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

Announcements

- Vocabulary for the week was posted in GoPost
- Reading
  - Ch 18 for today
  - Ch 21 for Friday

Basic Programming Concepts

Get with the Program

D.A. Clements

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

- 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

Currency, string, number, boolean, date/time

DATA TYPES

Strings

- The quick brown fox jumped over the lazy dog.

Strings

- 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!’ ”;

Strings

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

Strings

- 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

Numbers

- 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

What's in a name?

VARIABLES

Names, Values, and Variables

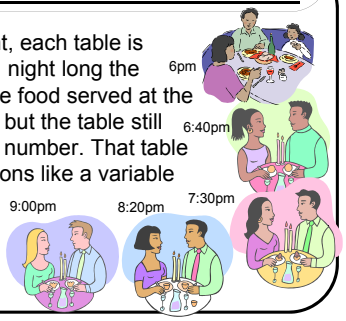
- Names in a Program Are Called *Identifiers*
- *Variables* store values and give you a handy way to refer to the current value in the variable
  - Like we say "The President" to refer to our current president
- Names Have Changing Values
  - U.S. President
    - current value is George W. Bush
    - 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

Variables

- In a restaurant, each table is numbered. All night long the people and the food served at the table change, but the table still has the same number. That table number functions like a variable name.



Quick Clicks

- One question

Identifiers and Their Rules

• Case sensitive:

HOME ≠ Home ≠ home

Valid	Invalid	Reason It's Invalid
firstone	1stOne	Begins with number
first1	first-1	JS thinks hyphen is a minus sign
firstOne	first$1	$ not allowed
first_one	first One	Space not allowed
first_1	First1!	Exclamation point

Quick Clicks

- Two questions

Example: **var home;**

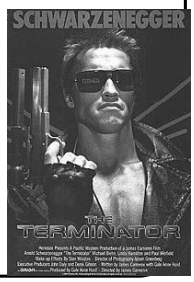
VARIABLE DECLARATIONS

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

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


All about assignment statements

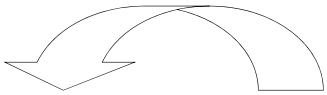
ASSIGNING VALUES TO VARIABLES

Assigning Values to Variables

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

```
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>.

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		
2	var myName = "Andrea";		
3	yourName = myName;		
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";		
3	yourName = myName;		
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	
3	yourName = myName;		
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	Sarah
3	yourName = myName;		
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	Sarah
3	yourName = myName;	Andrea	
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	Sarah
3	yourName = myName;	Andrea	Andrea
4	yourName = "myName";		

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	Sarah
3	yourName = myName;	Andrea	Andrea
4	yourName = "myName";	Andrea	

Assigning Values to Variables and Variables to Variables

We can also assign one variable to another:

Line	Assignment Statement	myName	yourName
1	var yourName = "Sarah";		Sarah
2	var myName = "Andrea";	Andrea	Sarah
3	yourName = myName;	Andrea	Andrea
4	yourName = "myName";	Andrea	myName

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	
2	myAge = myAge + 2;	
3	myAge += 2;	
4	myAge ++;	
5	myAge -= 3;	
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	
3	myAge += 2;	
4	myAge ++;	
5	myAge -= 3;	
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	34
3	myAge += 2;	
4	myAge ++;	
5	myAge -= 3;	
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	34
3	myAge += 2;	36
4	myAge ++;	
5	myAge -= 3;	
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	34
3	myAge += 2;	36
4	myAge ++;	37
5	myAge -= 3;	
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	34
3	myAge += 2;	36
4	myAge ++;	37
5	myAge -= 3;	34
6	myAge -- ;	

Other Assignment Operators

Line	Assignment Statement	Value in myAge
1	var myage = 32;	32
2	myAge = myAge + 2;	34
3	myAge += 2;	36
4	myAge ++;	37
5	myAge -= 3;	34
6	myAge -- ;	33

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
    - "four" + "five" produces "fourfive"
    - "four" + "5" produces "four5"
    - "four " + "five" produces "four five"

Quick Clicks

- Two questions

Comments and White Space

DOCUMENTING YOUR CODE

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!

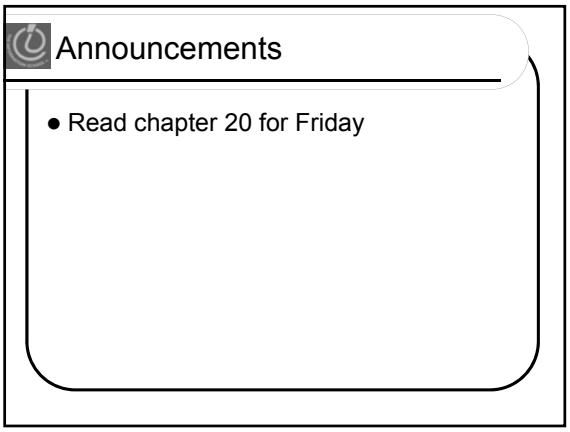
End papers…

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