CSc 110, Autumn 2017

Lecture 22: 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.”
What you like about lecture

- Examples: 60
- Everything: 30
- Most things: 10
- Nothing: 5
- Slides: 20
- Explanations: 10
- Ask/answer questions: 5
- Working with others: 2
- Attending: 2
What you dislike about lecture

- nothing: 20
- everything: 5
- hard to read screen: 1
- too slow: 10
- 8am: 25
- too fast: 20
- daily problems: 15
- examples not relatable: 10
- paying attention: 5
- powerpoints too bare: 5
- projects too long/hard: 5
- examples too long+: 5
- quiz: 5
- how to take notes?: 1
What you dislike about section

- nothing: 70
- boring and slow: 5
- everything: 1
- too fast: 1
- time: 1
- not enough time: 20
- more personal help: 1
- SL: 5
- bad time management: 1
Resources you would like

- Youtube videos
- More one on one help
- Study group
- More examples
- Extra practice problems
- List of allowed functions
- Lecture recordings
- Review sessions
- More office hours
- More SSL at tutoring...
Least favorite

• Handwritten exams
• Quizzes
• Time of deadlines
• Daily problems
• No collaboration

Favorite

• Problem solving
• Joy of getting a problem right
• Programming
• Resources
• Projects
• Cartoons

"The projects are probably one of the most satisfying pieces of schoolwork I've ever done. I just feel a real sense of accomplishment when I complete one."
Lists

- **list**: object that stores many values.
  - **element**: One value in a list.
  - **index**: A 0-based integer to access an element from a 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>

• Useful when you know what the list’s elements will be

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>
List initialization

\[
\text{name} = [\text{value}] \times \text{size}
\]

- Example:
  \[
  \text{numbers} = [0] \times 3
  \]

  Creates the following list

<table>
<thead>
<tr>
<th>index</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>26</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
Out-of-bounds

• Legal indexes to use []: between – list's length and the list's length - 1.
  • Reading or writing any index outside this range with [] will cause an
    IndexError: list assignment index out of range

• Example:

  data = [0] * 10
  print(data[0])   # okay
  print(data[9])   # okay
  print(data[-20]) # error
  print(data[10])  # error

index  0  1  2  3  4  5  6  7  8  9
value  0  0  0  0  0  0  0  0  0  0
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: 0 4 11 0 44 0 0 2
  ```

- 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()

- **Use** `len()` **to find the number of elements in a list.**

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

- **What expressions refer to:**
  - The last element of any list?
  - The middle element?
Lists and for loops

- You can also loop directly over lists, just as with strings

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>append(x)</code></td>
<td>Add an item to the end of the list. Equivalent to <code>a[len(a):] = [x]</code>.</td>
</tr>
<tr>
<td><code>extend(L)</code></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><code>insert(i, x)</code></td>
<td>Inserts an item at a given position. <code>i</code> 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><code>remove(x)</code></td>
<td>Removes the first item from the list whose value is <code>x</code>. Errs if there is no such item.</td>
</tr>
<tr>
<td><code>pop(i)</code></td>
<td>Removes the item at the given position in the list, and returns it. <code>a.pop()</code> removes and returns the last item in the list.</td>
</tr>
<tr>
<td><code>clear()</code></td>
<td>Remove all items from the list.</td>
</tr>
<tr>
<td><code>index(x)</code></td>
<td>Returns the index in the list of the first item whose value is <code>x</code>. Errs if there is no such item.</td>
</tr>
<tr>
<td><code>count(x)</code></td>
<td>Returns the number of times <code>x</code> appears in the list.</td>
</tr>
<tr>
<td><code>sort()</code></td>
<td>Sort the items of the list</td>
</tr>
<tr>
<td><code>reverse()</code></td>
<td>Reverses the elements of the list</td>
</tr>
<tr>
<td><code>copy()</code></td>
<td>Return a copy of the list.</td>
</tr>
</tbody>
</table>
"list mystery" problem

• **traversal**: An examination of each element of an list.

• What element values are stored in the following list?

```python
a = [1, 7, 5, 6, 4, 14, 11]
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>