**Questions:**

1. **Python Basics:**

**What is Python, and what are some of its key features that make it popular among developers? Provide examples of use cases where Python is particularly effective.**

**Python –** is a dynamically typed, high level; intergraded programming language

**Why python:**

1. It is simple
2. Is the best language for scripting and gaming programming

Example of uses cases:

* **Data Analysis and Visualization**: Libraries like Pandas, NumPy, and Matplotlib make it easy to analyze and visualize data.
* **Web Development**: Frameworks such as Django and Flask simplify web application development.
* **Machine Learning and AI:** Libraries like TensorFlow, Keras, and Scikit-learn are widely used for building and training machine learning models.

1. **Installing Python:**

**Describe the steps to install Python on your operating system (Windows, macOS, or Linux). Include how to verify the installation and set up a virtual environment.**

**Installation of python**

1. Python.org

1. **Choose the mode of your device**

* Navigate to the Downloads section and select Download Python (latest version).
* the go FILE – get the 64bit installer
* Check the PATH when installing python
* Click "Install Now"

1. **To test python installation**

* Go to CMD the type python –version

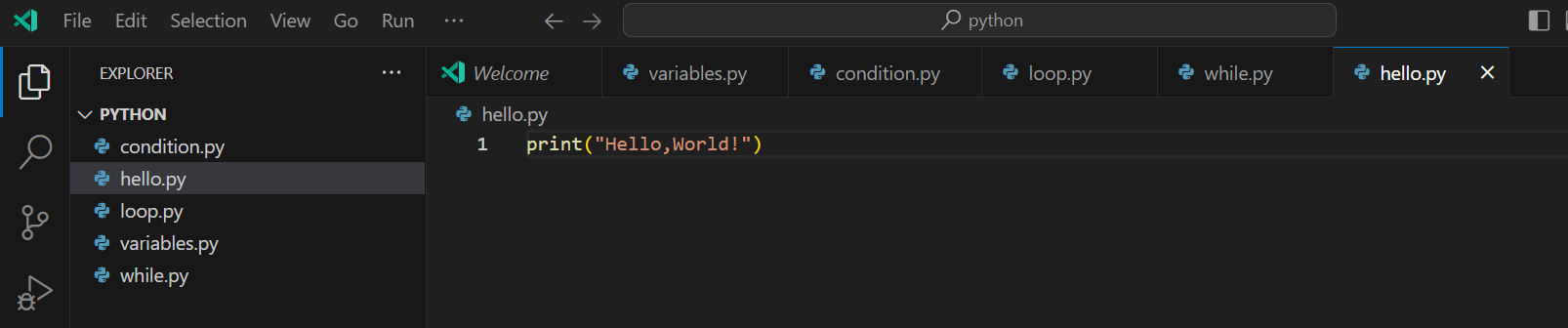
1. **Installation and set up a virtual environment.**

* In CMD Check pip –version
* In Command Prompt, type pip install virtualenv
* Then navigate to project directory { cd my\_project}
* The create the virtual environment { py –m venv myenv}
* Then activate the virtual environment{ myenv/Scripts/activate}
* Then deactivate afterwards {deactivate}

1. **Python Syntax and Semantics:**

**Write a simple Python program that prints "Hello, World!" to the console. Explain the basic syntax elements used in the program.**

**Print (“Hello, World!”)**

****

* **Print ():** This is a built-in Python function used to output text to the console.
* **"Hello, World!”** This is a string literal. In Python, a string is a sequence of characters enclosed in either single quotes (') or double quotes (").
* **The parentheses ()** are used to call functions in Python. They contain the arguments that are passed to the function. In this case, the string **"Hello, World!"** is the argument passed to the **print** function.
* **““**: Double quotation marks are used to denote the beginning and end of a string.

1. **Data Types and Variables:**

**List and describe the basic data types in Python. Write a short script that demonstrates how to create and use variables of different data types.**

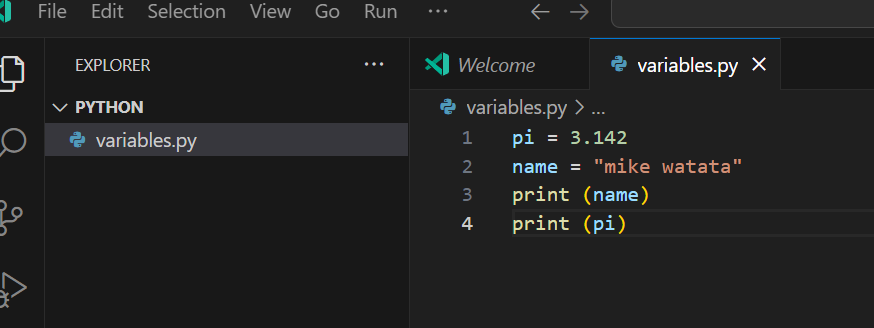
Data type:

1. Int – this refers to a whole number. Example 1, 2,
2. Float- is any number with a decimal point 2.5, 5.6
3. String – is collection of characters example – “Hello world”
4. Boolean- the show the value true or false.

Variable – this means something that can change.

* In programming variable is like a container for data

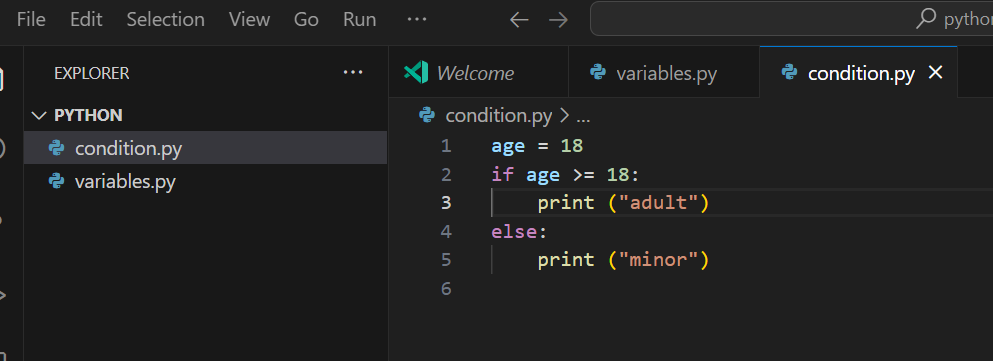
How to create and use variables of different data types.



1. **Control Structures:**

**Explain the use of conditional statements and loops in Python. Provide examples of an if-else statement and a for loop.**

**Conditional statement** – this allow the programmer to execute a block of code base on certain condition {the most common id **if, elif** and **else}**

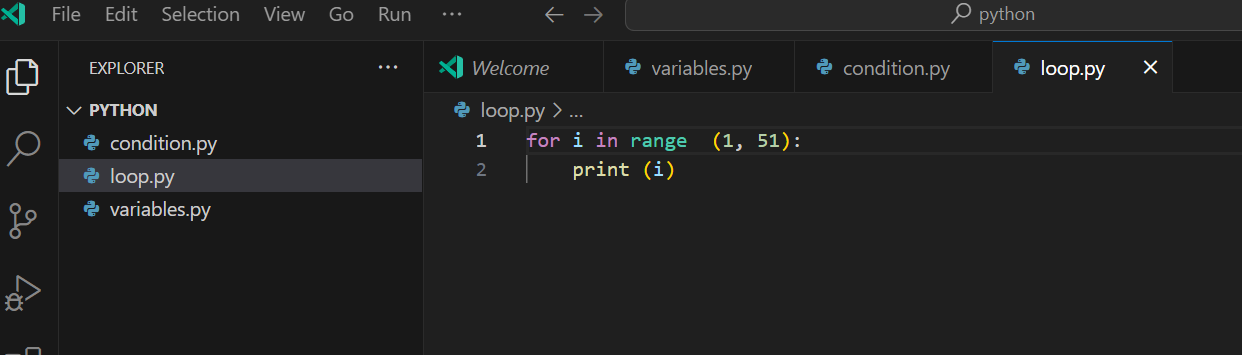
****

**Note:** **if statement** check if the condition {if age => 18} is true the it execute the intended code **while else statement** check **if** is false the it execute the code under the **else**

**Loops –** is used to execute a certain block of code repeatedly till the required condition is met

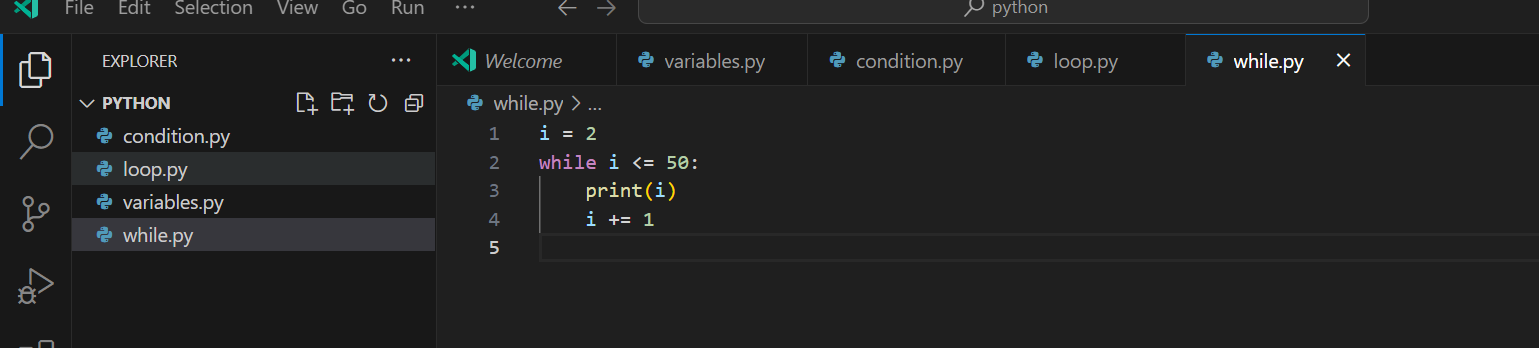
1. Iterative – doing something in repletion { for loop}

For loop: Iterates over a sequence (like a list, tuple, or string) and executes a block of code for each element.



1. Conditional – this is doing something until the condition is met { while loop}

* While loop: Repeats a block of code as long as a condition is true.



1. **Functions in Python:**

**What are functions in Python, and why are they useful? Write a Python function that takes two arguments and returns their sum. Include an example of how to call this function.**

**Functions** – is a reusable chunk of code

* It uses **def** keyword, followed by the function name and **parentheses ().** Arguments, if any, are placed inside the parentheses.

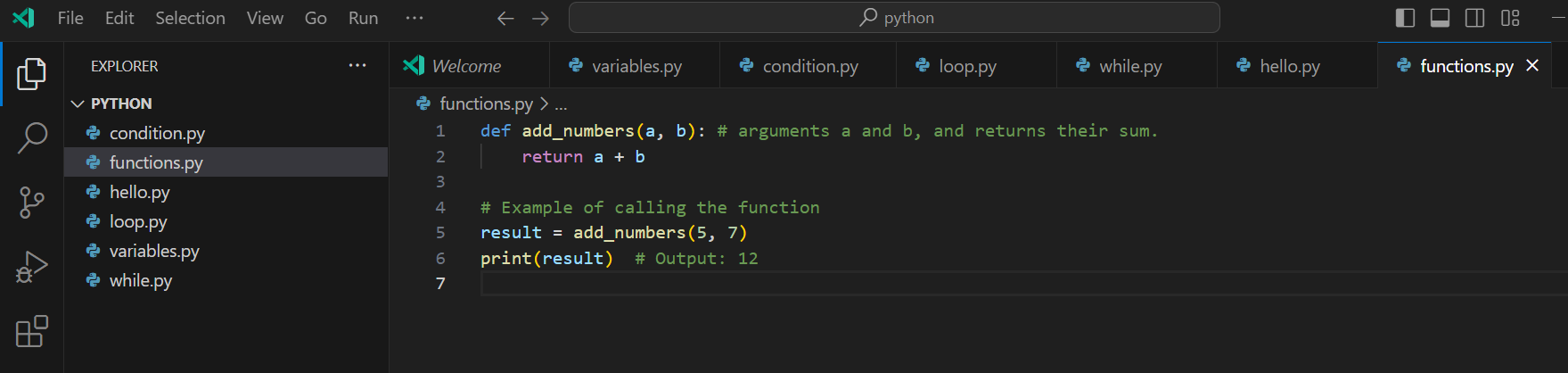
**Why are they useful**

* Organize code into manageable sections.
* Avoid repetition by allowing code reuse.
* Improve code readability and maintainability.

**Type of functions:**

* Pre-defined functions
* User-defined functions

**Example:**



1. **Lists and Dictionaries:**

**Describe the differences between lists and dictionaries in Python. Write a script that creates a list of numbers and a dictionary with some key-value pairs, then demonstrates basic operations on both.**

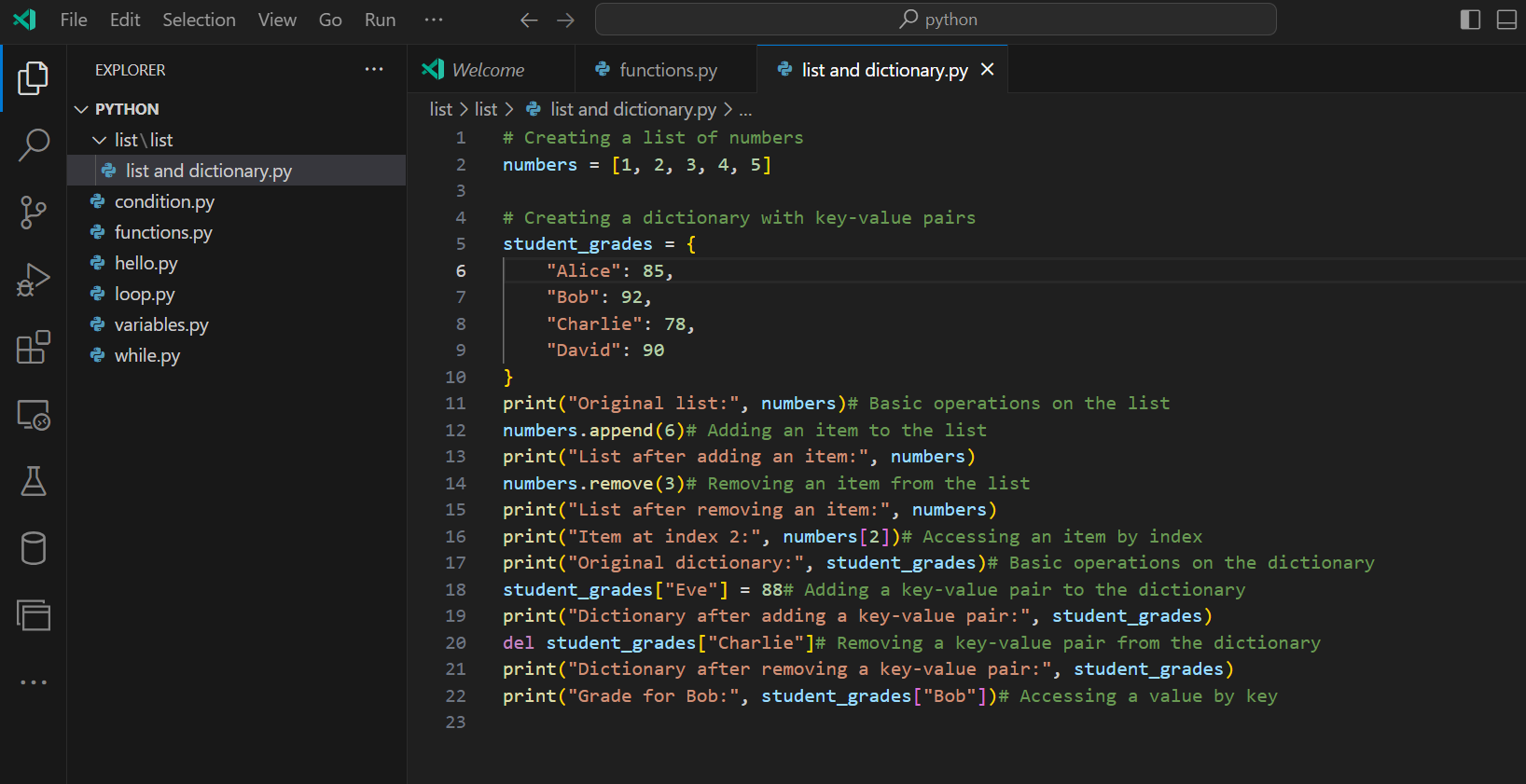
**Differences between Lists and Dictionaries in Python:**

Lists:

* Ordered collections of items.
* Items are accessed by their position (index) in the list.
* Syntax: my\_list = [item1, item2, item3]
* Can contain duplicate items.

Dictionaries:

* Unordered collections of key-value pairs.
* Items are accessed by their keys.
* Syntax: my\_dict = {key1: value1, key2: value2}
* Keys must be unique and immutable (e.g., strings, numbers, tuples).



**Explanation**

**List Operations:**

* **numbers.append(6):** Adds the number 6 to the end of the list.
* **numbers.remove(3):** Removes the first occurrence of the number 3 from the list.
* **numbers[2]:** Accesses the item at index 2 (the third item in the list).

**Dictionary Operations:**

* **student\_grades["Eve"] = 88:** Adds a new key-value pair ("Eve": 88) to the dictionary.
* **del student\_grades["Charlie"]:** Deletes the key-value pair with the key "Charlie" from the dictionary.
* **student\_grades["Bob"]:** Accesses the value associated with the key "Bob" (which is 92).

1. **Exception Handling:**

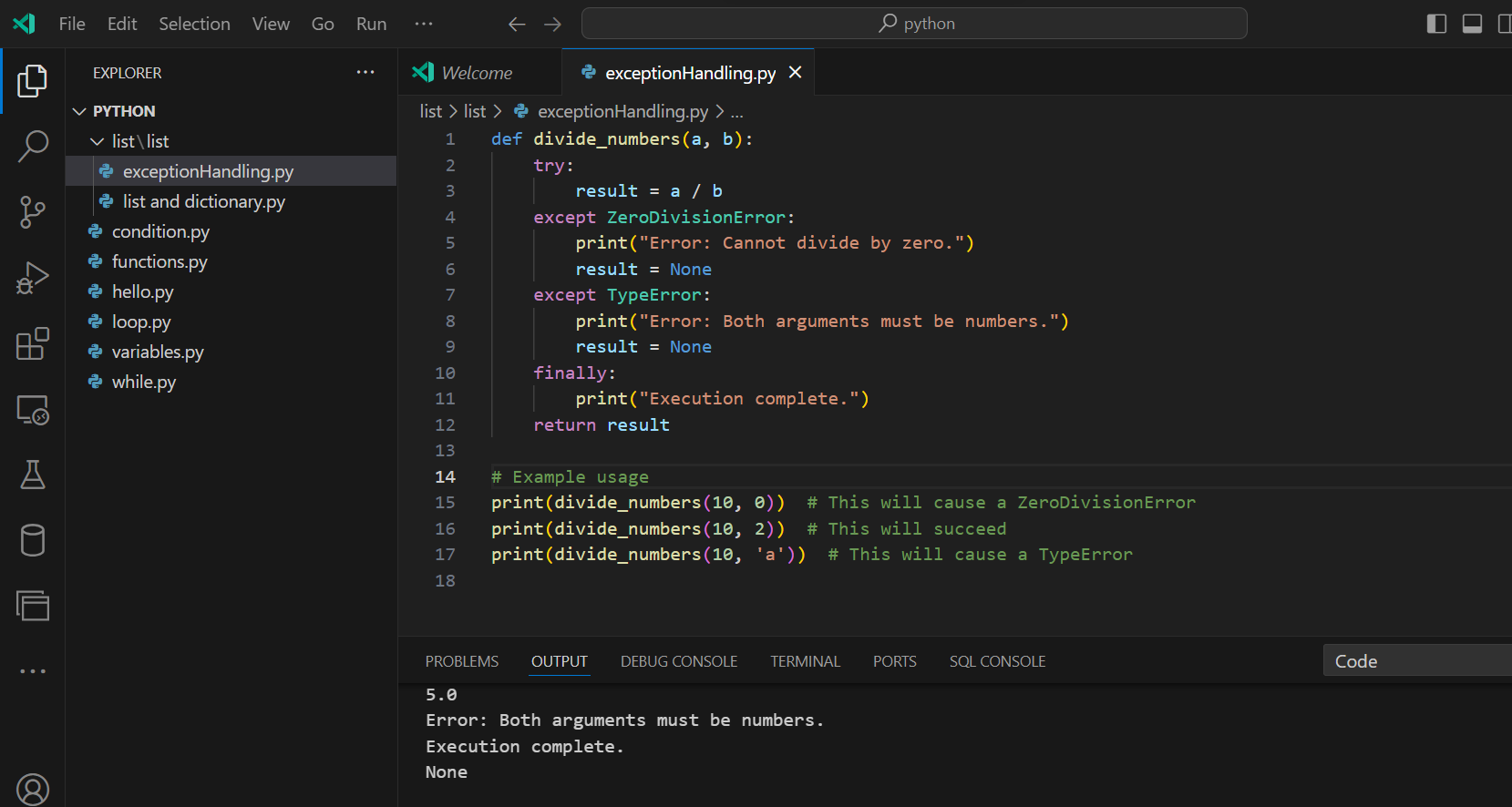
**What is exception handling in Python? Provide an example of how to use try, except, and finally blocks to handle errors in a Python script.**

**Exception handling in Python**

* Is a way to handle errors gracefully without crashing the program
* It allows the programmer to catch and respond to different exceptions that may occur during the execution of a program.

**Key Components of Exception Handling**

* **try block:** Code that might raise an exception is placed inside a try block.
* **except block:** Code to handle the exception is placed inside an except block.
* **finally block:** Code that will run no matter what, even if an exception occurs, is placed inside a finally block.



1. **Modules and Packages:**

**Explain the concepts of modules and packages in Python. How can you import and use a module in your script? Provide an example using the math module.**

**Modules and Packages in Python**

**A module** - is a single file containing Python definitions and statements. Modules allow you to organize your code into separate files for better readability and reusability.

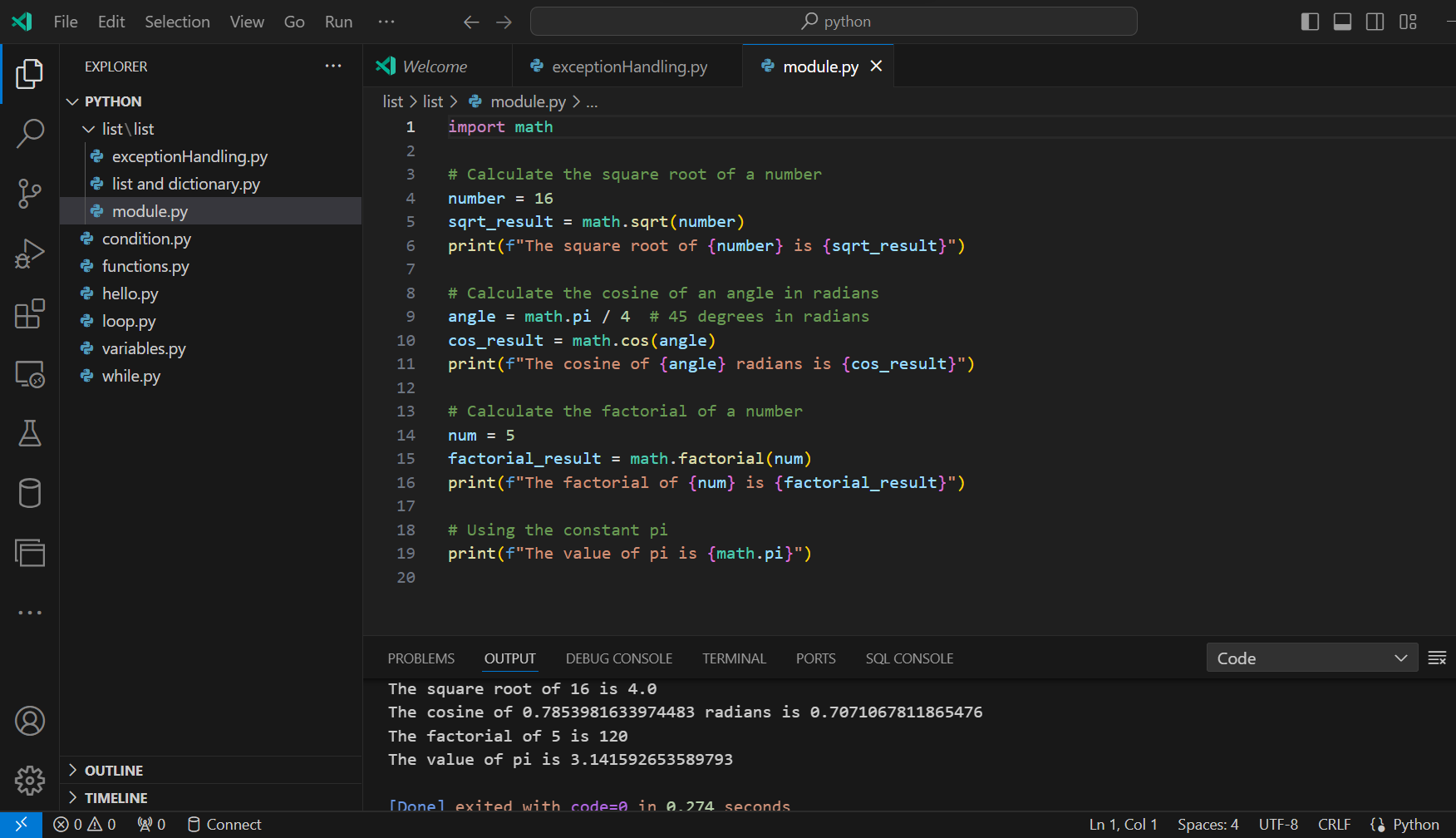
* You can import a module into another Python script to use its functions, classes, and variables.

Packages:

**A package** - is a collection of modules organized in directories that provide a hierarchical structure.

* **Note:** A package must contain a special file named \_\_init\_\_.py, which can be empty or contain initialization code for the package.
* Packages allow for a structured and organized collection of modules, making it easier to manage larger codebases.

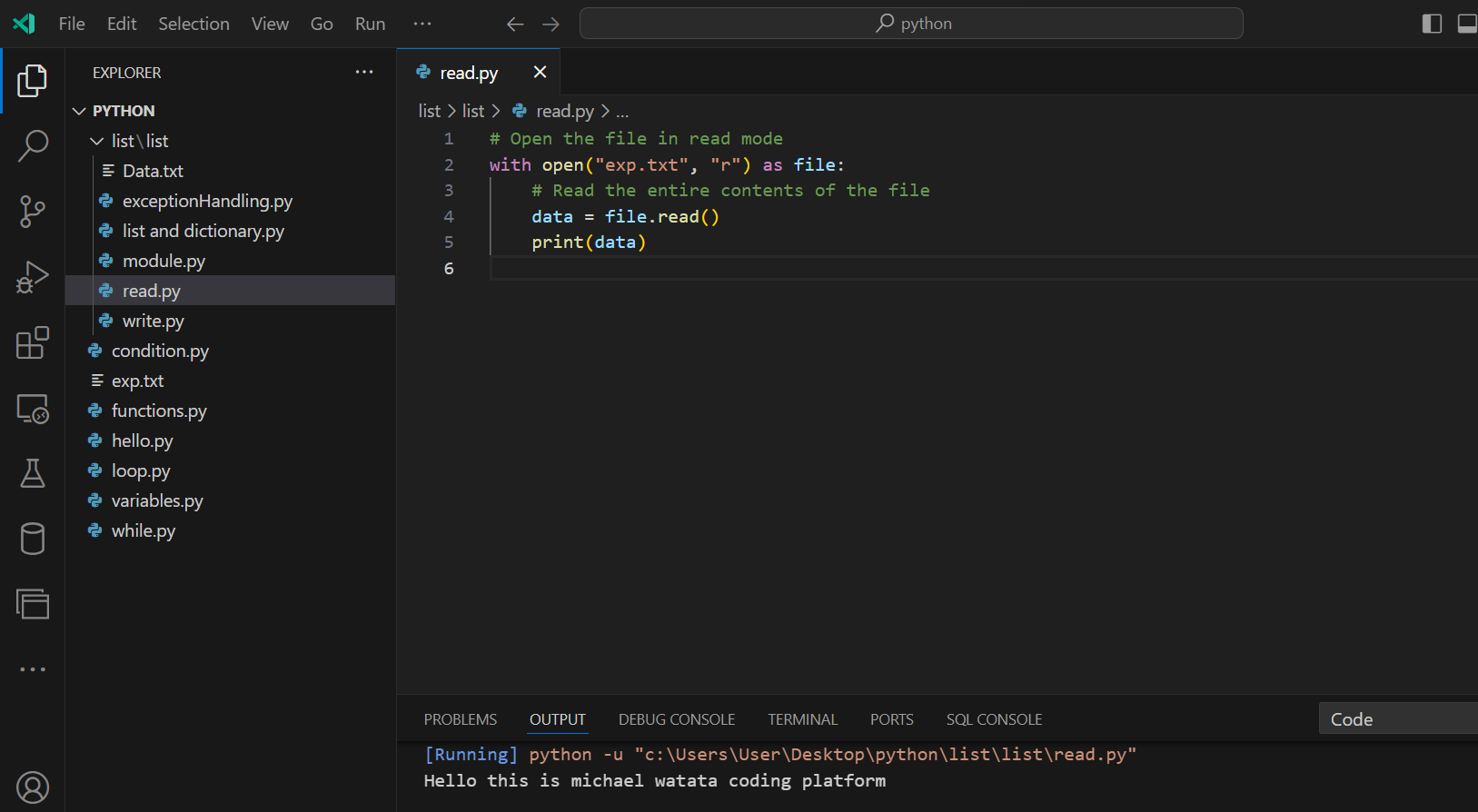
**How to Import and Use a Module in Your Script**



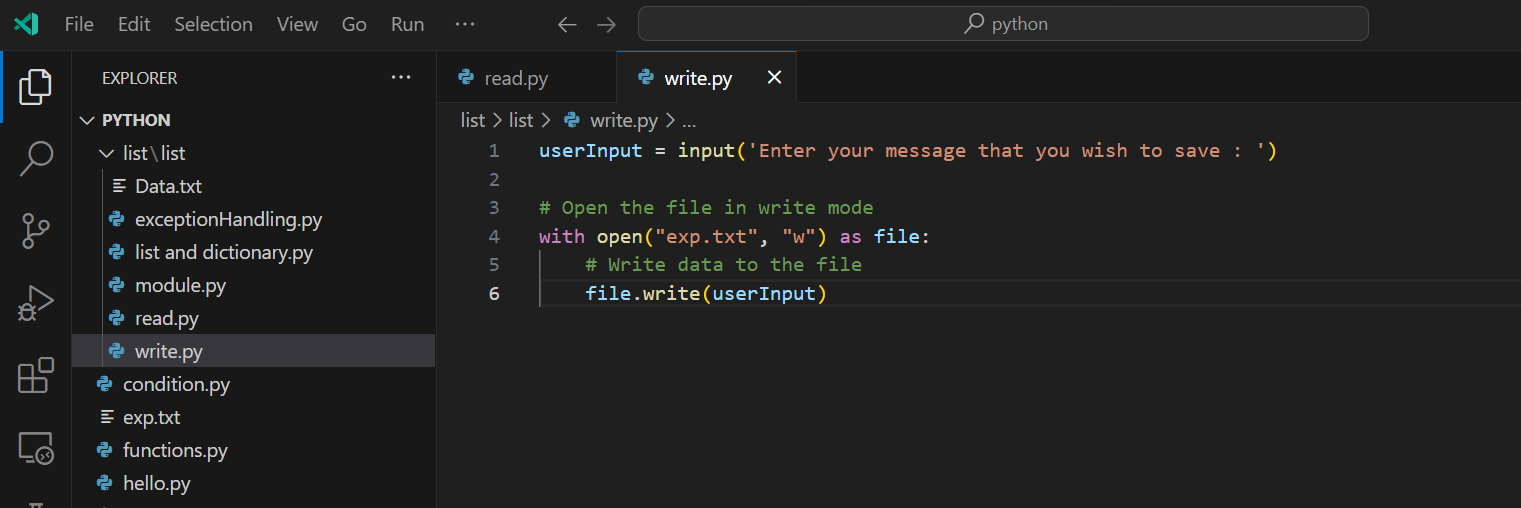
1. **File I/O:**

**How do you read from and write to files in Python? Write a script that reads the content of a file and prints it to the console, and another script that writes a list of strings to a file.**

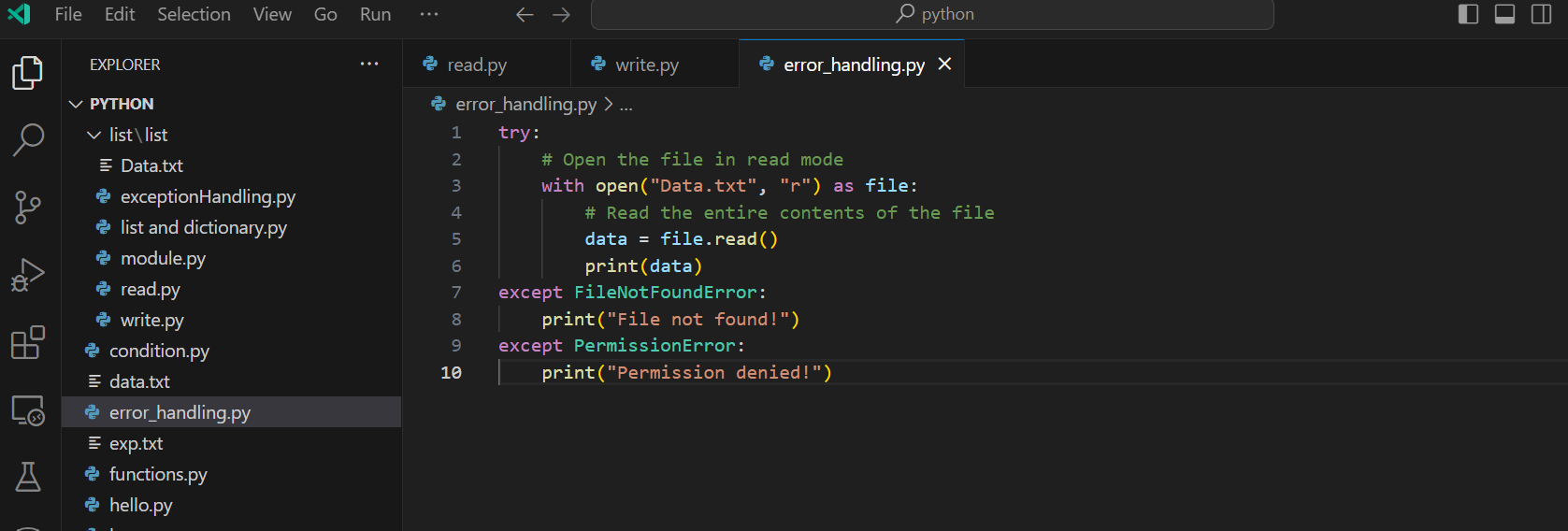
**Reading from a File:**

****

**Writing to a File:**

****

**Error Handling:**

****