

## Python In-class Programming Assignment

### Introduction To Lists

- Readings: *Liang* section 10.1 – 10.2

Lists in Python are incredibly powerful. A list is a sequential collection of elements stored using a single named reference (*variable name*) and applying indices to access individual items.

Creating lists:

Lists in Python are created using square brackets `[]`. Here's a simple list containing the values 1, 2, 3:

```
L = [1,2,3]

print(L[0]) prints 1
print(L[1]) prints 2
print(L[2]) prints 3
print(L) prints [1,2,3]
```

Lists are *mutable* (meaning changeable). You can change the value of an individual item in a list using its index.

```
L = [1,2,3]
print(L) prints [1,2,3]
L[1] = 8
print(L) prints [1,8,3]
```

By way of comparison, strings are *immutable* (not changeable). The following code causes an error:

```
S = "Go Navy"
S[0] = 'B'
TypeError: 'str' object does not support item assignment
```

Some useful list operations:

```
L = [1,2,3,4,5]
```

- append

```
L.append(10)
print(L)
```

- insert

```
L.insert(3,22)
print(L)
```

## Python In-class Programming Assignment

### Introduction To Lists

- `remove`

```
L.remove(3)
print(L)
```

*Question: Does it remove the item at position 3 or does it remove an item whose value is 3? What if you enter: `L.remove(500)`?*

- `del`

```
del(L[3])
print(L)
```

*Question: Does it remove the item at position 3 or does it remove an item whose value is 3? What if you enter `del(L[25])`?*

- `pop`

```
x = L.pop()
print(x)
print(L)
```

- `len`

```
total = len(L)
print(total)
```

- `copy`

```
L2 = L
print(L)
print(L2)
L2[0] = 8
print(L)
print(L2)
```

*Is that the behavior we expected?*

```
L2 = L.copy()
print(L)
print(L2)
L2[0] = 100
print(L)
print(L2)
```

*How about now?*

## Python In-class Programming Assignment

### Introduction To Lists

- `sort`

```
print(L)
L.sort()
print(L)
```

- `list(str)`

```
s = "Go Navy!"
L = list(s)
print(L)
```

- `split(str)`

```
s = "I am a Cyber Operations Major"
L = s.split(' ')
print(L)
L.sort()
print(L)
What did this give you?
```

- `readlines()`

```
f = open("gettysburg.txt", "r")
L = f.readlines()
print(L)
```

*Write a program to count the number of words in the Gettysburg Address, where a word is defined as something bounded by spaces.*