

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; font-family: monospace; font-size: small;"> 0 1 2 3 4 </div> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px; font-family: monospace; font-size: x-small;"> a b c 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; font-family: monospace; font-size: small;"> 0 1 2 3 4 5 </div> <div style="display: flex; justify-content: space-around; border: 1px solid black; padding: 2px; font-family: monospace; font-size: x-small;"> 5 6 5 9 15 23 </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]

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)`

Representing Lists

Wrong	Correct
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <code>x [5, 6, 7, -2]</code> </div> <p style="font-size: small; margin-top: 10px;">Box is "too small" to hold the list</p>	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <code>x id1</code> Variable holds id </div> <div style="border: 1px solid black; padding: 5px; font-family: monospace; font-size: x-small;"> id1 Unique tab identifier 0 5 1 7 2 4 3 -2 </div> </div> <p style="font-size: small; margin-top: 10px;">Put list in a "folder"</p>
<code>x = [5, 7, 4, -2]</code>	

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="margin-top: 10px;"> <code>x id2</code> list 0 5 1 7 2 4 3 -2 </div>	<ul style="list-style-type: none"> Attributes are named <ul style="list-style-type: none"> Example: <code>o.red</code> <div style="margin-top: 10px;"> <code>o id3</code> RGB red 128 green 64 blue 255 </div>

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

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 = [5, 7, 4, -2]`

`x id1`

0 5
 1 7
 2 4
 3 -2

- `x[1] = 8`

`x id1`

0 5
 1 8
 2 4
 3 -2

List Methods Can Alter the List

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

See Python API for more

- `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?

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)`

List Slices Make Copies

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

x `id5` y `id6`

copy = new folder

Exercise Time

- Execute the following:


```
>>> x = [5, 6, 5, 9, 10]
>>> x[3] = -1
>>> x.insert(1,2)
```
- Execute the following:


```
>>> x = [5, 6, 5, 9, 10]
>>> y = x[1:]
>>> y[0] = 7
```
- What is x[4]?
- What is x[1]?

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

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.RED
>>> b = intros.BLUE
>>> g = intros.GREEN
>>> x = [r,b,g]
```