



Lab Work – Loops

Objective

This lab helps develop a practical understanding of how Python loops work and how they are applied in engineering and AI-related tasks. The focus is on using for loops and while loops to iterate through data, perform calculations, and simulate real-world processes. Through examples such as resistor value verification, current power loss calculation, serial number generation, voltage and capacitor data handling, battery charging simulation, countdown timers, and motor speed control, the lab strengthens skills in data iteration, loop control, conditional execution, and practical automation in Python.

For Loop

Task 1 — Print resistor values

Create list of resistors.

Loop through list and print each value.

```
In [1]: values=[10,220,56,780.100,340,210]
        for v in values:
            print(v)
```

```
10
220
56
780.1
340
210
```

Task 2 — Square current readings

Create list of currents.

Loop and print I^2 for each current.

```
In [2]: currents=[110,78,89,220,120,56,11]
        for a in currents:
            power_loss= a*a
            print(power_loss)
```

```
12100
6084
7921
48400
14400
3136
121
```

Task 3 — Generate serial numbers

Use range(1,11).

Loop and print each serial number.

```
In [3]: for serial_num in range(1,11):  
        print(serial_num)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Task 4 — Print voltage values

Create list of voltages.

Loop and print each value.

```
In [4]: voltages=[220,11,0,110,560]  
        for v in voltages:  
            print(v)
```

```
220  
11  
0  
110  
560
```

Task 5 — Double capacitor values

Create list of capacitors.

Loop and print double of each value.

```
In [5]: capacitances=[10,45,11,34,6,89] #μF  
        for c in capacitances:  
            double=c*2  
            print(double)
```

```
20  
90  
22  
68  
12  
178
```

Task 6 — Print sensor temperatures

Create list of temperatures.

Loop and print each sensor's value.

```
In [6]: temps=[23.9,43,50,10.5]
        for t in temps:
            print(t)
```

```
23.9
43
50
10.5
```

Task 7 — Half solar energy readings

Create list of daily energy.

Loop and print half of each value.

```
In [7]: energy_usage=[5600,2400,12,10,6] #kWh
        for e in energy_usage:
            half=e/2
            print(half)
```

```
2800.0
1200.0
6.0
5.0
3.0
```

While Loop

Task 1 — Simulate battery charging

Start battery = 0.

While battery < 100: print "Charging..." and add 10.

```
In [8]: battery=0
        while battery<100:
            print("charging")
            battery+=10
```

charging
charging
charging
charging
charging
charging
charging
charging
charging
charging

Task 2 — Countdown timer

Start timer = 10.

While timer \geq 0: print "Test Running" and decrease timer.

```
In [9]: timer=10
while timer>=0:
    print("Test Running:",timer)
    timer-=1
```

Test Running: 10
Test Running: 9
Test Running: 8
Test Running: 7
Test Running: 6
Test Running: 5
Test Running: 4
Test Running: 3
Test Running: 2
Test Running: 1
Test Running: 0

Task 3 — Voltage input until 0

Use input() repeatedly.

Stop loop when user enters 0.

```
In [10]: voltage=1
while voltage!=0:
    voltage=float(input("Enter voltage:"))
    if voltage !=0:
        print("voltage entered was:",voltage)
```

voltage entered was: 45.0

Task 4 — Power doubling simulation

Start power = 1.

While power \leq 1000: double power each iteration.

```
In [11]: power=1
while power <=1000:
    print("power:", power,"w")
    power = power * 2
print("Power limit exceeded!")
```

```
power: 1 w
power: 2 w
power: 4 w
power: 8 w
power: 16 w
power: 32 w
power: 64 w
power: 128 w
power: 256 w
power: 512 w
Power limit exceeded!
```

Task 5 — Cycle running simulation

Start cycle = 1.

While cycle \leq 5: print "Cycle Running" and increment cycle.

```
In [12]: cycle=1
while cycle <=5:
    print("cycle running:",cycle)
    cycle +=1
print("process completed!")
```

```
cycle running: 1
cycle running: 2
cycle running: 3
cycle running: 4
cycle running: 5
process completed!
```

Task 6 — Motor speed simulation

Start speed = 0.

While speed \leq 3000: add 500 and print each speed.

```
In [1]: cycle=0
while cycle<=3000:
    print("Speed",cycle, "RPM")
    cycle+=500
print("Limit reached!")
```

Speed 0 RPM
Speed 500 RPM
Speed 1000 RPM
Speed 1500 RPM
Speed 2000 RPM
Speed 2500 RPM
Speed 3000 RPM
Limit reached!