

# CSE 8A: Intro to Programming in Python

## Fall 2021

Lecture 10 - break, continue

UC San Diego

# Announcement

- Midterm this Wednesday
- Q/A session: zoom on Monday night 7pm – 8pm
  - Zoom link: <https://ucsd.zoom.us/my/yic242>

# Topics for Today

- break
- continue
- exercises

# break and continue

- Both statements are used to alter the flow of your program
- Both are usually used in loops
  - break: get out of the loop prematurely
  - continue: skip the current iteration of the loop

# Other ways to **break** from a loop

- In the loops we wrote so far, the only way to exit was via the loop condition

```
val = int(input())  
while val != -1 :  
    val = int(input())
```

```
for i in range(5):  
    print(i + 1)
```

# break statement

- In the loops we wrote so far, the only way to exit was via the loop condition
- What if we want to exit before the loop condition is false?
  - Unexpected program state, user-specified “exit” command, etc.
- We can use the **break** statement to immediately exit the current loop
  - You’ll usually want it inside of a conditional statement
  - “If this specific condition occurs, break out of the loop immediately”

# Break Statement

- **Break:** Immediately exit the current loop

```
while or for loop:  
    # code before break  
    if breakCondition:  
        break  
  
    # code after break
```

# What will be printed by the following code?

```
message = "hello"
for i in range(len(message)):
    if message[i] == 'l': #letter l
        break

    print(message[i], end = '')
```

- A. he
- B. hel
- C. hell
- D. hello
- E. This will cause an error



# What if I want to only skip *this* iteration?

- The “break” statement will break out of the entire loop immediately
- What if we want to just skip *this* iteration, but try the next iteration as usual?
  - E.g. skipping a header line in a spreadsheet
- We can use the **continue** statement to end the current iteration of the loop
  - You’ll usually want it inside of a conditional statement
  - “If this specific condition occurs, skip to the next iteration of the loop”

# Continue Statement

- **Continue:** Immediately jump to the next iteration of the current loop

```
for or while loop:  
    # code before continue  
    if continueCondition:  
        continue  
  
    # code after continue
```

# What will be printed by the following code?

```
message = "hello"  
for i in range(len(message)):  
    if message[i] == 'l': #letter l  
        continue  
  
    print(message[i], end = '')
```

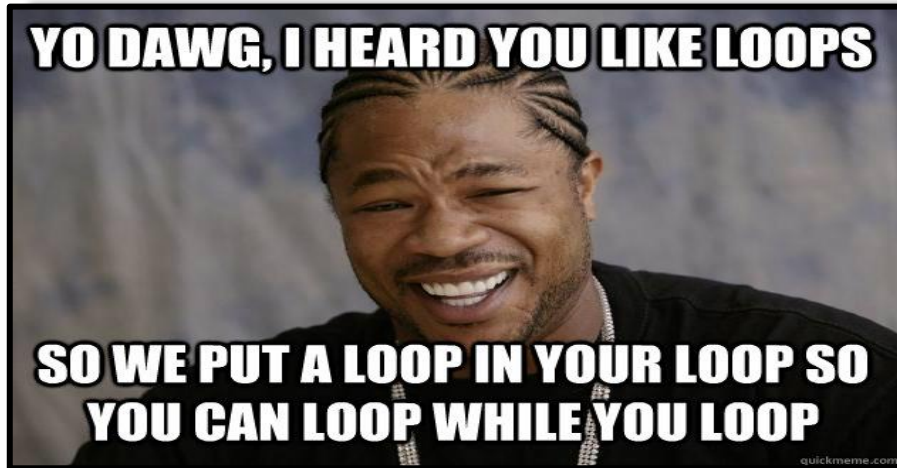
- A. he
- B. hell
- C. hello
- D. heo
- E. This will cause an error

# Nested Loops

- Loops help us automate performing repetitive tasks
- What about if, within a loop, we want to automate performing repetitive tasks?

# Nested Loops

- Loops help us automate performing repetitive tasks
- What about if, within a loop, we want to automate performing repetitive tasks?
- **Nested Loop:** A loop within a loop



What shape will be printed?

```
for i in range(4):  
    for j in range(2):  
        print('*', end = '')
```

A. \*\*\*\*\*

B. \*\*\*\*

\*\*\*\*

C. \*\*\*

\*\*\*

D. None of the above

# Nested Loops

- Loops help us automate performing repetitive tasks
- What about if, within a loop, we want to automate performing repetitive tasks?
- **Nested Loop:** A loop within a loop

```
for i in range(1, 10, 2):  
    for j in range(2, 9, 2):  
        print(i, '*', j, '=', i * j)
```

# Challenge

- A. 4
- B. 6
- C. 7
- D. 8
- E. This will cause an error

```
n = 4
x = 0
while n > 0:
    for i in range(n):
        x += 1
    n //= 2
print(x)
```