

Chapter 2

How to code a JavaScript application

The basic syntax rules for JavaScript

- JavaScript is case-sensitive.
- JavaScript statements end with a semicolon.
- JavaScript ignores extra whitespace in statements.
- Single-line comments begin with two forward slashes.
- Multi-line comments begin with `/*` and end with `*/`.

How to load JavaScript from an external file

```
<script type="text/javascript" src="sales_tax.js"></script>
```

A single-line comment

```
nextYear = thisYear + 1; // Add 1 to this year
```

A multi-line comment

```
/* The following line determines what the  
next year is by adding 1 to the current year  
*/  
nextYear = thisYear + 1;
```

The primitive data types

- Number
- String
- Boolean

Terms

- integer
- floating-point value
- exponent
- empty string
- escape sequence
- Boolean value

Rules for creating identifiers in JavaScript

- 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.
- Avoid using names that are similar to reserved words, global properties, or global methods.

Valid identifiers in JavaScript

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

Examples of number values

15 // an integer

-21 // a negative integer

21.5 // a floating-point value

-124.82 // a negative floating-point value

-3.7e-9 // a floating-point value
// equivalent to -0.0000000037

Examples of string values

```
"JavaScript"    // a string with double quotes  
'String Data'  // a string with single quotes  
""             // an empty string
```

How the `\n` escape sequence can be used to start a new line in a string

```
"A valid variable name\ncannot start with a number."  
// represents the string: A valid variable name  
// cannot start with a number.
```

The two Boolean values

```
true    // equivalent to true, yes, or on  
false   // equivalent to false, no, or off
```

Common arithmetic operators

+

-

*

/

%

++

--

The order of precedence for arithmetic expressions

Order	Operators
1	++
2	--
3	* / %
4	+ -

Precedence and the use of 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`

Terms

- arithmetic expression
- arithmetic operator
- order of precedence
- concatenate
- concatenation operator

How to declare variables without assigning values to them

```
var subtotal;  
// declares a variable named subtotal
```

```
var lastName, state, zipCode;  
// declares three variables
```

The assignment operators

=

+=

-=

*=

/=

%=

How to declare variables and assign values

```
var subtotal = 74.95;           // subtotal is 74.95

var salesTax = subtotal * .1;  // salesTax is 7.495

var isValid = false;          // Boolean value is false

var zipCode = "93711", state = "CA"; // two assignments

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

How to code compound assignment statements

```
var subtotal = 24.50;
```

```
subtotal += 75.50;           // subtotal is 100  
subtotal *= .9;              // subtotal is 90 (100 * .9)
```

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

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