Variables, Operators, and Expressions

February 14th 2006
Outline

- SFTP demo
- HW4 recap and notification
- Javascript and HTML review
- Define “Expressions”
- Discuss how to represent “data” in a program
  - variable name
  - variable type
- List and discuss basic operations on variables
Expressions

• A program is a sequence of statements
• A program is a collection of functions and the data those functions operate on
• The building blocks of statements are expressions: combinations of language key words, function calls, operators, and operands that evaluate to a value
Review: What is a Program?

- A program is a sequence of instructions that operates on data.
- A program is a collection of variables and functions that process the data held in those variables.
- Computers process long strings of 1's and 0's.
  - Need a way to refer to portions of those strings as higher-level data objects.
Variables
A variable is a container for data

Unstructured Data

345
Michael
Patient SSN: 999550000
3.141592658
Structured Data

Unstructured Data
What exactly is a Variable?

• A variable is the concept of a piece of structured data that can be accessed (read or modified) via well-known, standard rules
• A variable is NOT JUST the data it contains!
• A variable also has:
  - a **name** or identifier that provides a way to refer to it
  - a **type** that defines its size (how much memory it uses)
  - a **location** or memory address specifying where the data is stored
Example:: Simple Integer Values

- Suppose we want to write a program for processing a student's grades
- Need a variable to hold the total score

Variable name: total_score: 345
Example:: Variable Declaration

- Declaring a variable is a standard action to let the rest of the program know about a piece of data that will be used.
- The following **program statement** declares (that is, tells the computer to set aside a memory location) a variable called 'total_score'

```plaintext
var total_score;
```

- **var**: keyword
- **total_score**: variable name
Declaring Variables

• Variables are usually declared at the beginning of the program or function they are used in
• Variable names can be any combination of letters, numbers, or underscores, but must start with a letter or underscore
• Variable names should be descriptive; avoid names like 'ab', 'x', 'tmp', etc.
• Make sure you don't try to name a variable after a reserved word (if, for, while, switch ...)

A Simple Statement: Initialization

• You can initialize a variable when you declare it
  – This gives the variable a definite “starting value”
  – To initialize a variable, use an assignment statement.

• An assignment statement computes the value of the expression on the right hand side and stores it in the variable named on the left hand side.
  – “=” should be read “stores the result of”

var total_score = 0;
Initializing a Variable

```html
<script language="Javascript">
var total_score = 0;
//...
function foo()
{
    document.write(total_score);
}
</script>
```

<table>
<thead>
<tr>
<th>address</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1022</td>
<td></td>
</tr>
<tr>
<td>1023</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>1025</td>
<td></td>
</tr>
<tr>
<td>1026</td>
<td></td>
</tr>
<tr>
<td>1027</td>
<td>00000000</td>
</tr>
</tbody>
</table>
Subtle Point About Variable Names

- When you program, you see the variable name
- When the computer executes your program, it actually sees the variable memory address
- In both cases, the data is used behind the scenes
Language Operators
Operators Overview

• You are familiar with many operators from basic math and logic:
  – addition (+), subtraction (-), multiplication (*), division (/)
  – AND (&&), OR (||), NOT (!)

• Operators are basically common functions that take their input and produce some output

• Common enough to have their own symbols in a programming language (see above)
Operators (cont.)

- Javascript has many operators
  - Some you are familiar with (see previous page)
  - Some not: mod, bitwise AND, OR, XOR, relational

- Operators are:
  - unary (take one argument, e.g., !-)
  - binary (take two arguments, e.g., +-*/<>==)
  - ternary (take three arguments)

- Classifications:
  - arithmetic, logic, relational, assignment
**Operator Context**

- **Warning!** Operators are represented by symbols. Sometimes, the symbols may mean something completely different based on context. For example:

  ```javascript
  var x = -1; // the '-' operator is negation
  ```

  ```javascript
  var x = 4 - 3; // the '-' operator is subtraction
  ```
**Arithmetic Operators**

- Addition is represented by +
  
  e.g., \( \text{sum} = x + y \);
- Subtraction is rep. by -
  
  e.g., \( \text{diff} = x - y \);
- Multiplication is rep. by *
  
  e.g., \( \text{scale} = x \times y \);
- Division is rep. by /
  
  e.g., \( \text{quotient} = x / y \);
- Modulus is rep. by %
  
  e.g., \( \text{remainder} = x \% y \);
Relational Operators

• Assignment operator is =, e.g., var result = x;
• Equality operator is ==, e.g., result = (x==y);
• String equality is ===
• Not equal is expressed by !=, result = (x!=y);
• String inequality is !==
• Less than: <
• Greater than: >
• Less than or equal to <=
• Greater than or equal to >=
Logical Operators

- **AND**: \((x && y)\)
- **OR**: \((x || y)\)
- **NOT**: \((!x)\)
Order of Operations

- PEMDAS (power, exponent, mul, div, add, sub)
- For everything else, use parenthesis () to say what you mean
Types

- Three basic types:
  - string
  - number
  - Boolean

- Use “typeof” keyword to distinguish
Things We Haven't Covered Here

• Increment and decrement operators
• Assignment operators
• The ternary condition operator
• Short circuit boolean evaluation
• Collections of data types and variables (arrays, next session)