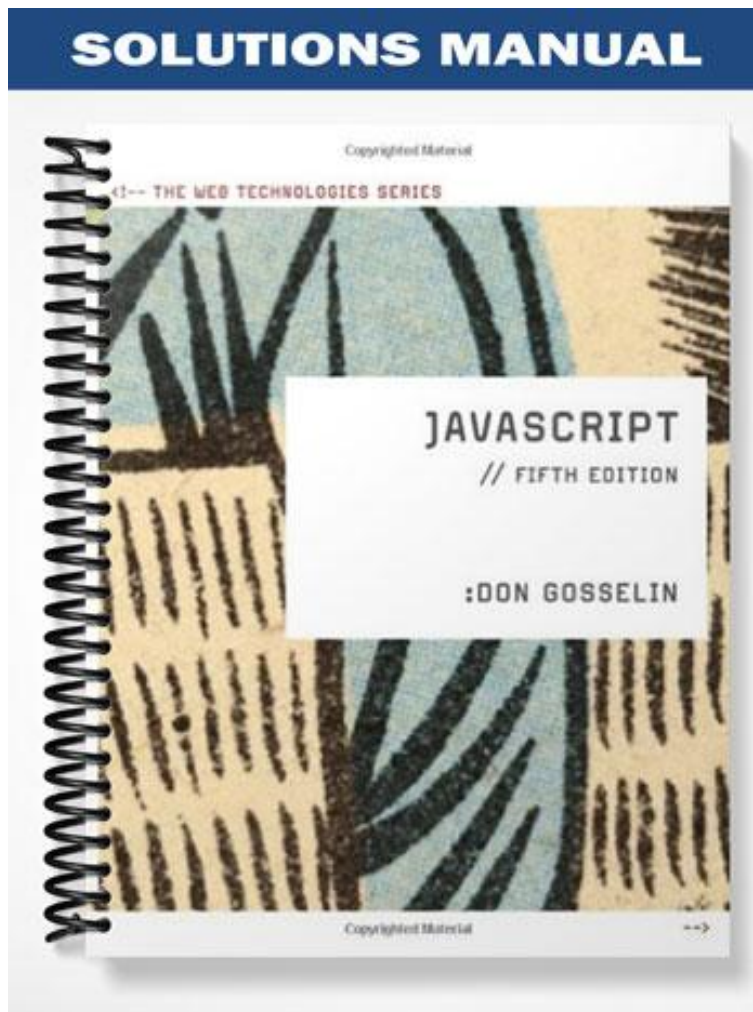


SOLUTIONS MANUAL



Chapter 2

Working with Data Types and Operators

At a Glance

Instructor's Manual Table of Contents

- Overview
- Objectives
- Teaching Tips
- Quick Quizzes
- Class Discussion Topics
- Additional Projects
- Additional Resources
- Key Terms

Lecture Notes

Overview

In this chapter, students will learn about JavaScript variables and data types, and the operations that can be performed on them. Students will start by learning how to declare, initialize, and use variables. Then, they will study the different data types of JavaScript. After that, students will learn about expressions, operands, and the different types of operators available in JavaScript. Then, students will learn more about working with strings. Finally, students will study operator precedence.

Chapter Objectives

- Work with variables
- Study data types
- Use expressions and operators
- Work with strings
- Study operator precedence

Teaching Tips

Using Variables

1. Explain that a variable is a specific location in the computer's memory. However, the term is commonly used to refer to the values a program stores in computer memory.
2. Mention that to use a variable in a program, you first have to write a statement that creates the variable and assigns it a name.

Naming Variable Names

1. Point out that the name you assign to a variable is called an identifier.
2. Mention the following rules and conventions that must be observed when naming a variable:
 - Identifiers must begin with an uppercase or lowercase ASCII letter, dollar sign (\$), or underscore (_).
 - You can use numbers in an identifier, but not as the first character.
 - You cannot include spaces in an identifier.
 - You cannot use reserved words for identifiers.
3. Define reserved words (also called keywords) as special words that are part of the JavaScript language syntax. Use Table 2-1 to identify the JavaScript reserved words.

Declaring and Initializing Variables

1. Mention that creating a variable is known as declaring the variable. In JavaScript, you usually use the reserved keyword `var` to declare variables. To declare a variable named `myVariable`, the following statement is used:

```
var myVariable;
```
2. Point out that when you declare a variable, you can also assign a specific value to, or initialize, the variable using the following syntax:

```
var variable_name = value;
```
3. Mention that the equal sign (`=`) in the preceding statement is called an assignment operator.
4. Describe how to declare multiple variables in a single statement using a single `var` keyword.

<i>Teaching Tip</i>	Mention that even though JavaScript lets you declare a variable without using the <code>var</code> keyword, it is good programming practice to always use the <code>var</code> keyword when declaring variables. Not doing so can lead to problems in some cases.
----------------------------	---

Displaying Variables

1. Describe how to print a variable (that is, display its value on the screen) using the `document.write()` or `document.writeln()` methods.
2. Note that when printing a variable, the plus sign (`+`) is used to combine a literal string with a variable containing a numeric value. The plus sign can also be used to perform arithmetic operations involving variables that contain numeric values.

Modifying Variables

1. Describe how to change a variable's value.
2. Point out that you can change a variable's value at any point in a script.

Quick Quiz 1

1. The values a program stores in computer memory are commonly called _____.
Answer: variables
2. The name you assign to a variable is called a(n) _____.
Answer: identifier

3. Reserved words are also called ____.
Answer: keywords
4. True or False: Identifiers must begin with an ASCII letter.
Answer: False
5. True or False: The plus sign (+) is called the assignment operator.
Answer: False

Working with Data Types

1. Define a data type as the specific category of information that a variable contains.
2. Define primitive types as data types that can be assigned only a single value. Use Table 2-2 to identify the JavaScript primitive types.
3. Define reference, or composite, data types as those that can contain multiple values or complex types of information.
4. Explain the difference between a strongly typed programming language and a loosely typed programming language. Mention that JavaScript is a loosely typed programming language.

Numeric Data Types

1. Mention that JavaScript supports two numeric data types: integers and floating-point numbers.
2. Define an integer as a positive or negative number with no decimal places.
3. Define a floating-point number as a number that contains decimal places or that is written in exponential notation.
4. Explain that exponential notation, or scientific notation, is a shortened format for writing very large numbers or numbers with many decimal places.

Boolean Values

1. Define a Boolean value as a logical value of true or false.
2. Mention that in JavaScript programming, you can only use the words true and false to indicate Boolean values.

Arrays

1. Mention that an array contains a set of data represented by a single variable name. Use Figure 2-8 to show a conceptual example of an array.

2. Explain how to create new arrays using the keyword `new` and the `Array()` constructor.
3. Mention that each piece of data contained in an array is called an element.
4. Define an index as an element's numeric position within the array. Point out that the numbering of elements within an array starts with an index number of zero (0).

<i>Teaching Tip</i>	The concept of an array will be confusing to students who have not dealt with arrays before. Be prepared to spend some time in introducing this subject to students. You may also want to draw a model of an array and its elements, along with the index of each element. The fact that the first element has an index number of 0 might cause a lot of confusion and even some errors in the work of students in the beginning.
----------------------------	---

5. Explain how to access the elements of an array.
6. Demonstrate how to modify the elements of an array.
7. Discuss how to use the `length` property to determine the number of elements in an array.

Quick Quiz 2

1. A(n) ____ is the specific category of information that a variable contains.
Answer: data type
2. Data types that can be assigned only a single value are called ____.
Answer: primitive types
3. True or False: The `String` data type is a reference data type in JavaScript.
Answer: False
4. A(n) ____ is a positive or negative number with no decimal places.
Answer: integer
5. True or False: The numbering of elements within an array starts with an index number of zero (0).
Answer: True

Building Expressions

1. Define an expression as a literal value or variable or a combination of literal values, variables, operators, and other expressions that can be evaluated by the JavaScript interpreter to produce a result.
2. Define operands as variables and literals contained in an expression. Mention that a literal is a value such as a literal string or a number.
3. Define operators as symbols used in expressions to manipulate operands. Use Table 2-3 to discuss the main types of JavaScript operators.
4. Explain the difference between a binary operator and a unary operator.

Arithmetic Operators

1. Mention that arithmetic operators are used in JavaScript to perform mathematical calculations, such as addition, subtraction, multiplication, and division.
2. Use Table 2-4 to show the binary arithmetic operators available in JavaScript.
3. Use Table 2-5 to show the unary arithmetic operators available in JavaScript.
4. Explain the difference between a prefix operator and a postfix operator.

Assignment Operators

1. Mention that assignment operators are used for assigning a value to a variable.
2. Use Table 2-6 to discuss the assignment operators available in JavaScript.

Comparison and Conditional Operators

1. Define comparison operators as those that are used to compare two operands, and determine if one numeric value is greater than another. A Boolean value of true or false is returned after two operands are compared.
2. Use Table 2-7 to point out the JavaScript comparison operators.
3. Point out that the conditional operator executes one of two expressions, based on the results of a conditional expression.
4. Describe the syntax for the conditional operator.

Logical Operators

1. Explain that logical operators are used for comparing two Boolean operands for equality. A Boolean value of true or false is returned after two operands are compared.

2. Use Table 2-8 to describe the JavaScript logical operators.

Quick Quiz 3

1. ____ are variables and literals contained in an expression.
Answer: Operands
2. True or False: A unary operator requires an operand before and after the operator.
Answer: False
3. True or False: A prefix operator is placed before a variable.
Answer: True
4. ____ operators are used for assigning a value to a variable.
Answer: Assignment
5. ____ operators are used for comparing two Boolean operands for equality.
Answer: Logical

Working with Strings

1. Remind students that a text string is text that is contained within double or single quotation marks.
2. Mention that you can use text strings as literal values or assign them to a variable.
3. Point out that literal strings can be assigned a zero-length string value called an empty string.

String Operators

1. Point out that when used with strings, the plus sign is known as the concatenation operator. It is used to combine two strings. Show how the operator works with strings.
2. Mention that the compound assignment operator (+=) can also be used to combine two strings. Show how this operator works with strings.

Escape Characters and Sequences

1. Define an escape character as a character that tells the compiler or interpreter that the character that follows it has a special purpose. In JavaScript, the escape character is the backslash (\).
2. Mention that when you combine the escape character with other characters, the combination is called an escape sequence. Most escape sequences carry out special functions.

3. Use Table 2-9 to show the JavaScript escape sequences.

Special Operators

1. Use Table 2-10 to discuss the special operators in JavaScript. These operators are used for various purposes and do not fit within any other category.
2. Discuss why the `typeof` operator is useful.
3. Describe the syntax of the `typeof` operator. Use Table 2-11 to list the values that can be returned by the `typeof` operator.

Understanding Operator Precedence

1. Explain what is meant by the term operator precedence.
2. Use Table 2-12 to show the order of precedence for JavaScript operators.
3. Define an operator's associativity as the order in which operators of equal precedence execute.
4. Use Figure 2-20 to explain left to right associativity, and use Figure 2-21 to explain right to left associativity.

Quick Quiz 4

1. True or False: Empty strings are valid values for literal strings.
Answer: True
2. True or False: You can use the compound assignment operator (`+=`) to combine two strings.
Answer: True
3. In JavaScript, the escape character is the _____.
Answer: backslash (\)
4. When you combine the escape character with other characters, the combination is called a(n) _____.
Answer: escape sequence
5. Operator _____ is the order in which operators of equal precedence execute.
Answer: associativity

Class Discussion Topics

1. Why does there need to be a set order of precedence for the JavaScript operators?
2. Is it *necessary* to use arrays in programming? Explain your reasoning. Is it *very useful* to use arrays in programming? Explain your reasoning.

Additional Projects

1. Write a script in which you:
 - a. Declare a variable called `employee1` and assign it the value: “Susan Harper”
 - b. Declare a variable called `hoursWorked1` and assign it the value: 45
 - c. Print the variables onto the screen in such a way that the following is displayed on the screen: Susan Harper has worked 45 hours.
2. Declare an array called `cities` that has six elements. Assign the following values to the elements:
 - a. First element – New York
 - b. Second element – Chicago
 - c. Third element – Los Angeles
 - d. Fourth element – Austin
 - e. Fifth element – Miami
 - f. Sixth element - Philadelphia

Additional Resources

1. JavaScript Operations on Variables:
www.functionx.com/javascript/Lesson04.htm
2. JavaScript Tutorial – Variables:
www.howtcreate.co.uk/tutorials/javascript/variables
3. JavaScript Arrays - Creating and Storing Values:
www.webdevelopersnotes.com/tutorials/javascript/javascript_arrays.php3

Key Terms

- **Arithmetic operators** are used in JavaScript to perform mathematical calculations, such as addition, subtraction, multiplication, and division.
- An **array** contains a set of data represented by a single variable name.
- **Assignment operators** are used for assigning a value to a variable.
- **Associativity** is the order in which operators of equal precedence execute.
- A **binary operator** requires an operand before and after the operator.
- A **Boolean value** is a logical value of true or false.

- **Comparison operators** are used to compare two operands and determine if one numeric value is greater than another.
- **Reference**, or **composite**, data types can contain multiple values or complex types of information, as opposed to the single values stored in primitive data types.
- **Compound assignment operators** perform mathematical calculations on variables and literal values in an expression, and then assign a new value to the left operand.
- The **concatenation operator** (+) is used to combine two strings.
- The **conditional operator** executes one of two expressions, based on the results of a conditional expression.
- A **constructor** is a special type of function that is used as the basis for creating reference variables (that is, variables whose data type is the reference data type).
- A **data type** is the specific category of information that a variable contains.
- Using a statement to create a variable is called **declaring** the variable.
- Loose typing is known as **dynamic typing** because data types can change after they have been declared.
- Each piece of data contained in an array is called an **element**.
- A zero-length string is called an **empty string**.
- An **escape character** tells the compiler or interpreter that the character that follows it has a special purpose.
- When you combine the escape character with other characters, the combination is called an **escape sequence**.
- **Exponential notation**, or **scientific notation**, is a shortened format for writing very large numbers or numbers with many decimal places.
- An **expression** is a literal value or variable or a combination of literal values, variables, operators, and other expressions that can be evaluated by the JavaScript interpreter to produce a result.
- A **floating-point number** is a number that contains decimal places or that is written in exponential notation.
- The name you assign to a variable is called an **identifier**.
- An **index** is an element's numeric position within the array.
- When you assign a specific value to a variable, you are said to **initialize** the variable.
- An **integer** is a positive or negative number with no decimal places.
- **Reserved words** (also called **keywords**) are special words that are part of the JavaScript language syntax.
- The **length property** returns the number of elements in an array.
- A **literal** is a value such as a literal string or a number.
- **Logical operators** are used for comparing two Boolean operands for equality.
- Programming languages that do not require you to declare the data types of variables are called **loosely typed programming languages**.
- **Operands** are variables and literals contained in an expression.
- The term **operator precedence** refers to the order in which operations in an expression are evaluated.
- **Operators**, such as the addition operator (+) and multiplication operator (*), are symbols used in expressions to manipulate operands.
- A **postfix operator** is placed after a variable.
- A **prefix operator** is placed before a variable.
- Data types that can be assigned only a single value are called **primitive types**.
- **Reference**, or **composite**, data types can contain multiple values or complex types of information, as opposed to the single values stored in primitive data types.

- **Reserved words** (also called **keywords**) are special words that are part of the JavaScript language syntax.
- **Exponential notation**, or **scientific notation**, is a shortened format for writing very large numbers or numbers with many decimal places.
- Strong typing is known as **static typing**, because data types do not change after they have been declared.
- Programming languages that require you to declare the data types of variables are called **strongly typed programming languages**.
- A **unary operator** requires a single operand either before or after the operator.
- The values a program stores in computer memory are commonly called **variables**.