

Objectives

- Indefinite Loops
- Exam review

Feb 3, 2010

Sprenkle - CSCI111

1

Lab Review

- 1 “Challenge” problem
- 1 Application problem

Feb 3, 2010

Sprenkle - CSCI111

2

Enhanced Lottery Game

- Check if user's pick matches the number you generated

Feb 1, 2010

Sprenkle - CS111

[pick4winner.py](#)

3

Indefinite Loops

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

Feb 3, 2010

Sprenkle - CSCI111

4

While Loop Syntax

```
while condition :  
    statement1  
    statement2  
    ...  
    statementn
```

keyword

body of while loop

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

Feb 3, 2010

Sprenkle - CSCI111

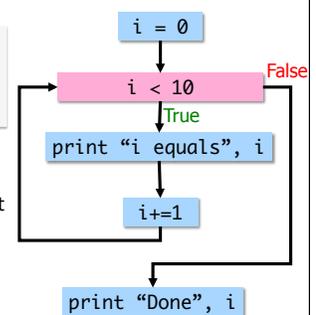
5

While Loop

```
i = 0  
while i < 10 :  
    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 3, 2010

Sprenkle - CSCI111

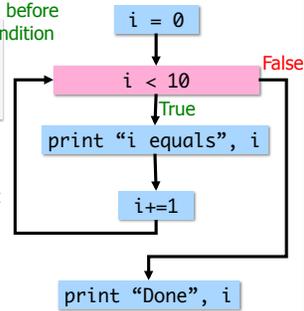
[while.py](#)

6

While Loop

```
i = 0
while i < 10 :
    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 3, 2010

Sprengle - CSCI111

while.py

7

While vs. For Loops

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

Feb 3, 2010

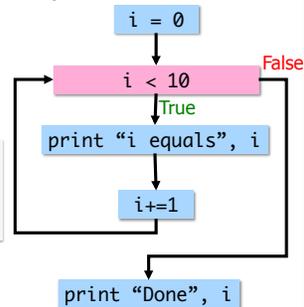
Sprengle - CSCI111

8

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 < 10 :
    print "i equals ", i
    i+=1
print "Done", i
```



Feb 3, 2010

Sprengle - CSCI111

9

Comparing while and for

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

```
i = 0
while i < 10 :
    print "i equals ", i
    i+=1
print "Done", i
```

```
for i in xrange(10):
    print "i equals", i
print "Done", i
```

Feb 3, 2010

Sprengle - CSCI111

whilevsfor.py

10

What Will This Loop Do?

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

Feb 3, 2010

Sprengle - CSCI111

loop.py

11

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 3, 2010

Sprengle - CSCI111

12

Infinite Loop Questions

- Is there ever a time that an infinite loop is wanted?
 - Yes! For example in web servers, we have something like
- Can a computer automatically detect infinite loops?
 - No that is an **undecidable** problem
 - Best to **prevent** infinite loops (more later)
 - Benefit of Python's **for** loops: definite loops

```
while True:
    listenForRequest()
    handleRequest()
```

Feb 3, 2010

Sprenkle - CSC1111

13

Unknown Number of Iterations

- Sums numbers input by user
 - Stop when the user inputs some designated stop value (**enter** key --> "")

Feb 3, 2010

Sprenkle - CSC1111

sumtillenter.py 14

Design Pattern: Sentinel Loop

- Sentinel: when to stop
 - "guard" to the loop

```
value = get input
while value != sentinel :
    process value
    value = get input
```

Feb 3, 2010

Sprenkle - CSC1111

15

Discussion

- How can we make sure that the loop actually stops (is not infinite)?

Feb 3, 2010

Sprenkle - CSC1111

16

Discussion

- How can we make sure that the loop actually stops (is not infinite)?
 - Update the condition's variable inside loop
 - Test
- How you'll usually detect an infinite loop...
 - "Why isn't my program giving me any output?"
 - If the program isn't exiting, probably an infinite loop

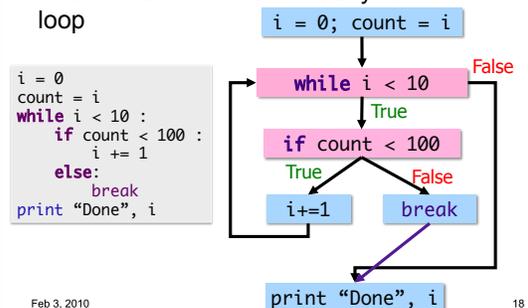
Feb 3, 2010

Sprenkle - CSC1111

17

Use of break statement

- **break** statement can "break you" out of a loop



Feb 3, 2010

18

while Loops: comparing use of break

```
# condition shows when loop # have to look inside loop to
# will stop executing      # know when it stops
x= input("Enter a number:") while True :
while x % 2 != 0 :         x = input("Enter a number:")
    x = input("Try again.  if x %2 == 0 :
    Enter a number:")     break
print x, "is an even     print x, "is an even number."
number."
```

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

Feb 3, 2010

Sprengle - CSCI111

19

While vs. For Loops

- Any **for** loop can be translated into a **while** loop
 - **Not vice versa**
- **while** loops are more **powerful** than **for** loops
 - Give an example of a **while** loop that can't be converted to a **for**

Feb 3, 2010

Sprengle - CSCI111

20

Summary of Control-Flow Building Blocks (so far)

- Conditional statements
 - if, if-else, if-elif-else
- Loops
 - while, for

Feb 3, 2010

Sprengle - CSCI111

21

Exam 1 Prep

- Cumulative up to today
 - We keep using the ideas from the first day of class
 - Basic Linux commands used during every lab
- Similar problems as in handouts, class discussion, labs
- Read code and explain what it does
 - What it displays as output
- Sections: Very Short Answer, Short Answer, Write Code
- Online prep document

Feb 3, 2010

Sprengle - CSCI111

22

Grading Overview

- Labs: 38%
- 2 Exams: 30%
- Final: 20%
- Broader Issues: 7%
- Participation: 5%

Feb 3, 2010

Sprengle - CSCI111

23