Chapter 2

Getting started with JavaScript

(Part 1)

Section 2-10

How to Include JavaScript in an HTML Document

Two attributes of the script element

```
src
type    (no longer needed with HTML5)
```

A script element in the head section that loads an external JavaScript file

```
<script src="calculate_mpg.js"></script>
```

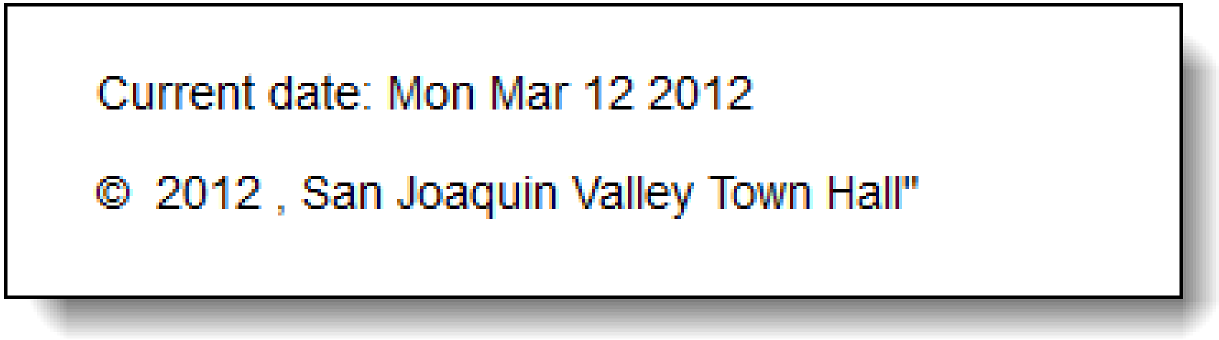
A script element that embeds JavaScript in the head section

```
<head>
    ...
    <script>
        alert("The Calculate MPG application");
        var miles = prompt("Enter miles driven");
        miles = parseFloat(miles);
        var gallons =
            prompt("Enter gallons of gas used");
        gallons = parseFloat(gallons);
        var mpg = miles/gallons;
        mpg = parseInt(mpg);
        alert("Miles per gallon = " + mpg);
    </script>
</head>
```

JavaScript in the body of an HTML document

```
<p>
    <script>
        var today = new Date();
        document.write("Current date: ");
        document.write(today.toDateString());
    </script>
</p>
<p>&copy;&nbsp;
    <script>
        var today = new Date();
        document.write( today.getFullYear() );
    </script>
, San Joaquin Valley Town Hall
</p>
```

The result of the JavaScript in a web browser



Current date: Mon Mar 12 2012

© 2012 , San Joaquin Valley Town Hall"

A noscript element in the body of the HTML

```
<p>&copy;&nbsp;
    <script>
        var today = new Date();
        document.write( today.getFullYear() );
    </script>
    <noscript>2012</noscript>
, San Joaquin Valley Town Hall
</p>
```

A noscript element at the start of the HTML

```
<h2>
    <noscript>
        To get the most from this web site,
        please enable JavaScript.
    </noscript>
</h2>
```

Exercise 2-1

- Click on the link for Exercise_2-1

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

Section 2-20

JavaScript Syntax

A block of JavaScript code

```
var joinList = function () {
    var emailAddress1 = $("email_address1").value;
    var emailAddress2 = $("email_address2").value;

    if (emailAddress1 == "") {
        alert("Email Address is required.");
    } else if (emailAddress2 == "") {
        alert("Second Email Address is required.");
    } else if (emailAddress1 !== emailAddress2) {
        alert("Second Email entry must equal " +
              "first entry.");
    } else if ($("first_name").value == "") {
        alert("First Name is required.");
    } else {
        $("email_form").submit();
    }
}
```

The basic syntax rules for JavaScript

- JavaScript is case-sensitive.
- JavaScript ignores extra whitespace within statements.
- Each JavaScript statement ends with a semicolon.

How to split a statement over two or more lines

- Split a statement after:

  an arithmetic or relational operator like +, -, *, /, =, ==, >, or <

  an opening brace ({), bracket ([ ), or parenthesis

  a closing brace (})

- Do not split a statement after:

  an identifier, a value, or the *return* keyword

  a closing bracket ( ]) or parenthesis

Rules for creating identifiers

- Identifiers can only contain letters, numbers, the underscore, and the dollar sign.
- Identifiers can't start with a number.
- Identifiers are case-sensitive.
- Identifiers can be any length.
- Identifiers can't be the same as *reserved words*.
- Avoid using global properties and methods as identifiers.

Valid identifiers in JavaScript

```
subtotal
index_1
$
taxRate
calculate_click
$log
```

Camel casing versus underscore notation

taxRate	tax_rate
calculateClick	calculate_click
emailAddress	email_address
firstName	first_name
futureValue	future_value

Naming recommendations

- Use meaningful names for identifiers.

  That way, your identifiers aren't likely to be reserved words or global properties.
- Be consistent: Use either camel casing or underscores.
- If you're using underscore notation, use lowercase for all letters.

JavaScript code that includes comments

```
/* this onload function sets up the events that display
   and hide the text that follows a series of h2 headings
*/
window.onload = function () {
    var fiveReasons = $("five_reasons"); // gets a div

    // gets the h2 and div elements within the div element
    var h2Headings =
        fiveReasons.getElementsByTagName("h2");
    var divTags = fiveReasons.getElementsByTagName("div");

    var i, headingNode, divNode;
    for (i = 0; i < h2Headings.length; i++ ) {
        // one loop for each h2
        headingNode = h2Headings[i];
        divNode = divTags[i];

        // Attaches an event handler for each h2
        headingNode.onclick = function () {
            var h2 = this;
```

The syntax rules for JavaScript comments

- Block comments begin with /* and end with */.
- Single-line comments begin with two forward slashes and continue to the end of the line.

Guidelines for using comments

- Use comments to describe portions of code that are hard to understand.
- Use comments to comment out portions of code that you don’t want to test.
- Don’t use comments unnecessarily.

Exercise 2-2

- Click on the link for Exercise_2-2

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

Section 2-30

Introduction to Objects in JavaScript

Common methods of the window object

The syntax for calling a method of an object

objectName.*methodName*(*parameters*)

```
window.alert(string)
        or just alert(string)

window.prompt(string, default)
        or just prompt(string, default)

window.print()
        or just print()
```

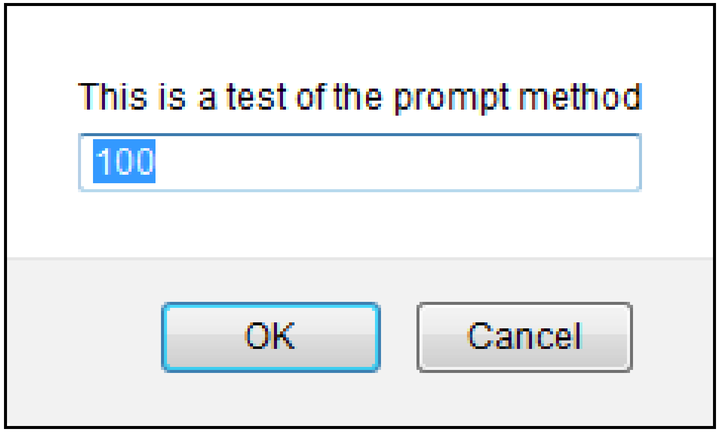
A statement that calls the alert method of the window object

```
window.alert("This is a test of the alert method");
```

A statement that calls the prompt method with the object name omitted

```
var userEntry =
    prompt("This is a test of the prompt method", 100);
```

The prompt dialog box that's displayed



This is a test of the prompt method

100

One property of the window object

```
location
```

The syntax for accessing a property of an object

```
objectName.propertyName
```

A method that displays the URL of the window object

```
alert(window.location);
```

Exercise 2-3

- Click on the link for Exercise_2-3

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

Section 2-40

JavaScript Data

Number, String, and Boolean values

Examples of number values

```
15
-21
21.5
-124.82
-3.7e-9
```

Examples of string values

```
"JavaScript"
'String Data'
""
```

The two Boolean values

```
true
false
```

Common arithmetic operators

Operator	Example	Result
`+`	`5 + 7`	`12`
`-`	`5 - 12`	`-7`
`*`	`6 * 7`	`42`
`/`	`13 / 4`	`3.25`
`%`	`13 % 4`	`1`
`++`	`counter++`	adds 1 to counter
`--`	`counter--`	subtracts 1 from counter

The order of precedence for arithmetic expressions

Order	Operators	Direction
1	`++`	Left to right
2	`--`	Left to right
3	`* / %`	Left to right
4	`+ -`	Left to right

Examples of precedence and parentheses

```
3 + 4 * 5              // Result is 23
(3 + 4) * 5            // Result is 35

13 % 4 + 9             // Result is 10
13 % (4 + 9)           // Result is 0

1000 + 1000 * .05     // Result is 1050
1000 + (1000 * .05)   // Result is still 1050
```

The most useful assignment operators

```
=
+=
```

How to declare numeric variables

```
var subtotal;
var investment, interestRate, years;
```

How to declare and assign values to variables

```
var subtotal = 74.00;              // subtotal = 74.00
var salesTax = subtotal * .1;      // salesTax = 7.4
```

How to code compound assignment statements

```
var subtotal = 74.95;                    // subtotal = 74.95
subtotal += 20.00;                       // subtotal = 94.95
```

Three ways to increment a counter variable

```
var counter = 1;                // counter = 1
counter = counter + 1;          // counter now = 2
counter += 1;                   // counter now = 3
counter++;                      // counter now = 4
```

The concatenation operator for strings

Operator	Example	Result
`+`	`"Ray " + "Harris"`	`"Ray Harris"`
	`"Months: " + 120`	`"Months: 120"`

Escape sequences that can be used in strings

Operator	Description
`\n`	Starts a new line in a string.
`\"`	Puts a double quotation mark in a string.
`\'`	Puts a single quotation mark in a string.

How to declare string variables

```
var zipCode;
var lastName, state, zipCode;
```

How to declare and assign values to variables

```
var firstName = "Ray", lastName = "Harris";
var fullName = lastName + ", " + firstName;
// fullName is "Harris, Ray"
```

How to code compound assignment statements

With string data

```
var firstName = "Ray", lastName = "Harris";
var fullName = lastName;
fullName += ", ";
fullName += firstName;
// fullName is "Harris, Ray"
```

With mixed data

```
var months = 120;
message = "Months: ";
message += months;
// message is "Months: 120"
```

How escape sequences can be used in a string

```
var message =
    "A valid variable name\ncannot start with a number.";
var message = "This isn\'t the right way to do this.";
```

How to declare Boolean variables and assign values to them

```
var isValid = false;
```

How JavaScript interprets the plus sign in mixed expressions

- If both values are numbers, JavaScript adds them.
- If both values are strings, JavaScript concatenates them.
- If one value is a number and one is a string, JavaScript converts the number to a string and concatenates them.

Two methods of the window object for converting string values

```
parseInt(string)
parseFloat(string)
```

Examples that use these methods

```
var entryA = prompt("Enter any value", 12345.6789);
alert(entryA);           // displays 12345.6789
entryA = parseInt(entryA);
alert(entryA);           // displays 12345

var entryB = prompt("Enter any value", 12345.6789);
alert(entryB);           // displays 12345.6789
entryB = parseFloat(entryB);
alert(entryB);           // displays 12345.6789

var entryC = prompt("Enter any value", "Hello");
alert(entryC);           // displays Hello
entryC = parseInt(entryC);
alert(entryC);           // displays NaN
```

Exercise 2-4

- Click on the link for Exercise_2-4

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

End of Chapter 2 – Part 1