



Python Programming

asaad.nayyef@uomus.edu.iq

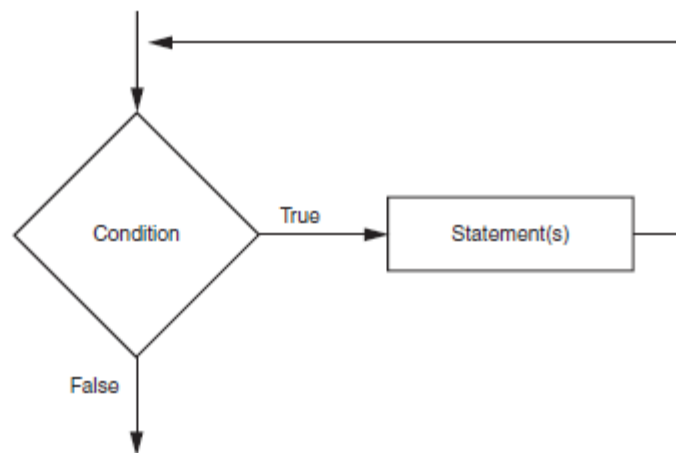


Week 3: Loops and Iterations

1. The while Loop: A Condition-Controlled Loop

A condition-controlled loop causes a statement or set of statements to repeat as long as a condition is true. In Python you use the *while* statement to write a condition-controlled loop. The while loop gets its name from the way it works: while a condition is true, do some task. The loop has two parts: (1) a condition that is tested for a true or false value, and (2) a statement or set of statements that is repeated as long as the condition is true. Figure 5-1 shows the logic of a while loop.

Figure The logic of a while loop



The general format of the while loop in Python:

```
while condition:
```

```
statement
```

```
statement
```

```
etc.
```

Using while loop with a condition

Print numbers from 1 to 5

```
# Print numbers from 1 to 5
count = 1
while count <= 5:
    print(f"Count: {count}")
    count += 1
```

Count: 1

Count: 2

Count: 3

Count: 4

Count: 5



Python Programming

asaad.nayyef@uomus.edu.iq



while loop with else

Check if a number is prime

```
# Check if a number is prime
num = int(input("Enter a number: "))
divisor = 2

while divisor < num:
    if num % divisor == 0:
        print(f"{num} is not a prime number.")
        break
    divisor += 1
else:
    print(f"{num} is a prime number.")
```

Enter a number: 7

7 is a prime number.

2. The for Loop: A Count-Controlled Loop

A count-controlled loop iterates a specific number of times. In Python you use the for statement to write a count-controlled loop.

The general format:

for variable in [value1, value2, etc.]:

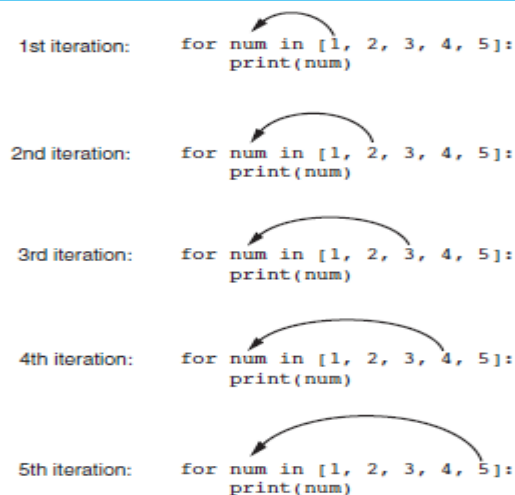
statement

statement

etc.

Using for loop with conditional checks

Figure The for loop





Python Programming

asaad.nayyef@uomus.edu.iq



```
count = 1
for count in range(1, 6):
    print(f"Count: {count}")
```

Count: 1
Count: 2
Count: 3
Count: 4
Count: 5

```
for i in range(5, 0, -1):
    print(i)
```

5
4
3
2
1

```
for i in range(5):
    print(5 - i)
```

5
4
3
2
1

```
# Print only even numbers from 1 to 10
for num in range(1, 11):
    if num % 2 == 0:
        print(f"Even number: {num}")
```

Even number: 2
Even number: 4
Even number: 6
Even number: 8
Even number: 10

3. Using break and continue for conditional checks

#Find the First Divisible Number (Using break)

Find the first number divisible by 7 in a range

```
# Find the first number divisible by 7 in a range
for num in range(1, 21):
    if num % 7 == 0:
        print(f"The first number divisible by 7 is
```



Python Programming

asaad.nayyef@uomosul.edu.iq



```
{num}." )  
    break
```

The first number divisible by 7 is 7.

#Skip Odd Numbers (Using continue)

Print only even numbers between 1 and 10

```
# Print only even numbers between 1 and 10  
for num in range(1, 11):  
    if num % 2 != 0:  
        continue  
    print(f"Even number: {num}")
```

Even number: 2

Even number: 4

Even number: 6

Even number: 8

Even number: 10

#Skip Specific Numbers (Using continue)

Skip printing the numbers 3 and 6

```
# Skip printing the numbers 3 and 6  
for num in range(1, 10):  
    if num == 3 or num == 6:  
        continue  
    print(f"Number: {num}")
```

Number: 1

Number: 2

Number: 4

Number: 5

Number: 7

Number: 8

Number: 9

#Find Prime Numbers (Using break)

Check for prime numbers in a range

```
# Check for prime numbers in a range  
for num in range(2, 20):  
    is_prime = True  
    for i in range(2, int(num ** 0.5) + 1):  
        if num % i == 0:  
            is_prime = False  
            break  
    if is_prime:  
        print(f"{num} is a prime number.")
```

2 is a prime number.



Python Programming

asaad.nayyef@uomus.edu.iq



3 is a prime number.
5 is a prime number.
7 is a prime number.
11 is a prime number.
13 is a prime number.
17 is a prime number.
19 is a prime number.

Print a right-angled triangle pattern

```
# Print a right-angled triangle pattern
rows = 5
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print('*', end='')
    print() # New line
```

*

**

Factorial Using for Loop

```
# Factorial calculation using a for loop
number = int(input("Enter a number: "))

if number < 0:
    print("Factorial is not defined for negative numbers.")
else:
    factorial = 1
    for i in range(1, number + 1):
        factorial *= i
    print(f"The factorial of {number} is {factorial}.")
```

Enter a number: 5

The factorial of 5 is 120.



Python Programming

asaad.nayyef@uomos.edu.iq



Factorial Using while Loop

```
# Factorial calculation using a while loop
number = int(input("Enter a number: "))

if number < 0:
    print("Factorial is not defined for negative numbers.")
else:
    factorial = 1
    i = 1
    while i <= number:
        factorial *= i
        i += 1
    print(f"The factorial of {number} is {factorial}.")
```

Enter a number: 5

The factorial of 5 is 120.