TOPIC : Artificial Intelligence with Python - INMOVIDU - Jahnavi N

10) LOOPS

Loops are control structures used to repeat a given section of code a certain number of times or until a particular condition is met. There are different loops

1) FOR LOOP

2) WHILE LOOP

3) DO WHILE - python does not have do while but we could do it our selves

#FOR

In [21]:

```
y=11

for i in range(1,y):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [23]:

```
for i in range(0,15,2):
    print(i)
```

```
0
2
4
6
8
10
12
14
```

In [3]:

```
for i in range(10,2,-1):
    print(i)
```

```
10
9
8
7
6
5
4
3
```

In [ ]:

```
#WHILE
```

In [24]:

```
a=1
while(a<=44):
    print(a)
    a=a+4
```

```
1
5
9
13
17
21
25
29
33
37
41
```

In [ ]:

```
#DO while
```

In [3]:

```
q=10
while True:
    print(q)
    q=q+1
    if(q==20):
        break
print("out of loop")
```

```
10
11
12
13
14
15
16
17
18
19
out of loop
```

Entry controlled loop is a loop in which the test condition is checked first, and then loop body will be executed. Exit controlled loop is a loop in which the loop body is executed first and then the given condition is checked afterwards.

* break will bring the program control out of the loop for that we use keywork break itself

In [26]:

```
a=2
while(a<10):
    print(a)
    break
    a=a+2
```

```
2
```

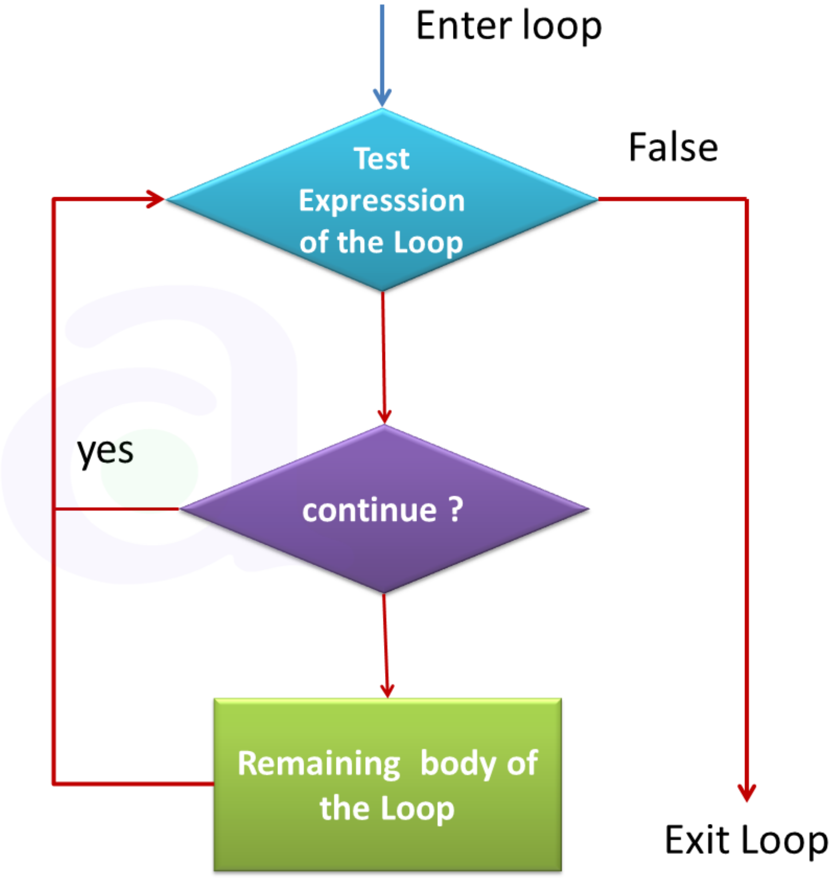
* The continue statement rejects all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop.

For this we use continue

In [27]:

```
from IPython import display
display.Image("http://www.atnyla.com/library/images-tutorials/c-continue-statement-i
```

Out[27]:



In [28]:

```
a=0
while(a<=10):
    a=a+2
    if(a==6):
        continue
    print(a)
```

```
2
4
8
10
12
```

In [ ]:

11) String Operations

Concat

In [29]:

```
s1="hai"
s2="buddies"
```

In [30]:

```
s1=s1+s2
print(s1)
```

```
haibuddies
```

In [23]:

```
s1="hai "
s2="buddies"
s1=s1+"                 "+ s2
print(s1)
```

```
hai                  buddies
```

Length

In [4]:

```
s1="machine"
print(len(s1))
```

```
7
```

upper, lower

In [32]:

```
s1="HAI"
print(s1.lower())
```

```
hai
```

In [28]:

```
s2="hai"
print(s2.upper())
```

```
HAI
```

count

In [5]:

```
s3="aaabbb"
print(s3.count("a"))
```

```
3
```

title

In [6]:

```
s4="we are learning"
print(s4.title())
```

```
We Are Learning
```

find

In [7]:

```
s4="machine learning "
```

In [8]:

```
s4.find('i')
```

Out[8]:

```
4
```

In [34]:

```
print(s4.find("z"))
```

```
-1
```

find() returns -1 if not found

Index

In [9]:

```
print(s4.index('e'))
```

```
6
```

In [32]:

```
print(s4.index('z'))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-32-035c2e90d636> in <module>
----> 1 print(s4.index('z'))

ValueError: substring not found
```

index() Returns an exception if substring isn't found

strip

In [10]:

```
s3="               hai i am human         "
print(s3)
print(s3.strip())
```

```
               hai i am human         
hai i am human
```

replace

In [12]:

```
s="hai people"
s3=s.replace("a","eeee")
```

In [13]:

```
print(s3)
```

```
heeeei people
```

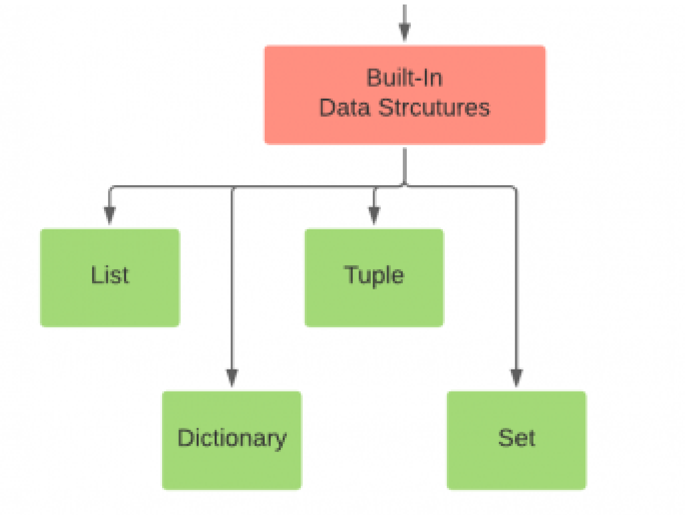
12) DATA STRUCTURES

data structures will give the flexibility to store and access data

In [14]:

```
from IPython import display
display.Image("q.png")
```

Out[14]:



List - allow you to store heterogenous datatypes (Heterogeneous Data Structure – Data elements may not be of same data type)

In [16]:
```
L=[]
```

In [17]:
```
print(type(L))
```

```
<class 'list'>
```

In [19]:
```
L=[1,2,3,4]
```

In [20]:
```
print(L)
```

```
[1, 2, 3, 4]
```

In [ ]:
```

```