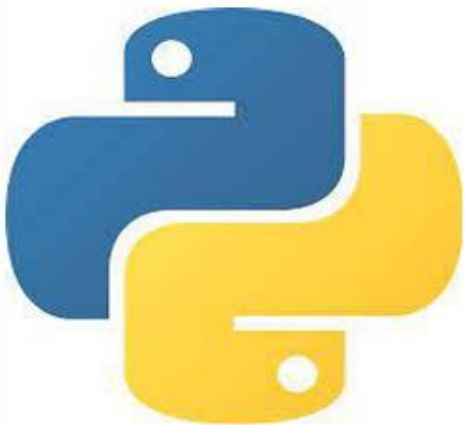
python Programming

K Vinod



K Vinod

Chapter Name
1. Conditional Functionality
2. Loop Functionality
3. User Defined Functions

Python Flow Control

K Vinod

Python supports the usual logical conditions from mathematics:

- Equals: a == b
- Not Equals: a != b
- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b

The if statement in Python is used to execute a specific block of code if a specified condition is true.

syntax: if condition:
code block

If the condition is true, the code block will be executed. If the condition is false, the code block will be skipped and the program will continue to the next statement.

You can also include an optional else statement to specify a code block that will be executed if the condition is false:

```
if condition:
    code block
else:
    code block
```

Additionally, you can include multiple elif statements to provide additional conditions to check:

The elif statements are only evaluated if the previous conditions are false, and the else statement is optional.

```
if condition:
    code block
elif condition:
    code block
else:
    code block
```

Practical 1:


```
# exercise NUMBER GUESSING GAME
# Make a variable, like winning_number and assign any
number to it.
# Ask user to guess a number.
# if user guess correctly then, print "you win !!!"
# if user didn't guess correctly then:
  # if user guessed lower than actual number then print
"too low"
  # if user guessed higher than actual number then
print "too high"
```

Practical 1: Nested if else


```
winning_number = 25
user_input = input("guess a number between 1 to 100 : ")
user_input = int(user_input)
if user_input == winning_number:
    print("you win !!!")

else:
    if user_input < winning_number:
        print("too low")

    else:
        print("too high")
```

And OR operator


```
# check two conditions at same time
# and , or
name = "abc"
age = 19

if name == "abc" or age == 190:
    print("condition true")
else:
    print("condition false")
```

Practical 2:

```
# exercise : WATCH COCO
# Ask user's name and age
# if user's name start with ('a' or 'A') and age is
above 10 then
# print 'you can watch coco movie'
# else print sorry, you cannot watch cocoo
```

In Keyword

```
name = "indian army"
if "t" in name:
    print("a is present in name")
else:
    print("not present")
```

```
# important

name = 'bvhgv'
if name: # true if string is not empty
    print("string is not empty")
else:
    print("string is empty")

name = input("enter your name : ")
if name: # true if string is not empty
    print(f"your name is {name}")
else:
    print("you didn't type anything")
```

Loop Functionality

While Loop


```
# loop ---> while loop for loop
# print("hello ") #10times
i = 1
while i<=10:
    print("hello ")
    i +=1 # i = i + 1
```

While Loop


```
# sum 1 to 10
total = 0
i = 1
while i <= 10:
    total = total + i
    i = i + 1
print(total)
```

Practical 3:

```
# # sum of n natural numbers
# ask a number from user
# print 1 to n
```

Practical 4:


```
# problem
# ask a user to input more than one digit
# example - 1243
# calculate 1+2+4+3 and print

# algorithum - (method to solve in human language)
# ask input in string....don't change string to int
# pick string character one by one and change to int
# int(example[0]+example[1])......go to len[example]
```

Practical 5:


```
# problem
# ask a user to input more than one digit
# example - 1243
# calculate 1+2+4+3 and print

# algorithum - (method to solve in human language)
# ask input in string....don't change string to int
# pick string character one by one and change to int
# int(example[0]+example[1])......go to len[example]
```

Practical 6:


```
# ask a user for name  while loop
# example - indian army
# print count of each word
# i : 2
# n : 1
# d : 1
```

Infinite Loop


```
# # infinite loop {ctrl + c}


i = 0
while i <= 10:
    print("hello world")
while True:
    print("hello")
```

For Loop

Thank You