Today's topics:

- while loops
- f string
- list comprehension

while Loops in Python

The while loop in Python is used to execute a block of code repeatedly as long as a given condition is True.

while loops

syntax

while condition: statements

In [1]:
```
# basic code
#1-5
choice=1
while choice<6: #true
    print(choice)
    choice+=1
```

```
1
2
3
4
5
```

In [3]:
```
#reverse loop
choice=5
while choice>0: #true
    print(choice)
    choice-=1
```

```
5
4
3
2
1
```

for loops

- when you have any sequence (list,sets,tuples etc)\
- when you know the number of iterations

while loops

- condition based
- when you don't know the number iterations

In [4]:
```
# if else
# marks validation
marks=int(input('enter your marks'))
while marks<0 or marks>100:
    marks = int(input('enter your marks'))
else:
    print('valid marks')
```

```
valid marks
```

In [6]:
```
# email authentications
email=input('enter yor email id')
while '@' not in email and email[-4:]!='.com':
    email=input('enter your email id')
else:
    print('valid email')
```

```
valid email
```

In [8]:
```
password='apkagaurav'
keys=input('enter your password')
count=1
while password!=keys:
    print('attempt number',count)
    keys=input('enter your password')
    count+=1
    if count>5:
        print('-'*30)
        print('device blocked')
        print('-'*30)
        break
else:
    print('valid password')
```

```
attempt number 1
attempt number 2
attempt number 3
valid password
```

In [11]:
```python
# game
# number guessing game

import random
n = random.randint(1,50)
key=int(input('enter a number'))
count=1
while  key!=n:
    print(5-count,'attempts left.')
    print('_'*30)
    if key>n:
        print('try a smaller number')
    elif key<n:
        print('try a bigger number')
    key = int(input('enter a number'))
    count+=1
    if count==5:
        print('0 attempts left'.center(30))
        print('you loss'.upper().center(30))
        print('right answer :',n)
        break
else:
    print('you won')
```

```
4 attempts left.
______________________________
try a smaller number
3 attempts left.
______________________________
try a bigger number
2 attempts left.
______________________________
try a bigger number
1 attempts left.
______________________________
try a smaller number
       0 attempts left
           YOU LOSS
right answer : 34
```

In [12]:
```python
n
```

Out[12]:
```
34
```

list comprehensions

List comphrehensions provide a concise and efficient way to create new lists in python. they offer a more readable and often faster alternative to traditional for loops for list creation and manipulations.

new_list=[expression for item in iterable if condition]

```
In [19]: l1=[1,2,3,4,5,6,7]
         n1=[]
         for i in l1:
             n1.append(i**2)
         n1

Out[19]: [1, 4, 9, 16, 25, 36, 49]

In [20]: [x**2 for x in l1]

Out[20]: [1, 4, 9, 16, 25, 36, 49]

In [21]: # if condition
         # filter
         print(l1)
         [x for x in l1 if x%2==0]

        [1, 2, 3, 4, 5, 6, 7]

Out[21]: [2, 4, 6]

In [ ]: [operation for variable in iterable if condition]
```