

Python is a widely used high-level, interpreted programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

### **What can we do with Python?**

Python is used for:

- **Web Development:** Frameworks like Django, Flask.
- **Data Science and Analysis:** Libraries like Pandas, NumPy, Matplotlib.
- **Machine Learning and AI:** TensorFlow, PyTorch, Scikit-learn.
- **Automation and Scripting:** Automate repetitive tasks.
- **Game Development:** Libraries like Pygame.
- **Web Scraping:** Tools like BeautifulSoup, Scrapy.
- **Desktop Applications:** GUI frameworks like Tkinter, PyQt.
- **Scientific Computing:** SciPy, SymPy.
- **Internet of Things (IoT):** MicroPython, Raspberry Pi.
- **DevOps and Cloud:** Automation scripts and APIs.
- **Cybersecurity:** Penetration testing and ethical hacking tools.

### **Key Features of Python**

- **Easy to Learn and Use:** Python's simple and readable syntax makes it beginner-friendly.
- **Cross-Platform Compatibility:** Python runs seamlessly on Windows, macOS, and Linux.
- **Extensive Libraries:** Includes robust libraries for tasks like web development, data analysis, and machine learning.
- **Dynamic Typing:** Variable types are determined automatically at runtime, simplifying code writing.
- **Versatile:** Supports multiple programming paradigms, including object-oriented, functional, and procedural programming.
- **Open Source:** Python is free to use, distribute, and modify.

## **Why Learn Python?**

Whether you are a beginner or an experienced developer, both have their own benefits.

### **For Beginners:**

- **Easy Syntax:** Python syntax is like plain English, which allows you to focus on logic instead of worrying about complex rules.
- **Built-in Libraries for Beginners:** Python has beginner friendly libraries like random, re, os etc, which can be used while learning fundamentals.
- **Error Friendly:** Python's error messages are easy to understand and debug.
- **Project Oriented Learning:** You can start making simple projects while learning the Python basics.

### **For Experienced:**

- **Easy Career Transition:** If you know any other programming language, moving to Python is super easy.
- **Great for Upskilling:** Moving to Python expands your skill sets and gives opportunity to work in areas like AI, Data Science, web development etc.
- **High Demand of Python in Emerging tech:** Python is widely used in trending domains, like Data Science, Machine Learning, Cloud Computing etc.
- **Bridge Between Roles:** For software developers working with different language, learning Python can help you integrate advanced features like AI in your projects.

## **Hello World in Python**

Hello, World! in python is the first python program which we learn when we start learning any program. It's a simple program that displays the message "Hello, World!" on the screen.

### **Here's the "Hello World" program:**

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

## **Output**

Hello, World!

### **How does this work:**

- `print()` is a built-in function in Python that tells the program to display something on the screen. We need to add the string in parenthesis of `print()` function that we are displaying on the screen.

- “Hello, World!” is a string text that we want to display. Strings are always enclosed in quotation marks.

## Python Comments

Comments in Python are the lines in the code that are ignored by the interpreter during the execution of the program.

- Comments enhance the readability of the code.
- Comment can be used to identify functionality or structure the code-base.
- Comment can help understanding unusual or tricky scenarios handled by the code to prevent accidental removal or changes.
- Comments can be used to prevent executing any specific part of your code, while making changes or testing.

### Example:

```
# I am single line comment
```

```
""" Multi-line comment used
```

```
print("Python Comments") """
```

### Explanation:

- In Python, single line comments starts with hashtag **symbol #**.
- Python ignores the string literals that are not assigned to a variable. So, we can use these string literals as Python Comments.

## Indentation in Python

In Python, indentation is used to define blocks of code. It tells the Python interpreter that a group of statements belongs to a specific block. All statements with the same level of indentation are considered part of the same block. Indentation is achieved using whitespace (spaces or tabs) at the beginning of each line.

### Example:

```
if 10 > 5:
```

```
    print("This is true!")
    print("I am tab indentation")
```

```
print("I have no indentation")
```

#### Explanation:

- The first two print statements are indented by 4 spaces, so they belong to the if block.
- The third print statement is not indented, so it is outside the if block.

### Famous Application Built using Python

- **YouTube:** World's largest video-sharing platform uses Python for features like video streaming and backend services.
- **Instagram:** This popular social media app relies on Python's simplicity for scaling and handling millions of users.
- **Spotify:** Python is used for backend services and machine learning to personalize music recommendations.
- **Dropbox:** The file hosting service uses Python for both its desktop client and server-side operations.
- **Netflix:** Python powers key components of Netflix's recommendation engine and content delivery systems (CDN).
- **Google:** Python is one of the key languages used in Google for web crawling, testing, and data analysis.
- **Uber:** Python helps Uber handle dynamic pricing and route optimization using machine learning.
- **Pinterest:** Python is used to process and store huge amounts of image data efficiently.

### Input and Output in Python

With the `print()` function, we can display output in various formats, while the `input()` function enables interaction with users by gathering input during program execution.

#### Printing Output using `print()` in Python

At its core, printing output in Python is straightforward, thanks to the `print()` function. This function allows us to display text, variables and expressions on the console. Let's begin with the basic usage of the `print()` function:

In this example, "Hello, World!" is a string literal enclosed within double quotes. When executed, this statement will output the text to the console.

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

## Output

Hello, World!

## Printing Variables

We can use the `print()` function to print single and multiple variables. We can print multiple variables by separating them with commas. **Example:**

*# Single variable*

```
s = "Akash"
```

```
print(s)
```

*# Multiple Variables*

```
s = "Adi"
```

```
age = 25
```

```
city = "Haridwar"
```

```
print(s, age, city)
```

## Output

Akash

Adi 25 Haridwar

## Taking input in Python

**Python `input()` function** is used to take user input. By default, it returns the user input in form of a string.

**Syntax:** `input(prompt)`

### How to Take Input in Python

The code prompts the user to input their name, stores it in the variable “name”, and then prints a greeting message addressing the user by their entered name.

```
name = input("Enter your name: ")
print("Hello,", name, "! Welcome!")
```

## **Output**

Enter your name: BSACET

Hello, BSACET! Welcome!

## **How to Change the Type of Input in Python**

By default `input()` function helps in taking user input as string. If any user wants to take input as int or float, we just need to typecast it.

## **How to Print Names in Python**

The code prompts the user to input a string (the color of a rose), assigns it to the variable `color`, and then prints the inputted color.

```
# Taking input as string  
  
color = input("What color is rose?: ")  
  
print(color)
```

## **Output**

What color is rose?: Red

Red

## **How to Print Numbers in Python**

The code prompts the user to input an integer representing the number of roses, converts the input to an integer using typecasting, and then prints the integer value.

```
# Taking input as int  
  
# Typecasting to int  
  
n = int(input("How many roses?: "))  
  
print(n)
```

## **Output**

How many roses?: 8

8

## **How to Print Float/Decimal Number in Python**

The code prompts the user to input the price of each rose as a floating-point number, converts the input to a float using typecasting, and then prints the price.

```
# Taking input as float  
  
# Typecasting to float
```

```
price = float(input("Price of each rose?: "))

print(price)
```

**Output**

Price of each rose?: 50.30

50.3