# **Introduction**

Python was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with focus on code readability and its syntax allows us to express concepts in fewer lines of code.

### **Key Features of Python**

* Python’s simple and readable syntax makes it beginner-friendly.
* Python runs seamlessly on Windows, macOS and Linux.
* Includes libraries for tasks like web development, data analysis and machine learning.
* Variable types are determined automatically at runtime, simplifying code writing.
* Supports multiple programming paradigms, including object-oriented, functional and procedural programming.
* Python is free to use, distribute and modify.

### **Understanding Hello World Program in Python**

**Example:**

**Input**

print("Hello, World!")

**Output**

Hello, World!

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

* print() is a built-in Python function that tells the computer to show something on the screen.
* The message "Hello, World!" is a string, which means it's just text. In Python, strings are always written inside quotes (either single ' or double ").
* Anything after # in a line is a comment. Python ignores comments when running the code, but they help people understand what the code is doing.
* Comments are helpful for explaining code, making notes or skipping lines while testing.

### **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.

### **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.

# **Input and Output in Python**

Understanding input and output operations is fundamental to [Python programming](https://www.geeksforgeeks.org/python-programming-language/). 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.

### **Taking input in Python**

**Python**[**input() function**](https://www.geeksforgeeks.org/python-3-input-function/) is used to take user input. By default, it returns the user input in form of a string.

**Example:**

**Input:**

name = input("Enter your name: ")

print("Hello,", name, "! Welcome!")

**Output:**

Enter your name: GeeksforGeeks  
Hello, GeeksforGeeks ! Welcome!

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.

### **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()](https://www.geeksforgeeks.org/python-output-using-print-function/) 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.

**Example:**

**Input**

print("Hello, World!")  
**Output**

Hello, World!

### **Printing Variables**

We can use the print() function to [print single and multiple variables](https://www.geeksforgeeks.org/g-fact-25-print-single-multiple-variable-python/). We can print multiple variables by separating them with commas.

**Input**

# Single variable

s = "Bob"

print(s)

​

# Multiple Variables

s = "Alice"

age = 25

city = "New York"

print(s, age, city)

**Output**

Bob

Alice 25 New York

### **Take Multiple Input in Python**

We are [taking multiple input](https://www.geeksforgeeks.org/taking-multiple-inputs-from-user-in-python/)from the user in a single line, splitting the values entered by the user into separate variables for each value using the [split() method](https://www.geeksforgeeks.org/python-string-split/). Then, it prints the values with corresponding labels, either two or three, based on the number of inputs provided by the user

**Example:**

**Input**

# taking two inputs at a time

x, y = input("Enter two values: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)

# taking three inputs at a time

x, y, z = input("Enter three values: ").split()

print("Total number of students: ", x)

print("Number of boys is : ", y)

print("Number of girls is : ", z)

**Output**

Enter two values: 15 10  
Number of boys: 15   
Number of girls: 10  
Enter three values: 15 10 5  
Total number of students: 15  
Number of boys is : 10   
Number of girls is : 15

### **Take Conditional Input from user in Python**

In this example, the program prompts the user to enter their age. The input is converted to an integer using the int() function. Then, the program uses conditional statements to check the age range and prints a message based on whether the user is a minor, an adult, or a senior citizen.

**Exmaple:**

**Input**

# Prompting the user for input

age\_input = input("Enter your age: ")

# Converting the input to an integer

age = int(age\_input)

# Checking conditions based on user input

if age < 0:

print("Please enter a valid age.")

elif age < 18:

print("You are a minor.")

elif age >= 18 and age < 65:

print("You are an adult.")

else:

print("You are a senior citizen.")

**Output**

Enter your age: 22  
You are an adult.

## **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](https://www.geeksforgeeks.org/type-casting-in-python/) it.

### **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 Colour.

**Example:**

**Input**

color = input("What color is rose?: ")

print(color)

**Output**

What color is rose?: Red  
Red

### **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.

**Example:**

**Input**

# Taking input as int

# Typecasting to int

n = int(input("How many roses?: "))

print(n)

**Output**

How many roses?: 88

### **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.

**Example**

**Input:**

# Taking input as float

# Typecasting to float

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

print(price)

**Output**

Price of each rose?: 50.3050.3

### **Find DataType of Input in Python**

In the given example, we are printing the type of variable x. We will determine the type of an object in Python.

**Example**

**Input:**

a = "Hello World"

b = 10

c = 11.22

d = ("Geeks", "for", "Geeks")

e = ["Geeks", "for", "Geeks"]

f = {"Geeks": 1, "for":2, "Geeks":3}

​

print(type(a))

print(type(b))

print(type(c))

print(type(d))

print(type(e))

print(type(f))

**Output**

<class 'str'>

<class 'int'>

<class 'float'>

<class 'tuple'>

<class 'list'>

<class 'dict'>

## **Output Formatting**

[Output formatting in Python](https://www.geeksforgeeks.org/python-output-formatting/) with various techniques including the format() method, manipulation of the [sep](https://www.geeksforgeeks.org/python-sep-parameter-print/" \t "_blank) and end parameters, f-strings and the versatile % operator. These methods enable precise control over how data is displayed, enhancing the readability and effectiveness of your Python programs.

### **Example 1:**[**Using Format()**](https://www.geeksforgeeks.org/python-output-formatting/)

**Input:**

amount = 150.75

print("Amount: ${:.2f}".format(amount))

**Output**

Amount: $150.75

### **Example 2:**[**Using sep and end parameter**](https://www.geeksforgeeks.org/difference-between-end-and-sep-in-python/)

**Input**

# end Parameter with '@'

print("Python", end='@')

print("GeeksforGeeks")

# Seprating with Comma

print('G', 'F', 'G', sep='')

# for formatting a date

print('09', '12', '2016', sep='-')

# another example

print('pratik', 'geeksforgeeks', sep='@')

**Output**

Python@GeeksforGeeks

GFG

09-12-2016

pratik@geeksforgeeks

### **Example 3:**[**Using f-string**](https://www.geeksforgeeks.org/formatted-string-literals-f-strings-python/)

**Input**

name = 'Saketh'

age = 21

print(f"Hello, My name is {name} and I'm {age} years old.")

**Output**

Hello, My name is Saketh and I'm 21 years old

### **Example 4:**[**Using % Operator**](https://www.geeksforgeeks.org/what-is-a-modulo-operator-in-python/)

We can use **'%'** operator. % values are replaced with zero or more value of elements. The formatting using % is similar to that of ‘printf’ in the C programming language.

* %d –integer
* %f – float
* %s – string
* %x –hexadecimal
* %o – octal

**Input**

# Taking input from the user

num = int(input("Enter a value: "))

add = num + 5

# Output

print("The sum is %d" %add)

**Output**

Enter a value: 50The sum is 55

# **Python Variables**

In [Python](https://www.geeksforgeeks.org/python-programming-language-tutorial/), variables are used to store data that can be referenced and manipulated during program execution. A variable is essentially a name that is assigned to a value. Unlike many other programming languages, Python variables do not require explicit declaration of type. The type of the variable is inferred based on the value assigned.

Variables act as placeholders for data. They allow us to store and reuse values in our program.

**Example:**

**Input:**

# Variable 'x' stores the integer value 10

x = 5

# Variable 'name' stores the string "Saketh"

name = "Saketh"

print(x)

print(name)

**Output**

5

Saketh

### **Table of Content**

* [Rules for Naming Variables](#_Rules_for_Naming)
* [Assigning Values to Variables](#_Assigning_Values_to)
* [Multiple Assignments](#_Multiple_Assignments)
* [Type Casting a Variable](#_Type_Casting_a)
* [Getting the Type of Variable](#_Getting_the_Type)
* [Scope of a Variable](#_Scope_of_a)
* Object Reference in Python
* Delete a Variable Using del Keyword

### **Rules for Naming Variables**

To use variables effectively, we must follow Python’s naming rules:

* Variable names can only contain letters, digits and underscores (\_).
* A variable name cannot start with a digit.
* Variable names are case-sensitive (myVar and myvar are different).
* Avoid using [Python keywords](https://www.geeksforgeeks.org/python-keywords/) (e.g., if, else, for) as variable names.

**Valid Example:**

age = 21

\_colour = "lilac"

total\_score = 90

**Invalid Example:**

1name = "Error" *# Starts with a digit*

**class** = 10 # '**class**' is a reserved keyword

user-name = "Doe" *# Contains a hyphen*

## **Assigning Values to Variables**

### **Basic Assignment**

Variables in Python are assigned values using the = [operator.](https://www.geeksforgeeks.org/python-operators/)

**Example:**

x = 5

y = 3.14

z = "Hi"

### **Dynamic Typing**

Python variables are dynamically typed, meaning the same variable can hold different types of values during execution.

**Example:**

x = 10

x = "Now a string"

## **Multiple Assignments**

Python allows multiple variables to be assigned values in a single line.

### **Assigning the Same Value**

Python allows assigning the same value to multiple variables in a single line, which can be useful for initializing variables with the same value.

**Example:**

**Input:**

a = b = c = 100

print(a, b, c)

**Output:**

100 100 100

### **Assigning Different Values**

We can assign different values to multiple variables simultaneously, making the code concise and easier to read.

**Example:**

**Input:**

x, y, z = 1, 2.5, "Python"

print(x, y, z)

**Output:**

1 2.5 Python

## **Type Casting a Variable**

[Type casting](https://www.geeksforgeeks.org/type-casting-in-python/) refers to the process of converting the value of one data type into another. Python provides several built-in functions to facilitate casting, including int(), float() and str() among others.

### **Basic Casting Functions**

* [**int()**](https://www.geeksforgeeks.org/python-int-function/)- Converts compatible values to an integer.
* [**float()**](https://www.geeksforgeeks.org/float-in-python/)- Transforms values into floating-point numbers.
* [**str()**](https://www.geeksforgeeks.org/python-str-function/)- Converts any data type into a string.

Example**:**

**Input:**

s = "10" # Initially a string

n = int(s) # Cast string to integer

cnt = 5

f = float(cnt) # Cast integer to float

age = 25

s2 = str(age) # Cast integer to string

print(n)

print(f)

print(s2)

**Output**

10

5.0

25

## **Getting the Type of Variable**

In Python, we can determine the type of a variable using the type() function. This built-in function returns the type of the object passed to it.

**Example Usage of type()**

**Input:**

# Define variables with different data types

n = 42

f = 3.14

s = "Hello, World!"

li = [1, 2, 3]

d = {'key': 'value'}

bool = True

# Get and print the type of each variable

print(type(n))

print(type(f))

print(type(s))

print(type(li))

print(type(d))

print(type(bool))

**Output**

<class 'int'>

<class 'float'>

<class 'str'>

<class 'list'>

<class 'dict'>

<class 'bool'>

## **Scope of a Variable**

There are two methods how we define scope of a variable in python which are [local and global](https://www.geeksforgeeks.org/global-local-variables-python/).

### **Local Variables:**

Variables defined inside a function are local to that function.

**Input:**

def f():

a = "I am local"

print(a)

f()

# print(a) # Error: 'local\_var' is not accessible outside the function.

**Output:**

I am local

### **Global Variables:**

Variables defined outside any function are global and can be accessed inside functions using the [global keyword](https://www.geeksforgeeks.org/global-keyword-in-python/).

**Input:**

a = "I am global"

def f():

global a

a = "Modified globally"

print(a)

f()

print(a)

**Output:**

Modified globally

Modified globally