Bloomberg

STARTUP

Bloomberg

STARTUP

Python Lesson

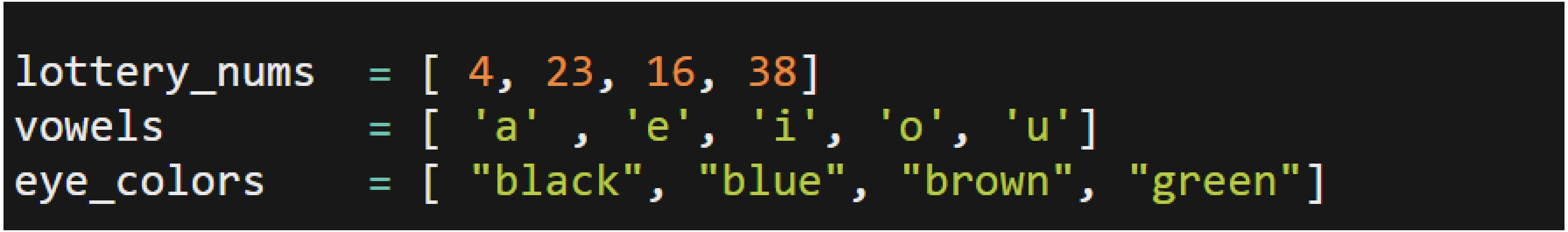
Lists

Lists: Introduction

- Recall, a **variable** is a name given to a piece of data that can be used to refer to it later.
- We have learned about some different variable types available
  - **int** or integer, e.g. favorite_number = 23
  - **float**, e.g. height_in_inches = 71.5
  - **string**, e.g. message = “Hello World!”

Lists: Introduction

- A **list** is a collection of data
- Examples

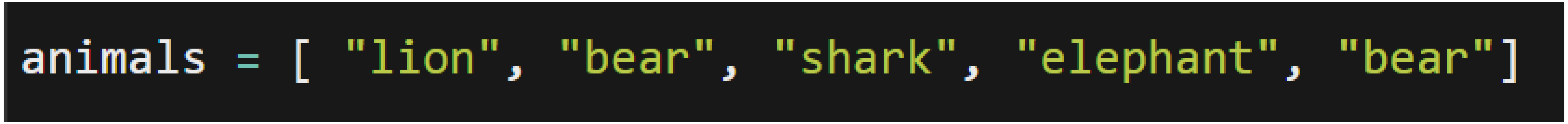


```
lottery_nums  = [ 4, 23, 16, 38]
vowels        = [ 'a' , 'e', 'i', 'o', 'u']
eye_colors    = [ "black", "blue", "brown", "green"]
```

Lists: Introduction

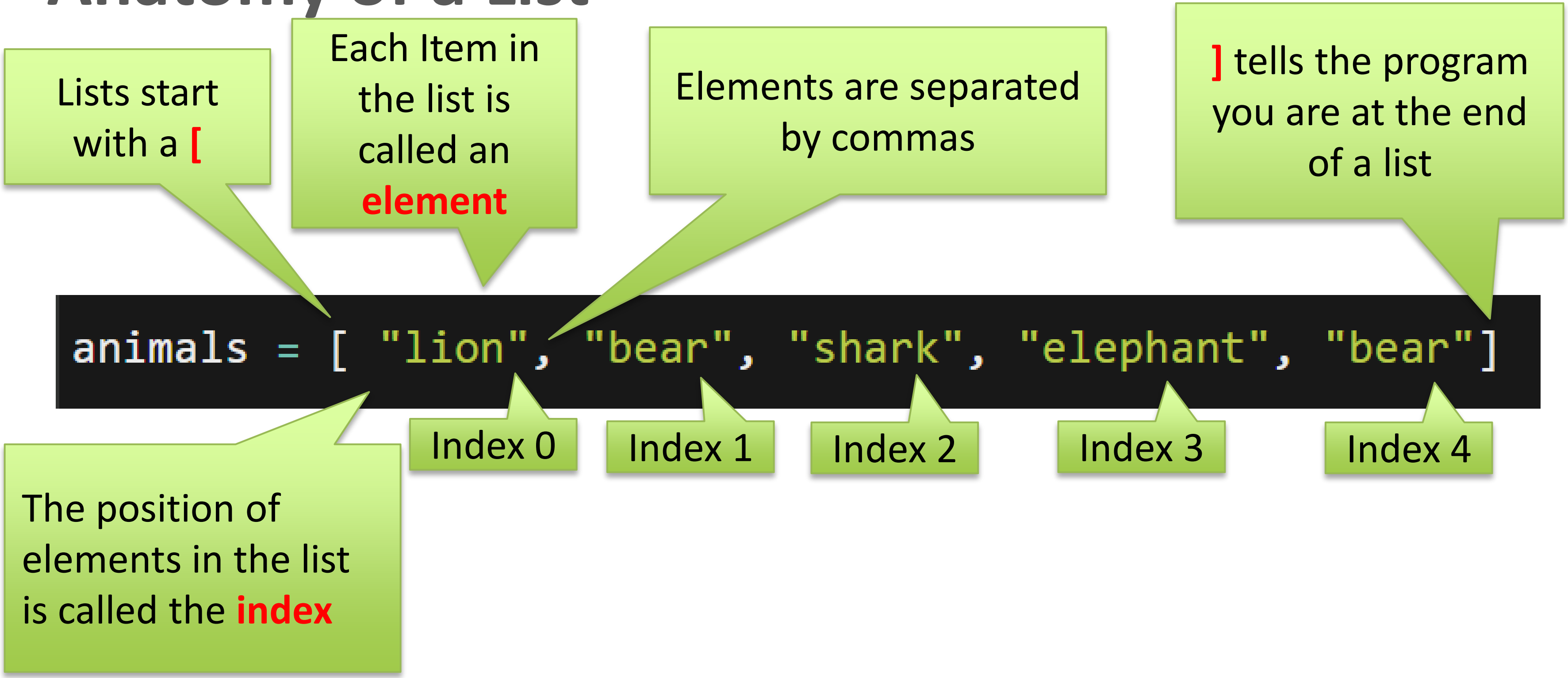
Definition: A **list** is a collection of elements which are organized in order from first to last

Example:



```
animals = [ "lion", "bear", "shark", "elephant", "bear"]
```

Anatomy of a List



Lists start with a [

Each Item in the list is called an **element**

Elements are separated by commas

] tells the program you are at the end of a list

```
animals = [ "lion", "bear", "shark", "elephant", "bear"]
```

Index 0

Index 1

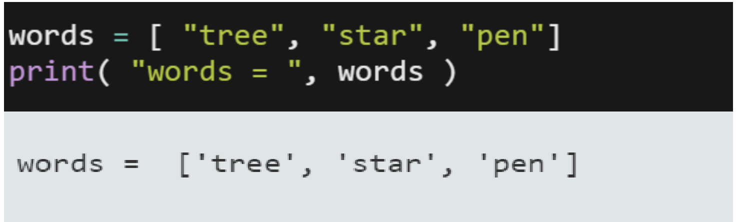
Index 2

Index 3

Index 4

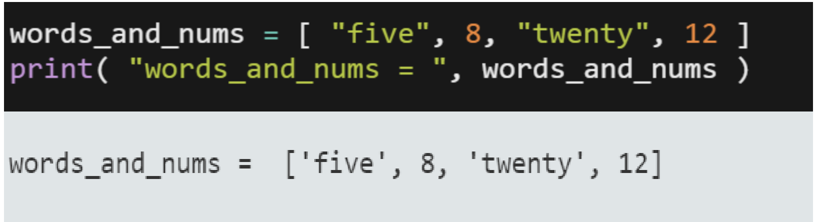
The position of elements in the list is called the **index**

List examples



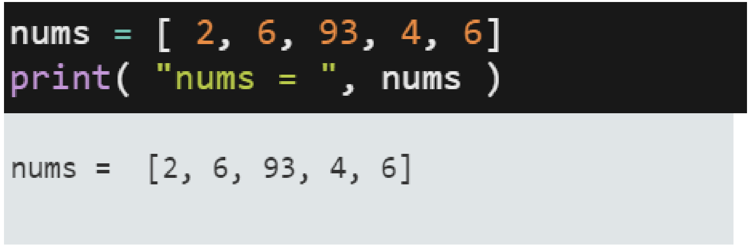
```
words = [ "tree", "star", "pen"]
print( "words = ", words )
```

```
words =  ['tree', 'star', 'pen']
```



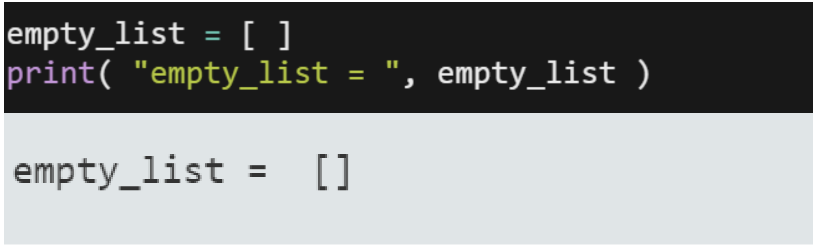
```
words_and_nums = [ "five", 8, "twenty", 12 ]
print( "words_and_nums = ", words_and_nums )
```

```
words_and_nums =  ['five', 8, 'twenty', 12]
```



```
nums = [ 2, 6, 93, 4, 6]
print( "nums = ", nums )
```

```
nums =  [2, 6, 93, 4, 6]
```



```
empty_list = [ ]
print( "empty_list = ", empty_list )
```

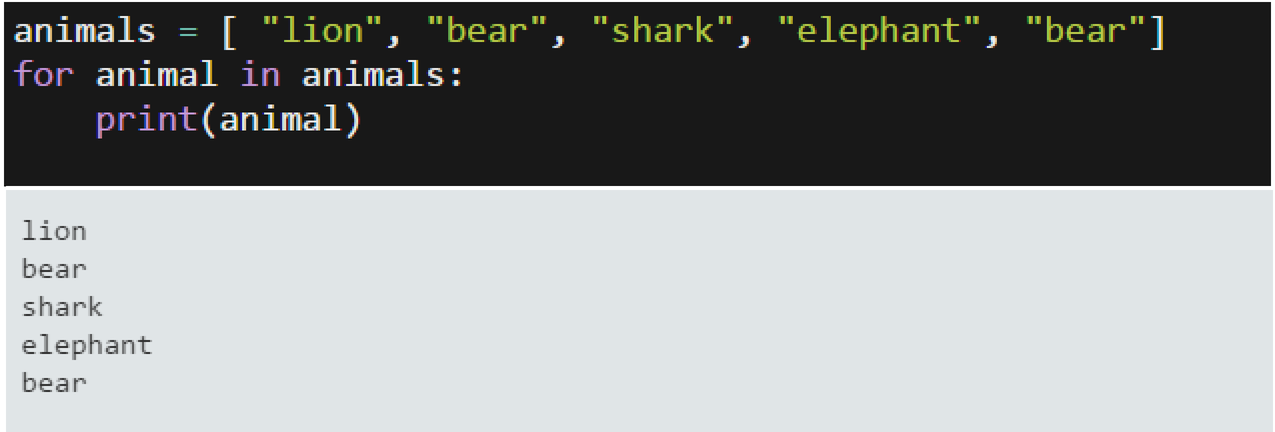
```
empty_list =  []
```

Iterating a list with for Loops

for element **in** list:

do something for each element

Example



```
animals = [ "lion", "bear", "shark", "elephant", "bear"]
for animal in animals:
    print(animal)
```

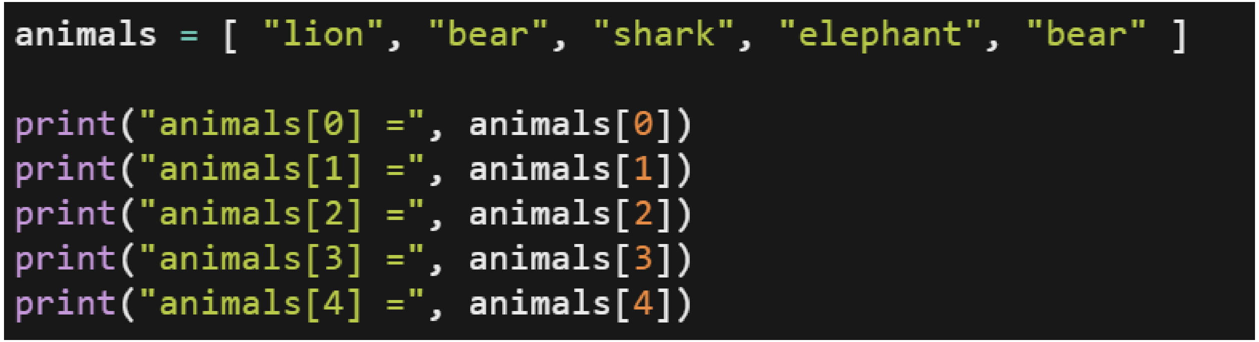
```
lion
bear
shark
elephant
bear
```

Accessing elements in a List

- To access a value in a list, use the square brackets with the index of the element you want

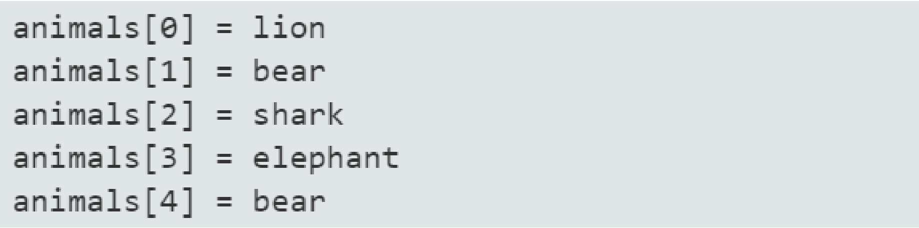
  list[index]

Accessing elements in a list



```
animals = [ "lion", "bear", "shark", "elephant", "bear" ]

print("animals[0] =", animals[0])
print("animals[1] =", animals[1])
print("animals[2] =", animals[2])
print("animals[3] =", animals[3])
print("animals[4] =", animals[4])
```



```
animals[0] = lion
animals[1] = bear
animals[2] = shark
animals[3] = elephant
animals[4] = bear
```

In Class Practice

Work on In Class Exercises.

List Functions

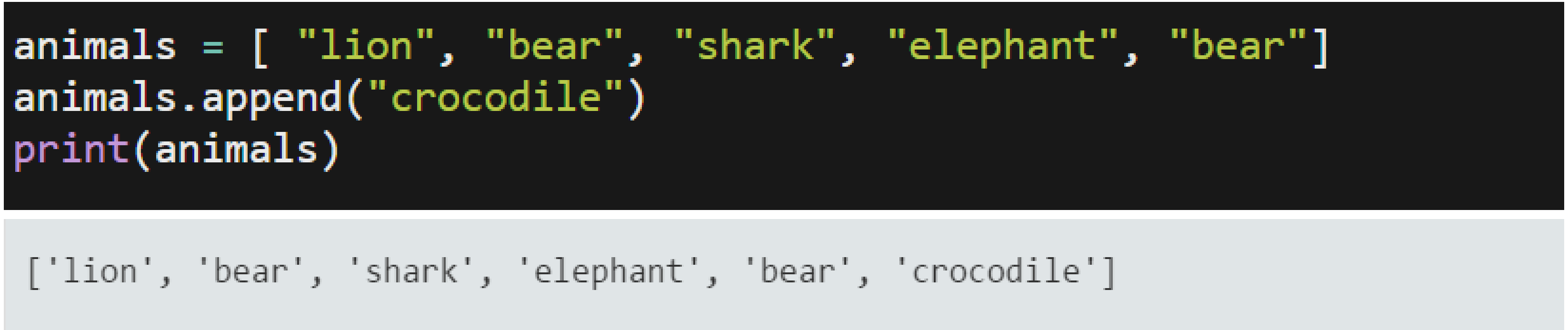
Python provides many built-in functions you can use with lists.

- Add an element to the end of a list
  - list.append(x)
- Remove an element from a list
  - list.remove(x)
- Organize a list
  - list.sort()

list.append(x)

- Add an element to the end of the list

list.append(x)



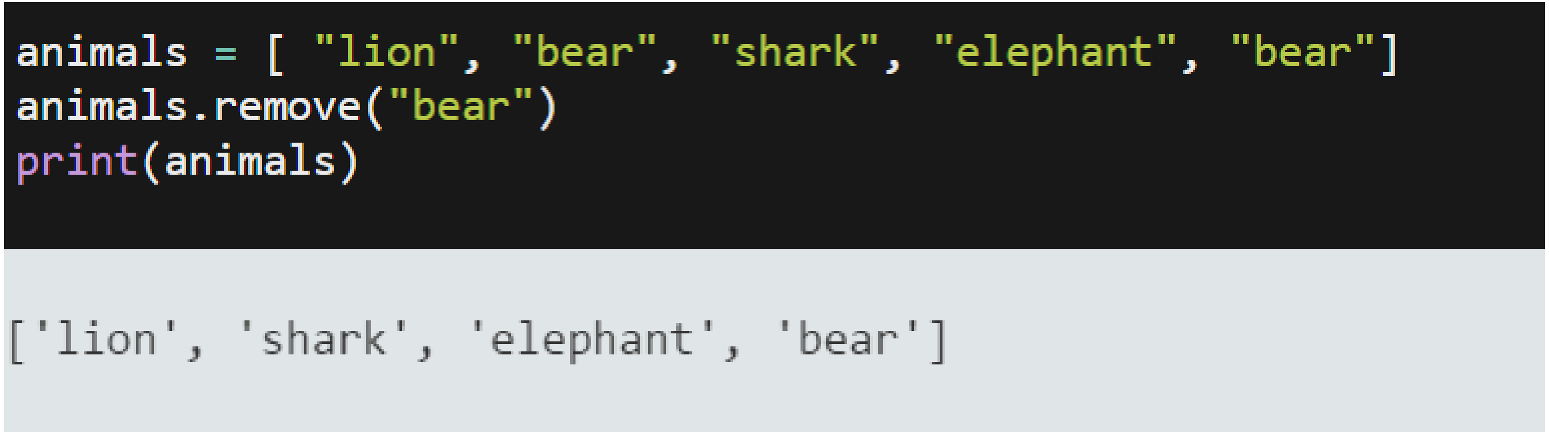
```
animals = [ "lion", "bear", "shark", "elephant", "bear"]
animals.append("crocodile")
print(animals)
```

```
['lion', 'bear', 'shark', 'elephant', 'bear', 'crocodile']
```

list.remove(x)

- Remove the **first** element from the list whose value is x.
- If no element in the list looks like x, then an error will occur

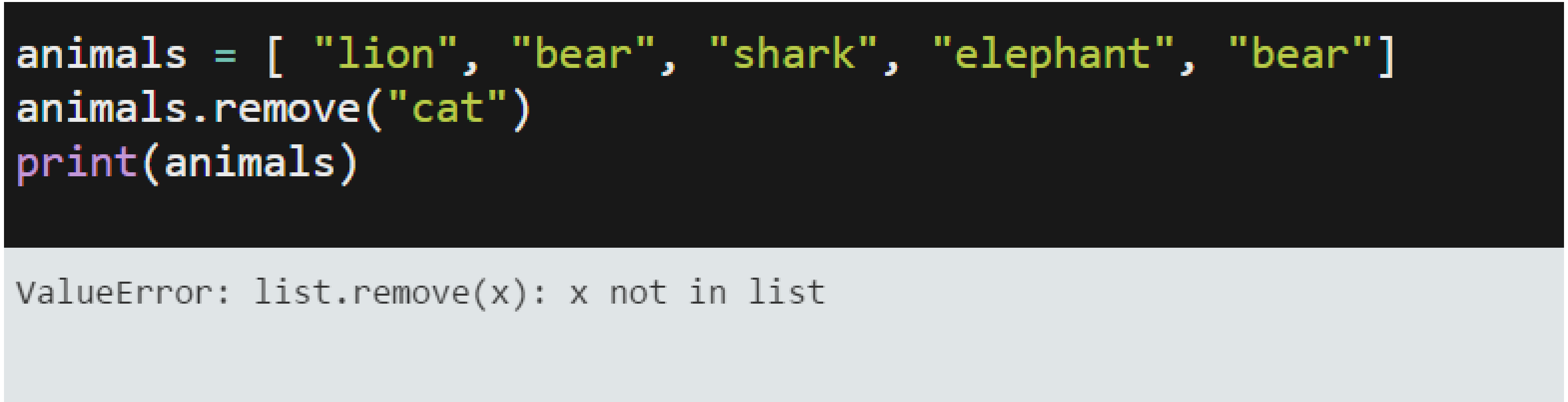
list.remove(x)



```python
animals = [ "lion", "bear", "shark", "elephant", "bear"]
animals.remove("bear")
print(animals)
```

```
['lion', 'shark', 'elephant', 'bear']
```

list.remove(x)



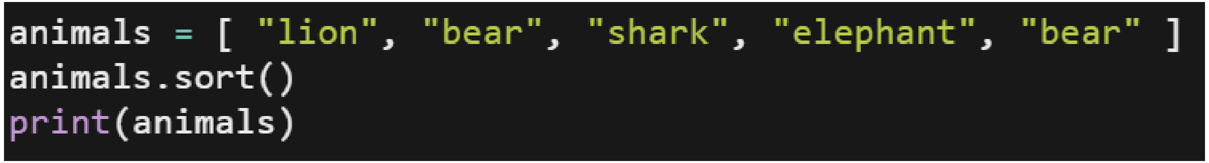
```python
animals = [ "lion", "bear", "shark", "elephant", "bear"]
animals.remove("cat")
print(animals)
```

```
ValueError: list.remove(x): x not in list
```

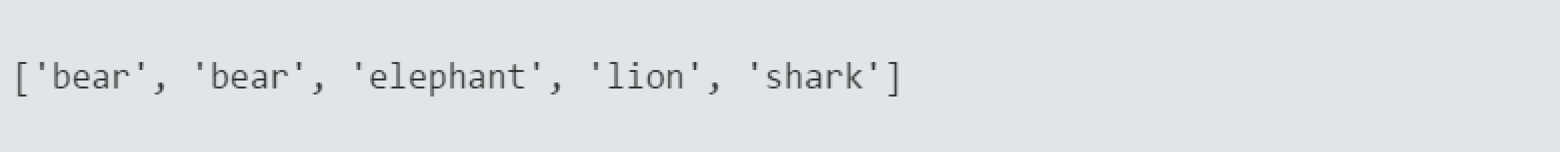
list.sort()

- Sorts the elements in the list

list.sort()

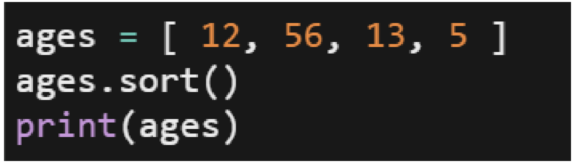


```
animals = [ "lion", "bear", "shark", "elephant", "bear" ]
animals.sort()
print(animals)
```

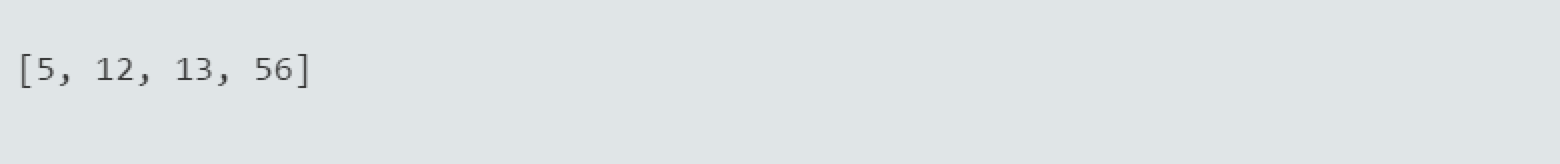


```
['bear', 'bear', 'elephant', 'lion', 'shark']
```

list.sort()



```
ages = [ 12, 56, 13, 5 ]
ages.sort()
print(ages)
```

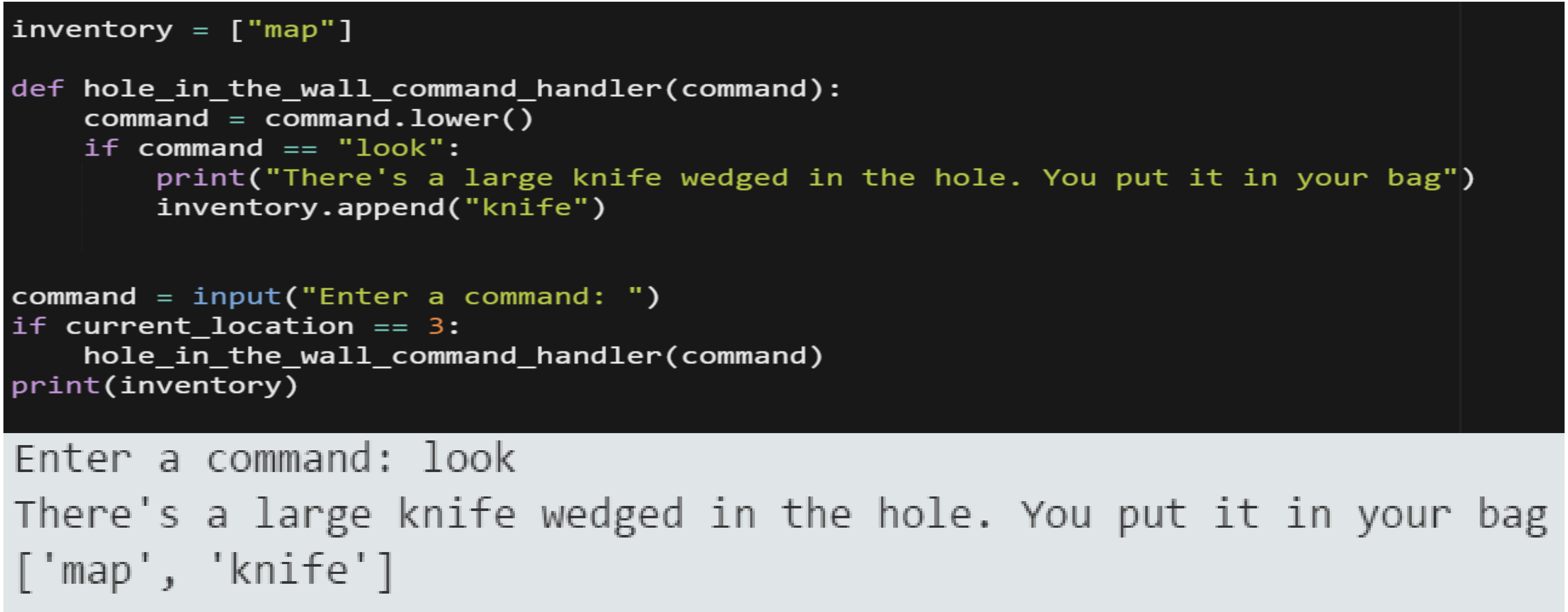


```
[5, 12, 13, 56]
```

In Class Practice

Work on In Class Exercises.

Python Explorer Game



```python
inventory = ["map"]

def hole_in_the_wall_command_handler(command):
    command = command.lower()
    if command == "look":
        print("There's a large knife wedged in the hole. You put it in your bag")
        inventory.append("knife")


command = input("Enter a command:  ")
if current_location == 3:
    hole_in_the_wall_command_handler(command)
print(inventory)
```

```
Enter a command: look
There's a large knife wedged in the hole. You put it in your bag
['map', 'knife']
```

Recap

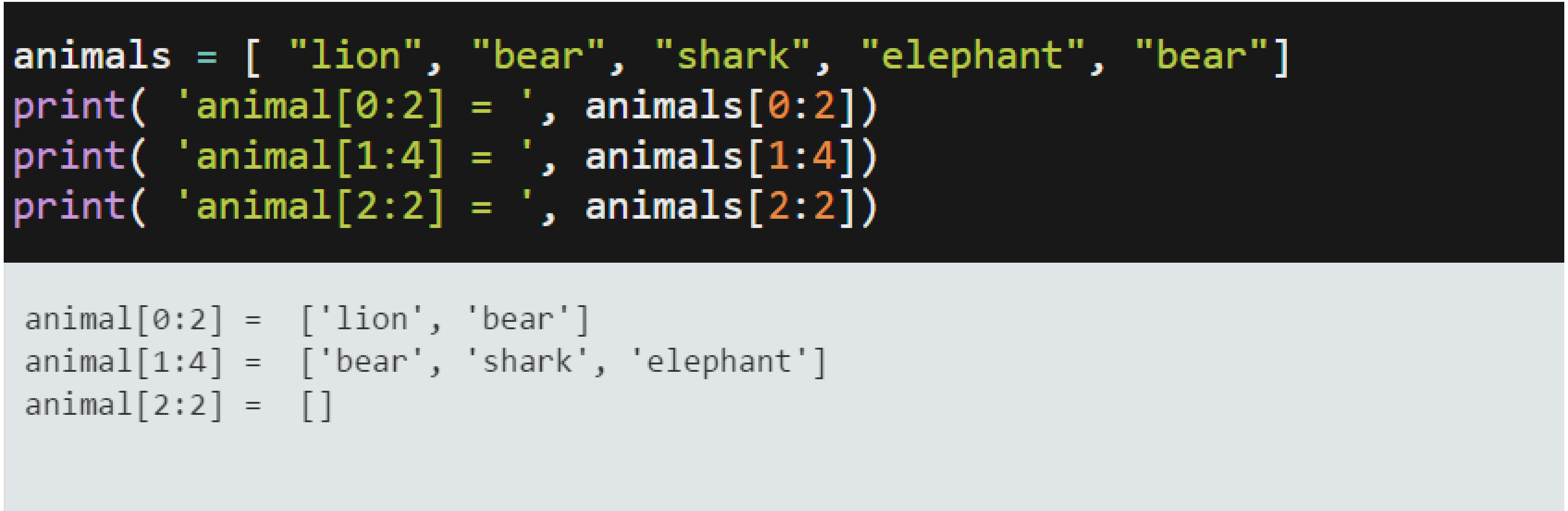
- A **list** is a convenient way to store an ordered collection of data
- It is possible to retrieve a value from a list by specifying the index
- A range of functions are available to allow you to modify a list

Slicing Lists

- To access a range of values in lists, use the square brackets with the indices of the range of elements you want

  list[start_index: end_index]

Slicing Lists



```
animals = [ "lion", "bear", "shark", "elephant", "bear"]
print( 'animal[0:2] = ', animals[0:2])
print( 'animal[1:4] = ', animals[1:4])
print( 'animal[2:2] = ', animals[2:2])
```

```
animal[0:2] =  ['lion', 'bear']
animal[1:4] =  ['bear', 'shark', 'elephant']
animal[2:2] =  []
```

List Functions

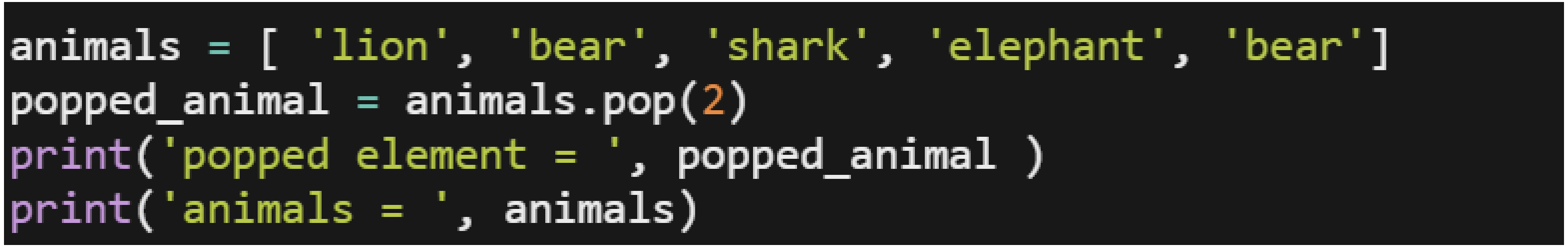
Python provides many built-in functions you can use with lists.

- Add elements to a list
  - list.append(x)
  - list.extend(x)
- Remove elements from a list
  - list.remove(x)
  - list.pop(x)
- Organize a list
  - list.sort()
  - list.reverse()
- Search a list
  - list.index(x)

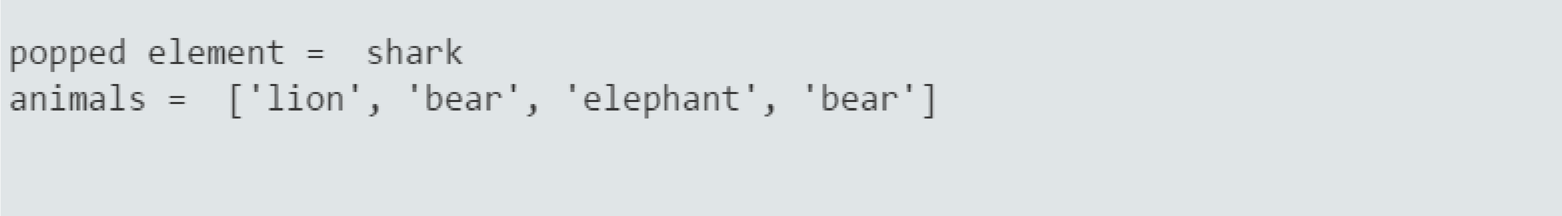
list.pop(x)

- Remove an element at index x from the list and return it
- If x isn’t specified, removes the last element from the list and returns it

list.pop(x)

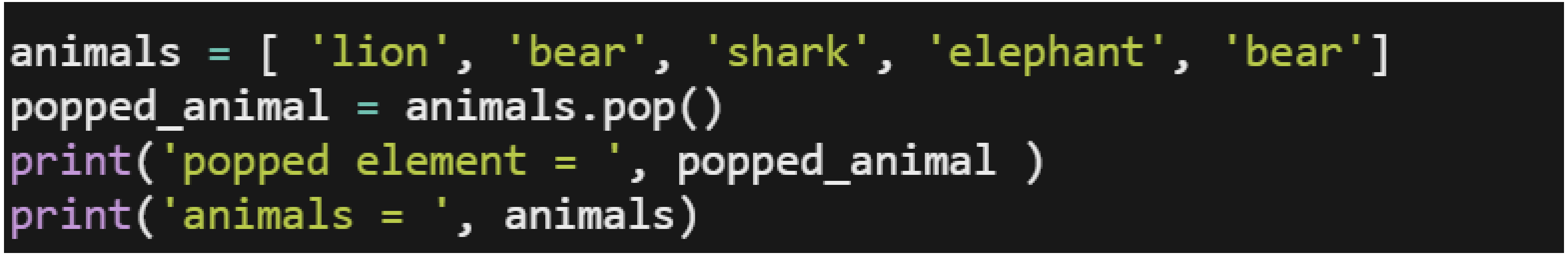


```
animals = [ 'lion', 'bear', 'shark', 'elephant', 'bear']
popped_animal = animals.pop(2)
print('popped element = ', popped_animal )
print('animals = ', animals)
```

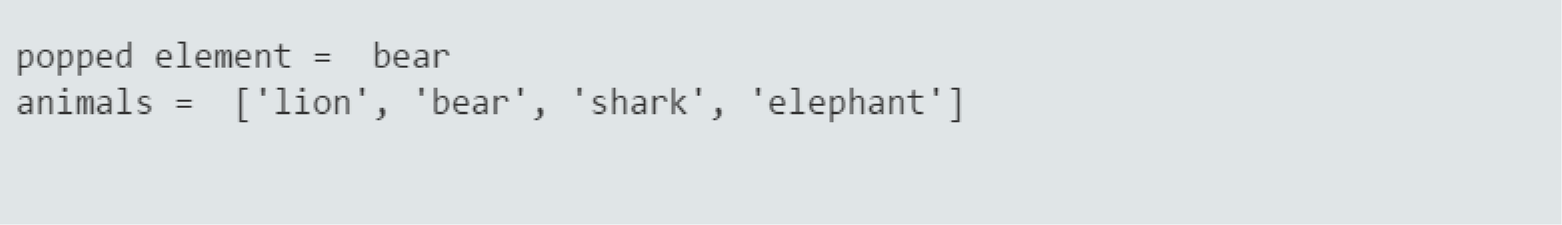


```
popped element =  shark
animals =  ['lion', 'bear', 'elephant', 'bear']
```

list.pop()



```python
animals = [ 'lion', 'bear', 'shark', 'elephant', 'bear']
popped_animal = animals.pop()
print('popped element = ', popped_animal )
print('animals = ', animals)
```

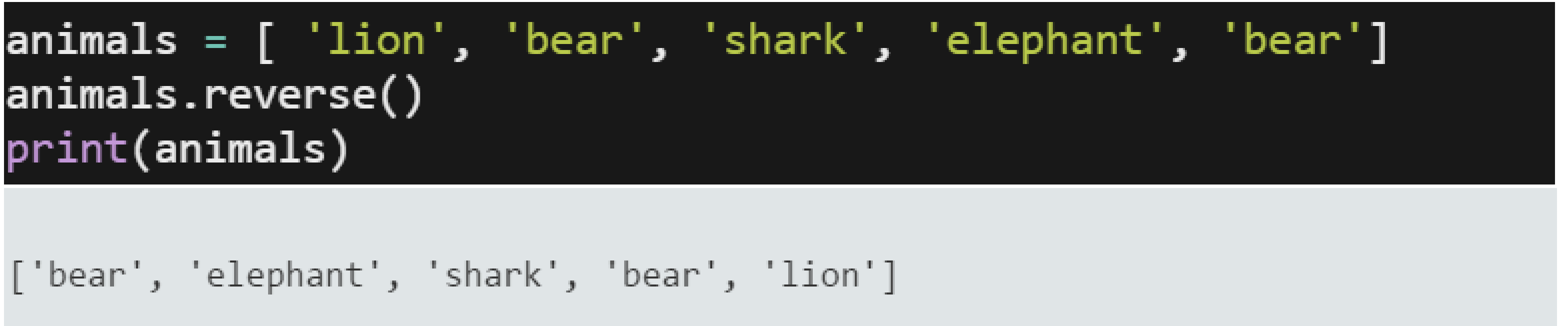


```
popped element =  bear
animals =  ['lion', 'bear', 'shark', 'elephant']
```

list.reverse()

- Reverse the elements in the list

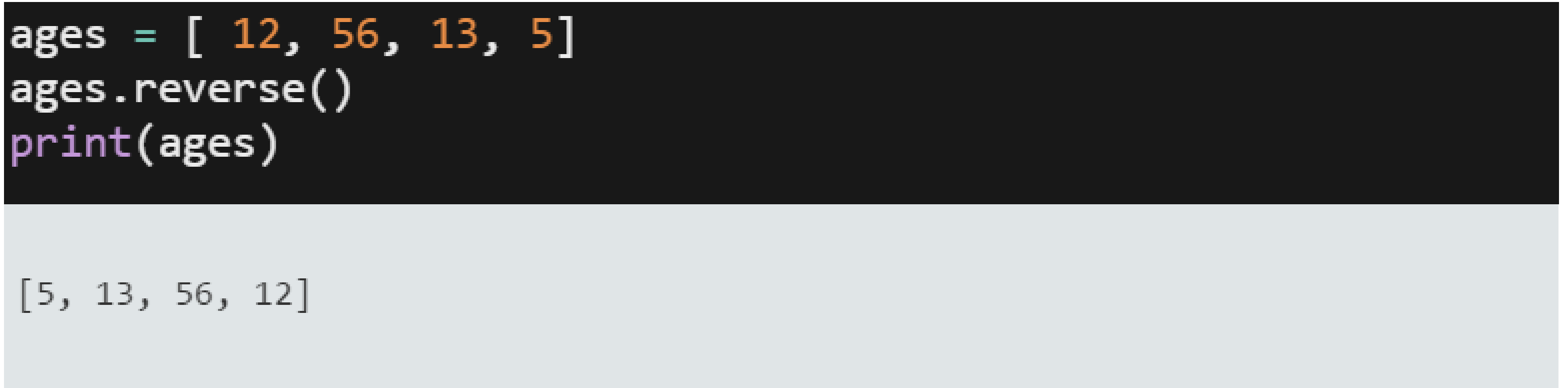
list.reverse()



```
animals = [ 'lion', 'bear', 'shark', 'elephant', 'bear']
animals.reverse()
print(animals)
```

```
['bear', 'elephant', 'shark', 'bear', 'lion']
```

list.reverse()



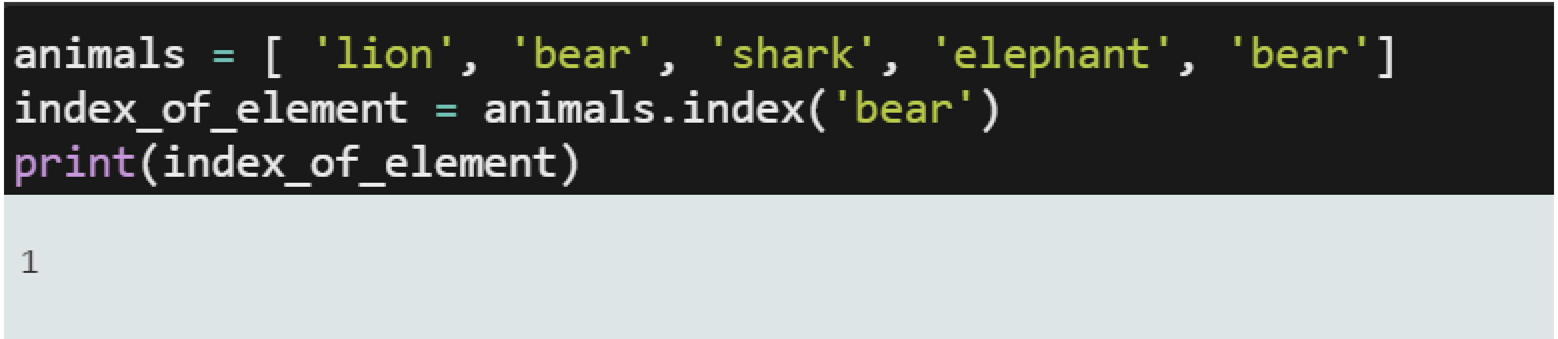
```
ages = [ 12, 56, 13, 5]
ages.reverse()
print(ages)
```

```
[5, 13, 56, 12]
```

list.index(x)

- Return the index of the **first** element from the list whose value is x.
- If no element in the list looks like x, then an error occurs

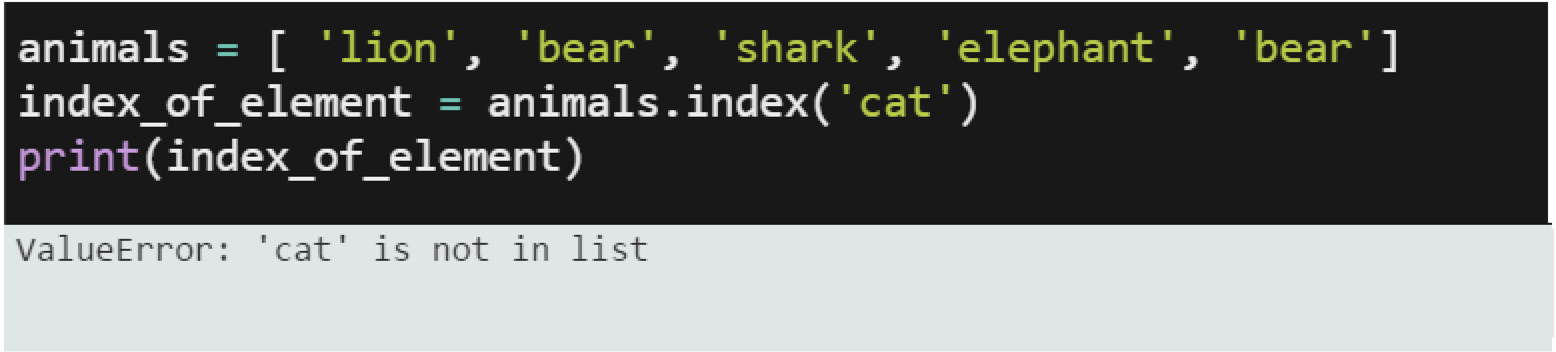
list.index(x)



```
animals = [ 'lion', 'bear', 'shark', 'elephant', 'bear']
index_of_element = animals.index('bear')
print(index_of_element)
```

```
1
```

list.index(x)



```
animals = [ 'lion', 'bear', 'shark', 'elephant', 'bear']
index_of_element = animals.index('cat')
print(index_of_element)
```

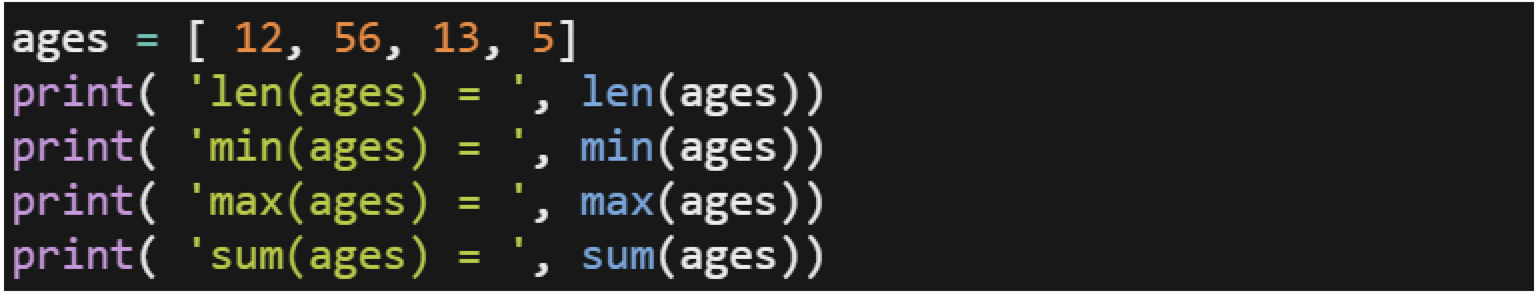
```
ValueError: 'cat' is not in list
```

List Functions

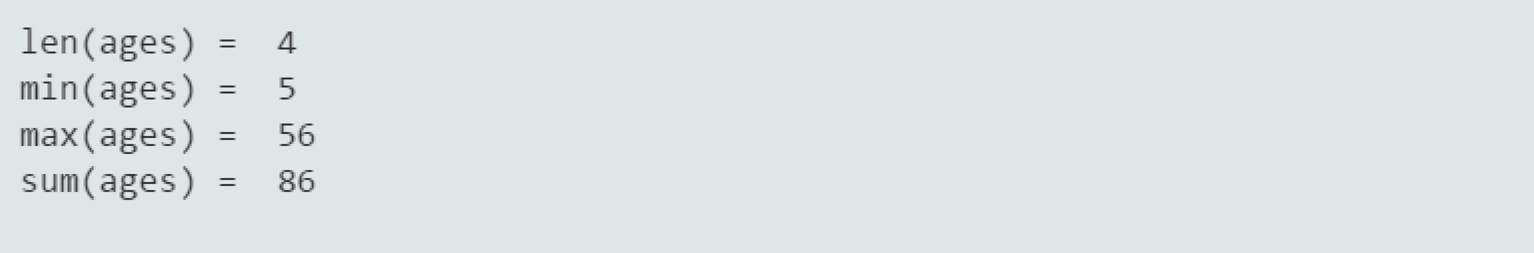
Python provides several functions that operate on lists:

- len(list)
- max(list)
- min(list)
- sum(list)

List Functions



```
ages = [ 12, 56, 13, 5]
print( 'len(ages) = ', len(ages))
print( 'min(ages) = ', min(ages))
print( 'max(ages) = ', max(ages))
print( 'sum(ages) = ', sum(ages))
```



```
len(ages) =  4
min(ages) =  5
max(ages) =  56
sum(ages) =  86
```