Compsci 101
Introduction
Live Lecture

Susan Rodger
Nicki Washington
February 11, 2021

<table>
<thead>
<tr>
<th>sum(lst)</th>
<th>sum of the elements in lst</th>
</tr>
</thead>
<tbody>
<tr>
<td>max(lst)</td>
<td>maximum value of lst</td>
</tr>
<tr>
<td>min(lst)</td>
<td>minimum value of lst</td>
</tr>
<tr>
<td>.append(elm)</td>
<td>Mutates the list by adding elm to the end of the list</td>
</tr>
<tr>
<td>.count(elm)</td>
<td>Number of times see elm in the list</td>
</tr>
</tbody>
</table>

Announcements

• Assignment 1 Totem due today, Feb 11, 11:30pm
  • Also REFLECT Form due same time
  • Get one grace day, but no consulting hours on Friday
• APT-2 out Tues, Feb 16
• Exam on Tuesday!, Feb 16
• Exam 1 Question session – Sunday Feb 14, 4pm
  • Use lecture zoom link, recorded
  • Bring questions, if no questions, it will be short!

Exam 1 Rules

• This is your own work, no collaboration
• Open book, Open notes
• Do not search for answers on the internet
• Do not type in code where it can be compiled and run
  • Do not use Pycharm, textbook code boxes, Python tutor or any other place the code can be run
• Do not talk to anyone about the exam during the exam, and until it is handed back!

PFTD

• Lists continued
• String methods and more
• Exam 1
Exam 1 Logistics

- Take on Tues. Feb 16 between 7am and 11pm
- You pick the start time
  - Must start by 9:15pm
- You get 1 hour 45 min
  - Longer if you have accommodations
- Once you start, your timer starts and you must finish in 1 hour, 45 minutes
- You cannot pause the timer

Exam 1 Logistics (2)

- Go to Gradescope to start
  - login with your netid
  - select the CompSci 101 Exam site
    - Different site than where you turn in assignments
- Click on Exam 1 to start
- Gradescope saves answers as you type them in
  - Type 4 spaces to indent code
- Disconnected? Just log back in to Gradescope
- Question? Post a private post on Piazza

Don't go to Gradescope site until you are ready to start!

You click it, you have started!

We do not restart it!

WOTO-1 Cloning
List Concatenation Steps

1. Calculate the **length** of the new list
2. _Create_ list of that length
3. _Copy_ values from first list
4. _Copy_ values from second list
5. Assign the variable to the new list

Concatenation:
length, create, copy, copy, assign

```python
1 lst0 = [1, 2]
2 lst1 = [3, 4, 5]
3 lst2 = lst0 + lst1
```

WOTO-2 – Mutable

List Mutation: `.append(…)`

- `.append()` – list function that adds element to end of list
  - Mutates list to left of “.”
  - “.” – call function to the right of the dot on the thing to the left of the dot (LEFT.RIGHT)

Same list!
Strings have functions too!

**TYPE_STRING.FUNCTION(PARAMETERS)**

- “.” means apply function to what is on the left

Example:

- `'one fish two fish'.split()` returns a list
  - `['one', 'fish', 'two', 'fish']`

Strings have functions too!

**TYPE_STRING.FUNCTION(PARAMETERS)**

- “.” means apply function to what is on the left

Example:

- `'red fish blue fish'.split()` returns a list
  - `['red', 'fish', 'blue', 'fish']`

String’s join(…)

- **TYPE_STRING.join(SEQ_OF_STRINGS)**
  - Opposite of .split()
  - Creates string from sequence's items separated by the string to the left of join

Example:

- `'ish'.join(['f','w','d','end'])`
  - `fishwishdishend`
### String’s `join(...)`

- **`TYPE_STRING.join(SEQ_OF_STRINGS)`**
  - Opposite of `.split()`
  - Creates string from sequence's items separated by the string to the left of `join`
- `’ish’.join([‘f’,’w’,’d’,’end’])`
- `’fishwishdishend’`

### More Methods

**String**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>.find(s)</code></td>
<td>index of first occurrence of s</td>
</tr>
<tr>
<td><code>.rfind(s)</code></td>
<td>index of last occurrence of s (from Right)</td>
</tr>
<tr>
<td><code>.upper()</code></td>
<td>uppercase/lowercase version of string</td>
</tr>
<tr>
<td><code>.lower()</code></td>
<td>remove leading/trailing whitespace</td>
</tr>
<tr>
<td><code>.strip()</code></td>
<td>remove leading/trailing whitespace</td>
</tr>
<tr>
<td><code>.count(s)</code></td>
<td>number of times see s in string</td>
</tr>
<tr>
<td><code>.startswith(s)</code></td>
<td>bool of whether the string begins with s</td>
</tr>
<tr>
<td><code>.endswith(s)</code></td>
<td>bool of whether the string ends with s</td>
</tr>
</tbody>
</table>

**List**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>.sum(lst)</code></td>
<td>sum of the elements in lst</td>
</tr>
<tr>
<td><code>.max(lst)</code></td>
<td>maximum value of lst</td>
</tr>
<tr>
<td><code>.min(lst)</code></td>
<td>minimum value of lst</td>
</tr>
<tr>
<td><code>.append(elm)</code></td>
<td>Mutates the list by adding elm to the end of the list</td>
</tr>
<tr>
<td><code>.count(elm)</code></td>
<td>Number of times see elm in the list</td>
</tr>
</tbody>
</table>