Announcements

- Due dates
  - Project 1B—Wednesday by 10pm
  - 1-1-1 rule Thursday by 10pm
  - Only once during quarter!
- Lab 5—Friday by 10pm
- Next week
  - Labs 6/7—Tuesday by 10pm

Objectives

- Learn basic programming concepts common to all programming languages
- Apply them to Web pages using JavaScript
- We’ll spend a couple weeks on this journey

Basic Programming Concepts

Get with the Program

- Documenting your code with comments
- Data types (math, string, boolean)
- Variables
- Assigning values to variables
- Expressions
- Conditionals, branches, or tests (all names for same thing)
- Loops, or iterations (both names for same thing)
- Arrays, lists, or collections (all names for same thing)
- Functions and Methods

Programming Concepts

- Basic concepts have been developed over last 50 years to simplify common programming tasks
- Programming concepts will be implemented in JavaScript in this course
  - Easy syntax
  - Immediate results
  - No special software required beyond NotePad++
  - All the major browsers include JavaScript interpreters
**DATA TYPES**

- Currency, string, number, boolean, datetime

---

**Strings**

- The quick brown fox jumped over the lazy dog.

---

- String = a sequence of keyboard characters
- Always surrounded by single (') or double quotes ("")
  - No smart quotes! (" " and ")
- Initialize a declaration
  - var hairColor = "black";
- Quotes can nest
  - firstLine = "Johnson called, 'Dude!' ";

---

- Any number of characters allowed in a string
- Minimum number of characters is none ("")
  - the empty string

---

**Numbers**

- How are they stored in the computer?
  - Quotes are removed (they are only used to 'delimit' the string literal)
    - Delimit means to mark the start and end of the literal

---

- Rules for Writing Numbers
  - No "units" or commas
    - 5884559 NOT $5,884,559
  - Up to 10 significant digits
  - Range from $10^{-324}$ to $10^{308}$
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

Names, Values, and Variables

- Names in a Program Are Called Identifiers
- Variables store values and give you a handy way to refer to the current value in the variable
  - Like we say “The President” to refer to our current president
- Names Have Changing Values
  - U.S. President
    - current value is George W. Bush
    - previous values were Bill Clinton, George Washington

Variables

- Variables are named areas in memory
- We can refer to the value in the memory area without knowing its value, just by calling its name

Quick Clicks

- One question
Identifiers and Their Rules

- Case sensitive:
  HOME ≠ Home ≠ home

<table>
<thead>
<tr>
<th>Valid</th>
<th>Invalid</th>
<th>Reason It's Invalid</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstOne</td>
<td>1stOne</td>
<td>Begins with number</td>
</tr>
<tr>
<td>first1</td>
<td>first-1</td>
<td>JS thinks hyphen is a minus sign</td>
</tr>
<tr>
<td>firstOne</td>
<td>first$1</td>
<td>$ not allowed</td>
</tr>
<tr>
<td>first_one</td>
<td>first One</td>
<td>Space not allowed</td>
</tr>
<tr>
<td>first_1</td>
<td>First1!</td>
<td>Exclamation point</td>
</tr>
</tbody>
</table>

Quick Clicks

- Two questions

Variable Declaration Statement

- Declare your variables at the top of your script so you can find them easily
- State what variables will be used
- Computer sets aside a named area in memory for each variable
- Declare each variable only once in your program
- The declaration is a type of statement
  - 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 Statement Terminator

- A program is a list of statements
- End each statement with the statement terminator symbol
  - In JavaScript, all statements terminate with the semicolon ( ; )

Quick Clicks

- One question
Rules for Declaring Variables

- Declare every variable before it is used in the program.
- In JavaScript declaration can be anywhere in the program.
- Best practice: Place them at the top of the program.
- Declare each variable only once in the program.
- Undefined values:
  - Variable has been declared but does not yet have a value

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

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

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

### Assignment Statement

- `<variable> <assignment> <expression>`
- 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>myName</th>
<th>yourName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var yourName = &quot;Sarah&quot;;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>var myName = &quot;Andrea&quot;;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>yourName = myName;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>yourName = &quot;myName&quot;;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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>myName</th>
<th>yourName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var yourName = &quot;Sarah&quot;;</td>
<td></td>
<td>Sarah</td>
</tr>
<tr>
<td>2</td>
<td>var myName = &quot;Andrea&quot;;</td>
<td>Andrea</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>yourName = myName;</td>
<td>Andrea</td>
<td>Andrea</td>
</tr>
<tr>
<td>4</td>
<td>yourName = &quot;myName&quot;;</td>
<td>Andrea</td>
<td>Andrea</td>
</tr>
</tbody>
</table>

9/24/2008  D.A. Clements, MLIS
<table>
<thead>
<tr>
<th>Line</th>
<th>Assignment Statement</th>
<th>Value in myAge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var myage = 32;</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>myAge = myAge + 2;</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>myAge += 2;</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>myAge ++;</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>myAge -= 3;</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>myAge -- ;</td>
<td></td>
</tr>
</tbody>
</table>

Other Assignment Operators
Other Assignment Operators

<table>
<thead>
<tr>
<th>Line</th>
<th>Assignment Statement</th>
<th>Value in myAge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var myage = 32;</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>myAge = myAge + 2;</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>myAge += 2;</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>myAge ++;</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>myAge =: 3;</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>myAge -= ;</td>
<td>33</td>
</tr>
</tbody>
</table>

Calculating values in variables

EXPRESSIONS

An Expression and its Syntax

- Algebra-like formula called an expression
- 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
- Anything within parentheses is calculated first
- Within parentheses multiply and divide are performed first
- JavaScript does not have an operator for exponents

Relational Operators

- Make comparisons between numeric values
- Used in if statements and stop tests in loops
- 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
- Note: Difference between = and ==
- == compares values
- = assigns a value to a variable

Logical Operators

- To test two or more relationships at once
  - 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
More about the + operator

- Addition
  - Adds numbers
  - $4 + 5$ produces 9
- Concatenation
  - Glues strings together
    - "Your" + "five" produces "Your five"
    - "Your" + "5" produces "Your5"
    - "Your " + "five" produces "Your five"

Quick Clicks

- Two questions

Comments

- Single-line JavaScript comment
- /*Multi-line JavaScript comment continues for more than one line*/
- Comments allow you to
  - Annotate your code
  - Remind yourself what you did and why
  - Notes for yourself—or someone else—six months from now when you’re making an update!

Comments and White Space

DOCUMENTING YOUR CODE

End papers...

Eagleson's law
- Any code of your own that you haven't looked at for six or more months might as well have been written by someone else.

White Space

- White space is your friend!
  - The statements may be run together on a single line
    - Use white space to help you
      - read your code
      - understand your program
Announcements

- Read chapter 20 for Friday