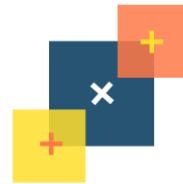


# WEEK 4

# LISTS / ARRAYS



TECH BHOI

JUL 20  
JUL 24  
2020



# LIST

MON  
JUL 20

## LIST = [MULTI VALUE]

A LIST is used to store MULTIPLE pieces of information that we want to reuse throughout our programs. A real world example of this is like your grocery list we can change it or have it store nothing or even have it store just one item. Similarly the number of items in your shopping cart stores the list of how many items you've picked up in the store, this can also change or vary over time.

### DISCUSSION:

Identify some things around you that can be represented as a list.

Three parts:

- 1) A ListName
- 2) An equal sign (=)
- 3) **open/close square brackets**

**OPTIONAL ) comma seperated values**

THINK OF A MAGIC BAG WITH UNLIMITED SPOTS FOR SINGLE ITEMS

### QUIZ:

What are the two most important parts in creating a list?

- a) the equals sign and parentheses
- b) square parentheses and quotes
- c) square parentheses and commas
- d) open square parentheses and a close square parentheses

**LIST\_NAME = [VAL1,...,VALN]**

**USE A LIST TO STORE  
MORE THAN ONE  
VALUE AT A TIME**



**TUE  
JUL 21**

# LIST INDEX

A list is similar to a cubby with space for an item. Each space has its own identifying number, and we call this the index. It is similar to counting, but instead of counting from 1, we start at 0 and increment.

Counting : 1,2,3,4,5,6,...

Indices : 0,1,2,3,4,5...

In the following list the index of one is 0 but the index of two is 1

index    0    1    2    3

mylist = [one,two,three,four]

```
one=1  
two = 2  
three = 3  
four = 4  
mylist = [one,two,three,four]
```

### Accessing an element:

#### **Accessing the element at an index**

```
print(mylist[0])
```

#### **Accessing elements using negative index**

```
print(mylist[-1])
```

#### **Accessing a range of elements in the list**

```
print(mylist[1:3])
```

### Quiz:

Complete the code below to print the value of the variable with the name three (we have two options)

```
mylist =[one,two,three,four]  
print(mylist[____])
```



# List Operations

WEEK 4 | WED JUL 22

## ADD / REMOVE FROM LIST

```
groceries = ["eggs", "bread"]
groceries.append("milk")
```

**listName.append(value)**

Whenever we append, we do one element at a time, and it goes to the end of the list  
our list now looks like [eggs,bread,milk]

While we go across the aisles in the grocery store, we want to cross off items (remove) products from our list that we have in our cart already. In python, this is represented by using the remove keyword (as seen below)

```
groceries = ["eggs", "bread", "milk", "fruit"]
```

**groceries.remove("bread")**

now our list is ["eggs", "milk", "fruit"]

## LIST FUNCTIONS

### Size of a list:

**len(listname)**

This will return to us the number of elements in the list

### Smallest value in the list:

**min(listname)**

This will return to us the smallest value in the list

### Largest value in the list:

**max(listname)**

This will return to us the largest value in the list

### Existence:

**value in list**

This will check whether a value exists in the list, will return a boolean

### Merging lists (Concatenation):

**list3 = list1 + list2**

This will add the two lists together and store the "new" list in list3 without changing list1 & list2

What is the appropriate keyword to add an item to the list at the back?

- a) insert
- b) add
- c) toend
- d) append

How would you add the location USA to the following list?

```
places = ["Canada",
"England", "United Kingdom"]

places._____("_____")
```