

# **CPSC 1101**

# **Introduction to Computing**

School of Engineering & Computing  
Dept. of Computer Science & Engineering  
Dr. Sidike Paheding

# Agenda

- Recap
- Lecture on for-loop
- In-class exercises
- Reading Assignment. Chapter 3: p.84-92

# Relational operators

## Operator Name

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

# Logical operators

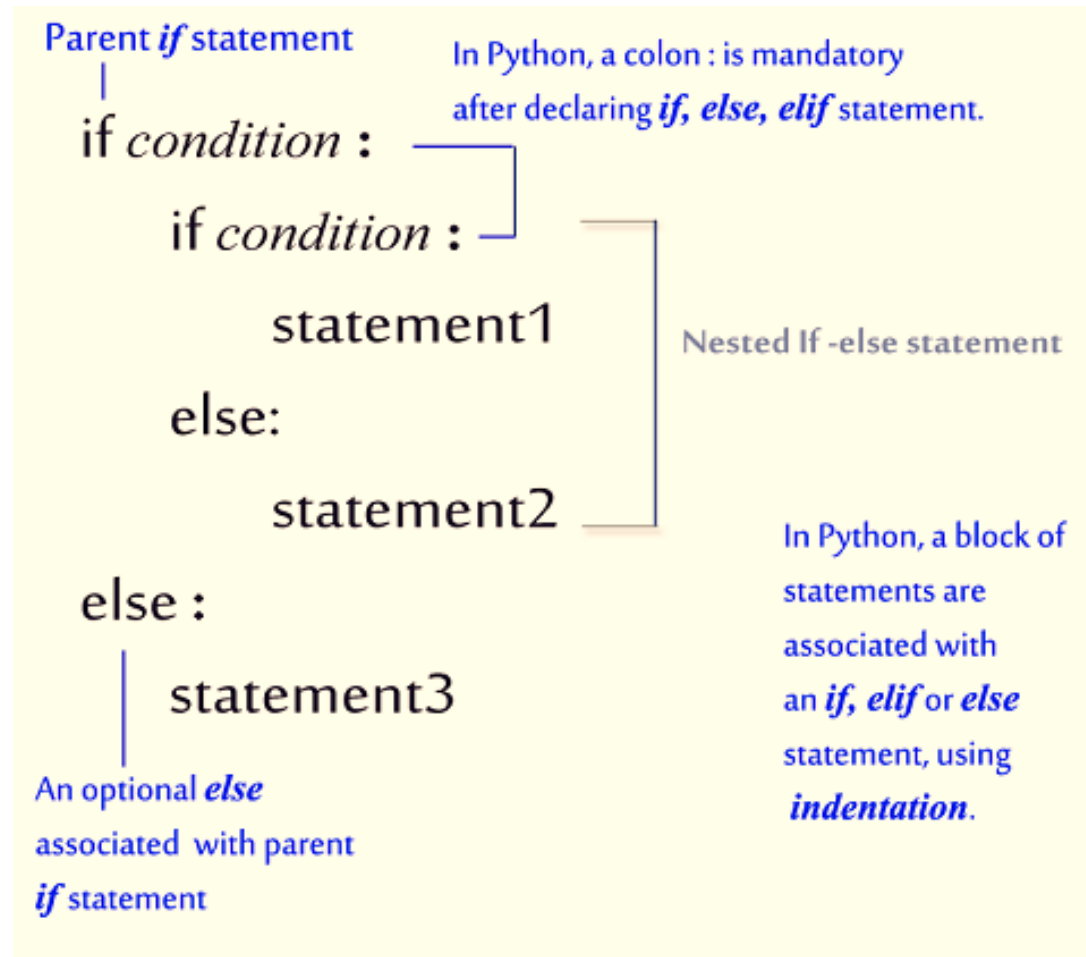
Operator	Name
----------	------

- |       |     |
|-------|-----|
| • and | AND |
| • or  | OR  |
| • not | NOT |

## Order of precedence

- NOT operator
- AND operator
- OR operator

# If-statements

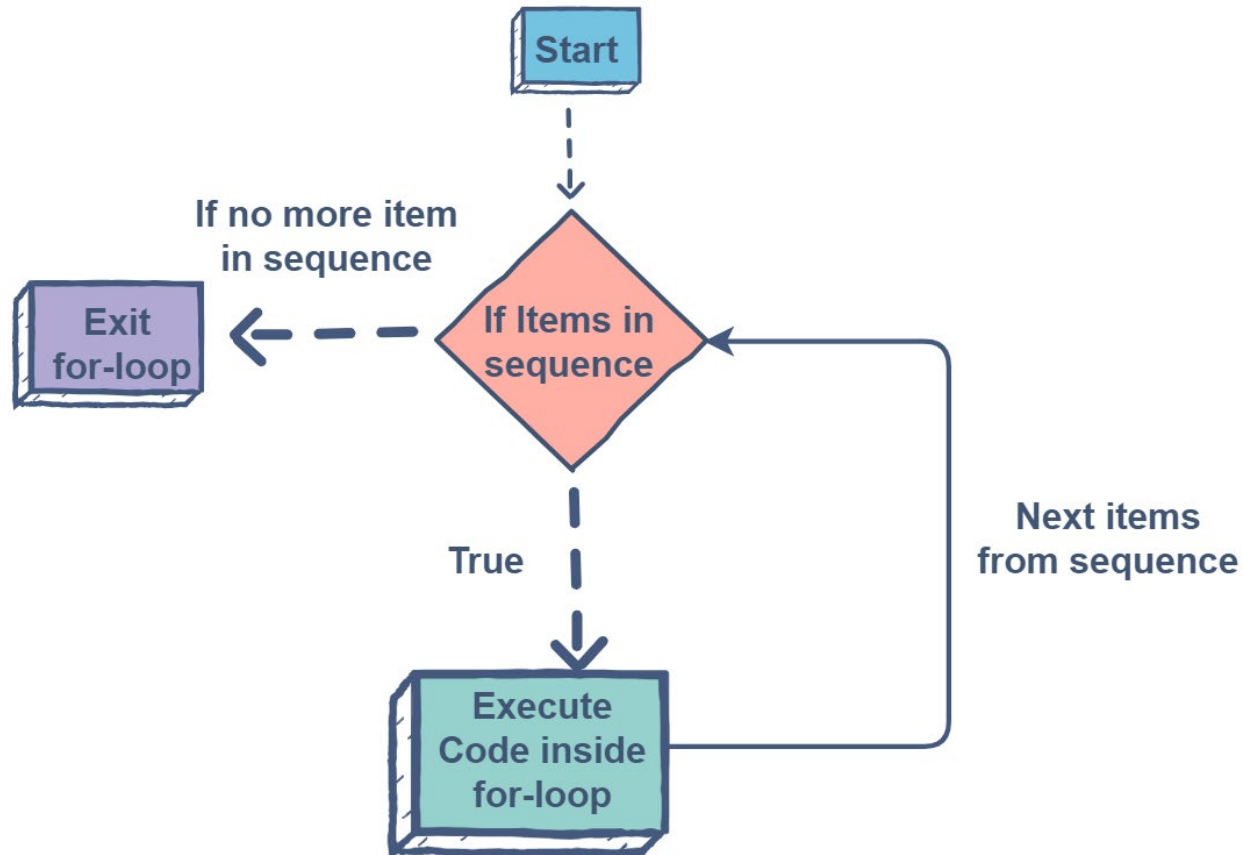


# Iteration Structure in Python

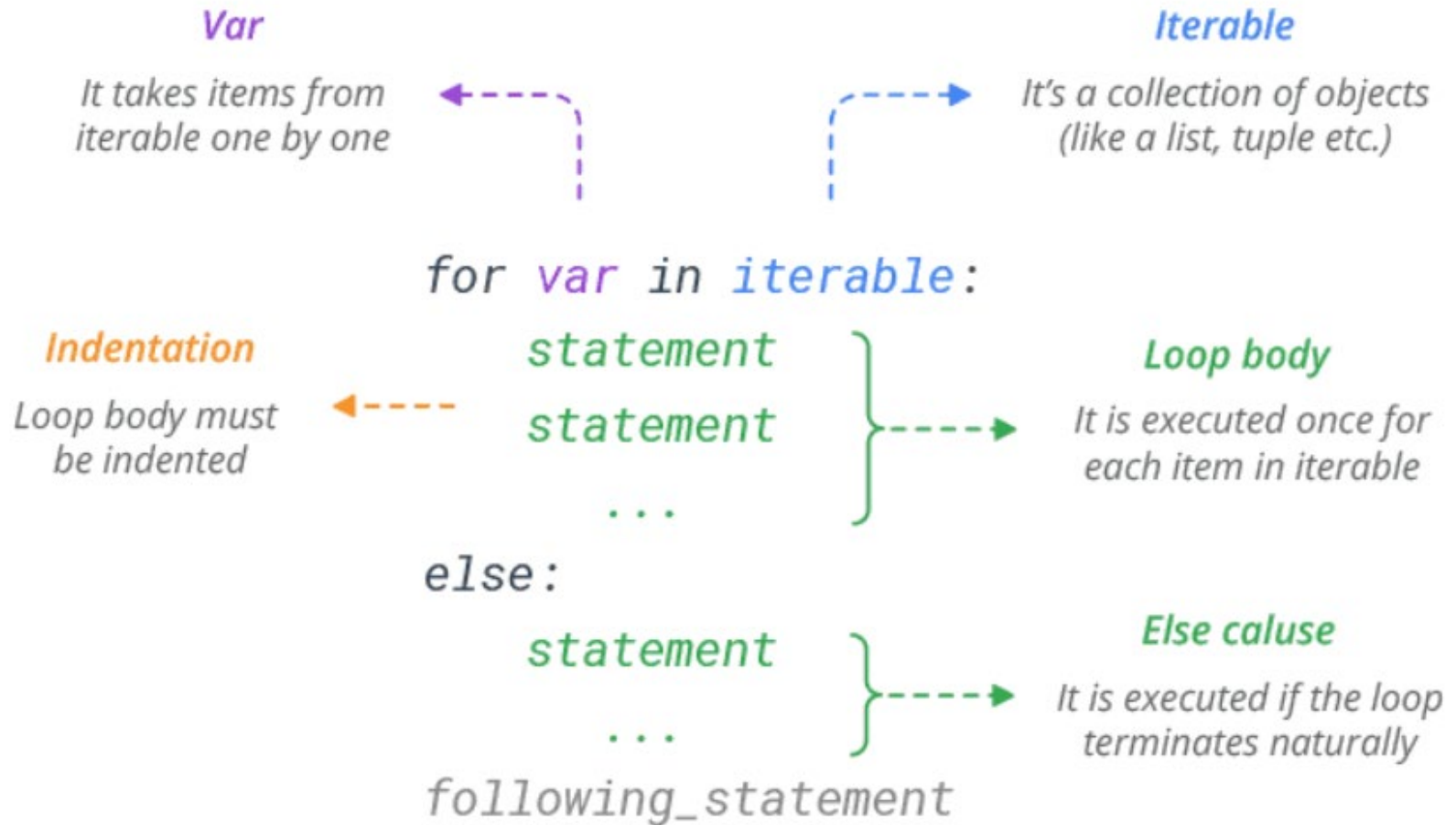
*for-loop*

# For loop

- A for loop is used for iterating over a sequence (e.g., a list, a tuple, a dictionary, a set, or a string).



# For loop Syntax





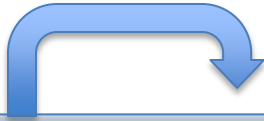
# For loop

- The book gives the following example:

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

- name: the loop variable
- The brackets contain loop values called a list
- The print statement in the loop body
- The “else” keyword in a for loop specifies a block of code to be executed when the loop is finished

# For loop



```
for name in ["Joe", "Amy", "Brad"]:
```

```
    print("Hi", name, "Please come to my party on Saturday!")
```

```
else:
```

```
    print("Be there or be square!")
```

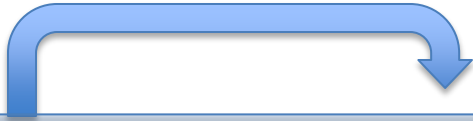
# For loop

```
for name in ["Joe", "Amy", "Brad"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!
```

# For loop



```
for name in ["Joe", "Amy", "Brad"]:
```

```
    print("Hi", name, "Please come to my party on Saturday!")
```

```
else:
```

```
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!
```

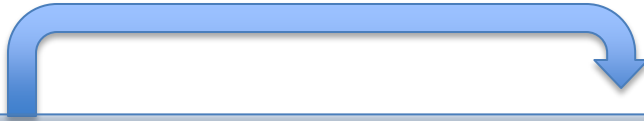
# For loop

```
for name in ["Joe", "Amy", "Brad"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!  
Hi Amy Please come to my party on Saturday!
```

# For loop



```
for name in ["Joe", "Amy", "Brad"]:
```

```
    print("Hi", name, "Please come to my party on Saturday!")
```

```
else:
```

```
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday! y!  
Hi Amy Please come to my party on Saturday!
```

# For loop

```
for name in ["Joe", "Amy", "Brad"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!  
Hi Amy Please come to my party on Saturday!  
Hi Brad Please come to my party on Saturday!
```

# For loop

```
for name in ["Joe", "Amy", "Brad"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!  
Hi Amy Please come to my party on Saturday!  
Hi Brad Please come to my party on Saturday!
```



# For loop

```
for name in ["Joe", "Amy", "Brad"]:  
    print("Hi", name, "Please come to my party on Saturday!")  
else:  
    print("Be there or be square!")
```

## Output:

```
Hi Joe Please come to my party on Saturday!  
Hi Amy Please come to my party on Saturday!  
Hi Brad Please come to my party on Saturday!  
Be there or be square
```

# For loop with range

- Iterates from the beginning to the end of the range. If not specified start at 0. Not inclusive of endpoint

#prints 0-19

```
for x in range(20):  
    print ("x is now: ", x)
```

```
for x in range(2, 6):  
    print(x)
```

# For loop with range

- The range() function defaults to increment the sequence by 1, however it is possible to specify the increment value by adding a third parameter:

```
for x in range(2, 30, 3):  
    print(x)
```

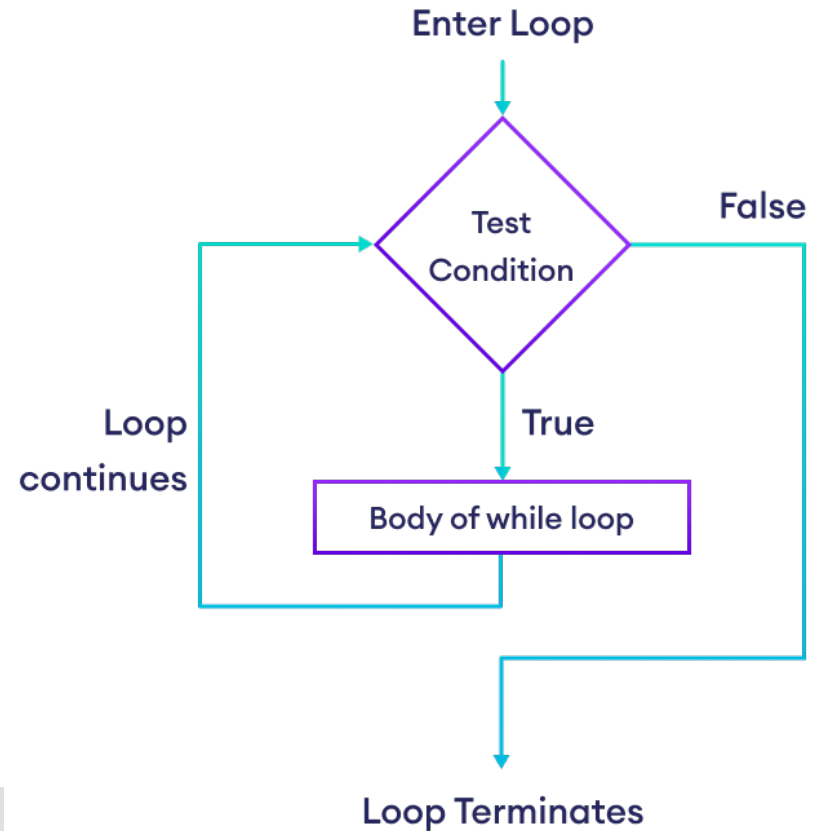
**Output:**

```
2  
5  
8  
11  
14  
17  
20  
23  
26  
29
```

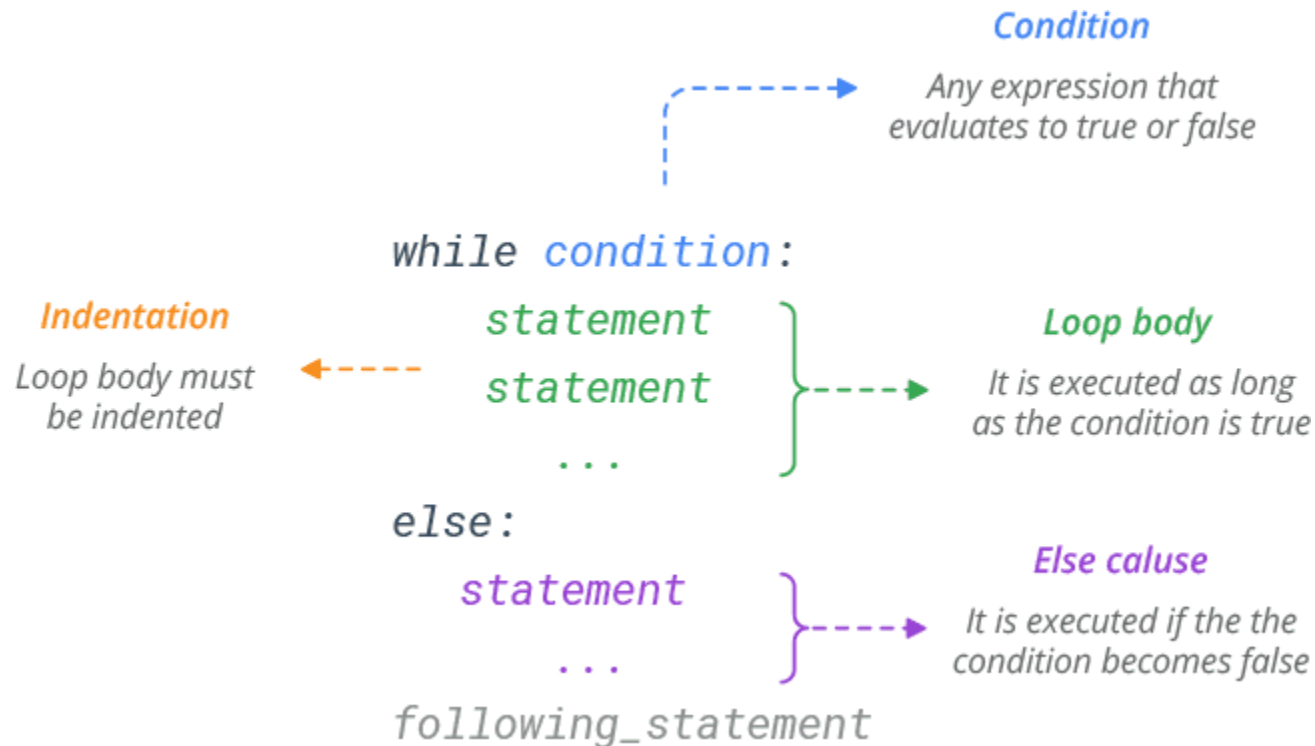
# While loop

- With the **while** loop we can execute a set of statements as long as a condition is **true**.
- Syntax of while loop in Python

**while** expression:  
statement(s)



# While loop



# While loop

- Can execute a set of statements as long as a condition is true. Result?

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```



# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
        print(i)
```

```
    i += 1
```

**Output:** 1

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
        i += 1
```

**Output:** 1

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:** 1

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:** 1

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
        print(i)
```

```
    i += 1
```

**Output:** 1  
2

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:** 1  
2



# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**   1  
             2

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**   1  
             2

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
        print(i)
```

```
    i += 1
```

**Output:**

```
1
2
3
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3
```

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
        print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3  
4
```



# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

1

2

3

4

# While loop

```
flag = True
```

```
i = 1
```

```
while i < 6 and flag == True:
```

```
    if i == 4:
```

```
        flag = False
```

```
    print(i)
```

```
    i += 1
```

**Output:**

```
1  
2  
3  
4
```

# While loop

```
flag = True
i = 1
while i < 6 and flag == True:
    if i == 4:
        flag = False
    print(i)
    i += 1
```

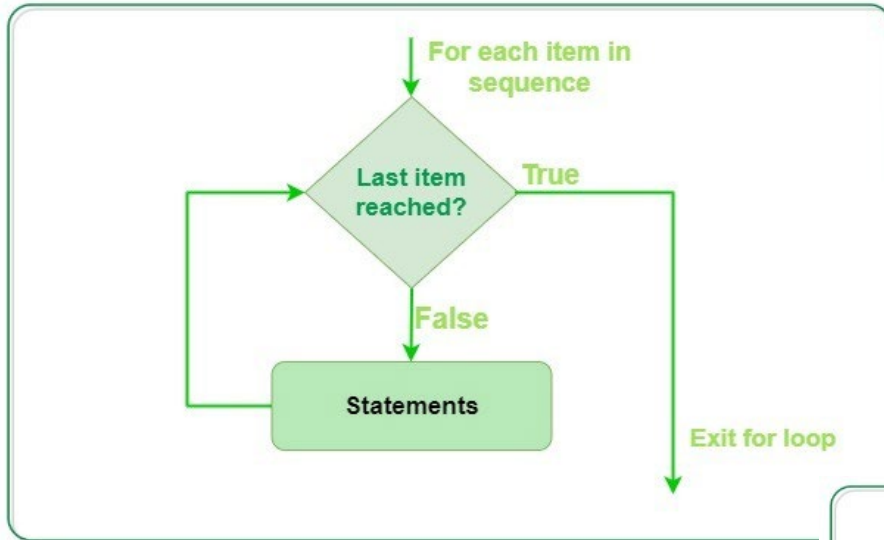
**Output:**

```
1
2
3
4
```

# For Loop v.s. While Loop

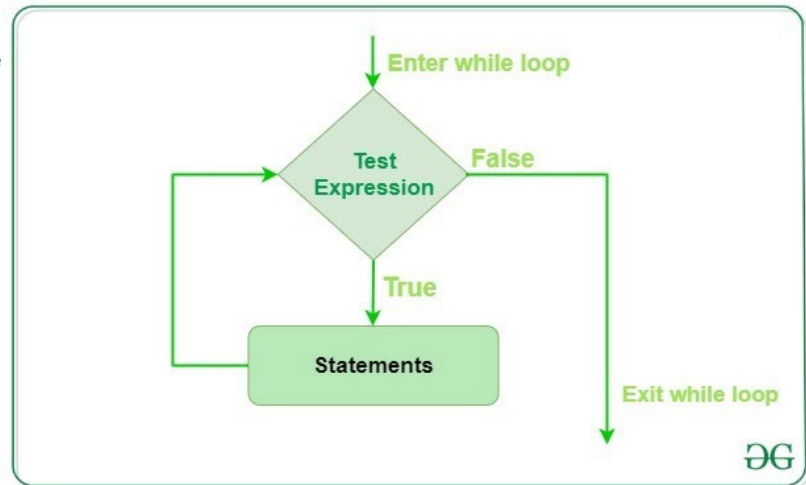
For loop	While loop
For loop is used to iterate over a sequence of items.	While loop is used to repeatedly execute a block of statements while a condition is true.
For loops are designed for iterating over a sequence of items. Eg. list, tuple, etc.	While loop is used when the number of iterations is not known in advance or when we want to repeat a block of code until a certain condition is met.
For loop require a sequence to iterate over.	While the loop requires an initial condition that is tested at the beginning of the loop.
For loop is typically used for iterating over a fixed sequence of items	While loop is used for more complex control flow situations.
For loop is more efficient than a while loop when iterating over sequences, since the number of iterations is predetermined and the loop can be optimized accordingly.	While a loop may be more efficient in certain situations where the condition being tested can be evaluated quickly.

# For Loop v.s. While Loop



for Loop

while Loop



# Exercise 1

# transform the following for loop into a while loop

# prints 0-19

```
for i in range(20):
```

```
    print ("i is now: ", i)
```

## Exercise 2

Write a Python script :

1. define a variable greeting = "With a cup of coffee in Hand, Many things to understand, Deep thoughts early this day, A very good morning for today!"
2. Print all the characters except "h" & "H"

# Exercise 3

Get a random number (i.e. use `random.randint(1500, 2500)` from the module `random` ) between 1500 and 2500 and print "All the digits are even" if that is the case.

Solve this using the following 2 approaches

1. if statements
2. loops

A couple of hints:

1. `# importing the module random`  
`import random`
2. `# obtaining a random number between 1500 and 2500`  
`n1 = random.randint(1500,2500)`
1. `# transforming the number into a string`  
`n1s = str( n1 )`
2. `# printing each character in the string`  
`for x in n1s:`  
 `print(x)`



# Exercise 4

The daily high temperature ( $^{\circ}\text{F}$ ) in Fairfield, CT and Denver, Colorado, during the month of January 2021 is given in the list below:

Fairfield=[33, 33, 18, 29, 40, 55, 19, 22, 32, 37, 58, 54, 51, 52, 45, 41, 45, 39, 36, 45, 33, 18, 19, 19, 28, 34, 44, 21, 23, 30, 39]

Denver=[39, 48, 61, 39, 14, 37, 43, 38, 46, 39, 55, 46, 46, 39, 54, 45, 52, 52, 62, 45, 62, 40, 25, 57, 60, 57, 20, 32, 50, 48, 28]

# Exercise 4

Write a python program in a script file that

1. uses relational and logical operations to determine (for each city):

- a. The number of days the temperature was above  $32^{\circ}$ .
- b. The number of days the temperature was between  $30^{\circ}$  and  $45^{\circ}$ .

2. determines and displays the following information:

- a. The average temperature for the month in each city.
- b. The number of days that the temperature was above the average in each city.
- c. The number of days that the temperature in Denver was higher than the temperature in Fairfield.

## Exercise 5

Write a Python program that prints all the possible combinations of a 3 digit code (i.e. 000, 001, 002, ... ).

# Exiting a statement

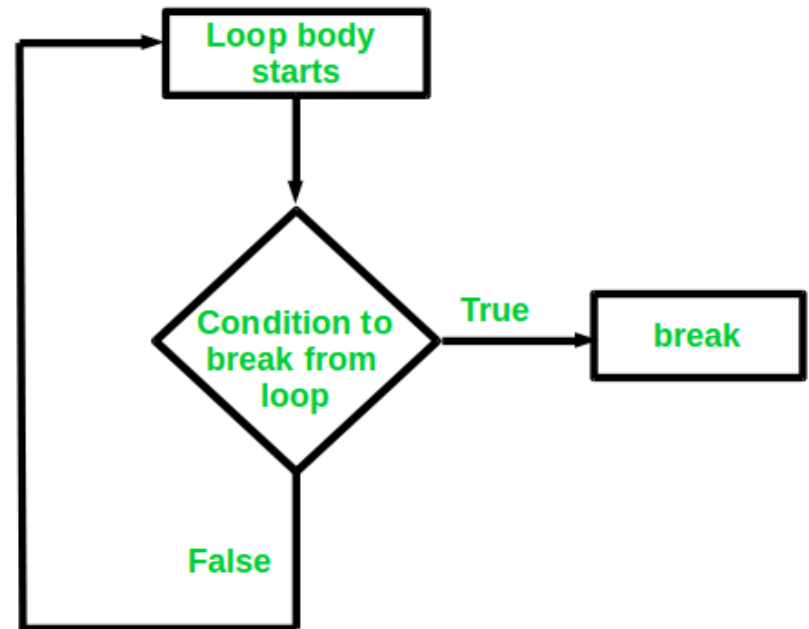
- **Break** keyword immediately terminates the loop
- **Continue** keyword immediately skips to the next iteration of the loop
- Return keyword will terminate a loop and its referenced function statement
- Exit commands:

<https://stackoverflow.com/questions/19747371/python-exit-commands-why-so-many-and-when-should-each-be-used/19747562>

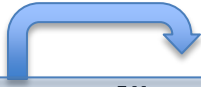
# Break Keyword

- **Break** keyword immediately terminates the loop

```
for name in ["Joe", "Amy", "Brad",  
"Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```



# Break Keyword



```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
```

```
    if name == "Zuki":
```

```
        print("Found it! ", name)
```

```
        break
```

# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

# Break Keyword



```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
```

```
    if name == "Zuki":
```

```
        print("Found it! ", name)
```

```
        break
```



# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

# Break Keyword



```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
```

```
    if name == "Zuki":
```

```
        print("Found it! ", name)
```

```
        break
```

# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

# Break Keyword



```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
```

```
    if name == "Zuki":
```

```
        print("Found it! ", name)
```

```
        break
```

# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

# Break Keyword



```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
```

```
    if name == "Zuki":
```

```
        print("Found it! ", name)
```

```
        break
```

# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

## Output:

```
found it Zuki
```



# Break Keyword

```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
    if name == "Zuki":  
        print("Found it! ", name)  
        break
```

## Output:

```
found it Zuki
```

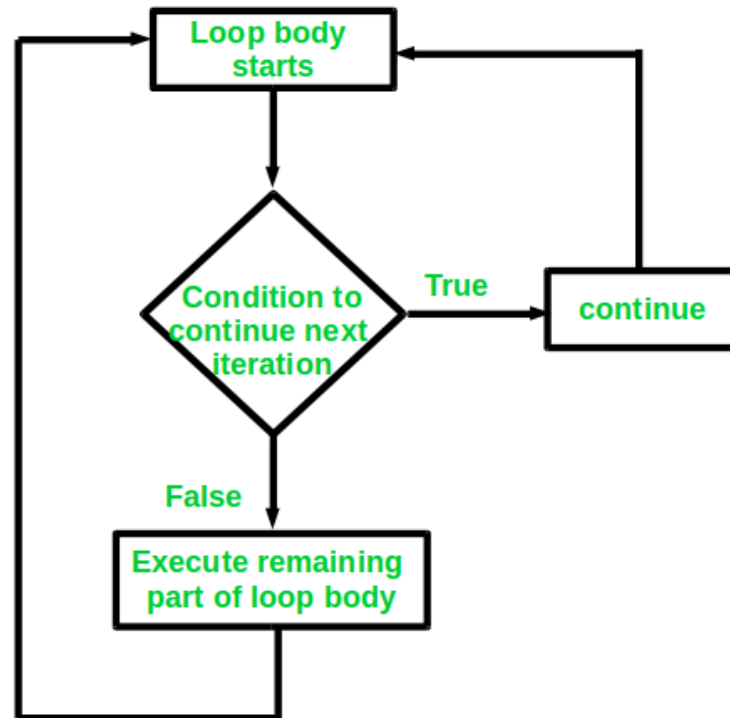
## Exercise 6

Write a Python script that validates a password (of your choice) entered by the user. The user has 3 attempts to enter the correct password.

# Continue Keyword

- **Continue** keyword immediately skips to the next iteration of the loop

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```



# Continue Keyword

```
i = 1
```

```
while i < 6:
```

```
    if i == 3:
```

```
        i = i + 1
```

```
        continue
```

```
    i = i + 1
```

```
    print(i)
```

# Continue Keyword

```
i = 1
```

```
while i < 6:
```

```
    if i == 3:
```

```
        i = i + 1
```

```
        continue
```

```
    i = i + 1
```

```
    print(i)
```

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2



# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3



# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3  
5

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3  
5

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3  
5

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3  
5

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:**

```
2
3
5
6
```



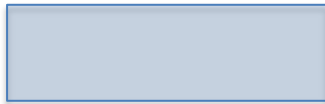
# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```

**Output:** 2  
3  
5  
6

# Continue Keyword

```
i = 1
while i < 6:
    if i == 3:
        i = i + 1
        continue
    i = i + 1
    print(i)
```



**Output:** 2  
3  
5  
6

# Exercise 6 Password Verification System

- Write a Python program that simulates a basic password verification system. The user should have three attempts to correctly guess a predefined password.
  - The program should store the correct password in a variable called `password`.
  - The user should be prompted to enter the password up to **three** times.
  - If the user enters the correct password, the program should print a success message and exit.
  - If the user fails to enter the correct password after three attempts, the program should print a failure message indicating they have used all their attempts.
  - At the end of the program, print "`over`" to indicate the process is complete.
  - Hint: exits the loop using **break**

# Important Links

**Python Documentation:**

<https://docs.python.org/3/index.html>

**Range():**

<https://docs.python.org/3/library/stdtypes.html#range>

# **CPSC 1101**

# **Introduction to Computing**

School of Engineering & Computing  
Dept. of Computer Science & Engineering  
Dr. Sidike Paheding