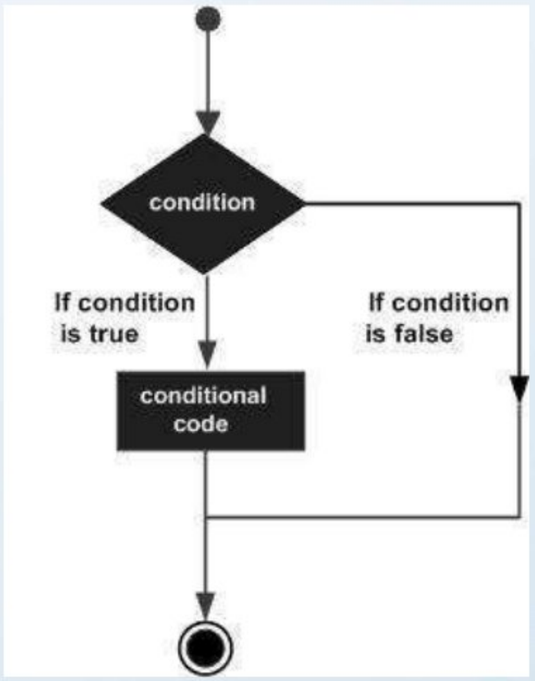
Decision Making

Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.



If Statement

The IF statement is similar to that of other languages. The if statement contains a logical expression using which the data is compared and a decision is made based on the result of the comparison

ifdescisionmaking.py - D:\Documents\Python\ifdescisionmaking.py (3.6.4)

File Edit Format Run Options Window Help

```
# Decision Making
a=1
if a:
    print ( True)

# Decision Making while a variable contain 0
a=0
if a:
    print ( True )
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC
Type "copyright", "credits" or "license () " for more information.
>>>
========== RESTART: D:\Documents\Python\ifdescisionm
True
>>>
```

If-else Statement

An if statement can be followed by an optional else statement, which executes when the boolean expression is FALSE.

ifdescisionmaking.py - D:\Documents\Python

File Edit Format Run Options Window

```
# If-Else statments
x=0
if x:
    print ( True)
else:
    print ( False)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.1900 32 bit
Type "copyright", "credits" or "license ()  " for more information.
>>>
============= RESTART: D:\Documents\Python\ifdescisionmaking.py =====
False
>>>
```

Elif Statement

Elif statement is replacement of else-if of other programming language.

File Edit Format Run Options Windov

```
# If-Elif statments
x,y,z=0,1,0
if x:
    print ( True)
    print  ( x)
elif y:
    print ( True)
    print ( y)
elif z:
    print ( True)
    print  ( z)
else:
    print ( False)
```

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.1900 32 b
Type "copyright", "credits" or "license ()  " for more information.
>>>
========== RESTART: D:\Documents\Python\ifdescisionmaking.py ==
True
1
>>>
```

Nested-If Statement

We can use one if or else if statement inside another if or else if statement(s).

ifdescisionmaking.py - D:\Documents\Python\ifdescisionmaking.py (3.6

File Edit Format Run Options Window Help

```
#Nested If
x,y,z=1,2,3
if x>y:
    if x>z:
        print ( "X is greatest")
    else:
        print ( "X is greater than Y but not Z")
else:
    if y>z:
        print ( "Y is greatest")
    else:
        print ( "Z is greatest")
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)
Type "copyright", "credits" or "license ()  " for more inform
>>>
============= RESTART: D:\Documents\Python\ifdesci
Z is greatest
>>>
```

Relational Operator in Decision Making

File Edit Format Run Options Window Help

```
#Relational Operator  ( ==,<,>,<=,>=,<>,!=)
#<> and != are similar
#<> deprecated in python 3.x
a,b,c,d,e=10,20,5,15,5
if a==b:
    print ( "A = B")
else:
    print ( "False: A!=B")

if a>c:
    print ( "A > C")
if c>=e:
    print ( "C >= E")
if c<=e:
    print ( "C <= E")
if a!=b:
    print ( "True: A != B")
if c==e:
    print ( "C = E")
```

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [M
Type "copyright", "credits" or "license ()  " for more informati
>>>
============= RESTART: D:\Documents\Python\ifdescisic
False: A!=B
A > C
C >= E
C <= E
True: A != B
C = E
```

Membership Operator in Decision Making

ifdescisionmaking.py - D:\Documents\Python\ifdescisionmaking.py (3.6.4)

File Edit Format Run Options Window Help

```
#Membership operator 'in' and 'not in'
string ="Python Programing"
if 'P' in string:
    print ( True)
else:
    print ( False)

if 'm' not in string:
    print ( True)
else:
    print ( False)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:
Type "copyright", "credits" or "license ()  " for mor
>>>
============= RESTART: D:\Documents\Pytho
True
False
>>>
```

Identical Operator in Decision Making

ifdescisionmaking.py - D:\Documents\Python\ifdescisionmaking.py (

File Edit Format Run Options Window Help

```
# Identical operator 'is' and 'is not'
a,b,c=10,20,10
if a is c:
    print ( True)
else:
    print ( False)

if a is not c:
    print ( True)
else:
    print ( False)
```

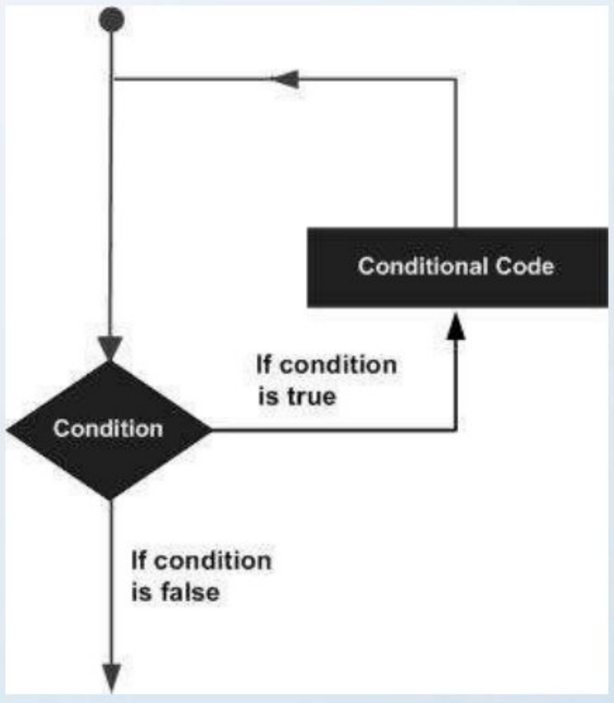
Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)
Type "copyright", "credits" or "license ()  " for more informa
>>>
============= RESTART: D:\Documents\Python\ifdescis
True
False
>>>
```

Looping and Iteration

A loop statement allows us to execute a statement or group of statements multiple times. The following diagram illustrates a loop statement



While Loop

A while loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

looping.py - D:/Documents/Python/looping.py (3

File Edit Format Run Options Window Hel

```
#While loop
cnt=1
while cnt<5:
    print ( "Counter: ",cnt)
    cnt+=1
else:
    print ( "Loop Complete")
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.19
Type "copyright", "credits" or "license ()  " for more information.
>>>
========== RESTART: D:/Documents/Python/looping.py
Counter:  1
Counter:  2
Counter:  3
Counter:  4
Loop Complete
```

For Loop

It has the ability to iterate over the items of any sequence, such as a list or a string.

looping.py - D:/Documents/Python/looping.py (3.6.4)

File Edit Format Run Options Window Help

```
# For loop
# Range is a function which set
# How many time loop will execute
for i in range ( 5)  :
    print ( i)

# Iteration of Characters in string
print ( "Iteration Example")
string="abc"
for s in string:
    print ( s)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04
Type "copyright", "credits" or "license ()  " for more i
>>>
========== RESTART: D:/Documents/P
0
1
2
3
4
Iteration Example
a
b
c
```

Range can accept 3 parameters like [range(1,10,2)] 1 is initial point, 10 is condition and 2 is step

Nested Loops

Python allow nested looping

looping.py - D:\Documents\Python\looping.py (3.6.4

File Edit Format Run Options Window Help

```
# Nested loop
n1=1
while n1<=10:
    n2=1
    while n2<=5:
        print ( n1*n2, end="\t ")
        n2+=1
    print ()
    n1+=1
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC
Type "copyright", "credits" or "license ()  " for more information.
>>>
========== RESTART: D:\Documents\Python\looping
1       2       3       4       5
2       4       6       8       10
3       6       9       12      15
4       8       12      16      20
5       10      15      20      25
6       12      18      24      30
7       14      21      28      35
8       16      24      32      40
9       18      27      36      45
10      20      30      40      50
```

Break and Continue

File Edit Format Run Options Window Help

```
# Break and Continue
a=1
while a<10:
    if ( a==4)  :
        break
    print ( a)
    a+=1
else:
    # if loop terminate on any step using break
    # Else does not working
    print ( "loop is complete")
```

File Edi

```
Pytho
Type
>>>
====
1
2
3
>>>
```

*looping.py - D:\Documents\Python\

File Edit Format Run Options W

```
# Continue
for n in range ( 1,8)  :
    if ( n%2==0)  :
        continue
    print ( n)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Wi

```
Python 3.6.4  ( v3.6.4:d
Type "copyright", "credit
>>>
================== I
1
3
5
7
```

Prime Number Using Nested Loop

primenumber.py - D:/Documents/Python/primenumber.py (3.6.4)

File Edit Format Run Options Window Help

```
# Prime Number Program using loop
for n in range ( 2, 10)  :
    for x in range ( 2, n)  :
        if n % x == 0:
            break
    else:
        print ( n, 'is a prime number')
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window H

```
Python 3.6.4  ( v3.6.4:d48ec
Type "copyright", "credits" or
>>>
================== RESTAR
2 is a prime number
3 is a prime number
5 is a prime number
7 is a prime number
```

Count Number of Digits Using Loop

digitcounter.py - C:/Users/RAJAT TYAGI/AppData/Local/Prog

File Edit Format Run Options Window Help

```
# Program to count number of digits
num=2548971
counter=0
while num!=0:
    counter +=1
    num//=10
print ( counter)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec
Type "copyright", "credits" or "license
>>>
 RESTART: C:/Users/RAJAT TYAGI/
7
>>>
```

Printing Pattern Using Loop

pattern.py - D:/Documents/Python/pattern.py (3.6.4)

File Edit Format Run Options Window Help

```
# Printing star pattern using loop
for n in range ( 5) :
    for c in range  ( n) :
        print ( '*',end=" ")
    print ()
print ()
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.19
Type "copyright", "credits" or "license ()  " for more information.
>>>
================== RESTART: D:/Documents/Python/pattern.py


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

Find a Number is Palindrome or Not

palindrome.py - D:/Documents/Python/palindrome.py (3.6.4)

File Edit Format Run Options Window Help

```
# Palindrome program
orgnum=input ( 'Enter a number: ')
orgnum=int ( orgnum)
num=orgnum
result=0
while ( num!=0)  :
    rem=num%10
    result=result*10+rem
    num=num//10
if ( orgnum==result)  :
    print ( orgnum, ' is palindrome')
else:
    print ( orgnum, ' is not palindrome')
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.1900
Type "copyright", "credits" or "license ()  " for more information.
>>>
================== RESTART: D:/Documents/Python/palindrome.py
Enter a number: 15251
15251  is palindrome
>>>
```

Find a Number is Armstrong or Not

armstrong.py - D:/Documents/Python/armstrong.py (3.6.4)

File Edit Format Run Options Window Help

```
# Armstrong program
orgnum=input ( 'Enter a number: ')
orgnum=int ( orgnum)
num=orgnum
result=0
while ( num!=0)  :
    rem=num%10
    result=result+rem**3
    num=num//10
if ( orgnum==result)  :
    print ( orgnum, ' is Armstrong')
else:
    print ( orgnum, ' is not Armstrong')
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.190(
Type "copyright", "credits" or "license ()  " for more information.
>>>
================== RESTART: D:/Documents/Python/armstrong.py
Enter a number: 371
371  is Armstrong
>>>
```

Printing a Fibonacci Series

fibonacci.py - D:/Documents/Python/fibonacci.py (3

File Edit Format Run Options Window Help

```
# Fibonacci Seriese Program
num1=0
num2=1
for i in range ( 10) :
    num = num1+num2
    num1=num2
    num2=num
    print ( num)
```

Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.4 ( v3.6.4:d48eceb, Dec 19 2017, 06:04:45)  [MSC v.19
Type "copyright", "credits" or "license ()  " for more information.
>>>
========== RESTART: D:/Documents/Python/fibonacci.py
1
2
3
5
8
13
21
34
55
89
```