Loop manipulation using pass, continue, break and else

Week-4 , Lecture-1

The *break* Statement

- ***break*** statement is used to terminate a loop or bring the control out of a loop when some external condition is triggered.

- ***break*** statement is generally used with ***while*** and ***for*** loop.

- ***if*** statement is used to provide the condition on which break will terminate the loop.

break Example

```
i=1
while i<=10:
    print (i)
    if i==5:
        break
    i=i+1
```

```
s = input("Enter a string: ")
# Using for loop
for letter in s:

    print(letter)

    if letter == 'a' or letter == 'i':
        break

print("Out of for loop")
```

The *continue* Statement

- The ***continue*** statement unconditionally allows the control to jumps to the beginning of the *loop* for the *next iteration.*
- This is just the opposite of the ***break*** statement.

- ***continue*** statement is also generally used with ***while*** and ***for*** loop.

```
i=0
while i<=10:
    i=i+1
    if(i==5):
        continue
    print (i)
```

```
for letter in 'Python‘:
    if letter == 'h':
        continue
    print (letter)
```

The ***pass*** Statement

- It is just a no operation statement.
- You can place a pass statement in the code where you may write the actual set of code latter on.

```
for letter in 'Python':
    if letter == 'h':
        pass
        print ("This is pass block")
    print (" Letter :", letter)
```

The ***else*** Statement

- Python supports the use of ***else*** statement with the ***for*** and ***while*** loop.

- The ***else*** statement when used with ***for***, is executed at the termination of the ***for loop***.
- The ***else*** statement when used with ***while***, is executed when the condition becomes false.

```
for letter in "Python":
    print(letter)
else:
    print("Complete")
```

```
count = 0
while (count > 1):
    count = count+1
    print(count)
    break
else:
    print("No Break")
```

Thank you