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 `cd`

**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 2nd 5)
  - `x[3:]` is `[9, 15, 23]`

Lists Have Methods Similar to Strings

- **index(value)**
  - Return position of the value
  - ERROR if value is not there
    - `x.index(9)` evaluates to 3

- **count(value)**
  - Returns number of times value appears in list
    - `x.count(5)` evaluates to 2

Representing Lists

- **Wrong**
  - `x = [5, 6, 7, -2]`
  - Box is "too small" to hold the list

- **Correct**
  - `x.id1`
  - Unique tab identifier
  - Variable holds id
  - Put list in a "folder"

Example: `x = [5, 6, 7, -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]**
  - id1
  - `x[1] = 8`

Exercise: List Assignment

- Assignment copies id into `y`
  - `x = [5, 7, 4, 8]`
  - `y = x`
  - Execute the assignments:
    - `x[2] = 8`
    - `y[8] = 3`
- What is value of `x[2]`?
  - A: 8
  - B: 3
  - C: id1
  - D: I don’t know

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, 5, 9, 15, 23, -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, -1]`

- **sort()**
  - What do you think this does?
Lists and Functions: Swap

```python
def swap(b, h, k):
    """Procedure swaps b[h] and b[k] in b
    Precondition: b is a mutable list, h and k are valid positions in the list"""
    temp = b[h]
    b[h] = b[k]
    b[k] = temp
```

```python
swap(x, 3, 4)
```

Swap $b[h]$ and $b[k]$, because parameter $b$ contains name of list.

List Slices Make Copies

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

List $y$ is a copy of slice of list $x$.

Two Dimensional Lists

<table>
<thead>
<tr>
<th>Table of Data</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
<tr>
<td>0 1 2 3</td>
<td>Each row, col has a value</td>
</tr>
<tr>
<td>0 5 4 7 3</td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
<tr>
<td>1 4 8 9 7</td>
<td>Each row, col has an color value</td>
</tr>
<tr>
<td>2 5 1 2 3</td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
<tr>
<td>3 4 1 2 9</td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
<tr>
<td>4 6 7 8 0</td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
<tr>
<td>d = [[5,4,7,3],[4,8,9,7],[5,1,2,3],[4,1,2,9],[6,7,8,0]]</td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
</tr>
</tbody>
</table>

Store them as lists of lists (row-major order)

How Multidimensional Lists are Stored

- $b = [[9, 6, 4], [6, 7, 7]]$
- $b$ holds name of a one-dimensional list
  - Has len($b$) elements
  - Its elements are (the names of) 1D lists
- $b[i]$ holds the name of a one-dimensional list (of ints)
  - Has len($b[i]$) elements

Slices and Multidimensional Lists

- Only “top-level” list is copied.
- Contents of the list are not altered
- $x = b[3]$

```python
x = b[3]
```

- $b = [[9, 6], [4, 5], [7, 7]]$
- $b$ holds name of a one-dimensional list
  - Has len($b$) elements
  - Its elements are (the names of) 1D lists
- $b[i]$ holds the name of a one-dimensional list (of ints)
  - Has len($b[i]$) elements