SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya Institute of Management

Department of Data Science and Technology

Practical No: 07

Subject: Python Programming Lab

MCA / Sem I / Python Programming [Course Code : 217P09L102]

ROLL No: 09 **DATE:18/09/2023**

FULL NAME: Atharv Ankush Desai

Aim:	**Understanding Python Loops break and continue statements**
Topics Covered:	continue,break,isdigit(),isalnum(),isalpha()
Problem Statement:	1. WAPP to find the first even number and print. 2. WAPP to skip printing numbers divisible by 3 within a for loop. 3. WAPP to terminate a while loop when the user enters a specific input (e.g.,"exit"). 4. WAPP to skip specific values (e.g., negative numbers) when iterating through a list. 5. WAPP to find and print all prime numbers from 1 to 50. Use a for loop and the break statement to optimize the search for prime numbers. 6. WAPP that takes a list of integers and prints all positive numbers, skipping negative ones using the continue statement. 7. WAPP that asks the user to enter a password. If the password contains at least one uppercase letter, one lowercase letter, one digit, and is at least 8 characters long, print "Password accepted." Otherwise, print an appropriate message and continue asking for a password until a valid one is entered. 8. WAPP that takes a list of numbers and a sum limit. Print the numbers in the list one by one until their sum exceeds the limit, then terminate the loop using the break statement. 9. WAPP that takes a list of numbers and prints unique values (skipping duplicates) using the continue statement.

Theory:

1. `break`:

```
for variable in iterable:
    if condition:
        break
```

2. `continue`:

```
for variable in iterable:
    if condition:
        continue
```

All of them return Boolean Value

3. `isdigit()`:

```
string_variable.isdigit()
```

4. `isalnum()`:

```
string_variable.isalnum()
```

5. `isalpha()`:

```
string_variable.isalpha()
```

Code:

1.

```
for i in range(1,100):
    if i % 2 == 0:
        print(f'{i} is the First Even Number')
        break
```

2.

```
for i in range(1,20):
    if i % 3 != 0:
        print(f'{i} is not divisible by 3')
```

3.

```
while True:
        myInput = input("(type 'Exit or exit' to Terminate) Enter Your
Input => ")
        if myInput.lower() in ["exit", "quit"]:
            break
        print(f"Your Input is => {myInput}")
```

```
print(f"Successfully terminated...")
```

4.

```
import random,math
myList = [math.floor(random.randint(-10, 20) * i) for i in range(11)]
print(f"{myList}")

for item in myList:
    if item > 0:
        continue
    print(f"{item} is less than zero")
```

5.

```
myList = [i for i in range(1,10)]

def isprime(num):
    # if 2 ot 1 case
    # base for any num
    if num == 1:
        return False
    else:
        for i in range(2,num + 1):
            if num % i == 0:
                break
        if i == num:
            return True
    return False

for i in myList:
    if isprime(i):
        print(f'{i} is prime')

print('\n---------------------\n')

for num in range(1,20):
    if num == 1:
        continue
    else:
        for i in range(2,num + 1):
            if num % i == 0:
                break
        if i == num:
            print(f'PRIME {num}')
```

6.

```
import random, math

myList = [math.floor(random.randint(-10, 20) * i) for i in range(11)]
print(f"{myList}")

for item in myList:
    if item < 0:
        continue
    print(f"{item} is less greater than zero")
```

7.

```
while True:
    myPass = input(f"Enter Password For Validation => ")
    if len(myPass) < 8:
        print(f'Weak Password')
    elif not any(char.isdigit() for char in myPass):
        print(f'Morderate Password')
    elif len(myPass) >=12 and any(char.isdigit() or not char.isalpha()
for char in myPass):
        print(f'Very Strong Password')
        break
    else:
        print(f'Strong Password {myPass}')
        break
```

8.

```
import random, math

# random.seed(69420)

myList = [math.floor(random.randint(10, 30) + i) for i in range(100)][2:]

maxLimit,counter = 200,int()

print(f"{myList}\n")

for item in myList:
    counter += item #106 206
    if counter > maxLimit : # 206 <200
        break
    print(f"Total => {counter} not hitting max Limit of {maxLimit}")
```

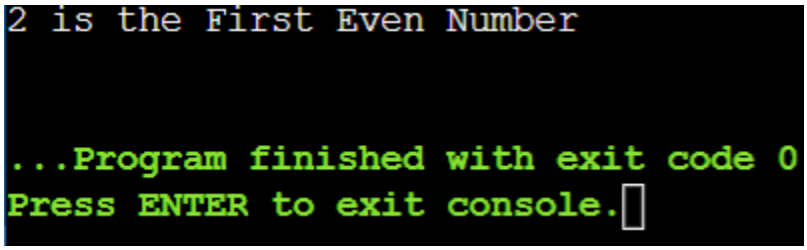
9.

```
import random, math

# random.seed(69420)
```

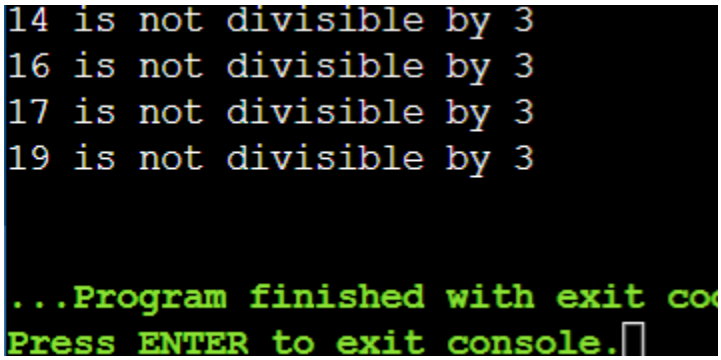
```
myList = [math.floor(random.randint(10, 30) + i) for i in range(100)]
print(myList)
myuniqueItem = []

for i in myList:
    if i in myuniqueItem:
        continue
    myuniqueItem.append(i)

print(f'Unique => {myuniqueItem}')
```

Screenshot of Output:

1.



2.



3.

```
(type 'Exit or exit' to Terminate) Enter Your Input => Lo
Your Input is => Lo
(type 'Exit or exit' to Terminate) Enter Your Input => LOl
Your Input is => LOl
(type 'Exit or exit' to Terminate) Enter Your Input => exit
Successfully terminated...


...Program finished with exit code 0
Press ENTER to exit console.
```

4.

```
[0, -3, -2, 36, -4, -30, -24, -35, 160, 18, -70]
0 is less than zero
-3 is less than zero
-2 is less than zero
-4 is less than zero
-30 is less than zero
-24 is less than zero
-35 is less than zero
-70 is less than zero


...Program finished with exit code 0
Press ENTER to exit console.
```

5.

```
2 is prime
3 is prime
5 is prime
7 is prime


...Program finished with exit code 0
Press ENTER to exit console.
```

6.

```
[0, 18, -12, -18, -24, -10, 30, 112, -56, -27, 130]
0 is less greater than zero
18 is less greater than zero
30 is less greater than zero
112 is less greater than zero
130 is less greater than zero


...Program finished with exit code 0
Press ENTER to exit console.
```

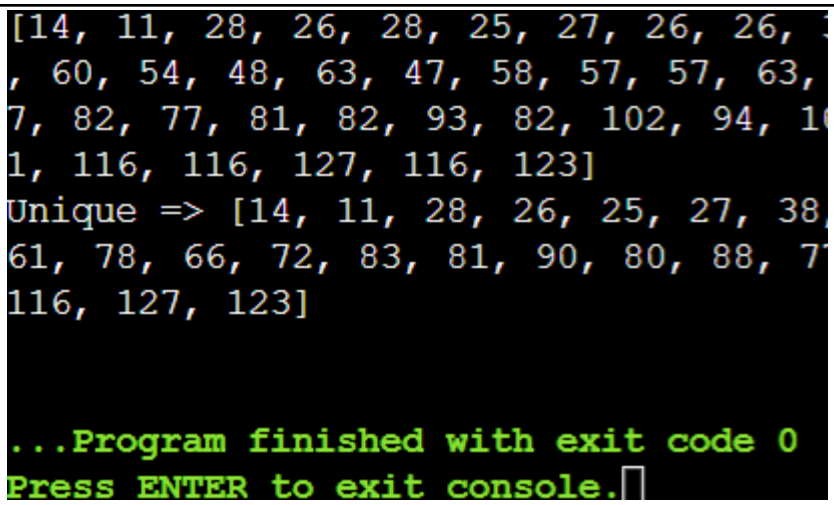
7.

```
Enter Password For Validation => Gokussj@12345
Very Strong Password


...Program finished with exit code 0
Press ENTER to exit console.
```

8.

```
, 64, 62, 56, 55, 62, 64, 70, 59, 60, 61, 55, 75, 69, 62, 5
0, 82, 84, 85, 94, 90, 86, 97, 85, 92, 94, 108, 100, 92, 10
8, 118, 118, 127]

Total => 32 not hitting max Limit of 200
Total => 54 not hitting max Limit of 200
Total => 78 not hitting max Limit of 200
Total => 99 not hitting max Limit of 200
Total => 115 not hitting max Limit of 200
Total => 140 not hitting max Limit of 200
Total => 178 not hitting max Limit of 200
```

9.

	
Observations:	- `break` in Python: Terminates the current loop execution when a specific condition is met. - `continue` in Python: Skips the current iteration of a loop and proceeds to the next iteration. - `break` is used to exit loops prematurely based on certain conditions, enhancing control flow. - `continue` is employed to skip specific iterations within a loop without terminating the entire loop.
Conclusion:	- In summary, `break` in Python serves to prematurely exit a loop when a particular condition is met, while `continue` skips the current iteration and proceeds to the next. - These two keywords enhance loop control and efficiency, providing valuable tools for Python programmers.

Subject-In-Charge:

Sign: _____

Prof. Mayura Nagar