Sequences: Lists of Values

<table>
<thead>
<tr>
<th>String</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ( s = \text{'abc'd'} )</td>
<td>• ( x = [5, 6, 5, 9, 15, 23] )</td>
</tr>
<tr>
<td>Put characters in quotes * Use `\ for quote character</td>
<td>Put values inside [ ] * Separate by commas</td>
</tr>
<tr>
<td>Access characters with [ ] * ( s[0] ) is ( x ) * ( s[0] ) causes an error * ( s[0:2] ) is [ 6, 5 ] * ( s[2:] ) is [ 5, 9, 15, 23 ]</td>
<td>Access values with [ ] * ( x[0] ) is 5 * ( x[0:2] ) is [ 5, 7 ] * ( x[2:] ) is [ 4, -2 ]</td>
</tr>
</tbody>
</table>

List vs. Class Objects

<table>
<thead>
<tr>
<th>List</th>
<th>RGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Attributes are indexed * Example: ( x[2] )</td>
<td>• Attributes are named * Example: ( c \cdot \text{red} )</td>
</tr>
<tr>
<td>• ( x = [5, 7, 4, -2] )</td>
<td>( x = [\text{id1}, \text{id2}, \text{id3}, \text{id4}, \text{id5}] )</td>
</tr>
<tr>
<td>List assignment: (&lt;\text{var}&gt;[&lt;\text{index}&gt;] = &lt;\text{value}&gt; ) * Reassign at index * Affects folder contents * Variable is unchanged</td>
<td>( x[1] = \text{8} )</td>
</tr>
</tbody>
</table>

When Do We Need to Draw a Folder?

- When the value contains other values
  - This is essentially what we mean by ‘object’
- When the value is mutable

<table>
<thead>
<tr>
<th>Type</th>
<th>Container?</th>
<th>Mutable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>float</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>str</td>
<td>Yes*</td>
<td>No</td>
</tr>
<tr>
<td>Point</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RGB</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>list</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Lists Have Methods Similar to String

<table>
<thead>
<tr>
<th>( x = [5, 6, 5, 9, 15, 23] )</th>
<th>[ \text{id1}, \text{id2}, \text{id3}, \text{id4}, \text{id5} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>• \text{index(value)} * Return position of the value * ERROR if value is not there</td>
<td>( x[1] = \text{8} )</td>
</tr>
<tr>
<td>• \text{count(value)} * Returns number of times value appears in list</td>
<td>( x\cdot\text{count}(5) ) evaluates to 2</td>
</tr>
</tbody>
</table>

When Do We Need to Draw a Folder?

- When the value contains other values
  - This is essentially what we mean by ‘object’
- When the value is mutable

Lists are Mutable

<table>
<thead>
<tr>
<th>( x = [5, 7, 4, -2] )</th>
<th>( x = [\text{id1}, \text{id2}, \text{id3}, \text{id4}, \text{id5}] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ( x[1] = \text{8} )</td>
<td>( x[1] = \text{8} )</td>
</tr>
</tbody>
</table>

Strings cannot do this

- \( s = \text{"Hello World!"} \)
  - \( s[0] = \text{"H"} \) \* ERROR
  - String are immutable
List Methods Can Alter the List

- **append(value)**
  - A *procedure method*, not a fruitful method
  - Adds a new value to the end of list
  - `x.append(-1)` changes the list to `[5, 6, 6, 9, -1]`
- **insert(index, value)**
  - Put the value into list at index; shift rest of list right
  - `x.insert(2,-1)` changes the list to `[5, 6, -1, 5, 9,]`
- **sort()**
  - What do you think this does?

List Slices Make Copies

- `x = [5, 6, 5, 9]`
- `y = x[1:3]`  

Exercise Time

- Execute the following:  
  - `>>> x = [5, 6, 5, 9, 10]`
  - `>>> y = x[1:]`
  - `>>> y[0] = 7`
  - What is `x[3]`?
  - Execute the following:  
    - `>>> x = [5, 6, 5, 9, 10]`
    - `>>> y = x[1:]`
    - `>>> y[0] = 7`
    - What is `x[1]`?

Lists and Expressions

- List brackets `[]` can contain expressions
- This is a list *expression*
  - Python must evaluate it
  - Evaluates each expression
  - Puts the value in the list
- Example:  
  - `>>> a = [1+2,3+4,5+6]`
  - `>>> a`  
    - `[3, 7, 11]`

Lists of Objects

- List positions are variables
  - Can store base types
  - But cannot store folders
  - Can store folder identifiers
- Folders linking to folders
  - Top folder for the list
  - Other folders for contents
- Example:  
  - `>>> r = colormodel.RED`
  - `>>> b = colormodel.BLUE`
  - `>>> g = colormodel.GREEN`
  - `>>> x = [r,b,g]`