

Conditional Loop

* To make decisions we use conditional statements.

* The conditional statements in Python

→ if, elif, else

if statement:-

Ex:- stat1
stat2
stat3
if cond:
 stat4
 stat5
stat6
stat7

if condition is True stat1,2,3,4,5,6,7
if condition is False stat1,2,3,6,7

Else statement:-

Ex:- stat1
stat2
stat3
if cond:
 stat4
 stat5
else:
 stat6
 stat7

stat 8
stat 9

if condition is True stat 1, 2, 3, 4, 5, 6, 7, 8, 9
if condition is False stat 1, 2, 3, 6, 7, 8, 9

Ex:- stat 1

stat 2

stat 3

if cond 1:

stat 4

stat 5

if cond 2:

stat 6

stat 7

if cond 3:

stat 8

stat 9

else:

stat 10

stat 11

This else will print when only
"cond 3 is False"

stat 12

stat 13

if cond 1 is True & cond 2, 3 is False - stat 1, 2, 3

if cond 3 is True 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Else if

```
stat 1  
stat 2  
stat 3  
if cond 1:  
    stat 4  
    stat 5  
else if cond 2:  
    stat 6  
    stat 7  
else if cond 3:  
    stat 8  
    stat 9  
else:  
    stat 10  
    stat 11  
stat 12  
stat 13
```

True : 1, 2, 3, 4, 5, 12, 13
True : 1, 2, 3, 6, 7, 12, 13

* Take 3 numbers from the user & find the max out of those 3 numbers :-

```
num 1 = int(input("Enter 1st number :"))
```

```
num 2 = int(input("Enter 2nd number :"))
```

```
num 3 = int(input("Enter 3rd number :"))
```

```
if (num 1 < num 2  
    print("num 1")
```

```
else if num_2 > num_3:  
    print ('num_2')
```

~~else~~:

```
else if num_3 > num_1:  
    print ('num_3')
```

else:

```
    print ('all are equal')
```

Output: num_1 = 10
 num_2 = 20
 num_3 = 30
 num_3

* Design a calculator where two numbers will be taken from user

* operation code will be entered by user

* add

* sub

* multiplication

```
num_1 = int(input("Enter the number :"))
```

```
num_2 = int(input("Enter the number :"))
```

```
op = int(input("Enter the operation"))
```

In 1 for addition

In 2 for subtraction

In 3 for multiplication " ")

- if op == 1:
 print ("num1 + num2")
- elif op == 2:
 print ("num1 - num2")
- elif op == 3:
 print ("num1 * num2")
- else:
 print ("no operation applied")

output: Enter number : 10
 : 20
Enter operation : 2
 -10

- * To find a year entered by user is a leap year or not
 - * If an year is divisible by 4; while it is divisible by 100, if yes then check whether it is divisible by 400.

```
year = int(input("Enter a year:"))
```

```
if year % 4 == 0:
```

```
    print("Divisible by 4")
```

```
if year % 100 == 0:
```

```
    print("Divisible by 100")
```

```
if year % 400 == 0:
```

```
    print("Divisible by 400")
```

```
    print("It's a leap year")
```

```
else:
```

```
    print("Not Divisible by 400")
```

```
    print("Year is not a leap year")
```

```
else:
```

```
    print("Not Divisible by 100")
```

```
    print("Year is not a leap year")
```

```
else:
```

```
    print("Not Divisible by 4")
```

```
    print("Year is not a leap year")
```

- * To check whether a number is a palindrome or not.

```
num = input("Enter a number for Palindrome")
print('Palindrome' if num == num[::-1] else
      'Not Palindrome')
```

output : Enter a number for Palindrome : 12321
Palindrome

Loops in Python

for, while, continuous & break

- * For loop is used to execute a block of code a number of times & in the case where no. of iterations are in advance.

```
for i in range(10):
    print("Hello")
```

Hello
Hello
Hello

- * If you want to do any operations on any wise

```
my-str = 'Python'
```

```
for i in range(range length(my-str)):  
    print(my-str [i] * (i + 1))
```

output : P
y y
t t t
h h h h
o o o o o
n n n n n

- * If you want to do the operation on the seq element wise

List

```
lst = [10, 20, 30, 40]
```

```
for ele in lst:  
    print(ele * 2)
```

output : 20
40
60
80

string
my_str = 'Python'

j = 1

for ele in my_str:

 print(ele * j)

j += 1 → increment in every iteration

output: p
 yy
 tt t
 h h h h
 oo ooo
 nn nnn n

Tuple

tup1 = 10, 20, 30, 40

for ele in tup1:

 print(ele * 2)

output: 20
 40
 60
 80

Set:

```
s = { 20, 30, 40, 50 }  
for ele in s:  
    print (ele * 2)
```

[Output: 40
60
40
100]

while

[white cond :] [If the condition is True then
code] → the code is been executed

↓
is used till a condition is satisfied

initialization

white cond :

Code

Increment/decrement

Ex:-

my - str = 'Python'

ind = 0

```
while ind < len(mystr):  
    print(mystr[ind:(ind+1)])  
    ind += 1
```

Output:-

p
yy
ttt
hhh
ooo
nnn

Break & continue in loop

- * Break is used to exit the current loop completely.
- * continue is used to skip the current iteration.

(Case 1):

```
for i in range(10):  
    stat1  
    stat2  
    print()
```

Output:

stat1	stat1	stat1
stat2	stat2	stat2
stat1	stat1	stat1
stat2	stat2	stat2

(Case 2:

for i in range(5):

 stat 1

 stat 2

 stat 3

 break

output: stat 1
 stat 2
 stat 3

(Case 3:

for i in range(5):

 print(i)

 stat 1

 stat 2

 stat 3

 if print()

 if i == 3:

 break

output: 0

 stat 1

 stat 2

 stat 3

 1

 stat 1

 stat 2

 stat 3

 2
 stat 1
 stat 2
 stat 3

Case 4:

```
for i in range(5):  
    for j in range(5)  
        print(i, j)
```

stat 1

stat 2

stat 3

print()

if j == 3:

if i == 3:
 break

output:-
 break
 0, 0

stat 1

stat 2

stat 3

0, 1

:

0, 2

:

0, 3

:

1, 0

1, 1

:

1, 2

:

1, 3

:

2, 0

:

2, 1

:

2, 2

:

2, 3

Continue

for i in range(3) :

 stat1

 stat2

 print()

 continue

 stat3

Output:

stat1

;

stat2

;

stat1

stat2

stat3

→ stat3 is never
executed bcz after
continues

Write a python program to find below output using loop :-

input :- 'peter piper picked a peck of picked p

output:- ['peter', 'piper', 'picked', 'a',]

```
? f
str1 = 'peter piper picked a peck '
lst1 = []
str2 = ''
for i in range(0, len(str1)):
    if str1[i] != ' ':
        str2 += str1[i]
    else:
        lst1.append(str2)
        str2 = ''
lst1.append(str2)
print(lst1)
```