Week 4

Intro to Python

Please sign in!
https://goo.gl/LwZ8CA

Objectives

Chapter 5
(part 1)

- While loops
- Counter-controlled loops
- Sentinel value controlled loops
- User confirmation controlled loops
- Break/continue

Recap

Last week…

Boolean expressions

If statements

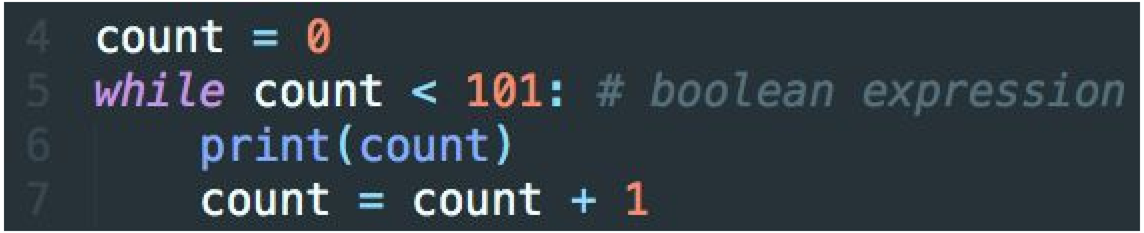
If, elif, else statements

Logical operators (and, or, not)

Random numbers (random.*randint()*)

Intro to While Loops

Say you wanted to print every number from 0 to 100...

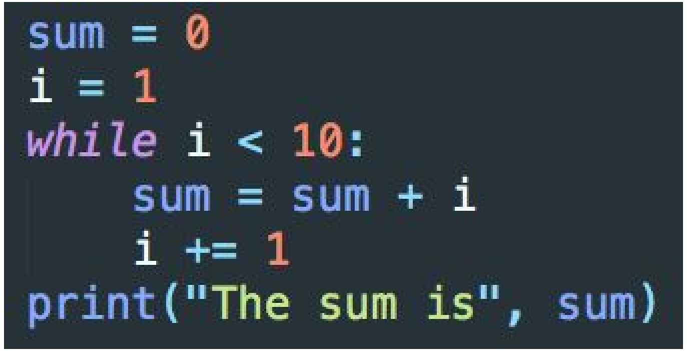


```python
count = 0
while count < 101: # boolean expression
    print(count)
    count = count + 1
```

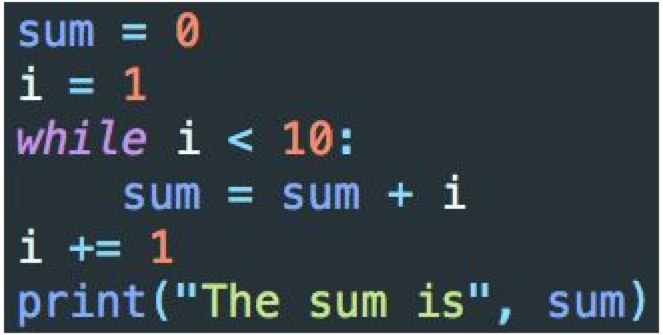
A while loop executes statements repeatedly as long as a condition remains **true**.

Another example and Indentation

Just like in if statements, indentation is important for loops.



```
sum = 0
i = 1
while i < 10:
    sum = sum + i
    i += 1
print("The sum is", sum)
```

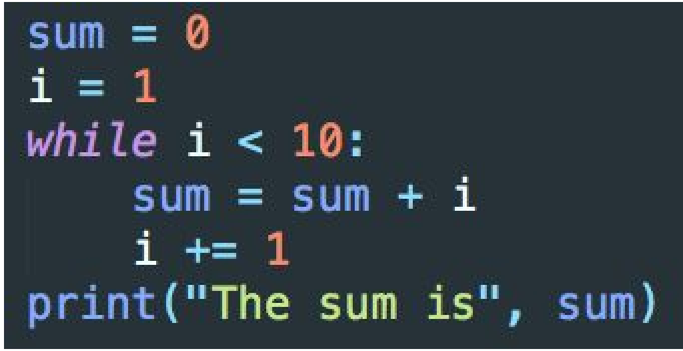


```
sum = 0
i = 1
while i < 10:
    sum = sum + i
i += 1
print("The sum is", sum)
```

Some loopy rules

The loop body starts and stops with the indentations.

The variable in the loop continuation condition and any variables updated in the loop body must be defined before the loop.



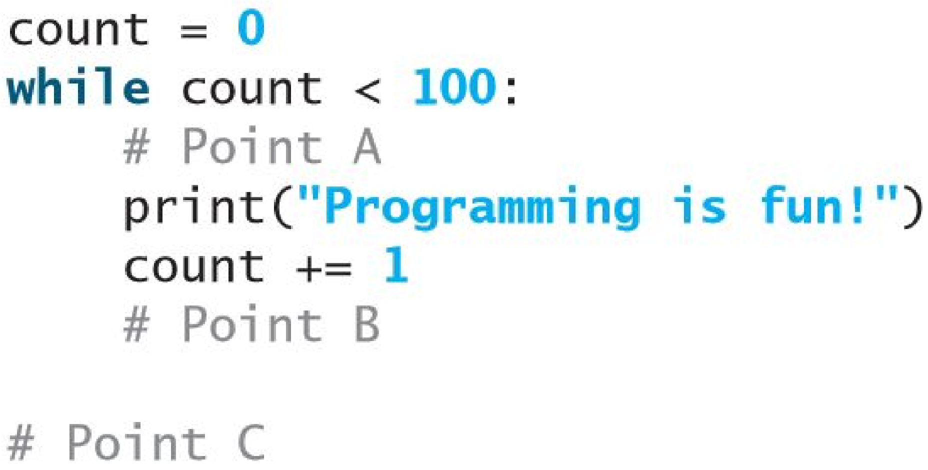
```
sum = 0
i = 1
while i < 10:
    sum = sum + i
    i += 1
print("The sum is", sum)
```

The loop continuation condition eventually MUST BECOME FALSE or you'll create an infinite loop.

Be careful in deciding if you want < or <= in the loop continuation condition.

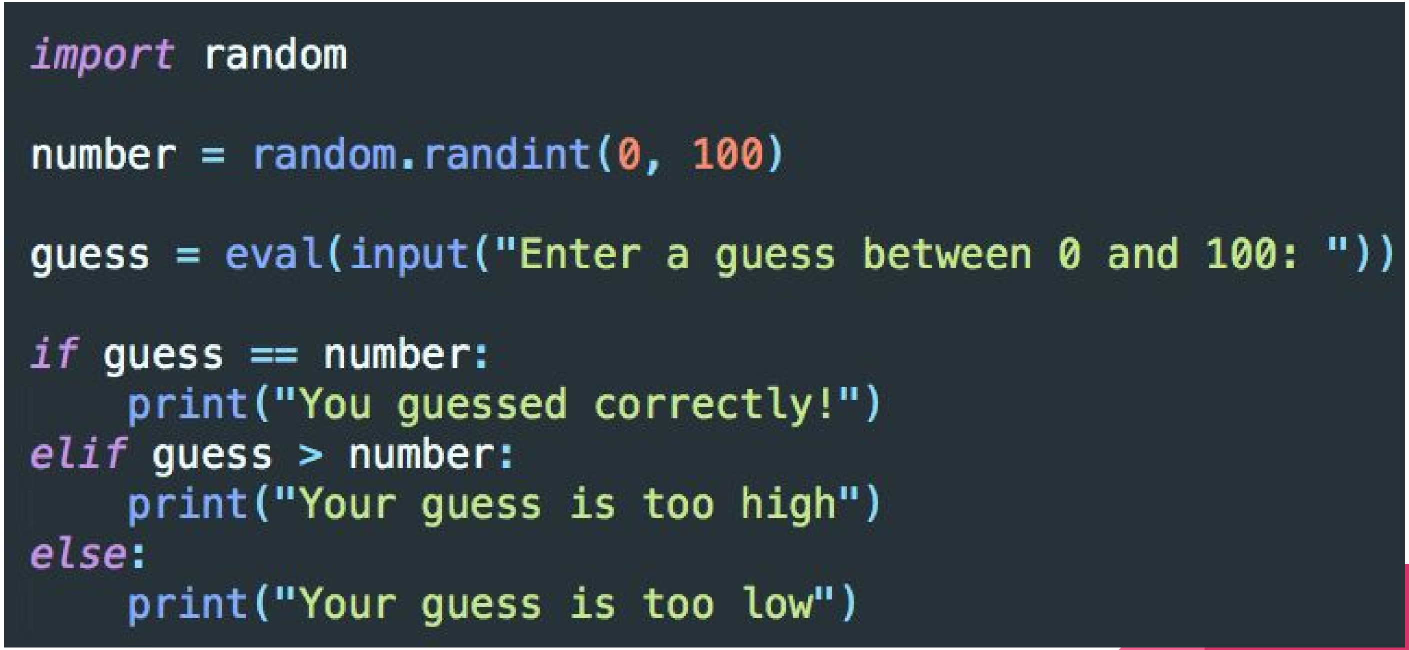
Practice!

At each point - is count < 100 always True, always False, or sometimes True/ sometimes False?



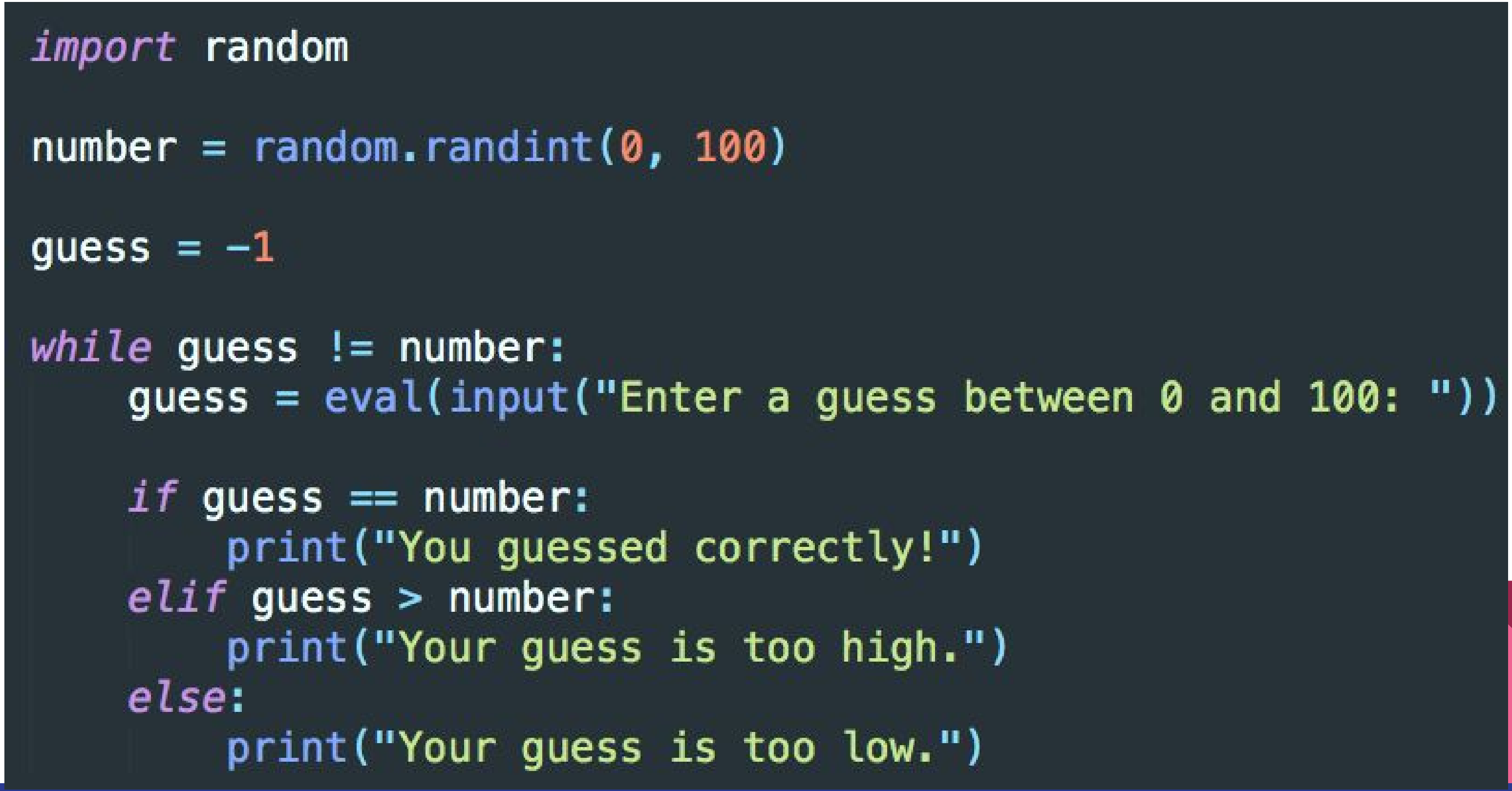
```
count = 0
while count < 100:
    # Point A
    print("Programming is fun!")
    count += 1
    # Point B

# Point C
```

Guessing Game



```python
import random

number = random.randint(0, 100)

guess = eval(input("Enter a guess between 0 and 100: "))

if guess == number:
    print("You guessed correctly!")
elif guess > number:
    print("Your guess is too high")
else:
    print("Your guess is too low")
```

Guessing Game



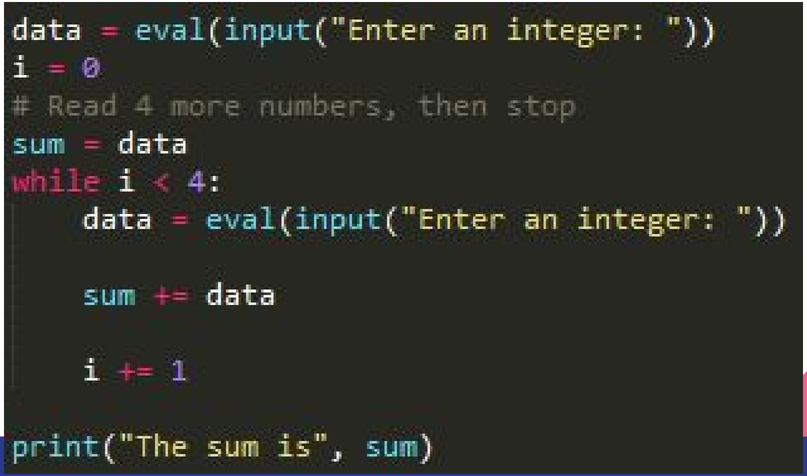
```python
import random

number = random.randint(0, 100)

guess = -1

while guess != number:
    guess = eval(input("Enter a guess between 0 and 100: "))

    if guess == number:
        print("You guessed correctly!")
    elif guess > number:
        print("Your guess is too high.")
    else:
        print("Your guess is too low.")
```

Counter-controlled loops

A *counter* variable starts at 0 and increases by one each time the loop iterates.

The loop condition is false when the counter reaches or passes a loop bound.

It is already known how many times the loop will have to iterate.

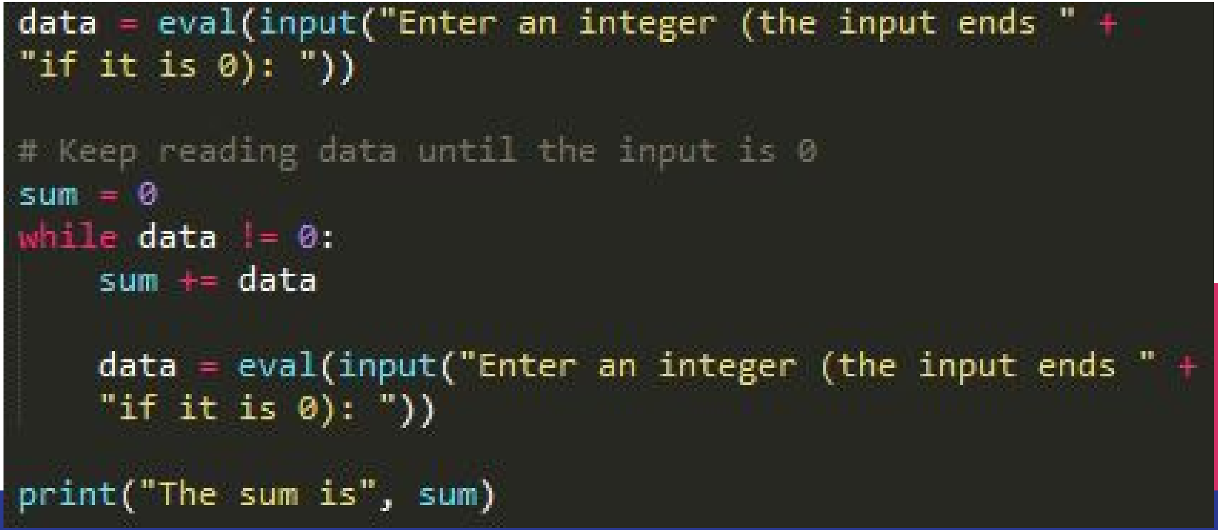


```
data = eval(input("Enter an integer: "))
i = 0
# Read 4 more numbers, then stop
sum = data
while i < 4:
    data = eval(input("Enter an integer: "))

    sum += data

    i += 1

print("The sum is", sum)
```

Sentinel value controlled loops

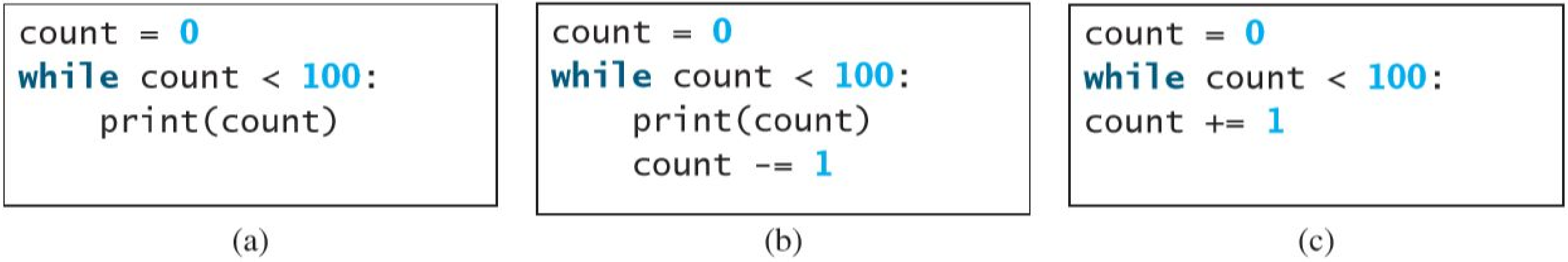
Each iteration, the user inputs data.

If the data is equal to a sentinel value, the loop continuation condition becomes false.



```python
data = eval(input("Enter an integer (the input ends " +
"if it is 0): "))

# Keep reading data until the input is 0
sum = 0
while data != 0:
    sum += data

    data = eval(input("Enter an integer (the input ends " +
    "if it is 0): "))

print("The sum is", sum)
```

Point out the errors!



```
count = 0
while count < 100:
    print(count)
```

(a)

```
count = 0
while count < 100:
    print(count)
    count -= 1
```

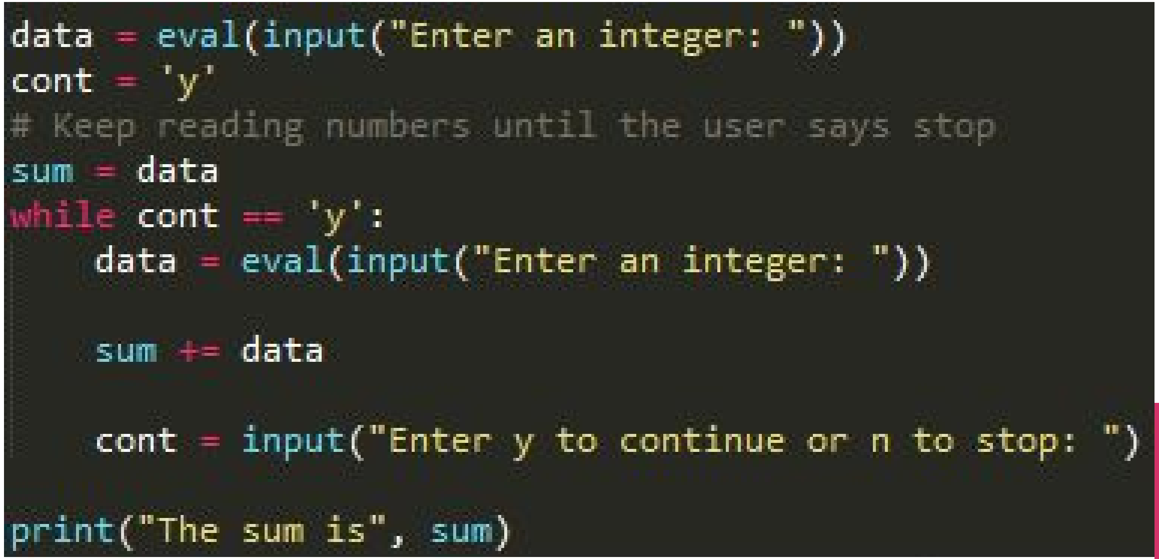
(b)

```
count = 0
while count < 100:
count += 1
```

(c)

User confirmation controlled loops

At the end of each iteration, the user must decide whether the loop should continue or not.



```
data = eval(input("Enter an integer: "))
cont = 'y'
# Keep reading numbers until the user says stop
sum = data
while cont == 'y':
    data = eval(input("Enter an integer: "))

    sum += data

    cont = input("Enter y to continue or n to stop: ")

print("The sum is", sum)
```