**PYTHON**

**What is python?**

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility.

It was created by Guido van Rossum, and released in 1991.

Python is used in various fields, including:

Web Development, Data Science & Machine Learning, Artificial Intelligence & Deep Learning, Automation & Scripting, Game Development, Cybersecurity, Internet of Things (IoT), Finance & Trading.

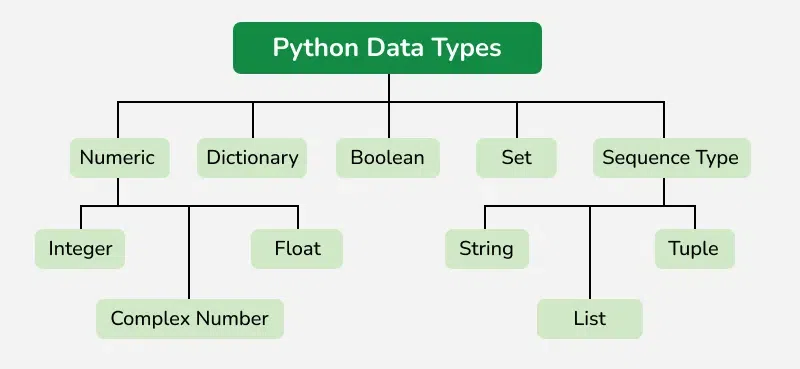
**Why python?**

Beginner-Friendly, Versatility, High Demand, Large Community.

**Python Data types:**

Python data type is the classification or categorization of the data items. It represents the kind of value that tells what operations can be performed on a particular data.

* Numerical -int, float, complex
* Sequence type -string, list, tuple
* Mapping type -dict
* Boolean -bool
* Set type -set, frozen set
* Binary type -bytes, byte array, memory view



Data types define the kind of values that a variable can hold. Python provides several built-in data types:

**1. Numeric Types**

* int (Integer): Stores whole numbers (both positive and negative) without decimal points. (e.g., 10, -5)
* float (Floating Point): Decimal numbers (e.g., 3.14, -0.001)
* complex (Complex Numbers): Numbers with real and imaginary parts (e.g., 3 + 4j)

**2. Sequence Types**

* str (String): Text data (e.g., "Hello", 'Python')
* list (List): Ordered, mutable collection (e.g., [1, 2, 3])
* tuple (Tuple): Ordered, immutable collection (e.g., (1, 2, 3))

**3. Set Types**

* set (Set): Unordered, unique collection (e.g., {1, 2, 3})
* frozenset (Immutable Set): Similar to set but immutable
* animals = frozen set (["cat", "dog", "lion"])
* print ("cat" in animals)
* print ("elephant" in animals)

**4. Mapping Type**

* dict (Dictionary): Key-value pairs (e.g., {"name": "Alice", "age": 25})

**5. Boolean Type**

* bool: Represents True or False
* x = bool (1)
* print(x)
* y = bool ()
* print(y)

true or false

**6. Binary Types**

* bytes: Immutable sequence of bytes
* bytearray: Mutable sequence of bytes
* memoryview:
* Memory view object for handling binary data

**Variables:**

In Python, variables are used to store data values. A variable is like a container that holds information that can be changed later in the program.

* Dynamic Typing – You don’t need to specify the type of a variable; Python automatically determines it.
* Can Change Type – A variable can hold different types of values at different times.
* Case-Sensitive – age and Age are different variables.
* Assignment Using = – The equal sign (=) is used to assign values.

X = 5

*# Variable 'x' stores the integer value 10*

Name = Samantha

*# Variable 'name' stores the string "Samantha"*

Print (x)

Print(name)

Out put

5

Samantha

**Python operators**

In general, Operators are the symbols used to perform a specific operation on different values and variables.

### Different Types of Operators in Python:

1. Arithmetic Operators
2. Comparison Operators
3. Assignment Operators
4. Logical Operators
5. Bitwise Operators
6. Membership Operators
7. Identity Operators

1.Arithmetic Operators:

Used to perform operations on both variables & values types of operations.

Python Arithmetic operators are used to perform basic mathematical operations like addition, subtraction, multiplication, and division.

**Addition: sum**

* 1. Adding two numbers

a = 5

b = 3

sum = a + b

print ("Sum:", sum)

# Output: Sum: 8

* 1. Adding numbers in list

Numbers = [1,2,3,4,5,]

Sum\_ of\_ numbers = sum(numbers)

Print (“sum of list:”, sum\_ of\_- numbers)

Output: 15

* 1. Adding a function

Def add (a, b):

Return a + b

Result = add (10, 20)

Print (“sum:”, result)

Output: 30

* 1. Adding a floating point

X = 2. 5

Y= 3. 7

Sum result = x + y

Print (sum result)

Output: 7. 12

**Subtraction: difference**

1.

A = 10

B = 5

Result = a – b

Print(results)

Output: 5

2. Negative numbers

X = -10

Y = 7

Print (x – y)

Output: -17

3.Subtraction with Floats

python

Copyedit

a = 5.5

b = 2.2

print (a - b)

# Output: 3.3

**Multiplication**

a = 5

b = 3

product = a \* b

print(product)

# Output: 15

**Multiplication with User Input**

python

CopyEdit

a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

product = a \* b

print("Product:", product)

**Multiplication in a Function**

python

CopyEdit

def multiply(a, b):

return a \* b

result = multiply(4, 6)

print(result) # Output: 24

**Multiplication in a Loop**

* 1. A = 10

B = 20

Product = a \* b

Print(product)

#output: 200

* 1. Multiplication in a Function

def multiply (a, b):

return a \* b

result = multiply (4, 6)

print(result)

# Output: 24

3. Multiplication in a Loop

numbers = [2, 3, 4]

product = 1

for num in numbers:

product \*= Num

print(product) #

Output: 24

**Division**

* 1. a = 10

b = 3

result = a // b

print(result)

# Output: 3

* 1. a = 10

b = 2

result = a / b

print(result)

# Output: 5.0

**Comparison operators (returns true or false)**

x = 5

y = 10

print (x == y) # Equal -> False

print (x! = y) # Not equal -> True

print (x > y) # Greater than -> False

print (x < y) # Less than -> True

print (x >= y) # Greater than or equal -> False

print (x <= y) # Less than or equal -> True

**Logical Operators (Used with Boolean values)**

a = True

b = False

print(a and b) # AND -> False

print(a or b) # OR -> True

print(not a) # NOT -> False

**Assignment Operators (Used to assign values)**

x = 5

x += 3 # x = x + 3 -> 8

x -= 2 # x = x - 2 -> 6

x \*= 2 # x = x \* 2 -> 12

x /= 4 # x = x / 4 -> 3.0

x %= 2 # x = x % 2 -> 1.0

x \*\*= 3 # x = x \*\* 3 -> 1.0

**Conditional statement**

if else statement

If Elif else statement

Nested if statements

1.if Statement

If statement is the simplest form of a conditional statement.

Executes a block of code if the condition is True.

python

Copyedit

age = 18

if age >= 18:

print ("You are eligible to vote.") # Output: You are eligible to vote.

2. if-else Statement

Executes one block of code if the condition is True and another if it's False.

python

Copyedit

age = 16

if age >= 18:

print ("You are eligible to vote.")

else:

print ("You are not eligible to vote.") # Output: You are not eligible to vote.

3. if-Elif-else Statement

Checks multiple conditions and executes the first True condition.

python

Copyedit

marks = 85

if marks >= 90:

print ("Grade: A+")

Elif marks >= 80:

print ("Grade: A”) #

Output: Grade: A

Elif marks >= 70:

print ("Grade: B")

else:

print ("Grade: C")

4. Nested if Statements

An if statement inside another if statement.

python

Copyedit

Num = 10

if Num > 0:

print ("Positive number")

if Num % 2 == 0:

print ("Even number”) # Output: Positive number, even number