

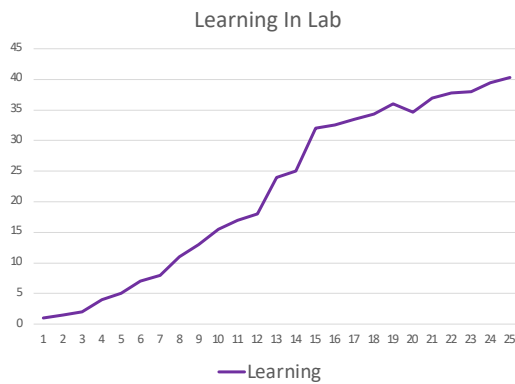
# Objectives

- Indefinite Loops

1

# Your Learning Journey

- You're learning a lot
  - Struggle is part of the learning

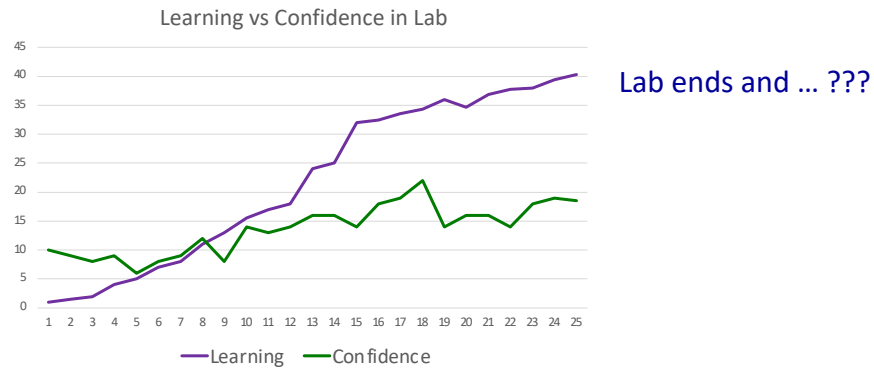


2

## Your Learning Journey

- But struggle affects your confidence

➤ Confidence != Learning

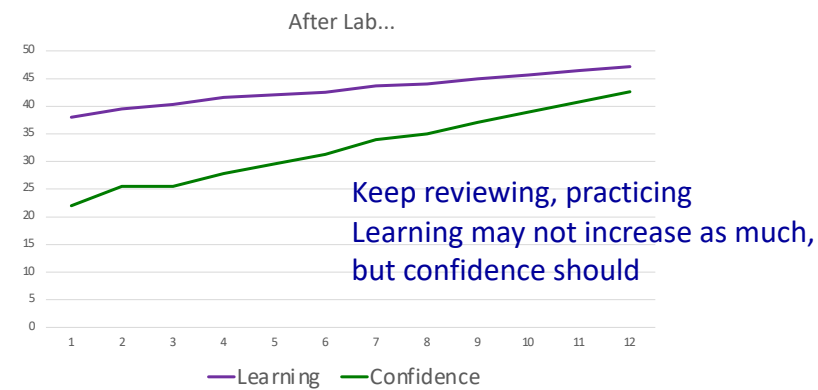


3

## Your Learning Journey

- But struggle affects your confidence

➤ Confidence != Learning



4

## Review

- We are judging a science fair.
- There is different criteria for winning a first place ribbon, depending on what grade the student is in.
- Given the variables `scienceScore` and `grade`
  - Write a condition that will evaluate to True if (and only if) the student's score is above the first place threshold of 60 points and the student's grade is 8.
    - Otherwise, the condition should evaluate to False

Feb 24, 2021

Sprenkle - CSCI111

5

5

## Review

- We are judging a science fair.
- There is different criteria for winning a first place ribbon, depending on what grade the student is in.
- Given the variables `scienceScore` and `grade`
  - Write a condition that will evaluate to True if (and only if) the student's score is above the first place threshold of 60 points and the student's grade is 8.
    - Otherwise, the condition should evaluate to False

```
scienceScore > 60 and grade == 8
```

Feb 24, 2021

Sprenkle - CSCI111

6

6

## Short-circuit Evaluation

- Don't necessarily need to evaluate all expressions in a compound expression
- A **and** B
  - If A is **False**, compound expression is **False**
- A **or** B
  - If A is **True**, compound expression is **True**
- No need to evaluate B
  - Put more important/limiting expression first
  - Example: `if count != 0 and sum/count > 10:  
do something`

Feb 24, 2021

Sprenkle - CSCI111

7

7

## INDEFINITE LOOPS

Feb 24, 2021

Sprenkle - CSCI111

8

8

## Definite vs Indefinite Loops

- **for** loops are *definite* loops
  - Execute a *fixed* number of times
- *Indefinite* loops: keep iterating until certain conditions are met
  - Depending on condition, no guarantee in advance of how many times the loop body will be executed

Feb 24, 2021

Sprenkle - CSCI111

9

9

## While Loop Syntax

```
keyword → while condition :  
          statement1  
          statement2  
          ...  
          statementn  
          } body of while loop
```

- Like a *looped if* statement
  - Execute statements **only** when condition is true
  - Stop executing when condition is false

Feb 24, 2021

Sprenkle - CSCI111

10

10

## While Loop

```
i = 0
while i < 5 :
    print("i equals", i)
    i+=1
print("Done", i)
```

Feb 24, 2021

Sprenkle - CSCI111

while.py

11

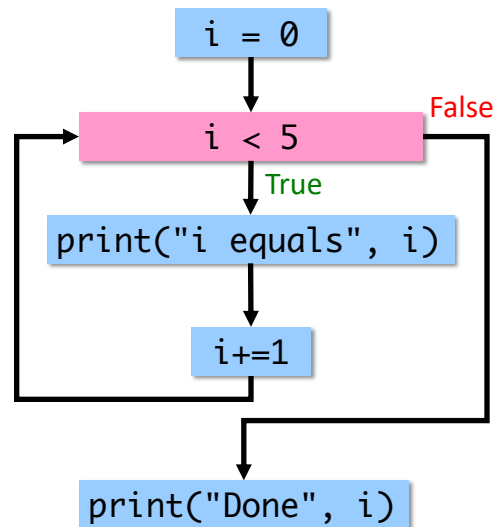
11

## While Loop

```
i = 0
while i < 5 :
    print("i equals", i)
    i+=1
print("Done", i)
```

Questions:

- How many times will *i* get printed out?
- How many times is the condition evaluated?
- What is the value of *i* after the loop?



Feb 24, 2021

Sprenkle - CSCI111

while.py

12

12

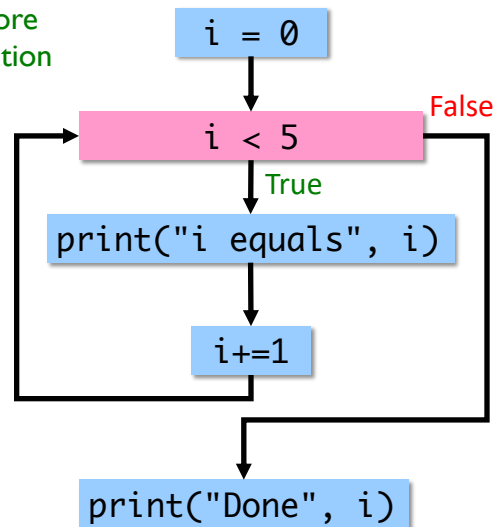
## While Loop

```
i = 0  
while i < 5 :  
    print("i equals", i)  
    i+=1  
print("Done", i)
```

Initialize i before using in condition

Questions:

- How many times will `i` get printed out?
- How many times is the condition evaluated?
- What is the value of `i` after the loop?



Feb 24, 2021

Sprenkle - CSCI111

while.py

13

13

## While vs. For Loops

- Any **for** loop can be translated into a **while** loop
  - But **NOT** vice versa
- **while** loops are more *powerful* than **for** loops

Feb 24, 2021

Sprenkle - CSCI111

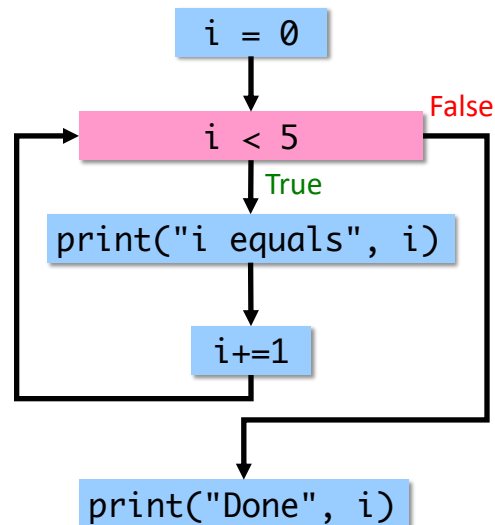
14

14

## Convert to a **for** loop

We *can* convert this **while** loop into a **for** loop because it executes a *fixed* number of times.

```
i = 0
while i < 5 :
    print("i equals", i)
    i+=1
print("Done", i)
```



Feb 24, 2021

Sprenkle - CSCI111

15

15

## Comparing **while** and **for**

- What are the main differences between these loops?
- What are the advantages and disadvantages of each?

```
i = 0
while i < 5 :
    print("i equals", i)
    i+=1
print("Done", i)
```

```
for i in range(5):
    print("i equals", i)
print("Done", i+1)
```

Feb 24, 2021

Sprenkle - CSCI111

[whilevsfor.py](http://whilevsfor.py)

16

16



## What Will This Loop Do?

```
count = 1
while count > 0:
    print(count)
    count += 1
```

Feb 24, 2021

Sprenkle - CSCI111

loop.py

17

17

## Infinite Loop

- Condition will never be **False** so keeps executing

```
count = 1
while count > 0:
    print(count)
    count += 1
```

- To stop an executing program in Linux use  
➤ Control-C

Feb 24, 2021

Sprenkle - CSCI111

18

18

## Infinite Loop Discussion

- Is there ever a time that an infinite loop is wanted?
  - Yes! For example in web servers, we have something like

```
while True:
    listenForRequest()
    handleRequest()
```
- Can a computer automatically detect infinite loops?
  - No that is an *undecidable* problem
  - Best to **prevent** infinite loops (more later)
    - Benefit of **for** loops: definite loops

Feb 24, 2021

Sprenkle - CSCI111

19

19

## A Very Simple Therapist

- Whenever a user tells the computer/program what they think, the program asks, "How does that make you feel?"
- Ends when user enters nothing ("")

Feb 24, 2021

Sprenkle - CSCI111

[therapist.py](#)

20

20

## A Very Simple Therapist

- Whenever a user tells the computer/program what they think, the program asks, "How does that make you feel?"
- Ends when user enters nothing ("")
- Partial example output:

```
Tell me what is bothering you.  
There is too much going on in my life.  
How does that make you feel?  
I feel like I am out of control and can't juggle it all.  
How does that make you feel?  
Really stressed and tired.  
How does that make you feel?  
  
Thank you! Come again!
```

Feb 24, 2021

Sprenkle - CSCI111

therapist.py

21

21

## Design Pattern: Sentinel Loop

- Sentinel: when to stop
  - “guard” to the loop
- “Keep going until you see the sentinel”

```
value = set value  
while value != sentinel :  
    process value  
    value = set value
```

- Options for “set value”
  - From a simple assignment, user input, calling a function, reading from a file, ...

Feb 24, 2021

Sprenkle - CSCI111

22

22

## while Loops: Sentinel Loop

```
# condition says when loop
# will continue
x=eval(input("Enter number:"))
while x % 2 != 0 :
    print("Error!")
    x = eval(input("Enter
number: "))
print(x, "is an even number.")
```

Example of a while loop that cannot be transformed into a for loop

Feb 24, 2021

Sprenkle - CSCI111

23

23

## while Loops: comparing use of break

```
# condition says when loop
# will continue
x=eval(input("Enter number:"))
while x % 2 != 0 :
    print("Error!")
    x = eval(input("Enter
number: "))
print(x, "is an even number.")
```

Loop condition says when to keep going

```
# have to look inside loop to
# know when it stops
while True :
    x = eval(input("Enter number:"))
    if x % 2 == 0 :
        break "breaks" out of a loop
    print("Error!")
print(x, "is an even number.")
```

Internal condition says when to stop

Using break statements:  
Best when loop has to execute at least once.

Feb 24, 2021

Sprenkle - CSCI111

24

## Summary: While vs. For Loops

- Any **for** loop can be translated into a **while** loop
- But **not** vice versa
- **while** loops are more *powerful* than **for** loops

Feb 24, 2021

Sprenkle - CSCI111

26

26

## Flipping Coins

- Problem: How many flips does it take to get 3 consecutive heads?
  - How can we simulate flipping a coin?
- In the game module:
  - Constants: HEADS and TAILS
  - Method:

```
def flipCoin():  
    """  
    Simulates flipping a non-biased coin.  
    returns either HEADS or TAILS.  
    """
```

[consecutiveHeads.py](#)

Feb 24, 2021

Sprenkle - CSCI111

27

27

## Flipping Coins

- Problem: How many flips does it take to get 3 consecutive heads?
  - How can we simulate flipping a coin?
- Challenge: write without a break and with a break

`game.py`  
`consecutiveHeads.py`

Feb 24, 2021

Sprenkle - CSCI111

28

28

## Looking Ahead

- Lab 5 due Friday
- Broader Issue: Self-driving Cars

Feb 24, 2021

Sprenkle - CSCI111

29

29