Welcome to Week 5

Tutorial 11

Lists

A list is an ordered collection of objects. ie:

```
[True,  3,  “bob”, 2.39]

[“Sara”, “Anna”, “Karen”]

[2, 4, 8, 16, 32, 64]
```

Lists

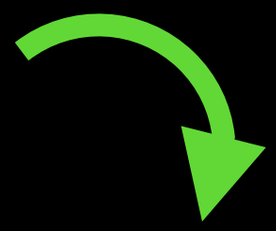
We can also slice and index lists the way we do with strings

```
myList = [True,  3,  “bob”, 2.39]

myList[:2]  →  [True, 3]
```

List Operations

We can concatenate two lists
using the + operator.

```
Girls = [“Sara”,  “Anna”,  “Karen”]
Boys  = [“Billy”,  “John”]

Girls + Boys
```

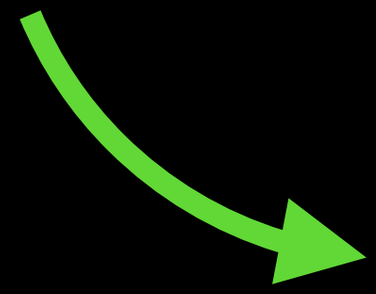


```
[“Sara”, “Anna”, “Karen”, “Billy”, “John”]
```

List Operations

We can repeat elements within a list using the * operator.

```
["Sara"] * 3
```



```
["Sara", "Sara", "Sara"]
```

We can check for membership within a list using the in keyword.

Lists and Strings

Both Lists and Strings are actually Arrays (Ordered Collections)

A String is an array of characters

“H”	“E”	“L”	“L”	“O”	“ “	“W”	“O”	“R”	“L”	“D”

A list is an array of objects

“Hello”	3.14	True	42	False	“bloop”

List Functions

`cmp(list1, list2)`

Compares the two given lists

`min(list)`

Returns the minimum in the list

`max(list)`

Returns the maximum in the list

`len(list)`

Returns the length of the list

List Methods

- `List.append(obj)`

  Inserts obj to the end of the list

  `List.count(obj)`

  Returns the number of occurrences of obj

  `List.index(obj)`

  Returns the first index of obj in the list

- `List.insert(index,obj)`

  Inserts obj at the given index in the list

List Methods

`List.pop()`

Removes and returns the last object in list

- `List.remove(obj)`

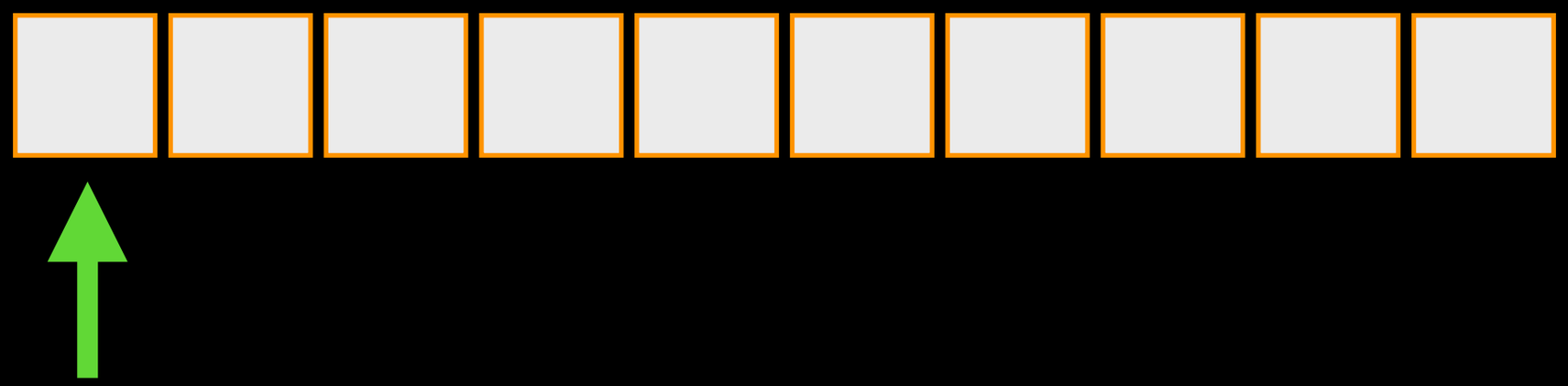
  Removes obj from the list

- `List.reverse(obj)`

  Reverses the order of objects in the list

Iterating Through a List

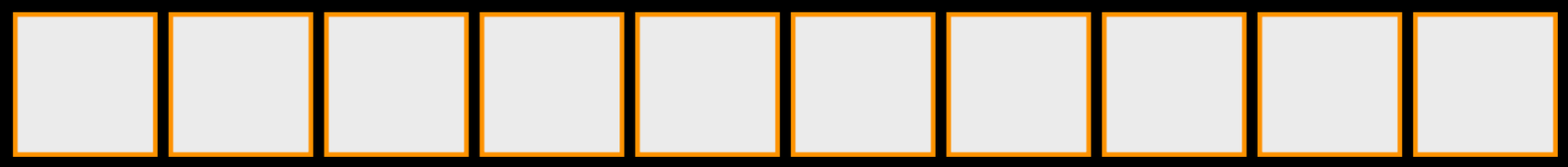
myList:




```
1   for i in myList:
```

Iterating Through a List

myList:

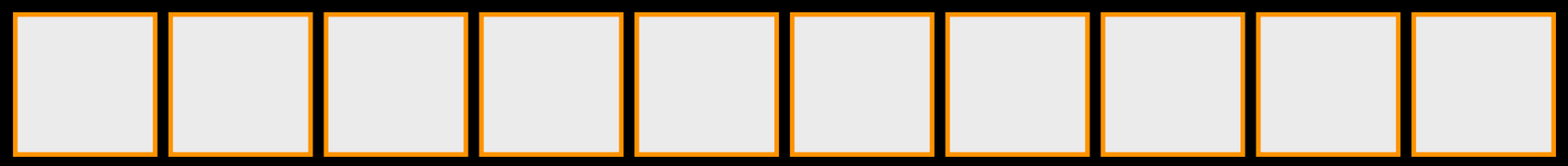


i:

```
1   for i in myList:
```

Iterating Through a List

myList:

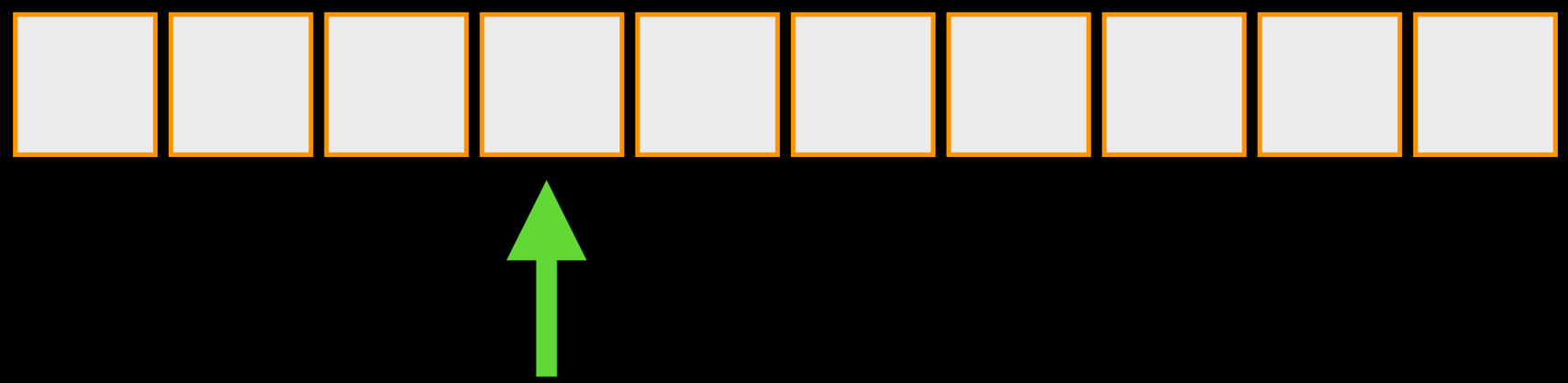


i:

```
1   for i in myList:
```

Iterating Through a List

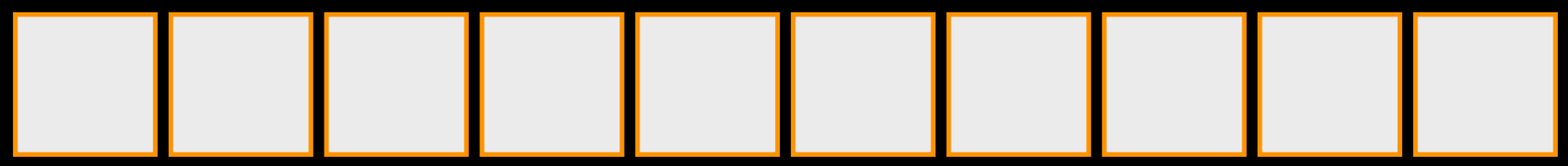
myList:




```
1   for i in myList:
```

Iterating Through a List

myList:

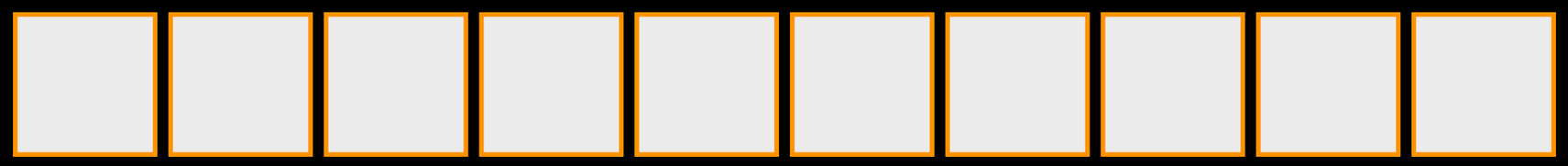


i:

```
1   for i in myList:
```

Iterating Through a List

myList:

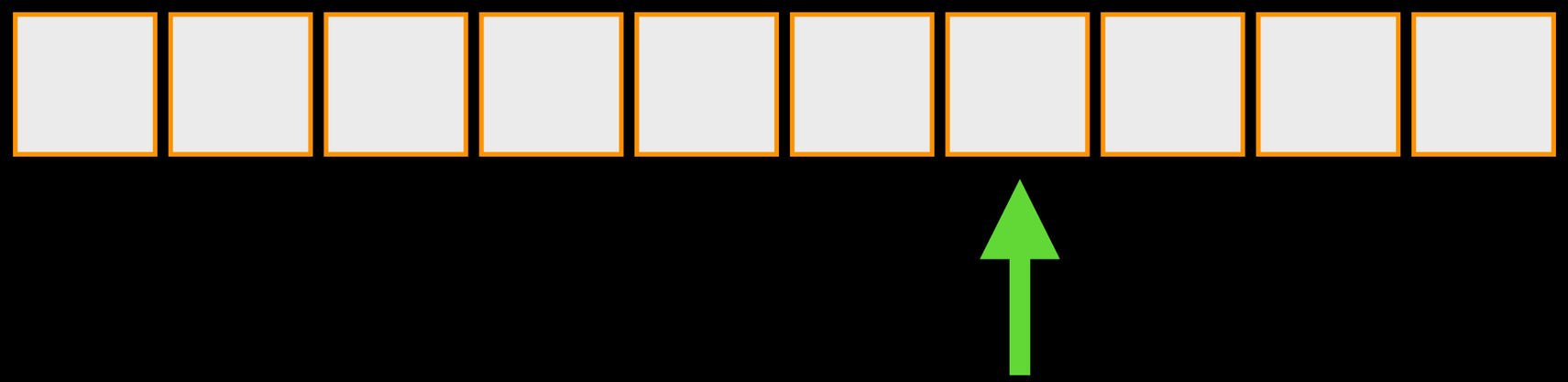


i:

```
1   for i in myList:
```

Iterating Through a List

myList:

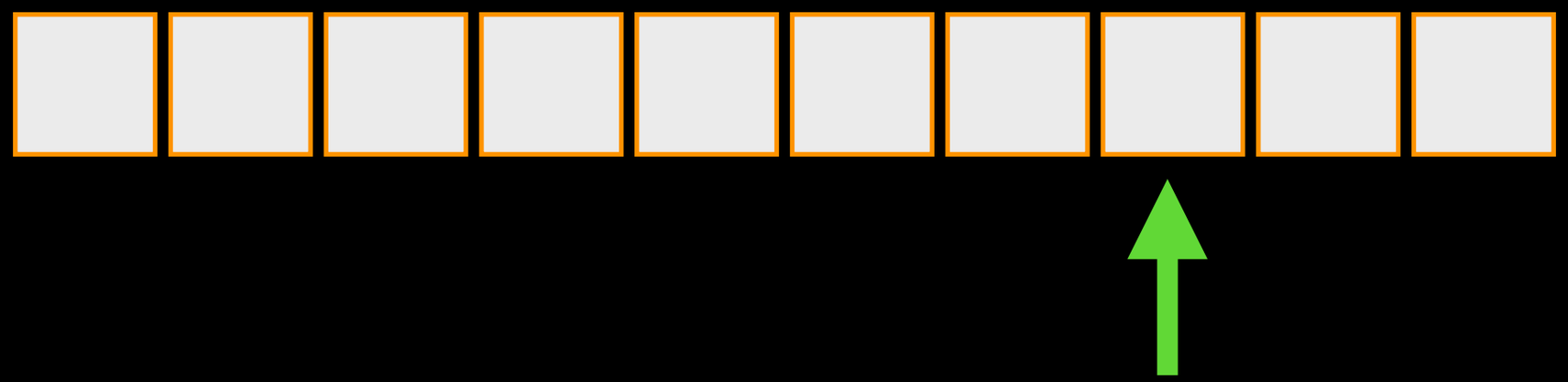


i:

```
1   for i in myList:
```

Iterating Through a List

myList:

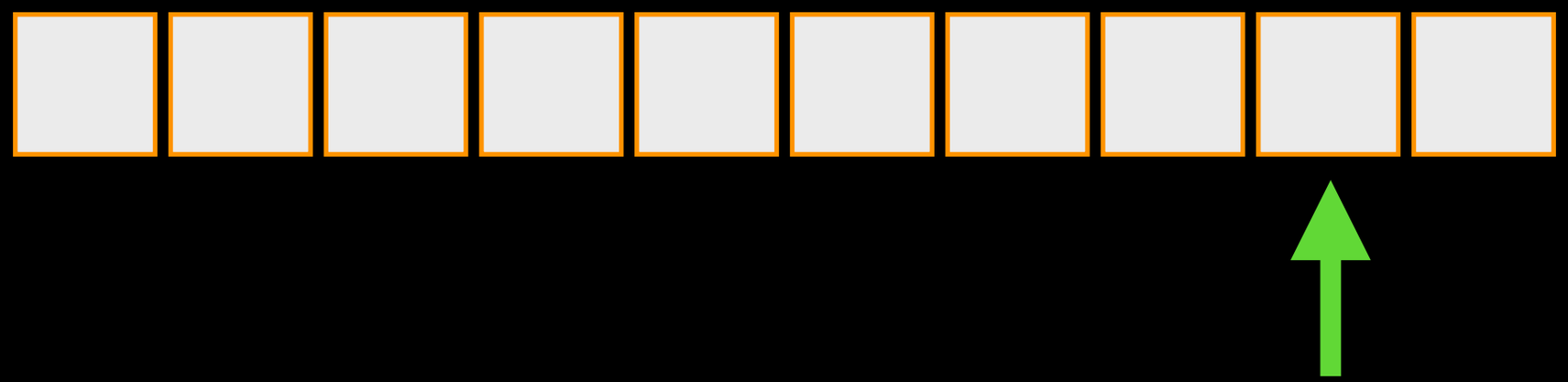


i:

```
1   for i in myList:
```

Iterating Through a List

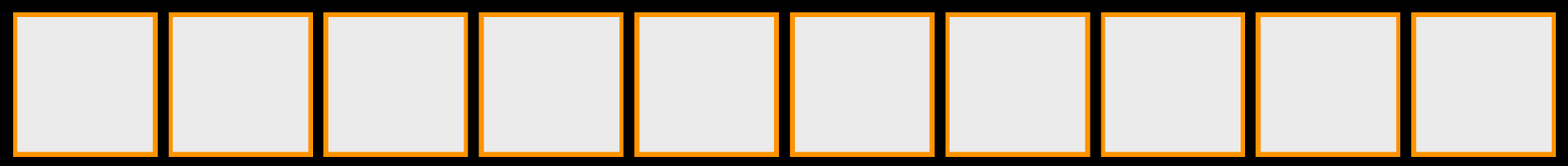
myList:




```
1   for i in myList:
```

Iterating Through a List

myList:



i:

```
1   for i in myList:
```

Weekly Challenge

A lot of different names can be made by *taking a consonant and adding "ara"* to it.
ie: `Bara, Cara, Dara, Fara, … , Zara`

Using only 4 lines of code:

- Make "`ara`" names for every consonant, (Including Y) and add them to an empty list

- Print the completed list of names.

Using only 4 lines of code:

- Make “ara” names for every consonant, (Including Y) and add them to an empty list

- Print the completed list of names.

```
1  list = []
2  for letter in “BCDFGHJKLMNPQRSTVWXYZ”:
3      list.append(letter + “ara”)
4  print(list)
```