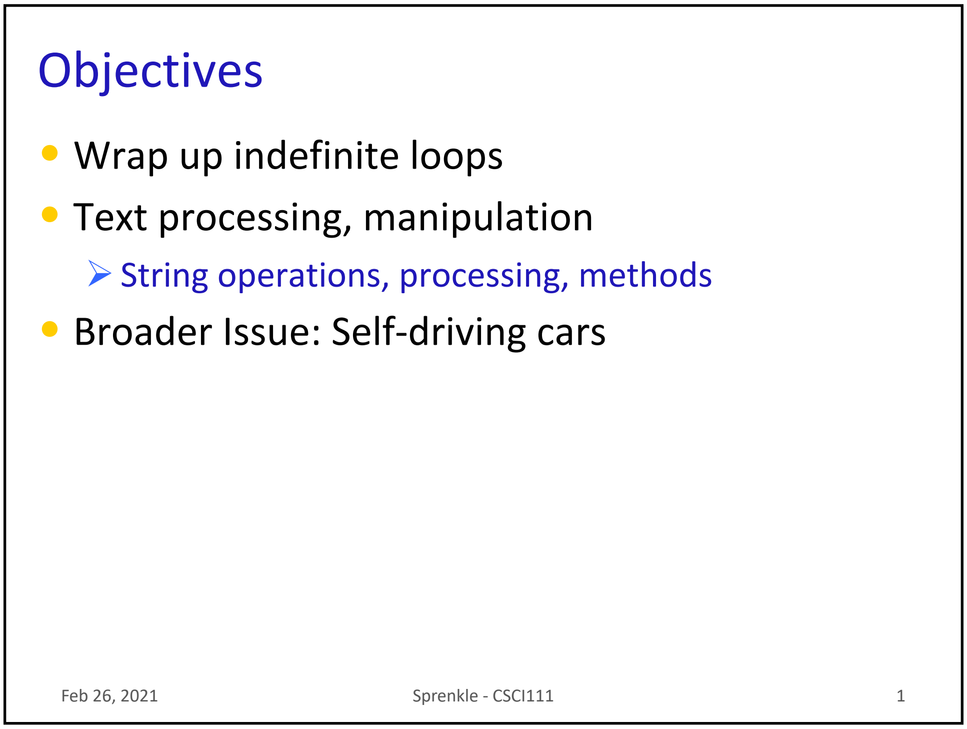
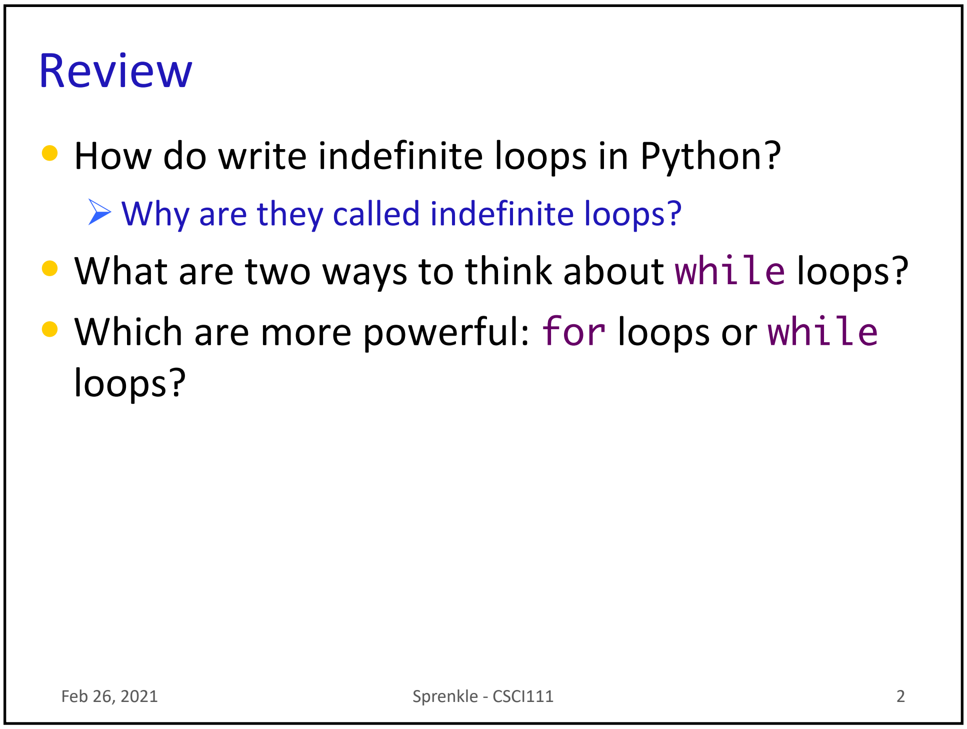


Objectives

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



Review

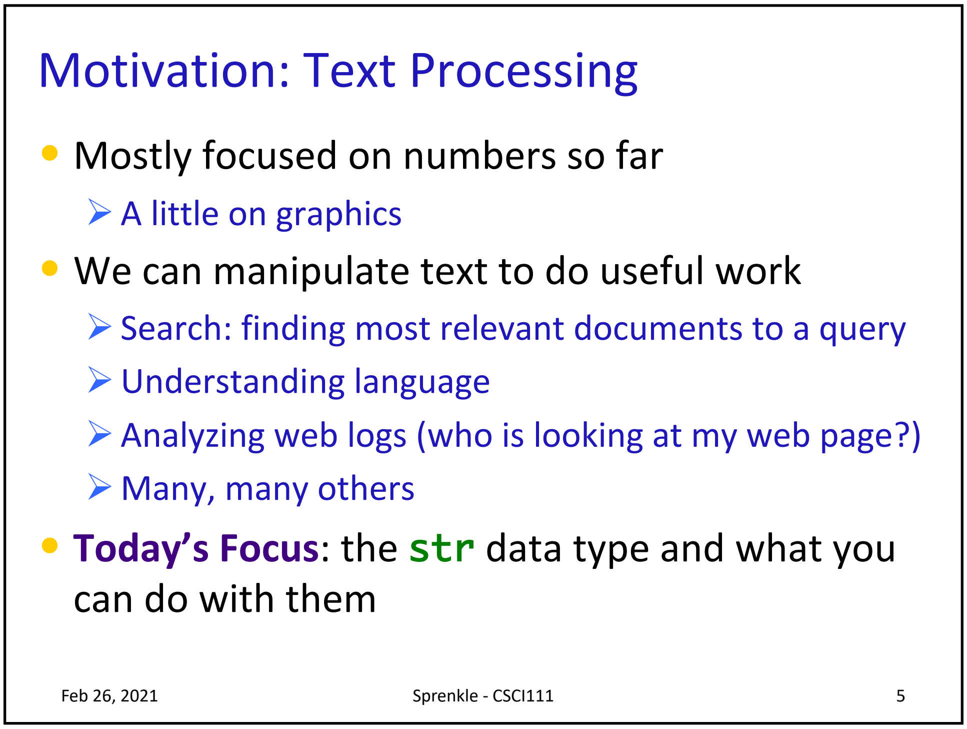
- How do write indefinite loops in Python?
  - Why are they called indefinite loops?
- What are two ways to think about `while` loops?
- Which are more powerful: `for` loops or `while` loops?

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`
  - Wrote solution with twist on sentinel design pattern
- Now: write solution using a `while True` loop and `break`

`consecutiveHeads.py`
`game.py`

TEXT PROCESSING

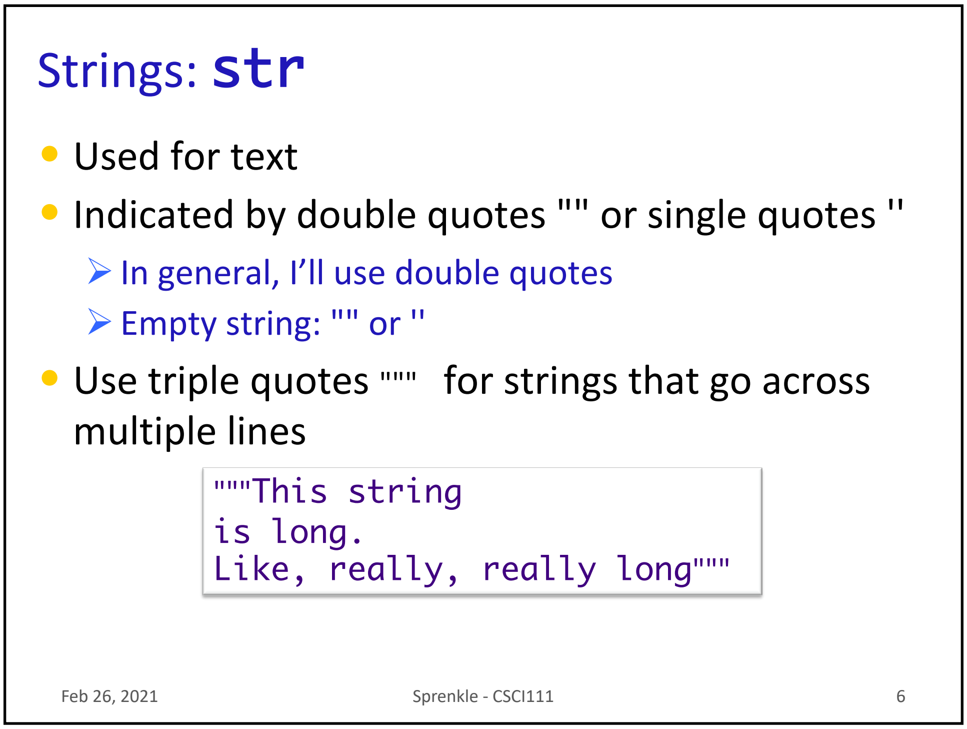


Motivation: Text Processing

- Mostly focused on numbers so far
  - A little on graphics
- We can manipulate text to do useful work
  - Search: finding most relevant documents to a query
  - Understanding language
  - 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 26, 2021 Sprenkle - CSCI111 5

5



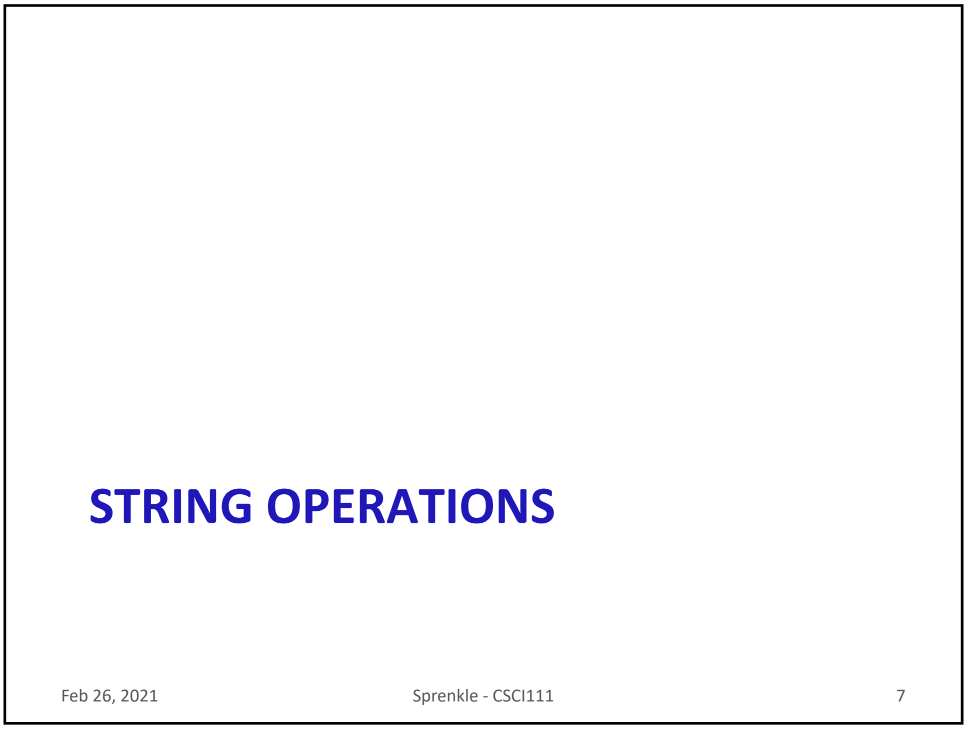
Strings: `str`

- 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

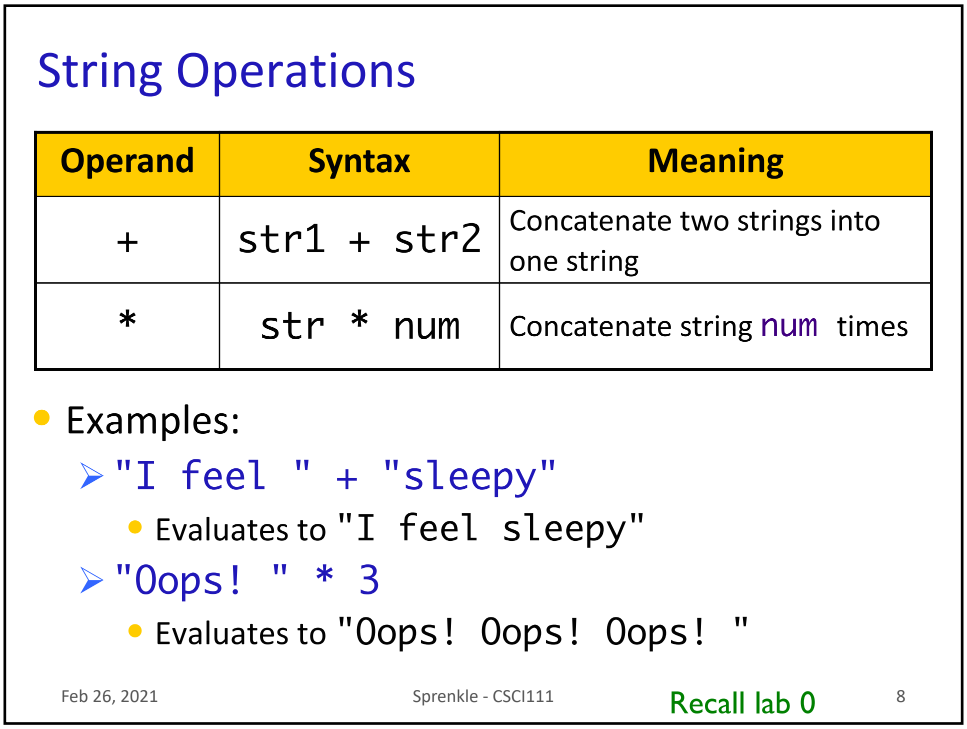
```
"""This string
is long.
Like, really, really long"""
```

Feb 26, 2021 Sprenkle - CSCI111 6

6



STRING OPERATIONS

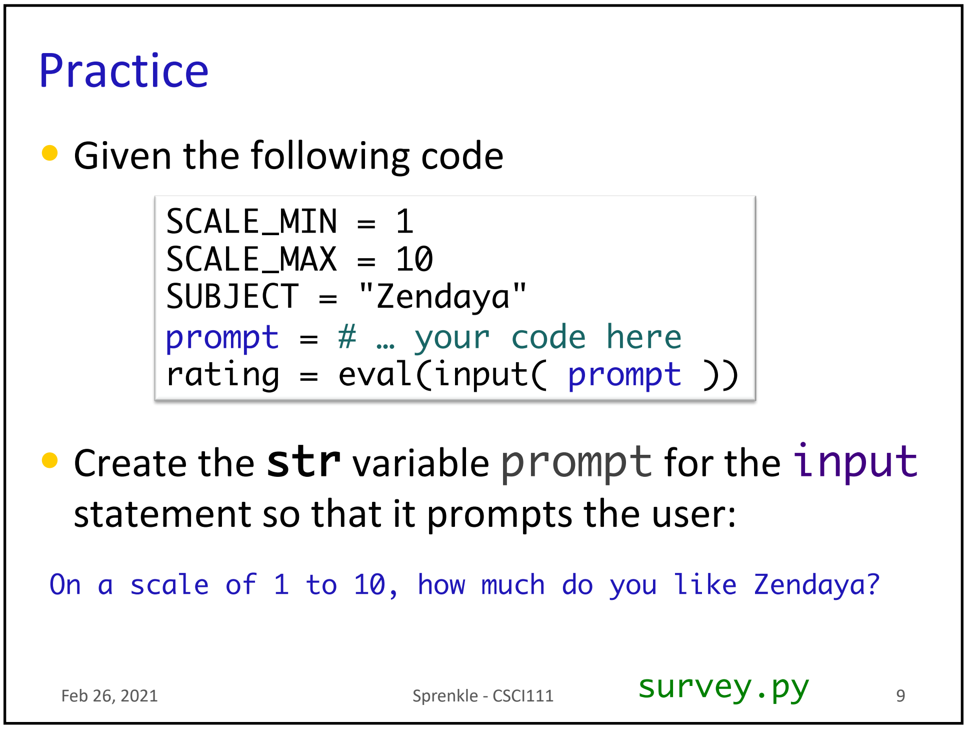


String Operations

Operand	Syntax	Meaning
+	`str1 + str2`	Concatenate two strings into one string
*	`str * num`	Concatenate string `num` times

- Examples:
  - `"I feel " + "sleepy"`
    - Evaluates to `"I feel sleepy"`
  - `"Oops! " * 3`
    - Evaluates to `"Oops! Oops! Oops! "`

Recall lab 0



Practice

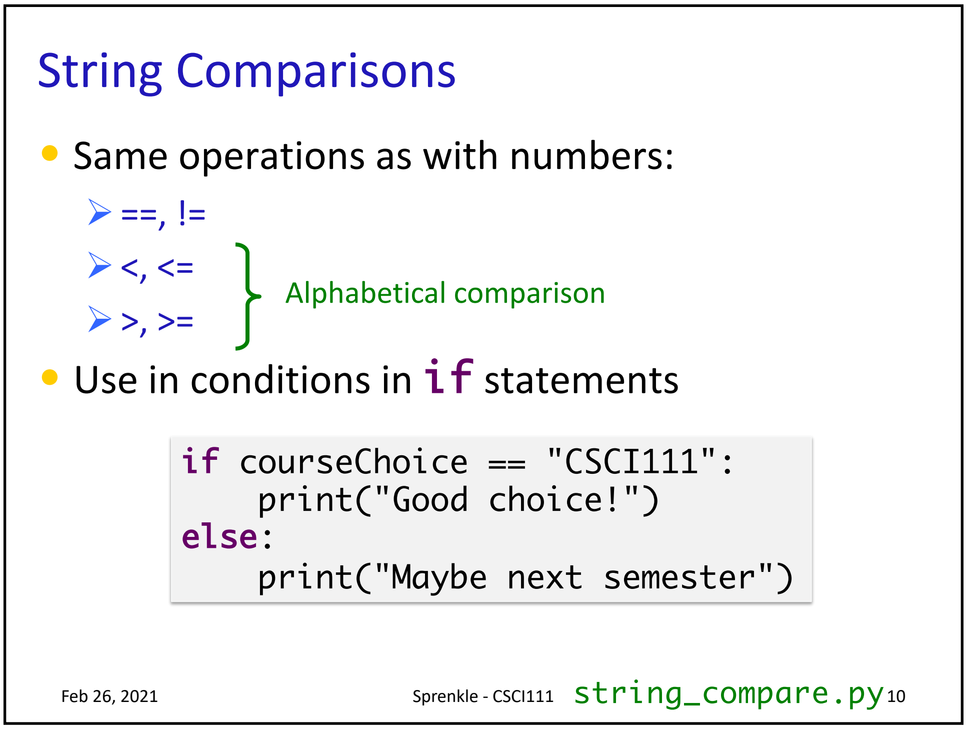
- Given the following code

```
SCALE_MIN = 1
SCALE_MAX = 10
SUBJECT = "Zendaya"
prompt = # … your code here
rating = eval(input( prompt ))
```

- Create the `str` variable `prompt` for the `input` statement so that it prompts the user:

```
On a scale of 1 to 10, how much do you like Zendaya?
```

survey.py



String Comparisons

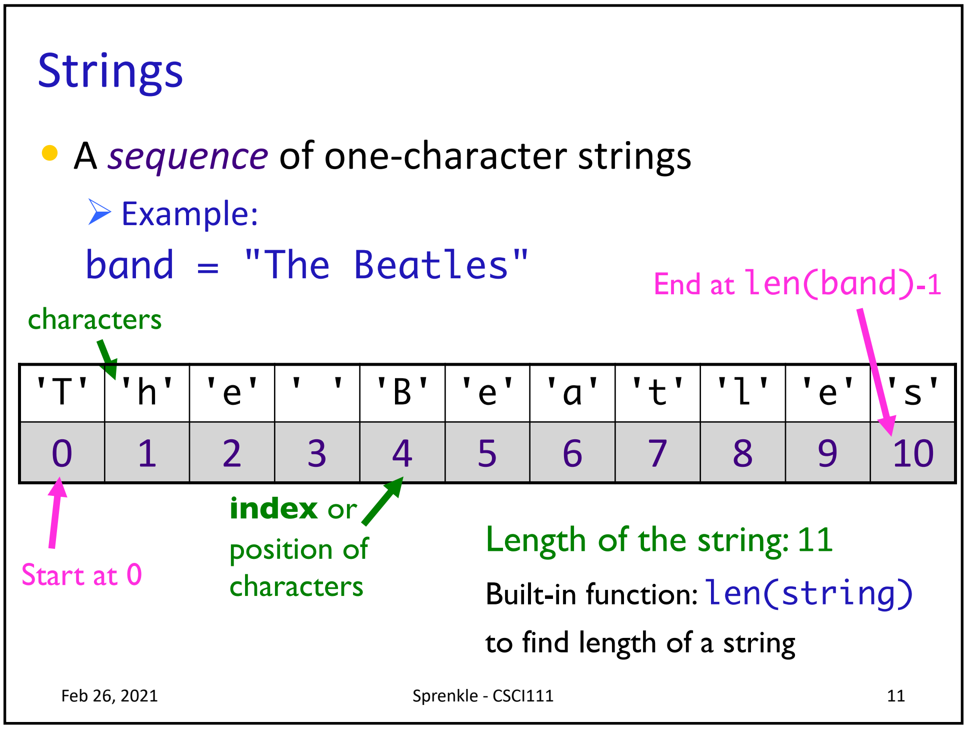
- Same operations as with numbers:
  - ==, !=
  - <, <=
  - >, >=

Alphabetical comparison

- Use in conditions in `if` statements

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

string_compare.py



Strings

- A *sequence* of one-character strings
  - Example:

```
band = "The Beatles"
```

End at len(band)-1

characters

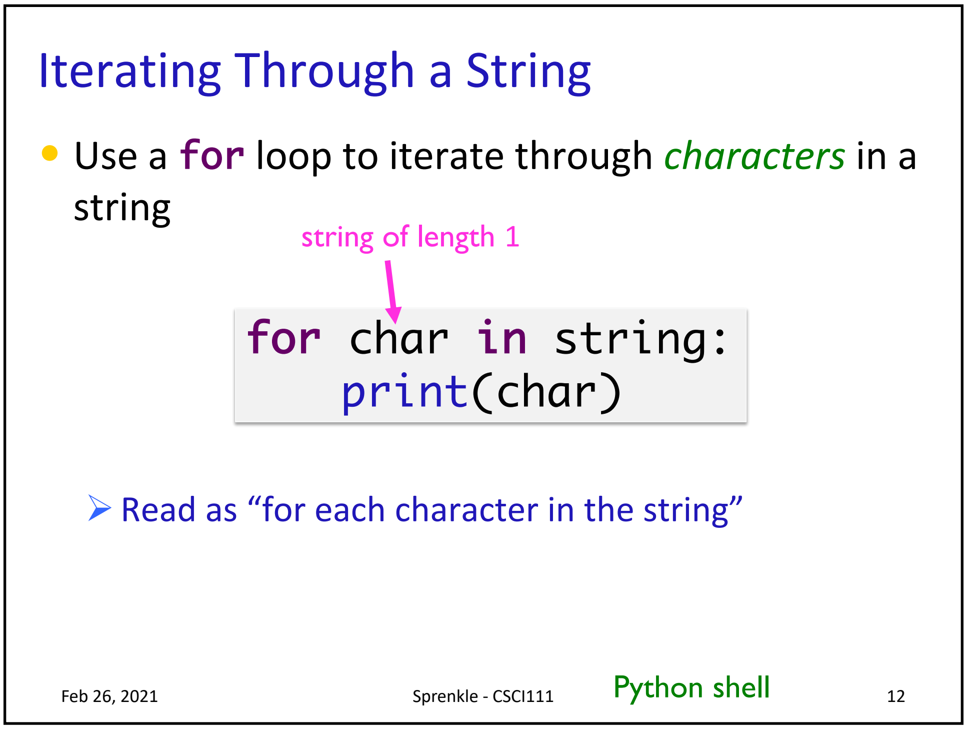
'T'	'h'	'e'	' '	'B'	'e'	'a'	't'	'l'	'e'	's'
0	1	2	3	4	5	6	7	8	9	10

Start at 0

index or position of characters

Length of the string: 11

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



Iterating Through a String

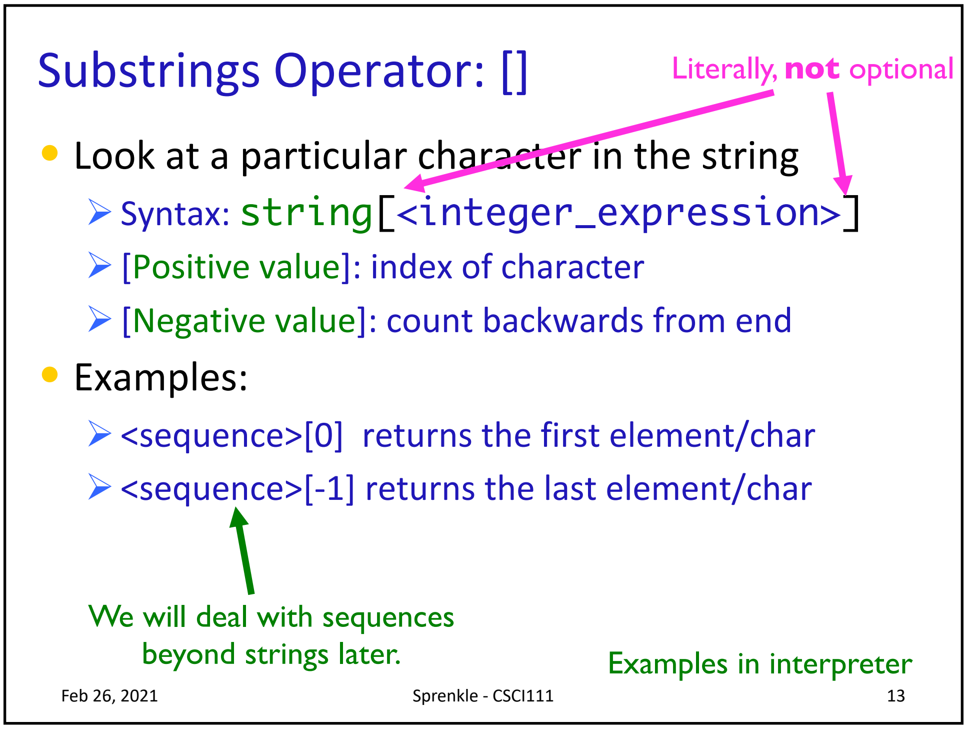
- Use a **for** loop to iterate through *characters* in a string

string of length 1

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

- Read as "for each character in the string"

Python shell



Substrings Operator: []

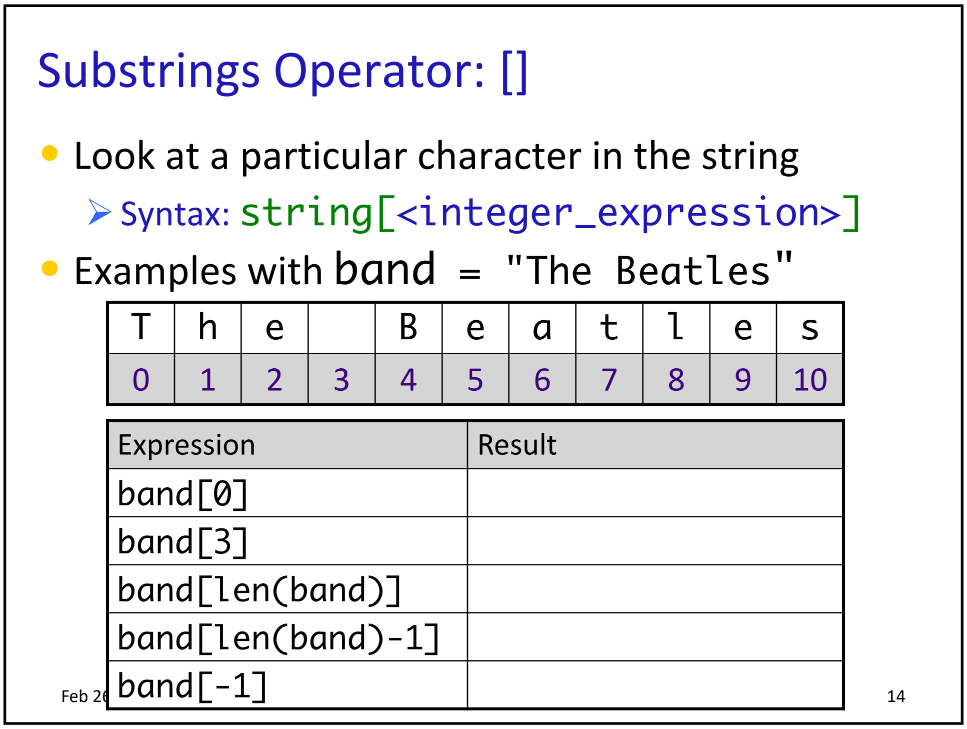
Literally, **not** optional

- 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

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
band[0]	
band[3]	
band[len(band)]	
band[len(band)-1]	
band[-1]	

14

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
`band[0]`	`"T"`
`band[3]`	`" "`
`band[len(band)]`	`IndexError`
`band[len(band)-1]`	`"s"`
`band[-1]`	`"s"`

Broader Issue: Self-Driving Cars

- Why do I still assign an article from 2007?
- 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 about the ethics of tough decisions?
- Consider the development process to create self-driving cars
  - What are the steps? What makes it hard?
- What should the next DARPA challenge be?

Broader Issue: Self-Driving Cars

- Development Process
  - Writing code
    - Breaking complex problem into smaller pieces
    - Iteration
  - Testing code – safety-critical system
    - Iteration
  - Dealing with changing conditions
    - Emphasis on “soft”, easily changed code
  - Dealing with other systems
    - Need a common API

Looking Ahead: Next Week

- Non-Lab 6 Prep Assignment: Tuesday
  - Focus on Strings (Chapter 9)
  - Skipping over while loops (Chapter 8) until the next pre lab
- Non-Lab 6: a few problems due Friday
- No broader issue because short week

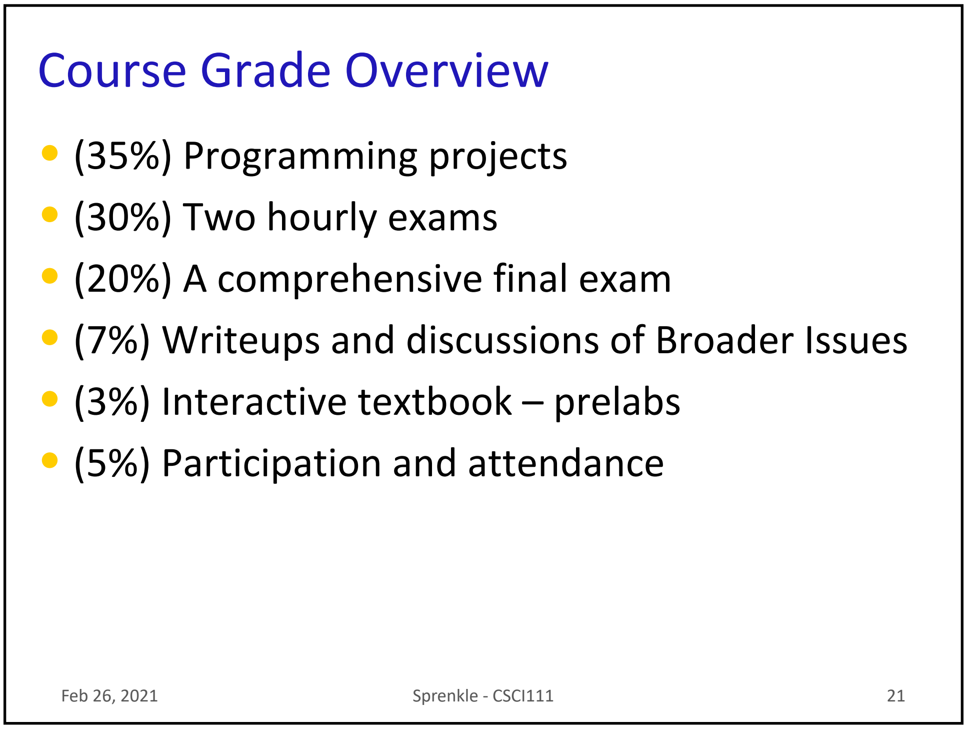
Midterm Grade Calculation

- 50% - Exam 1
- 50% - Labs

Exam 1

- Reflection
  - What strategies did you use to study?
    - What strategies did you use in the course in general?
  - 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	92.73	74.31	91.64	90.41
Median	93.02	72.22	93.75	91.25



Course Grade Overview

- (35%) Programming projects
- (30%) Two hourly exams
- (20%) A comprehensive final exam
- (7%) Writeups and discussions of Broader Issues
- (3%) Interactive textbook – prelabs
- (5%) Participation and attendance