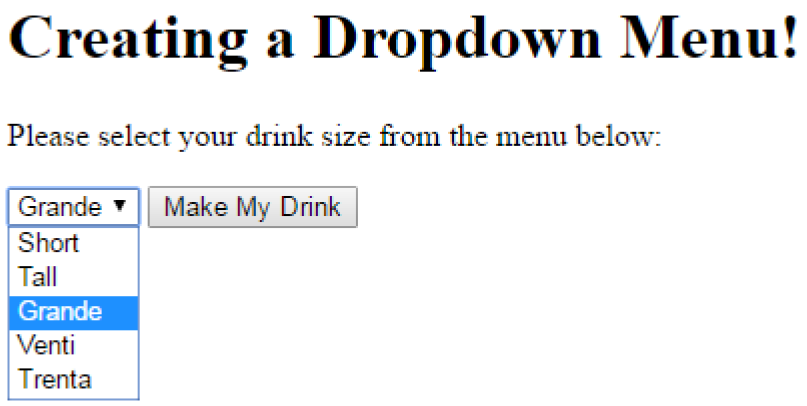
Arrays!

(and drop down boxes!)

Another HTML Thing

- <select>
- Creates a dropdown box on the webpage
  - Allows the user to select from several choices
- The advantage of using select is that we are guaranteed that the input will be one of our choices (no invalid user input)
- To specify options in a dropdown box we use the <option> tag inside the <select> tag

<select>



Creating a Dropdown Menu!

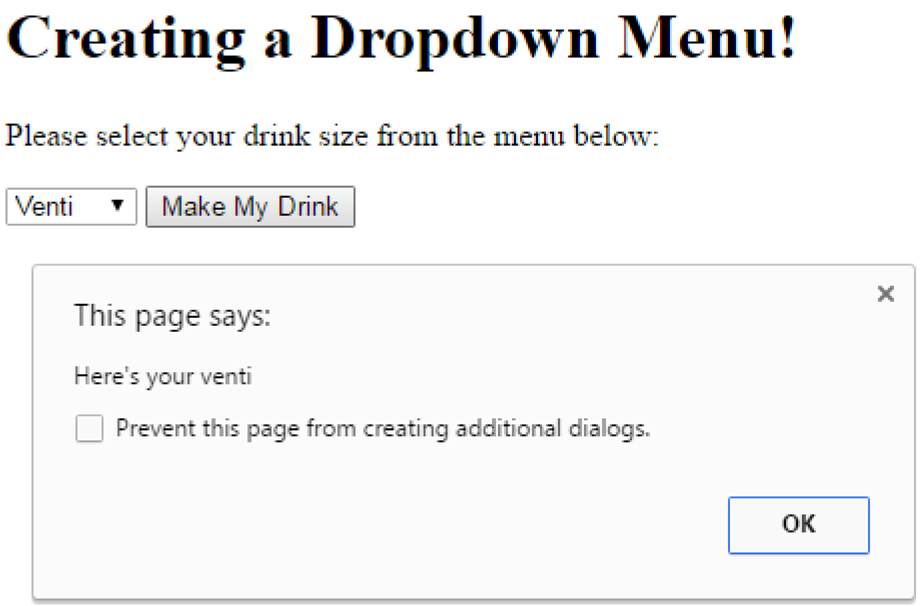
Please select your drink size from the menu below:

```
<h1>Creating a Dropdown Menu!</h1>
<p>Please select your drink size from the menu below: </p>
<select id=“drinkSize”>
    <option value="short">Short</option>
    <option value="tall">Tall</option>
    <option value="grande">Grande</option>
    <option value="venti">Venti</option>
    <option value="trenta">Trenta</option>
</select>
<button type="button" onclick="makeDrink()">Make My
Drink</button>
```

- We must set the value attribute of the option, so we can access the value contained in it using JavaScript

<select>

- How do you think we access the currently selected option in the JavaScript?



Creating a Dropdown Menu!

Please select your drink size from the menu below:

This page says:

Here's your venti

-

```
function makeDrink(){
   var size = 
document.getElementById("drinkSize");
   alert("Here's your " + size.value);
}
```

Remember

- Values
    - Have type
    - Represent a concrete number, string, or boolean
    - Ex. 1, “cow”, true
- Variables
    - Hold values in memory
    - Can change their values
    - Are used to carry information through our programs

Arrays Make Life Easier

- Have you ever thought “Man, I have so much information to store, I don’t want to have to create a million variables to store it all!”
- If you have thought this, you’re going to love arrays
- Arrays are an ordered set of elements
- Arrays are used to store and number a list of things
  - We start numbering these items at 0
  - This number is called the index

Creating Arrays

- Concept:
- We store an array in a variable
- Each item in the array is a value and has an index number
- We can then refer to an item in the array using its array name and index number

Method 1 to Create Arrays

- Using [ ]

```
var myArray = [];
```

Creates an empty array

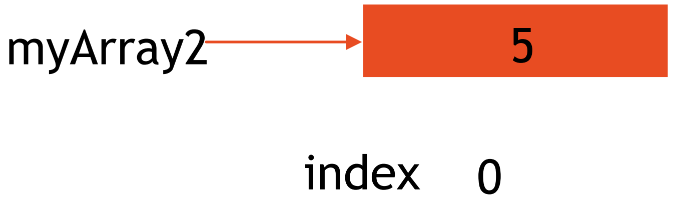
$$\begin{tikzcd} \text{myArray} \arrow[r] & {[\ ]} \end{tikzcd}$$

Method 1 to Create Arrays

- Using [ ]

```
var myArray = [];
var myArray2 = [5];
```

Creates an array of size 1
That contains the number 5



Method 1 to Create Arrays

- Using [ ]

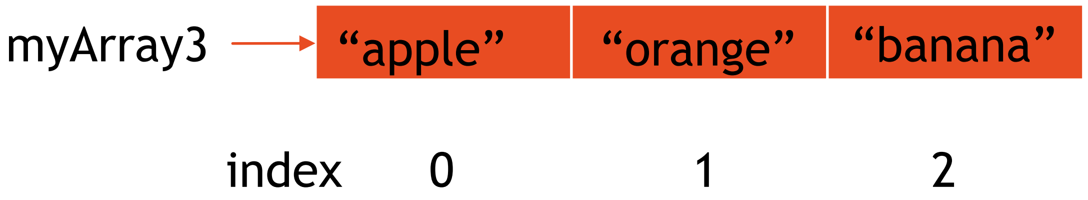
```
var myArray = [];
var myArray2 = [5];
var myArray3 = [“apple”, “orange”, “banana”];
```

Creates an array of size 3, containing 3 strings

Method 1 to Create Arrays

- Using [ ]

```
var myArray = [];
var myArray2 = [5];
var myArray3 = [“apple”, “orange”, “banana”];
```

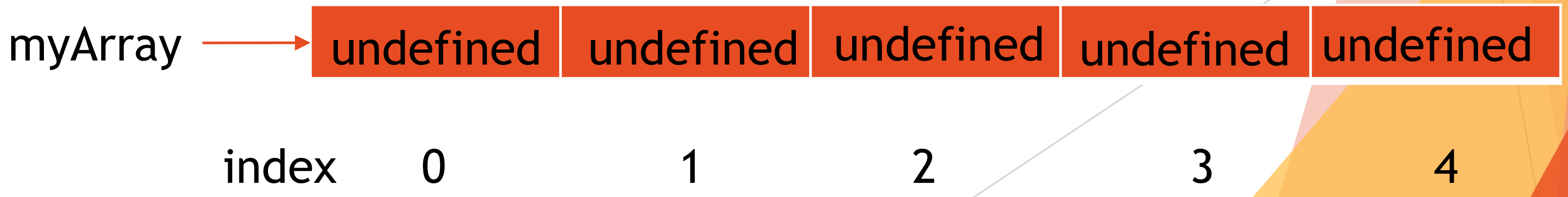


Method 2 to Create Arrays

- Array() function

```
var myArray = new Array(5);
```

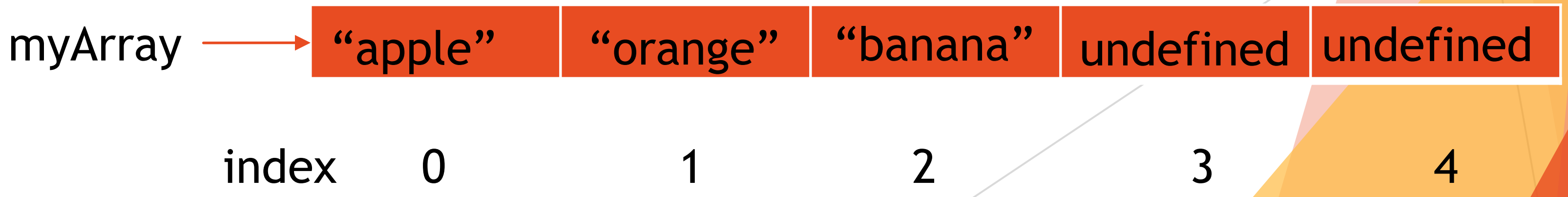
- Creates an array of length 5, with the value "undefined" stored in each space



Method 2 to Create Arrays

- Then we can fill each space the array directly

```
myArray[0] = "apple";
myArray[1] = "orange";
myArray[2] = "banana";
```



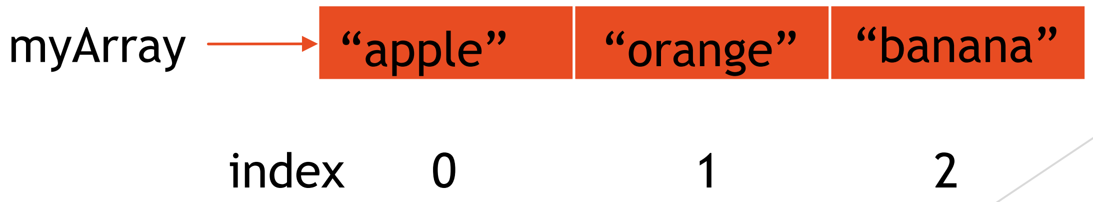
Accessing Array Items

- We can store items in the array and get items from the array
- We need two pieces of information to do this
  - Array name and index the item is stored at

`console.log(myArray[0])` prints “apple”

`console.log(myArray[1])` prints “orange”

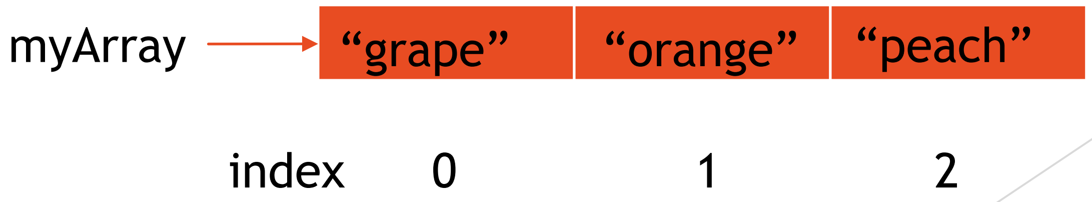
`console.log(myArray[2])` prints “banana”



Modifying Array Items

- We still need the same two pieces of information to do this

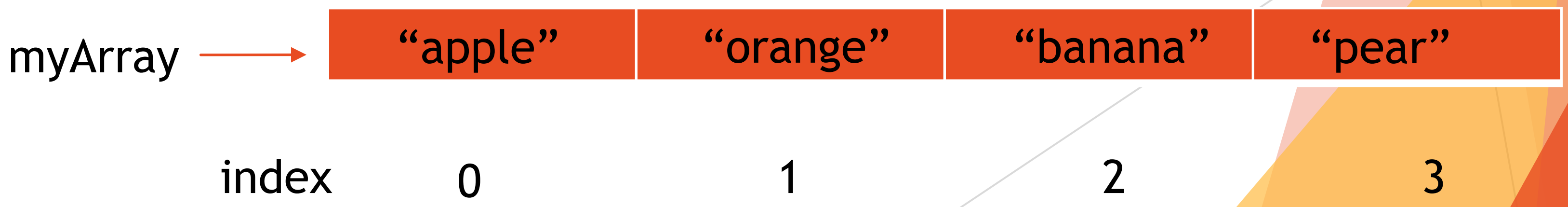
```
myArray[0] = "grape";
myArray[2] = "peach";
```



Adding Items to an Array

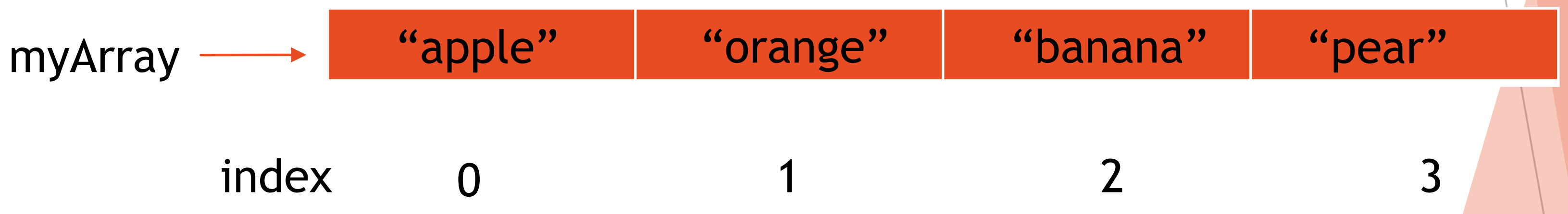
- Arrays in JavaScript can change size (this is not true in other languages)
- So we can add new elements to our arrays
- The function that allows us to do this is `.push(item)`

```
var myArray3 = [“apple”, “orange”,
“banana”];
myArray3.push(“pear”);
```

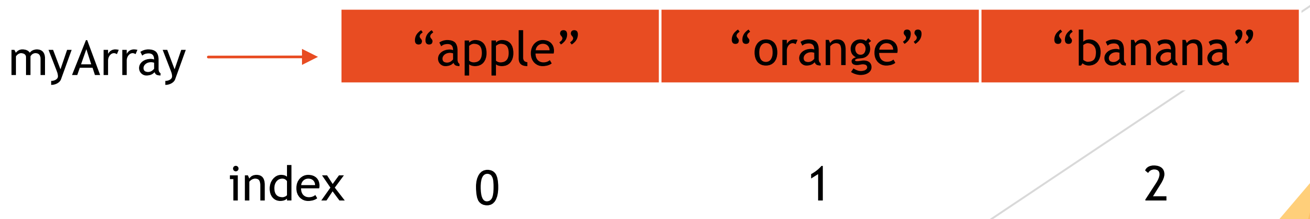


Removing Items From an Array

- We can also remove items from **the end of** an array
- The function that allows us to do this is pop()
  - pop() will remove the LAST item from the array

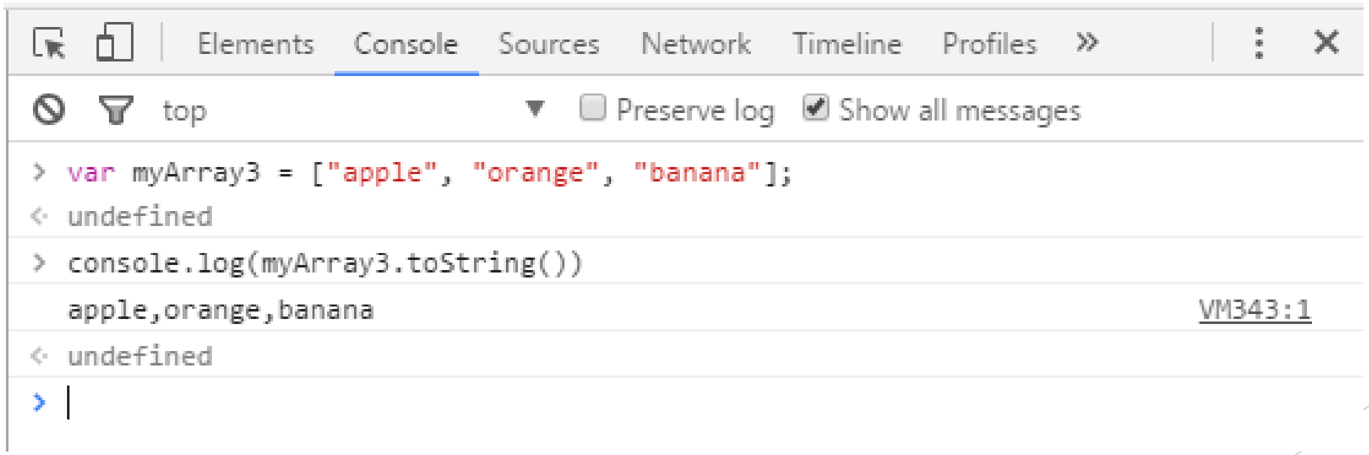


```
var item = myArray.pop();
```



Printing Arrays

- You can print the contents of an entire array to the console using
- `console.log(array.toString());`



```
> var myArray3 = ["apple", "orange", "banana"];
< undefined
> console.log(myArray3.toString())
  apple,orange,banana                                          VM343:1
< undefined
> |
```

Exercise

- Write the JavaScript to create an array to store the names of 25 students (but do not fill the array)
- Write the JavaScript to create an array to store a credit card transaction. This array should store the following information:
    - The name on the card: “Dr. Pepper”
    - The credit card number “1111222233334444”
    - The security number 555
    - The balance of the credit card, which is 3000

Did You Notice?

- Did you notice that the array index starts at 0 and counts the number of items stored in the array?
- Did that make you think “Hey, I could probably use a for loop with arrays!”
- If you thought this, you’re right, and you’re thinking like a computer scientist.

Example

- Let’s create an array of size 5 and fill it with zeroes

```
var zeroArray = new Array(5);
for(var i = 0; i < zeroArray.length; i++){
   zeroArray[i] = 0;
}
```

- What do you think zeroArray.length returns?
- Why did we use < and not <= ?

For Loops and Arrays

- Because we can access each element of an array via an index, it makes sense that we can then process arrays with loops
- Ex. Let’s try to double the value of each element stored in an array

```
var someNums = [5,10,20,30];
for(var j = 0; j < someNums.length; j++){
    someNums[j] = someNums[j] * 2;
}
```

- Can you think of another way we could have written this?

Copying Arrays

- We can use the slice() method to create copies of arrays
- slice() method returns the selected elements in an array, as a new array object
- slice() method selects the elements starting at the given *start* argument, and ends at, *but does not include*, the given *end* argument
- If we don't specify any parameters the whole array is copied

```
var fruits =
["Banana", "Orange", "Lemon", "Apple", "Mango"];
var citrus = fruits.slice(1, 3);
var variety = fruits.slice();
```

Finding Values in Arrays

- The indexOf(someValue) method searches through the array and looks to see if someValue is stored in the array
- If someValue is in the array, the method returns the first index the value is located at
- If someValue is not in the array, the method will return -1

```
var fruits =
["Banana", "Orange", "Lemon", "Apple", “Lemon"];
console.log(fruits.indexOf(“Lemon”));
console.log(fruits.indexOf(“Potato”));
```

Exercise

- What is the index of Big White in the following array?

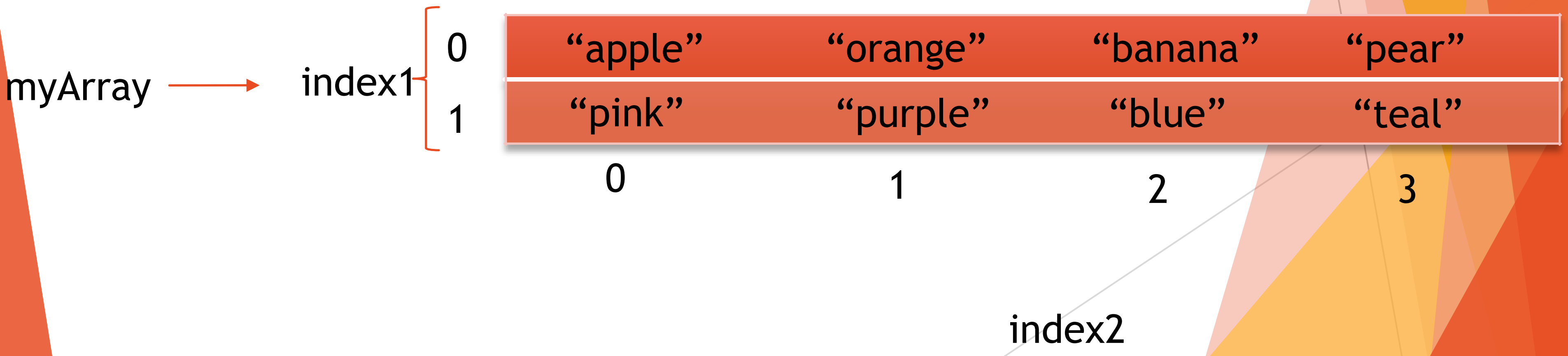
```
var resorts = ["Whistler", “Silverstar",
                "Big White", “Revelstoke",
                "Sun Peaks", "Red Mntn"];
```

- Write an expression that refers to the string Revelstoke within the array.
- What is the value of the expression skiResorts.length?
- What is the index of the last item in the array?
- What is the value of the expression skiResorts[5]?
- Write an expression to find the first index of "Silverstar" within the array

2D Arrays

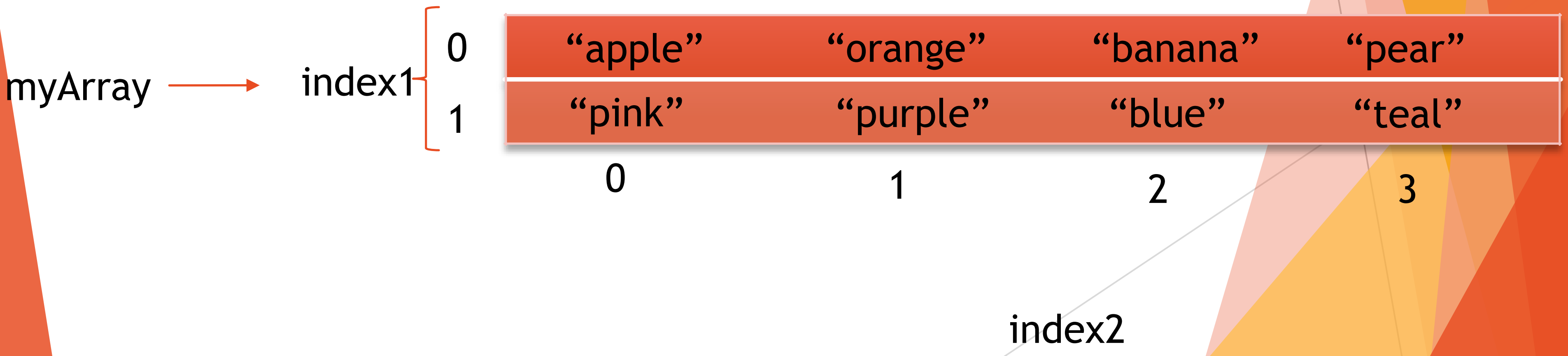
- 2D arrays are just arrays that store arrays
- They are handing for representing grids or tables of information in our programs

```
var myArray = Array(2);
myArray[0] = ["apple", "orange", "banana", "pear"];
myArray[1] = ["pink", "purple", "blue", "teal"];
```



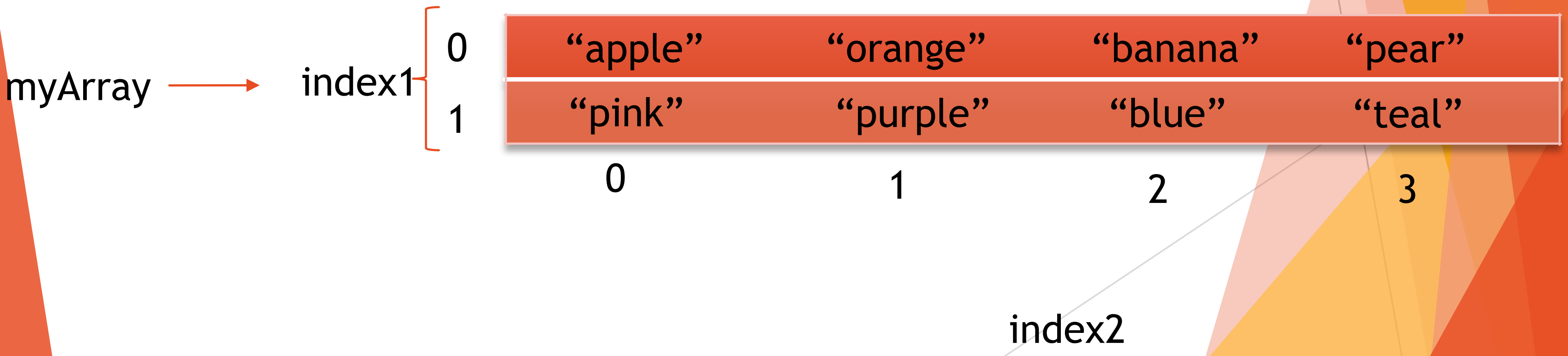
2D Arrays

- `myArray[1] is the array [“pink”, “purple”, “blue”, “teal”]`
- `myArray[0][0] is apple`
- `myArray[1][1] is purple`



Question

- `What is stored at myArray[0][3]?`
- `What is stored at myArray[1][2]?`



Exercise

- Write the JavaScript to create a 2D array. This array should store the following names and midterm exam grades for each of the following students:
  - Jamie 100
  - Amir 75
  - Joel 75
  - Hillary 50
  - Donald 25
- Now, use a for loop to calculate the average of the midterm exam grades stored in the array

Adding Items to Arrays

- splice(position, numberOfItemsToRemove, itemToAdd);
- Parameters: start position, number of elements to delete, elements to add

```
var array = ["one", "two", "four"];
array.splice(2, 0, "three");
array would contain ["one", "two", "three", "four"]
```

- The splice method returns an empty array when no elements are removed; otherwise it returns an array containing the removed element

```
var ar = [1, 2, 3, 4, 5, 6];
ar.splice(3, 0, “a”, “b”, “c”);
console.log( ar );
//prints [1, 2, 3, "a", "b", "c", 4, 5, 6]
```

Putting Arrays Together

- the concat() method will join two or more arrays together
- This method doesn't change the existing array, it returns a new array containing the values of all joined arrays

```
var heros = ["Batman", "Robin"];
var villains = ["Joker",
"Penguin", "Riddler"];
var characters =
heros.concat(villains);
```

Exercise

- Given the following array

```
var arr = ["dog", "cat", "bird"];
What is the value of result:
var result = arr[0] = arr[2];
```

- Write a function called oddArray(N) that accepts the size of an array as input. This function should then return an array filled with the first N odd numbers.