LAB 5

Topic: Lists, Tuples, Strings

Lists - Python Data Structure

Can contain multiple data types

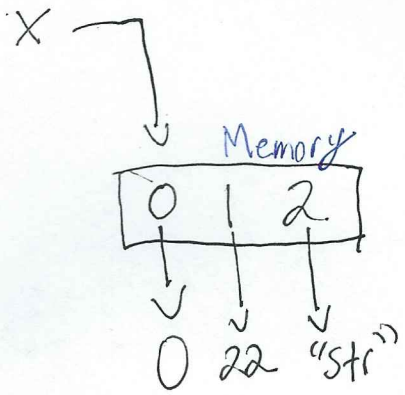
*Passed by reference in function calls

* Index starts at 0

```
-5 -4 -3 -2 -1   Index
[0, 1, 3, (1,2), "str"]
 0  1  2  3   4   Index
```

Stored in memory as pointers to objects

```
X = [0, 22, "str"]
```



In place List modifications req. an index loop.

This means using:

```
for i in range(len(list)):
    list[i] = ...
```

OR

```
for index, val in enumerate(list)):
    ...
```

Slicing - getting a sub-sequence of an object

Assume a = List

a[start:stop] —> items start through stop - 1

a[start:] —> items start through rest of list

a[:stop] —> items from beginning through stop - 1

a[:] —> copy whole array / List

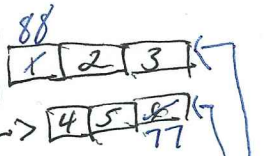
a[::-1] —> reverse the List

Examples

```
x = [1,2,3,4,5]
for val in x:
    print(val)
```

```
x = [[1,2]]
for val in x:
    for val_ in val:
        print(val_)
```

```
xs = [1,2,3]
ys = [4,5,6]
```




```
lists = [xs, ys]
```

```
lists[0][0] = 88
ys[2] = 77
```

```
xs = [6,7,8,9,10]
for x in xs:
    xs.pop()
```

How many iterations for this loop? 3

```
for i in range(len(xs)):
    xs.pop()
```

itrs? 5

```
xs = [[1,2,3],[4,5,6]]
z = []
for x in xs:
    for y in x:
        z.append(y)
```

Tuples – sequence of immutable Python objects

(1,2,3), ("hello", "there")

Access tuples elements with indices

```
x = (1,2,3)
print(x[0])
```

Tuple can contain mutable objects

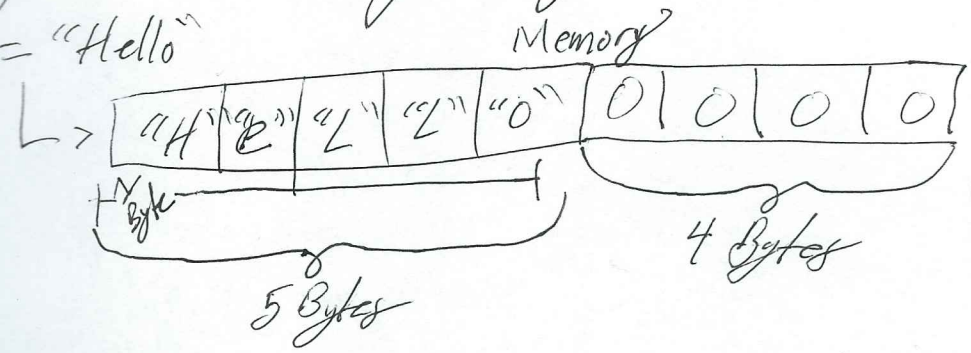
Strings – sequence of immutable characters

Why immutability?

Ensures things dont change through execution

Strings exist in memory contiguously so it has a set size

x = "Hello"



If we can modify x and make the string longer, it "spills" into other data used by the program – we dont want this

Additionally consider the following:

A hash table data structure which maps (key value) pairs

The hash() function "hashes" a key value pair into an index

Assume hash("dog") outputs an index of 0

Mapped K,V pair of "dog","cat" to index 0 in hash table

Now assume we mutate the key "dog" to "Bob"

The lookup() function for "Bob" is messed up as "Bob" maps to a different index,