

Objectives

- A new data type: Lists
- Broader Issue: self-driving cars

Feb 19, 2016

Sprengle - CSC1111

1

Review

- What are the various things we can do with strings?

Feb 19, 2016

Sprengle - CSC1111

2

Sequences of Data

- Sequences so far ...
 - `str`: sequence of characters
 - `range`: generator (sequence of numbers)
- We commonly group a sequence of data together and refer to them by one name
 - Days of the week: Sunday, Monday, Tuesday, ...
 - Months of the year: Jan, Feb, Mar, ...
 - Shopping list
- Can represent this data as a **list** in Python
 - Similar to **arrays** in other languages

Feb 19, 2016

Sprengle - CSC1111

3

Lists: A Sequence of Data Elements

element	daysInWeek					
"Sun"	"Mon"	"Tue"	"Wed"	"Thu"	"Fri"	"Sat"
0	1	2	3	4	5	6

Position/
index
in the list

`len(daysInWeek)` is 7

- Elements in lists can be *any* data type

What does this look similar to, in structure?

Feb 19, 2016

Sprengle - CSC1111

4

Example Lists in Python

- Empty List: `[]`
- List of `str`s:
 - `daysInWeek=["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]`
- List of `float`s
 - `highTemps=[60.4, 70.2, 63.8, 55.7, 54.2]`
- Lists can contain >1 type
 - `wheelOfFortune=[250, 1000, "Bankrupt", "Free Play"]`

Syntax for list: `[]`
How different from accessing a string?

Feb 19, 2016

Sprengle - CSC1111

5

Benefits of Lists

- Group related items together
 - Instead of creating separate variables
 - `sunday = "Sun"`
 - `monday = "Mon"`
- Convenient for dealing with large amounts of data
 - Example: could keep all the temperature data in a list if needed to reuse later
- Functions and methods for handling, manipulating lists

Feb 19, 2016

Sprengle - CSC1111

6

List Operations

Similar to operations for strings

Concatenation	<code><seq> + <seq></code>
Repetition	<code><seq> * <int-expr></code>
Indexing	<code><seq>[<int-expr>]</code>
Length	<code>len(<seq>)</code>
Slicing	<code><seq>[:]</code>
Iteration	<code>for <var> in <seq>:</code>
Membership	<code><expr> in <seq></code>

Feb 19, 2016

Sprengle - CSC111

7

Lists: A Sequence of Data Elements

daysInWeek						
"Sun"	"Mon"	"Tue"	"Wed"	"Thu"	"Fri"	"Sat"
0	1	2	3	4	5	6

Position in the list len(daysInWeek) is 7

- `<listname>[<int_expr>]`
 - Similar to accessing characters in a string
 - `daysInWeek[-1]` is "Sat"
 - `daysInWeek[0]` is "Sun"

Feb 19, 2016

Sprengle - CSC111

8

Iterating through a List

- Read as
 - For every element in the list ...

An item in the list list object

```
for item in list:
    print(item)
```

Iterates through items in list

- Equivalent to

```
for x in range(len(list)):
    print(list[x])
```

Iterates through positions in list

Feb 19, 2016

Sprengle - CSC111

daysOfWeek.py

9

Practice

- Get the *list* of weekend days from the days of the week list
 - `daysInWeek=["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]`

Feb 19, 2016

Sprengle - CSC111

10

Practice

- Get the *list* of weekend days from the days of the week list

```
➤ daysInWeek=["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]

➤ weekend = daysInWeek[:1] + daysInWeek[-1:]
```

Gives back a *list* or

```
➤ weekend = [daysInWeek[0]] + [daysInWeek[-1]]
```

Gives back an element of list, which is a *str*¹¹

Feb 19, 2016

Sprengle - CSC111

¹¹

Membership

- **Check if a list contains an element**
- Example problem
 - `enrolledstudents` is a list of students who are enrolled in the class
 - Want to check if a student who attends the class is enrolled in the class

```
if student not in enrolledstudents:
    print(student, "is not enrolled")
```

Feb 19, 2016

Sprengle - CSC111

12

Example Code

```
friends = ["Alice", "Bjorn", "Cayman", "Duanphen", \
          "Esfir", "Farah"]

for name in friends:
    print("I know " + name + ".")
    print(name, "is a friend of mine.")

print("Those are the people I know.")
```

friends.py

Feb 19, 2016

Sprengle - CSCI111

13

List Methods

Method Name	Functionality
<list>.append(x)	Add element <i>x</i> to the end
<list>.sort()	Sort the list
<list>.reverse()	Reverse the list
<list>.index(x)	Returns the index of the first occurrence of <i>x</i> , Error if <i>x</i> is not in the list
<list>.insert(i, x)	Insert <i>x</i> into list at index <i>i</i>
<list>.count(x)	Returns the number of occurrences of <i>x</i> in list
<list>.remove(x)	Deletes the first occurrence of <i>x</i> in list
<list>.pop(i)	Deletes the <i>i</i> th element of the list and returns its value

Note: methods do **not** return a *copy* of the list ...

Feb 19, 2016

Sprengle - CSCI111

14

Lists vs. Strings

- Strings are **immutable**
 - Can't be mutated?
 - Err, can't be modified/changed
- Lists are **mutable**
 - Can be changed
 - Changes how we call/use methods

```
groceryList=["milk", "eggs", "bread", "Doritos", "0J", \
            "sugar"]
groceryList[0] = "skim milk"
groceryList[3] = "popcorn"
groceryList is now ["skim milk", "eggs", "bread", \
                  "popcorn", "0J", "sugar"]
```

Feb 19, 2016

Sprengle - CSCI111

15

Practice in Interactive Mode

- list = [7,8,9]
- string = "abc"
- list[1]
- string[1]
- string.upper()
- list.reverse()
- string
- list
- string = string.upper()
- list = list.reverse()
- string
- list

Feb 19, 2016

Sprengle - CSCI111

16

Special Value: None

- Special value we can use
 - E.g., Return value from function when there is an error
- Similar to **null** in Java
- If you execute

```
list = list.sort()
print(list)
```

 - Prints None because `list.sort()` does not return anything

Feb 19, 2016

Sprengle - CSCI111

17

Fibonacci Sequence

- Goal: Solve using *list*
- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$
- Example sequence: 1, 1, 2, 3, 5, 8, 13, 21, ...

Feb 19, 2016

Sprengle - CSCI111

18

Fibonacci Sequence

- Create a list of the 1st 15 Fibonacci numbers

➤ $F_0=F_1=1$; $F_n=F_{n-1}+F_{n-2}$

Grow list as we go

```
fibs = [] # create an empty list
fibs.append(1) # append the first two Fib numbers
fibs.append(1)
for x in range(2, 16): # compute the next 13 nums
    newfib = fibs[x-1]+fibs[x-2]
    fibs.append(newfib)

print(fibs) # print out the list
```

Feb 19, 2016

Sprengle - CSCI111

fibs.py

19

Fibonacci Sequence

- Create a list of the 1st 15 Fibonacci numbers

➤ $F_0=F_1=1$; $F_n=F_{n-1}+F_{n-2}$

- Create whole list
- Update values

```
fibs = [0]*15 # creates a list of size 15,
              # containing all 0 s
fibs[0] = 1
fibs[1] = 1
for x in range(2, len(fibs)):
    newfib = fibs[x-1]+fibs[x-2]
    fibs[x] = newfib

for num in fibs: # print each num on sep line
    print(num)
```

Feb 19, 2016

Sprengle - CSCI111

fibs2.py

20

Practice: Wheel of Fortune

- Modify to keep track of previous guesses
 - If user made that guess before, print message
- What are the data types of the data we're modeling?

Feb 19, 2016

Sprengle - CSCI111

21

Practice: Wheel of Fortune

- Model the wheel
 - Money
 - Bankruptcy, lose a turn, free spin
- Simulate spinning the wheel

Feb 19, 2016

Sprengle - CSCI111

22

Broader Issue: Self-Driving Cars

Abdur
Austin F
Corson
Eric
Gunnar

Alicia
Allie
JT
Michael

Bennett
Emily
Ethiopia
Viktor

Austin J
Honor
Margaret
Rachel R

Alice
Collin
Josie
Madhav

Andrew
Holly
Max
Rachel B

Chris
George
Lily
Stuart

Feb 19, 2016

Sprengle - CSCI111

23

Broader Issues Discussion

- What are the pros and cons to automated cars?
- We talk a lot about testing in this class.
 - When would you consider that the car has been tested enough?
 - What are good test cases for automated cars?
- Would you feel safe/safer with a self-driving car or an Uber/Lyft driver?
 - Does your feeling depend on anything else?
- What should the next DARPA Challenge be?

Feb 19, 2016

Sprengle - CSCI111

24

Looking Ahead

- Extra credit opportunities
 - Review CS articles (similar to broader issues)