Using Color Objects in A3

- New classes in colormap
- RGB, CMYK, and HSV
- Each has its own attributes
  - RGB: red, blue, green
  - CMYK: cyan, magenta, yellow, black
  - HSV: hue, saturation, value
- Attributes have invariants
  - Limits the attribute values
  - Example: red is int in 0..255

```
>>> import colormap
>>> c = colormap.RGB(128,0,0)
>>> r = c.red
>>> c.red = 500  # out of range
AssertionError: 500 outside [0,255]
```

How to Do the Conversion Functions

```python
def rgb_to_cmyk(rgb):
    """Returns: color rgb in space CMYK
    Precondition: rgb is an RGB object"
    # DO NOT CONSTRUCT AN RGB OBJECT
    # Variable rgb already has RGB object
    # 1. Access attributes from rgb folder
    # 2. Plug into formula provided
    # 3. Compute the new cyan, magenta, etc. values
    # 4. Construct a new CMYK object
    # 5. Return the newly constructed object
```

Sequences: Lists of Values

<table>
<thead>
<tr>
<th>String</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = 'abc d'</td>
<td>x = [5, 6, 9, 15, 23]</td>
</tr>
<tr>
<td>Put characters in quotes</td>
<td>Put values inside [ ]</td>
</tr>
<tr>
<td>Use \ for quote character</td>
<td>Separate by commas</td>
</tr>
<tr>
<td>Access characters with [ ]</td>
<td>Access values with [ ]</td>
</tr>
<tr>
<td>s[0] is 'a'</td>
<td>x[0] is 5</td>
</tr>
<tr>
<td>s[2] is 'c'</td>
<td>x[0:2] is [5, 6]</td>
</tr>
<tr>
<td>s[2:] is 'c d'</td>
<td>x[3:] is [9, 15, 23]</td>
</tr>
</tbody>
</table>

Lists Have Methods Similar to String

```
x = [5, 6, 9, 15, 23]
```

- `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

Lists Are Mutable

- Can alter their contents
  - Use an assignment:
    - `<var>[<index>] = <value>`
  - Index is position, not slice
- Does not work for strings
  - `s = 'Hello World!'`
  - `s[0] = 'H'` ERROR
- Represent list as a folder
  - Variable holds tab name
  - Contents are attributes

When Do We Need to Draw a Folder?

- When the value contains other values
  - This is what we are calling 'objects'
- 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>list</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 vs. Class Objects

<table>
<thead>
<tr>
<th>List</th>
<th>RGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Attributes are indexed</td>
<td>• Attributes are named</td>
</tr>
<tr>
<td>- Example: x[2]</td>
<td>- Example: c.red</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id2</th>
<th>id3</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>o</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id3</th>
<th>RGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>255</td>
</tr>
<tr>
<td>green</td>
<td>64</td>
</tr>
<tr>
<td>blue</td>
<td>0</td>
</tr>
</tbody>
</table>

List Methods Can Alter the List

- append(value)
  - A procedure method, not a function method
  - Adds a new value to the end of list
  - Example: x.append(-1) changes the list to [5, 6, 5, 9, -1]

- insert(index, value)
  - Put the value into list at index; shift rest of list right
  - Example: 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

swap(x, 3, 4)
```

List Slices Make Copies

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

Exercise Time

- Execute the following:
  - x = [5, 6, 5, 9, 10]
  - x[3] = -1
  - x.insert(1, 2)
  - What is x[4]?
  - What is x[1]?

- Execute the following:
  - y = x[1::]
  - y[0] = 7
  - What is x[2]?

Lists and Expressions

- List brackets [ ] can contain expressions
  - Example:
    - a = [1+2, 3+4, 5+6]
    - a = [3, 7, 11]

- This is a list expression
  - Python must evaluate it
  - Evaluates each expression
  - Puts the value in the list

- Execute the following:
  - a = 5
  - b = 7
  - x = [a, b, a*b]
  - What is x[2]?