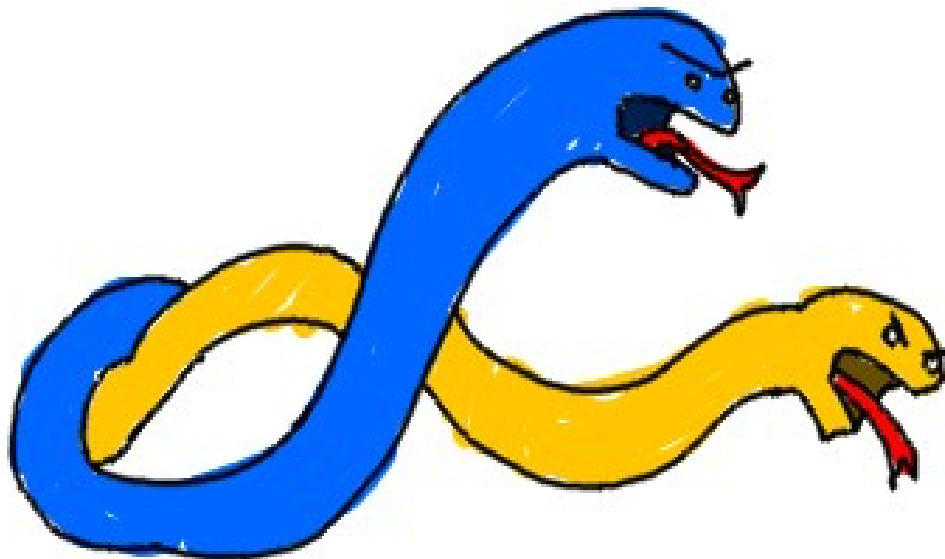


# Python vs. C++



# Arrays

- C++ doesn't have built-in data-structures like Python's List or Dictionary.
- Similar things exist in the STL (Standard Template Library), and we can write data-structures like these ourselves using classes (& other things).
- C++ has Arrays, which are *similar* to Lists.



# Arrays

- C++ Arrays are a fixed-size; you set a size when you declare it and that size cannot change.
- C++ Arrays can only store one type of datatype, unlike Python.

# Arrays

## Creating a list in Python:

```
bookChapters = [ "C++ Basics", "Flow of Control", "Function Basics" ]
```

## Creating an array in C++:

```
string bookChapters[] = { "C++ Basics", "Flow of Control", "Function Basics" };
```

If you ***initialize*** the array (where we're storing values in it on the same line we declare it), we don't need to specify the array size; it will automatically take on the size of the amount of elements you give it.

In this example, bookChapters will automatically be given a size of 3.



# Arrays

## Declaring Arrays:

```
string bookChapters[] = { "C++ Basics", "Flow of Control", "Function Basics" };  
float prices[20];
```

If you're not going to initialize the array after declaring it, you need to specify a size inside of the square brackets [ ].

Here, we have **float prices[20];**. After we declare this array, we can store **float** values in the **prices** array. Valid array indices are between 0 and 19 (inclusive).

```
prices[0] = 9.99;  
prices[1] = 13.99;  
prices[2] = 3.99;  
...  
prices[19] = 4.99;
```

The square brackets [ ] are known as the **subscript operator**.

# Arrays

## Dynamic Arrays & Linked Lists

Later in the class, we will learn how to deal with **memory management**, which will allow us to create **dynamic arrays**. This will allow us to create an array and not immediately specify its size. However, these won't resize on their own and you have to copy values to-and-fro and reallocate memory to resize them.

We will also learn how to create **linked list** data structures, which allow us to store data in a **resizable list**.

By creating objects with **classes**, and knowing how to deal with **memory management** with **pointers**, we can create more functional data structures.



# Arrays

## Additional Reading

- <http://www.cplusplus.com/doc/tutorial/arrays/>
- <http://www.learncpp.com/cpp-tutorial/61-arrays-part-i/>