浙江理工大学
ZHEJIANG SCI-TECH UNIVERSITY

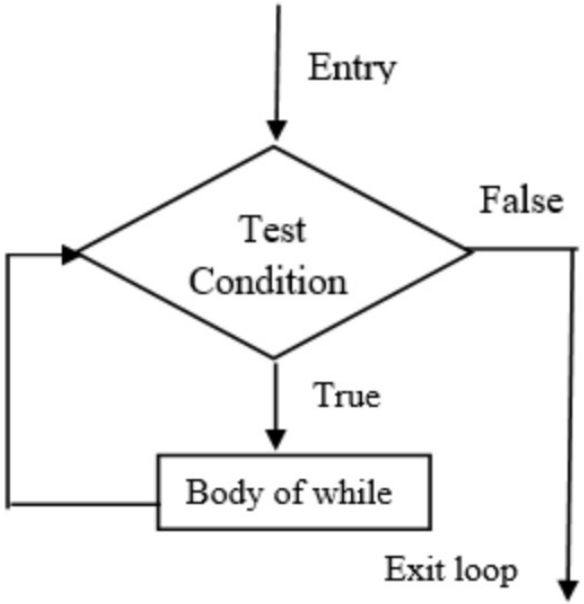
6. Control Flow (2)

《Python programming》 / Lecturer : Zhiyi Luo (罗志一)

School of Computer Science and Technology
计算机科学与技术学院

While loop

- The while loop statement repeatedly executes a code block while a particular condition is true.



Flow chart of while loop

While loop

- The while statement checks the condition. The condition must return a boolean value. Either True or False.
- Next, if the condition evaluates to true, the while statement executes the statements present inside its block.
- The while statement continues checking the condition in each iteration and keeps executing its block until the condition becomes false.

Syntax of while-loop

```
while condition:
    body of while loop
```

While loop

- In a while-loop, every time the condition is checked at the beginning of the loop, and if it is true, then the loop’s body gets executed. When the condition became False, the controller comes out of the block.

While loop

- Example to calculate the sum of first ten numbers

```
num = 10
sum = 0
i = 1
while i <= num:
    sum = sum + i
    i = i + 1
print("Sum of first 10 number is:", sum)
```

Infinite while loop

- If the code inside the while loop doesn’t modify the variables being tested in the loop condition, the loop will run forever.

```
# Infinite while loop
while True:
    print('Hello')
```

If-else in while loop

- Example: Print even and odd numbers between 1 to the entered number

```
n = int(input('Please Enter Number '))
while n > 0:
    # check even and odd
    if n % 2 == 0:
        print(n, 'is a even number')
    else:
        print(n, 'is a odd number')
    # decrease number by 1 in each iteration
    n = n - 1
```

Break in while loop

- Example: Write a while loop to display each character from a string and if a character is number then stop the loop.

```
name = 'Jesaa29Roy'
size = len(name)
i = 0
# iterate loop till the last character
while i < size:
    # break loop if current character is number
    if name[i].isdecimal():
        break;
    # print current character
    print(name[i], end=' ')
    i = i + 1
```

Continue in while loop

- Example: Write a while loop to display only alphabets from a string.

```
name = 'Jesaa29Roy'
size = len(name)
i = -1
# iterate loop till the last character
while i < size - 1:
    i = i + 1
    # skip while loop body if current character is not alphabet
    if not name[i].isalpha():
        continue
    # print current character
    print(name[i], end=' ')
```

Nested while loops

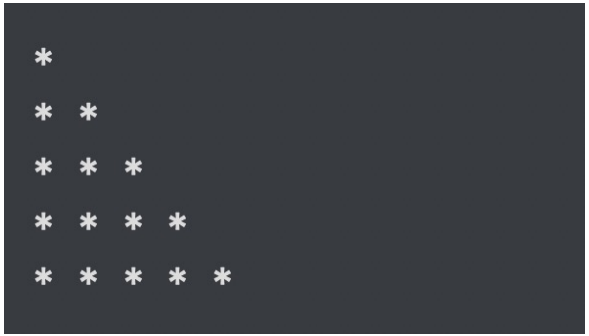
syntax

```
while expression:
    while expression:
        statemen(s) of inner loop

    statemen(s) of outer loop
```

Nested for loops

- Example: Nested while loop to print the following pattern




```
i = 1
# outer while loop
# 4 rows in pattern
while i < 5:
    j = 0
    # nested while loop
    while j < i:
        print('*', end=' ')
        j = j + 1
    # end of nested while loop
    # new line after each row
    print('')
    i = i + 1
```

While loop with else block

- We can use the else block in the while loop, which will be executed when the loop terminates normally.
- Else block will not execute when:
  - While loop terminate abruptly
  - The break statement is used to break the loop
- Defining the else block with a while loop is optional.

While loop with else block

```
i = 1
while i <= 5:
    print(i)
    i = i + 1
else:
    print("Done. while loop executed normally")
```

```
i = 1
while i <= 5:
    print(i)
    if i == 3:
        break
    i = i + 1
else:
    print("Done. while loop executed normally")
```

Exercise

- 以下代码的输出结果是？

```
for s in "Hangzhou, Zhe Jiang":
    if s == "i":
        break
    print(s, end="")
```

Exercise

- 以下代码的输出结果是？

```
namelist = ["any"]
namelist.extend(["mary", "susan"])
namelist.append(["mary", "susan"])
print(namelist[3])
```

Exercise

- 以下代码的输出结果是？

```
str1 = "k:1|k1:2|k2:3|k3:4"
str_list = str1.split('|')
d = {}
for L in str_list:
    k, v = L.split(':')
    d[k] = v
print(d)
```

Exercise

```
x = int(input())
y = 0
if x % 2 == 1:
    y = x - 3
else:
    y = x + 6

if x > 6:
    y = (y + 2) * 3 % 5
elif x < 3:
    y = -2

print(y)
```

(1) 用户输入-6，输出结果为?
(2) 用户输入5，输出结果为?
(3) 用户输入8，输出结果为?