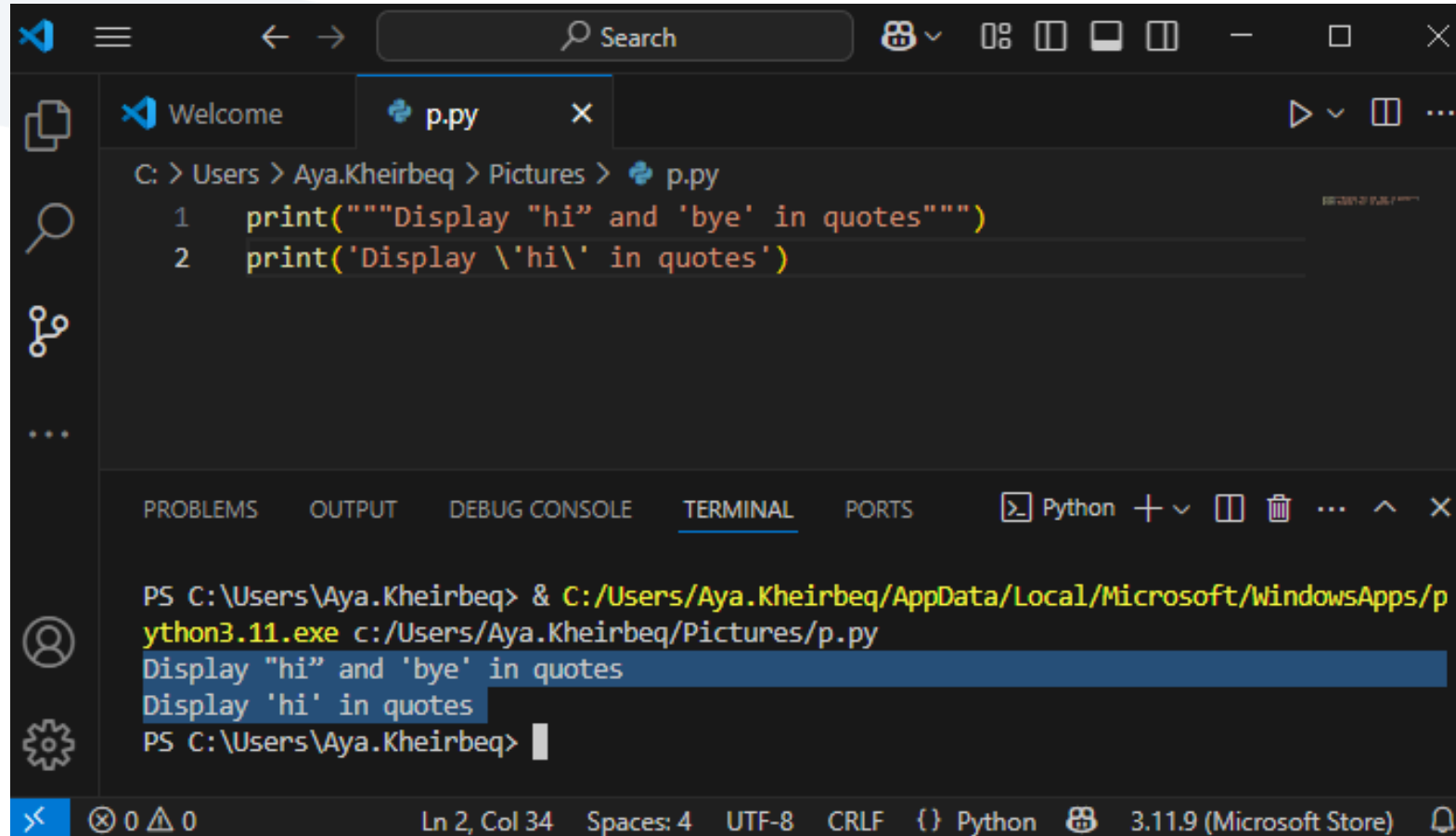
The terminal window shows the execution of the script. The first two lines execute successfully, outputting:

```
Sum is 10
Display "hi" in quotes
```

The third line causes a `SyntaxError` due to an invalid syntax in the string: `print('Display 'hi' in quotes')`. The error message is:

```
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

The status bar at the bottom indicates the file is at line 3, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and is a Python 3.11.9 file from the Microsoft Store.



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `p.py` with the following code:

```
1 print("""Display "hi" and 'bye' in quotes""")
2 print('Display \'hi\' in quotes')
```

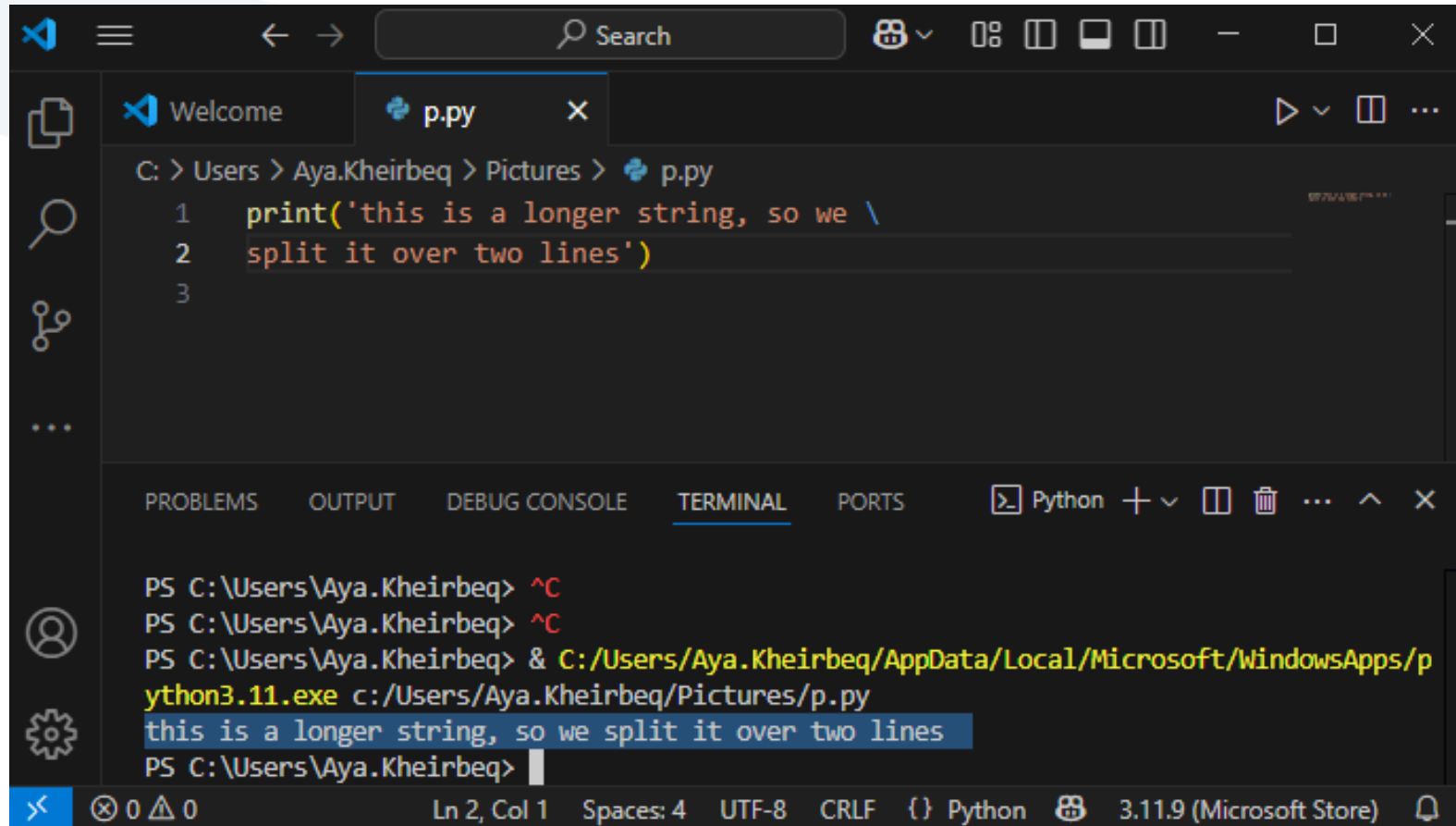
The terminal window at the bottom shows the command to run the script using Python 3.11.9 from the Microsoft Store:

```
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
```

The output of the script is displayed in the terminal:

```
Display "hi" and 'bye' in quotes
Display 'hi' in quotes
```

The status bar at the bottom indicates the file is `p.py`, line 2, column 34, using UTF-8 encoding and CRLF line endings.



```
C: > Users > Aya.Kheirbeq > Pictures > p.py
1  print('this is a longer string, so we \
2  split it over two lines')
3

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python + v [icon] [icon] ... ^ X

PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/p
ython3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
this is a longer string, so we split it over two lines
PS C:\Users\Aya.Kheirbeq>
```

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store)

Within Python, a backslash (\) means a continuation character. Also, if it is set at the edge of a line, it is estimated that the line is continued, disregarding following newlines.

Variable Names

A variable can have a short name (like `x` and `y`) or a more descriptive name (`age`, `carname`, `total_volume`).

- Rules for Python variables:
 - A variable name must start with a letter or the underscore character
 - A variable name cannot start with a number
 - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and `_`)
 - Variable names are case-sensitive (`age`, `Age` and `AGE` are three different variables).

Variable Names

اسماء المتحولات المسموحة :

- `myvar = "John"`
- `my_var = "John"`
- `_my_var = "John"`
- `myVar = "John"`
- `MYVAR = "John"`
- `myvar2 = "John"`

اسماء المتحولات غير المسموحة :

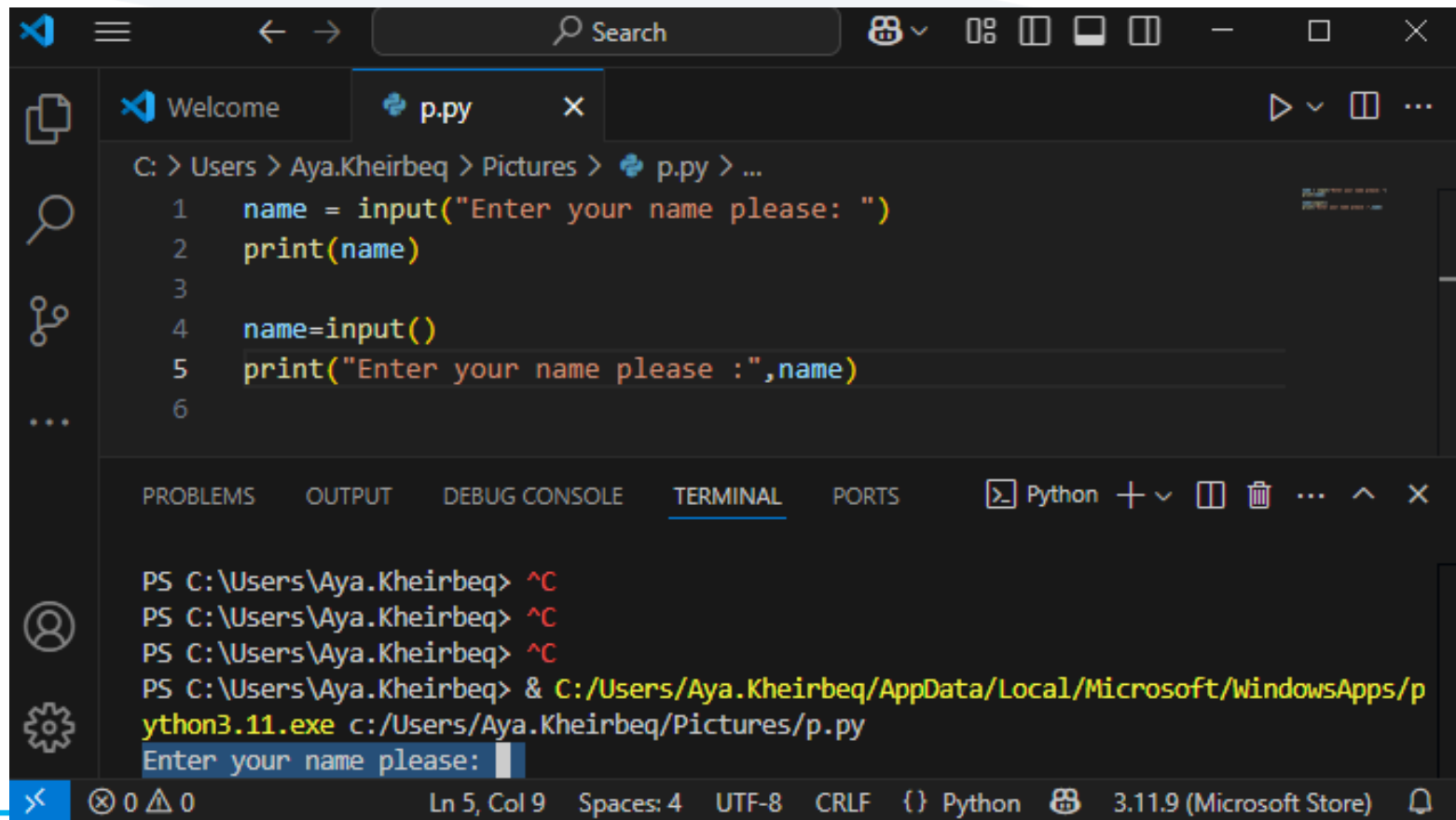
- `2myvar = "John"`
- `my-var = "John"`
- `my var = "John"`

Input

```
name = input("Enter your name please: ")  
print(name)
```

```
name=input()  
print("Enter your name please :",name)
```

Input



The screenshot shows the Visual Studio Code interface. The editor window displays a file named `p.py` with the following Python code:

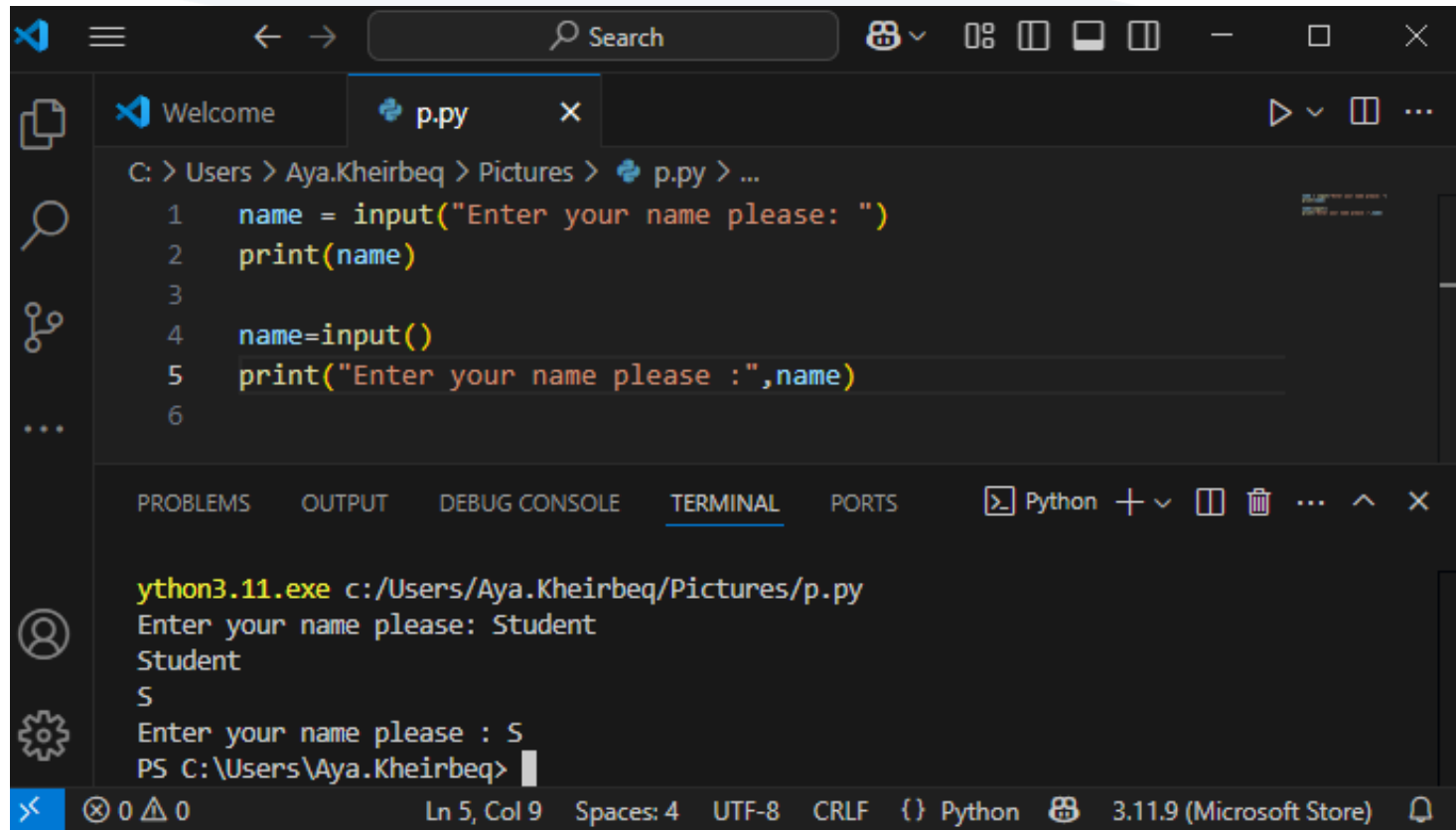
```
1 name = input("Enter your name please: ")
2 print(name)
3
4 name=input()
5 print("Enter your name please :",name)
6
```

The bottom panel shows the `TERMINAL` tab with the following commands and output:

```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
Enter your name please: 
```

The status bar at the bottom indicates the current position is `Ln 5, Col 9`, with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the file is a `Python` file using `3.11.9 (Microsoft Store)` interpreter.

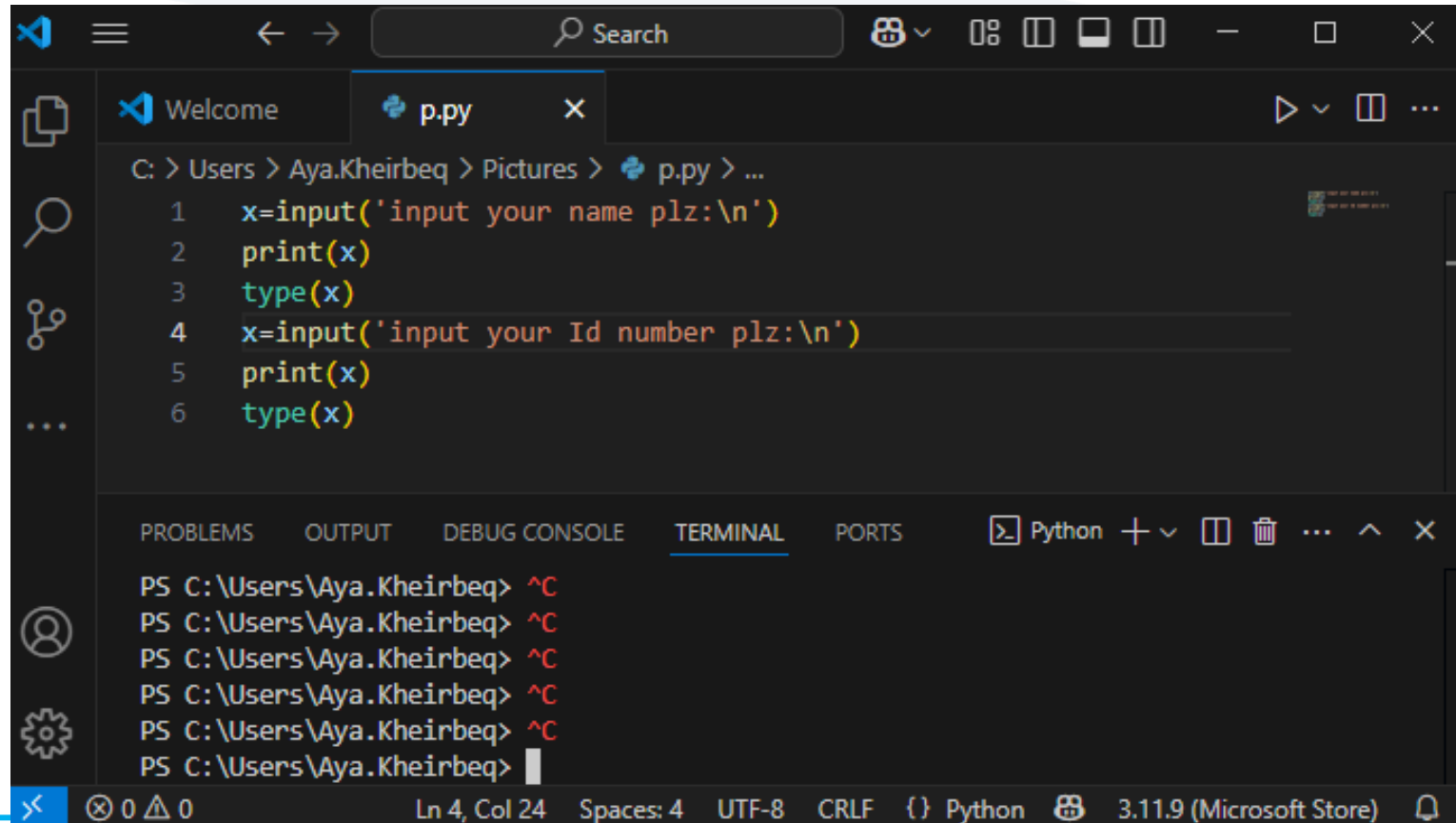
Input



```
C: > Users > Aya.Kheirbeq > Pictures > p.py > ...  
1  name = input("Enter your name please: ")  
2  print(name)  
3  
4  name=input()  
5  print("Enter your name please :",name)  
6  
  
Python  
python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py  
Enter your name please: Student  
Student  
S  
Enter your name please : S  
PS C:\Users\Aya.Kheirbeq>
```

Ln 5, Col 9 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store)

Controlling Input type

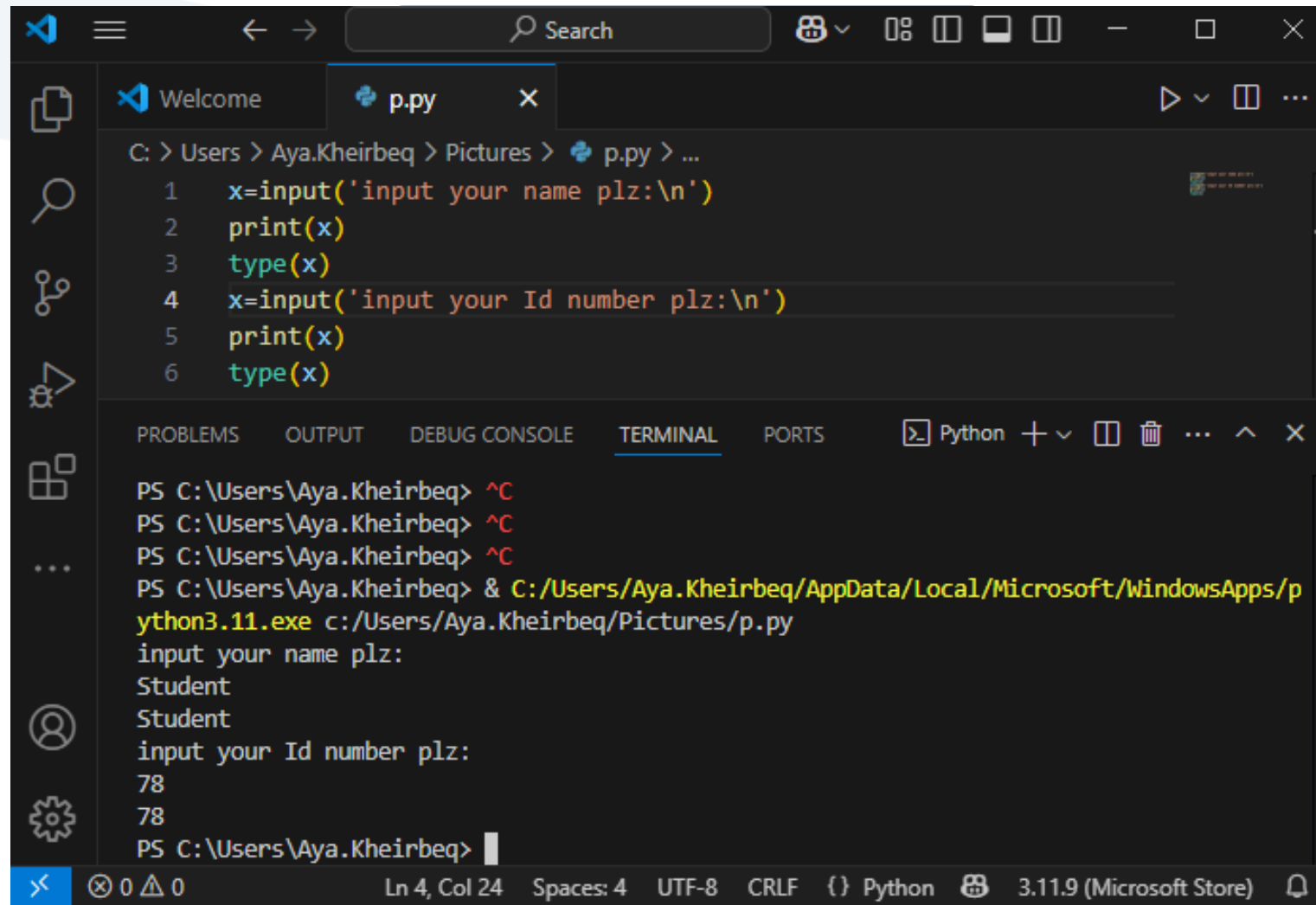


```
C: > Users > Aya.Kheirbeq > Pictures > p.py > ...
1  x=input('input your name plz:\n')
2  print(x)
3  type(x)
4  x=input('input your Id number plz:\n')
5  print(x)
6  type(x)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - Python 3.11.9 (Microsoft Store)

```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> 
```

Ln 4, Col 24 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store)



The screenshot shows the Visual Studio Code interface. The editor window displays a Python file named `p.py` with the following code:

```
1 x=input('input your name plz:\n')
2 print(x)
3 type(x)
4 x=input('input your Id number plz:\n')
5 print(x)
6 type(x)
```

The terminal window at the bottom shows the command prompt output:

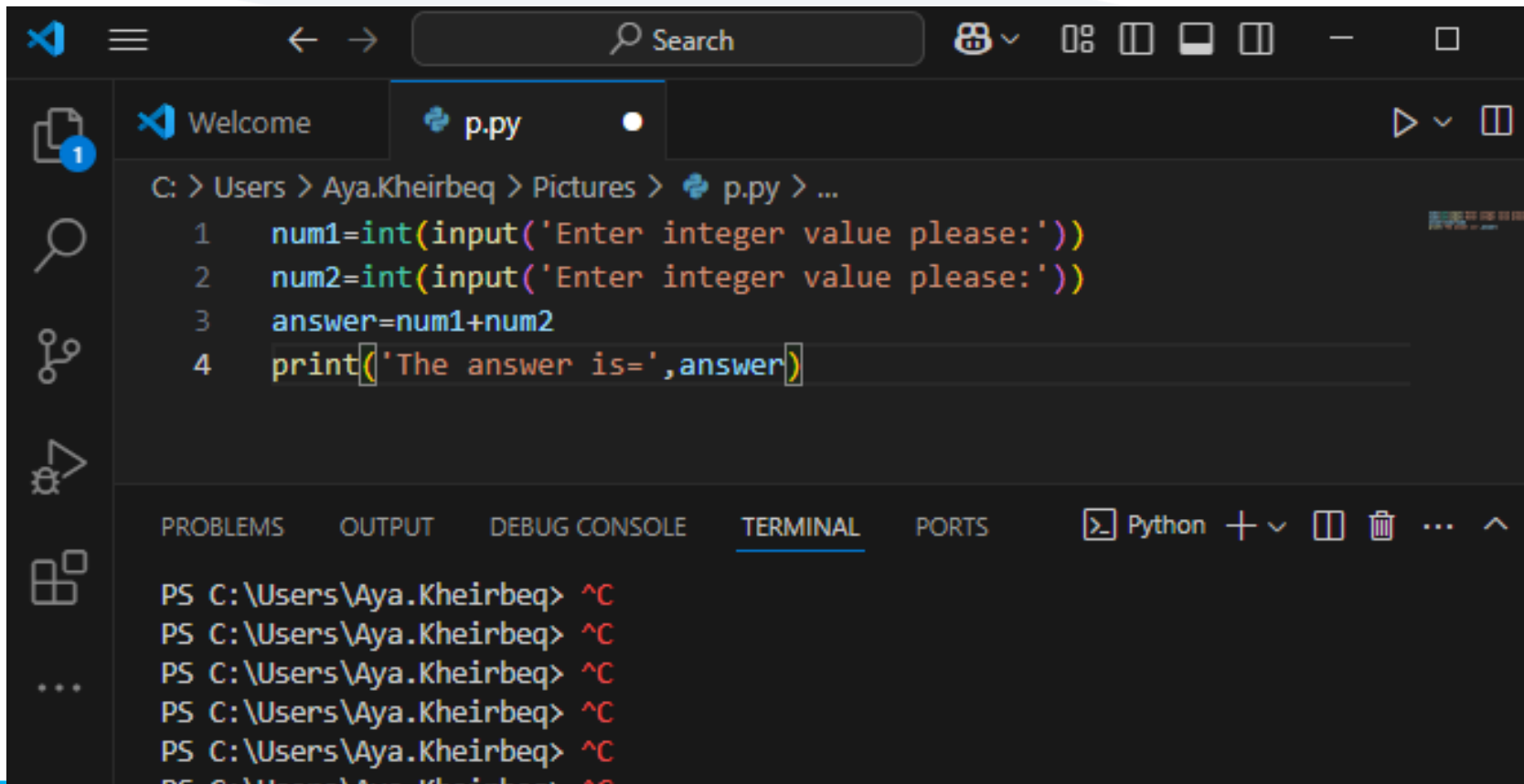
```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
input your name plz:
Student
Student
input your Id number plz:
78
78
PS C:\Users\Aya.Kheirbeq>
```

The status bar at the bottom indicates the file is at line 4, column 24, using UTF-8 encoding, CRLF line endings, and is a Python 3.11.9 file from the Microsoft Store.

Casting

What about numbers?

Casting

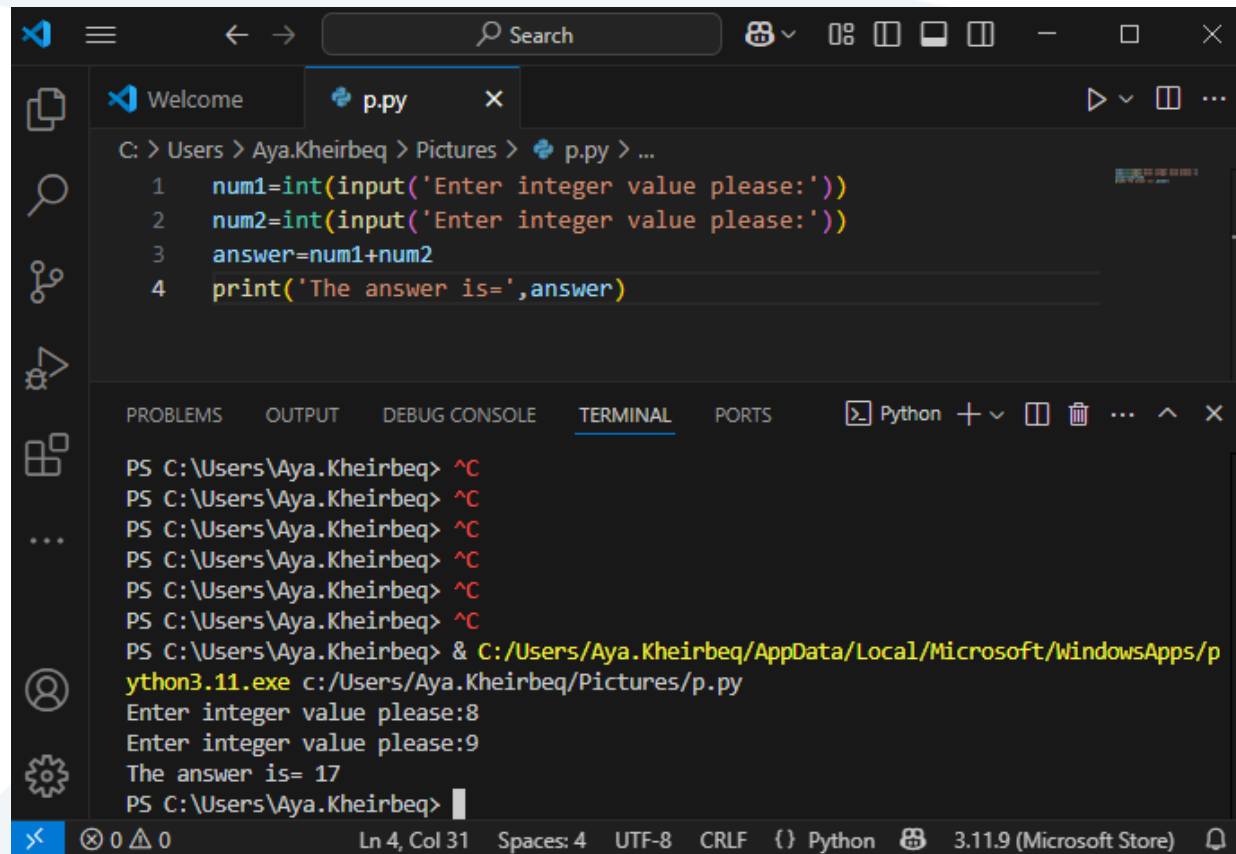


```
1 num1=int(input('Enter integer value please:'))
2 num2=int(input('Enter integer value please:'))
3 answer=num1+num2
4 print('The answer is=',answer)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [icon] [icon] ... ^

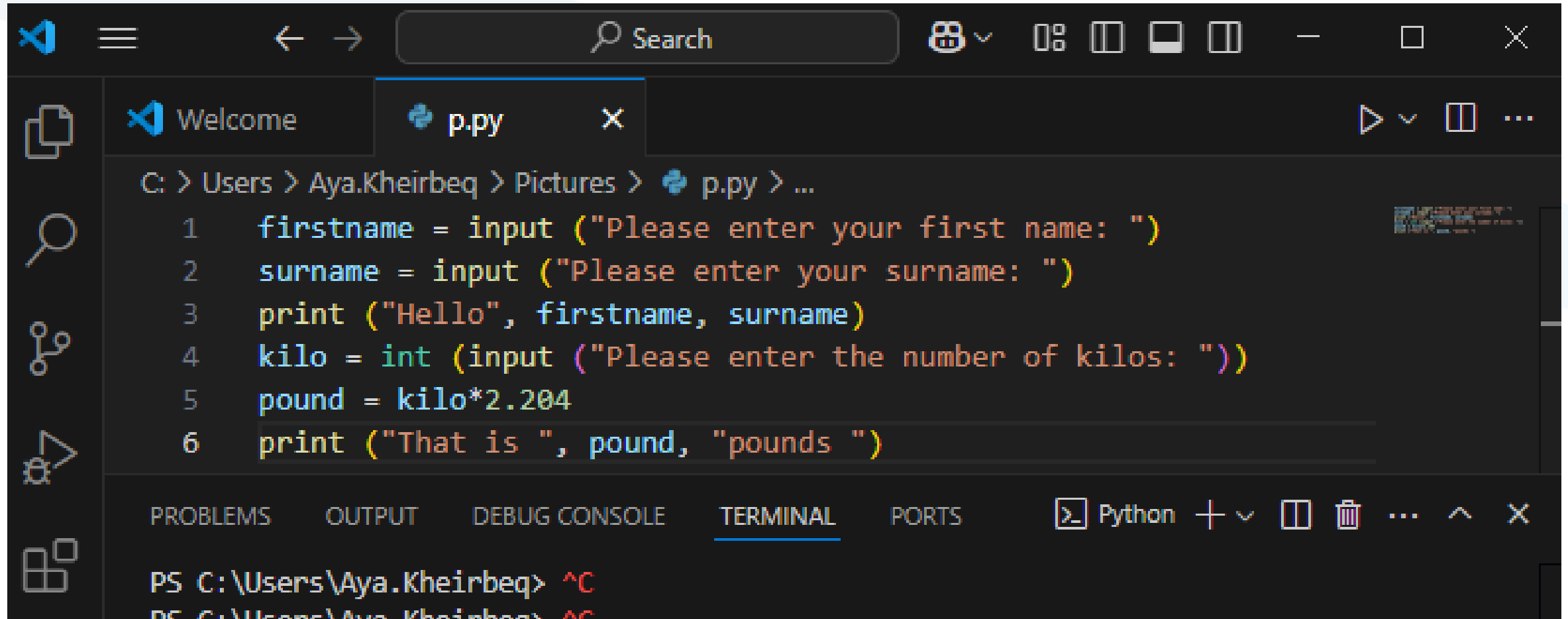
```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
```

Casting

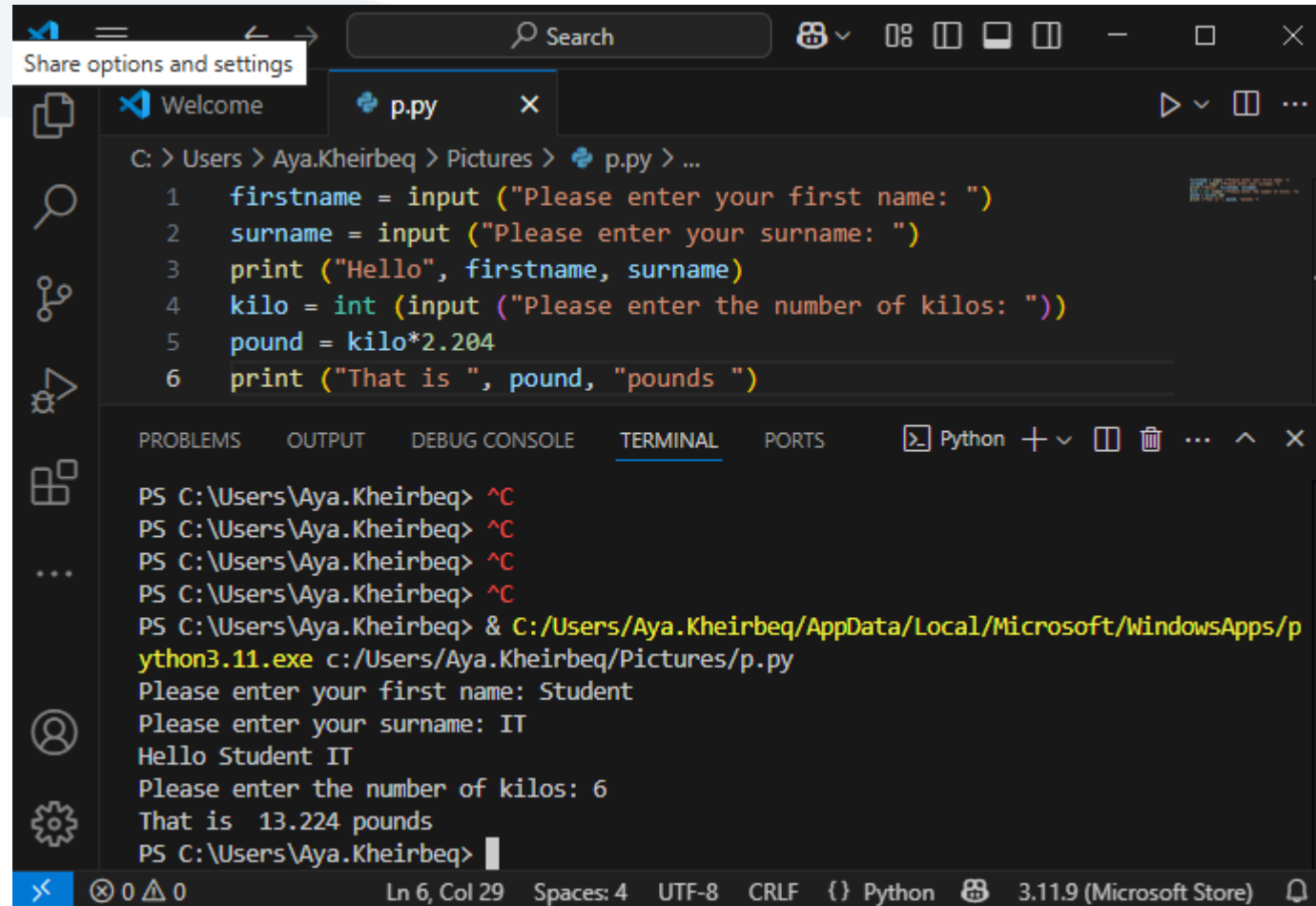


```
1 num1=int(input('Enter integer value please:'))
2 num2=int(input('Enter integer value please:'))
3 answer=num1+num2
4 print('The answer is=',answer)
```

```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
Enter integer value please:8
Enter integer value please:9
The answer is= 17
PS C:\Users\Aya.Kheirbeq>
```



```
C: > Users > Aya.Kheirbeq > Pictures > p.py > ...  
1  firstname = input ("Please enter your first name: ")  
2  surname = input ("Please enter your surname: ")  
3  print ("Hello", firstname, surname)  
4  kilo = int (input ("Please enter the number of kilos: "))  
5  pound = kilo*2.204  
6  print ("That is ", pound, "pounds ")  
  
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python + v [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
  
PS C:\Users\Aya.Kheirbeq> ^C  
PS C:\Users\Aya.Kheirbeq> ^C
```



The screenshot shows the Visual Studio Code interface. The top bar includes a search bar and window management icons. The left sidebar shows the Explorer, Search, and Run and Debug views. The main editor area displays a Python file named `p.py` with the following code:

```
1  firstname = input ("Please enter your first name: ")
2  surname = input ("Please enter your surname: ")
3  print ("Hello", firstname, surname)
4  kilo = int (input ("Please enter the number of kilos: "))
5  pound = kilo*2.204
6  print ("That is ", pound, "pounds ")
```

Below the editor, the TERMINAL view is active, showing the command prompt output:

```
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> ^C
PS C:\Users\Aya.Kheirbeq> & C:/Users/Aya.Kheirbeq/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Aya.Kheirbeq/Pictures/p.py
Please enter your first name: Student
Please enter your surname: IT
Hello Student IT
Please enter the number of kilos: 6
That is  13.224 pounds
PS C:\Users\Aya.Kheirbeq>
```

The status bar at the bottom indicates the current line and column (Ln 6, Col 29), encoding (UTF-8), line endings (CRLF), and the Python version (3.11.9).

Data Types

Numeric Data Types	String Data Types	Boolean Type
Integer -,0,+ <code>>>> print(42)</code> # <class 'int'> 42 Note $8 \neq 8.0$ integer float	Str <code>>>> print('hello')</code> # <class 'str'> hello Type (Rami) str	Bool True False <code>>>> print(True)</code> # <class 'bool'> True <code>>>> my_bool = 5 > 8</code> <code>>>> print(my_bool)</code> False
Float 8.0 <code>>>> print(3.14)</code> # <class 'float'> 3.14	<code>x=7+6j</code> type (x) complex <code>x=8</code> type (x) Int	
Complex 5+9j <code>>>> print(1 + 2j)</code> # <class 'complex'> (1+2j)	type (10.5) float	

Data Types

Sequence Types	Set Data Types
List <pre>>>> print([1, 2, 3]) # <class 'list'> [1, 2, 3]</pre>	Set <pre>>>> print({'red', 'green', 'blue'}) # <class 'set'> {'red', 'green', 'blue'}</pre>
Tuples <pre>>>> print((1, 2, 3)) # <class 'tuple'> (1, 2, 3)</pre>	frozenset
Range <pre>range(start, stop, step) >>> range(2, 6, 2) 2, 4 >>> range(25, 0, -5) # reverse range 25, 20, 15, 10, 5 >>> range(10)</pre>	

Thanks for Listening