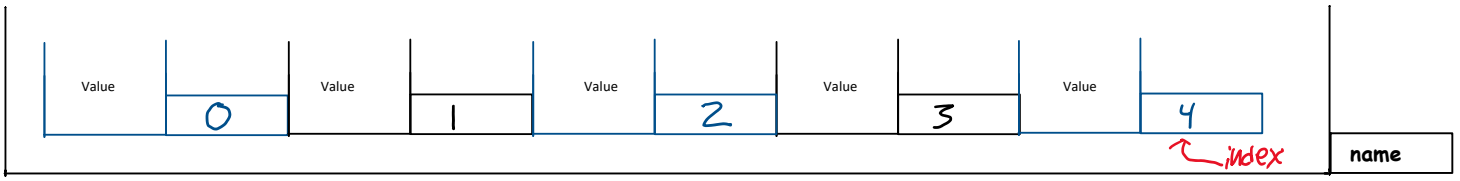
# Arrays

Sunday, October 14, 2018 6:13 PM




Since we look at variables as a box with a name and a value, we can extend the metaphor to arrays as a box of boxes.

## Things to know about arrays

- Array sizes are fixed
- Arrays DO NOT know their size
- You access the contents of an array using "box notation", i.e. name[index]
- You cannot just print an array, you must iterate through each value to print its contents

## Parallel arrays

Arrays give us a unique structure that we can exploit when we have two pieces of data that relate to a third piece. We call this structure a parallel array. Consider the model below:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| "Here" | "is" | "a" | "story" | "all" | "about" | "how" | "My" | "life" | "got" | "flipped" |
| "Here" | "is" | "a" | "story" | "all" | "about" | "how" | "My" | "life" | "got" | "flipped" |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

stringArr1

stringArr2

As you can see above we can use a structure like this to see if set of strings is equivalent to another simply by checking all instances of:
stringArr1[i] == stringArr2[i]
This very simplified example should start the mind working on how else this type of structure can be used. The clue is when you have two different pieces of information that are related by a third piece, such as a count or an instance.

## Basic Forms

```
type name[int]; //declaration with a size declared, remember that arrays do not
                // know their size later, we usually will use a variable to declare
                //and track the size of our array
type name[]; //declaration, we don't really use this until later in the semester
name = new type[int]; //initialization, we don't really use this until later in the semester
for i < arraySize {arrayName[i] = value;} //array initialization, happens using a loop
type name[int] = {value1, value2, … valueInt}; //array definition, declare and initialize in
                                               //one step
name[int] //array access, the int in this case is the index of the array that we want a
          //value from
```