**Python 3 Fundamentals:-**

**Explore Data Types:-**

Numeric data types are used to store numbers. There are three types of numeric data types:

Integers are whole numbers, positive or negative.

Floats are numbers with decimal points.

Complex numbers are numbers that have a real and imaginary part.

String data types are used to store text. Strings are enclosed in single or double quotes.

Sequence data types are used to store a collection of data. There are three types of sequence data types:

Lists are ordered collections of data.

Tuples are ordered collections of data that cannot be changed.

Ranges are a special type of sequence that represents a sequence of numbers.

Mapping data types are used to store a collection of key-value pairs. There is one type of mapping data type:

Dictionaries are unordered collections of key-value pairs.

Boolean data types are used to store true or false values.

Set data types are used to store a collection of unique values.

NoneType is a special data type that represents the absence of a value.

**Conditions:-**

It’s used to control the flow of execution of a program. They are used to check if a certain condition is true or false, and then execute a block of code if the condition is true, or another block of code if the condition is false.

There are three types of conditions in Python:

Comparison operators are used to compare two values. The following are the comparison operators in Python:

Equal to: ==

Not equal to: !=

Less than: <

Less than or equal to: <=

Greater than: >

Greater than or equal to: >=

Logical operators are used to combine multiple conditions. The following are the logical operators in Python:

And: and

Or: or

Not: not

Conditional statements are used to execute a block of code if a certain condition is true, or another block of code if the condition is false. The following are the conditional statements in Python:

If statement

If-else statement

Elif statement

**List:-**

A list in Python is a data structure that allows you to store a collection of items. Lists are ordered, which means that the items in the list are stored in a specific order. Lists are also mutable, which means that you can change the contents of a list after it has been created.

**List Methods:-**

**append() –** Adds an element at the end of the list

**clear() –** Removes all the elements from the list

**copy() –** Returns a copy of the list

**count() –** Returns the number of elements with the specified value

**extend() –** Add the elements of a list (or any iterable), to the end of the current list

**index() –** Returns the index of the first element with the specified value

**insert() –** Adds an element at the specified position

**pop() –** Removes the element at the specified position

**remove() –** Removes the item with the specified value

**reverse() –**  Reverses the order of the list

**sort() -** Sorts the list

**Loops:-**

Loops are used to iterate over a collection of data. There are two main types of loops in Python: for loops and while loops.

**For loops:-**

For loops are used to iterate over a collection of data that has a known length.

The range() function generates a sequence of numbers from 0 to the length of the list minus

**While loops:-**

While loops are used to iterate over a collection of data that does not have a known length.

**Dictionaries:-**

A dictionary in Python is a data structure that allows you to store a collection of key-value pairs. Dictionaries are unordered, which means that the order of the key-value pairs does not matter. Dictionaries are also mutable, which means that you can change the contents of a dictionary after it has been created.

**JSON**

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

JSON is a text format that is based on JavaScript object notation. JSON is a popular data format for exchanging data between different programming languages.

JSON is a text format that is easy to read and write. It is also easy for machines to parse and generate. This makes JSON a popular data format for exchanging data between different programming languages.

To convert a Python dictionary to JSON, you can use the json.dumps() function. The json.dumps() function takes one argument, which is the dictionary that you want to convert to JSON. The json.dumps() function returns a JSON string.

**Pip**

Pip is a package manager for Python. Pip allows you to install and manage Python packages.

**Functions:-**

A function is a block of code that is used to perform a specific task. Functions can be called multiple times, and they can be used to organize code and make it easier to read and understand.

**Types:-**

**Built-in functions** are pre-defined in the Python language and can be used without being defined first. Some examples of built-in functions include print(), len(), and max().

**User-defined functions** are defined by the user and can be used to perform specific tasks. User-defined functions are defined using the def keyword.

**Features:-**

Functions can be used to improve the readability and maintainability of code. By grouping related code into functions, it becomes easier to understand and modify the code. Additionally, functions can be used to reuse code in multiple places. This can help to reduce the amount of code that needs to be written and maintained.

**Class:-**

A class is a blueprint for creating objects. It defines the data and behavior of an object. An object is an instance of a class. It has the data and behavior defined by the class.

Classes and objects are used to organize and structure code. They make it easier to create complex programs by providing a way to group related data and behavior together.

**Features:-**

* Classes are defined using the class keyword.
* The \_\_init\_\_() method is called when an object is created. It is used to initialize the object's attributes.
* Methods are defined using the def keyword.
* Attributes can be accessed using the dot notation.
* Methods can be called using the dot notation.

**Objects:-**

In Python, an object is a data structure that contains data and methods. Objects are the building blocks of Python programs, and they can be used to represent anything from simple numbers to complex data structures.

To create an object, you first need to define a class. A class is a blueprint for an object. It defines the data and methods that will be available to the object.

**Inheritance:-**

Inheritance is a powerful feature in Python that allows you to create new classes that inherit the properties and methods of existing classes. This can save you a lot of time and effort, as you don't have to re-write code that already exists.

**Features:-**

* The inherits keyword is used to create a child class that inherits from a parent class.
* The child class inherits all of the properties and methods of the parent class.
* The child class can override the properties and methods of the parent class.
* Inheritance can be used to make code more concise and reusable.

**Types:-**

**Single inheritance:** A child class inherits from a single parent class. Single inheritance is the most common type of inheritance in Python. It is a simple and straightforward way to inherit from a parent class.

**Multiple inheritance:** A child class inherits from multiple parent classes. Multiple inheritance is a more complex type of inheritance. It allows a child class to inherit from multiple parent classes. This can be useful when a child class needs to inherit from multiple different types of objects.

**Multilevel inheritance:** A child class inherits from a parent class, which in turn inherits from another parent class. Multilevel inheritance is a type of inheritance where a child class inherits from a parent class, which in turn inherits from another parent class. This can be useful when a child class needs to inherit from a more specialized version of an object.

**Hierarchical inheritance:** A child class inherits from multiple parent classes, and the parent classes are arranged in a hierarchy. Hierarchical inheritance is a type of inheritance where a child class inherits from multiple parent classes, and the parent classes are arranged in a hierarchy. This can be useful when a child class needs to inherit from a variety of different types of objects.

**Hybrid inheritance:** A combination of two or more of the above types of inheritance. Hybrid inheritance is a combination of two or more of the above types of inheritance. This can be useful when a child class needs to inherit from a variety of different types of objects in a complex way.