

Objectives

- Review: Testing
- Exception handling
- Indefinite Loop

Mar 9, 2012

Sprenkle - CSCI111

1

Review

- How are parameters passed to functions?
- How do you set a default value for a parameter in a function?
- How do you create a module?

Mar 9, 2012

Sprenkle - CSCI111

2

TESTING FUNCTIONS

Mar 9, 2012

Sprenkle - CSCI111

3

Testing Functions

- Functions make it easier for us to test our code
- We can write code to test the functions
 - Input: parameters
 - Output: what is returned
 - We can verify programmatically

What are good tests for
`binaryToDecimal(binnum)` and `isBinary(candidate)?`

`binaryToDecimal.test.py`

Mar 9, 2012

Sprenkle - CSCI111

4

Debugging Advice

- Build up your program in steps
 - Always write small pieces of code
 - Test, debug. **Repeat**
- Write function body as part of **main**, test
 - Then, separate out into its own function
 - Similar to process using in lab probs
- Test function separately from other code

Mar 9, 2012

Sprenkle - CSCI111

5

EXCEPTION HANDLING

Mar 9, 2012

Sprenkle - CSCI111

6

Handling Exceptions

- Using try/except statements

- Syntax:

```
try:
    <body>
except [<errorType>]:
    <handler>
```

Optional: use this to handle specific error types appropriately

- Example:

```
try:
    age = eval(input("Enter your age: "))
    currentyear = int(input("Enter the current year: "))
except:
    print("ERROR: Your input was not in the correct form.")
    print("Enter integers for your age and the current year")
    sys.exit()
```

Mar 9, 2012

Sprenkle - CSCI111 yearborn.py

7

Handling Exceptions

- Other types of exceptions

- File exceptions:

- File doesn't exist
- Don't have permission to read/write file

Mar 9, 2012

Sprenkle - CSCI111 file_handle.py

8

INDEFINITE LOOPS

Mar 9, 2012

Sprenkle - CSCI111

9

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

Mar 9, 2012

Sprenkle - CSCI111

10

While Loop Syntax

```
while condition:
```

loop stops when condition is False

```
    statement1
```

```
    statement2
```

```
    ...
```

```
    statementn
```

keyword

body of while loop

- Like a looped **if** statement

- Execute statements **only** when condition is true

Mar 9, 2012

Sprenkle - CSCI111

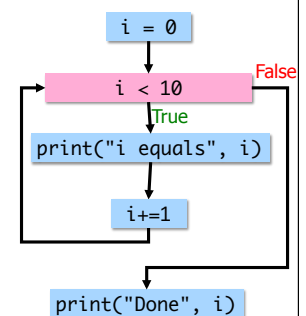
11

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?



Mar 9, 2012

Sprenkle - CSCI111

while.py

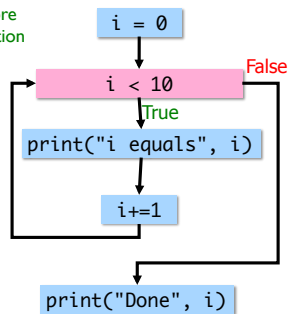
12

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?



Mar 9, 2012

Sprenkle - CSCI111

while.py

13

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

Mar 9, 2012

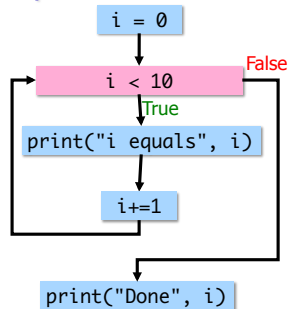
Sprenkle - CSCI111

14

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)
```



Mar 9, 2012

Sprenkle - CSCI111

15

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 range(10):
    print("i equals", i)
print("Done", i+1)
```

Mar 9, 2012

Sprenkle - CSCI111

whilevsfor.py

16

What Will This Loop Do?

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

Mar 9, 2012

Sprenkle - CSCI111

loop.py

17

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**

Mar 9, 2012

Sprenkle - CSCI111

18

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

Mar 9, 2012

Sprenkle - CSCI111

19

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!
```

Mar 9, 2012

Sprenkle - CSCI111

therapist.py

20

Design Pattern: Sentinel Loop

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

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

Mar 9, 2012

Sprenkle - CSCI111

21

Summary: 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**

Mar 9, 2012

Sprenkle - CSCI111

22

Broader Issues: Digital Humanities

- Process large amounts of information to learn new information
 - **Not necessarily numeric information**
 - Text-based, images, multi-media, etc.
- Area growing in importance

Trang
Gabi
Gaurav
Mary
John K

Haley
Luke
Jonathan
Will

Connor
Hang
Colby
Josh
Emily

Phil
Kari
Koven
Shannon

John G
Deirdre
Sam
Drew

Mar 9, 2012

Sprenkle

Katherine Crowley's Talk

Discussion

- What problem were the researchers solving?
- How did they solve the problem?
 - What were some of the challenges in solving it?
- How would you solve/implement ...
 - Processing the words
 - Keeping track of how many words there are
 - When there is a new word
- Are there any follow up questions that can be asked/answered?
- What other gains do you see possible by applying computing to the humanities?
- Will digital humanities take the "art" out of the humanities?

Mar 9, 2012

Sprenkle - CSCI111

24

Discussion

- Were you surprised by this use of computer science in the humanities?
- What other gains do you see possible by applying computing to the humanities?
- Will digital humanities take the “art” out of the humanities?
- Did anyone try out <http://ngrams.googlelabs.com/>?

Mar 9, 2012

Sprenkle - CSCI111

25

Relation to Class

- Most concrete example of something we could write so far
- Analyzing texts, data
 - [Strings](#)
- Files containing the data
- Information retrieval, Natural Language Processing
 - [Growing fields of computer science](#)
 - E.g., Speech recognition
 - [Process sentences](#)
 - Determine subject/verb/direct object, etc.

Mar 9, 2012

Sprenkle - CSCI111

26

Analyzing Shakespeare

- hyphenated compound words
- Relative clauses per thousand
 - [Shakespeare used less often](#)
- Grade-level of writing, as measured by word- and sentence-length
- Percentage of open- and feminine-ended lines
 - [Open: no punctuation](#)
 - [Feminine: unstressed syllable](#)
 - [Shakespeare used more often](#)

Mar 9, 2012

Sprenkle - CSCI111

27