

Sequences: Lists of Values

String	List
<ul style="list-style-type: none"> <code>s = 'abc d'</code> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px;"> 01234 </div> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px;"> abc d </div> <ul style="list-style-type: none"> Put characters in quotes <ul style="list-style-type: none"> Use <code>\</code> for quote character Access characters with <code>[]</code> <ul style="list-style-type: none"> <code>s[0]</code> is 'a' <code>s[5]</code> causes an error <code>s[0:2]</code> is 'ab' (excludes c) <code>s[2:]</code> is 'c d' 	<ul style="list-style-type: none"> <code>x = [5, 6, 5, 9, 15, 23]</code> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px;"> 012345 </div> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px;"> 56591523 </div> <ul style="list-style-type: none"> Put values inside <code>[]</code> <ul style="list-style-type: none"> Separate by commas Access values with <code>[]</code> <ul style="list-style-type: none"> <code>x[0]</code> is 5 <code>x[6]</code> causes an error <code>x[0:2]</code> is [5, 6] (excludes 2nd 5) <code>x[3:]</code> is [9, 15, 23]

1

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)`

2

Representing Lists

Wrong	Correct								
<code>x [5, 6, 7, -2]</code> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> Box is "too small" to hold the list </div>	<code>x id1</code> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Variable holds id</div> <div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; align-items: center;"> id1 <div style="border: 1px solid black; padding: 2px;">Unique tab identifier</div> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>5</td></tr> <tr><td>1</td><td>7</td></tr> <tr><td>2</td><td>4</td></tr> <tr><td>3</td><td>-2</td></tr> </table> </div> </div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> Put list in a "folder" </div>	0	5	1	7	2	4	3	-2
0	5								
1	7								
2	4								
3	-2								
<code>x = [5, 7, 4, -2]</code>									

3

Lists vs. Class Objects

List	RGB																		
<ul style="list-style-type: none"> Attributes are indexed <ul style="list-style-type: none"> Example: <code>x[2]</code> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">x id2</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td colspan="2">list</td></tr> <tr><td>0</td><td>5</td></tr> <tr><td>1</td><td>7</td></tr> <tr><td>2</td><td>4</td></tr> <tr><td>3</td><td>-2</td></tr> </table> </div> </div>	list		0	5	1	7	2	4	3	-2	<ul style="list-style-type: none"> Attributes are named <ul style="list-style-type: none"> Example: <code>c.red</code> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">c id3</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td colspan="2">RGB</td></tr> <tr><td>red</td><td>128</td></tr> <tr><td>green</td><td>64</td></tr> <tr><td>blue</td><td>255</td></tr> </table> </div> </div>	RGB		red	128	green	64	blue	255
list																			
0	5																		
1	7																		
2	4																		
3	-2																		
RGB																			
red	128																		
green	64																		
blue	255																		

4

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**

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

5

Lists are Mutable

- 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 id1

id1	
0	5
1	X 8
2	4
3	-2

6

List Methods Can Alter the List

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

- **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]`
- **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()**

7

Lists and Functions: Swap

1. `def swap(b, h, k):`
2. `""" Swaps b[h] and b[k] in b`
3. `Precond: b is a mutable list,`
4. `h, k are valid positions"""`
5. `temp= b[h]`
6. `b[h]= b[k]`
7. `b[k]= temp`

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

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

8

List Slices Make Copies

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

9

Exercise Time

- Execute the following:


```
>>> x = [5, 6, 5, 9, 10]
>>> x[3] = -1
>>> x.insert(1,2)
```
- What is x[4]?
 - A: 10
 - B: 9
 - C: -1
 - D: **ERROR**
 - E: I don't know
- Execute the following:


```
>>> x = [5, 6, 5, 9, 10]
>>> y = x[1:]
>>> y[0] = 7
```
- What is x[1]?
 - A: 7
 - B: 5
 - C: 6
 - D: **ERROR**
 - E: I don't know

10

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]
```
- Execute the following:


```
>>> a = 5
>>> b = 7
>>> x = [a, b, a+b]
```
- What is x[2]?
 - A: 'a+b'
 - B: 12
 - C: 57
 - D: **ERROR**
 - E: I don't know

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 = intros.RGB(255,0,0)
>>> g = intros.RGB(0,255,0)
>>> b = intros.RGB(0,0,255)
>>> x = [r,g,b]
```

12