- **IDLE:** This is the default code editor that comes with Python, and is a good option for beginners. It has a simple interface and basic features, such as syntax highlighting and auto-indentation.
- **Sublime Text:** This is a popular text editor that supports multiple programming languages, including Python. It has a wide range of features, including syntax highlighting, code snippets, and plugins for customizing the interface.
- **PyCharm:** This is a powerful integrated development environment (IDE) that is designed specifically for Python development. It has a wide range of features, including code completion, debugging, and version control integration.

To install a code editor, simply download the installer from the editor's website and follow the prompts to install it on your computer.

❖ **Writing your first Python program (Hello World!).**

- To write your first Python program, follow these steps.
  - Open your code editor and create a new file.
  - Type the following code:

```
print("Hello, world!")
```


  - Save the file with a .py extension (e.g. helloworld.py).
  - Open a command prompt (Windows) or terminal (macOS/Linux) and navigate to the directory where you saved the file.
  - Type "python helloworld.py" to run the program.

This program will output the text "Hello, world!" to the console.

❖ **Running Python programs from the command line and code editor.**

- To run a Python program from the command line, navigate to the directory where the program is saved and type "python filename.py", where "filename.py" is the name of the Python file. This will execute the program and output any results to the console.
- To run a Python program from a code editor, simply open the file in the editor and click the "run" button or use the keyboard shortcut to run the program. The exact steps may vary depending on the code editor you are using.

❖ **How to use comments in Python.**

- Comments are used to explain the purpose and functionality of the code and are ignored by the Python interpreter. To add a comment in Python, start the line with the "#" character. For example:

```
# This is a comment
print("Hello, world!") # This is also a comment
```

Comments can also be used to temporarily disable a line of code for testing purposes. For example:

```
# print("Hello, world!")
```

In this example, the print statement is commented out and will not be executed, but can be uncommented later to enable the code again.

❖ Variables and Simple Data Types:

- Variables are used to store values in Python.
- Simple data types include strings, integers, floats, and booleans.

Examples:

```
message = "Hello, World!"
age = 25
height = 1.75
is_raining = True
```

❖ Introducing Lists:

- Lists are used to store multiple values in a single variable.
- Lists are denoted by square brackets and each item in the list is separated by a comma.

Examples:

```
colors = ['red', 'green', 'blue']
numbers = [1, 2, 3, 4, 5]
```

```
    # Some code that might raise MyException
    raise MyException('This is a custom exception')
except MyException as e:
    # Handle the MyException exception
    print(e)
```

In this example, the MyException class is defined as a subclass of the built-in `Exception` class. The try block raises an instance of this exception, and the except block handles it by printing the exception message.

- **How to work with JSON data?**
  - To work with JSON data in Python, you can use the built-in `json` module. Here's an example of reading JSON data from a file:

```
import json

with open('data.json', 'r') as file:
    data = json.load(file)
    print(data)
```


And here's an example of writing JSON data to a file:

```
import json

data = {'name': 'John', 'age': 30}

with open('data.json', 'w') as file:
    json.dump(data, file)
```

- **How to handle errors when working with files and exceptions?**