Test Your Tech

JavaScript is:
A. The earliest known writing by Java Man.
B. Programming language for Web pages.
C. Instructions in the Starbucks bag on how to brew good coffee.

Help in HTML

- W3 Schools
  - HTML tag reference
  - Tutorials

Get With the Program:

Fundamental Programming Concepts Expressed in JavaScript

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

Programming Concepts

- Names, values, variables
- Declarations
- Data types, numbers, string literals and Booleans
- Assignment
- Expressions
- Conditionals, or branches
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 (something = something else)

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!)

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
The Statement Terminator

• A program is a list of statements
• The statements may be run together on a line
  * Use whatever spacing you need to read your code and understand your program
• Each statement is terminated by the statement terminator symbol
  * In JavaScript, all statements terminate with the semicolon (;)

Names, Values, And Variables

• Declaring a variable
  * Names a particular area in computer memory where you can store values
  * Gives you a name, or handle, that is independent of the current value

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

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;

Three Basic Date Types of Javascript

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

Numbers

• Rules for Writing Numbers
  * There are no “units” or commas
  * Can have about 10 significant digits and can range from $10^{-308}$ to $10^{308}$
Strings

- Strings are sequences of keyboard characters
- Strings are always surrounded by single (') or double quotes ("")
  - No smart quotes!
- Strings can initialize a declaration
  - `var hairColor = "black";`
- Quotes can nest
  - `firstLine = "Johnson called, 'Dude!'"`

Rules for Writing Strings

- Must be surrounded by single or double quotes
- Allow most characters except return (Enter), backspace, tab, \n- 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

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 (\)

Table 18.1: Escape sequences for characters prohibited from string literals

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Character</th>
<th>Sequence</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>\b</td>
<td>Backspace</td>
<td>\f</td>
<td>Form feed</td>
</tr>
<tr>
<td>\n</td>
<td>New line</td>
<td>\r</td>
<td>Carriage return</td>
</tr>
<tr>
<td>\t</td>
<td>Tab</td>
<td>&quot;</td>
<td>Apostrophe or single quote</td>
</tr>
<tr>
<td>&quot;</td>
<td>Double quote</td>
<td>&quot;</td>
<td>Backslash</td>
</tr>
</tbody>
</table>

Comments

- HTML
  - <!-- HTML comments -->
- JavaScript
  - `// Single-line JavaScript comment`
  - `/* Multi-line JavaScript comment continues for more than one line */`

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
Assigning Values to Variables

- Assign values to variables with an assignment operator.
- We’ll use = for now.

```javascript
var yourAge, acctBal, custName;
yourAge = 32; // store 32 in yourAge
acctBal = 100.75; // store 100.75 in acctBal
custName = "Jeff"; // store 'Jeff' in custName
isCustomer = true; // store boolean true in isCustomer (no quotes)
var yourName = "Jeff"; // alternate all-in-one line assignment statement
```

Assignment Statement

- Flow moves from right to left.
- Results of the <expression> replace the value stored in the <variable>.

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

<table>
<thead>
<tr>
<th>Line</th>
<th>Assignment Statement</th>
<th>Value in</th>
<th>Value in</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var yourName = &quot;Sarah&quot;;</td>
<td>yourName</td>
<td>&quot;Sarah&quot;</td>
</tr>
<tr>
<td>2</td>
<td>var myName = &quot;Andrea&quot;;</td>
<td>myName</td>
<td>&quot;Andrea&quot;</td>
</tr>
<tr>
<td>3</td>
<td>var yourName = myName;</td>
<td>yourName</td>
<td>myName</td>
</tr>
<tr>
<td>4</td>
<td>var yourName = &quot;myName&quot;;</td>
<td>yourName</td>
<td>&quot;myName&quot;</td>
</tr>
</tbody>
</table>

Other Assignment Operators

<table>
<thead>
<tr>
<th>Line</th>
<th>Assignment Statement</th>
<th>Value in</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var myAge = 32;</td>
<td>myAge</td>
</tr>
<tr>
<td>2</td>
<td>myAge = myAge + 2;</td>
<td>myAge</td>
</tr>
<tr>
<td>3</td>
<td>myAge += 2;</td>
<td>myAge</td>
</tr>
<tr>
<td>4</td>
<td>myAge ++;</td>
<td>myAge</td>
</tr>
<tr>
<td>5</td>
<td>myAge = 3;</td>
<td>myAge</td>
</tr>
<tr>
<td>6</td>
<td>myAge -= 3;</td>
<td>myAge</td>
</tr>
</tbody>
</table>

Assignment

- Three Key Points
  - All three of the components must be given
  - If anything is missing, the statement is meaningless
  - Flow of value to identifier 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

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
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
  - Within parentheses multiply and divide are performed first
- 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

Relational Operators

- Make comparisons between numeric values
- Outcome is a Boolean value, true or false
- < less than
- <= less than or equal to
- == equal to
- != not equal to
- >= greater than or equal to
- > greater than

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

A Conditional Statement

if ( <Boolean expression> )
  <then-statement>;
- Boolean expression is a relational expression; then-statement is any JavaScript statement

Operators (cont’d)

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

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
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.

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.

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.

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.
Summary

Programming is the exact specification of an algorithm
JavaScript is typical... with many rules

• Learning strategy
  • Do the reading first
  • Practicing is better than memorizing for learning the rules
  • Use the program-save-reload-check plan
  • Precision is your best friend