CSc 110, Spring 2017

Lecture 15: lists

Adapted from slides by Marty Stepp and Stuart Reges

"The machine learning algorithm wants to know if we'd like a dozen wireless mice to feed the Python book we just bought."
Can we solve this problem?

- Consider the following program (input underlined):

<table>
<thead>
<tr>
<th>Day</th>
<th>High Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>46</td>
</tr>
<tr>
<td>7</td>
<td>53</td>
</tr>
</tbody>
</table>

Average temp = 44.6

4 days were above average.
Why the problem is hard

• We need each input value twice:
  • to compute the average (a cumulative sum)
  • to count how many were above average

• We could read each value into a variable... but we:
  • don't know how many days are needed until the program runs
  • don't know how many variables to declare

• We need a way to hold a sequence of values (and of course a way to reference them...)
Lists

- **list**: a type that holds a sequence of zero or more values.
  - **element**: One value in a list.
  - **index**: A 0-based integer used to access an element from an list.

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>12</td>
<td>49</td>
<td>-2</td>
<td>26</td>
<td>5</td>
<td>17</td>
<td>-6</td>
<td>84</td>
<td>72</td>
<td>3</td>
</tr>
</tbody>
</table>

- element 0
- element 4
- element 9
List initialization

name = [value, value, ... value]

• Example:
  numbers = [12, 49, -2, 26, 5, 17, -6]

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>12</td>
<td>49</td>
<td>-2</td>
<td>26</td>
<td>5</td>
<td>17</td>
<td>-6</td>
</tr>
</tbody>
</table>

• An alternate form when the values are the same:

name = [value] * count

• Example:
  numbers = [0] * 4

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Accessing elements

name[index]  # access
name[index] = value  # modify

• Example:
  numbers = [0] * 2
  numbers[0] = 27
  numbers[1] = -6

  print(numbers[0])
  if (numbers[1] < 0):
    print("Element 1 is negative.")
    index  0   1
    value  27  -6
Accessing list elements

numbers = [0] * 8
numbers[0] = 3
numbers[1] = 99
numbers[2] = 6

x = numbers[0]
numbers[x] = 42
numbers[numbers[2]] = 11  # use numbers[2] as index

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>3</td>
<td>99</td>
<td>6</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>
Out-of-bounds

• Legal indexes to use []: between list's length and the list's length - 1.

• Note: this is just like strings
  • Reading or writing any index outside this range with [] will cause an `IndexError: list assignment index out of range`

• Example:
  ```python
data = [0] * 10
print(data[0])       # okay
print(data[9])       # okay
print(data[-20])     # error
print(data[10])      # error
```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Lists and `for` loops

• It is common to use `for` loops to access list elements.

```python
for i in range(0, 8):
    print(str(numbers[i]) + " ", end='')
print()  # output: 3 99 6 42 0 0 11 0
```

• Sometimes we assign each element a value in a loop.

```python
for i in range(0, 8):
    numbers[i] = 2 * i
```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>
len

• \texttt{len(list)} returns the number of elements in a list.

```python
for i in range(0, \texttt{len(numbers)}):
    \texttt{print(numbers[i] \+ \" \\+ \texttt{end=\"\")
# output: 0 2 4 6 8 10 12 14
```

• What value is produced by:
  • \texttt{len([10,20,30])}
  • \texttt{len([3,4] \* 2)
Lists and for loops

• You can use the `len` function to loop through a list

```python
counts = [1, 3, 6, 23, 43, 12]
for i in range(0, len(counts)):
    print(str(counts[i]) + " ", end='')
print()  # output: 1 3 6 23 43 12
```

• Or, you can also loop directly over lists, just as with strings

```python
counts = [1, 3, 6, 23, 43, 12]
for number in counts:
    print(str(number) + " ", end='')
print()  # output: 1 3 6 23 43 12
```
Weather question

• Use a list to solve the weather problem:

How many days' temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.
Weather answer

# Reads temperatures from the user, computes average and # days above average.

def main():
    days = int(input("How many days' temperatures? "))

    temps = [0] * days  # list to store days' temperatures
    sum = 0

    for i in range(0, days):  # read/store each day's temperature
        temps[i] = int(input("Day " + (i + 1) + "'s high temp: "))
        sum = sum + temps[i]
    average = sum / days

    count = 0  # see if each day is above average
    for i in range(0, days):
        if (temps[i] > average):
            count = count + 1

    # report results
    print("Average temp = " + str(average))
    print(str(count) + " days above average")
Weather question 2

• Modify the weather program to print the following output:

Type in a temperature or "done" to finish
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Day 7's high temp: done
Average temp = 44.6
4 days were above average.
### List functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>append(x)</td>
<td>Add an item to the end of the list. Equivalent to <code>a[len(a):] = [x]</code>.</td>
</tr>
<tr>
<td>extend(L)</td>
<td>Extend the list by appending all the items in the given list. Equivalent to <code>a[len(a):] = L</code></td>
</tr>
<tr>
<td>insert(i, x)</td>
<td>Inserts an item at a given position. i is the index of the element before which to insert, so <code>a.insert(0, x)</code> inserts at the front of the list.</td>
</tr>
<tr>
<td>remove(x)</td>
<td>Removes the first item from the list whose value is x. Errs if there is no such item.</td>
</tr>
<tr>
<td>pop(i)</td>
<td>Removes the item at the given position in the list, and returns it. a.pop() removes and returns the last item in the list.</td>
</tr>
<tr>
<td>clear()</td>
<td>Remove all items from the list.</td>
</tr>
<tr>
<td>index(x)</td>
<td>Returns the index in the list of the first item whose value is x. Errs if there is no such item.</td>
</tr>
<tr>
<td>count(x)</td>
<td>Returns the number of times x appears in the list.</td>
</tr>
<tr>
<td>sort()</td>
<td>Sort the items of the list</td>
</tr>
<tr>
<td>reverse()</td>
<td>Reverses the elements of the list</td>
</tr>
<tr>
<td>copy()</td>
<td>Return a copy of the list.</td>
</tr>
</tbody>
</table>
# Reads temperatures from the user, computes average and # days above average.
def main():
    print('Type in a temperature or "done" to finish')

    temps = []  # list to store days' temperatures
    sum = 0
    done = input("Day 1's high temp: ")
    day = 1

    while(done != "done"):
        # read/store each day's temperature
        done = int(done)
        sum = sum + done
        temps.append(done)
        done = input("Day " + str(day + 1) + "'s high temp: ")
        day = day + 1
    average = sum / day

    count = 0  # see if each day is above average
    for i in range(0, day - 1):
        if (temps[i] > average):
            count = count + 1

    # report results
    print("Average temp = " + str(average))
    print(str(count) + " days above average")
Weather question 3

• Modify the weather program to print the following output:

How many days’ temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.

Temperatures: [45, 44, 39, 48, 37, 46, 53]
Two coldest days: 37, 39
Two hottest days: 53, 48
# Reads temperatures from the user, computes average and # days above average.
def main():
    days = int(input("How many days' temperatures? ")
    temps = [0] * days  # list to store days' temperatures
    sum = 0
    for i in range(0, days):
        # read/store each day's temperature
        temps[i] = int(input("Day " + (i + 1) + "'s high temp: "))
        sum = sum + temps[i]
    average = sum / days
    count = 0  # see if each day is above average
    for i in range(0, days):
        if (temps[i] > average):
            count += 1
    # report results
    print("Average temp = " + str(average))
    print(str(count) + " days above average")
    print("Temperatures: " + str(temps))
    temps.sort()
    print("Two coldest days: " + str(temps[0]) + ", " + str(temps[1]))
    print("Two hottest days: " + str(temps[-1]) + ", " + str(temps[-2]))
"list mystery" problem

• **traversal**: A sequential processing of the elements of a list.

• What element values are stored in the following list?

\[
a = [1, 7, 5, 6, 4, 14, 11]
\]

```python
for i in range(0, len(a) - 1):
    if (a[i] > a[i + 1]):
        a[i + 1] = a[i + 1] * 2
```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>1</td>
<td>7</td>
<td>10</td>
<td>12</td>
<td>8</td>
<td>14</td>
<td>22</td>
</tr>
</tbody>
</table>