Date
Page

Break & Continue in Loops

Break :

eg: 1

```
for number in range(10):
    if number == 5:
        break
    print(number)
print("Loop ended")
```

output:
0
1
2
3
4
Loop Ended

eg: 2

```
numbers = [10, 20, 30, 40, 50, 60]

for num in numbers:
    if num == 40:
        print(f"found {num}, stopping the search")
        break
    print(f"{num} is not 40")
```

Output:

10 is not 40
20 is not 40
30 is not 40
found 40, stopping the search

break with else in a for loop:

eg:3

```
numbers = [1, 2, 3, 4, 5]

for num in numbers:
    if num == 10:
        print("found 10!")
        break
else:
    print("10 not found in the list.")
```

Output:

10 not found in the list.

Continue Statement:

In Python, the continue statement is allowed to be used with a for loop. Inside the for loop, you should include an if statement to check for a specific condition. If the condition becomes TRUE, the continue statement will skip the current iteration and proceed with the next iteration in the loop.

eg: 2

```
for letter in 'Python':
    if letter == 'h':
        continue
    print('Current Letter :', letter)
print("good bye!")
```

Output:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : o
Current Letter : n
good bye!
```

eg: 2 : Checking Prime factors

```
num = 60
print("Prime factors for:", num)
d = 2
while num > 1:
    if num % d == 0:
        print(d)
        num = num/d
        continue
    d = d + 1
```

Output:

```
Prime factors for: 75
3
5
5
```

classmate
Date
Page

Nested Loop:

In Python, when you write one or more loops within a loop statement that is known as a nested loop.

The main loop is considered as outer loop and loop(s) inside the outer loop are known as inner loops.

eg: 1

```
months = ["jan", "feb", "mar"]
days = ["sun", "mon", "tue"]
for x in months:
    for y in days:
        print(x, y)
print("good bye!")
```

eg: 2

```
for i in range(3):
    for j in range(2):
        print(f"i = {i}, j = {j}")
```

Output:

```
i = 0, j = 0
i = 0, j = 1
i = 1, j = 0
i = 1, j = 1
i = 2, j = 0
i = 2, j = 1
```

eg: 3 Nested for loop to print a multiplication table.

```
for i in range(1, 4):
    for j in range(1, 5):
        print(f"{i} * {j} = {i * j}")
```

Output:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

eg:4 To print both table of 3 and 4 using nested 4 loops.

```
for num in range(3,5):
    print(f"\n Table of {num}:")
    for i in range(1,11):
        print(f"{num} x {i} = {num*i}")
```

Output:

```
Table of 3:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Table of 4:

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
```