CS 30 Discussion

Week 9

for loop

For loops are used for sequential traversal.

Syntax:

```
for iterator_var in sequence:
    statements(s)
```

It can be used to iterate over iterators and a range.

```
# Iterating over a list
print("List Iteration")
l = ["cs", "30", "exciting"]
for i in l:
    print(i)
```

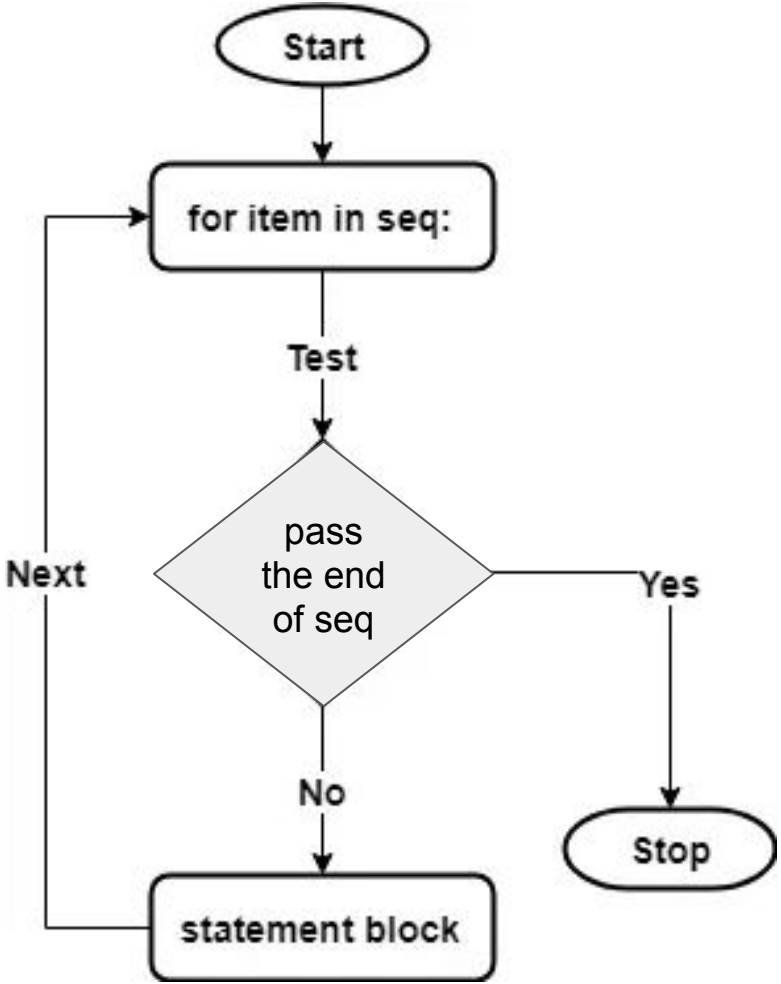
```
List Iteration
cs
30
exciting
```

```
# Iterating over a String
print("String Iteration")
s = "Geeks"
for i in s :
    print(i)
```

```
String Iteration
G
e
e
k
S
```

```
# Iterating by index
print("List Iteration")
l = ["cs", "30", "exciting"]
for index in range(len(l)):
    print(l[index])
```

```
List Iteration
cs
30
exciting
```


Start
for item in seq:
Test
pass
the end
of seq
Next
Yes
No
Stop
statement block

for loop practice

```
Replaces all negative numbers in list l with 0, leaving all
other numbers unchanged.
    >>> zeroNegatives([1,3,-4,-6,5,-5])
    [1, 3, 0, 0, 5, 0]

Use for loop to check whether a string is a palindrome.
    >>> palindrome('dogeeseseegod')
    True
```

for loop solution

```
def zeroNegatives(l):
    for i in range(len(l)):
        if l[i] < 0:
            l[i] = 0
    return l

def palindrome(s):
    for i in range(len(s)//2):
        if s[i] != s[len(s)-1-i]:
            return False
    return True
```

for loop practice

```
return nth fibonacci number: 1, 1, 2, 3, 5, 8, ...
def fibonacci(n):
    return

>> fibonacci(5)
5
```

for loop solution

```
def fibonacci(n):
    a = 0
    b = 1
    for i in range(0, n):
        temp = a
        a = b
        b = temp + b
    return a

# Display the first 15 Fibonacci numbers.
for c in range(0, 15):
    print(fibonacci(c))
```

nested for loop

```
for iterating var in sequence:
   for iterating var in sequence:
      statements(s)
   statements(s)
```

Practice: print the following pattern

```
*
**
***
****
*****
```

```
def print_pattern(n):
    for i in range(1,n):
        acc = ''
        for j in range(i):
            acc += '*'
        print(acc)
```

```
def print_pattern(n):
    for i in range(1,n):
        acc = '*' * i
        print(acc)
```

while loop

In python, while loop is used to execute a block of statements **repeatedly until a given a condition is satisfied**. And when the condition becomes false, the line immediately after the loop in program is executed.

Syntax :

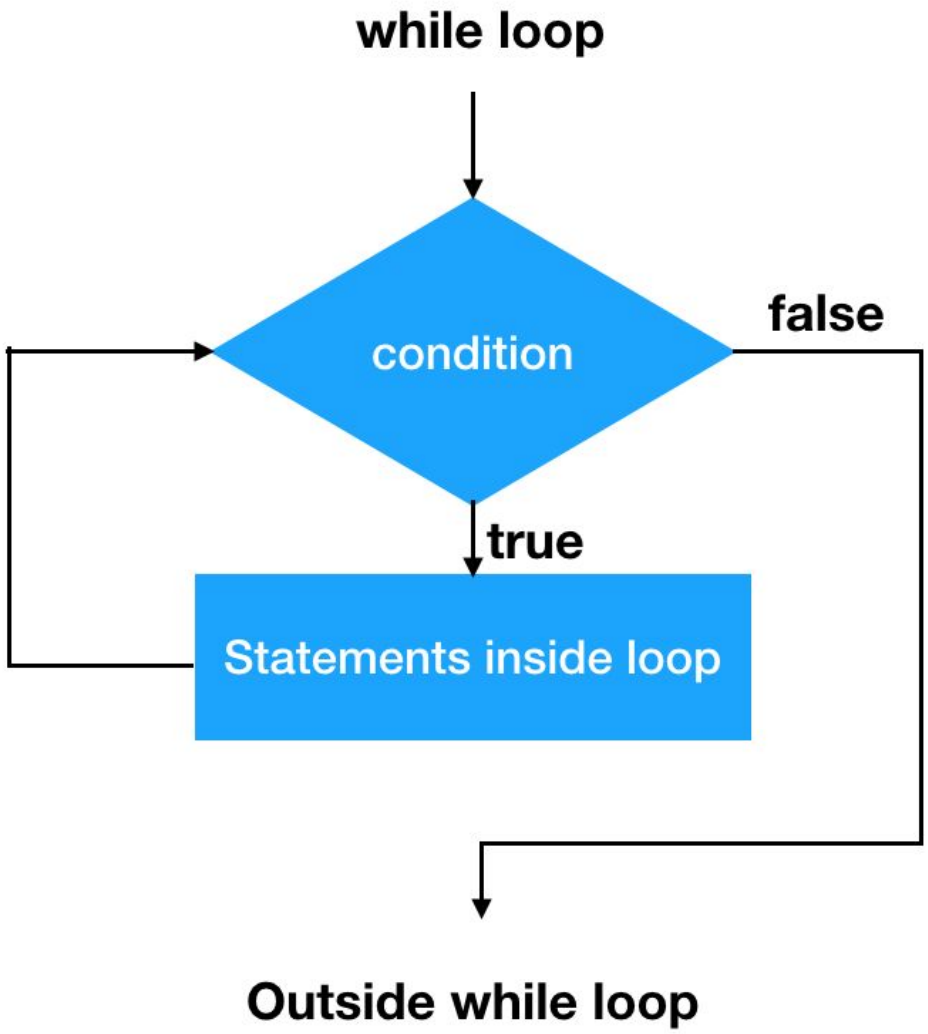
```
while expression:
    statement(s)
```

```
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("exciting CS30")
```

```
exciting CS30
exciting CS30
exciting CS30
```

```
i = 1
while i < 6:
  print(i)
  i += 1
```

```
1
2
3
4
5
```


while loop
condition
false
true
Statements inside loop
Outside while loop

while loop practice

```
Replaces all negative numbers in list l with 0, leaving all
other numbers unchanged.
    >>> zeroNegatives([1,3,-4,-6,5,-5])
    [1, 3, 0, 0, 5, 0]

Use for loop to check whether a string is a palindrome.
    >>> palindrome('dogeeseseegod')
    True
```

while loop solution

```
def zeroNegatives(l):
    i = 0
    while i < len(l):
        if l[i] < 0:
            l[i] = 0
        i += 1
    return l
```

```
def palindrome(s):
    i = 0
    while i <= len(s)//2:
        if s[i] != s[len(s)-1-i]:
            return False
        i += 1
    return True
```

while loop solution

```
def zeroNegatives(l):
    i = 0
    while i < len(l):
        if l[i] < 0:
            l[i] = 0
        i += 1
    return l
```

```
def palindrome(s):
    i = 0
    j = len(s)-1
    while i<j:
        if s[i] != s[j]:
            return False
        i += 1
        j -= 1
    return True
```

while loop practice

return n^{th} fibonacci number: 1, 1, 2, 3, 5, 8, ...

```
def fibonacci(n):
    return

>> fibonacci(5)
5
```

while loop solution

```
def fibonacci(n):
    a = 0
    b = 1
    while n > 0:
       temp = a
       a = b
       b = temp + b
       n -= 1
    return a
```

WorkSheet

Please work together on the problem 1-7(1).