Python

BRIDGE COURSE

TOPICS COVERED

95%

- Python
- Programming Modes in Python
- Variables in Python
- Comments in Python
- Saving a Python Program
- Opening a Saved Python Program
- More Programs
- Getting Started with Python
- Input and Output
- Data Types in Python
- Operators in Python
- Executing a Python Program
- Exiting Python Idle

PYTHON

Python is a high-level object-oriented programming language. It is a general-purpose language with interactive features. Python is an open-source software and was founded by **Guido Van Rossum in 1991**. It derives its name from a comedy program named **Python's Flying Circus**. Python is extensively used in:

1. Web Development
2. GUI Application Development
3. Data Analysis
4. Software Development
5. Scientific Applications

FEATURES OF PYTHON PROGRAMMING

Python is a simple and powerful language. Following are the features of python:

Find on Google

What language is google written in?

- **Easy to understand:** The syntax or grammar of Python is simpler, hence making it easier to learn and understand. Writing programs in Python is easy as compared to other high-level programming languages.
- **Open-source:** Python is an open-source language and can be easily downloaded from the Internet for free.
- **Portable:** Programs written in Python can run on almost every known platform such as Windows, Linux and Macintosh.

- **Libraries and Tools:** Programming in Python is easier because the libraries and supporting tools make even a complicated task easy.
- **Interpreted:** The interpreted language allows for easy debugging as it executes code line by line.
- **Integrated and Extensible language:** We can easily integrate Python with other languages such as C, C++. We can also write and compile a Python code in C or C++.
- **Dynamic Memory Allocation:** When an object is created in Python, memory is dynamically allocated to it. This memory is also retracted when it is no longer being used.
- **Object-Oriented:** Python has an object-oriented approach. This means that the programs are designed using objects and classes that interact with each other.

GETTING STARTED WITH PYTHON

Let us learn how to download and install python in the system.

DOWNLOADING AND INSTALLING PYTHON

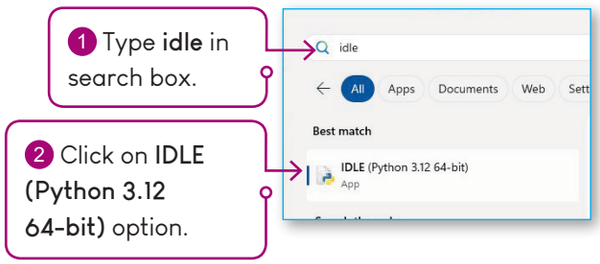
The first step to install Python is to download the latest version of the installation package from the given link: https://www.python.org/downloads/

After downloading the Python setup, run it to install.

Once Python is installed, we will use **IDLE** to write and run Python programs. IDLE stands for **Integrated Development Learning Environment** and it is the most popular graphics-based development environment that allows us to type, edit, run and debug the Python programs.

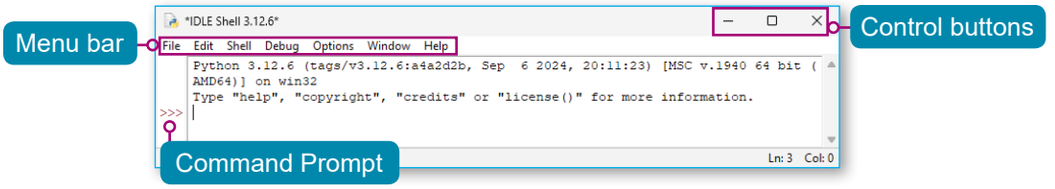
STARTING IDLE PYTHON

To open IDLE, follow the given steps:



COMPONENTS OF THE IDLE SHELL WINDOW

Let's see the main components of the IDLE Shell window:



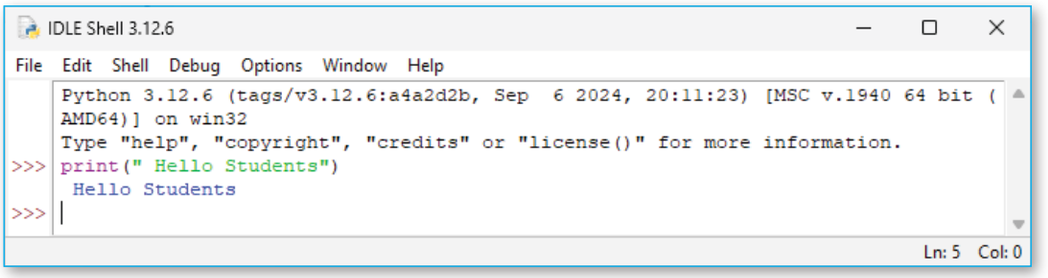
- **Menu bar:** It has various menus such as File, Edit, Shell.
- **Command Prompt:** It allows the user to enter commands directly into Python and get an output instantly by pressing the Enter key.
- **Control buttons:** These buttons are used to resize and close the window.

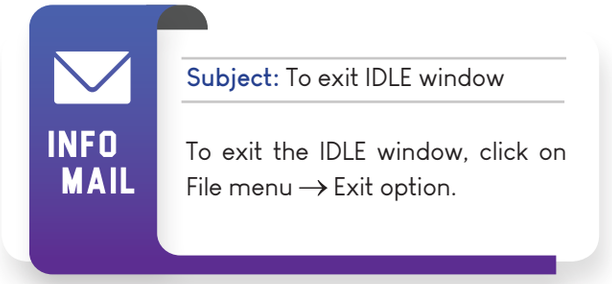
PROGRAMMING MODES IN PYTHON

Python provides two basic programming modes: **Interactive Mode** and **Script Mode.**

INTERACTIVE MODE (IDLE SHELL WINDOW)

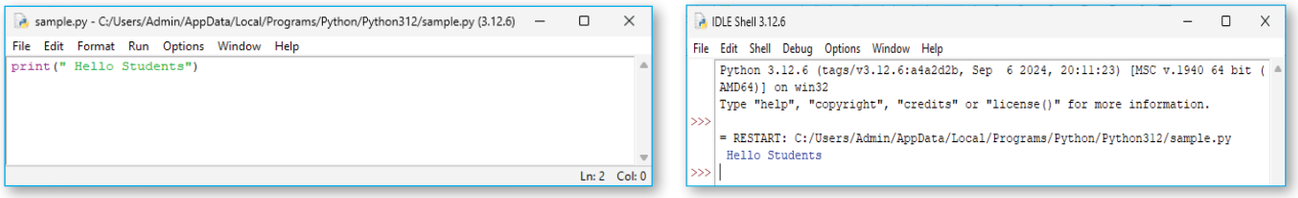
The **IDLE Shell** window is an interactive window where we can type a Python code and view the output in the same window. In the IDLE Shell window, the >>> sign appears. Here, we can type our code and get output by pressing the Enter key.





SCRIPT MODE (EDITOR'S WINDOW)

Python Shell executes commands immediately and doesn't save them for later use. For multi-line code and to save your work for future use, you should use Python script mode. This allows you to write larger programs and run the same code multiple times. Save your code in a file, for example, '**filename.py**', and run it to see all the output after execution. To open Python Editor's Window: click on the File menu → click on the New File option.



The output is displayed in the Python Shell window.

INPUT AND OUTPUT

Python provides two commonly used functions `input()` and `print()` for input and output respectively.

THE input() FUNCTION

We use the `input()` function to take the user's input while a program is being executed. (This function also evaluates the expression whether the user has entered a string, number or list right after receiving input.) The general syntax of the `input()` function is as follows:

```
Variable_name = input(<message to be displayed>)
```

Example:

```
>>> name = input("Enter your name: ")
    Enter your name: Trackpad
```

By default, Python considers the input as string even if you entered a number as input. You cannot perform calculations on string. To perform calculations, you need to convert the string into integer or floating-point value.

To do so, Python provides following functions:

- **int():** It is used to change the input data value into integer. For example,

  ```
  a = int(input("Enter a number "))
  ```

  Now, the entered value is converted into integer value.
- **float():** It is used to change the input value into floating-point value. For example,

  ```
  a = float(input("Enter a number "))
  ```

  Now, the entered value is converted into floating-point value.

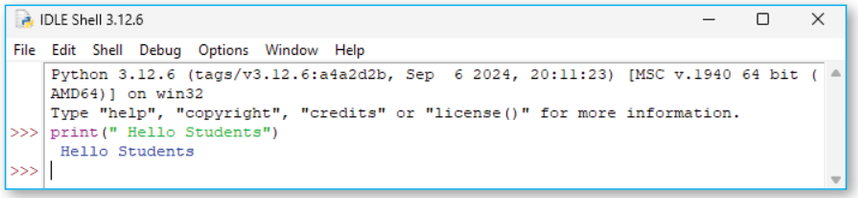
THE print() FUNCTION

We use the print() function to display the result on the screen. The content is written within the parenthesis. To get the output, follow the given steps:

Step 1 Open the Python Shell window.

Step 2 In front of the command prompt, type **print("Hello Students")** and press the Enter key. Do not forget to include the double quotes **(" ")**.

Step 3 The output is displayed in the next line in the same Python Shell window.



VARIABLES IN PYTHON

A **Variable** is the name of the memory location that is used to store data values that can be accessed or changed later. The names given to the variables are known as **identifiers**. In Python, we do not need to specify the type of variable because Python is a dynamic language and it also identifies the variable type automatically.

DECLARING AND INITIALISING A VARIABLE

In Python, variables are declared and initialised at the same time in the following way:

```
a = 10
b = 20
print ("+a=", a)
print ("b=", b)
```

On the output screen, `a = 10` and `b = 20` will be printed. You can also assign the same value to multiple variables at the same time in the following way:

```
a = b = 20
print ("a=", a)
print ("b=", b)
```

On the output screen, `a = 20` and `b = 20` will be printed.

You can also assign multiple values to multiple variables in the same line in the following way:

```
name, age, Class= "Anaya", 12, 'VII'
print ("Name is", name)
print ("Age is", age)
print ("Grade is", Class)
```

On the output screen, `Name is Anaya, Age is 12,` and `Grade is VII` will be printed in three lines.

You must follow the given rules while creating and naming the variables:

- A variable name must start with a letter or underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alphanumeric characters (all the letters of the alphabet and numbers) and underscores (_).
- Variable names are case-sensitive.
- Variable names cannot contain any special character or symbol.

DATA TYPES IN PYTHON

A data type is used to define the type of value a variable can contain. For example, a person's name must be stored as a string value whereas the person's age must be stored as an integer. Commonly used data types in Python are:

- **int:** Positive or negative whole numbers (without any fractions), for example: a = 3
- **float:** Any real number in which a fraction is denoted by a decimal symbol, for example: a = 3.5
- **string:** A collection of one or more characters put in single or double-quotes, for example: a="hello"

COMMENTS IN PYTHON

Comments in Python can be used to explain parts of the code. It can also be used to hide the code as well. Python does not execute comments.

Comments are not a part of the program, but they do enhance the interactivity of the program and make them readable. Python supports two types of comments: Single-line-comments and multiline comments.

Single-line Comment

Python uses the hash (#) symbol for writing single-line comments.

Example:

Program:
```
#printing a string
print("Hello TrackPad")
```

Output:
```
Hello TrackPad
```

Multiple-line Comment

Python does not have a syntax for multiline comments. To add a multiple-line comment, you could insert a **#** for each line. Example:

Program:
```
#print three names
print("Roli")
print("Proloy")
print("Devansh")
```

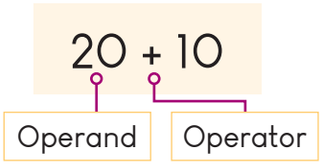
Output:
```
Roli Proloy Devansh
```

However, the text written inside triple quotes (''' or " " ") is also considered a comment. You can use the triple quotes to write multiline comments. For example:

```
'''
This is a multiple-line comment that spans two lines in Python.
'''
```

OPERATORS IN PYTHON

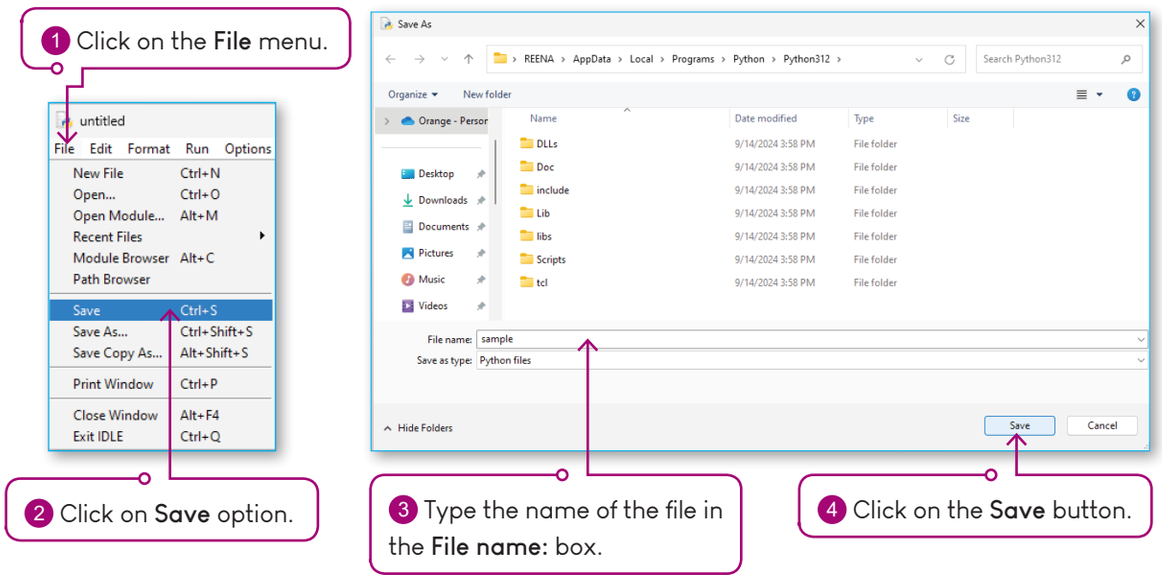
Operators are the special symbols that are used to perform computations. The values which the operators use to get the output are called **operands**.



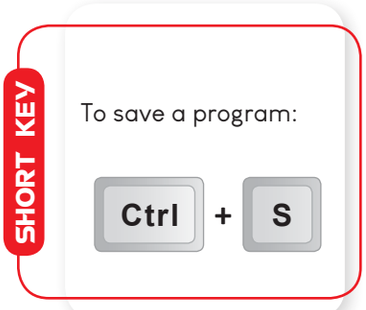
- **(+) Addition operator:** It is an arithmetic operator which is used to perform addition. Arithmetic operators perform arithmetic operations between two operands. Some more arithmetic operators are:
- **(–) Subtraction operator:** It subtracts the right-hand operand from the left-hand operand.
- **(*) Multiplication operator:** It multiplies values on either side of the operator.
- **(/) Division operator:** It divides the left-hand operand by the right-hand operand.
- **(%) Modulus operator:** It divides the left-hand operand by the right-hand operand and returns the remainder.
- **(**) Exponentiation operator:** It performs exponential (power) calculation on operands.
- **(//) Floor or Integer division operator:** It divides and cuts the fractional part from the result.

SAVING A PYTHON PROGRAM

To save a Python program, follow the steps given below:



The file will be saved with the .py extension and ready to be executed as a Python script.

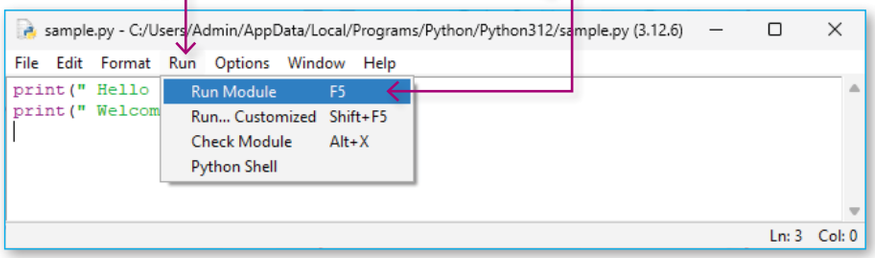


EXECUTING A PYTHON PROGRAM

To run a Python program, follow the steps given below:

1 Click on the **Run** menu.

2 Click on the **Run Module** option.

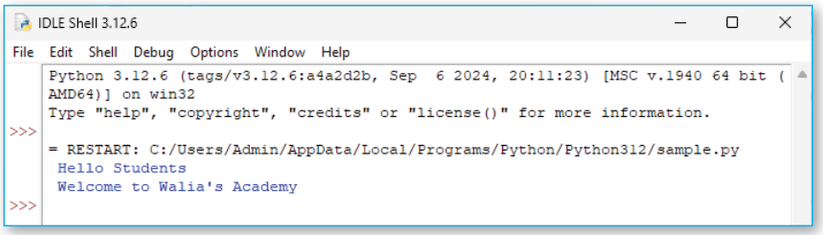


SHORT KEY

To execute a program:

F5

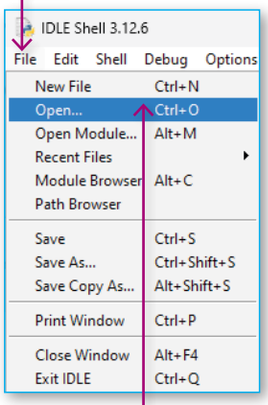
The output will be displayed in the Python Shell window.

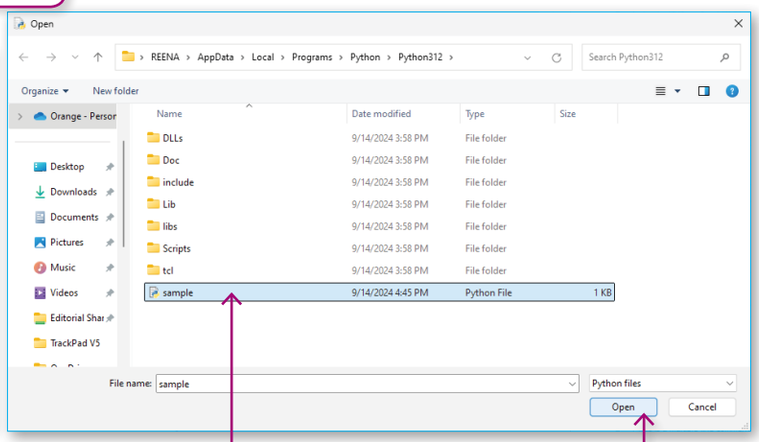


OPENING A SAVED PYTHON PROGRAM

To open a previously saved program, follow the given steps:

1 Click on the **File** menu.





2 Click on **Open** option.

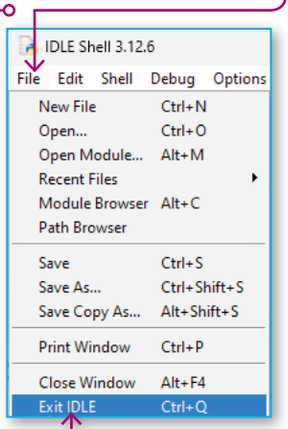
3 Select the program to be opened or type the name in the **File name:** box.

4 Click on the **Open** button.

EXITING PYTHON IDLE

After we are done writing our program, we can exit Python IDLE. To exit from Python IDLE, follow the steps given below:

1 Click on the **File** menu.



2 Click on the **Exit IDLE** option.

INFO MAIL

Subject: Exit Python IDLE

To exit Python IDLE. Type exit() in front of the command prompt and click on OK button.

MORE PROGRAMS

Program 1. Write a program to input your name, school name and also print.

Program1.py

File Edit Format Run Options Window Help

```
# Store Input Data
name = input(" Enter your name: ")
school_name = input(" Enter your school name: ")
# Display the Input Data
print ("Name:",name)
print ("School Name:", school_name)
```

On running the above program, we get the following output:

Output

```
Enter your name: Jaspreet
Enter your school name: DPS
Name: Jaspreet
School Name: DPS
```

Program 2. Write a program to add two numbers.

Program2.py

File Edit Format Run Options Window Help

```
#Program to add two numbers
num1 = int(input(" Enter first number: "))
num2 = int(input(" Enter Second number: "))
ans=num1+num2
print(" The sum is : ", ans)
```

On running the above program, we get the following output:

Output

```
Enter first number: 3
Enter Second number: 2
The sum is : 5
```

Program 3. Write a program to calculate the average marks of three subjects.

Program3.py

File Edit Format Run Options Window Help

```
#Program to calculate the average marks of three subjects
Eng = float(input(" Enter English Marks: "))
Math = float(input(" Enter Math Marks: "))
Hindi = float(input(" Enter Hindi Marks: "))
avg=( Eng+ Math + Hindi)/3
print("Average is :", avg)
```

On running the above program, we get the following output:

Output

```
Enter English Marks: 90
Enter Math Marks: 95
Enter Hindi Marks: 80
Average is : 88.33333333333333
```