程式設計概論
Programming 101
—Repetition structures (程式迴圈)

授課老師：邱淑怡

Outline

- Repetition structure
  - for loop (計數器迴圈)
    - A count-controlled loop iterates a specific number of times. In Python. you use the for statement to write a count-controlled loop.
  - while loop (條件式迴圈)
    - A condition-controlled loop causes a statement or set of statements to repeat as long as a condition is true.

Statement

- for loop
  - Introduce list
- range() function
- while loop
- break
- continue
- pass

list(串列)

- A *list* is an object that contains multiple data items.
- Each item that is stored in a list is called an *element*.
- Lists are dynamic data structures, meaning that items may be added to them or removed from them.
- Use indexing, slicing, and various methods to work with lists.
- Example: even_numbers=[2,4,6,8,10]

for loop

```
for num in [0,1,2,3,4]:
    print(num)

for name in [‘Jeff’, ‘Chuck’, ‘Mark’]:
    print(name)
```

range() function

- range(start, stop[, step]): the values are integer, either positive or negative.
  - start: the starting value(included), default value=0
  - stop: the ending value(excluded)
  - step: increment value(遞增值), default value=1

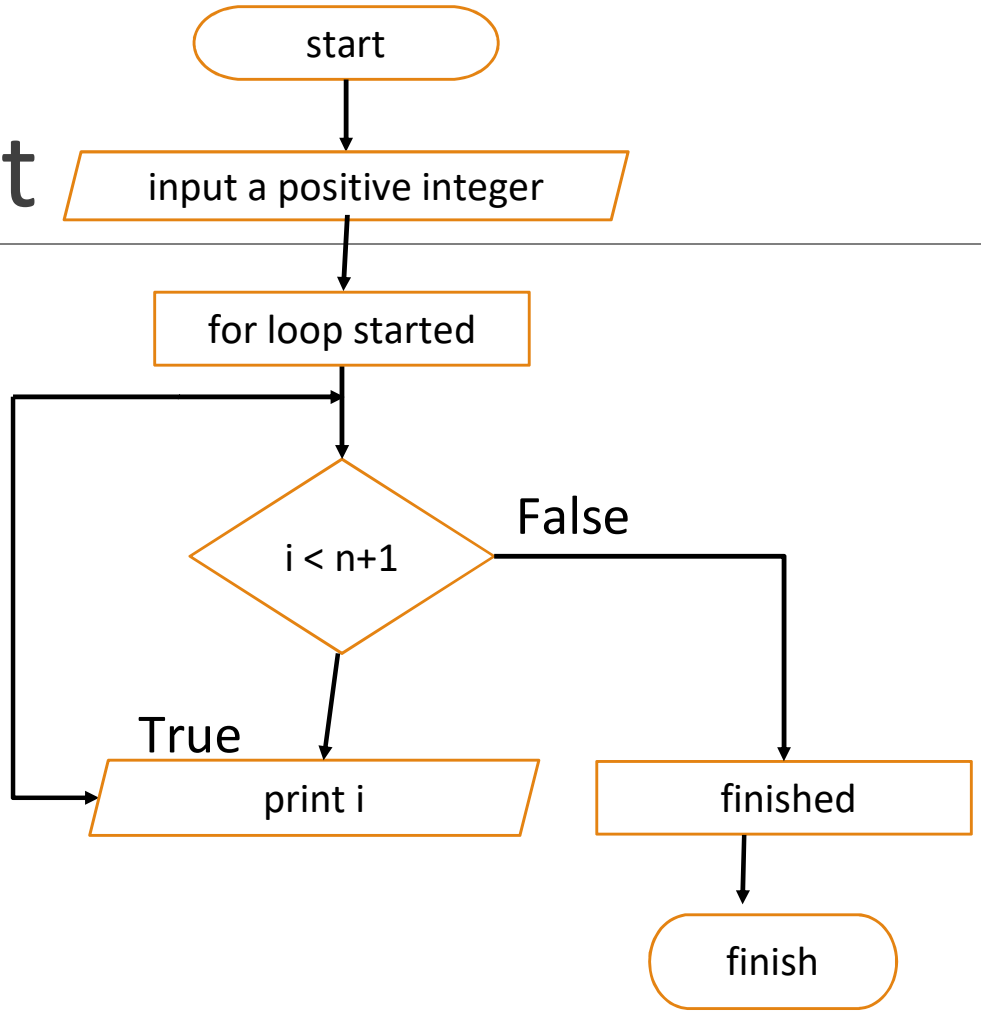
String and list share the same concept

```
r=range(5)
print(r)
print(list(r))
```

```
print(list(range(10)))  # range(0,10)
print(list(range(0,15,5)))   #step 5, [0,5,10]
print(list(range(0,10,3)))   # [0, 3, 6, 9]
print(list(range(0,-10,-2)))   # [0, -2, -4, -6, -8]
```

for loop structure

for loop flowchart



The structure of for loop

Use control variable to control the number of times of for loop executed

statements in for loop, determined by the value that controls variable execute repeatedly.

- for loop is count-controlled loop, and counter is the control_variable.
- for loop syntax:

for **count_variable** in list or any numbers or words:

statements

```
n=int(input("input a positive integer:"))
for i in range(1,n+1):
    print(i,end=' ')
print("for loop completed")
```

range(a,b) means (a,a+1,…,b-1)

Use: http://pythontutor.com/

for loop

can be regarded as index or counter

- statements in for loop, determined by the value that controls variable execute repeatedly.

for **count(variable)** in list or any numbers or words:

    statement 3   # more than one line is allowed

```
for index1 in range(1,10):
    print(index1)
print("for loop completed")
```

Python range() function syntax:
range(start, stop[, step])
Ex:

```
list(range(10))  # range(0,10)
list(range(0,15,5))   #step value 5, [0,5,10,15]
list(range(0,10,3))   # [0, 3, 6, 9]
list(range(0,-10,-2))   # [0, -2, -4, -6, -8, -10]
```

for loop(cont.)

Please write a Python program to add from 1 to 5.

```
Result=0
for i in range(1,6):
        Result=Result+i # Result+= i
        print(Result, i)
print("for loop completed, Result:", Result)
```

loop times	=result on the left	i	=result in the right
first time	0	1	0+1
second time	1	2	1+2 (3)
third time	3	3	3+3 (6)
fourth time	6	4	6+4 (10)
fifth time	10	5	10+5 (15)

for loop

Write Python program

- Q1: Calculate 1+2+3+…+100=?
- Q2: Calculate 2+4+6+8+…+94+96+98=?
- Q3: Find all the numbers that are divisible by 13 between 1 and 100, and print the result.

Repetition structure

can be regarded as index or counter

- Statements in the for loop are determined by the value that control_variable executes finally.

```
for j in list or any numbers or words:
        statement1
        statement2
for i in range(1,10):
    print(i)
print("for loop completed")
```

range() function can specify the change of index (i) in the loop

```
for i in range(1,20,3):     #i is the index
    print(i)
print("for loop completed 1/2")


for i in range(20,1,-2):
    print(i)
print("for loop completed 2/2")
```

Combine the data type”list”with for loop

- Each element in the list prints its value and corresponding index value
  - range(len(a)) = range(6) = range(0:6) ==range The function will return in order: 0,1,2,3,4,**5(6-1**)

```
a= [3,6,-7,1,-4,12]
for i in range(len(a)):
        print(i, a[i])
print(“list loop completed”)
```

Multiple for loops

```
for i in range(a,b):
    for j in range(c,d):
            print(i,j)
            print("inner loop")
        print("outer loop")
print("all loops are completed")
```

c, d value may be related to i

Multiple for loops: example

```
for i in range(2,6):
    for j in range(1,i):
        print(i,j)
        print("inner for loop")
    print("outer for loop")
print("all loops are completed")
```

Multiple for loops: example

```
for i in range(1,10):
    for j in range(1,10):
        s=  i*j
        print('%d * %d = %d ' %(i, j , s))
```

```
for i in range(1,10):
    for j in range(1,10):
        s=  i*j
        print('%d * %d = %d   ' %(i, j , s), end='')
    print('\n')
```

end=' 'means adding a space at the end of the line instead of a newline

Multiple for loops: example(cont.)

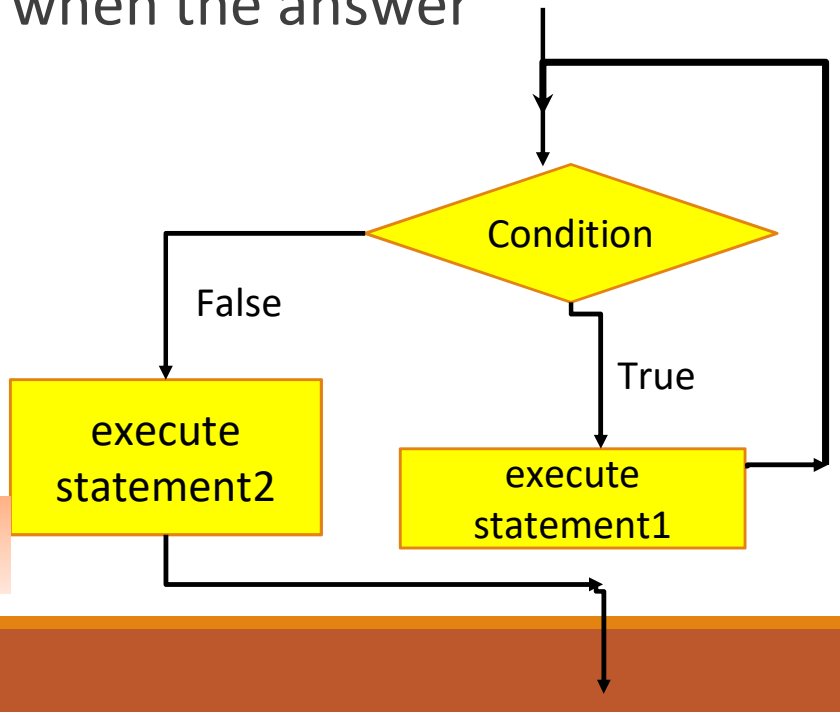
```
k=[1,2,3,4,5,6,7,8,9]   # k is a list
for i in k :
        for j in k:
                s= i*j
                print('%d * %d = %d ' %(i, j , s), end="")
        print('\n')   # newline
```

while loop

- The execution of the while loop is based on whether the condition is true. It's a condition-controlled loop.
- In the while loop, the most important thing is whether the condition changes or not. The loop ends when the answer changes from True to False.

```
while condition:
    statement1
[else:
    statement2]
```

Use http://pythontutor.com/



while loop_example

- While loop avoid infinite loop
- Use the while loop to print 0 1 2 3 4

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

Use http://pythontutor.com/

while…else

```
i=0
while i<5:
    print(i)
     i=i+1
else:
    print(“while else”)
```

break statement

- break
  - break can make you leave the loop. No matter where you are in the loop, it makes you leave the loop immediately and execute the next statements

```
while True:
        data = input('print any string. If you leave the loop, input stop:')
         if data.lower()=='stop':
                 break
        print(data)
print('Finished')
```

continue statement

- continue
  - continue don’t make a program leave, but also interrupt the loop. Continue will bring you back to the beginning of the loop and continue the loop again.

```
while True:
    data = input('print any string. You stay the loop but do not print string, input continue:')
    if data.lower() == ‘cont':
        print('在continue 內')
        continue
    print(data)
print("OK, stop it.")
```

pass statement

- Empty statement

```
for var in ‘Python code’:
        if var==‘ ’:
                pass
        else:
                print(var)
```

Multiple while loops_example

```
i=1   # Need to give i,j starting values
while i < 10:
        j=1
        while j < 10:
                s= i*j
                print ('%d * %d = %d ' %(i, j , s))
                j = j+1      # j +=1
        i=i+1   # i += 1
        print('\n')
```

Exercise_4
<< basic >>

Question 1: input three edges and determine if they can construct a triangle.

- let the user input three edges and determine if they can construct a triangle, if can't, let the user input again. repeat until they can.
- rules of constructing a triangle(either way)：
  - the sum of any two edges is always greater than the third one
  - (or)The difference between any two edges of the triangle is always less than the third one

Question 2

Print the following pattern on the screen using for loop

```
#
 #
  #
   #
    #
     #
    #
   #
  #
#
```

Question 3

- Add up -1+(-3)+(-5)+…+(-99)
- (hint: range() function)

Question 4

- Write a loop that calculates the total of the following series of numbers: $\frac{1}{30} + \frac{2}{29} + \frac{3}{28} + \ldots + \frac{30}{1}$

Question 5:
Complete the “number guessing program”

- Write a program to make computer pick a random positive integer (0-100), then let user guess the number. (“Please input a positive integer (0-100):”) ◦
- The program determine the value user input, if it greater than the truth value, print ”The number you guessed is too big” ; if it less than the truth value, print ”The number you guessed is too small” ; if they are the exact same number, print ”congrat, you’re right.” Let the user keep guessing until the right answer appears.

Advanced exercise_4-A

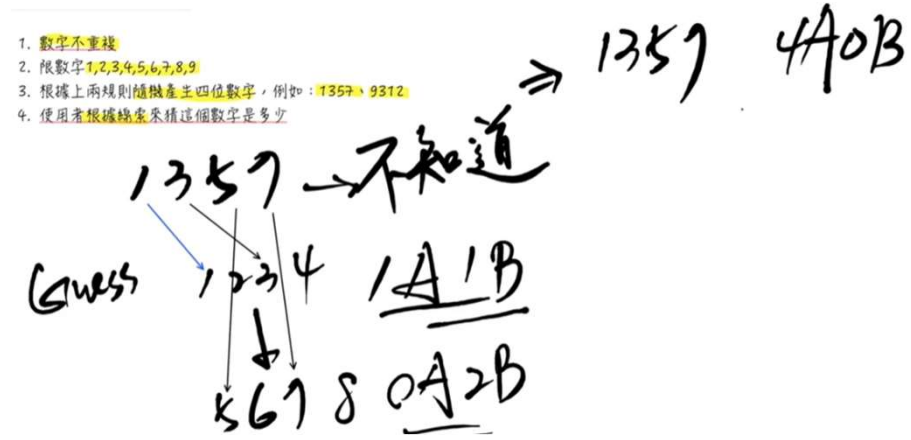
Question 4_A-1

Write a program which repeatedly reads numbers until the user enters “done”. Once “done” is entered, print out the total, count, and average of the numbers.

Question 4_A-2:
Guess the number

- Write a program to make the computer pick a random positive integer (0-50), then let users guess the number (“Please input a positive integer (0-50):”).
- If they guess outside that range, you prompt with an error encouraging them to choose a number within the proper range.
- Whenever they guess the wrong number you ask if they want to keep playing or if they'd like to quit
- Finally, when the user eventually guesses the right number you congratulate them and show the number of attempts they had.

Question 4_A-3: guess number(1A2B)



1. Can’t choose the same number
2. can only choose 1,2,3,4,5,6,7,8,9
3. pick four random number based on rules above, ex:9312
4. User guess the value based on the clue

Review

- Textbook: chapter 5: all