Chapter 5

How to create and use arrays

Section 5-10

Creating and Filling Arrays

The syntax for creating an array

Using the new keyword with the Array object name

```
var arrayName = new Array(length);
```

Using the brackets literal

```
var arrayName = [];
```

Two ways to create an empty array

```
var rates = new Array();
var names = [];
```

The syntax for creating an array and assigning values in one statement

Using the new keyword with the Array object name

```
var arrayName = new Array(arrayList);
```

Using the brackets literal

```
var arrayName = [arrayList];
```

How to create an array and assign values

```
var rates = new Array(14.95, 12.95, 11.95, 9.95);
var names = ["Ted Lewis", "Sue Jones", "Ray Thomas"];
```

The syntax for referring to an element of an array

```
arrayName[index]
```

Code that refers to the elements in an array

```
rates[2]     // Refers to the 3rd element in the array
names[1]     // Refers to the 2nd element in the array
```

How to assign rates to an array

```
var rates = new Array(4);
rates[0] = 14.95;
rates[1] = 12.95;
rates[2] = 11.95;
rates[3] = 9.95;
```

How to assign strings to an array

```
var names = [];
names[0] = "Ted Lewis";
names[1] = "Sue Jones";
names[2] = "Ray Thomas";
```

One property and one operator for an array

```
length
delete
```

How to add an element to an array

To the end of an array

```
var numbers = [1, 2, 3, 4];
// array is 1, 2, 3, 4

numbers[numbers.length] = 5;
// array is 1, 2, 3, 4, 5
```

At a specific index

```
var numbers = [1, 2, 3, 4];
// array is 1, 2, 3, 4

numbers[6] = 7;
// array is 1, 2, 3, 4, undefined, undefined, 7
```

How to delete an element at a specific index

```
var numbers = [1, 2, 3, 4];
// array is 1, 2, 3, 4

delete numbers[2];
// array is 1, 2, undefined, 4
```

Exercise 5-1

- Do the exercises for this section

  (shown to the right for the link to this presentation)

Section 5-20

Processing Arrays

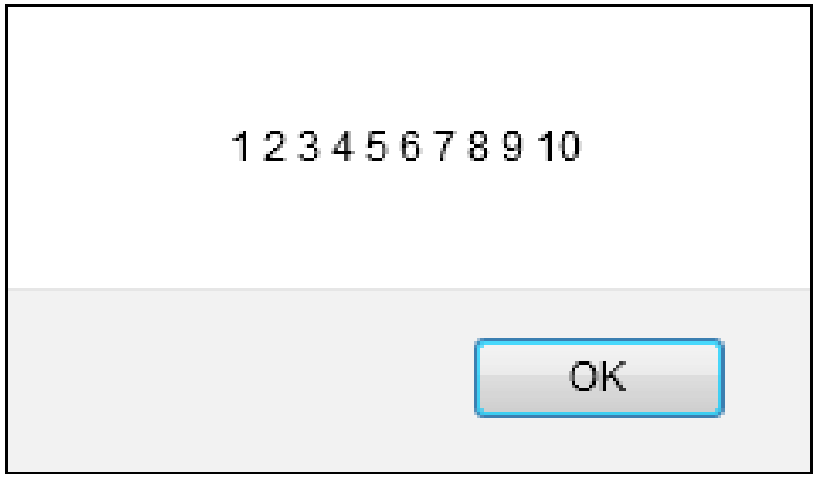
Code that puts 1 through 10 into an array

```
var numbers = [];
for (var i = 0; i < 10; i++) {
    numbers[i] = i + 1;
}
```

Code that displays the numbers array created above

```
var numbersString = "";
for (var i = 0; i < numbers.length; i++) {
    numbersString += numbers[i] + " ";
}
alert (numbersString);
```

The message that's displayed



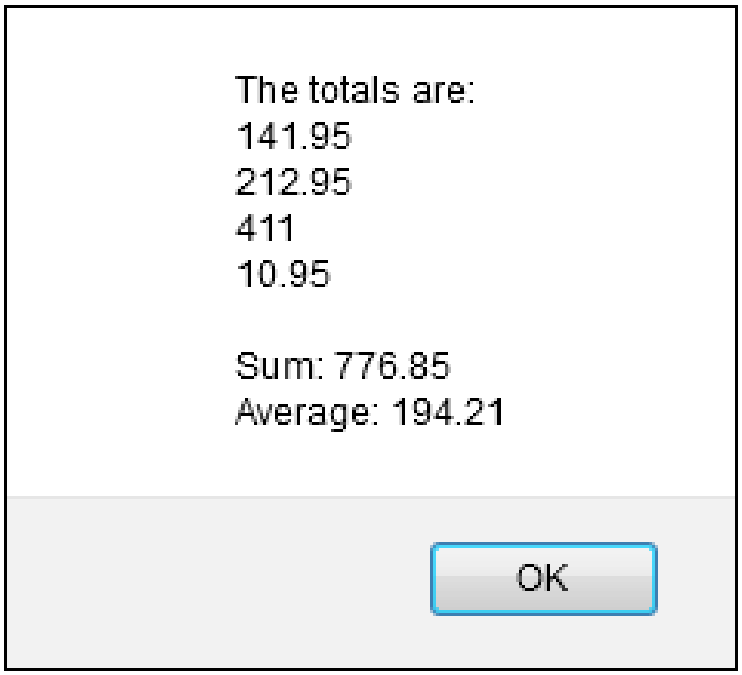
Code that gets the sum and average of an array

```
var totals = [141.95, 212.95, 411, 10.95];
var sum = 0;
for (var i = 0; i < totals.length; i++) {
    sum += totals[i];
}
var average = sum / totals.length;
```

Code that displays the array, sum, and average

```
var totalsString = "";
for (var i = 0; i < totals.length; i++) {
    totalsString += totals[i] + "\n";
}
alert ("The totals are:\n" + totalsString + "\n" +
       "Sum: " + sum.toFixed(2) + "\n" + "Average: " +
        average.toFixed(2) );
```

The message that's displayed



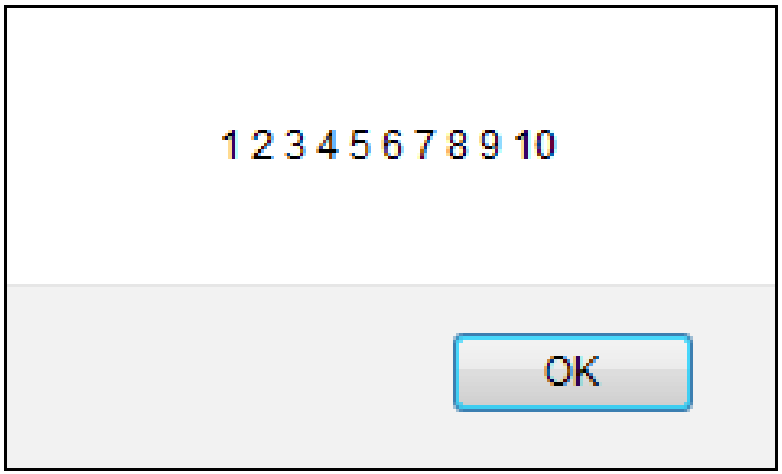
The syntax of a for-in loop

```
for (var elementIndex in arrayName) {
    // statements that access the elements
}
```

A for-in loop that displays an array

```
var numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
var numbersString = "";
for (var index in numbers) {    // The start of the loop
    numbersString += numbers[index] + " ";
}
alert(numbersString);
```

The message that's displayed

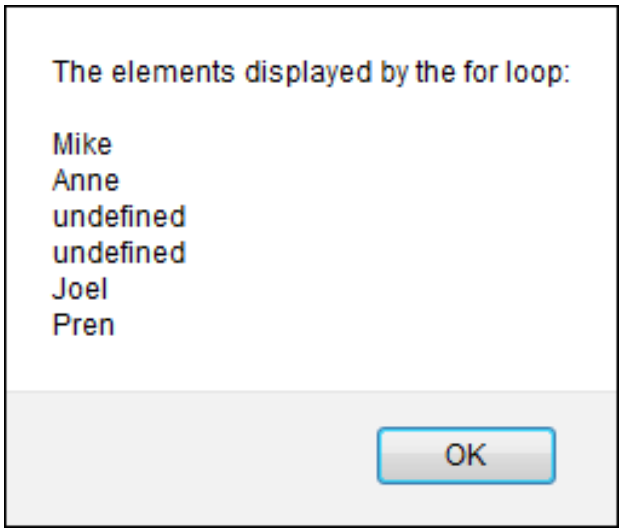


The difference between for and for-in loops

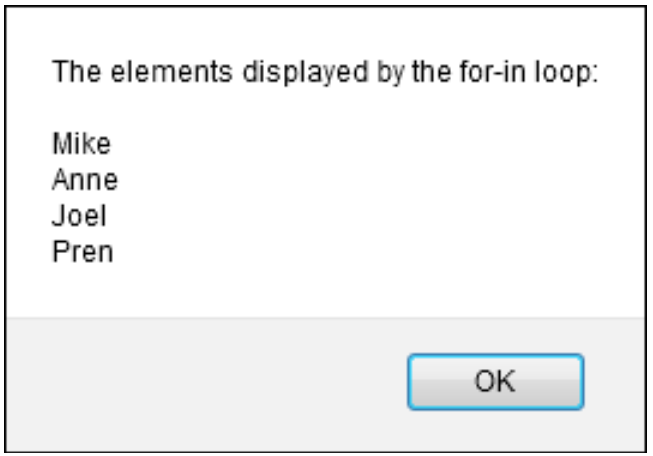
```
var names = ["Mike", "Anne", "Ray"];
names[4] = "Joel";

names[names.length] = "Pren";
delete names[2];

var namesString1 =
    "The elements displayed by the for loop:\n\n";
for (var i = 0; i < names.length; i++) {
    namesString1 += names[i] + "\n"; }
    // Includes undefined elements

var namesString2 =
    "The elements displayed by the for-in loop:\n\n";
for (var i in names) {
    namesString2 += names[i] + "\n"; }
    // Omits undefined elements

alert (namesString1);
alert (namesString2);
```

The messages that are created by the loops

For loop



For-in loop



Exercise 5-2

- Do the exercises for this section

  (shown to the right for the link to this presentation)

Section 5-30

Array Methods

Six of the methods of an Array object

```
push(elements_list)
pop()
unshift(elements_list)
shift()
join(separator)
toString()
```

How to use the push and pop methods

```
var names = ["Mike", "Anne", "Joel"];

names.push("Ray", "Pren");
// names is Mike, Anne, Joel, Ray, Pren

var removedName = names.pop();
// removedName is Pren

alert (names.join());
// displays Mike,Anne,Joel,Ray
```

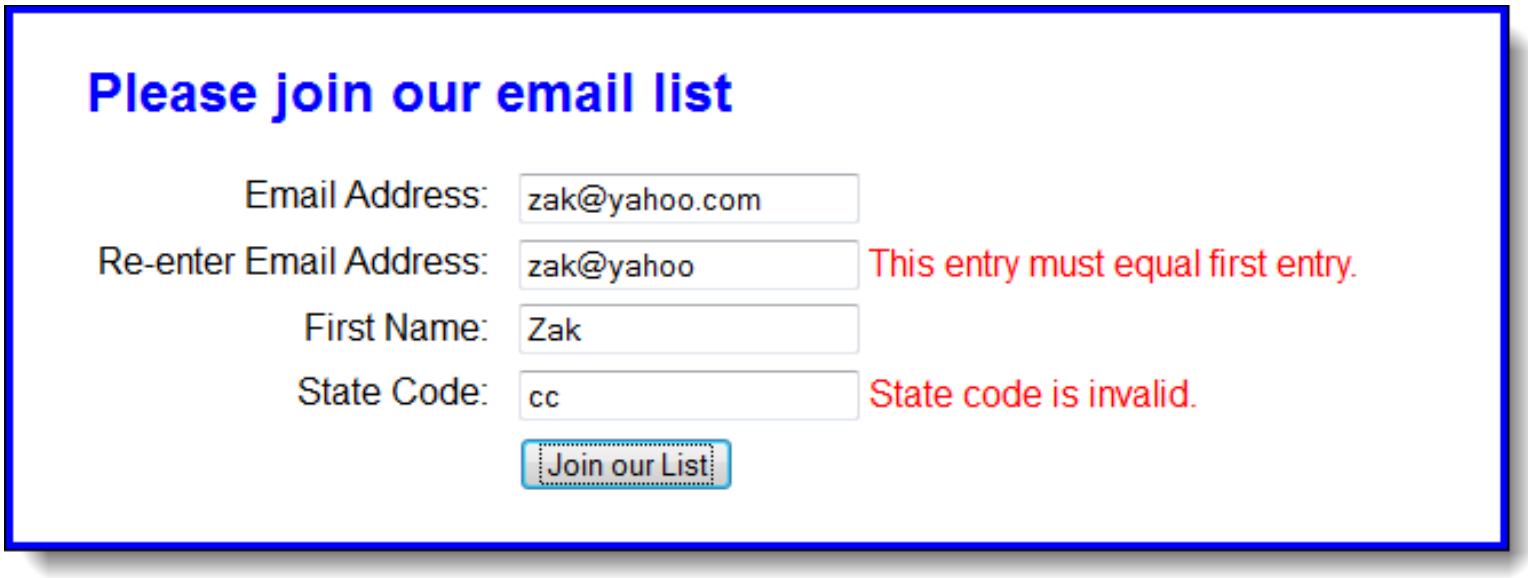
How to use the unshift and shift methods

```
var names = ["Mike", "Anne", "Joel"];

names.unshift("Ray", "Pren");
// names is Ray, Pren, Mike, Anne, Joel

removedName = names.shift();
// removedName is Ray

alert (names.join());
// displays Pren,Mike,Anne,Joel
```

How to use the join and toString methods

```
var names = ["Mike", "Anne", "Joel", "Ray"];

alert (names.join());
// displays Mike,Anne,Joel,Ray

alert (names.join(", "));
// displays Mike, Anne, Joel, Ray

alert (names.toString());
// displays Mike,Anne,Joel,Ray
```

The Email List app with state code validation



The HTML file for the page

```
<section>
    <h1>Please join our email list</h1>
    <form id="email_form" name="email_form"
          action="join.html" method="get">
        <label for="email_address1">Email Address:
        </label>
        <input type="text" id="email_address1
               name="email_address1">
        <span id="email_address1_error">*</span><br>
        <!-- next 2 fields same as in earlier version -->

        <label for="state_code">State Code:</label>
        <input type="text" id="state_code"
               name="state_code">
        <span id="state_code_error">*</span><br>

        <label> </label>
        <input type="button" id="join_list"
               value="Join our List">
    </form>
</section>
```

The JavaScript for the application

```
var $ = function (id) {
    return document.getElementById(id);
}
var stateCodeLookup = function (stateCode) {
    var states =
        ["CA", "WA", "OR", "NV", "NM", "AZ", "WY", "MT"];
    stateCode = stateCode.toUpperCase();
    for (var i = 0; i < states.length; i++) {
        if (states[i] == stateCode) {
            return true;
        }
    }
    return false;
}
```

The JavaScript for the application (continued)

```
var joinList = function () {
    // same as before for validating the first 3 entries

    // validate the state code entry
    var stateCode = $("state_code").value;
    if (!stateCodeLookup(stateCode)) {
        $("state_code_error").firstChild.nodeValue =
            "State code is invalid.";
        isValid = false;
    }
    else {
        $("state_code_error").firstChild.nodeValue = "";
    }

    // submit the form if all entries are valid
    if (isValid) { $("email_form").submit(); }
}
window.onload = function () {
    $("join_list").onclick = joinList;
    $("email_address1").focus();
}
```

Exercise 5-3

- Do the exercises for this section

  (shown to the right for the link to this presentation)

End of Lesson 5