Class starts at 9:05 pm :)

ENJOY THE SONG :)

Iterations

Introduction

```
print(1)
print(2)
print(3)
print(4)
print(5)
#DRY -> Don't Repeat Yourself
```

```
1
2
3
4
5
```

```
'''
1. Initialize the starting point
2. Condition till when to repeat
3. Action
4. Updation
'''
```

```
'''
#While Loop

variable initialize #start
while condition:
  action
  update variable

'''
```

Print 1 to n

```
n=int(input())
start=1 #initialize
while start<=n: #condition  1<=10, 2<=10, 3<=10, 4<=10, 5<=10.....10<=10, 11<=10
  print(start) #action      1 2 3 4 5...10
  start+=1 #update          2 3 4 5 6...11

print("Im out of the while loop") #executes once the while condition is false
```

```
10
1
2
3
4
5
6
7
8
9
10
Im out of the while loop
```

```
n=int(input())
start=1 #initialize
while start<=n: #condition
  print(start) #action

print("Im out of the while loop") #executes once the while condition is false

#infinite loop -> no updation
```

Break statement example

```
n=int(input())
start=1 #initialize
while start<=n: #condition
  if(start==5): #1==5 false 2==5 false 3==5 false 4==5 false 5==5 true
    print("Break statement is encountered")
    break
  print(start) #action
  start+=1 #update

print("Im out of the while loop") #executes once the while condition is
```

```
10
1
2
3
4
Break statement is encountered
Im out of the while loop
```

Continue statement example

Skips that iteration

```
n=int(input())
start=0 #initialize
while start<n: #condition  0<=5, 1<=5, 2<=5 3<=5, 4<=5 5<=5, 6<=5 false
  start+=1 #update          1, 2, 3, 4, 5, 6
  if(start==3): #1==3 false  2==3 false 3==3 true 4==3 false, 5==3 false, 6==3 fa
    print("Skip this iteration")
```

```
    continue
  print(start) #action      1 2 4 5 6

print("Im out of the while loop") #executes once the while condition is false
```

```
5
1
2
Skip this iteration
4
5
6
Im out of the while loop
```

```
n=int(input())
start=0 #initialize
while start<=n: #condition  0<=5, 1<=5, 2<=5 3<=5, 4<=5 5<=5, 6<=5 false
  if(start==3): #1==3 false  2==3 false 3==3 true 4==3 false, 5==3 false, 6==3 fa
    print("Skip this iteration")
    continue
  print(start) #action      1 2 4 5 6
  start+=1 #update          1, 2, 3, 4, 5, 6


print("Im out of the while loop") #executes once the while condition is false

#this is an infinte loop because updation comes after conitnue
```

While Else

```
n=int(input())
start=1 #initialize
while start<=n: #condition
  if(start==5):
    print("Break statement is encountered")
    break
  print(start) #action
  start+=1 #update
else: #this block only runs when the loop terminates without comming across the b
  print("Loop terminated without break statement")
```

```
7
1
2
3
4
Break statement is encountered
```

Range Function

range function return values from 0 to n-1

default:

start=0

jump=1

user gives the end value=n

```
'''
# range(end) -> 0 to end-1
range(5) -> 0,1,2,3,4

# range(start,end) -> start to end-1
range(1,5) -> 1,2,3,4

# range(start,end,jump) -> jump defines the step
range(1,6,2) -> 1,3,5

range(-5,-1) -> start=-5, end= -1-1= -2, jump=1
-5,-4,-3,-2

range(0,-5,-1) -> start=0, end = -5-1 = -6, jump =-1
0,-1,-2,-3,-4


range(-5) -> start =0, end=-5-1, jump = 1
nothing

range(-4,-10) -> start = - 4 , end= -10, jump =1
nothing

range(5,1)
nothing

range(5,1,-1) -> start =5, end=1-1=0, jump=-1
5,4,3,2

jump cant be zero
range always takes integer values
```

Quiz 1

```
range(5) # start =0, end=5-1=4, jump = 1
#0 1 2 3 4
```

```
range(0, 5)
```

Quiz 2

```
range(5,15, 2) # start =5, end=15-1=14, jump = 2
# 5 7 9 11 13
```

Quiz 3

```
range(-5,0,1) # start =-5, end= 0 - 1 = -1, jump = 1
# -5 -4 -3 -2 -1
```

Quiz 4

```
range(-10,-5,-1)  # start = -10, end= -5, jump =-1
#nothing
```

```
'''
Break : 10:20 pm - 10:30 pm
'''
```

Iterables and Iterators

```
'''
Iterables - list, string, tuple, set, or range
Itertor: Loops -> for
'''
```

For loop

```
for variable in iterable:
  action
```

```
for i in range(1,101): # start=1, end=100, jump=1
  print(i, end=" ")
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
```

Quiz 5

```
for i in range(0,1): #Start=0, end=1-1=0, jump=1 range-> 0
    #LOOP GETS ONLY 1 VALUE WHICH IS 0, HENCE IT RUNS 1 TIME
    print('Hello')
```

```
Hello
```

Variable name

```
for _ in range(1,10):
  print(_)
```

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

Break and Continue in for loop

```
for i in range(10): #range(0,1,2,3,..9)
  if(i==4):
    break
  print(i)
```

```
0
1
2
3
```

```
for i in range(10): #start=0, end=9, jump=1, 0, 1, 2, 3
  if(i%2==0): #0%2==0 t 1%2==1 f, 2%2==0 t, 3%2==1 f
    continue
  print(i)
```

```
1
3
5
7
9
```

Pass Statement: only used in testing

pass does absolutely nothing, it acts like a placeholder for future code

```
for i in range(10):
  if(i%2==0):
    pass
  print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

Quiz 6

```
for i in range(0,10): # 1 2 3 4 5 6 7 8 9
    if(i % 3 == 0): #3%3==0, 6%3==0, 9%3==0
        continue
    print(i, end = ' ') # 1 2 4 5 7 8
```

```
 1 2 4 5 7 8
```

Quiz 7

```
for i in range(1,10): # 1 2 3
    if(i % 3 == 0): #1%3==0 f, 2%3==0 f, 3%3==0 t
        break
    print(i, end = ' ') # 1 2
```

```
 1 2
```

Nested Loops

```
'''
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
'''


for i in range(1,6):
  print(i, end=" ")
print()
for i in range(1,6):
  print(i, end=" ")
print()
for i in range(1,6):
  print(i, end=" ")
print()
for i in range(1,6):
  print(i, end=" ")
print()
for i in range(1,6):
  print(i, end=" ")
```

```
 1 2 3 4 5
    1 2 3 4 5
    1 2 3 4 5
    1 2 3 4 5
    1 2 3 4 5
```

```
# how many times you have to perform the task -> repeat 5 times
# figure out the task -> print(1,5)


for i in range(5): #0, 1, 2, 3, 4
  for j in range(1,6): #1,2,3,4,5
    print(j, end=" ")
  print()
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

Difference btw for and while loop

```
#FOR loop
1. Best for known range
2. All 3 steps are in the same line
3. Few chances to goto an infinite loop

#WHILE loop
1. Best known for unknown range
2. All steps are seperate
3. Can easily go into infinte loop
```