Loops

Leen-Kiat Soh
Computer Science & Engineering
University of Nebraska, Lincoln, NE

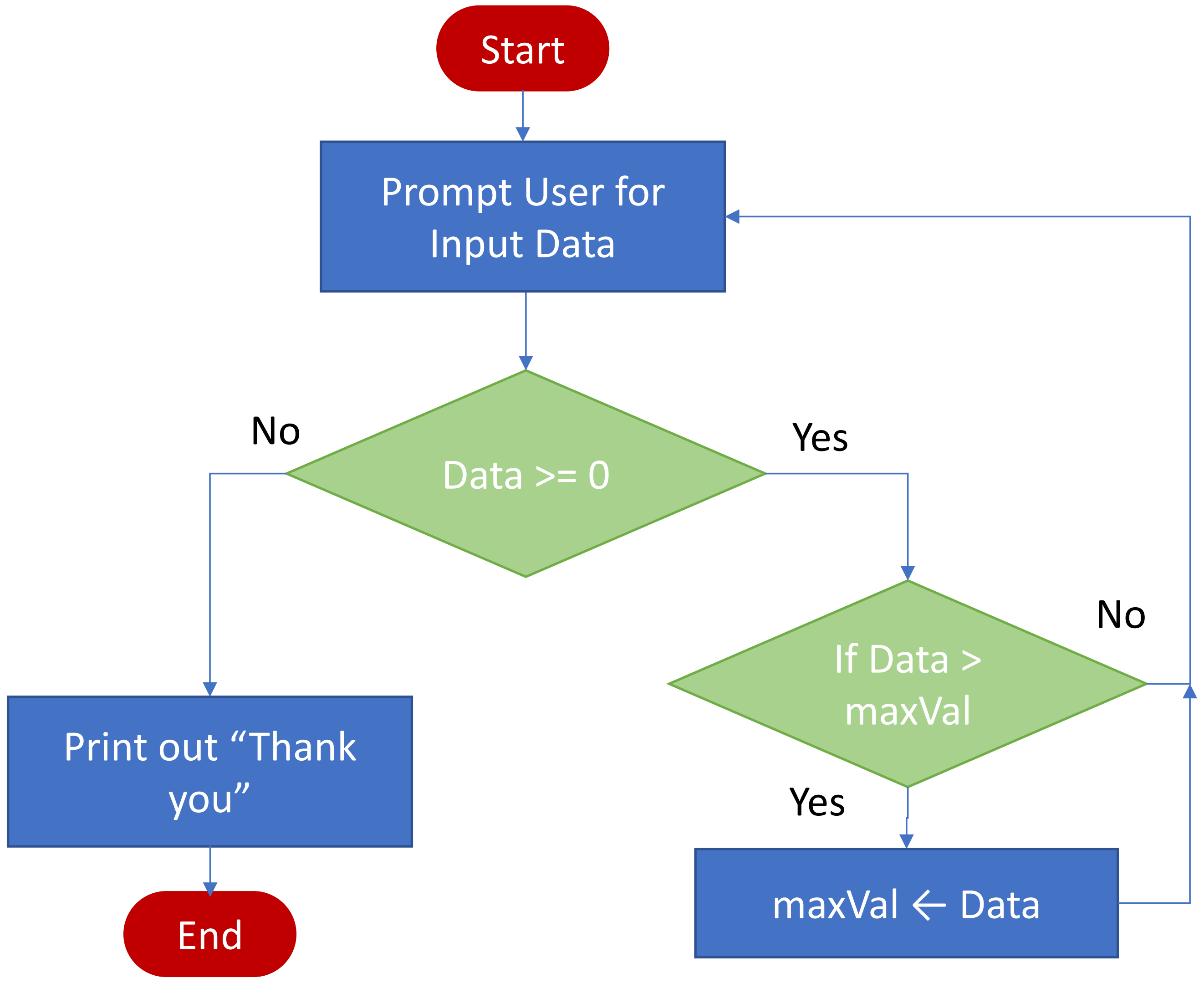
Loops

- An important "control structure" in programming concepts
  - **Controls the execution flow of a sequence**

Another important "control structure" in programming concepts is "conditionals" or "selection"

Flowchart

- Decision points
  - Conditions
  - Iteration



Loops

- **Pretest loops**
  - Test the conditions BEFORE entering the loop, or BEFORE executing the next iteration
  - The body of the loop may *not* be executed at all
- **Posttest loops**
  - Test the conditions AFTER entering the loop, or AFTER executing the next iteration
  - The body of the loop will *always* be executed at least once

Examples?

Getting a CAT scan before brain surgery and continuing to do so until recovery

Submitting a project proposal until it is approved

Going through a list of data values to find the average

Looking for whether a particular keyword is present in a list of keywords

Circling around at a parking lot to find a vacant parking space

Loops

- **Count-controlled loops**
    - The number of iterations is known before executing the loop
    - **For** loops
- **Sentinel-controlled loops**
    - The number of iterations is not known and depends on certain conditions (or sentinels) becoming true/false
    - **While** loops

Examples?

Getting a CAT scan before brain surgery and continuing to do so until recovery

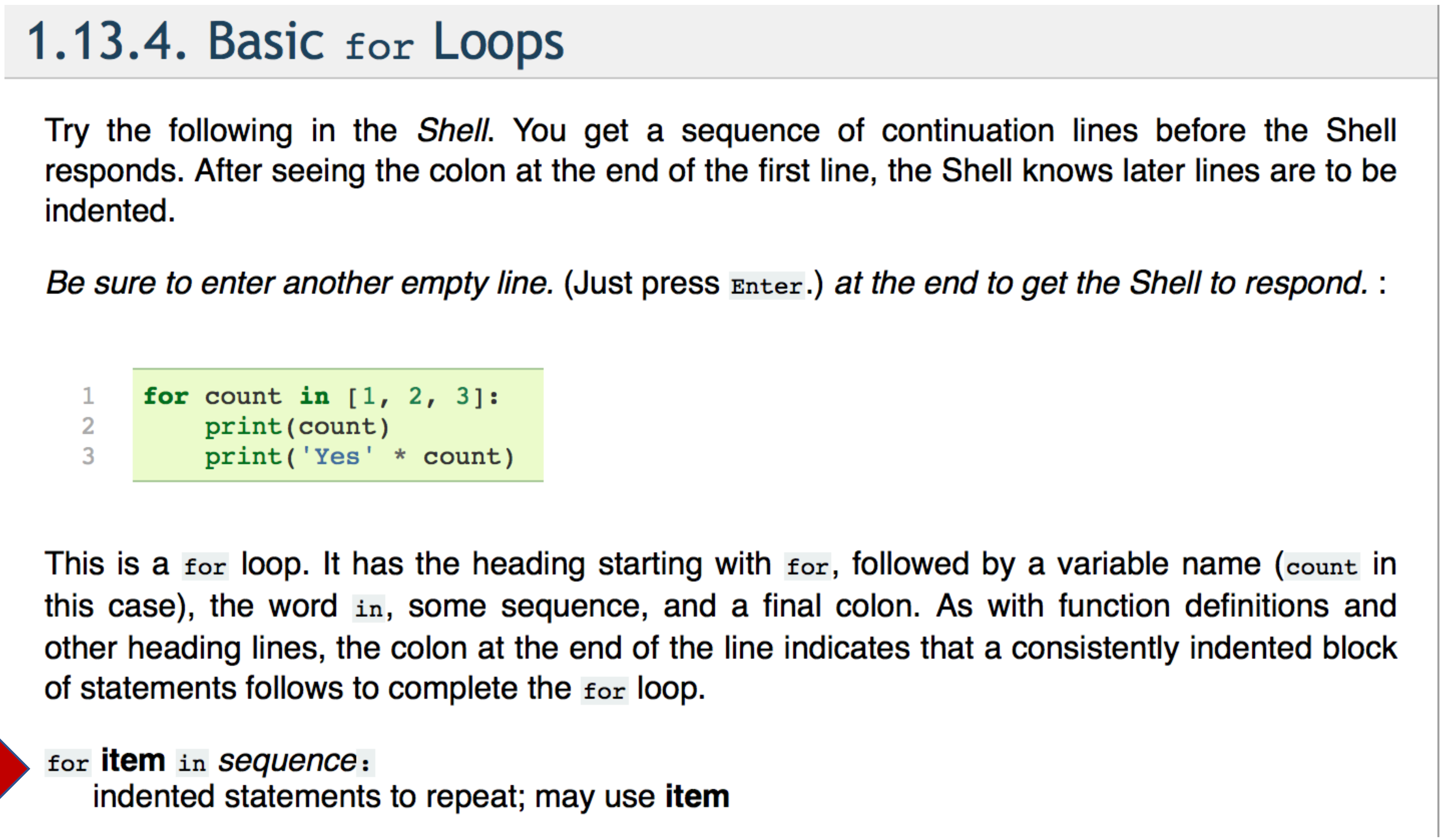
Submitting a project proposal until it is approved

Going through a list of data values to find the average

Looking for whether a particular keyword is present in a list of keywords

Entering a password to access an account

For Loops Syntax



1.13.4. Basic `for` Loops

Try the following in the *Shell*. You get a sequence of continuation lines before the Shell responds. After seeing the colon at the end of the first line, the Shell knows later lines are to be indented.

Be sure to enter another empty line. (Just press `Enter`.) *at the end to get the Shell to respond.* :

```
for count in [1, 2, 3]:
    print(count)
    print('Yes' * count)
```

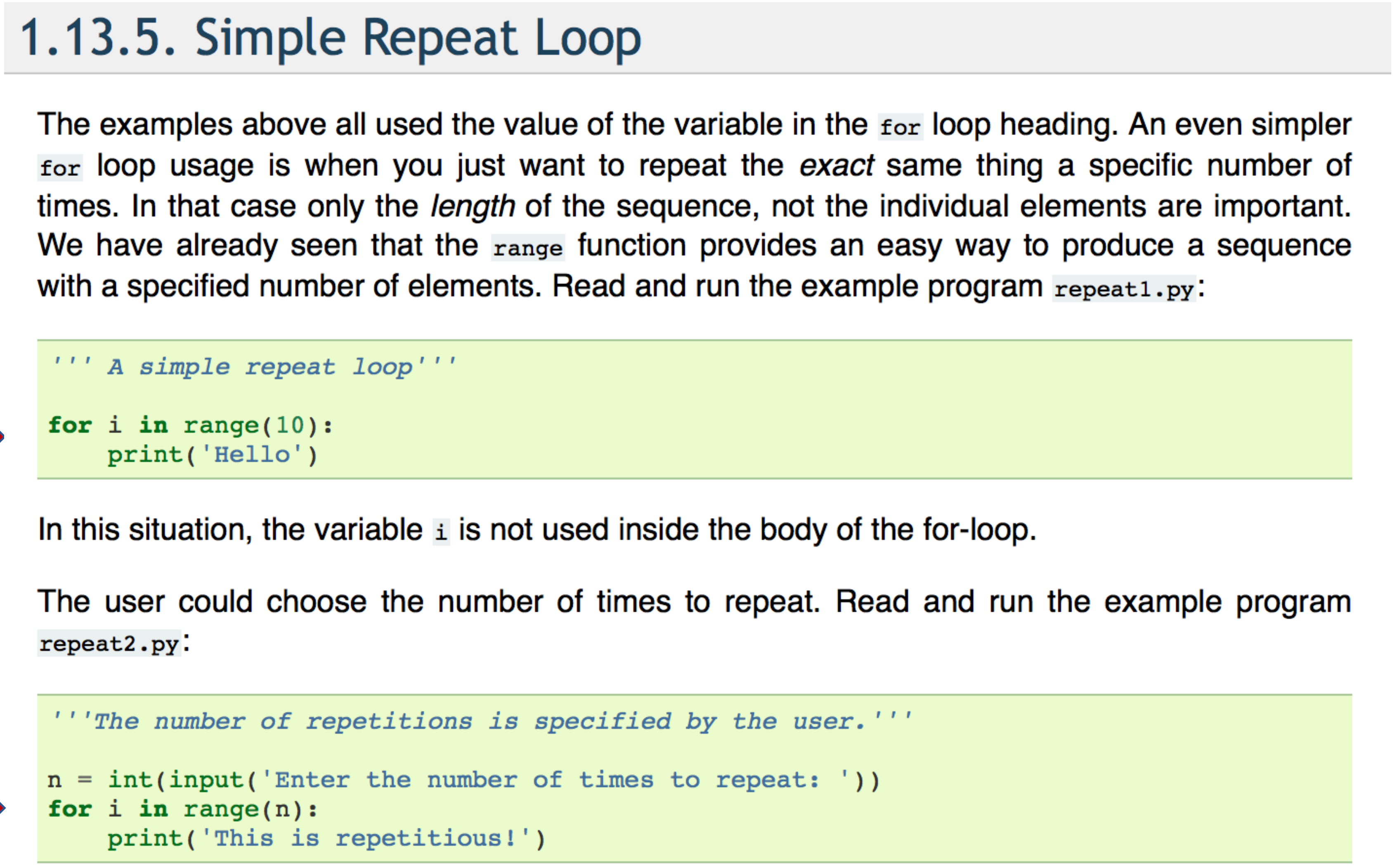
This is a `for` loop. It has the heading starting with `for`, followed by a variable name (`count` in this case), the word `in`, some sequence, and a final colon. As with function definitions and other heading lines, the colon at the end of the line indicates that a consistently indented block of statements follows to complete the `for` loop.

Syntax

`for` **item** `in` *sequence*`:`
    indented statements to repeat; may use **item**

http://anh.cs.luc.edu/handsonPythonTutorial/loops.html

For Loops Syntax 2



1.13.5. Simple Repeat Loop

The examples above all used the value of the variable in the `for` loop heading. An even simpler `for` loop usage is when you just want to repeat the *exact* same thing a specific number of times. In that case only the *length* of the sequence, not the individual elements are important. We have already seen that the `range` function provides an easy way to produce a sequence with a specified number of elements. Read and run the example program `repeat1.py`:

Syntax

```
''' A simple repeat loop'''

for i in range(10):
    print('Hello')
```

In this situation, the variable `i` is not used inside the body of the for-loop.

The user could choose the number of times to repeat. Read and run the example program `repeat2.py`:

Syntax

```
'''The number of repetitions is specified by the user.'''

n = int(input('Enter the number of times to repeat: '))
for i in range(n):
    print('This is repetitious!')
```

http://anh.cs.luc.edu/handsonPythonTutorial/loops.html

For Loops
Syntax 3

1
```
for x in (0, 9, 4, 5, 10):
    print(x)
```

2
```
for x in range(0, 4):
    print(x)
```

3
```
for x in range(0, 4, 2):
    print(x)
```

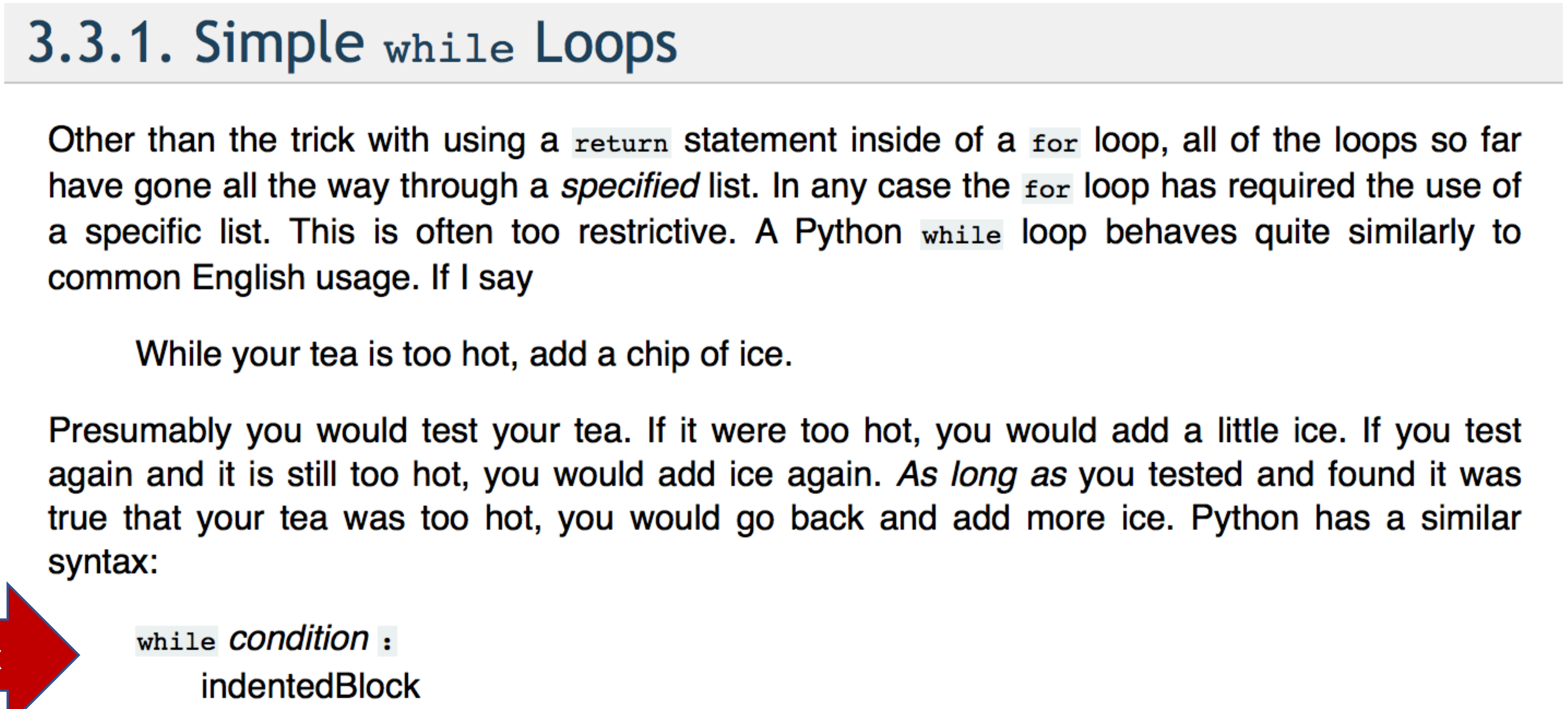
4
```
for x in range(0,3):
    print "Hello World %d" % (x)
```

Output?

What does "range()" do?

http://anh.cs.luc.edu/handsonPythonTutorial/loops.html

While Loops Syntax



3.3.1. Simple `while` Loops

Other than the trick with using a `return` statement inside of a `for` loop, all of the loops so far have gone all the way through a *specified* list. In any case the `for` loop has required the use of a specific list. This is often too restrictive. A Python `while` loop behaves quite similarly to common English usage. If I say

While your tea is too hot, add a chip of ice.

Presumably you would test your tea. If it were too hot, you would add a little ice. If you test again and it is still too hot, you would add ice again. *As long as* you tested and found it was true that your tea was too hot, you would go back and add more ice. Python has a similar syntax:

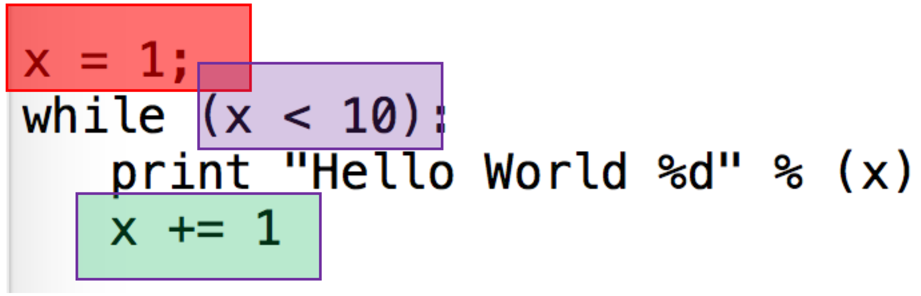
```
while condition :
    indentedBlock
```

Syntax

http://anh.cs.luc.edu/handsonPythonTutorial/whilestatements.html#while-statements

While Loops Syntax

- Initialization (before entering loop)
- Conditions (compound conditions) as the gatekeeper of the loop
  - To decide whether to continue or exit the loop
- Increment (inside loop, last line)



```
x = 1;
while (x < 10):
    print "Hello World %d" % (x)
    x += 1
```

? Why do we need to initialize?

? Why do we need to increment?

While Loops Syntax 3

1
```
x = 1;
while (x < 10):
    print "Hello World %d" % (x)
    x -= 1
```

2
```
x = 1;
while (x > 10):
    print "Hello World %d" % (x)
    x += 1
```

3
```
x = 1;
while (x < 10):
    print "Hello World %d" % (x)
    x = x*2
```

4
```
x = 0;
while (x < 10):
    print "Hello World %d" % (x)
    x = x*2
```

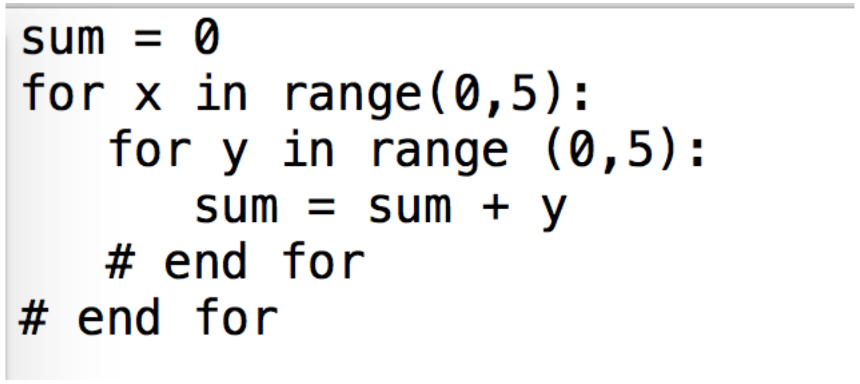
5
```
x = 1;
while (x == 10):
    print "Hello World %d" % (x)
    x = x + 2
```

6
```
x = 0;
while (x > 10):
    print "Hello World %d" % (x)
    x = x*2
```

Output?

? What is an infinite loop?

Nested For Loops



```
sum = 0
for x in range(0,5):
    for y in range (0,5):
        sum = sum + y
    # end for
# end for
```

What is the final value of the variable “sum”?

TRACE

x (0, 1, 2, 3, 4)	y (0, 1, 2, 3, 4)	sum
		0

Nested For Loops

```
sum = 0
for x in range(0,5):
    for y in range (x,5):
        sum = sum + x + y
    # end for
# end for
```

What is the final value of the variable “sum”?

TRACE

x (0, 1, 2, 3, 4)	y (x, …., 4)	sum
		0

Nested While Loops

```
sum = 0
x = 0
while (x < 5):
    y = 0
    while (y < 5):
        sum = sum + y
        y = y + 1
    # end while
    x = x + 2
# end while
```

What is the final value of the variable “sum”?

TRACE

x (0, ???, 4)	y (0, ???, 4)	sum
		0

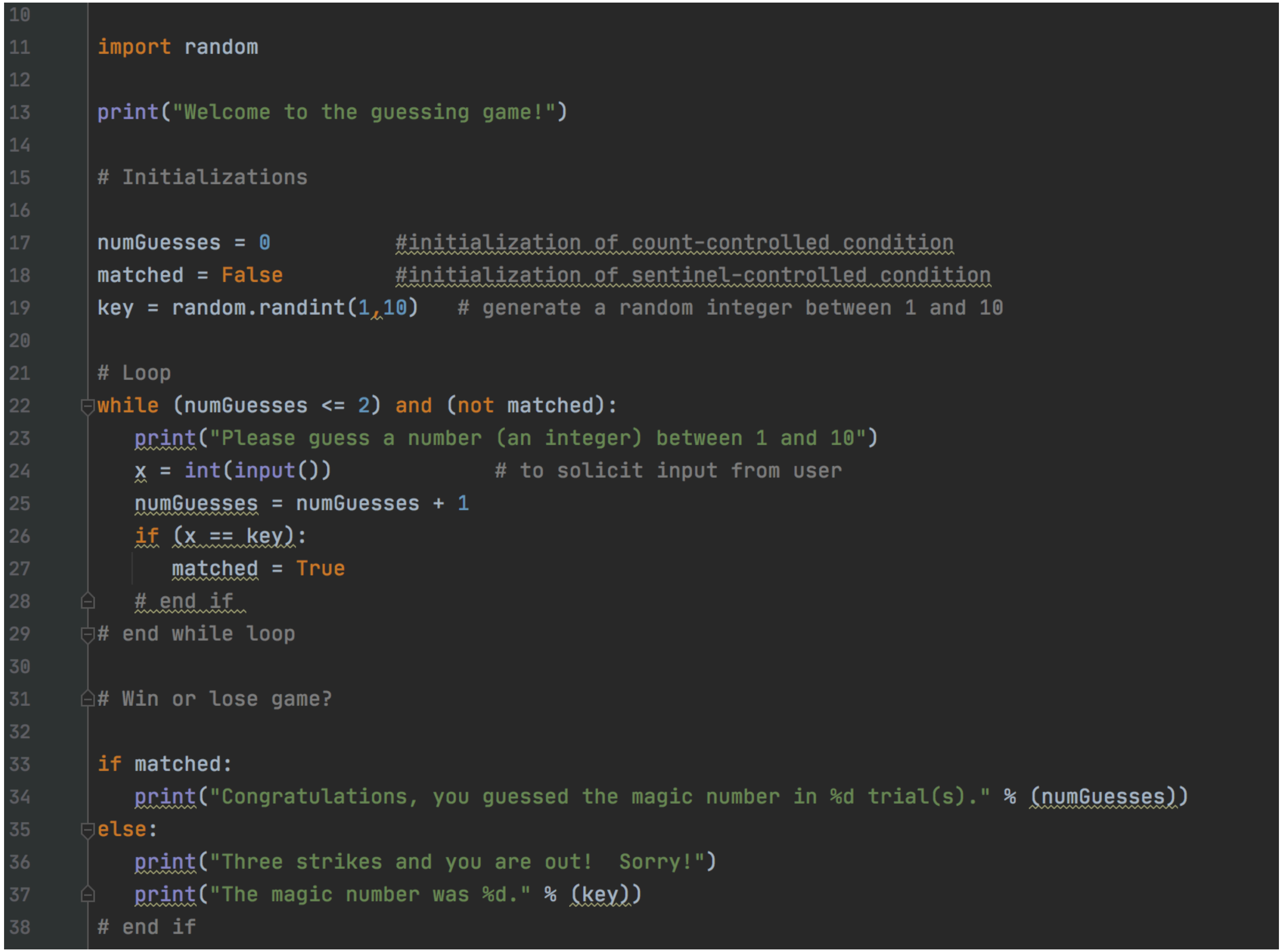
Nested Loops MORE

- Often times, the body of a loop is yet another loop!
- Think about how to process data that is 2D, or 3D, or N-dimensional
- **Can you think of an example data that has 2 dimensions?**

- **We will come back to this topic after we discuss arrays/lists**

Guessing Game

- Count-controlled loop?
- Sentinel-controlled loop
- Combination of loops + conditionals
- Compound Boolen expression



```python

import random

print("Welcome to the guessing game!")

# Initializations

numGuesses = 0          #initialization of count-controlled condition
matched = False         #initialization of sentinel-controlled condition
key = random.randint(1,10)   # generate a random integer between 1 and 10

# Loop
while (numGuesses <= 2) and (not matched):
   print("Please guess a number (an integer) between 1 and 10")
   x = int(input())              # to solicit input from user
   numGuesses = numGuesses + 1
   if (x == key):
      matched = True
   # end if
# end while loop

# Win or lose game?

if matched:
   print("Congratulations, you guessed the magic number in %d trial(s)." % (numGuesses))
else:
   print("Three strikes and you are out!  Sorry!")
   print("The magic number was %d." % (key))
# end if
```

Pitfalls

- **IMPORTANT**: Use indentations to designate “scope” or “block”

```
x = 1;
while (x < 10):
    print("Hello World")
    x -= 1
```

= ?

```
x = 1;
while (x < 10):
    print("Hello World")
x -= 1
```

- **Infinite loop**: Conditions never becoming false
- **Never-enter-the-loop**: Conditions never becoming true in the first place
- **Off-by-one-error:**
    - E.g., how many hours are there between 2 pm of Sep 8 and 5 pm of Sep 9?
    - E.g., how many posts are needed to build a fence 1000 feet long with 100 feet between posts?