Mini-Lecture 13

Lists (& Sequences)
# Sequences: Lists of Values

## String
- `s = 'abc d'`
- Put characters in quotes
  - Use `\` for quote character
- Access characters with []
  - `s[0]` is 'a'
  - `s[5]` causes an error
  - `s[0:2]` is 'ab' (excludes c)
  - `s[2:]` is 'c d'

## List
- `x = [5, 6, 5, 9, 15, 23]`
- Put values inside [ ]
  - Separate by commas
- Access values with []
  - `x[0]` is 5
  - `x[6]` causes an error
  - `x[0:2]` is [5, 6] (excludes 2\textsuperscript{nd} 5)
  - `x[3:]` is [9, 15, 23]
Sequences: Lists of Values

String

- $s = 'abc \text{d}'$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Put characters in quotes
  - Use \ for quote character
- Access characters
  - $s[0]$ is 'a'
  - $s[5]$ causes an error
  - $s[0:2]$ is 'ab' (excludes c)
  - $s[2:]$ is 'c d'

List

- $x = [5, 6, 5, 9, 15, 23]$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td>15</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

- Put values inside [ ]
- Access values
  - $x[0]$ is 5
  - $x[6]$ causes an error
  - $x[0:2]$ is [5, 6] (excludes 2\text{nd} 5)
  - $x[3:]$ is [9, 15, 23]

Sequence is a name we give to both...

9/26/18
Representing Lists

Wrong

```
x = [5, 6, 7, -2]
```

Box is “too small” to hold the list

Correct

```
x = id1
```

id1

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>id1</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>-2</td>
</tr>
</tbody>
</table>

```
x = [5, 7, 4, -2]
```
Representing Lists

Wrong

Box is “too small” to hold the list

Correct

Put list in a “folder”

$$x = [5, 7, 4, -2]$$
Modifying List Contents

- **List assignment:**
  <var>[<index>] = <value>
  - Reassign at index
  - Affects folder contents
  - Variable is unchanged

- **Strings cannot do this**
  - s = 'Hello World!'
  - s[0] = 'J'  **ERROR**
  - String are immutable

- **x = [5, 7, 4, -2]**

- **x[1] = 8**
Modifying List Contents

- **List assignment:**
  
  `<var>[<index>] = <value>`
  
  - Reassign at index
  - Affects folder contents
  - Variable is unchanged

- Strings cannot do this
  
  - `s = 'Hello World!'`
  - `s[0] = 'J'`  **ERROR**
  - String are **immutable**

- `x = [5, 7, 4, -2]`

  
  - `x[1] = 8`
Exercise: List Assignment

- Assignment copies id into y
  >>> x = [5, 7, 4, -2]
  >>> y = x

- Execute the assignments:
  >>> x[2] = 8
  >>> y[2] = 3

- What is value of x[2]?

A: 8
B: 3
C: id1
D: I don’t know
Exercise: List Assignment

• Assignment copies id into y
  >>> x = [5, 7, 4, -2]
  >>> y = x

• Execute the assignments:
  >>> x[2] = 8
  >>> y[2] = 3

• What is value of x[2]?
  A: 8
  B: 3  CORRECT
  C: id1
  D: I don’t know

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>-2</td>
</tr>
</tbody>
</table>
Exercise: List Assignment

• Assignment copies id into y
  >>> x = [5, 7, 4, -2]
  >>> y = x

• Execute the assignments:
  >>> x[2] = 8
  >>> y[2] = 3

• What is value of x[2]?
  A: 8
  B: 3  CORRECT
  C: id1
  D: I don’t know
List Slices Make Copies

\[ x = [5, 6, 5, 9] \]

\[ y = x[1:3] \]

copy = new folder
**Advanced List Features: Method Calls**

- Function call with a “list in front”
  - Usage: `list.method(x,y...)`
  - Example: `count()`
    - `x = [1,3,5,3,2]`
    - `x.count(3) == 2`
    - `x.count(0) == 0`


Searching Lists

• $x$.index($y$)
  - Position of the first instance of $y$ in $x$

• $s_1$.count($s_2$)
  - Number of times $s_2$ appears inside of $s_1$

• $s$.strip()
  - Like index_str in introcs

• $x = [1,3,5,3,2]$
  - $s$.index('a') == 0
  - $s$.index('rac') == 2

• $s$.count('a') == 5
  - $s$.count('b') == 2
  - $s$.count('x') == 0

• ' a b '.strip() == 'a b'
Methods Can Alter the List

• x.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, 5, 9, -1]

• x.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]