

2) Control Flow

Here, we are going to review and learn the most fundamentals of Python control flow: Decision making statement and loops.

2.1 Decision Making Statement

- It is kind of making decision during occurred situation of program execution and action can be taken according to specified conditions.
- Structure of decision making evaluate several expressions that provide True or False as a result.
- It is up to you to decide which type of action want to take and execute the statements based upon True and False.

These are several topics of decision making which going to be discussed below:

1. **if**
2. **if else**
3. **if elif else**
4. **Nested if**
5. **Single Statement Suites**

2.1.1 If statement

- In python `if` statement is same as other languages.
- It is having a logical expression where data is getting compared and based upon comparison result a decision is made.

Examples

In [2]:

```
## if else statements
age=int(input("Enter the age"))
if age>=18 and age <=45:
    print("You are young blood")
else:
    print("Sorry only Young bloods are allowed")
```

Enter the age66
Sorry only Young bloods are allowed

2.1.2 If-else statement

- With `if` statement an `else` statement is combined.
- `if-else` statement have a block of code, where `if` is executed when it is 1 or True and `else` when `if` is 0 or False

Note: The `else` statement is consider to be an optional statement, so there could be only one `else` statement in block of code.

Examples

2.1.3 If-elif-else statement

- The `elif` statement is used to check for the multiple `True` expressions and so on if the condition is `True` it execute a block of code.
- Similar `elif` statement is consider to be an optional like `else` statement.
- Though, we can use `elif` statement multiple times in a block of code, but `else` statement can be used only ones

Examples

In [3]:

```
## mall - input the product price
## product >3000 rs 20% off
## price is ==4000 you get a trip to Goa
## print the product price after removing the discount
## product >=2000 rs and <=3000 30% off
## print the product price after removing the discount
## price ==2999 u will get aditional gift
## product 100 rs and <=2000 40% off
## print the product price after removing the discount### 4.1.2 If-else statement

- With `if` statement an `else` statement is combined.
- `if-else` statement have a block of code, where `if` is executed when it is `1` or `
> **Note:** The `else` statement is consider to be an optional statement, so there coul

**Examples**### 4.1.3 If-elif-else statement

- The `elif` statement is used to check for the multiple `True` expressions and so on
- Similar `elif` statement is consider to be an optional like `else` statement.
- Though, we can use `elif` statement multiple times in a block of code, but `else` st

**Examples**
```

In [8]:

```
price=int(input("What's the price of the Product"))
if price>=3000:
    if price==4000:
        print("You are Going to Goa!!!")
        print(f"please pay:{price*0.8} at the counter.")

elif price>=2000 and price<3000:
    if price==2999 :
        print("You get an additional gift!!!")
        print(f"please pay:{price*0.7} at the counter.")
elif price>=100 and price<2000:
    print(f"please pay:{price*0.6} at the counter.")
```

What's the price of the Product2999
 You get an additional gift!!!
 please pay:2099.29999999999997 at the counter.

In []:

#

2.1.4 Nested If statement

- In some of the cases, it is required to have other check condition inside a check condition (when it True)
- Such type of scenarios required to have a nested arrangement.
- Under a nested if arrangement, an `if-elif-else` can be construct inside an another `if-elif-else` construct.

Examples

In [9]:

```
'''Its a simple example to check whether
a number is zero or positive or negative
using nested if-else statement'''

val = float(input("Enter a number: "))
if val >= 0:
    if val == 0:
        print("Zero")
    else:
        print("Entered number is Positive")
else:
    print("Entered number is Negative")
```

Enter a number: -42
Entered number is Negative

2.1.5 Single Statement Suites

- If the suite of `if` section have only a single line, then the header statement may go at same line.

Examples

In [12]:

```
val = int(input("Enter a number less than 1000:- "))
if (val <= 999 ): print ("Value is less than equal to 999") #whole statement in a single
else:print ("I said less than 1000....GoodBye!")
```

Enter a number less than 1000:- 1000
I said less than 1000....GoodBye!

2.2 Loops Statements

- In loop, the statements are sequentially executed, execution of first function statement is done first, then second and so on.
- In situation where you need to perform a block of code many times then loop statement will come in to picture.
- It allows to execute/run group of statements or a statement multiple time.

These are several topics of loop statements which going to be discussed below:

1. **while Loop**
2. **for Loop**
3. **Nested loops**
4. **Loop Control**

- **break**
- **continue**
- **pass**

2.2.1 While Loop

- It repeats a statement or multiple statements when a provided condition is `True` .
- Before performing a loop body it check for a condition

Examples

In [13]:

```
# while loop while-else
joining_age=25
while joining_age<=60:

    joining_age=joining_age+1
    print(joining_age)

else:
    print("Its time for retirement")
### in this case the loop tops after 61.
```

26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

Its time for retirement

In [14]:

```
#to solve the above problem write print statement befor condition  
# while loop while-else  
joining_age=25  
while joining_age<=60:  
    print(joining_age)  
    joining_age=joining_age+1  
  
else:  
    print("Its time for retirement")
```

```
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
Its time for retirement
```

2.2.2 For Loop

- Here for loop statement is used to iterates over an item at any order, items can be a string or a list.

Examples

In [20]:

```
fruits_list = ["Mango", "Cherry", "Apple", "Papaya", "Banana"] ## List containing fruits name
for x in fruits_list:
    print(x)
else:
    print("List has no fruit left!")
```

Mango
Cherry
Apple
Papaya
Banana
List has no fruit left!

In [15]:

```
fruit='mango'
for x in fruit:
    print(x)
#this show that string is also lis(i.e is basically a collection of characters)t by default
```

m
a
n
g
o

In [17]:

```
#so we can use slicing in a string for example
fruit[2]
```

Out[17]:

'n'

In [19]:

```
"""
range(): range(start, stop[, step]) -> range object Return an object that produces a sequence of integers from start (inclusive) to stop (exclusive) by step. range(i, j)
"""
```

Out[19]:

'\nrange(): range(start, stop[, step]) -> range object Return an object that produces a sequence of integers from start (inclusive) to stop (exclusive) by step. range(i, j)\n'

2.2.3 Nested Loops

- Nested loops is used to create one and more number of loop inside an existing for or while loop.

Examples

Q1)- Create a right angle triangle as shown below using loop

```
*
* *
```

In [25]:

[illegible]

In [26]:

* * * * *

In []:

#concept-2

#use "\r" instead of "\n" because in \n a new line is added below which we do not want.

In [28]:

```
n=10 #n is length and width of triangle
for i in range(0, n): #Loop to handle the rows

    for j in range(0, i+1): #Loop to handle the columns

        print("* ",end="") #printing the stars

    print("\r") #after each row line will end
```

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
```

Q2)- Create a equilateral triangle as shown below using loop

```
  *
 ***
*****
*****
*****
```

In [44]:

```

num_rows = 5
row = 0 #row intialize
while(row < num_rows):
    row += 1 #Rows count increase
    s = num_rows - row #Spaces

    sc = 0 #Space counter intialize
    while(sc < s):
        print(" ", end='')
        sc += 1

    stars = 2*row-1 #Number of stars
    while(stars > 0):
        print("*", end='')
        stars -= 1

    print()

```

```

*
***
*****
*****
*****
*****

```

In [49]:

```

nbr_rows=int(input("Enter number of rows"))
row =0
for row in range (1,nbr_rows+1):
    sc=0
    s = nbr_rows-row
    while (sc<s):
        print(" ",end='')
        sc+=1
    str=2*row-1
    str_count = str
    while(str_count>0):
        print("*",end='')
        str_count-=1
    print("\r")

```

Enter number of rows10

```

*
***
*****
*****
*****
*****
*****
*****
*****
*****

```

In []: