Software Programming

Software programming is the process of writing instructions that a computer can understand and execute. It involves using programming language to create software, applications, or systems. The goal is to solve problems or perform tasks efficiently through automation.

Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and flexibility. It allows developers to write fewer lines of code compared to many language, making it beginner-friendly and powerful for advanced projects. Supports multiple programming style such as object-oriented, functional and procedural programming.

History of python

- Python was created by Guido Vann Rossum in 1991 at Centrum Wiskunde & Informatica in the Netherlands.
- It was designed as a successor to the ABC language, with an emphasis on code readability and simplicity
- Name python comes from British comedy series 'Monty Python's flying circus'. not from the snake.
- Python 2.0 was released in 2000
- Python 3.0 was released in 2008
- Today, Python is one of most popular programming language widely used in education, research and industries in the world

Features of Python

- simple and easy to learn
- Interpreted language
- Cross-platform
- Free and open source
- large standard library
- Extensible and Embeddable
- Portable

Applications of Python

- web development
- Data Science and Data Analysis
- Machine learning & Artificial Intelligence
- Automation
- Game development
- Desktop application
- Network application

Disadvantages

- Slower Execution speed
- High memory usage
- Not ideal for mobile development
- weak in Database Access
- Runtime errors
- Not best for high-performance compute

Comments

Comments are line in the code that are ignored by the Python Interprete
They are used to explain code, make it easier to read and help other programmers understand it comments do not effect the Program's output

1. Single-line comment

Starts with # symbol

```
ex : # This is a single-line comment
```

2. Multi-line Comment

Written using triple quotes ''' or """

```
Ex : '''
        This is a
          multi line comment
     '''
```

Keywords

Keywords are reserved in python that have special meaning
They are used to define syntax and structure of the python languag
keywords cannot be used as variable names, function names, or identifi
Python keyword are case-sensitive
Ex: if, else, while, for, break, continue, True, False, class, def, return etc

```
Ex code: if True :
              Print (" This is a keyword example")
```

Variables

A variable is a name used to store a value in memory.
It acts like a container for data that can be changed during program execution.

Rules for variable naming

→ Must start with a letter or underscore (_)

→ cannot start with a number

→ can only contains letters, numbers and underscores.

→ cannot be a keyword

→ Variables names are case-sensitive

ex: x = 10 # integer ~~value~~ variable

name = "John" # String variable

Pi = 3.14 # float variable

_is_valid = True # boolean variable

Datatypes

Datatypes define the kind of value a variable can hold. Python has built in data types, and it ~~[illegible]~~ decides the type automatically when a value is assigned

*. Numeric types

int → integer nos

float → Decimal number

Complex → Complex number

* Sequence type

str → String

list → Ordered, changeable collection

~~typ~~ tuple → ordered, unchangeable collection

* Set types

set → unordered, unique elements

frozenset → immutable set

* Mapping type

dict → key-value pairs

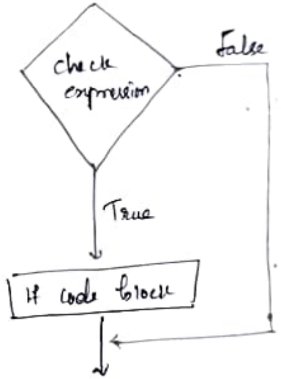
* Boolean type

bool → Represents True or False

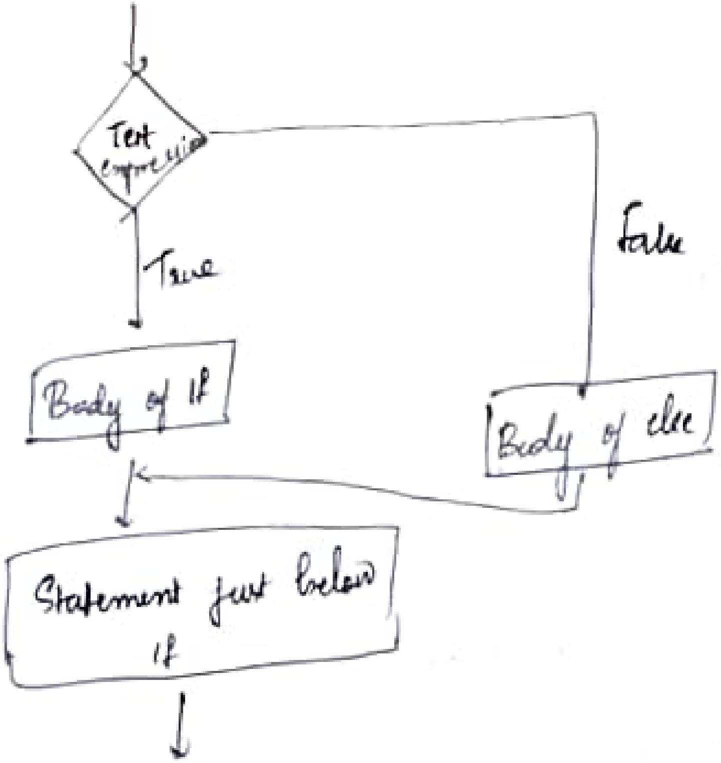
* None type

None → Represents to null values

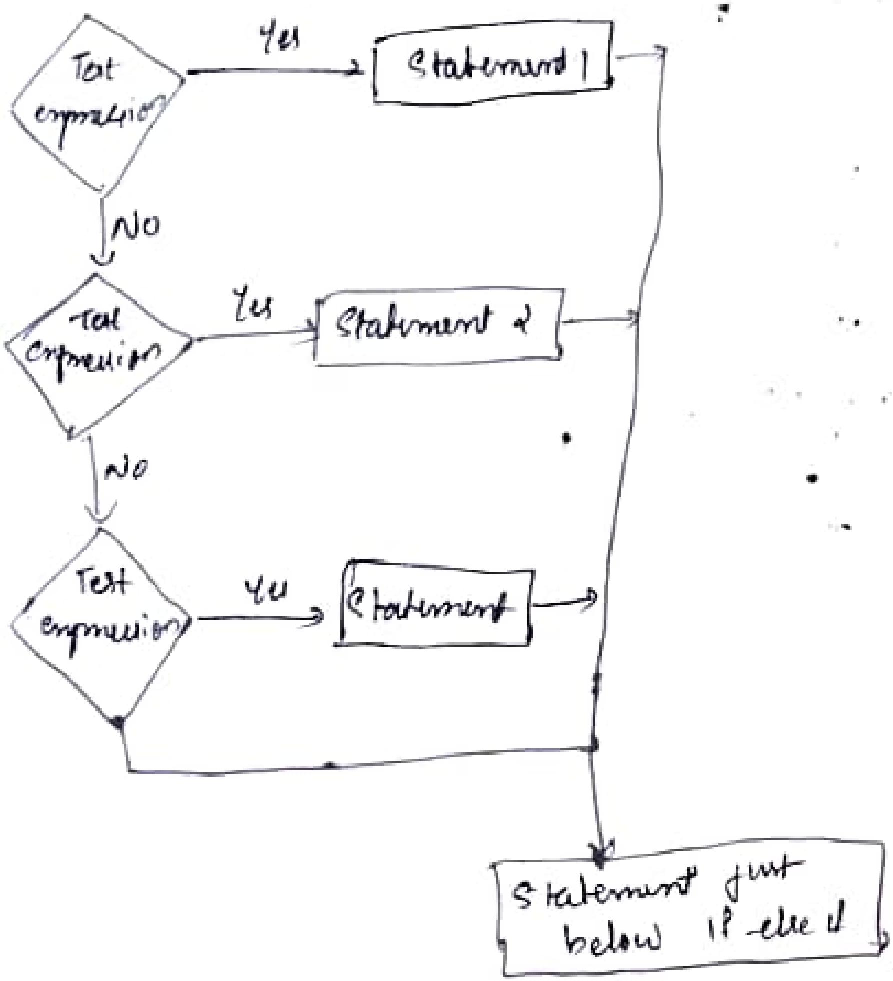
If conditional statements flow control



if - else flow chart



if - else ladder (Elif statement)



Input functions

- The Input() function is used to take user input from the keyboard.
- It always returns the input as a string, even if you type a number.

eg:

```
name = input ("Enter your name: ")
Print ("Hello", name)
```

Print() functions

It is used to display output on the screen

Syntax:

~~Print(object(s), sep= ' ', end= '\n')~~

```
# Simple print
Print ("welcome to python")
# printing variables
x = 10
Print ("value of x is ", x)
# f-string (modern & recommended way)
name = 'Alu"
Print = (f"Hello, {name} ,")
# using format()
marks = 95
Print ("you scored {} marks ".format (marks))
```

conditional statement

Execute code only if the condition is true

If syntax

```
if condition
    statement
```

4

```
eg:- age = int (input ("Enter the age :"))
     if (age >= 18)
         Print ("you are eligible to vote")
```

If else ~~sy~~

Syntax:

```
if condition
    # code of true
else
    # code of false
eg:- num = int (input ("Enter a number"))
  if (num % 2 == 0):
    Print ("even no")
  else:
    Print ("odd no")
```

elif syntax

```
if condition 1:
    Statement cond1
elif cond 2:
    statement cond 2
elif cond 3:
    statement cond 3
else:
    code if all conditions are false

eg:- marks = int (input ("Enter your marks :"))
  if (marks >= 90):
    Print ("grade A")
  elif (marks >= 60)
    Print ("grade B")
  elif (marks >= 50):
    Print ("grade C")
  else:
    Print ("grade fail")
```

Nested if else syntax

```
if cond 1
  if cond 2 :
    # code if both condition true
  else
    # code of cond1 true but cond 2 false
```

```
else :
    # code of cond 1 false

eg: num= int (input ("Enter a no:"))
    if (num>=0):
        if (num ==0):
            Print ("no is zero")
        else :
            Print ("no is +ve")
    else :
        Print ("no is -ve")
```