Chapter 2

Getting started with JavaScript

(Part 2)

Section 2-60

Comparisons and Decisions

The relational operators

Operator	Description	Example
`==`	Equal	`lastName == "Harris"` `testScore == 10`
`!=`	Not equal	`firstName != "Ray"` `months != 0`
`<`	Less than	`age < 18`
`<=`	Less than or equal	`investment <= 0`
`>`	Greater than	`testScore > 100`
`>=`	Greater than or equal	`rate / 100 >= 0.1`

The syntax of the global isNaN method

```
isNaN(expression)
```

Examples of the isNaN method

```
isNaN("Harris")      // Returns true
isNaN("123.45")      // Returns false
```

The logical operators in order of precedence

Operator	Description	Example
`!`	NOT	`!isNaN(age)`
`&&`	AND	`age > 17 && score < 70`
`||`	OR	`isNaN(rate) || rate < 0`

How the logical operators work

- Both tests with the AND operator must be true for the overall test to be true.
- At least one test with the OR operator must be true for the overall test to be true.
- The NOT operator switches the result of the expression to the other Boolean value.
- To override the order of precedence when two or more logical operators are used in a conditional expression, you can use parentheses.

Terms for conditional expressions

- conditional expression
- relational operator
- compound conditional expression
- logical operator

The syntax of the if statement

```
if ( condition-1 ) { statements }
[ else if ( condition-2 ) { statements }
  ...
  else if ( condition-n ) { statements } ]
[ else { statements } ]
```

An if statement with an else clause

```
if ( age >= 18 ) {
    alert ("You may vote.");
} else {
    alert ("You are not old enough to vote.");
}
```

An if statement with else if and else clauses

```
if ( isNaN(rate) ) {
    alert ("You did not provide a number for the rate.");
} else if ( rate < 0 ) {
    alert ("The rate may not be less than zero.");
} else if ( rate > 12 ) {
    alert ("The rate may not be greater than 12.");
} else {
    alert ("The rate is: " + rate + ".");
}
```

An if statement with a compound conditional expression

```
if ( isNaN(userEntry) || userEntry <= 0 ) {
    alert ("Please enter a valid number > zero.");
}
```

How to test a Boolean variable

To see if it's true

```
if ( isValid == true ) { }
if ( isValid ) { }           // same as isValid == true
```

To see if it's false

```
if ( isValid == false ) { }
if ( !isValid == true ) { }
if ( !isValid ) { }          // same as !isValid == true
```

The syntax of a while loop

```
while ( condition ) { statements }
```

A while loop that adds the numbers 1 through 5

```
var sumOfNumbers = 0;
var numberOfLoops = 5;
var counter = 1;
while (counter <= numberOfLoops) {
    sumOfNumbers += counter;
    counter++;                  // adds 1 to counter
}
alert(sumOfNumbers);            // displays 15
```

Section 2-70

Loops

The syntax of a do-while loop

```
do { statements } while ( condition );
```

A do-while loop that adds 1 through 5

```
var sumOfNumbers = 0;
var numberOfLoops = 5;
var counter = 1;
do {
    sumOfNumbers += counter;
    counter++;                      // adds 1 to counter
}
while (counter <= numberOfLoops);
alert(sumOfNumbers);                // displays 15
```

A while loop that gets the average of numbers

```
var total = 0, count = 0, number;
number = parseFloat( prompt("Enter a number:") );
while ( !isNaN(number) ) {
    total += number;
    count++;
    number = parseFloat(
        prompt("Enter another number " +
               "or click Cancel to stop:") );
}
var average = total / count;
alert("The average is: " + average);
```

The syntax of a for statement

```
for ( counterInitialization;
      condition;
      incrementExpression ) {
    statements
}
```

A for loop that adds the numbers 1 through 5

```
var sumOfNumbers = 0;
var numberOfLoops = 5;
for ( counter=1; counter <= numberOfLoops; counter++ ) {
    sumOfNumbers += counter;
}
alert(sumOfNumbers);              // displays 15
```

A for loop that calculates future value

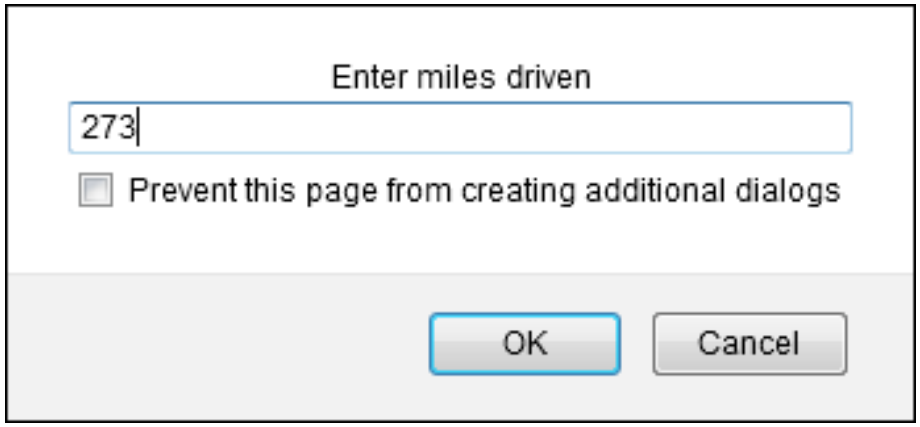
```
var investment = 10000;
var annualRate = 7.0;
var years = 10;
var futureValue = investment;
for ( i = 1; i <= years; i++ ) {
    futureValue += futureValue * annualRate / 100;
}
alert (futureValue);                    // displays 19672
```

Other ways that this could be coded

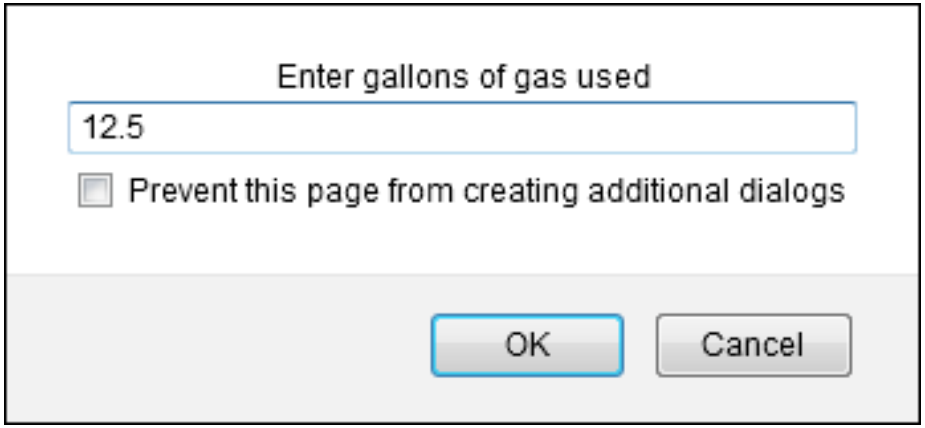
```
futureValue = futureValue +
              (futureValue * annualRate / 100);
futureValue = futureValue * (1 + (annualRate / 100))
```

The Calculate MPG application

The first prompt dialog box

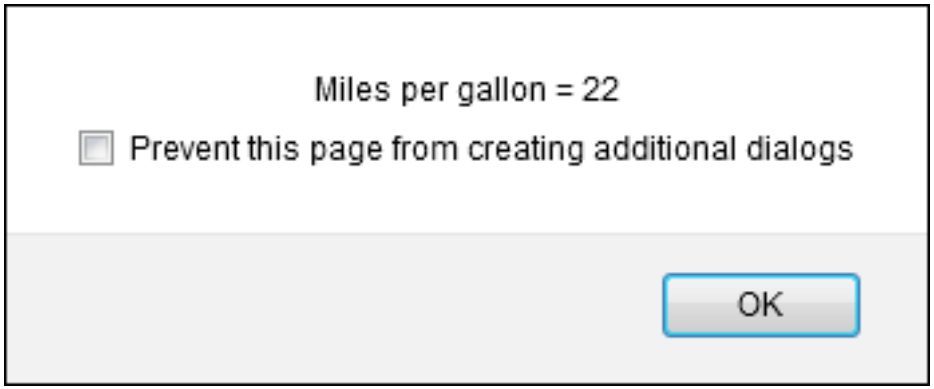


The second prompt dialog box



The Calculate MPG application (continued)

The alert dialog box that displays the result

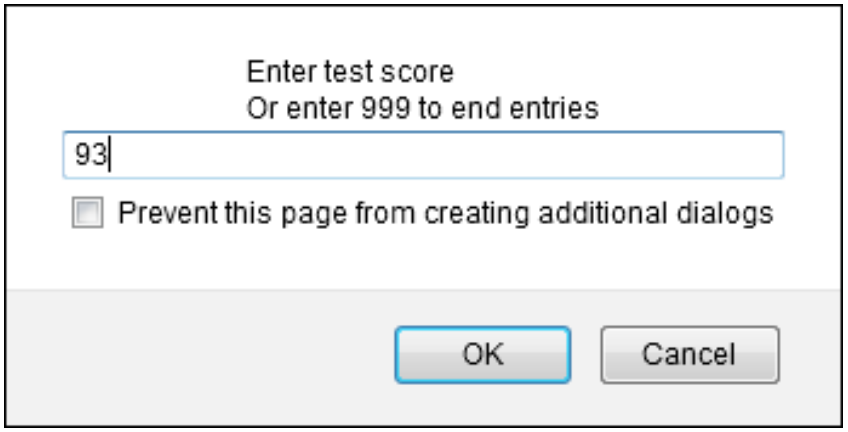


The HTML and JavaScript for the application

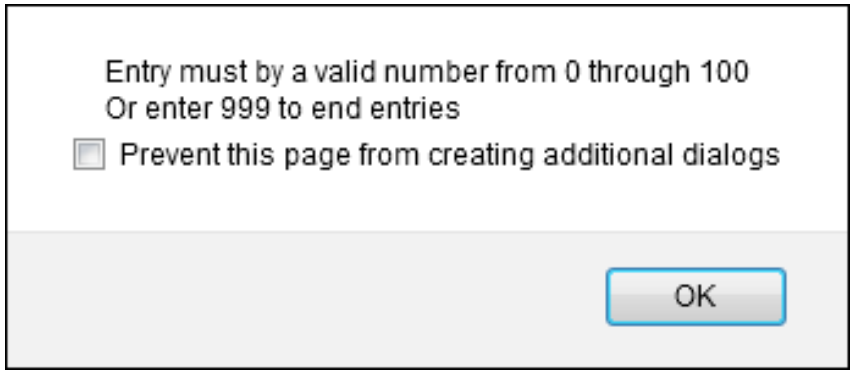
```
<!doctype html>
<html>
<head>
    <title>The Calculate MPG Application</title>
    <script>
        alert("The Calculate MPG application");
        var miles = prompt("Enter miles driven");
        miles = parseFloat(miles);
        var gallons = prompt("Enter gallons of gas used");
        gallons = parseFloat(gallons);
        var mpg = miles/gallons;
        mpg = parseInt(mpg);
        alert("Miles per gallon = " + mpg);
    </script>
</head>
<body>
<section>
  <h1>This page is displayed after the JavaScript runs</h1>
</section>
</body>
</html>
```

The dialog boxes for the Test Scores application

The prompt dialog box for the next test score

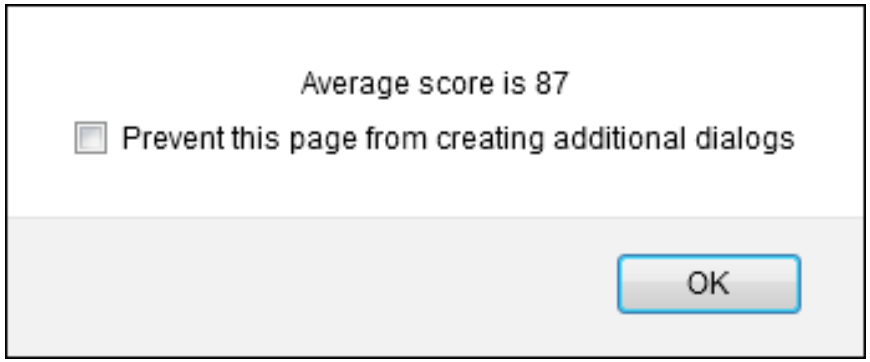


The alert dialog box for an entry error



The Test Scores application (continued)

The alert dialog box that displays the result



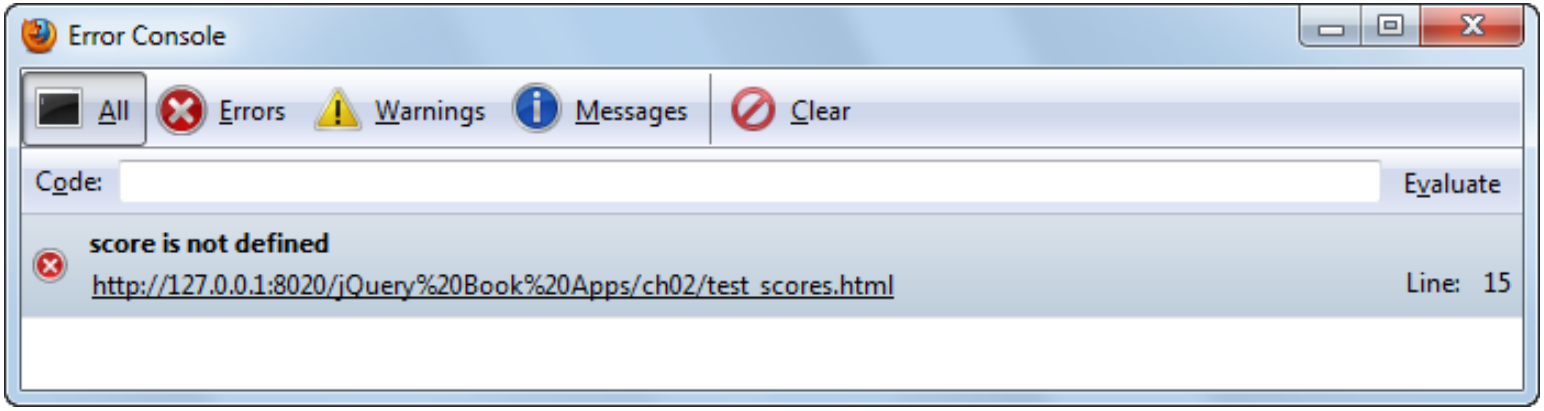
The JavaScript in the HTML head section

```
<script>
    var total = 0;
    var entryCount = 0;
    var entry;
    do {
        entry = prompt("Enter test score\n" +
            "Or enter 999 to end entries", 999);
        entry = parseInt(entry);
        if (entry >= 0 && entry <= 100) {
            total = total + entry;
            entryCount++; }
        else if (entry != 999){
            alert("Entry must by a valid number " +
                  "from 0 through 100\n" +
                  "Or enter 999 to end entries"); }
    }
    while (entry != 999);
    var average = total/entryCount;
    average = parseInt(average);
    alert("Average score is " + average);
</script>
```

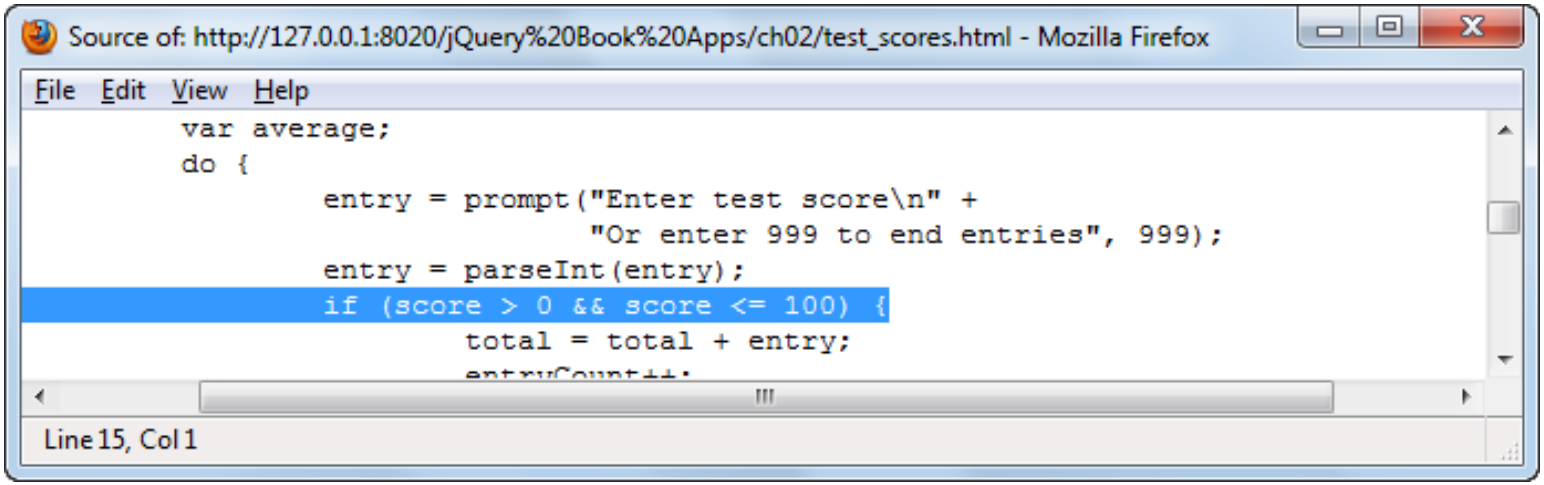
Section 2-80

Viewing Error Messages

The Firefox Error Console with an error



The source code when you click on the link



How to display the Firefox Error Console and the source code for an error

- To display the console, use the Tools→Web Developer→Error Console command, or press Ctrl+Shift+J.
- To display the source code for the error, click on the link in the Error Console.

Common syntax errors

- Misspelling keywords, like coding parseInteger instead of parseInt.
- Omitting required parentheses, quotation marks, or braces.
- Not using the same opening and closing quotation mark.
- Omitting the semicolon at the end of a statement.
- Using one equal sign instead of two when testing for equality.
- Misspelling or incorrectly capitalizing an identifier, like defining a variable named salesTax and referring to it later as salestax.