

ICS-104

Chapter 4

Notes

By: Naif Alqahtani

 Twitter: @NaifAlqahtani

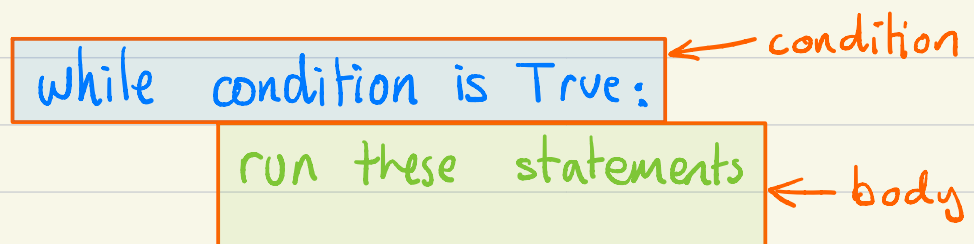
 Youtube.com/MWNLLT

• Loops

- loops are used to repeat a set of programming statements
- loops can help make code easier to understand and debug.

• While loop

- The while loop repeats statements inside the **body** as long as the header condition is **True**.



- No statement inside the **body** will run if the condition is **False**.
- Using **while True:** will make the inside statements run Forever
- To avoid making the loop run Forever, we need to modify part of the condition inside the **block** while block.

- An example to count to 10.

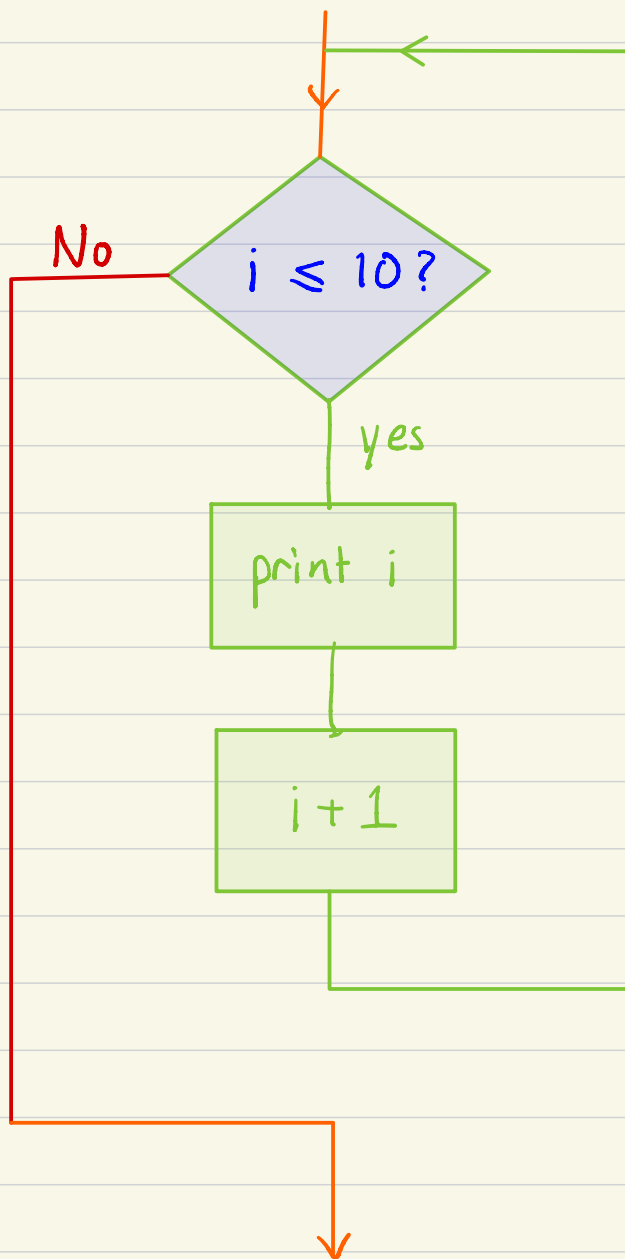
- This example is called a count-controlled loop

$i = 0$ \leftarrow initialize variable

while $i \leq 10$: \leftarrow stop once i reaches 10

print(i) \leftarrow repeat this statement

$i += 1$ \leftarrow this updates the condition making sure it does not run forever.



- revise common loop algorithms page 19,20,21

• The for loop

- a for loop can be used to iterate inside elements of a container.
- a container can be a string, list, tuple, etc...

important → - the variable iterator (the letter *i* for example) will hold the actual character instead of just a number when used with a string

- example

```
for i in "hello":  
    .....
```

this is a major difference between while loop and for loop. → - the letter *i* will take the value of each char in the string. So in the first loop *i* = "h" in the second loop *i* = "e" and so on.

- this loop will repeat from *i* = "h" to *i* = "o" ie, the loop will repeat `len("hello")` or for as many chars in the string.

— to achieve the previous example using a while loop, we can set i to 0 and repeat the while loop $\text{len}(\text{string}) - 1$ times.

— Since the value of i will be numbers, we must index the string. So we use:
 $\text{string}[i]$

```

3 string = "hello"
4
5
6 # _____
7 for i in string:
8     print(i)
9 # _____
10 i=0 < set i to 0
11 while i <= (len(string)-1):
12     print(string[i])
13     i += 1
14

```

since the values of i are letters, we can print them

loop from $i = 0$ to $i = \text{len}(\text{string}) - 1$ because counting starts from zero.

modify condition

notice, we indexed the string at i instead of printing i . This is because the values of i are numbers.

Output (stdout)

h
e
l
l
o

— both snippets of code produce the exact same result

— While both are the same, the For loop is easier and cleaner!

• range() in for loops

- range() is a function that returns a list of numbers

- since range returns a list of numbers, we can use it in for loops to iterate over it.

- range can have multiple arguments.

• range(5) returns [0, 1, 2, 3, 4]

• range(3, 8) returns [3, 4, 5, 6, 7]

• range(0, 7, 2) returns [0, 2, 4, 6]

start value (included) end value (not included) increment by 2

0 7 2

0 2 4 6

+2 +2 +2

- An example:

```
for i in range(0, 7, 2):  
    print(i)
```

- Output:

0
2
4
6

• Nested loops

- nested loops are loops inside loops
- they are sometimes used to make 2D text.

- lets say we want to print 10 "." in one line. we can do this.

```
1 for i in range(10):
2     print(".", end="")
3
```

- This outputs.

Output (stdout)

.....

- now we want to print 10 lines of the same 10 ".", to make a 10x10 grid that contains 100 ".".

- All we do is we make a for loop that repeats 10 times, then we insert our previous code inside it.

- Like this:

```
1 for i in range(10):
2     for i in range(10):
3         print(".", end="")
4     print()
```

→ repeat 10 times.

→ same code as last time

→ we want to print a new line after every row.

- This will be the output:



• Using loops for counting

- We can loop over a string letter by letter and increase a value if a condition is met.
- example: count how many letter "i" in the word "Curiosities"

```
1 word = "Curiosities" ← string
2 n = 0 ← counter to count letter i
3
4 for letter in word: ← here the value of i will be a letter in the string
5     if letter.lower() == "i":
6         n += 1
7
8 print(n)
```

→ convert current letter to lower then check if it is the letter i

→ increment the counter only if current letter is i

→ notice, print function is OUTSIDE of loop. This means that there will only be one output and it will be after the loop is finished.

OUTPUT:

• Random numbers

- The `random` module is used to return random numbers.

- The `randint()` function returns a random integer.

- example:

```
from random import randint
```

```
print(randint(1, 6))
```

- prints one random integer in the range 1 to 6

- both 1 and 6 are included.