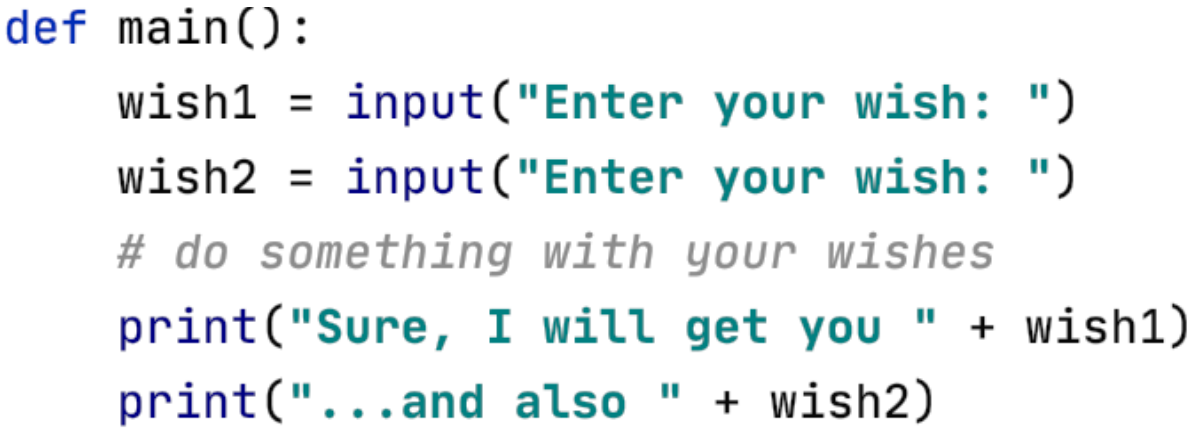
CS Bridge, Lecture 11

Lists

Wish List

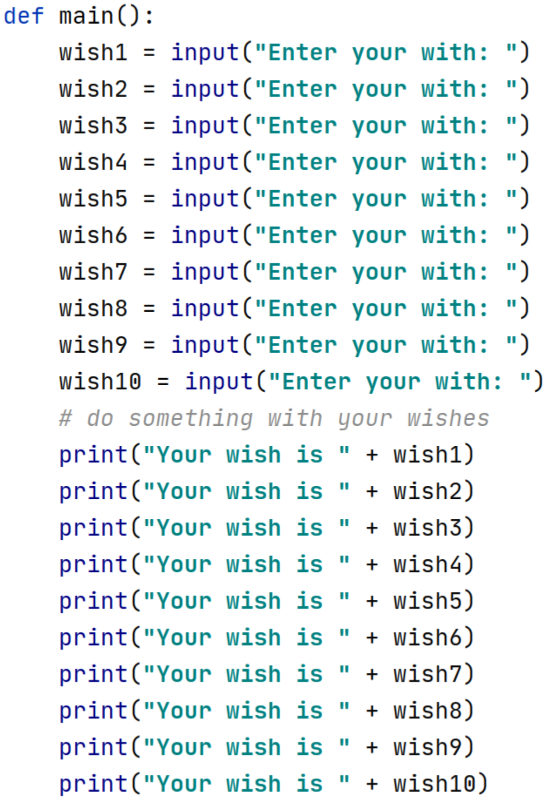
Consider this very kind program that asks for wishes



```
def main():
    wish1 = input("Enter your wish:  ")
    wish2 = input("Enter your wish:  ")
    # do something with your wishes
    print("Sure, I will get you " + wish1)
    print("...and also " + wish2)
```

Wish List

What if the user has many wishes?



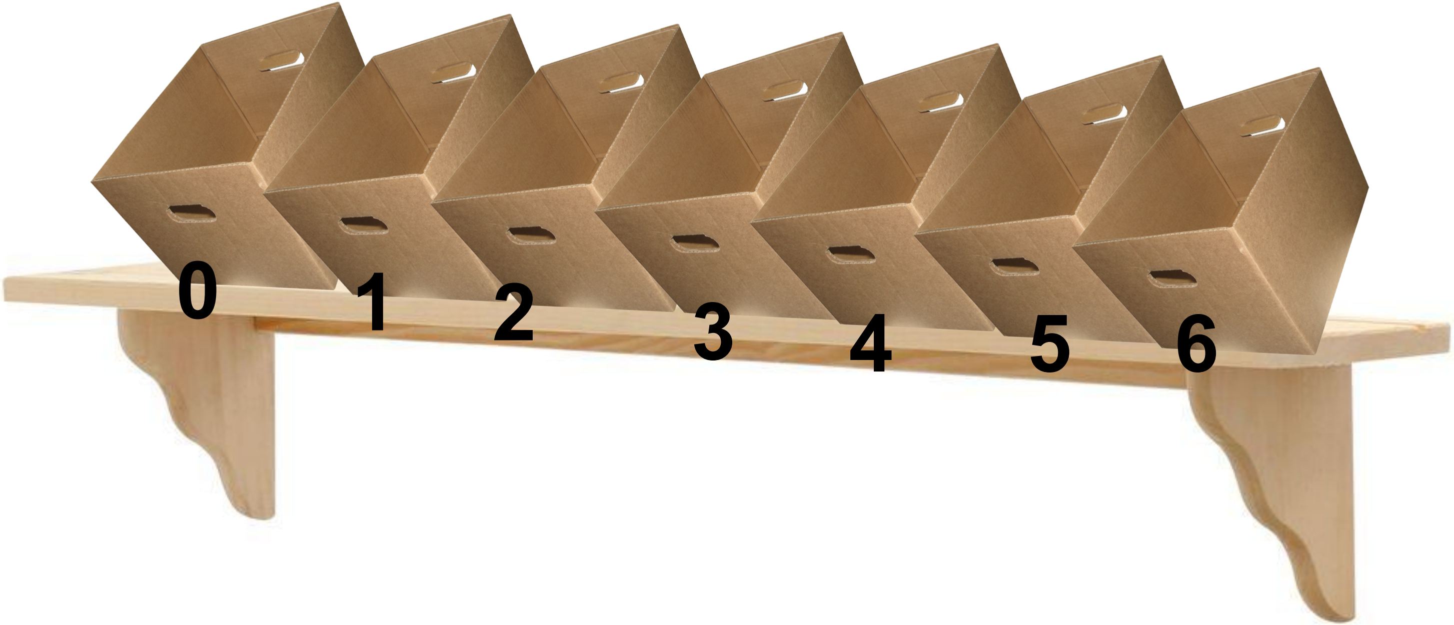
```
def main():
    wish1 = input("Enter your with: ")
    wish2 = input("Enter your with: ")
    wish3 = input("Enter your with: ")
    wish4 = input("Enter your with: ")
    wish5 = input("Enter your with: ")
    wish6 = input("Enter your with: ")
    wish7 = input("Enter your with: ")
    wish8 = input("Enter your with: ")
    wish9 = input("Enter your with: ")
    wish10 = input("Enter your with: ")
    # do something with your wishes
    print("Your wish is " + wish1)
    print("Your wish is " + wish2)
    print("Your wish is " + wish3)
    print("Your wish is " + wish4)
    print("Your wish is " + wish5)
    print("Your wish is " + wish6)
    print("Your wish is " + wish7)
    print("Your wish is " + wish8)
    print("Your wish is " + wish9)
    print("Your wish is " + wish10)
```

wishlist10.py

Then we would keep a wish list

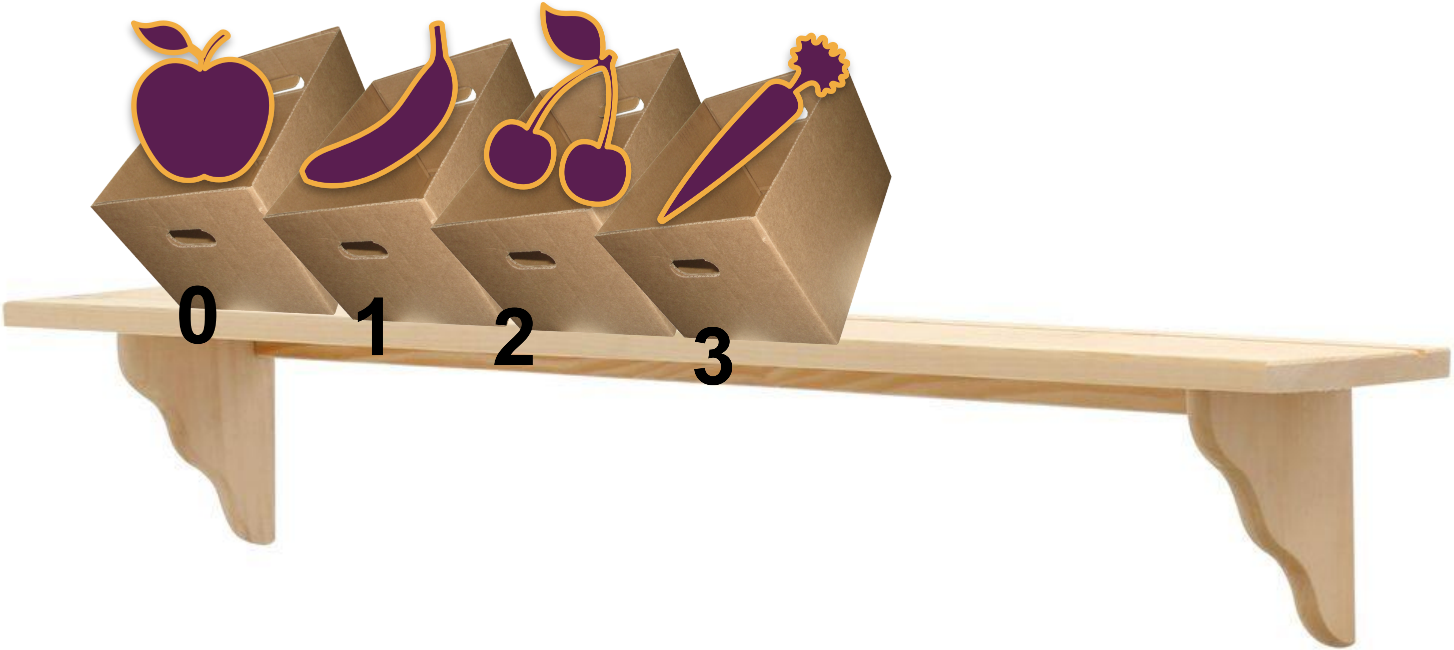
Lists

- A **list** is way to keep track of an *ordered collection* of items



Lists

- The list dynamically adjusts its size as elements are added or removed



Lists

- You can create them using [], empty or containing elements separated by comma

```
fruits_list = []
```

Lists

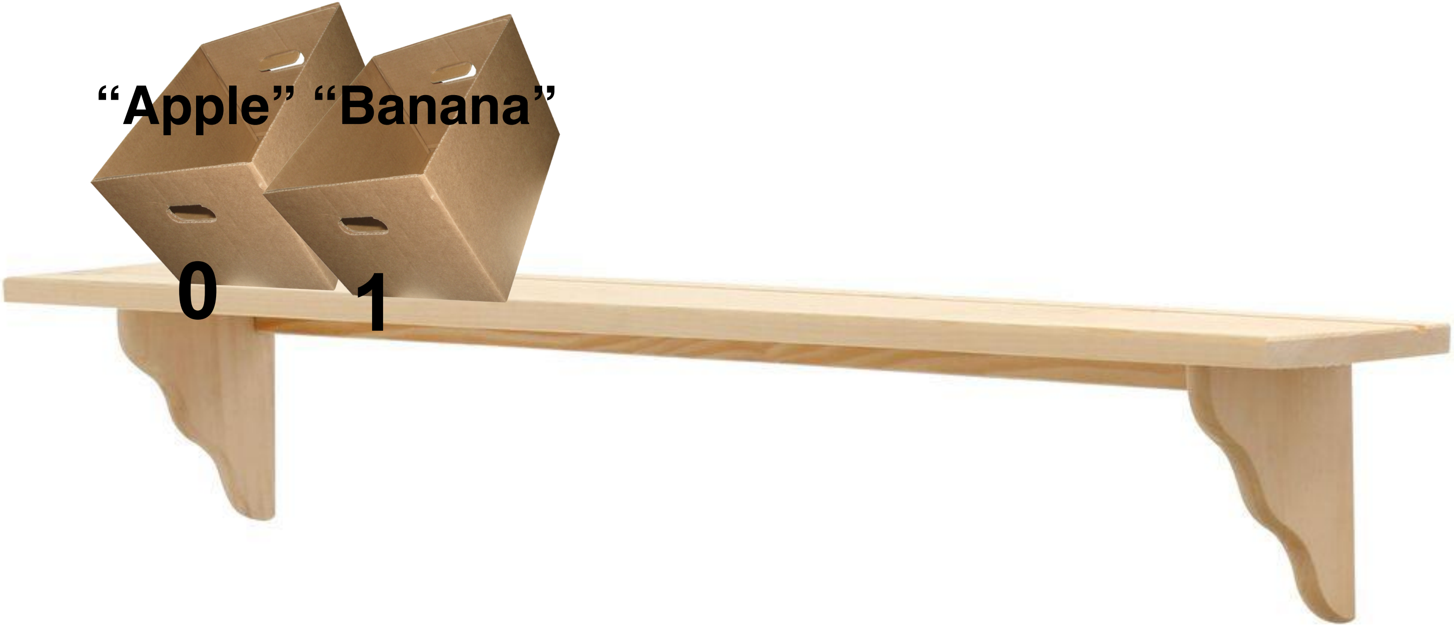
- You can create them using [], empty or containing elements separated by comma

```
fruits_list = []
fruits_list.append(“Apple”)
```

Lists

- You can create them using [], empty or containing elements separated by comma

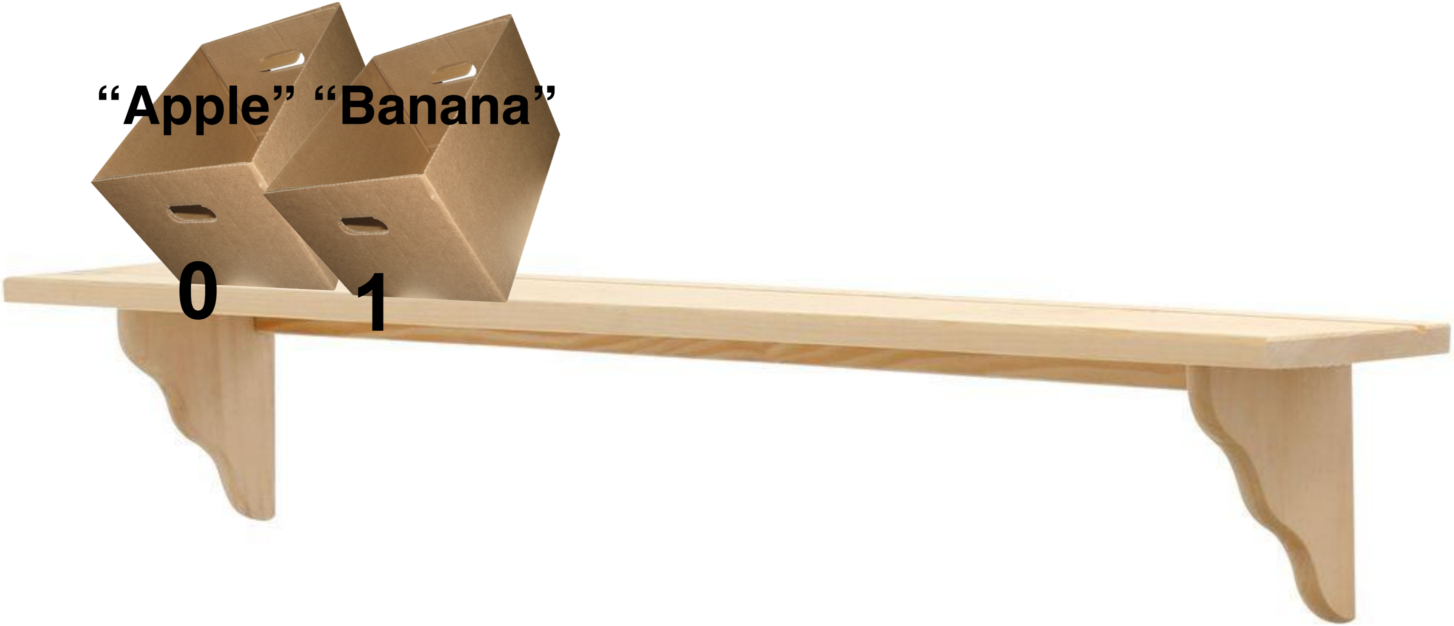
```
...
fruits_list.append(“Banana”)
```



Lists

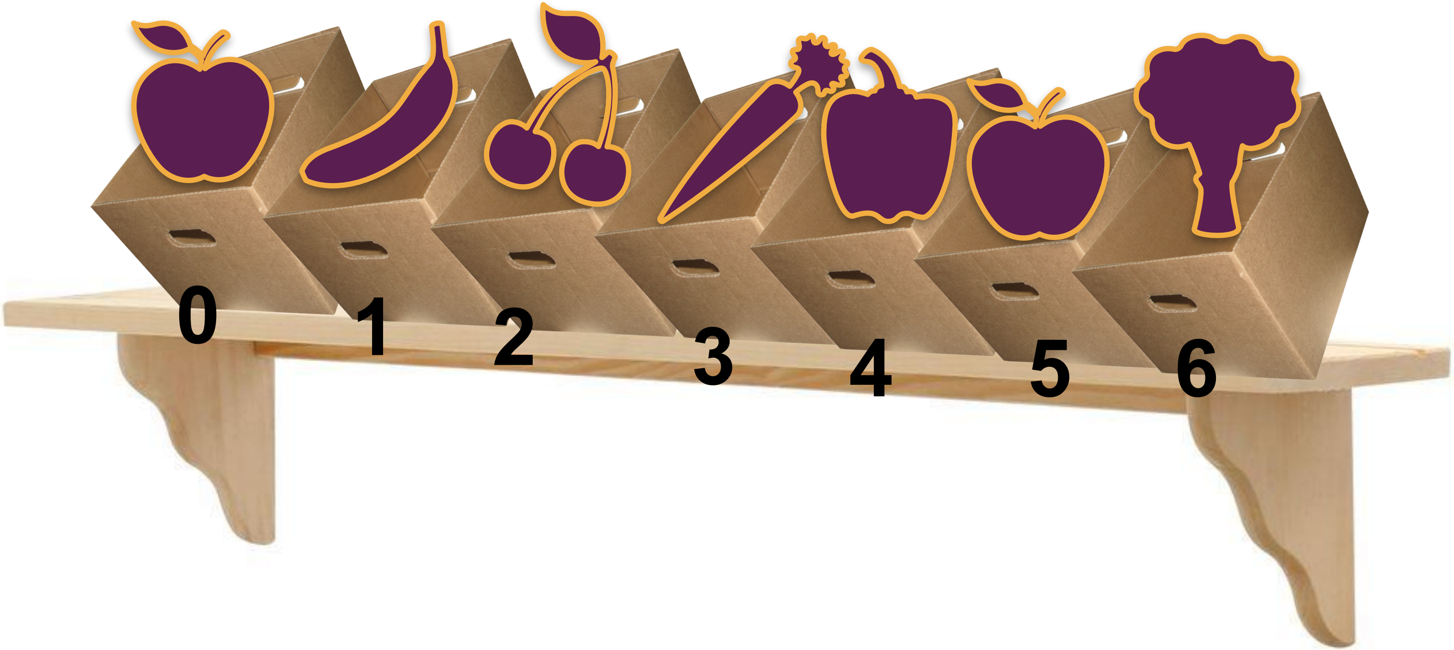
- You can create them using [], empty or containing elements separated by comma

fruits_list = [“Apple”, “Banana”]



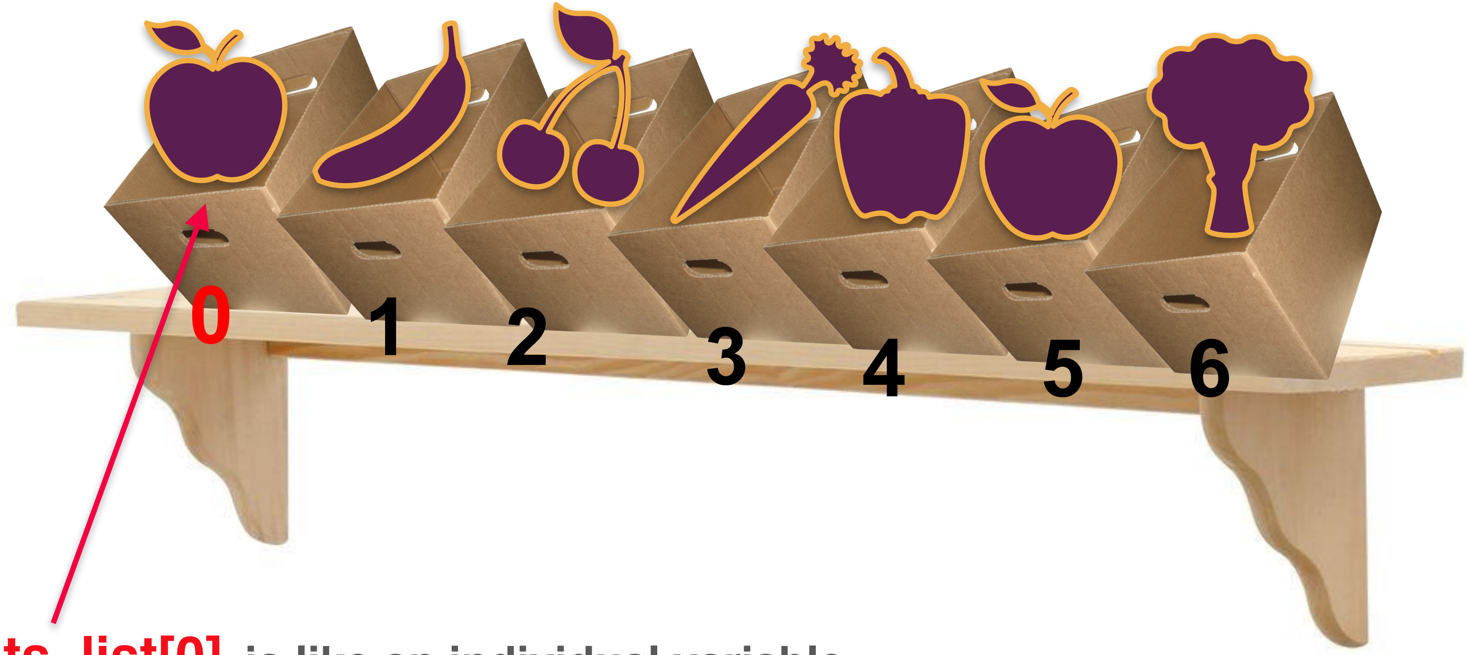
Lists

- Items in the list are called "elements"



Lists

- Ordered: can refer to elements by their position



fruits_list[0] is like an individual variable

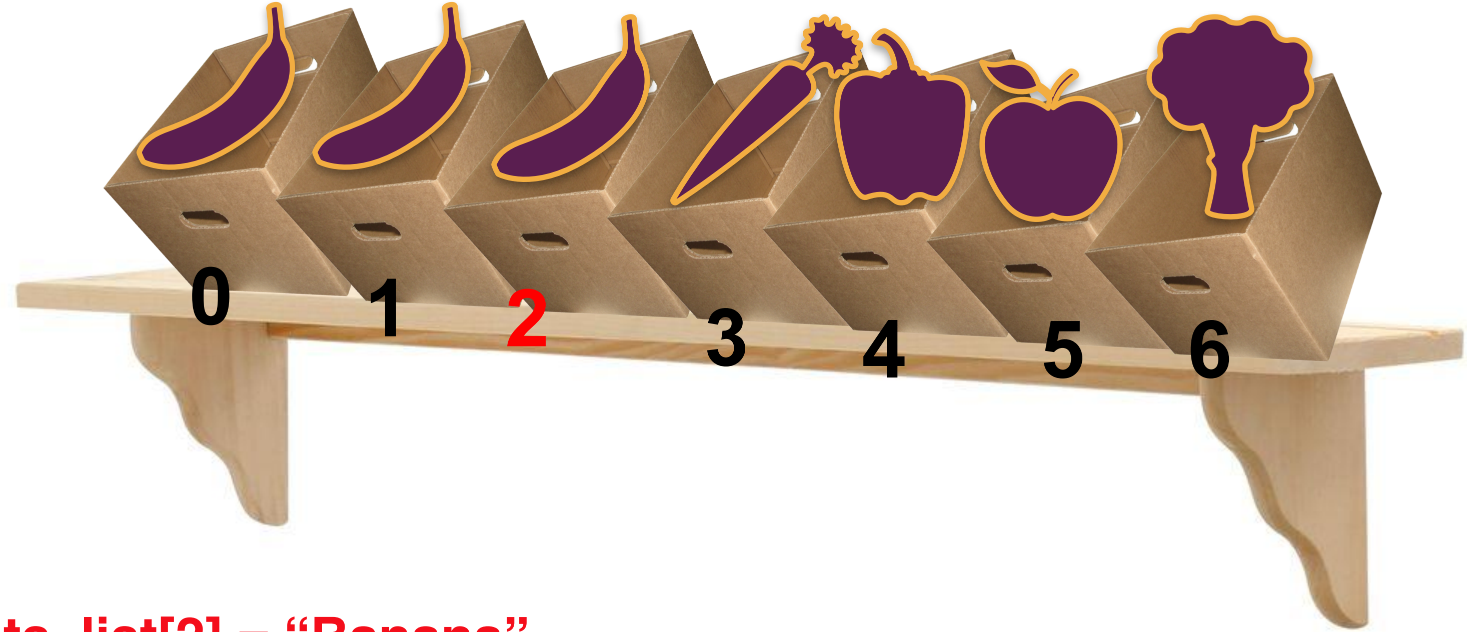
Lists

- You can access and use the value and/or modify the value

fruits_list[0] = “Banana”

Lists

- You can access and use the value and/or modify the value



fruits_list[2] = “Banana”

Intro to Lists

```
# A list of strings
some_str_list = ["Elma", "Armut"] # initialising the list with two elements
some_str_list.append("Kel Mahmut") # adding new element to the list
print(some_str_list)
```

Outputs:

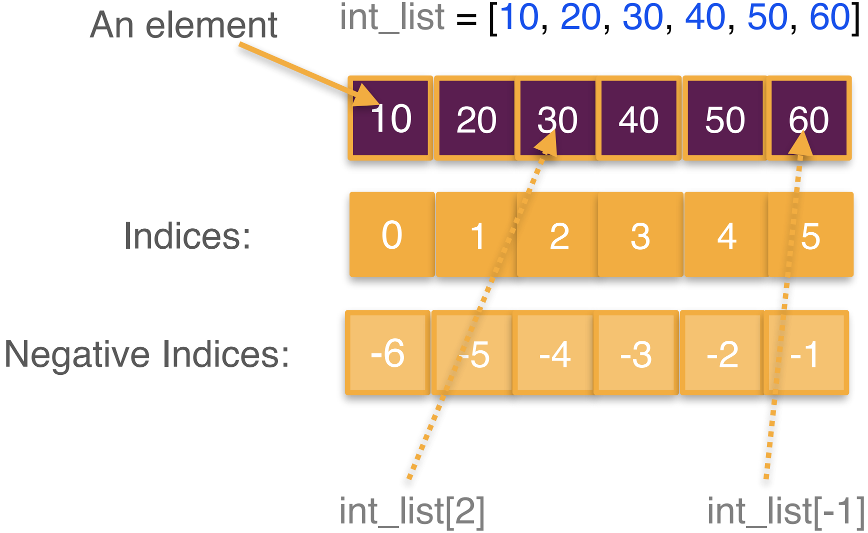
```
['Elma', 'Armut', 'Kel Mahmut']
```

```
# A list containing different types of data
mixed_list = [2, True, "Hi", 6.8]
print(mixed_list)
```

Outputs:

```
[2, True, 'Hi', 6.8]
```

```
# A list of integers
int_list = [10, 20, 30, 40, 50, 60]
```



lists_intro.py

Cycling through elements of a list

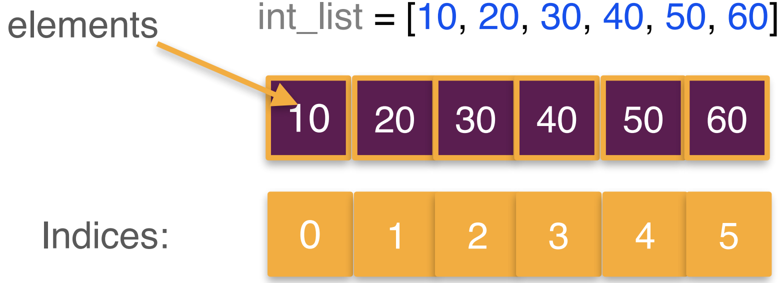
It is often practical to create indices using a for loop, use the loop variable as the index and access elements in order

```
for i in range(len(int_list)):
    print(int_list[i])
```

Outputs:

```
10
20
30
40
50
60
```

```
# A list of integers
int_list = [10, 20, 30, 40, 50, 60]
```



lists_intro.py

Cycling through elements of a list

```
# Creating an empty list and appending random integer values
random_ints_list = []
for i in range(10):
    random_ints_list.append(random.randint(0, 9))
print(random_ints_list)
```

Outputs: (a different set of values in each run)

```
[6, 2, 8, 7, 9, 8, 8, 7, 9, 0]
```

```
# Accessing values of a list and modifying them
for i in range(len(random_ints_list)):
    random_ints_list[i] *= 10
print(random_ints_list)
```

Outputs:

```
[60, 20, 80, 70, 90, 80, 80, 70, 90, 0]
```

lists_intro.py

Wish List

Write a program that asks for wishes from a user. The program should continue receiving new wishes until the user clicks enter without typing a character. Then the program should inform the user that some of her wishes will be fulfilled.

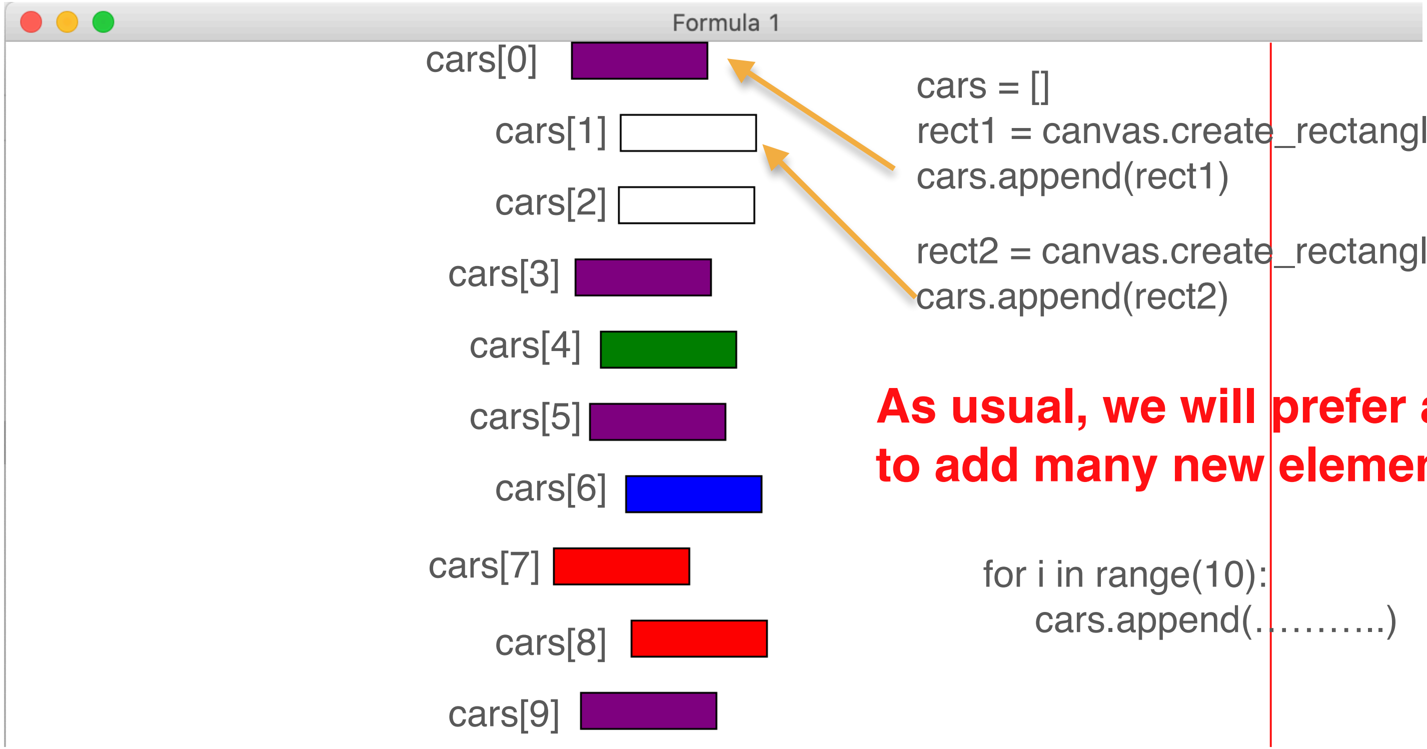
Example run:

```
Enter your wish: icecream          I'll get you icecream, no worries
Enter your wish: wings             Sorry, I cannot get you wings
Enter your wish: hotdog            I'll get you hotdog, no worries
Enter your wish: water             I'll get you water, no worries
Enter your wish: coffee            I'll get you coffee, no worries
Enter your wish: rocket            I'll get you rocket, no worries
Enter your wish:
```

wishlist.py

List of graphical objects

Sometimes you need to create many graphical objects and move them



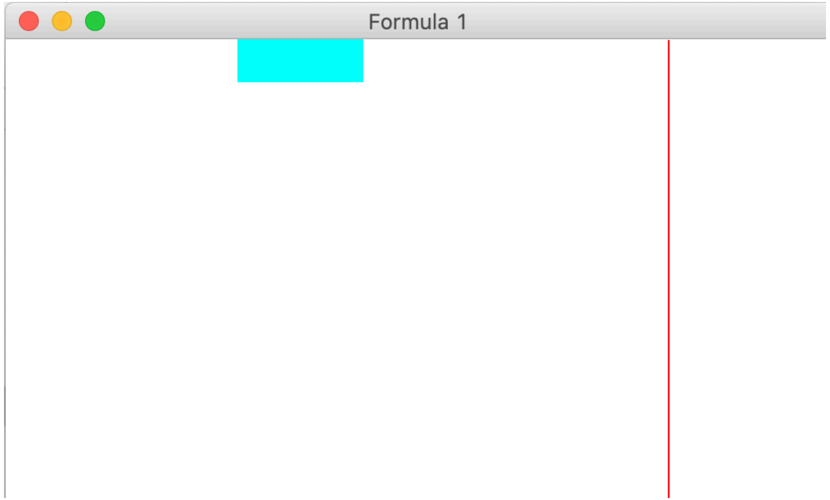
```
cars = []
rect1 = canvas.create_rectangle(x1,y1,x2,y2)
cars.append(rect1)

rect2 = canvas.create_rectangle(x3,y3,x4,y4)
cars.append(rect2)
```

As usual, we will prefer a for loop to add many new elements

```
for i in range(10):
    cars.append(……….)
```

Time to implement ‘Car Race’ using the starter code



car_race_starter.py