

Objectives

- A new data type: Lists
- Broader Issue: cryptography

Reminders:

- You can and should practice programming regularly
- Use <http://pythontutor.com/>
- You can install Python3 on your own computer, if it's not already

Feb 17, 2017

Sprengle - CSC1111

1

Review

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

Feb 17, 2017

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 17, 2017

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 17, 2017

Sprengle - CSC1111

4

Example Lists in Python

- Empty List: `[]`
- List of `strs`:
 - `daysInWeek=["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]`
- List of `floats`
 - `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 17, 2017

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 17, 2017

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 17, 2017

Sprengle - CSC1111

7

Lists: A Sequence of Data Elements

element → 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 17, 2017

Sprengle - CSC1111

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

- Output equivalent to

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

Iterates through
positions in list

Feb 17, 2017

Sprengle - CSC1111

daysOfWeek.py

9

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 17, 2017

Sprengle - CSC1111

10

Practice

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

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

Feb 17, 2017

Sprengle - CSC1111

11

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:]

or

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

Gives back a *list*

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

Feb 17, 2017

Sprengle - CSC1111

12

Membership

- **Check if a list contains an element**
- Example usage
 - `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 17, 2017

Sprenkle - CSC1111

13

List Methods

| Method Name | Functionality |
|--|--|
| <code><list>.append(x)</code> | Add element <i>x</i> to the end |
| <code><list>.sort()</code> | Sort the list |
| <code><list>.reverse()</code> | Reverse the list |
| <code><list>.index(x)</code> | Returns the index of the first occurrence of <i>x</i> , Error if <i>x</i> is not in the list |
| <code><list>.insert(i, x)</code> | Insert <i>x</i> into list at index <i>i</i> |
| <code><list>.count(x)</code> | Returns the number of occurrences of <i>x</i> in list |
| <code><list>.remove(x)</code> | Deletes the first occurrence of <i>x</i> in list |
| <code><list>.pop(i)</code> | 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 17, 2017

Sprenkle - CSC1111

14

Lists vs. Strings

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

```
groceryList=["milk", "eggs", "bread", "Doritos", "OJ", \  
            "sugar"]
```

```
groceryList[0] = "skim milk"  
groceryList[3] = "popcorn"
```

```
groceryList is now ["skim milk", "eggs", "bread", \  
                  "popcorn", "OJ", "sugar"]
```

Feb 17, 2017

Sprenkle - CSC1111

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 17, 2017

Sprenkle - CSC1111

16

Broader Issues Discussion

- What is cryptography?
 - What is public key cryptography?
 - Why is cryptography useful?
- Who is Alan Turing?
- Why are computer scientists involved in politics?
 - Has this class informed your politics? (tread lightly)