

Sequences: Lists of Values

String	List																						
<ul style="list-style-type: none"> <code>s = 'abc d'</code> <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>a</td><td>b</td><td>c</td><td></td><td>d</td></tr> </table> <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 e) <code>s[2:]</code> is 'c d' 	0	1	2	3	4	a	b	c		d	<ul style="list-style-type: none"> <code>x = [5, 6, 5, 9, 15, 23]</code> <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>5</td><td>6</td><td>5</td><td>9</td><td>15</td><td>23</td></tr> </table> <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] 	0	1	2	3	4	5	5	6	5	9	15	23
0	1	2	3	4																			
a	b	c		d																			
0	1	2	3	4	5																		
5	6	5	9	15	23																		

Lists Have Methods Similar to Strings

`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)`

Representing Lists

Wrong	Correct								
<code>x = [5, 6, 7, -2]</code> <div style="border: 1px solid gray; padding: 5px; width: fit-content; margin: 10px auto;">Box is "too small" to hold the list</div>	<code>x = [5, 7, 4, -2]</code> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;"> <code>x</code> → <code>id1</code> Variable holds id </div> <div style="border: 1px solid gray; padding: 5px;"> <table border="1" style="border-collapse: collapse;"> <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 style="margin-left: 10px;"> Unique tab identifier </div> </div> <div style="margin-top: 10px; border: 1px solid gray; padding: 5px; width: fit-content;">Put list in a "folder"</div>	0	5	1	7	2	4	3	-2
0	5								
1	7								
2	4								
3	-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]`

0	5	7	4	-2
	X			
		8		

`x[1] = 8`

0	5
1	X 8
2	4
3	-2

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

id1	
0	5
1	7
2	4
3	-2

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()` What do you think this does?

See Python API for more

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
```

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

swap(x, 3, 4)

swap		3	
b	id4	h	3
temp	6	k	4

id4	
0	5
1	4
2	7
3	5
4	5

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

Nested Lists

- Lists can hold any objects
- Lists are objects
- Therefore lists can hold other lists!

```
a = [2, 1]
b = [3, 1]
c = [1, 4, b]
x = [1, a, c, 5]
```

x[1]	
x[2]	
x[2][2]	

x = [1, [2, 1], [1, 4, [3, 1]], 5]

x[0]	x[1][1]	x[2][0]	x[2][2][1]
------	---------	---------	------------

Two Dimensional Lists

Table of Data

	0	1	2	3
0	5	4	7	3
1	4	8	9	7
2	5	1	2	3
3	4	1	2	9
4	6	7	8	0

Each row, col has a value

Images

	0	1	2	3	4	5	6	7	8	9	10	11	12
0													
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

Each row, col has an color value

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

d = [[5,4,7,3],[4,8,9,7],[5,1,2,3],[4,1,2,9],[6,7,8,0]]

How Multidimensional Lists are Stored

```
b = [[9, 6, 4], [5, 7, 7]]
```

9	6	4
5	7	7

b id1

id1	
id2	
id3	

id2	
9	
6	
4	

id3	
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

Slices and Multidimensional Lists

- Only “top-level” list is copied.
- Contents of the list are not altered

```
b = [[9, 6], [4, 5], [7, 7]]
x = b[:2]
```

id1	
id2	
id3	

id2	
9	
6	

id3	
4	
5	

id4	
7	
7	

id5	
id2	
id3	