Bloomberg

STARTUP

Bloomberg

STARTUP

Python Lesson

Lists and For Loops 1

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!”`

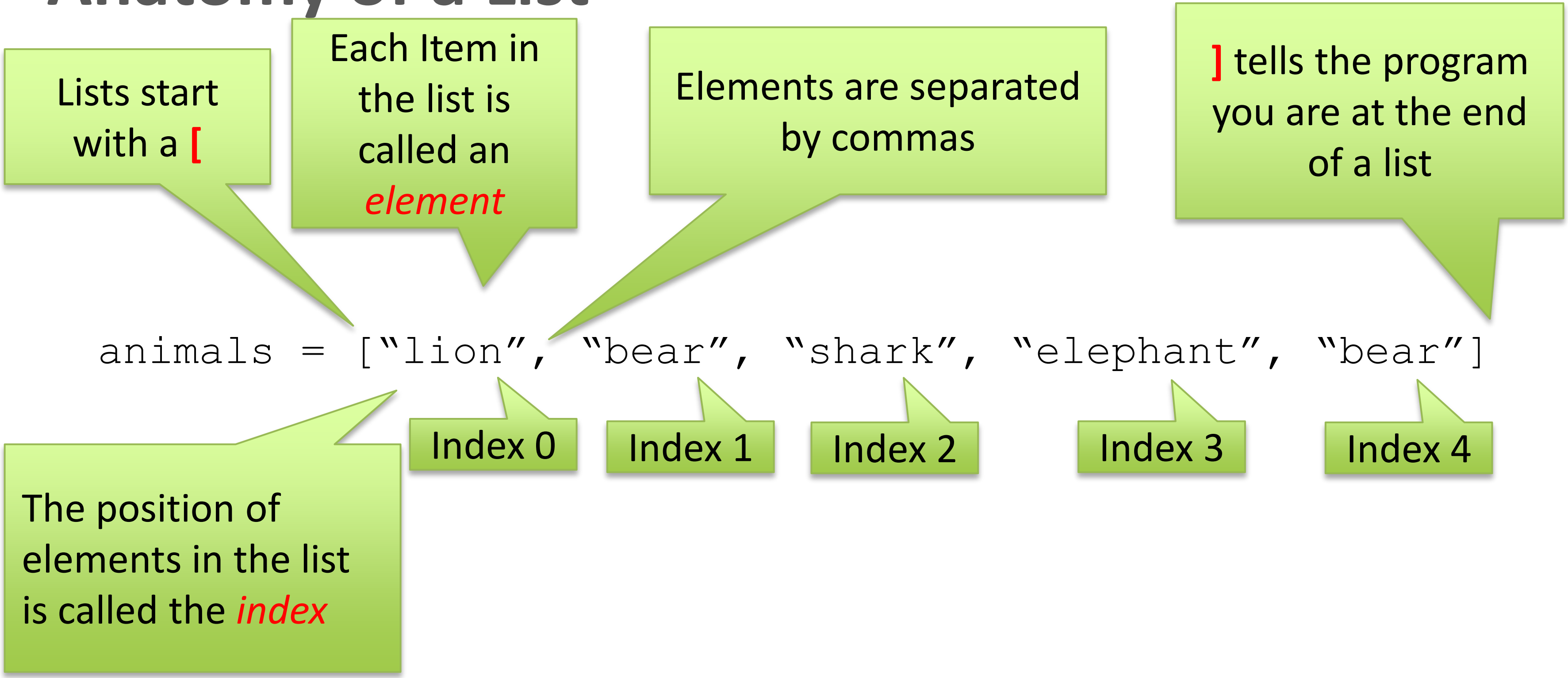
Lists: Introduction

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"]
```

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

List examples

```
words = [“tree”, “star”, “pen”]

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

words_and_nums = [“five”, 8, “twenty”, 12]

empty_list = []
```

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(“First element =”, animals[0])
print(“Second element =”, animals[1])
print(“Third element =”, animals[2])
print(“Fourth element =”, animals[3])
print(“Fifth element =”, animals[4])
```

```
First element = lion
Second element = bear
Third element = shark
Fourth element = elephant
Fifth element = bear
```

For Loops: Introduction

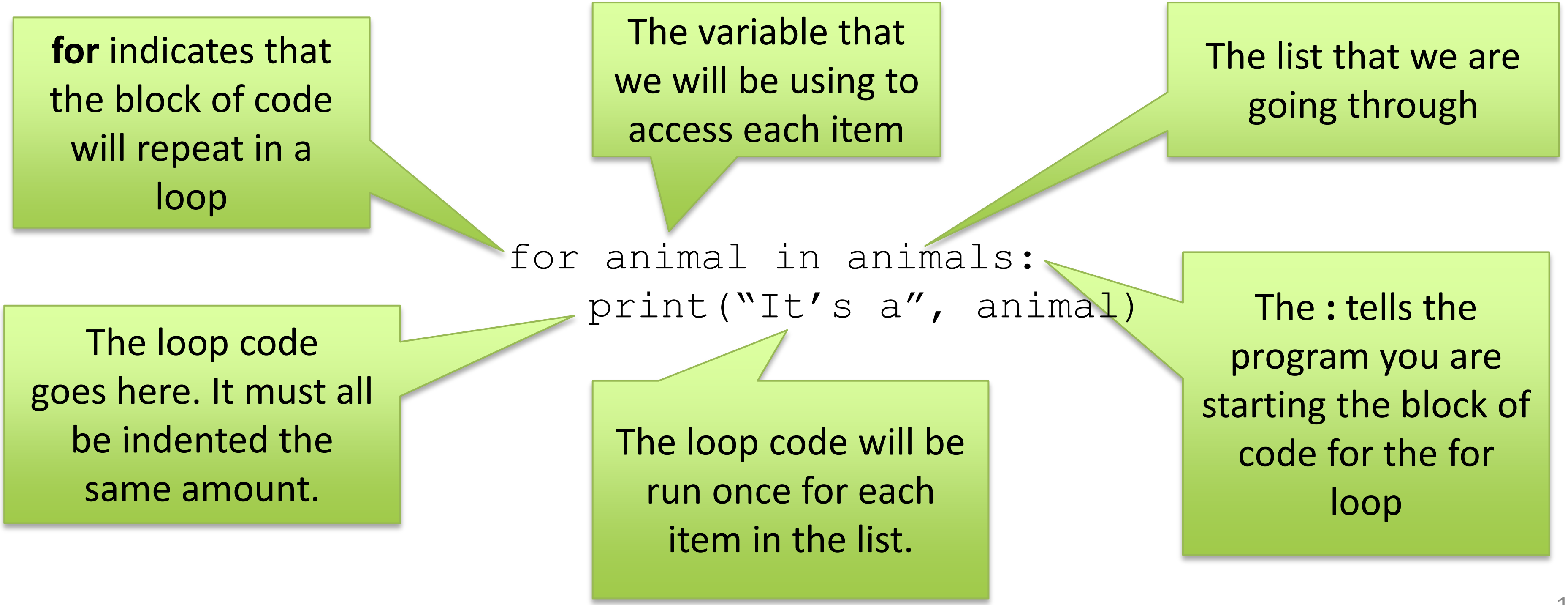
There’s an easier way to do something to each element of a list!

For loops allow us to repeat a block of code for each item in a list.

```
animals= [“lion”, “bear”, “shark”, “elephant”, “bear”]
for animal in animals:
   print(“It’s a”, animal)
```

Bloomberg STARTUP

Anatomy of a For loop



```
for animal in animals:
         print("It's a", animal)
```

for indicates that the block of code will repeat in a loop

The variable that we will be using to access each item

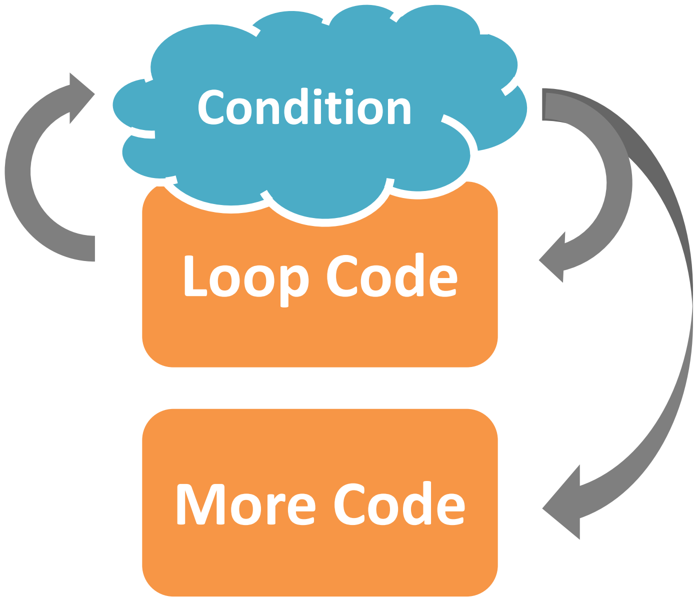
The list that we are going through

The loop code goes here. It must all be indented the same amount.

The loop code will be run once for each item in the list.

The **:** tells the program you are starting the block of code for the for loop

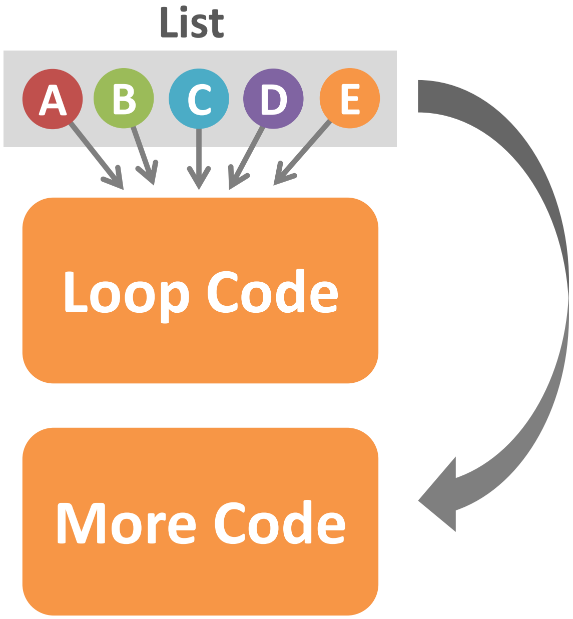
While Loops VS For Loops



Condition

Loop Code

More Code



List

A B C D E

Loop Code

More Code

Example

Write a for loop that iterates over the list below and adds up all the numbers in the list.

```
numbers = [3, -5, -7, 11, 2]
sum = 0
for number in numbers:
    sum = sum + number
print(sum)
```

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
- **For loops** let you execute a block of code once for every item in a list.
- To write a for loop:

```
for list_element in list:
    # do something to list_element
```