Loops: Repeating Without Retyping

Python for Absolute Beginners · Lesson 10

What you'll learn

- What a loop is, and why programmers love them
- How a for loop repeats a set number of times
- How range() counts for you
- How a while loop repeats until a condition changes
- How to avoid the one loop mistake every beginner makes

Why loops exist

- A loop repeats a block of code for you, so you write the action once and say how many times to do it
- Printing 1 to 100 without a loop would take a hundred lines — and changing the plan means rewriting all of them

Without a loop: five lines for 1 to 5

```
print(1)
print(2)
print(3)
print(4)
print(5)

Output:
1
2
3
4
5
```

That works, but it does not scale.

For loops and range()

- A for loop repeats its block once for each item in a sequence
- The simplest sequence comes from range(), which counts
- range(5) produces 0, 1, 2, 3, 4 — five numbers, starting at 0
- range(5) means “five times”, and the counting variable takes each value in turn
- range(start, stop) counts from start up to, but not including, stop



A for loop with range()

```
for number in range(5):
    print(number)

Output:
0
1
2
3
4
```

That zero start surprises everyone once, then becomes habit.

Range with a start and stop

```
                    for number in range(1, 6):
    print(number)

Output:
1
2
3
4
5
```

The stop value is always one past the last number you get.

A loop through a string

```
for letter in "cat":
    print(letter)

Output:
c
a
t
```

Same shape: “for each thing in this sequence, do the indented block” — here the sequence is characters.

While loops: repeat until something changes

- A while loop repeats its block as long as its condition stays true, and stops the moment it becomes false
- The block must change something the condition tests — otherwise nothing ever changes

A while loop in code

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

Output:
1
2
3
4
5
```

When count reaches 6, the condition is false and the loop stops.

The one loop mistake every beginner makes

- A while loop whose condition never becomes false runs forever — an infinite loop
- It usually comes from forgetting the update line, so the tested value never changes
- If a program prints the same thing endlessly, stop it — in IDLE, use Shell → Restart Shell, or close the shell window
- A for loop cannot run forever this way, because range() always runs out

An infinite loop (do not run this as-is)

```
                              count = 1
while count <= 5:
    print(count)
```

This prints 1 forever, because count never changes.

Hands-on exercise

- Use a for loop with range() to print the numbers 1 to 10
- Use a while loop to count down from 5 to 1, then print “Blast off!” after the loop
- Use a for loop to print each letter of your name on its own line
- Predict what range(3) produces before running it, then check

Key takeaways

- A loop repeats an indented block so you write an action once
- for repeats once per item in a sequence; range(n) counts 0 to n−1
- range(start, stop) counts from start up to, but not including, stop
- while repeats while its condition is true; the block must change something the condition tests
- A while loop that never changes its condition runs forever

Take the quiz

Take the Lesson 10 quiz — 5 quick questions to check what you learned.

Next lesson

Drawing with Turtle