python

High level, interpreted, oops, dynamically typed, indented programming language

Features -

1. simple syntax | easy to learn
2. indent
3. Open source
4. libraries
5. Support all kind of development
6. Less memory
7. platform independent

Father of python
Guido van Rossum - Python 1 - 1987 - 1990
python2 - 2000 - macbook
python3 - 2008

Applications of python
1. web development
2. Mobile app
3. Machine Learning / deep-Le
4. AI
5. Cyber Security
6. Graphics
7. IOT
8. automation
9. desktop app development

Comments

There are part of code but they won't consider in execution type.

Two types
* single line comment
* Multi-line comment

① Single line comments are represented by
# my program

② Multi - line comment
''' ''' or " " " or " " "

Advan
* if some error pops up we can have idea to solve the logic in the comments

Keywords
They are reserved words used for particular task and they cannot be used as identifier (variables, fun name.. etc)
Ex: True, False, for, if, else, while, break, try, ~~write~~, Except, finally .. etc

Variables -
it is place ~~where we~~ to store our value

Ex : a = 5 b = 10
s = a + b
M = s x 10
D = m / s

Ex a = 5
a is variable 5 is value

Rules
* valid variable declaration
a = 5 , A = 7 , num = 11 , Num1 = 89 Num2 = 11
empid = 123 , emp.name = "priyak" or 'priya'

To assign in single line {
a = 4 ; b = 6 ; c = 7
a, b, c = 4, 6, 7

* invalid variable declaration.
variable not allow to use the number.
1n = 35
stuff name @ = "arun" — not allow
stu id = 345
space not allowed.

Data types
* its pre-defined component and specify the data category.

Type — ① int, float, complex — 4+5j 5j
② str
③ Bool
single value data type

special datatypes (data structure)
① List
② Set
③ tuple
④ dict

a = 6
type (a)
O/P = int

input / output function
to any taking — taking from user
anything giving to the user is called O/P
input function = input()
Output function = Print()

Date / /
Page
SPLASH

Ex

```
print ("hello gm")
 → hello ~~wor~~ gm

a = int(input("Enter a number"))
Enter a number 7

type (a)
 int

b = int(input())
 2

type (b)
 int

a+b
 "9"
```

Q) Write a program to read employee id, employee name, employee phone num and print the details

Q) take 2 float number from user

```
a = float (~~float~~ input ( " Enter 1 number: "))
b = float (~~float~~ input ( " Enter 2 number: "))
print (a)
print (b)        or
print (a+b)
print (a, "+", y, "=", x+y)
print ("sum of:", a, "+", y, "=",
      x+y)
```

Variable → without Codes
~~Number~~ → ~~sa~~ Manually value — should be codes — separated by [,]

Date / /
Page
SPLASH

dot . Format method.

print ("sum of {} + {} = {} ". Format
(x, y, x+y))

f string method.
print (f "sum of {x} + {y} = {x+y}")

Operator

① identity Operator
* is * is not

Ex : a = 9 a = 9
b = 9 b = 8

a is b a is not b

o/p True o/p True

② membership Operator
in not in

Ex: i am a member of your family - False in
i am not a member of your family - True not

You are a member of your family - True in
You are not a member of your family - False
not in.

pet = ['dog', 'cat', 'cow', 'rabbit']
'cat' in pet → True
'~~loin~~ loon' in pet → False
'cat' not in pet — False
'loin' not in pet → True.

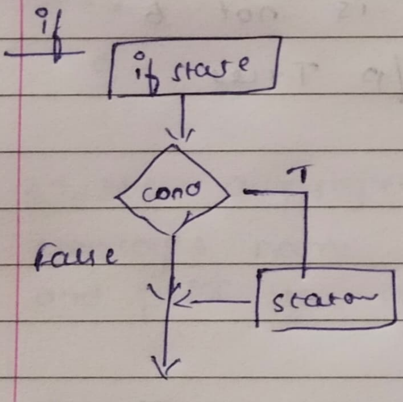
Conditional Statement

-: it is allow us to make decision in code, They check condition (Expression that results in True or False) and Execute different blocks of code accordingly

Types of Conditional Statements

① if statement - Execute a block only if the condition is true.
② if -- else statement - provides two paths, One if condition is True, another if False
③ if -- elsif -- else ladder :- multiple condition checked one by one.
④ Nested if - using One if inside another.

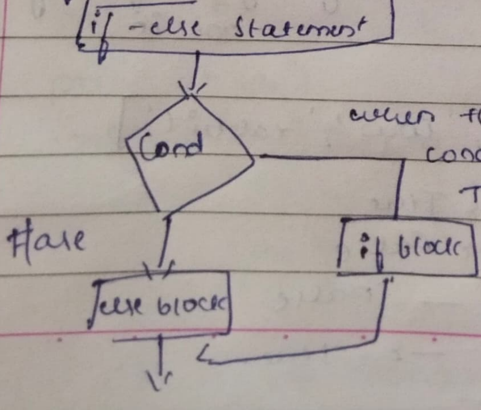
① if



Ex:
```
x = 10
if x > 5:
    print ("x is greater than
    5")
```

Syntax
```
if (condition):
        statements
```

② if. else

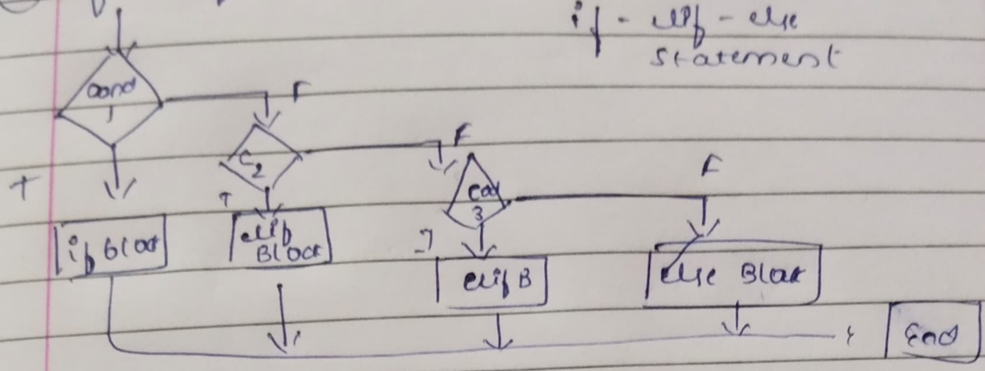


```
x = 2
if (x > 5):
    print ("x is greater
else:
    print (" x. is greater
```
O/p : x is not greater than 5

Date / /
Page
SPLASH

③ if - elif . else [ladder]

if - elif - else
statement



Syntax

```
if (condition 1):
    statement 1
else:
    if (condition 2):
        statement 2
    else:
        if (condition 3):
            statement 4
        else:
            statements.
```

```
if (condition 1):
    statement 1
elif (condition 2):
    statement 2
elif (condition 3):
    statement 3
else:
    statement 4
```

n 5")

than 5")