

https://github.com/ibraheem-02/Python_Programs/blob/main/Repititive_Structure.ipynb

Python Loops

Python has two primitive loop commands: while loops, for loops.

while Loop: With the while loop we can execute a set of statements as long as a condition is true.

Example

print the diagonal of stars but take number of star lines from user using while loop.

```
n = int(input("enter the number of lines: "))
count = 1
while (count <= n):
    print("*****")
    count += 1
print("out of loop")
or
```

```
count = 1
while (count <= 5):
    print("*****")
    count += 1
```

take numbers from user and add them using while loop number of repetition is not known when user input a negative number exit the loop.

```
sum = 0
x = int(input("enter the number: "))
while (x >= 0):
    sum = sum + x
    x = int(input("enter the number: "))
print("Sum = ", sum)
# sum = 0
# x = 1
# while (x >= 0):
#     x = int(input("enter the number: "))
#     if (x >= 0):
#         sum = sum + x
#     print("Sum = ", sum)
```

take four number from user and add them using while loop number of repitation is known.

```
sum = 0
n = 1
while (n <= 4):
    x = int(input("enter the number: "))
    sum = sum + x
    n += 1
print("Sum = ", sum)
```

Continue Statement: With the continue statement we can stop the current iteration, and continue with the next:

Break Statement: With the break statement we can stop the loop even if the while condition is true:

Example.

Take numbers from user and add them using while loop number of repitation is not known when user input a Even no sum up odd no ignored and negative number exit the loop Hint use break and continue keyword.

```
sum = 0
n = 1
while (n > 0):
    n = int(input("enter the number: "))
    if (n < 0):
        break
    elif (n % 2 != 0):
        continue
    else:
        sum = sum + n
print("Sum = ", sum)
```

else Statement: With the else statement we can run a block of code once when the condition no longer is true:

Example:

Using else keyword with while loop loop will be terminated when user input a negative number.

```
sum = 0
n = int(input("enter the number: "))
while (n > 0):
    sum = sum + n
    n = int(input("enter the number: "))
else:
    print("Loop ended here")
print("Sum = ", sum)
```

print odd numbers using while from 1-100.

```
c= 1
while (c <= 100):
    if ((c % 2 )== 1):
        print(c)
    c += 1
```

or

```
c= 1
while (c <= 100):
    if ((c % 2 )== 1):
        c = c + 1
        continue
    print(c)
    c += 1
```

or

```
c = 1
while (c <= 100):
    print(c)
    c += 2 # c = c +2
```

Take a number from user and print a table.

```
c =1
n = int(input("enter the number: "))
while (c <= 10):
    print(n, "*", c, "=", n*c)
    c += 1
```

Nested While Loops: To create a nested loop in Python, we can simply place a loop structure inside of the block of statements that is repeated by another loop structure. This is very similar to how we can nest conditional statements as well.

E.g.: Program that print 5 lines 1 2 3 4 5 in one line and sum all these values now the outer loop add the sum 4 times.

https://github.com/ibraheem-02/Python_Programs/blob/main/Nested_WhileLoop.ipynb

```
sum = 0
c2 = 1
while (c2 <= 4):
    c = 1
    while( c <= 5):
        sum = sum + c
        c = c + 1
    c2 = c2 + 1
print("Final sum is : ",sum)
```