

Arrays¹

Organizing and storing data is a foundational concept of programming.

One way we organize data in real life is by making lists. Let's make one here:

New Year's Resolutions:

1. Keep a journal
2. Take a falconry class
3. Learn to juggle

Let's now write this list in JavaScript, as an *array*:

```
var newYearsResolutions = ["Keep a journal", "Take a falconry class",  
"Learn to juggle"];
```

Arrays are JavaScript's way of making lists. Arrays can store any data types (including strings, numbers, and booleans). Like lists, arrays are ordered, meaning each item has a numbered position.

Here's an array of the concepts we'll cover:

```
let concepts = ["creating arrays", "array structures", "array  
manipulation"]
```

By the end of this assignment, you'll have another tool under your belt that helps you manage chunks of data!

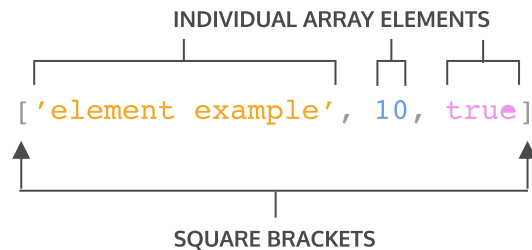
Instructions – Experiment

Open the project at <https://repl.it/@ROYMARQUEZ/csp-arrays-intro-1>. Run the code to see what is logged to the console.

¹ This lesson adapted from material provided by the generous folks at codeAcademy (<https://www.codecademy.com/courses/introduction-to-javascript/lessons/arrays/>)

Create an Array

One way we can create an array is to use an *array literal*. An array literal creates an array by wrapping items in square brackets `[]`. Remember from the previous exercise, arrays can store any data type — we can have an array that holds all the same data types or an array that holds different data types.



Let's take a closer look at the syntax in the array example:

- The array is represented by the square brackets `[]` and the content inside.
- Each content item inside an array is called an *element*.
- There are three different elements inside the array.
- Each element inside the array is a different data type.

We can also save an array to a variable. You may have noticed we did this in the previous experiment:

```
var newYearsResolutions = ["Keep a journal", "Take a falconry class",  
"Learn to juggle"];
```

Let's practice by making an array of our own.

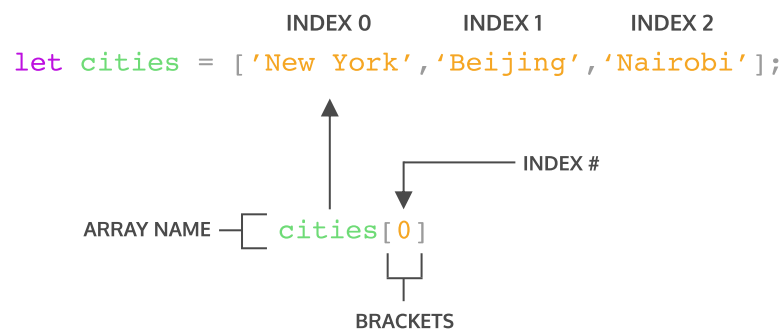
Instructions – Exercise 1

1. Create a new repl.it project. Declare a variable using `var` named `hobbies` and set it equal to an array with three strings inside of it.
2. Use `console.log()` to print `hobbies` to the console.

Accessing Elements

Each element in an array has a numbered position known as its *index*. We can access individual items using their index, which is similar to referencing an item in a list based on the item's position.

Arrays in JavaScript are *zero-indexed*, meaning the positions start counting from `0` rather than `1`. Therefore, the first item in an array will be at position `0`. Let's see how we could access an element in an array:



In the code snippet above:

- `cities` is an array that has three elements.
- We're using bracket notation, `[]` with the index after the name of the array to access the element.
- `cities[0]` will access the element at index `0` in the array `cities`. You can think of `cities[0]` as accessing the space in memory that holds the string `"New York"`.

You can also access individual characters in a string using bracket notation and the index. For instance, you can write:

```
var hello = "Hello World";  
console.log(hello[6]);  
// Output: W
```

The console will display `W` since it is the character that is at index `6`.

Instructions – Exercise 2

1.

Individual elements in arrays can also be stored to variables.

Fork the project at <https://repl.it/@ROYMARQUEZ/csp-arrays-intro-3>.

Create a variable named `listItem` and set it equal to the first item in the `famousSayings` array using square bracket notation (`[]`).

Then use `console.log()` to print the `listItem` variable to the console.

2.

Now, `console.log()` the third element in the `famousSayings` array using bracket notation to access the element.

Do not save the third element to a new variable before you display it.

3.

Awesome, you can access each element in an array using the index. But what happens if you try to access an index that is beyond the last element?

Try to log the item at index `[3]` of `famousSayings` to the console. What is logged to the console?

Update Elements

In the previous section, you learned how to access elements inside an array or a string by using an index. Once you have access to an element in an array, you can update its value.

```
var seasons = ["Winter", "Spring", "Summer", "Fall"];  
  
seasons[3] = "Autumn";  
console.log(seasons);  
//Output: ["Winter", "Spring", "Summer", "Autumn"]
```

In the example above, the `seasons` array contained the names of the four seasons.

However, we decided that we preferred to say `'Autumn'` instead of `'Fall'`.

The line, `seasons[3] = "Autumn";` tells our program to change the item at index 3 of the `seasons` array to be `"Autumn"` instead of what is already there.

Instructions – Exercise 3

1.

Fork the project at <https://repl.it/@ROYMARQUEZ/csp-arrays-intro-4>. Change the second element of the array `groceryList` to `"avocados"`.