Chapter 18: Get With the Program: Fundamental Concepts Expressed in JavaScript

Fluency with Information Technology
Third Edition

by
Lawrence Snyder

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Overview: Programming Concepts

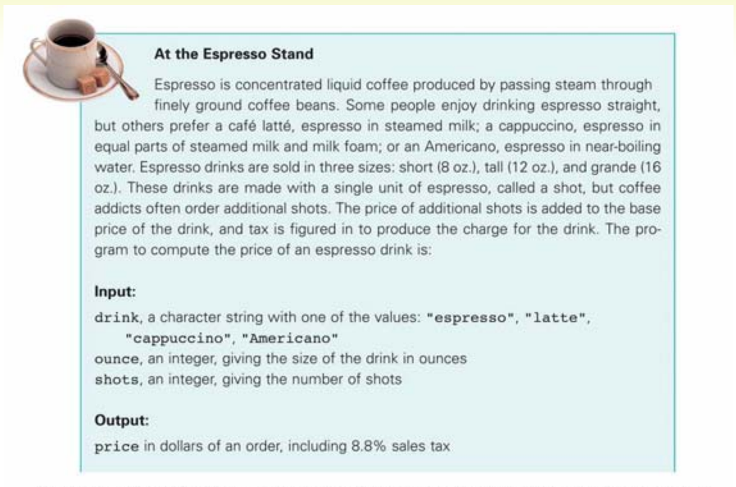
- Programming: Act of formulating an algorithm or program
- Basic concepts have been developed over last 50 years to simplify common programming tasks
- Concepts will be expressed in JavaScript

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Programming Concepts

- Names, values, variables
- Declarations
- Data types, numbers, string literals and Booleans
- Assignment
- Expressions
- Conditionals

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley



At the Espresso Stand

Espresso is concentrated liquid coffee produced by passing steam through finely ground coffee beans. Some people enjoy drinking espresso straight, but others prefer a café latté, espresso in steamed milk; a cappuccino, espresso in equal parts of steamed milk and milk foam; or an Americano, espresso in near-boiling water. Espresso drinks are sold in three sizes: short (8 oz.), tall (12 oz.), and grande (16 oz.). These drinks are made with a single unit of espresso, called a shot, but coffee addicts often order additional shots. The price of additional shots is added to the base price of the drink, and tax is figured in to produce the charge for the drink. The program to compute the price of an espresso drink is:

Input:

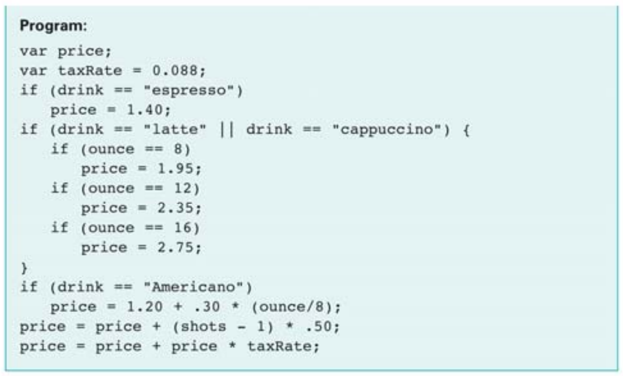
`drink`, a character string with one of the values: `"espresso"`, `"latte"`, `"cappuccino"`, `"Americano"`
`ounce`, an integer, giving the size of the drink in ounces
`shots`, an integer, giving the number of shots

Output:

`price` in dollars of an order, including 8.8% sales tax

Figure 18.1. Sample JavaScript computation to figure the cost of espresso drinks. (continues next page).

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley



Program:

```
var price;
var taxRate = 0.088;
if (drink == "espresso")
   price = 1.40;
if (drink == "latte" || drink == "cappuccino") {
   if (ounce == 8)
      price = 1.95;
   if (ounce == 12)
      price = 2.35;
   if (ounce == 16)
      price = 2.75;
}
if (drink == "Americano")
   price = 1.20 + .30 * (ounce/8);
price = price + (shots - 1) * .50;
price = price + price * taxRate;
```

Figure 18.1 (continued). Sample JavaScript computation to figure the cost of espresso drinks.

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Names, Values, And Variables

- Names Have Changing Values
  - Example: U.S. President has current value of George W. Bush, previous values of Bill Clinton, George Washington
- Names in a Program Are Called *Variables*
  - Values associated with a name change in programs using the *assignment* statement (=)

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Identifiers and Their Rules

- *Identifier* is the character sequence that makes up a variable's name
  - Must have a particular form
    - Must *begin* with a letter or underscore (_) *followed by* any sequence of letters, digits, or underscore characters
    - Cannot contain spaces
    - Case sensitive (Capitalization matters!)

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Identifiers and Their Rules

Valid	**Invalid**
firstOne	1stOne
first1	first-1
first_1	first$1
first_One	first One
FirstOne	First1!

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

A Variable Declaration Statement

- Declaration: State what variables will be used
  - Command is the word *var*
  - For example, a program to calculate area of circle given radius, needs variables area and radius:
    - var radius, area;
- The declaration is a type of *statement*

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

The Statement Terminator

- A program is a list of statements
- The statements may be run together on a line
- Each statement is terminated by the *statement terminator* symbol
  - In JavaScript, it is the *semicolon* (;)

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Rules for Declaring Variables

- Every variable used in a program must be declared (before it is used)
  - In JavaScript declaration can be anywhere in the program
  - Programmers prefer to place them first
- Undefined values
  - Variable has been declared but does not yet have a value

```
var number1;           // undefined value
var number2 = 42;      // initialized to the value 42
```

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Initializing a Declaration

- We can set an initial value as part of declaration statement:
  - var taxRate = .088;
- Related variables may be grouped in one declaration/initialization; unrelated variables are usually placed in separate statements

```
var num1 = 42, num2, num3;
```

```
var num1 = 42;
var num2;
var num3;
```

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Three Basic Date Types of Javascript

- Numbers
- Strings
- Booleans
  - These kind of values are called *data types* or just *types*

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Numbers

- Rules for Writing Numbers
  - There are no "units" or commas
  - Can have about 10 significant digits and can range from 10^{-324} to 10^{308}

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Strings

- Strings are sequences of keyboard characters
- Strings are always surrounded by single (' ') or double quotes (" ")
- Strings can initialize a declaration
  - var hairColor = "black";

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Rules for Writing Strings in JavaScript

- Must be surrounded by single or double quotes
- Allow most characters except return (Enter), backspace, tab, \
- Double quoted strings can contain single quoted strings and vice versa
- The apostrophe (') is the same as the single quote
- Any number of characters allowed in a string
- Minimum number of characters is zero (""), which is the *empty string*

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Literals

- String Literals stored in the computer
  - Quotes are removed (they are only used to delimit the string literal)
  - Any character can be stored in memory
    - Even a character that cannot be typed can be stored, using escape mechanism – in JavaScript, the backslash (\ )

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Table 18.1 Escape sequences for characters prohibited from string literals

Sequence	Character	Sequence	Character
`\b`	Backspace	`\f`	Form feed
`\n`	New line	`\r`	Carriage return
`\t`	Tab	`\'`	Apostrophe or single quote
`\"`	Double quote	`\\`	Backslash

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Boolean Values

- Two logical values: True and False
- They are values, not identifiers or strings
- Used implicitly throughout programming process; only occasionally for initializing variables
  - Mostly used to compare data or make decisions

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

The Assignment Statement

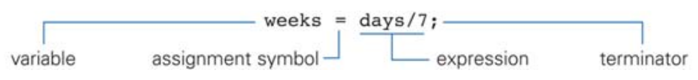
- Changes a variable's value

  <variable> <assignment symbol> <expression>;

- Assignment Symbol:
  - In JavaScript, the equal sign (=)
  - Example:
    - weeks = days / 7;

      ←

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley



Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Interpreting an Assignment Statement

- Value flows from the right side to the left side
- Read the assignment symbol as "is assigned" or "becomes" or "gets"
- The expression (right side) is computed or evaluated first
  - If there are any variables in it, their current value is used
- Then this computed value becomes the value of the variable on the left side

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Three Key Points about Assignment

- All three of the components must be given
  - if anything is missing, the statement is meaningless
- Flow of value to name is always right to left
- Values of any variables used in the expression are always their values before the start of the execution of the assignment

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

An Expression and its Syntax

- Algebra-like formula called an *expression*
  - Describe the means of performing the actual computation
  - Built out of values and *operators*
    - Standard *arithmetic operators* are symbols of basic arithmetic

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Arithmetic Operators

- Multiplication must be given explicitly with the asterisk (*) multiply operator
- Multiply and divide are performed before add and subtract
  - Unless grouped by parentheses
- JavaScript does not have an operator for exponents
- *Binary operators* operate on two *operands* (like + and *)
- *Unary operators* operate on one operand (like - for negate)
- *Modulus* or mod (%) divides two integers and returns the remainder

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Relational Operators

- Make comparisons between numeric values
- Outcome is a Boolean value, true or false
- < less than
- <= less than or equal to
- == equal to

  (Note difference between = and ==)
- != not equal to
- >= greater than or equal to
- > greater than

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Logical Operators

- To test two or more relationships together
  - Teenagers are older than 12 and younger than 20
- Logical And
  - Operator is &&
  - Outcome of a && b is true if both a and b are true; otherwise it is false
- Logical Or
  - Operator is ||
  - Outcome of a || b is true if either a is true or b is true
- Logical Not
  - Operator is !
  - Unary operator. Outcome is opposite of value of operand

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Operators (cont'd)

- Operator Overload
  - Use of an operator with different data types
  - Case of interest in JavaScript is +
- Addition
  - When used with numbers, it adds
    - 4 + 5 produces 9
- Concatenation
  - When + is used with strings, it concatenates or joins the strings together
    - "four" + "five" produces "fourfive"

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

A Conditional Statement

```
if ( <Boolean expression> )
   <then-statement>;
```

- Boolean expression is a relational expression; then-statement is any JavaScript statement

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

If Statements and Their Flow of Control

- The Boolean statement, called a predicate, is evaluated, producing a true or false outcome
- If the outcome is true, the then-statement is performed
- If the outcome is false, the then-statement is skipped
- Then-statement can be written on the same line as the Boolean or on the next line

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Compound Statements

- Sometimes we need to perform more than one statement on a true outcome of the predicate test
- You can have a sequence of statements in the then clause
- Group these statements using curly braces {}
  - They are collected as a compound statement

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

if/else Statements

- To execute statements if a condition is false

```
if ( <Boolean expression> )
{
    <then-statements>;
}
else
{
    <else-statements>;
}
```

- The Boolean expression is evaluated first
  - If the outcome is true, the then-statements are executed and the else-statements are skipped
  - If the outcome is false, the then-statements are skipped and the else-statements are executed

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Nested if/else Statements

- The then-statement and the else-statement can contain an if/else
- The else is associated with the immediately preceding if
- Correct use of curly braces ensures that the else matches with its if

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

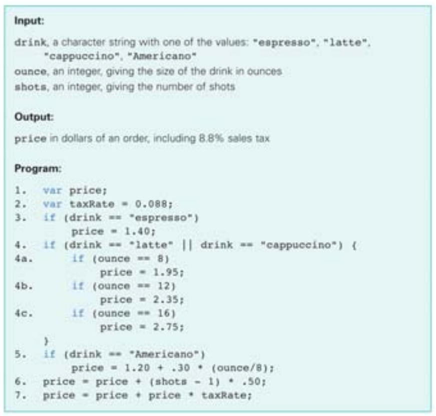
Nested if/else Statements

```
if (<Boolean exp1>)
   if (< Boolean exp2>)
   {
      <then-stmts for exp2>;
   }
   else
   {
      <else-stmts for exp2>;
   }
```

```
if (<Boolean exp1>)
{
   if (< Boolean exp2>)
   {
      <then-stmts for exp2>;
   }
}
else
{
   <else-stmts for exp1>;
}
```

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

The Espresso Program



Input:

drink, a character string with one of the values: "espresso", "latte", "cappuccino", "Americano"
ounce, an integer, giving the size of the drink in ounces
shots, an integer, giving the number of shots

Output:

price in dollars of an order, including 8.8% sales tax

Program:

```
1.   var price;
2.   var taxRate = 0.088;
3.   if (drink == "espresso")
         price = 1.40;
4.   if (drink == "latte" || drink == "cappuccino") {
4a.      if (ounce == 8)
             price = 1.95;
4b.      if (ounce == 12)
             price = 2.35;
4c.      if (ounce == 16)
             price = 2.75;
     }
5.   if (drink == "Americano")
         price = 1.20 + .30 * (ounce/8);
6.   price = price + (shots - 1) * .50;
7.   price = price + price * taxRate;
```

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

The Espresso Program

- Line 3 is a basic conditional statement
- Lines 4-4c use an if statement with conditionals in the then statement
- Line 5 uses basic if statement
- Lines 6, 7 compute using arithmetic operators

Copyright © 2008 Pearson Education, Inc. Publishing as Pearson Addison-Wesley