

Day12_Unconditional Statements

These are the kind of statements which make the control to jump to the different parts of the program without any specific condition but by using a simple keyword

The various unconditional statements are

1. break
2. continue
3. pass
4. return
5. exit
6. assert

break : It is a type of unconditional statement where the control jumps out of the program or any loop depending on the specific criteria

Ex:

```
lst = [5,-9,6,0,2,-3,1,0,7,4]
```

```
for i in lst:
```

```
    if i==0:
```

```
        break
```

```
    print(i)
```

Ex: break in nested loops

When we are using break in any of the inner loops it will get broken but not the whole program, the outer loop continue its flow.

```
for i in range(1,4):
    for j in range(4,7):
        if j==5:
            break
        print(i,":",j)
```

1: 4 ✓
2: 4 ✓
3: 4 ✓

for i in range(1,4):
 for j in range(4,7):
 if i==2:
 break
 print(i,":",j)

1: 4 ✓
1: 5 ✓
1: 6
3: 4 ✓
3: 5 ✓
3: 6 ✓

Handwritten notes and diagrams:
 - `range(1,4)` is exclusive, values 1, 2, 3.
 - `range(5)` values 0, 1, 2, 3, 4.
 - `range(1,10,3)` values 1, 4, 7, 10.
 - Diagrams showing nested loops with 'break' stopping the inner loop but allowing the outer loop to continue.

continue : It is similar to a break but instead of breaking the sequence it makes the control continue the flow only by skipping the specified criterion

```
lst = [5,-9,6,0,2,-3,1,0,7,4]
```

```
for i in lst:
```

```
    if i==0:
```

```
        continue
```

```
    print(i)
```

```
5
-9
6
2
-3
1
7
4
```

pass : It is used for bypassing the control without performing any break or skip.

```

lst = [5,-9,6,0,2,-3,1,0,7,4]
for i in lst:
    if i<=0:
        pass
    elif i>0:
        print(i)

```

```

5
6
2
1
7
4

```

In python programming we also have else block for the while and for statements

while..else : In this we can have an else followed by while and it will get executed after all the loop finishes the execution, the else in the while fails to get executed if we have a break in the while

Syntax:
While ..
 code
else:
 statements

```

i = 1
while i<=5:
    print(i)
    i=i+1
else:
    print("While is finished")
1
2
3
4
5
While is finished

```

```

i = 1
while i<1:
    print(i)
    i=i+1
else:
    print("While is finished")

```

#While is finished

```

i = 1
while i<=5:
    if i==2:
        break
    print(i)
    i=i+1
else:
    print("While is finished")

```

```

1

```

```

i = 0
while i<=5:
    i=i+1
    if i==3:
        continue
    print(i)

```

```
else:  
    print("While is finished")
```

for..else : In this we can have an else followed by for and it will get executed after all the loop finishes the execution, the else in the for fails to get executed if we have a break in the for

Syntax:

```
for loop:  
    statements  
else:  
    statements
```

Example:

```
for i in range(1,11) :  
    print(i)  
else:  
    print("End of for")
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
End of for
```

```
for i in range(1,11) :  
    if i==4:  
        break  
    print(i)  
else:  
    print("End of for")
```

```
1  
2  
3
```

```
for i in range(1,11) :  
    if i==4:  
        continue  
    print(i)  
else:  
    print("End of for")
```

```
2  
3  
5  
6  
7  
8  
9  
10  
End of for
```

