

Python, Variables ,Data types ,operators and if else

Install Python from <https://www.python.org/downloads/>

What is Python

(Make sure to check “Add Python to PATH” during installation

<https://code.visualstudio.com/download>, Install python extension

Python is a **high-level, Interpreted , general-purpose programming language** known for its **simplicity** and **dynamic typing**.

- 1- **High level :** A **high-level language** is a **programming language designed to be easy for humans to read, write, and understand**, abstracting away the complex details of the computer's hardware. Python converts it to bytecode - a low level platform independent representation.(Mac, Windows , Linux etc)

Example: Python (High-Level) vs Assembly (Low-Level)

High-Level (Python):

```
python
x = 10
y = 20
z = x + y
print(z)
```

Low-Level (Assembly - pseudocode):

```
MOV AX, 1
MOV BX, 20
ADD AX, BX
CALL PRINT_AX
```

- 2- **Interpreted :** An **interpreted language** is a programming language where code is **executed line-by-line by an interpreter at runtime, without being compiled into machine code ahead of time.**

```
print("Hello!")
x = 5 + 3
print(x)
```

1. Reads print("Hello!") → prints Hello!
2. Reads x = 5 + 3 → calculates 8 and assigns it
3. Reads print(x) → prints 8

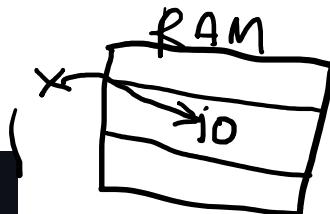
Each line is read, executed, and then discarded, rather than compiled all at once.

Variables

A **variable in Python** is a **name** that refers to a **value stored in memory**.

It's like a label you stick on a box to remember what's inside.

```
x = 10
name = "Ankit"
is_active = True
```



Here:

- x is a variable holding the integer 10
- name holds the string "Ankit"
- is_active holds the boolean True

Variables are stored in RAM and their location can be printed using id(variable name)

```
print(id(name))
```

Variables with the same value will have same address on RAM.

Key Points about Python Variables:

Feature	Description
Case-sensitive	name and Name are different variables
Must start with	A letter or underscore (_), not a number
Can hold any type	Strings, numbers, lists, functions, objects, etc.
Dynamically typed	A variable's type can change at runtime
No need to declare	You don't have to specify the type (e.g., int, string)

Data Types in Python

A data type in Python (or any programming language) tells the computer what kind of value a variable holds and what operations can be performed on it.

Data Type	Type Name	Example	Description
Integer	int	10, -5	Whole numbers
Float	float	3.14, 0.0	Decimal (floating-point) numbers
String	str	"Hello"	Sequence of characters (text)
Boolean	bool	True, False	Logical values (yes/no)

```
x = 10      # int (integer)
name = "Ankit" # str (string)
pi = 3.14    # float (decimal)
is_on = True  # bool (boolean)
```

You can check the type of variable using function type(variable name).

3- Dynamically typed :

A **dynamic type** means that the **type of a variable is determined at runtime**, not when you write the code.

In Python, you don't have to declare the data type of a variable — Python figures it out automatically when the code runs.

Dynamic Typing vs Static Typing

Feature	Dynamic Typing (Python)	Static Typing (C, Java)
Type declaration	Not needed	Required
Type checked at	Runtime	Compile time
Flexible	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> No (strict types)
Example	x = 10	int x = 10;

Print Statement

The print() function in Python is used to display output to the screen.

Use Cases:

Purpose	Example
Show variable value	print(name)
Display message	print("Welcome!")
Debug your code	print("Step 1 done")
Print results	print("Total:", total_area)

Add new line using \n or enter multiline text in 3 single quotes

```
# string method
print("The length of my rectangle is",length,"and breadth is" , breath)

# f string method
print(f"The length of my rectangle is {length} and breadth is {breath}")

# .format method
print("The length of my rectangle is {} and breadth is {}".format(length,breath))
```

--To print more details of print
pip install loguru
from loguru import logger

```
logger.info("The length of my rectangle is {} and breadth is {}".format(length,breath))
```

Operators in Python

An operator in Python is a special symbol or keyword that performs an operation on one or more operands (values or variables).

```
x+y
x*y
x/y
x-y
x%y
```

Full list can be checked here :

<https://docs.python.org/3/library/operator.html>

```
Area = l * b
Print(name + length)
a="2"
b="3"
print(a+b)
```

Explicit type conversion

```
print(int(a)+int(b))
print(str(length)+str(breadth))
```

Implicit conversion

```
a=1.1  
b=2  
print(a+b)  
  
--perimeter , --bodmas rule  
p = 2*(l+b)
```

input() in Python

The `input()` function is used to **take input from the user** during program execution.

```
variable = input("Enter something: ")
```

- The text inside the quotes is shown as a **prompt**.
- The user types something and presses **Enter**.
- The input is stored as a string (always!).

Find area of plot by taking input from user for length and breadth

⚠ Important:

Even if the user enters a number, `input()` returns it as a string. You need to convert it:

```
age = input("Enter your age: ")  
print(age + 5) # ✗ Error: can't add string and int  
 Fix:  
  
age = int(input("Enter your age: "))  
print(age + 5)
```

if else in python

The `if...else` statement is used for decision making in Python. It lets your program choose between different paths based on conditions.

```
if condition:  
    # code if condition is True  
elif another_condition:  
    # code if another condition is True  
else:  
    # code if all conditions are False
```

Comparison operators need to be used in If condition like `>`, `==`, `<`

```
score = int(input("Enter your score: "))  
  
if score >= 90:  
    print("Grade: A")  
elif score >= 75:  
    print("Grade: B")
```

```
elif score >= 60:  
    print("Grade: C")  
else:  
    print("Grade: F")
```

Remember:

- Conditions are checked **top to bottom**.
- Only the **first True** block runs.
- Indentation (spacing) is very important in Python.

Multiple conditions using and:

```
age = int(input("Enter your age: "))  
has_id = input("Do you have an ID? (yes/no): ")  
  
if age >= 18 and has_id.lower() == "yes":  
    print("You are allowed to enter.")  
else:  
    print("Access denied.")
```

Multiple conditions using or:

```
city = input("Enter your city: ")  
  
if city == "Dubai" or city == "Abu Dhabi":  
    print("You are in the UAE.")  
else:  
    print("You are outside the UAE.")
```

NOT reverses the condition -> True to False and False to True

```
city = input("Enter your city: ")  
  
if not (city == "Dubai" or city == "Abu Dhabi"):  
    print("You are outside the UAE.")  
else:  
    print("You are in the UAE.")
```