

Objectives

- Indefinite Loops

Feb 2, 2011

Sprenkle - CSCI111

1

Lab Review

- 1 “Challenge” problem
 - Double Jeopardy! Clue:
 - The largest number in the Fibonacci sequence that is also a day of the month
- 1 Application problem

Feb 2, 2011

Sprenkle - CSCI111

2

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 2, 2011

Sprenkle - CSCI111

3

While Loop Syntax

- ```
while condition :
 statement1
 statement2
 ...
 statementn
```
- keyword
- body of while loop
- loop stops when condition is False
- Like a looped **if** statement
    - Execute statements **only** when condition is true

Feb 2, 2011

Sprenkle - CSCI111

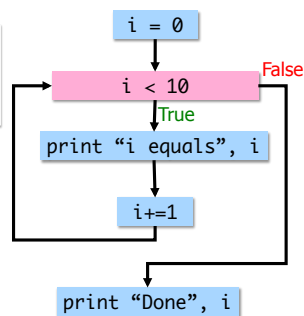
4

## 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 2, 2011

Sprenkle - CSCI111

while.py

5

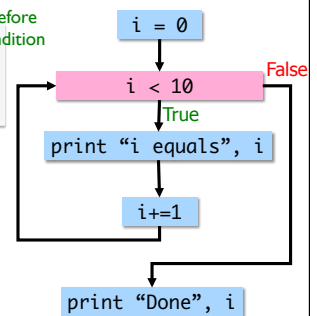
## While Loop

```
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 2, 2011

Sprenkle - CSCI111

while.py

6

## 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 2, 2011

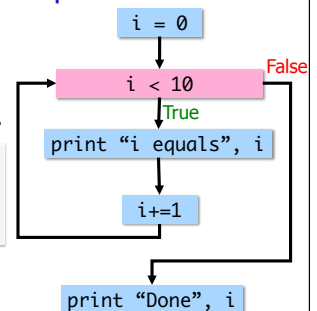
Sprenkle - CSCI111

7

## 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 2, 2011

Sprenkle - CSCI111

8

## 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 2, 2011

Sprenkle - CSCI111

whilevsfor.py

9

## What Will This Loop Do?

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

Feb 2, 2011

Sprenkle - CSCI111

loop.py

10

## 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 2, 2011

Sprenkle - CSCI111

11

## 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 Python's **for** loops: definite loops

Feb 2, 2011

Sprenkle - CSCI111

12

## 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 2, 2011

Sprenkle - CSCI1111

therapist.py

13

## Design Pattern: Sentinel Loop

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

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

Feb 2, 2011

Sprenkle - CSCI1111

14

## Unknown Number of Iterations

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

Feb 2, 2011

Sprenkle - CSCI1111

sumtillzero.py

15

## Discussion

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

Feb 2, 2011

Sprenkle - CSCI1111

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

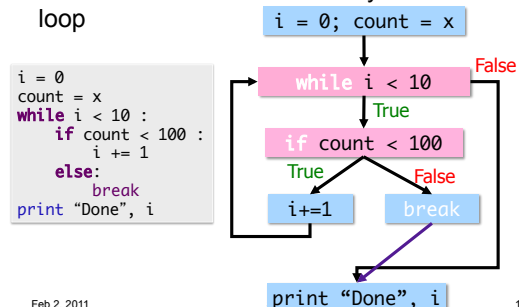
Feb 2, 2011

Sprenkle - CSCI1111

17

## Use of break statement

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



Feb 2, 2011

18

## while Loops: comparing use of break

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

Says when to keep going

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

Says when to stop

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

Feb 2, 2011

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 2, 2011

Sprengle - CSCI111

20

## Summary of Control-Flow Building Blocks (so far)

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

for

while

if

if  
else

if  
elif  
else

Can use in all sorts of different combinations

Feb 2, 2011

Sprengle - CSCI111

21

## This Week...

- For Friday
  - Lab 3 completed (although not printed)
  - Read, summarize article about the DARPA Urban Challenge
- Friday: starting Object-oriented Programming

Feb 2, 2011

Sprengle - CSCI111

22