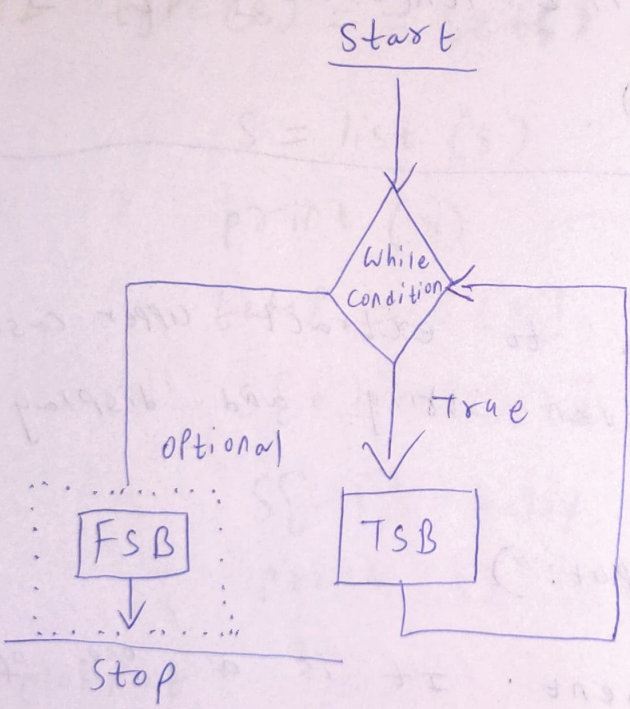
while loop :- It is one of the control and looping statements, it will iterate the set of statements 'n' no. of times. In while statement we have to write the condition or decision. while loop will iterate 'n' no. of times until condition is false.

Flow Chart:

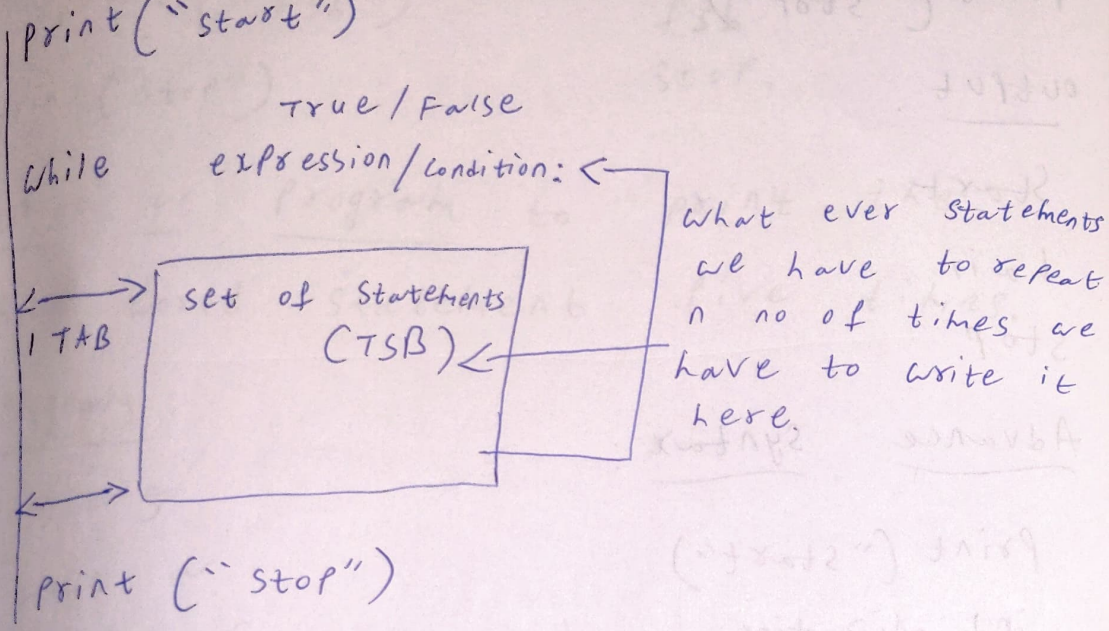


-> When while statement condition is satisfied then true statement block of code is executed then once again control will move on to while condition, until while condition is false.

-> If while condition is unsatisfied or false the control will move

on to while statement or if in case any else block the control will move on to else block then false statement block of code will execute. control will move to while.

Basic while syntax :-



Program:

```
print ("start")
a = True
while a :
    print ("hai")
print ("stop")
```

Output :

```
Start
hai
hai
hai
...
```

Program:

```
print ("start")
a = True        # initilization
while a :       # while with condition
    print ("haii")
    a = False   # updation.
print ("stop")
```

output:

```
start
haii
stop
```

Advance syntax:

```
print ("start")
intialization
while  expression / condition :
    ..... set of statements (TSB) .....
    ..... .....
    print ("stop")
else :
    ..... FSB .....
```

Note: else Statement Block is optional.

program:

```
print("start")
a = True
while a :
    print("haii")
    a = False
else :
    print("fsb")
print("stop")
```

output:

start
haii
fsb
stop.

write a program to print the hello world statement five times.

program:

```
a = 5
while a :
    print ("helloworld")
    a - = 1
```

5 4

While 5 : True
print ("Hello")
a = a - 1
5 - 1
4

3

While 4 : True
Print ("Hello")
a = a - 1
4 - 1
3

2

while (a) : True

print ("Hello")

a = a - 1

3 - 1

2

1

while (a) : True

print ("Hello")

a = a - 1

2 - 1

1

0

while (a) : True

print ("Hello")

a = a - 1

1 - 1

0

a = 0

while (a) : False

Output : 'Hello'

'Hello'

'Hello'

'Hello'

'Hello'.

program:

```
a = 5
while a == 0:
    print ("helloworld")
    a -= 1
```

output:

```
>>>
```

program:

```
a = 5
while a >= 0:
    print ("helloworld")
    a -= 1.
```

output:

```
helloworld
helloworld
helloworld
helloworld
helloworld
helloworld
```

program :

```
a = 5
while a > 0:
    Print ("helloworld")
    a -=1
```

output :

helloworld
helloworld
helloworld
helloworld
helloworld

program :

```
a = 0
while a :
    print("helloworld")
    a+ =1.
```

output :

>>> _____

program :

```
a = 0
while a < 5 :
    print ("Helloworld")
    a + = 1
```

output :

Helloworld
Helloworld
Helloworld
Helloworld
Helloworld.

program:

```
a = 0
while a != 5:
    print ("helloworld")
    a += 1
```

output:

helloworld
helloworld
helloworld
helloworld
helloworld.

program:

```
a = 0
while a != 5 and 5:
    print ("helloworld")
    a += 1
```

output:

helloworld
helloworld
helloworld
helloworld
helloworld.

program:

```
a = 0
while a != 5 or 5:
    print ("helloworld")
    a += 1
```

output:

helloworld
.
.
.
.
.

program:

```
a = 0
while a != 5 or 5-a:
    print ("helloworld")
    a += 1
```

output:

helloworld
helloworld
helloworld
helloworld
helloworld.
~~helloworld~~

21-2-23

write a program to print the even number in reverse order.

```
n = int(input("enter the value"))
while n > 0:
    if n % 2 == 0:
        print(n, end=" ")
    n -= 1
```

output

n = 20

20 18 16 14 12 10 8 6 4 2.

WAP to print the odd numbers in given range values by using a while loop

```
start = int(input("enter the value"))
end = int(input("enter the value"))
while start <= end:
    if start % 2 != 0:
        print(start)
    start += 1
```

output:

Start = 10

end = 20

11
13
15
17
19

WAP to print the numbers which are divisible by 4 and 9 from 1 to 200.

```
start = 1
end = 200
while start <=end:
    if start % 4 == 0 and
        start % 9 == 0:
    print (start)
    start += 1.
```

Output:

36
72
108
144
180

WAP to print the sum of odd numbers within the given range.

```
start = int (input ("enter the value"))
end = int (input ("enter the value"))
sum1 = 0
while start <= end:
    if start % 2 != 0:
        sum1 += start
    start += 1
print (sum1)
```

output :

Start = 0

end = 10

>>> 25

WAP to print sum of even numbers, product of odd numbers in given range.

```
Start = int(input('enter the value'))
end = int(input('enter the value'))
sum1=0
prod = 1
while Start<=end:
    if Start%2==0:
        sum1 += Start
    else
        prod *= Start
    Start+=1
Print(sum1, prod).
```

output : 0 -225

-5	-4	-3	-2	-1	0
-5 x 1	-4+0	-5 x -3	-4-2	+15 x -1	-6
-5	-4	+15	-6	-15	-6

1	2	3	4	5
1 x -15	2+-6	-15x3	-4+4	-45x5
-15	-4	-45	(0)	(-225)

WAP to print product of 1 to n^{th} value.

```
start = 1
end = int(input('enter the value'))
prod = 1
while start <= end :
    prod *= start
    start += 1
print(prod)
```

output : end = 5

120

WAP to print factorial.

```
start = 1
n = int(input('enter the value'))
fact = 1
while start <= n :
    fact *= start
    start += 1
Print(fact)
```

output : n = 5 >>> 120.

n = 5

120 .

Write a program to print the fibonacci series.

10 + 20 = 30

20 + 30 = 50

30 + 50 = 80

50 + 80 = 130

80 + 130 = 210

1

2

3

4

5

c = a + b

print(c)

a = b

b = c.

Program

```
a = 10
b = 20
n = 5
print(a)
print(b)
start = 1
while start <= n:
    c = a + b
    print(c)
    a = b
    b = c
    start += 1
```

Output:

10

20

30

50

80

130

210

Write a program to print the nth fibonocci series.

program

```
a = 0
b = 1
n = int(input("enter the value"))
print(a)
print(b)
start = 1
while start <= n
    c = a + b
    print(c)
    a = b
    b = c
    start += 1
```

output:

n = 5

0
1
1
2
3
5
8.

Write a program to print nth multiplication table from 1 to 20.

n = 5

1	2	3	4	. . . . .	20
*	*	*	*		
n	n	n	n		
5	10	15	20		

Program : Case 1

```
n = 5
start = 1
end = 20
while start <= end:
    print (n * start)
    start += 1
```

Output:

5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100.

Program : Case 2

```
n = 5
start = 1
end = 10
while start <= end:
    print (n, " * ", start, " = ", n * start)
    start += 1
```

Output:

5 * 1 = 5
" " 2 = 10
" " 3 = 15
" " 4 = 20
" " 5 = 25
" " 6 = 30
" " 7 = 35
" " 8 = 40
" " 9 = 45
" " 10 = 50.

WAP to print the factors of given number in the range of 1 to 30.

```
n = 4
start = 1
end = 30
while start <= end:
    if n % start == 0:
        print(start)
    start += 1
```

output:

1
2
4

n = 24

output:

1
2
3
4
6
8
12
24.

WAP to print the even factors of given number ~~in the range~~ ~~of 1 to 30~~.

n = 24

```
start = 1
while start <= n:
    if n % start == 0 and start % 2 == 0:
        print(start)
    start += 1
```

output:

2
4
6
8
12
24.

WAP to print the sum of odd factors of given number.

```
n = 24
start = 1
sum1 = 0
while start <= n:
    if n % start == 0 and start % 2 == 0:
        sum1 += start
    start += 1
print(sum1)
```

output:

4.

WAP to given number is divisible by 1 and itself to display the number. (prime number)

case 1:

```
n = 11
start = 1
count = 0
while start <= n:
    if n % start == 0:
        count += 1
    start += 1
if count == 2:
    print("prime")
```

```
else:
    print("not a prime")
```

output:

prime.

Case 2:

```
n=9
start = 2
count = 0
while start < n:
    if n % start == 0:
        count += 1
    start += 1
if count == 0:
    print("prime")
else:
    print("not a prime")
```

output:

not a prime.

Case 3:

```
n=17
start = 2
while start < n:
```

```
if n%i == 0:
    print("not a prime")
    break
    start += 1
else:
    print("prime")
```

output:

prime.

22-2-23

Break: It is a pre-defined key word. Whenever the controller see a break keyword, automatically controller will stop the iteration of the looping statements.

-> It is a one of the termination statements.
-> It will terminate entire loop.
-> Break keyword, we can use inside the looping statements only either 'while' loop or 'for' loop.
-> After the break keyword whatever statements are present those statements will not execute.
-> once controller see a break statement the controller will move out of the looping statements.

WAP to extract and print the all the characters from given string.

program:

```
st = "helloworld"
start = 0
while start < len(st):
    print(st[start])
    start += 1
```

st = "helloworld"

st[0] = 'h'
st[1] = 'e'
st[2] = 'l'
st[3] = 'l'
st[4] = 'o'
st[5] = 'w'
st[6] = 'o'
st[7] = 'r'
st[8] = 'l'
st[9] = 'd'
st[10] ↓ Index error.

WAP to extract and print from 2nd position to length of the collection.

```
st = "helloworld"
start = 1
while start < len(st):
    print(st[start])
    start += 1
```

WAP to extract and print even position values from given string. or (odd index position)

```
st = "helloworld"
start = 0
while start < len(st):
    if start % 2 != 0:
        print(st[start])
    start += 1.
```

WAP to extract and print the index position values (or) odd position

Program:

```
St = 'helloworld'
Start = 0
while Start < len(st):
    if Start % 2 == 0:
        print(st[start])
    Start += 1
```

WAP to extract and store all the characters from the string.

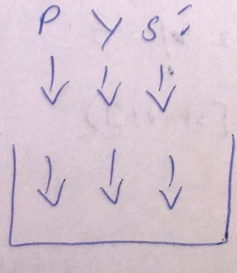
program: (string)

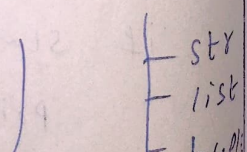
```
res = ""
St = 'hello'
Start = 0
while Start < len(st):
    res = res + st[start]
    Start += 1
print(res)
```

program Logic:

extract "PYS"

store





we cannot use dict ✗ set ✗

Str

```
res = ''  ""
res = res + "val"
   (or)
rest = 'val'
```

eg: hello

Forward Concat

list

```
res = [ ]
res = res + [val]
   (or)
res += [val]
```

```
res = 'val' + res
```

eg: olleh

Backward concat

```
res = [val]
    + res
```

---

tuple

```
res = ( )
res = res + (value,)
   (or)
res += (value,)
```

Forward Concat.

```
res = (value,) + res
```

Backward Concat

WAP to extract and store all the charac. in reverse order from the given string.

Program: case 1:

res = [ ]

```
st = "helloworld"
start = 0
while start < len(st):
    res = [st[start]] + res
    start += 1
print(res)
```

output:

['d', 'l', 'r', 'o', 'w', 'o', 'l', 'l', 'e', 'h']

Program: case 2:

```
st = "HOW ARE YOU"
res = ""
start = -1
while start >= -(len(st)):
    res = res + st[start]
    start -= 1
print(res)
```

output:

UOY ERA WOH

case 2 is another method.

WAP to extract and store odd index position values from given string.

program:

```
res = ""
st = "HOW ARE YOU"
start = 0
while start < len(st):
    if start % 2 != 0:
        res = res + st[start]
    start += 1
print(res)
```

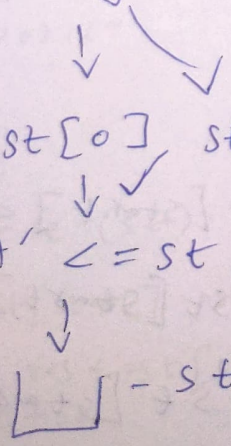
output:

O R O .

WAP to extract and store upper case charac. from given string.

logic:

st = "PysIDErs"



Program:

```
St = "HOW ARE YOU"
res = ""
start = 0
while start < len(st):
    if 'A' <= st[start] <= 'z':
        res = res + st[start]
    start += 1
print(res)
```

output:

HARYU

WAP to seperate the all the charac. from given string and store.

program:

```
St = "HELLO world 123 *&%$# Woo"
UC, LC, SP, oth = '', '', '', ''
start = 0
while start < len(st):
    if not ('A' <= st[start] <= 'z' or
            'a' <= st[start] <= 'z' or
            '0' <= st[start] <= '9')
```

```
        sp+ =st[start]
elif    'o' <=st<= 'q':
                [start]
        an+ =st[start]
elif    'A'<  =St[start]<=  'z':
        uc+ =st[start]
elif    'a'<  =St[start]<=  'z':
        lc+ =st[start]
start+=1
print(uc)
print(lc)
print(sp)
print(an)
```

output:

HELLOWOO
world
*&%.$#
123

WAP to eliminate duplicate Charac. from given string.

Program:

```
st = "helloworld"
res = ""
start = 0
while start < len(st):
        if st[start] not in res:
                res+ =st[start]
        start+=1
Print(res)
```

output:

helowrd,

Logic :

St = "Hello"

check element is present or not → Helo

If element is not present then store the value inside the reference variable

WAP to extract and convert all the charac. upper case to lower case.

Program :

```
St = "HELLO"
res = ""
start = 0
While start < len(st):
    if 'A' <= St[start] <= 'Z':
        res += chr(ord(st[start])+32)
    else:
        res += st[start]
    start += 1
print(res)
```

output :

hello.

23-2-23

WAP to Convert upper case to lower case and vice versa. In given string.

program:

```
st = "HELLOworld@123"
res = ""
start = 0
while start < len(st):
    if 'A' <= st[start] <= 'Z':
        res += chr(ord(st[start])+32)
    elif 'a' <= st[start] <= 'z':
        res += chr(ord(st[start])-32)
    else:
        res += st[start]
    start += 1
print(res)
```

output:

helloWORLD@123.

WAP to extract vowels, special Charac. in given string and display the all the Charac.

```
St = "HELLOworld@123"
res = ""
start = 0
while start < len(st):
    if st[start] in ['A','E','I','O','U'
                     'a','e','i','o','u'
        res += st[start]
    elif ~~st[start]~~ not ('A' <= st[start] <=
                           or 'a' <= st[start] <=
                           or 'o' <= st[start] <=
        res += st[start]
    start += 1
print(res)
```

WAP to find out given Charac. is a special Charac, if it is a special Charac. to extract previous Charac, given Charac., next Charac and store it inside the string.

program : case 1 : (only one character)

```
ch = "$"
res = ""
start = len(st) :
while
if not ('A' <= st[start] <= 'Z'
    or 'a' <= st[start] <= 'z'
    or '0' <= st[start] <= '9'):
    print(chr(ord(ch)+1))
    res += chr(ord(ch)-1) + ch + chr(ord(ch)+1)
print(res)
```

output :

#$%.

case 2 : (string)

program :

```
st = "hello@+%.5"
res = ""
start = 0
while start < len(st):
    if not ('A' <= st[start] <= 'Z'
        or 'a' <= st[start] <= 'z'
        or '0' <= st[start] <= '9'):
        res += chr(ord(ch)-1)+ch+chr(ord(ch)+1)
    start += 1
print(res)
```

output :

?@A*+,$%.&

WAP to Count the number of chr in given str using while loop.

Program: case 1:

```
st = "helloworld"
Count = 0
start = 0
while start < len(st):
    Count += 1
    start += 1
print(count)
```

output:

10

case 2: without using len. function

```
St = "hello"
count = 1
start = -1
while st:
    st = st[start::-1]
    Count += 1
    start -= 1
print(count)
```

output:

5

Logic: "Hello"

		Count
[-1::-1]	"olleH"	1
[-2::-1]	"lleH"	2

[-3 : : -1] `` leh = 3

[-4 : : -1] ``eh'' 4

[-5 : : -1] ``h'' 5

[-6 : : -1] `` ''

Case 3 :

```
st = "hello"
st = list(st)
count = 0
start = 0
while st:
    del st[start]
    count += 1
print(count)
```

output :

5

WAP to count the number of upper case charac. in given string.

```
st = "HELLOworld"
count = 0
start = 0
while start < len(st):
    if 'A' <= st[start] <= 'Z':
        count += 1
    start += 1
Print(count)
```

output :

5

Program I/P "HELLOWORLD$123"

program O/P "hEllO2wO3rld $123"

```
st = "HELLoWoRLD$123"
res = ""
count = 0
start = 0
while start < len(st):
    if st[start] in 'AEIOUaeiou':
        count += 1
    if st[start] in 'aeiou':
        res += chr(ord(st[start])-32)
                + str(count)
    else:
        res += st[start] + str(count)
else
    if 'A' <= st[start] <= 'Z':
        res += chr(ord(st[start])+32)
    else
        res += st[start]
    start += 1
print(res)
```

output: hEllO2wO3rld $123.

WAP to Count the duplicate Charac. ~~value~~ in the string. case 1:

```
st = "PYSPIDERS"
start = 0
count = 0
while start < len(st):
    if st[start] in st:
        count += 1
    start += 1
print(count)
```

output:

2

case 2:

```
st = "PYSPIDERS"
start = 0
res = ""
count = 0
while start < len(st):
    if st[start] not in res:
        res += st[start]
    else:
        count += 1
    start += 1
print(res)
print(count)
```

O/P:

PYSIDER

2

WAP to print prime numbers in given range.

```
i = 1
j = 50
while i <= j:
    n = i
    count = 0
    start = 1
    while start <= n:
        if n % start == 0:
            count += 1
        start += 1
    if count == 2:
        print (i, "is a prime number")
    i += 1
```

Output:

2 is a prime number
3 " " " "
5
7 " " " "
11
13 " " " "
.
.
.
47 " " " " .

24-2-23

WAP to check whether given String is uppercase or not.

Case 1:

```
st = "HELLO*!@123"
start = 0
while start < len(st):
    if 'a' <= st[start] <= 'z':
        print(False)
        break
    start += 1
else:
    print(True)
```

O/P:

True

-> If we use break in while then the controller will go out of 'else'.

Case 2:

```
st = "HELLO@123"
res = ""
start = 0
while start < len(st):
    if not('a' <= st[start] <= 'z'):
        res += st[start]
    start += 1
if st == res:
    print(True)
else:
    print(False)
```

O/P:

True.

Case 3:

```
st = "HEllo@123"
print(st.isupper())
```

O/P:

False

isupper: It is a pre-defined string built-in function. It will find out all the alphabets are upper case or not. If all the alphabets are upper case then it return true, else it return false.

Note:- It will not consider numeric Characters and special Characters.

WAP to Check whether given string is lowercase or not.

```
st = "hello@123"
start = 0
while start < len(st):
    if 'A' <= st[start] <= 'Z':
        print(False)
        break
    start += 1
else:
    print(True)
```

Output:

True.

islower : It is a pre-defined string built-in function. It will find out all the alphabets are lower case or not. If all the alphabets are lower case then it return true, other wise it return false.

upper() : It is a pre-defined string built-in function. This function will Convert ~~uppercase to~~ lower case ~~and~~ to upper case and it return upper case ~~Chi~~ alphabets.

Note :- It will not consider ASCII numbers and special characters.

lower() : It is a pre-defined string built-in function. This function will Convert uppercase to lower case and it return lowercase characters.

Note : It will not consider ASCII numbers and special characters.

WAP to Convert given string to Snake case.

```
st = "hello world hai"
start = 0
res = ""
while start < len(st):
    if st[start] == ' ':
        res += '_'
    else:
        res += st[start]
    start += 1
print(res)
```

O/P:

hello_world_hai

WAP to Convert given string to Snake case with 1st character to be in upper case. Case 1:

```
st = "hello world hai"
res = ""
if 'a' <= st[0] <= 'z':
    res += chr(ord(st[0]) - 32)
else:
    res += st[0]
start = 1
```

```
while start<len(st):
    if st[start]==' ':
        res+='_'
    else:
        res+=st[start]
    start+=1
print(res)
```

O/P :

Hello_world_haii.

Case 2 :

```
st = "Hello world haii"
res = ''
if st[0]==' ':
    res+='_'
elif 'a'<=st[0]<='z':
    res+=chr(ord(st[0])-32)
else:
    res+=st[0]
start=1
while start<len(st):
    if st[start]==' ':
        res+='_'
    else:
        res+=st[start]
    start+=1
print(res)
```

O/P :

Hello_world_haii.

WAP to print CAP first in given string.

```
St = "HelloW@123"
res = ''
if 'a' <= st[0] <= 'z':
    res += chr(ord(st[0]) - 32)
else:
    res += st[0]
res += st[1:]
print(res)
```

O/P:
HelloW@123.

Program:

```
St = "*11pyspider@!@#133"
res =
Start = 0
while Start < len(st):
    if not ('A' <= st[start] <= 'Z'
            or 'a' <= st[start] <=
        res += ''
    else:
        res += st[start]
        break
```

start += 1

```
st=res
res=''
if 'a' <= st[0] <= 'z':
    res += chr(ord(st[0])-32)
else:
    res += st[0]
rest = st[1:]
print(res)
```

O/P:

pyspider@!@#133

```
st = "*123pyspIDER@!@#133"
print(st.capitalize())

st = "py spIDER@!@#133"
print(st.capitalize())
```

#123pyspider@!@#133

~~Pyth~~ → lowercase.

Pyspider@!@#133

↓

uppercase.

25-2-23.

Capitalize:

Program:

```
st = 'HEllo WORLD@gg123''
# print (st.capitalize())
res = ""
if 'a' <= st[0] <= 'z':
    res += chr(ord(st[0])-32)
else :
    res += st[0]
start=1
while start < len(st):
    if 'A' <= st[start] <= 'z':
        res = chr(ord(st[0])+32)
    else :
        res = st[start]
    start += 1
print (res)
```

olp :-

Hello world @gg123.

Title Case:

```
st = "3hEllo  WORLD@gg123pYS"
# print(st.title())
res = ""
if 'a' <= st[0] <= 'z':
    res += chr(ord(st[0]) - 32)
    start = 1
else:
    start = 0
while start < len(st):
    if not ('A' <= st[start] <= 'Z' or
            'a' <= st[start] <= 'z'):
        res += st[start]
        if 'a' <= st[start+1] <= 'z':
            res += chr(ord(st[start+1]) - 32)
        else:
            res += st[start+1]
        start += 1
    elif 'A' <= st[start] <= 'Z':
        res += chr(ord(st[start] + 32)
    else:
        res += st[start]
    start += 1
Print(res)
```

O/P:

3Hello World@Gg123Pys

WAP to Check whether given String is title case or not. (istitle())

Program:

```
St = "Hello World@*1G glA3Pys"
if 'a' <= st[0] <= '}':
    print (False)
elif 'A' <= st[0] <= 'z':
    start = 1
else:
    start = 0
while start < len(st):
    if not ('A' <= st[start] <= 'z' or
            'a' <= st[start] <= 'z'):
        if 'a' <= st[start+1] <= 'z':
            Print (False)
            break
        elif 'A' <= st[start+1] <= 'z':
            start += 1
        elif not ('a' <= st[start] <= 'z'):
            print ('False')
            break
    start += 1
else:
    Print (True)
```

O/p:

True.

27-2-23.

WAP given string is converted to dragonCase.

```
st = "_he llOW*l"
res = ""
start = 0
while start < len(st):
    if 'a' <= st[start] <= 'z':
        res += st[start]
    elif 'A' <= st[start] <= 'Z':
        res += chr(ord(st[start]+32)
    start += 1
res = "__" + res + "__"
print(res)
```

O/P:

__hellow__

WAP given string is dragon case or not.

```
st = "__hello__"
if not (st[0] == '_' and st[1] == '_'
        and st[-1] == '_' and
        st[-2] == '_'):
    Print (False)
```

```
else:
    start = 2
    while start < len(st)-2:
        if not ('a'<=st[start]<=
            print(False)
            break
        start+=1
    else:
        print(True)
```

o/p:

True.

WAP to Check whether given Charac is present in String or not, if Charac is present display true or else false with out using in operator.

```
st = 'PYSPIDERS'
ch = 'I'
i = 0
while i < len(st):
    if ch == st[i]:
        print(True)
        break
    i+=1
else:
    print(False)
```

o/p:

True.

WAP to check whether given Charac is present in String or not, if present display index value or else present '-1'.

```
st = 'PYSPIDERS'
ch = 's'
i = 0
while i < len(st):
    if ch == st[i]:
        print(i)
        break
    i += 1
else
    print(-1)
```

O/P:

2

Program

```
st = 'PYSPIDERS'
print(st.find('y'))
```

>>> 1

Program

```
st = 'PYSPIDERS'
ch = 's'
i = 0
while i<len(st):
```

```
        if (h==st[i]:
            print(i)
            break
        i+=1
else:
    raise valueerror("Element is not present")
```

>>>Value error

```
->raise valueerror("Element is not present")
Value error: Element is not present.
```

WAP to Check whether sub string or sub collection is present or not if present Print true or else false.

```
St = 'PYSPIDERS'
S = 'PID'
start = 0
while start < len(st):
    if S = st[start : start+len(s)]
        print(True)
        break
    start+=1
else:
    print(False)
```

O/p:

True.

WAP to check whether sub string is present in given string if sub string is present to display starting index and ending index of sub string (of given string) or else false.

```
st = 'PYSPIDERS'
s = 'PID'
start = 0
while start < len(st):
    if s == st[start]:
        if s == st[start : start + len(s)]:
            print(start, start + len(s))
            break
        start += 1
    else:
        print(False)
```

O/P:

3 5

WAP to ~~display the~~ Find out the given char is present in given string or not to display the all the occurances of Charac. index values.

```
st = 'PYSPIDERS'
ch = 'S'
flag = False
start = 0
```

```
while start < len(st):
    if ch == st[start]:
        print(start)
        flag = true
    start += 1
if flag == False:
    print(flag)
```

O/P:

2
8

Program: (sub string)

```
st = 'PYSPIDERSPIDG'
ch = 'IG'
flag = False
start = 0
while start < len(st):
    if ch == st[start : start+len(ch)]:
        print(start, start+len(ch)-1)
        flag = True
    start += 1
if flag == False:
    print(flag)
```

O/P:

False

WAP to display ~~second~~ nth occurance index values.

```
st = 'PYSPIDERSPIDGI'
ch = 'I'
occr = 3
count = 0
flag = False
start = 0
while start < len(st):
    if ch == st[start]:
        count += 1
        flag = True
        if count == occr:
            print(start)
    start += 1
if flag == False:
    print(flag)
```

o/p :

13