

Objectives

- Wrap up indefinite loops
- Text processing, manipulation
 - String operations, processing, methods
- Broader Issue: Self-driving cars

Review

- How do write indefinite loops in Python?
 - Why are they called indefinite loops?
- What are two ways to think about **while** loops?
- Which are more powerful: **for** loops or **while** loops?

Flipping Coins

- Problem: How many flips does it take to get 3 consecutive heads?
 - How can we simulate flipping a coin?
- Recap:
 - Have the `game` module
 - `flipCoin()` and constants for HEADS and TAILS

`game.py`
`consecutiveHeads.py`

TEXT PROCESSING

Motivation: Text Processing

- Mostly focused on numbers so far
 - A little on graphics
- We can manipulate text to do useful work
 - Web search: finding most relevant documents to a query
 - Analyzing web logs (who is looking at my web page?)
 - Many, many others
- **Today's Focus:** the **str** data type and what you can do with them

Feb 15, 2019

Sprenkle - CSCI111

5

Strings: **str**

- Used for text
- Indicated by double quotes "" or single quotes '
 - In general, I'll use double quotes
 - Empty string: "" or ''
- Use triple quotes """ for strings that go across multiple lines

```
"""This string  
is long.  
Like, really, really long"""
```

Feb 15, 2019

Sprenkle - CSCI111

6

STRING OPERATIONS

Feb 15, 2019

Sprenkle - CSCI111

7

String Operations

Operand	Syntax	Meaning
+	<code>str1 + str2</code>	Concatenate two strings into one string
*	<code>str * num</code>	Concatenate string num times

- Examples:

- `"I feel " + "sleepy"`

- Evaluates to `"I feel sleepy"`

- `"Oops! " * 3`

- Evaluates to `"Oops! Oops! Oops! "`

Feb 15, 2019

Sprenkle - CSCI111

Recall lab 0

8

String Comparisons

- Same operations as with numbers:

➤ ==, !=
➤ <, <= } Alphabetical comparison
➤ >, >=

- Use in conditions in **if** statements

```
if courseChoice == "CSCI111":  
    print("Good choice!")  
else:  
    print("Maybe next semester")
```

Feb 15, 2019

Sprenkle - CSCI111 `string_compare.py` 9

Strings

- A *sequence* of one-character strings

➤ Example:

`band = "The Beatles "`

characters

'T'	'h'	'e'	' '	'B'	'e'	'a'	't'	'l'	'e'	's'
0	1	2	3	4	5	6	7	8	9	10

Start at 0

index or
position of
characters

End at `len(band)-1`

Length of the string: 11

Built-in function: `len(string)`
to find length of a string

Feb 15, 2019

Sprenkle - CSCI111

10

Iterating Through a String

- Use a **for** loop to iterate through *characters* in a string

string of length 1

```
for char in string:  
    print(char)
```

➤ Read as “for each character in the string”

Substrings Operator: []

- Look at a particular character in the string
 - Syntax: `string[<integer_expression>]`
 - [Positive value]: index of character
 - [Negative value]: count backwards from end
- Examples:
 - `<sequence>[0]` returns the first element/char
 - `<sequence>[-1]` returns the last element/char

We will deal with sequences
beyond strings later.

Examples in interpreter

Substrings Operator: []

- Look at a particular character in the string
 - Syntax: `string[<integer_expression>]`
- Examples with `band = "The Beatles"`

T	h	e		B	e	a	t	l	e	s
0	1	2	3	4	5	6	7	8	9	10

Expression	Result
<code>band[0]</code>	
<code>band[3]</code>	
<code>band[len(band)]</code>	
<code>band[len(band)-1]</code>	
<code>band[-1]</code>	

Feb 19

13

Substrings Operator: []

- Look at a particular character in the string
 - Syntax: `string[<integer expression>]`
- Examples with `band = "The Beatles"`

T	h	e		B	e	a	t	l	e	s
0	1	2	3	4	5	6	7	8	9	10

Expression	Result
<code>band[0]</code>	"T"
<code>band[3]</code>	" "
<code>band[len(band)]</code>	IndexError
<code>band[len(band)-1]</code>	"s"
<code>band[-1]</code>	"s"

Feb 19

14

Broader Issue: Self-Driving Cars

Andrew
Bobby
Callie
Ellis
Karel

Cat
Charlotte
Nate
Matt

Dan
Giovanni
Kassi
Melissa
Natalie

Danielle
Hayden
Jake
Jenna

Alice
August
James
Laurie
Mike

Feb 15, 2019

Sprenkle - CSCI111

15

Broader Issue: Self-Driving Cars

- Self-driving cars: love 'em or loathe 'em
 - As a passenger?
 - As a driver (or passenger) in another car?
 - As a pedestrian?
- What are the tradeoffs of self-driving cars?
 - What guarantees about the cars would you want from the company/government?
 - Are there situations that would be particularly difficult for software to handle that a person would be better equipped to handle?
- What should the next DARPA challenge be?
- Can ethical choices be automated?

Feb 15, 2019

Sprenkle - CSCI111

16

Midterm Grade Calculation

- 50% - Exam 1
- 50% - Labs

Feb 15, 2019

Sprenkle - CSCI111

17

Exam 1

- Reflection
 - What strategies did you use to study?
 - What did you do well on? What did you miss?
 - What strategies should you keep? What should change?

- Stats

	Section			
	A	B	C	Total
Average	84.93	73.73	84.84	85.35
Median	84.88	77.78	89.38	87.75

Feb 15, 2019

Sprenkle - CSCI111

18

Course Grade Overview

- (34%) Programming projects
- (30%) Two hourly exams
- (20%) A comprehensive final exam
- (7%) Writeups and discussions of Broader Issues
- (4%) Interactive textbook – prelabs
- (5%) Participation and attendance

Feb 25, 2019

Sprenkle - CSCI111

19

Looking Ahead

- Lab 6 Prep Assignment: Tuesday

Feb 15, 2019

Sprenkle - CSCI111

20