LOOPS in PYTHON

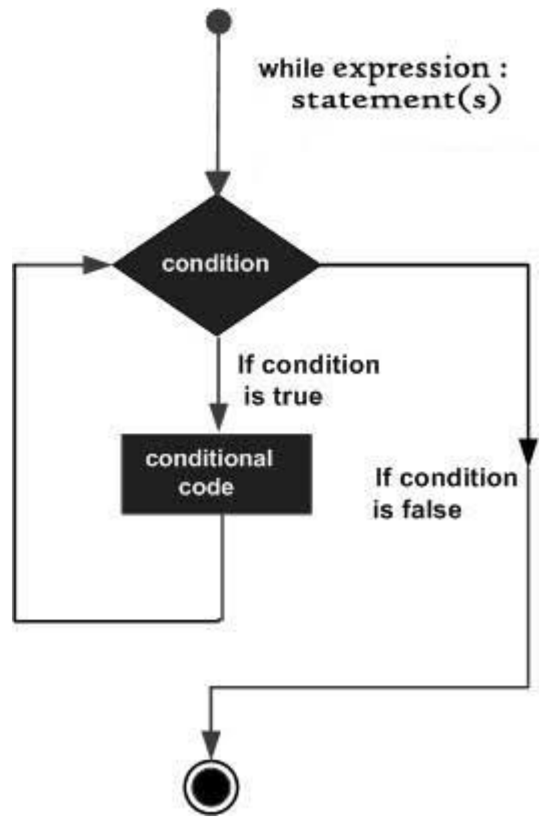
softserve

Agenda

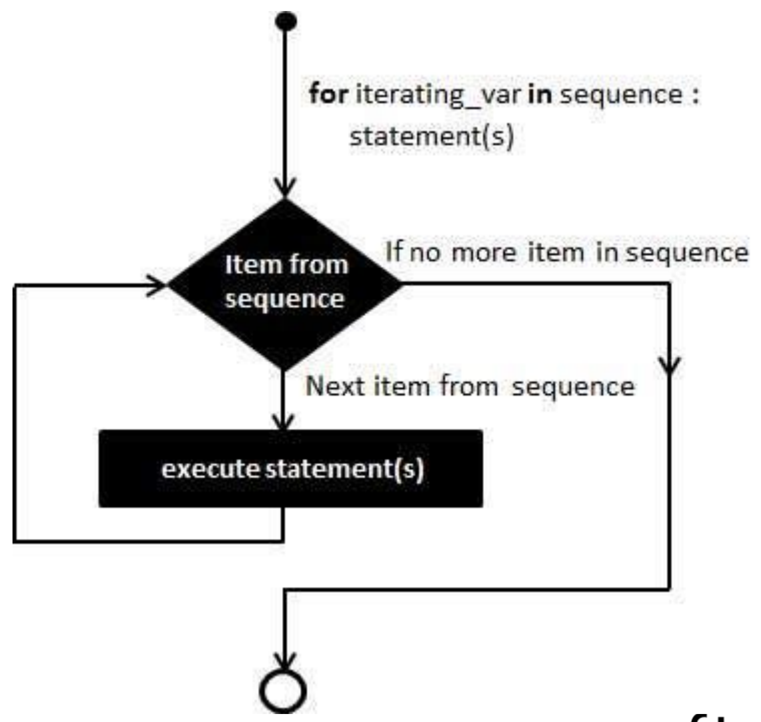
- while loop
- for loop

Flowcharts

while Loop Statements



for Loop Statements



softserve

while loop

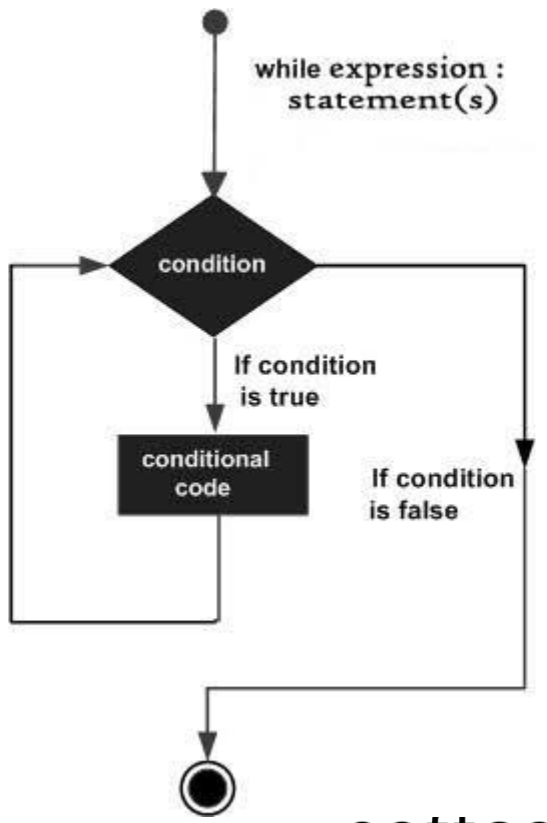
while expression **:**

    suite

else:

    suite

```
start = 0
finish = 10
while start < finish:
    print(start)
    start += 1
else:
    print (“The end”)
```



softserve

for loop

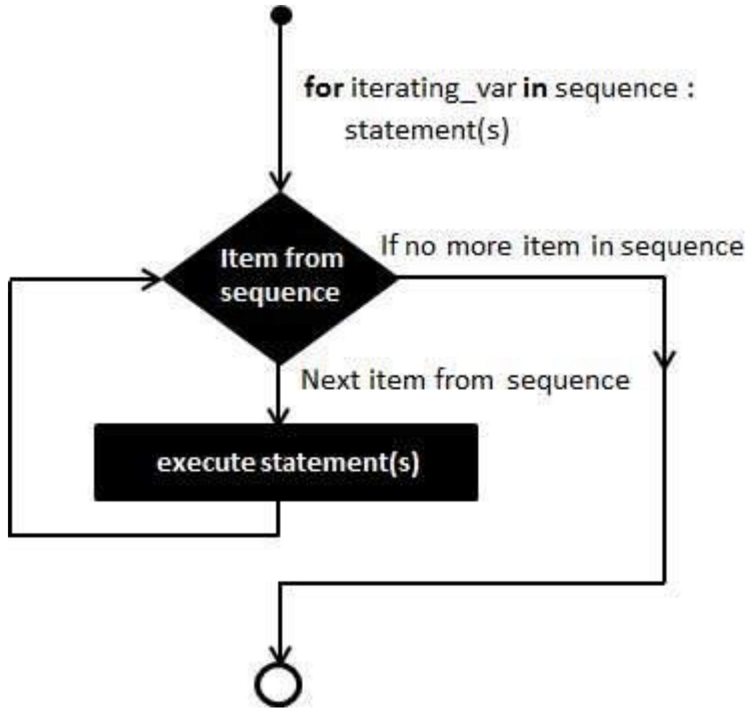
for target_list **in** expression_list **:**

    suite

else:

    suite

```
for j in [0, 1, 2, 3, 4]:
    print(j)
else:
    print(j, “- is the last”)
```



softserve

range

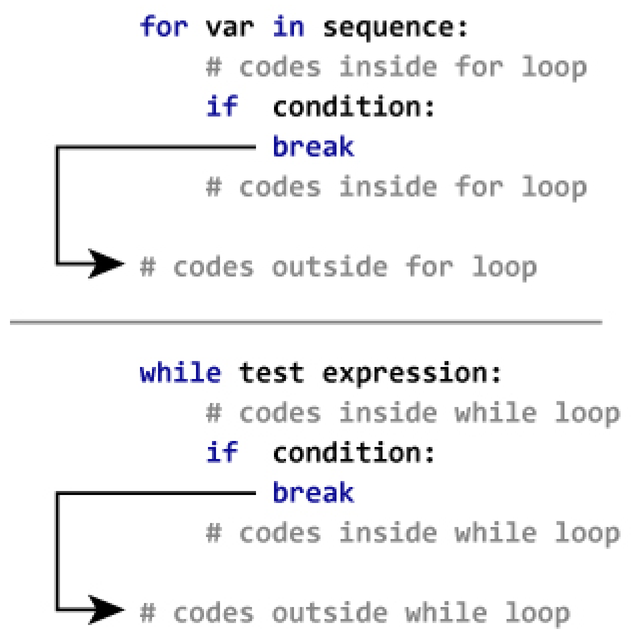
- range(stop) -> list of integers
- range(start, stop) -> list of integers
- range(start, stop, step) -> list of integers

```
print([i for i in range(10)])
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print([i for i in range(5,10)])
[5, 6, 7, 8, 9]
print([i for i in range(0,10,2)])
[0, 2, 4, 6, 8]
```

softserve

break and continue

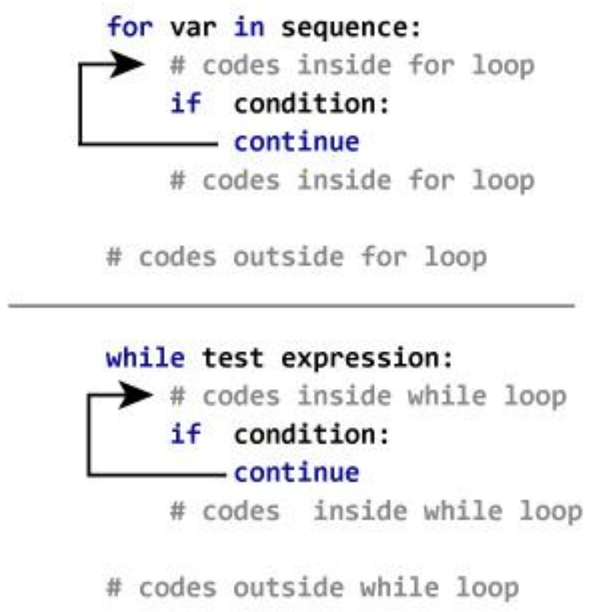
break



```
for var in sequence:
    # codes inside for loop
    if  condition:
        break
    # codes inside for loop

# codes outside for loop
```

```
while test expression:
    # codes inside while loop
    if  condition:
        break
    # codes inside while loop

# codes outside while loop
```

continue



```
for var in sequence:
    # codes inside for loop
    if  condition:
        continue
    # codes inside for loop

# codes outside for loop
```

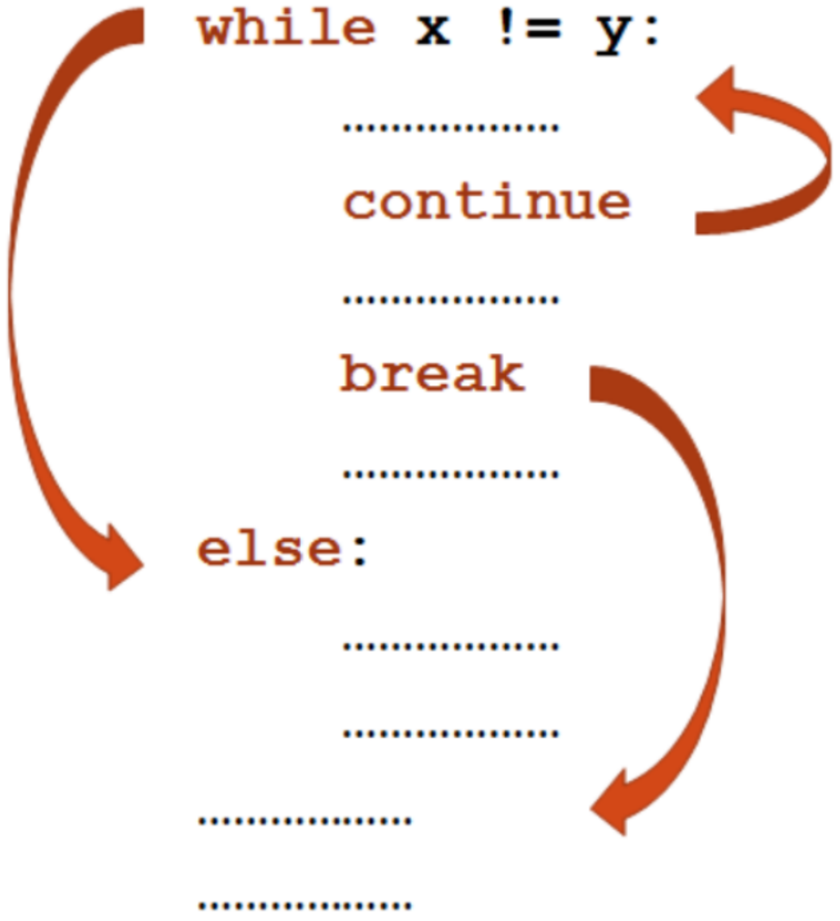
```
while test expression:
    # codes inside while loop
    if  condition:
        continue
    # codes  inside while loop

# codes outside while loop
```

```
for val in "string":
    if val == "i":
        break
    print(val)
print("The end")
```

```
for val in "string":
    if val == "i":
        continue
    print(val)
print("The end")
```

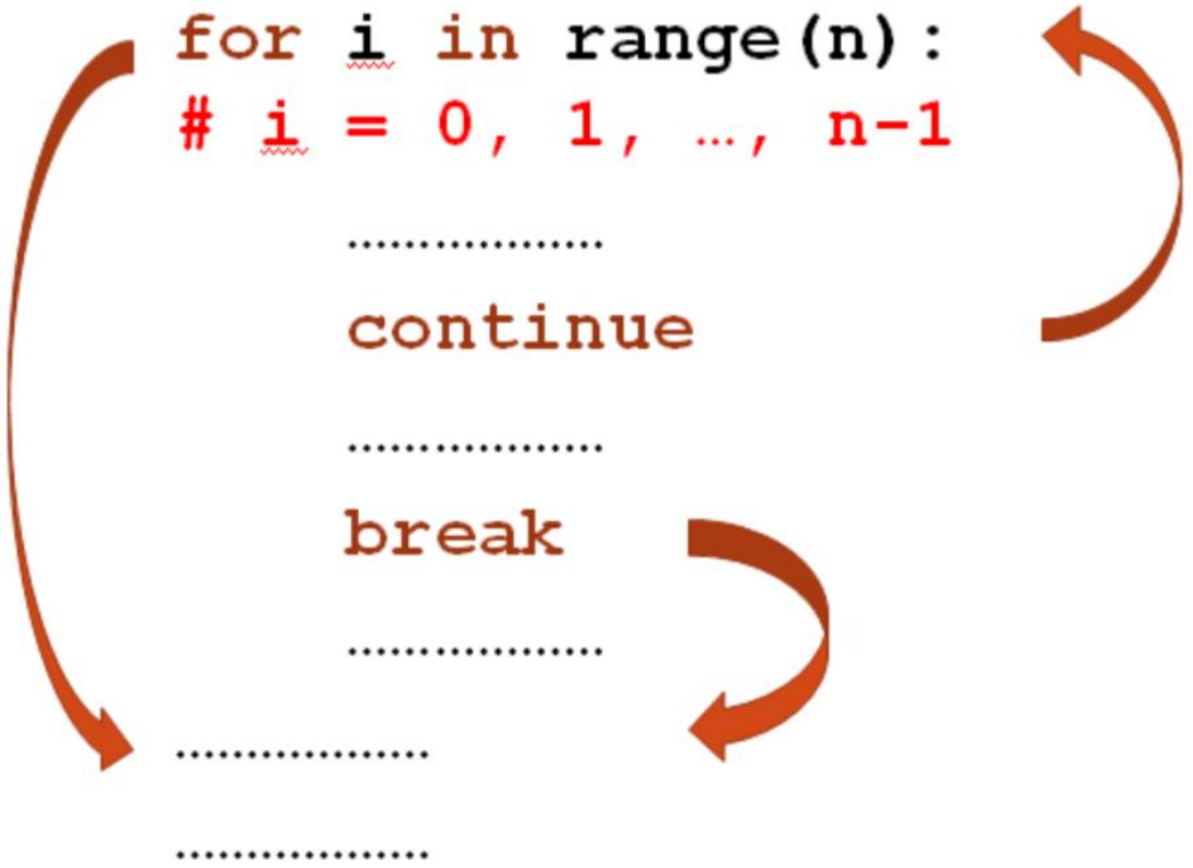
softserve

break and continue



```
while x != y:
          ..................
          continue
          ..................
          break
          ..................
else:
          ..................
          ..................
..................
..................
```

break and continue



```
for i in range(n):
# i = 0, 1, …, n-1
    ..................
    continue
    ..................
    break
    ..................
..................
..................
```

softserve

Python pass statement

pass - We generally use it as a placeholder.

Suppose we have a loop or a function that is not implemented yet, but we want to implement it in the future. They cannot have an empty body. The interpreter would complain. So, we use the pass statement to construct a body that does nothing.

```
# pass is just a placeholder for
# functionality to be added later.

sequence = {'p', 'a', 's', 's'}
for val in sequence:
    pass
```

We can do the same thing in an empty block of if statement or function or class as well.

NOW
THE
FUTURE
softserve