Date
May 25, 2021

classmate
Date
Page

WEEK 3

INTRODUCTION TO WHILE LOOP

The while loop executes a group of statements as long as the given expression is True.

SYNTAX : while expression :
statement(s)

Here is a basic example -

CODE →

```
i = 0
while i < 5 :
    print(i)
    i += 1
```

OUTPUT →

```
0
1
2
3
4
```

The while loop above kept executing codes until i (the counter) is no longer less than 5.

NOTE → Incrementing the counter is important. The counter in the previous example is the i variable.

If you don't increment the counter, the while loop will not stop until forever.

Using else with while loop

The else statement is executed when the iteration of the while loop ends.

CODE →

```
i = 1
while i <= 2 :
    print ("Hello")
    i += 1
else:
    print ("Loop end")
```

OUTPUT →

```
Hello
Hello
Loop end
```

Practice

(1) Write a code to find a factorial of a no. using while loop.

CODE 1

```
print ("Enter a no. : ")
n = int(input())
i, fact = 1, 1
while (i <= n):
    fact = fact * i
    i += 1
print ("Factorial is :", fact)
```

OUTPUT:

```
Enter a no. :
5
Factorial is : 120
```

CODE 2

```
n = int(input("Enter : "))
i = 1
fact = 1
while (i <= n):
    fact = fact * i
    i += 1
print ("factorial :", fact)
```

OUTPUT →

```
Enter : 5
factorial : 120
```

(2) Find the number of digits in the number input by user.

CODE 1:

```
x = int ( input ("Enter a number:")
print ( len (str(x))          # Using len() function
```

CODE 2:

```
#My Code - using while loop

x = input ("Enter a no. :")

i = 0
count = 0
while ( i <= x.index (x[-1])):
        count = count + 1
        i += 1

if  int (x) < 0 :
    print ("No. of digits :", count-1)
else :
    print ("No. of digits :", count )
```

CODE 3:

```
# Instructor's Code - using while loop

num = abs ( int ( input ("Enter a no. :")
digits = 1
while ( num > 9)
        num = num // 10
        digits = digits + 1
print ( digits )
```

(3) Reverse the digits in the given number.

CODE 1: #My Code

```
x= input ("Enter a no.")

i = 1
rev = " "

while  i<= len(x):
        rev + = x[-i]
        i= i+1

print ( rev)
```

```
if int(x) >0:
    while i<= len(x):
            rev + = x[-i]
            i = i+1
    print (rev)
else:
    while i< len(x):
            rev + = x[-i]
            i = i+1
    print (-int (rev))
```

CODE 2: #Instructor's Code

```
num = int ( input (" Enter a number: ")
absNum = abs (num)
rev = absNum % 10
absNum =  absNum // 10

while ( absNum > 0):
        r = absNum % 10
        absNum = absNum // 10
        rev = rev * 10 + r
if (num > 0):
    print (rev)
else:
    print( rev - 2 * rev)
```

14) Find whether the entered no. is palindrome or not

CODE 1 :

```
#My Code

x = input ("Enter a no.: ")
i = 1
y = x
rev = " "

if int(x) >0 :
    while i<= len(x):
            rev += x[-i]
            i += 1
    print (rev)
    if rev == y :
        print("palindrome")
    else:
        print ("not a palindrome")

else :
    while i< len(x) :
            rev += x[-i]
            i += 1
    reverse = - int(rev)
    print (reverse)
    if abs(reverse) == abs(int(y)):
        print ("palindrome")
    else:
        print ("not a palindrome")
```

CODE 2: # Instructor's Code

```
num = int(input("Enter a number: "))
absNum = abs(num)
rev = absNum % 10
absNum = absNum // 10

while (absNum > 0):
    r = absNum % 10
    absNum = absNum // 10
    rev = rev * 10 + r

if (num < 0):
    rev = rev - 2 * rev

if (rev == num):
    print("palindrome")
else:
    print("Not a palindrome")
```

Date
May 26, 2021

classmate
Date
Page

Introduction to for loop

The for loop is used to loop through or iterate over a sequence or iterable objects.

Iterable objects are strings, lists, sets etc.

SYNTAX: for variable in sequence:
statement(s)

Looping through a list

The for loop is commonly used with lists.

Example,
```
pets = ["cats", "goats", "dogs"]

for pet in pets:
    print(pet)
```

OUTPUT
cats
goats
dogs

Looping through a String

A string is also an iterable object because it is a sequence of characters.

CODE:
```
txt = "Hello"
for x in txt:
    print(x)
```

OUTPUT:
H
e
l
l
o

The range() function

→ When programming, sometimes you will need to repetitively execute a block of codes but you don't have something to loop over.

→ This is where the range() function comes in handy.

→ The range() function creates a sequence of numbers starting from 0.

→ In the following example, the range(5) function will create a sequence of five numbers starting from 0-4.

CODE:
```
for x in range(5):
    print(x)
```

OUTPUT:
```
0
1
2
3
4
```

→ There are three parameters of range() function.

```
range(0, 10, 2)
      ↑   ↑   ↑
   initial end  step
   point point+1
```

CODE:
```
for x in range(3, 8, 2)
    print(x)
```

OUTPUT:
```
3
5
7
```

→ What if we want to print in reverse order :

CODE:
```
for x in range (5, -1, -1):
    print (x)
```

OUTPUT :
5
4
3
2
1
0

The end point should be 1 greater than the expected end point.
For end point as 9, we should fill in 10.

PRACTICE CODES

(1) Write a code using for loop to add the first n numbers.

```
sum = 0
n = int(input("Enter a no.:"))
n = n+1
for x in range (1, n):
    sum = sum + x

print("Sum of first", n-1, "numbers is", sum)
```

(2) Write a code to print multiplication tables :

```
for i in range (11):
    print ("2 x", i, "=", 2*i)
```

Date May 27, 2021

classmate Date Page

Tutorial on 'for' loop AND DIFFERENCE BETWEEN 'for' and 'while' LOOP

PRACTICE CODES

(1) Accept integers using input() function to find max until the input is -1.

"not sure what is asked in question!"

```
x = int ( input ("Enter a no. : ")

while ( x != -1)
    x = int ( input ("Enter a no. : ")
```

---

(2) Find whether the no. is prime or not.

```
x = int ( input ('Enter a number : '))

if x>1 :
    flag = True
    if flag :
        for i in range (2, x) :
            if x% i == 0 :
                flag = False
    if flag :
        print ("prime Number")
```

```
        else:
            print ("Not a prime numbers")
else:
    print ("Not a prime number")
```

(3) Find sum of all digits in a given number:

```
x = int(input('Enter a number:'))

sum = 0
for i in str(x):
    sum = sum + int(i)

print ('sum of all digits is', sum)
```

(4) Find all positive numbers divisible by 3 or 5 which are smaller than the given no.

```
x = int(input("Enter a number:"))

for i in rangl(x):
    if i%3 == 0 or i%5 == 0:
        print(i)
```

(5) Find all the factors of given number.

```
x = int(input("enter a no.:")

for i in range(1, x+1):
    if x% i == 0:
        print(i)
```

Date
May 28, 2021

classmate
Date
Page

Formatted Printing

(1) Formatting Output Using String Modulo Operator (%)

OUTPUT : 2 x 1 = 2

Normal code → print (num, 'x', i, '=', num*i)

Formatted printing (SYNTAX)

```
print('%d x %d = %d' % (num, i, num*i))
```

d → integer
f → float

---

(2) Formatting Output Using format() method :

Normal code : print (num, 'x', i, '=', num*i)

Formatted printing : ~~print~~

```
print('{0} x {1} = {2}'.format(num, i, num*i))
```

{0} {1} {2} ...

0 → num
1 → i
2 → num*i

---

(3) Formatting Output Using f-strings

SYNTAX:

```
print( f " {num} x {i} = {num*i} " )
```

variables inside curly brackets

(4) Format specifiers (for decimal places)

```
# Let us calc. value of pi upto 3 decimal places

x = 22/7
# String Modulo Operator
print ('Value of pi : %.3f' % (x))

# format() method
print ('Value of pi = {0:.2f}' .format(x))

# f-string
print ( f'Value of pi : {x:.3f}')
```

OUTPUT:

```
Value of pi : 3.141
Value of pi : 3.14
Value of pi : 3.141
```

(5) Print statement formatting method (Parameters)

1. end = ''

→ By default python's print function ends with a newline.

→ Python's print() fn comes with a parameter called 'end'.

→ By default, the value of this parameter is '\n', i.e, new line character.

→ You can end a print statement with any character/string using this parameter.

Example Code1:

```
# ends the output with a <space>
print("Welcome to", end = ' ')
print("Python", end = ' ')
```

Output: Welcome to Python

Example Code2:

```
# ends the output with @
print("Stuck", end = '@')
print("home")
```

Output: Stuck @ home

2. **sep = ' '**

The separator between the arguments to print ( ) function is <space> by default, which can be modified and can be made to any characters, integer or strings as per our choice.

The 'sep' parameter is used to achieve the same.

Example Code :
```
# code to remove space
print ('G', 'F', 'E', sep = '')

# for formatting a date
print ('09', '12', '2016', sep = '-')
```

Output :
```
GFE
09-12-2016
```

Date
May 29, 2021

classmate
Date
Page

Nested Loops

Nested for loop

A for loop can be placed inside another for loop.

Example :

```
animal = ["tiger", "cat", "dog"]
sounds = ["roars", "meows", "barks"]

for x in animal :
    for y in sounds :
        print ("the {x} {y}")
```

Output :

```
the tiger roars
the tiger meows
the tiger barks
the cat roars
the cat meows
the cat barks
the dog roars
the dog meows
the dog barks
```

Practice Codes

Problem 1 : Find all the prime no.s less than the entered number.

My CODE :

```
x = int(input())

for i in range(2,x):
    flag = True
    for j in range(2,i):
        if i!=j and i%j==0:
            flag = False
    if flag:
        print(i, end=' ')
```

Instructor's Code :

```
num = int(input('Enter a no.:'))
if (num >= 2):
    print(2, end=' ')
for i in range(3, num):
    flag = False
    for j in range(2,i):
        if (i%j==0):
            flag = False
            break
        else:
            flag = True
    if flag:
        print(i, end=' ')
```

Problem 2: Find the total profit / loss of each trader working in a trading firm. Information regarding no. of traders & no. of transactions is unknown.

Code:

```
empId = input('Enter employee Id:')
While (empId != '-1'):
    trade = int(input('Enter trade amount:'))
    pro_loss = 0
    While (trade != 0):
        pro_loss = pro_loss + trade
        trade = int(input('Enter trade amount:'))
    print(f'Profit / loss for emp {empId} is {pro_loss}')
    empId = input('\n Enter emp Id')
```

Problem 3: Find the day wise total rainfall for the entered duration of time e.g. week, month etc.

Code:

```
days = int(input('Enter the no. of days:')
for i in range(1, days+1):
    total = 0
    rainfall = int(input('Enter the rainfall'))
    While (rainfall != -1):
        total = total + rainfall
        rainfall = int(input('Enter the rainfall'))
    print(f'Total rainfall for day {i} is {total}')
```

Problem 4: Find the length of the longest word from the set of words entered by the user

Code:

```
word = input ("Enter a word :")
maxlen = 0
while (word != '-1'):
    count = 0
    for letter in word:
        count = count + 1
    if (count > maxlen):
        maxlen = count
    word = input ('Enter a word :')
print ('The length of the longest word is :', maxlen)
```

Date
May 30, 2021

classmate
Date
Page

break, continue & pass

(1) break statement

The break statement stops the execution of the current loop.

#1 EXAMPLE : x = 'Hello @ World'

```
for i in x :
    if i == '@' :
        break
    print ( i , end='')
```

OUTPUT : Hello

#2 EXAMPLE :

```
i = 1
while i <= 5 :
    if i == 3 :
        break
    print(i)
    i += 1
```

OUTPUT : 1
2

(2) continue statement

The continue statement terminates the execution of the current iteration of the loop.

And continues the execution of the loop with the next iteration.

EXAMPLE:

```
txt = 'hello'
for x in txt:
    if x == 'l':
        continue
    print(x)
```

OUTPUT

```
h
e
o
```

(3) pass statement

The pass statement is used as a placeholder when a statement is syntactically required.
When it is executed, nothing happens.

EXAMPLE:

```
if 5 == 5:
    pass
else
    pass
print("Hello World")
```

OUTPUT : Hello World