

Using Color Objects in A3

- New classes in colormodel
 - 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
 - Get an error if you violate

```

c = id1
r = 128
            
```

```

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

How to Do the Conversion Functions

```

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
            
```

Only time you will ever call a constructor

Sequences: Lists of Values

String

```

s = 'abc d'
0 1 2 3 4
a b c 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]
0 1 2 3 4 5
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 String

```

x = [5, 6, 5, 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

But you get length of a list with a regular function, not method: `len(x)`

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] = 'J'` **ERROR**
- Represent list as a folder
 - Variable holds tab name
 - Contents are attributes

```

x = [5, 7, 4, -2]
0 1 2 3
5 7 4 -2
8
x[1] = 8
            
```

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**

Type	Container?	Mutable?
int	No	No
float	No	No
str	Yes*	No
Point	Yes	Yes
RGB	Yes	Yes
list	Yes	Yes

Lists vs. Class Objects

List	RGB														
<ul style="list-style-type: none"> Attributes are indexed <ul style="list-style-type: none"> Example: x[2] 	<ul style="list-style-type: none"> Attributes are named <ul style="list-style-type: none"> Example: c.red 														
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">x id2</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">id2</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">list</div> <table style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">x[0]</td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">x[1]</td><td style="padding: 2px;">7</td></tr> <tr><td style="padding: 2px;">x[2]</td><td style="padding: 2px;">4</td></tr> <tr><td style="padding: 2px;">x[3]</td><td style="padding: 2px;">-2</td></tr> </table>	x[0]	5	x[1]	7	x[2]	4	x[3]	-2	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">c id3</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">id3</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">RGB</div> <table style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">red</td><td style="padding: 2px;">128</td></tr> <tr><td style="padding: 2px;">green</td><td style="padding: 2px;">64</td></tr> <tr><td style="padding: 2px;">blue</td><td style="padding: 2px;">255</td></tr> </table>	red	128	green	64	blue	255
x[0]	5														
x[1]	7														
x[2]	4														
x[3]	-2														
red	128														
green	64														
blue	255														

List Methods Can Alter the List

x = [5, 6, 5, 9]

See Python API for more

- append(value)**
 - A **procedure method**, not a function method
 - Adds a new value to the end of list
 - `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
 - `x.insert(2,-1)` changes the list to [5, 6, -1, 5, 9,]
- sort()** What do you think this does?

Lists and Functions: Swap

```
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"""
    1 temp= b[h]
    2 b[h]= b[k]
    3 b[k]= temp
```

swap(x, 3, 4)

swap	b	id4	h	3	k	4
temp	6					

id4	0	5
	1	4
	2	7
	3	X 5
	4	X 6

x id4

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

List Slices Make Copies

x = [5, 6, 5, 9]

x id5

id5

list

x[0]	5
x[1]	6
x[2]	5
x[3]	9

y = x[1:3]

y id6

id6

list

y[0]	6
y[1]	5

copy = new folder

Exercise Time

<ul style="list-style-type: none"> Execute the following: <pre>>>> x = [5, 6, 5, 9, 10] >>> x[3] = -1 >>> x.insert(1,2)</pre> What is x[4]? 	<ul style="list-style-type: none"> Execute the following: <pre>>>> x = [5, 6, 5, 9, 10] >>> y = x[1:] >>> y[0] = 7</pre> What is x[1]?
---	--

9/30/14
Lists & Sequences
11

Lists and Expressions

<ul style="list-style-type: none"> List brackets [] can contain expressions This is a list expression <ul style="list-style-type: none"> Python must evaluate it Evaluates each expression Puts the value in the list Example: <pre>>>> a = [1+2,3+4,5+6] >>> a [3, 7, 11]</pre> 	<ul style="list-style-type: none"> Execute the following: <pre>>>> a = 5 >>> b = 7 >>> x = [a, b, a+b]</pre> What is x[2]?
---	--