

Introduction to JavaScript Training

Variables, Arrays and Operators

Lesson 1, Activity 2: JavaScript Variables

Variables are used to hold data in memory. JavaScript variables are declared with the `var` keyword.

```
var age;
```

Multiple variables can be declared in a single step.

```
var age, height, weight, gender;
```

After a variable is declared, it can be assigned a value.

```
age = 34;
```

Variable declaration and assignment can be done in a single step.

```
var age = 34;
```

A Loosely-typed Language

JavaScript is a loosely-typed language. This means that you do not specify the data type of a variable when declaring it. It also means that a single variable can hold different data types at different times and that JavaScript can change the variable type on the fly. For example, the `age` variable above is an integer. However, the variable `strAge` below would be a string (text) because of the quotes.

```
var strAge = "34";
```

If you were to try to do a math function on `strAge` (e.g, multiply it by 4), JavaScript would dynamically change it to an integer. Although this is very convenient, it can also cause unexpected results, so be careful.

Variable Naming

1. Variable names must begin with a letter, underscore (`_`), or dollar sign (`$`).
2. Variable names cannot contain spaces or special characters (other than the underscore and dollar sign).
3. Variable names can contain numbers (but not as the first character).
4. Variable names are case sensitive.
5. You cannot use keywords (e.g, `window` or `function`) as a variable name.

Storing User-Entered Data

The following example uses the `prompt()` method of the `window` object to collect user input. The value entered by the user is then assigned to a variable, which is accessed when the user clicks on one of the `span` elements.

Code Sample:

VariablesArraysOperators/Demos/Variables.html

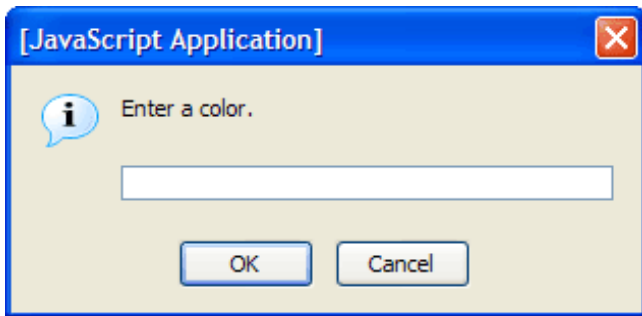
```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Variables</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  //Pop up a prompt
  var userColor = window.prompt("Enter a color.", "");
</script>
</head>
```

```

<body>
<p>
  <span onclick="document.bgColor = 'red';">Red</span> |
  <span onclick="document.bgColor = 'white';">White</span> |
  <span onclick="document.bgColor = 'green';">Green</span> |
  <span onclick="document.bgColor = 'blue';">Blue</span> |
  <span onclick="document.bgColor = userColor;">
  <script type="text/javascript">
    document.write(userColor);
  </script>
</span>
</p>
</body>
</html>

```

As the page loads, a prompt pops up asking the user to enter a color.



This is done with the `prompt()` method of the `window` object. The `prompt()` method is used to get input from the user. It takes two arguments:

1. The message in the dialog box (e.g., "Enter a color.").
2. The default value that appears in the text box. In the example above this is an empty string (e.g., "").

If the OK button is pressed, the prompt returns the value entered in the textbox. If the Cancel button or the close button (the red X) is pressed, the prompt returns `null`. In JavaScript, `null` is datatype with only one value: `null`. It represents a value that we don't know or

that is missing. The line below assigns whatever is returned to the variable `userColor`.

```
var userColor = window.prompt("Enter a color.", "");
```

A `script` block with a call to `document.write()` is then used to output the color entered by the user. This output is contained within a `span` element, which has an `onclick` event handler that will be used to turn the background color of the page to the user-entered color.

```
<span onclick="document.bgColor = userColor;">  
  <script type="text/javascript">  
    document.write(userColor);  
  </script>  
</span>
```

Lesson 1, Activity 4: Using Variables

Duration: 5 to 15 minutes.

In this exercise, you will practice using variables.

1. Open [VariablesArraysOperators/Exercises/Variables.html](#) for editing.
2. Below the `ADD PROMPT HERE` comment, write code that will prompt the user for her first name and assign the result to a variable.
3. Add a button below the Ringo button that reads "Your Name". Add functionality so that when this button is pressed an alert pops up showing the user's first name.
4. Test your solution in a browser.

Code Sample:

[VariablesArraysOperators/Exercises/Variables.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Variables</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  //ADD PROMPT HERE
</script>
</head>
<body>
<form>
  <input type="button" value="Paul" onclick="alert('Paul');">
  <input type="button" value="John" onclick="alert('John');">
  <input type="button" value="George" onclick="alert('George');">
  <input type="button" value="Ringo" onclick="alert('Ringo');">
  <!--ADD BUTTON HERE-->
</form>
</body>
</html>
```

Solution:

VariablesArraysOperators/Solutions/Variables.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Variables</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var firstName = window.prompt("What's your name?", "");
</script>
</head>
<body>
<form>
  <input type="button" value="Paul" onclick="alert('Paul');">
  <input type="button" value="John" onclick="alert('John');">
  <input type="button" value="George" onclick="alert('George');">
  <input type="button" value="Ringo" onclick="alert('Ringo');">
  <input type="button" value="Your Name" onclick="alert(firstName);">
</form>
</body>
</html>
```

Lesson 1, Activity 5: Arrays

An array is a grouping of objects that can be accessed through subscripts. At its simplest, an array can be thought of as a list. In JavaScript, the first element of an array is considered to be at position zero (0), the second element at position one (1), and so on. Arrays are useful for storing related sets of data.

Arrays are declared using the `new` keyword.

```
var myarray = new Array();
```

It is also possible and very common to use the `[]` literal to declare a new Array object.

```
var myarray = [];
```

Values are assigned to arrays as follows:

```
myarray[0] = value1;  
myarray[1] = value2;  
myarray[2] = value3;
```

Arrays can be declared with initial values.

```
var myarray = new Array(value1, value2, value3);  
//or, using the [] notation:  
var myarray = [value1, value2, value3];
```

The following example is similar to the previous one, except that it prompts the user for four different colors and places each into the

userColors array. It then displays the values in the userColors array in the spans and assigns them to document.bgColor when the user clicks on the spans.

Unlike in some languages, values in JavaScript arrays do not all have to be of the same data type.

Code Sample:

VariablesArraysOperators/Demos/Arrays.html

```

---- C O D E   O M I T T E D ----

<script type="text/javascript">
  //Pop up four prompts and create an array
  var userColors = new Array();
  userColors[0] = window.prompt("Choose a color.", "");
  userColors[1] = window.prompt("Choose a color.", "");
  userColors[2] = window.prompt("Choose a color.", "");
  userColors[3] = window.prompt("Choose a color.", "");
</script>
</head>
<body>
<p>
  <span onclick="document.bgColor = userColors[0];">
    <script type="text/javascript">
      document.write(userColors[0]);
    </script>
  </span> |
  <span onclick="document.bgColor = userColors[1];">
    <script type="text/javascript">
      document.write(userColors[1]);
    </script>
  </span> |
  <span onclick="document.bgColor = userColors[2];">
    <script type="text/javascript">
      document.write(userColors[2]);
    </script>
  </span> |
  <span onclick="document.bgColor = userColors[3];">
    <script type="text/javascript">
      document.write(userColors[3]);
    </script>
  </span>
</p>

```

```
</body>
</html>
```

As the page loads, an array called `userColors` is declared.

```
var userColor = new Array();
```

The next four lines populate the array with user-entered values.

```
userColors[0] = window.prompt("Choose a color.", "");
userColors[1] = window.prompt("Choose a color.", "");
userColors[2] = window.prompt("Choose a color.", "");
userColors[3] = window.prompt("Choose a color.", "");
```

The body of the page contains a paragraph with four `span` tags, the text of which is dynamically created with values from the `userColors` array.

Associative Arrays

Whereas regular (or enumerated) arrays are indexed numerically, associative arrays are indexed using names as keys. The advantage of this is that the keys can be meaningful, which can make it easier to reference an element in an array. The example below illustrates how an associative array is used.

Code Sample:

<VariablesArraysOperators/Demos/AssociativeArrays.html>

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
```

```

<title>Associative Arrays</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var beatles = [];
  beatles["singer1"] = "Paul";
  beatles["singer2"] = "John";
  beatles["guitarist"] = "George";
  beatles["drummer"] = "Ringo";
</script>
</head>
<body>
<p>
  <script type="text/javascript">
    document.write(beatles["singer1"]);
    document.write(beatles["singer2"]);
    document.write(beatles["guitarist"]);
    document.write(beatles["drummer"]);
  </script>
</p>
</body>
</html>

```

Array Properties and Methods

The tables below show some of the most useful array properties and methods. All of the examples assume an array called `beatles` that holds "Paul", "John", "George", and "Ringo".

```
var beatles = ["Paul", "John", "George", "Ringo"];
```

Array Properties

Property	Description
length	Holds the number of elements in an array. beatles.length // 4

Array Methods

Property	Description
join(delimiter)	Returns a delimited list of the items indexed with integers in the array. The default delimiter is a comma.

	<code>beatles.join(":") // Paul:John:George:Ringo</code>
<code>push()</code>	<p>Appends an element to an array.</p> <code>beatles.push("Steve")</code>
<code>pop()</code>	<p>Removes the last item in an array and returns its value.</p> <code>beatles.pop() // Returns Ringo</code>
<code>shift()</code>	<p>Removes the first item in an array and returns its value.</p> <code>beatles.shift() // Returns Paul</code>
<code>slice(start, end)</code>	<p>Returns a subarray from start to end. If end is left out, it includes the remainder of the array.</p> <code>beatles.slice(1, 2) //Returns [John, George]</code>
<code>splice(start, count)</code>	<p>Removes count items from start in the array and returns the resulting array.</p> <code>beatles.splice(1, 2) //Returns [Paul, Ringo]</code>
<code>sort()</code>	<p>Sorts an array alphabetically.</p> <code>beatles.sort() //Returns [George, John, Paul, Ringo]</code>

Lesson 1, Activity 7: Working with Arrays

Duration: 15 to 25 minutes.

In this exercise, you will practice working with arrays.

1. Open [VariablesArraysOperators/Exercises/Arrays.html](#) for editing.
2. Below the comment, declare a `rockStars` array and populate it with four values entered by the user.
3. Add functionality to the buttons, so that alerts pop up with values from the array when the buttons are clicked.
4. Test your solution in a browser.

Code Sample:

VariablesArraysOperators/Exercises/Arrays.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Arrays</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  /*
    Declare a rockStars array and populate it with
    four values entered by the user.
  */
</script>
</head>
<body>
<form>
  <input type="button" value="Favorite">
  <input type="button" value="Next Favorite">
  <input type="button" value="Next Favorite">
  <input type="button" value="Next Favorite">
</form>
</body>
</html>
```

Solution:

VariablesArraysOperators/Solutions/Arrays.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Arrays</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var rockStars = new Array();
  rockStars[0] = window.prompt("Who is your favorite rock star?", "");
  rockStars[1] = window.prompt("And your next favorite rock star?", "");
  rockStars[2] = window.prompt("And your next favorite rock star?", "");
  rockStars[3] = window.prompt("And your next favorite rock star?", "");
</script>
</head>
<body>
<form>
  <input type="button" value="Favorite" onclick="alert(rockStars[0]);">
  <input type="button" value="Next Favorite" onclick="alert(rockStars[1]);">
  <input type="button" value="Next Favorite" onclick="alert(rockStars[2]);">
  <input type="button" value="Next Favorite" onclick="alert(rockStars[3]);">
</form>
</body>
</html>
```

Lesson 1, Activity 9: JavaScript Operators

Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (remainder)
++	Increment by one
--	Decrement by one

Assignment Operators

Operator	Description
=	Assignment
+=	One step addition and assignment (a+=3 is the same as a=a+3)
-=	One step subtraction and assignment (a-=3 is the same as a=a-3)
=	One step multiplication and assignment (a=3 is the same as a=a*3)
/=	One step division and assignment (a/=3 is the same as a=a/3)
%=	One step modulus and assignment (a%=3 is the same as a=a%3)

String Operators

Operator	Description
+	Concatenation (var greeting = "Hello " + firstname;)
+=	One step concatenation and assignment (var greeting = "Hello "; greeting += firstname;)

The following code sample shows these operators in use:

Code Sample:

[VariablesArraysOperators/Demos/Operators.html](#)

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Operators</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var userNum1 = window.prompt("Choose a number.", "");
  alert("You chose " + userNum1);
  var userNum2 = window.prompt("Choose another number.", "");
  alert("You chose " + userNum2);
  var numsAdded = userNum1 + Number(userNum2);
  var numsSubtracted = userNum1 - userNum2;
  var numsMultiplied = userNum1 * userNum2;
  var numsDivided = userNum1 / userNum2;
  var numsModulused = userNum1 % userNum2;
</script>
</head>
<body>
<p>
  <script type="text/javascript">
    document.write(userNum1 + " + " + userNum2 + " = ");
    document.write(numsAdded + "<br>");
    document.write(userNum1 + " - " + userNum2 + " = ");
    document.write(numsSubtracted + "<br>");
    document.write(userNum1 + " * " + userNum2 + " = ");
    document.write(numsMultiplied + "<br>");
    document.write(userNum1 + " / " + userNum2 + " = ");
    document.write(numsDivided + "<br>");
    document.write(userNum1 + " % " + userNum2 + " = ");
    document.write(numsModulused + "<br>");
  </script>
</p>
</body>
</html>

```

The file above illustrates the use of the concatenation operator and several math operators. It also illustrates a potential problem with loosely-typed languages. This page is processed as follows:

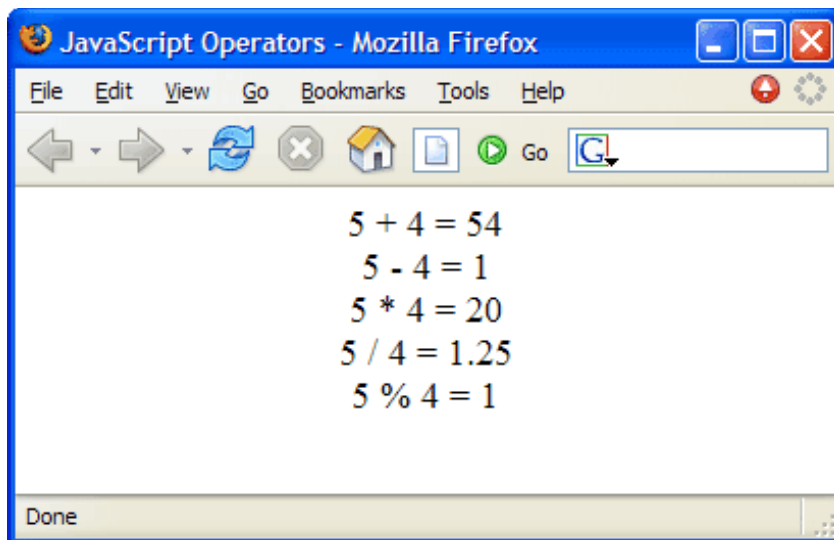
1. The user is prompted for a number and the result is assigned to `userNum1`.
2. An alert pops up telling the user what number she entered. The concatenation operator (+) is used to combine two

strings: "You chose " and the number entered by the user. Note that all user-entered data is always treated as a string of text, even if the text consists of only digits.

3. The user is prompted for another number and the result is assigned to `userNum2`.
4. Another alert pops up telling the user what number she entered.
5. Five variables are declared and assigned values.

```
var numsAdded = userNum1 + userNum2;  
var numsSubtracted = userNum1 - userNum2;  
var numsMultiplied = userNum1 * userNum2;  
var numsDivided = userNum1 / userNum2;  
var numsModulus = userNum1 % userNum2;
```

6. The values the variables contain are output to the browser.



So, $5 + 4$ is 54! It is when 5 and 4 are strings, and, as stated earlier, all user-entered data is treated as a string. In the upcoming lesson on JavaScript Functions, you will learn how to convert a string to a number.

Ternary Operator

Operator	Description
?:	Conditional evaluation (var evenOrOdd = (number % 2 == 0) ? "even" : "odd";)

The code sample below shows how the ternary operator works:

Code Sample:

VariablesArraysOperators/Demos/Ternary.html

```

---- C O D E   O M I T T E D ----

<script type="text/javascript">
  var num = prompt("Enter a number.", "");

  //without ternary:
  if (num % 2 == 0) {
    alert(num + " is even.");
  } else {
    alert(num + " is odd.");
  }

  //with ternary:
  var term = num % 2 == 0 ? "even" : "odd";
  alert(num + " is " + term);
</script>
---- C O D E   O M I T T E D ----

```

Lines 7-11 show a regular if-else statement, which we will cover in detail in the Conditionals and Loops lesson.

Lines 14-15 shows how to accomplish the same thing in a couple lines of code with the ternary operator.

Default Operator

Operator	Description
	Used to assign a default operator (var yourName = prompt("Your Name?", "") "Stranger";)

The code sample below shows how the default operator works:

Code Sample:

VariablesArraysOperators/Demos/Default.html

```
---- C O D E   O M I T T E D ----  
  
<script type="text/javascript">  
  var yourName = prompt("Your Name?", "") || "Stranger";  
  
  alert("Hi " + yourName + "!");  
</script>  
---- C O D E   O M I T T E D ----
```

If the user presses **OK** without filling out the prompt or presses **Cancel**, the default value "Stranger" is assigned to `yourName`.

Lesson 1, Activity 10: **Working with Operators**

Duration: 15 to 25 minutes.

In this exercise, you will practice working with JavaScript operators.

1. Open [VariablesArraysOperators/Exercises/Operators.html](#) for editing.
2. Add code to prompt the user for the number of albums she owns of her favorite and second favorite rockstars.
3. In the body, let the user know how many more of her favorite rockstar's albums she has than of her second favorite rockstar's albums.
4. Test your solution in a browser.

Code Sample:

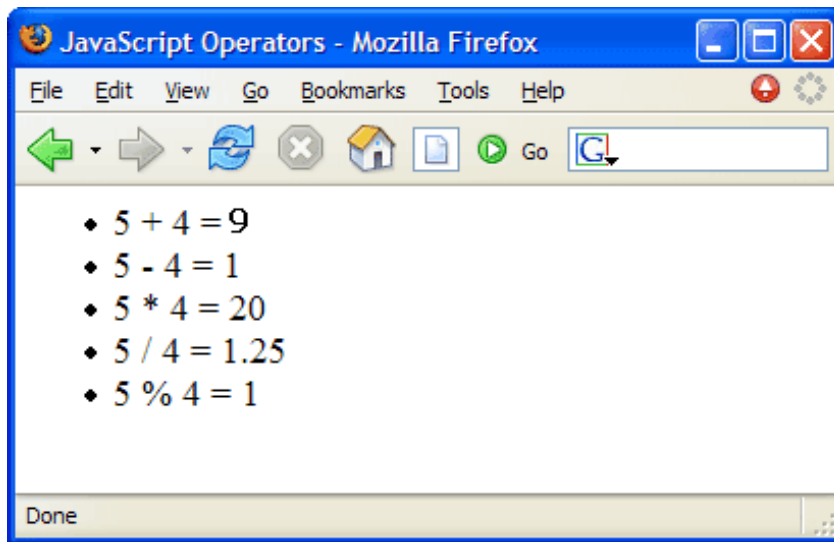
[VariablesArraysOperators/Exercises/Operators.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Operators</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var rockStars = [];
  rockStars[0] = prompt("Who is your favorite rock star?", "");
  /*
  Ask the user how many of this rockstar's albums she owns and store
  the result in a variable.
  */
  rockStars[1] = prompt("And your next favorite rock star?", "");
  /*
  Ask the user how many of this rockstar's albums she owns and store
  the result in a variable.
  */
</script>
</head>
<body>
<!--
  Let the user know how many more of her favorite rockstar's albums
  she has than of her second favorite rockstar's albums.
-->
```

```
</body>
</html>
```

Challenge

1. Open <VariablesArraysOperators/Exercises/Operators-challenge.html> for editing.
2. Modify it so that it outputs an unordered list as shown below:



Solution:

<VariablesArraysOperators/Solutions/Operators.html>

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Operators</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var rockStars = [];
  var cdTotals = [];
  rockStars[0] = prompt("Who is your favorite rock star?", "");
  cdTotals[0] = prompt("How many " + rockStars[0] + " albums do you own?", "");
  rockStars[1] = prompt("And your next favorite rock star?", "");
  cdTotals[1] = prompt("How many " + rockStars[1] + " albums do you own?", "");
</script>
```

```

</head>
<body>
  <script type="text/javascript">
    var diff = cdTotals[0] - cdTotals[1];
    document.write("You have " + diff + " more albums of " + rockStars[0]);
    document.write(" than you have of " + rockStars[1] + ".");
  </script>
</body>
</html>

```

Challenge Solution:

VariablesArraysOperators/Solutions/Operators-challenge.html

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Operators</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  var userNum1 = window.prompt("Choose a number.", "");
  alert("You chose " + userNum1);
  var userNum2 = window.prompt("Choose another number.", "");
  alert("You chose " + userNum2);
  var numsAdded = Number(userNum1) + Number(userNum2);
  var numsSubtracted = userNum1 - userNum2;
  var numsMultiplied = userNum1 * userNum2;
  var numsDivided = userNum1 / userNum2;
  var numsModulused = userNum1 % userNum2;
</script>
</head>
<body>
<ul>
  <script type="text/javascript">
    document.write("<li>" + userNum1 + " + " + userNum2 + " = ");
    document.write(numsAdded + "</li>");
    document.write("<li>" + userNum1 + " - " + userNum2 + " = ");
    document.write(numsSubtracted + "</li>");
    document.write("<li>" + userNum1 + " * " + userNum2 + " = ");
    document.write(numsMultiplied + "</li>");
    document.write("<li>" + userNum1 + " / " + userNum2 + " = ");
    document.write(numsDivided + "</li>");
    document.write("<li>" + userNum1 + " % " + userNum2 + " = ");
    document.write(numsModulused + "</li>");
  </script>
</ul>
</body>
</html>

```

