Computer Science & Engineering 120
Learning to Code

Manipulating Data I

Christopher M. Bourke
cbourke@cse.unl.edu

Part I: Variables

Topic Overview

Variables

- Declaring and Using Variables in JavaScript
- Basic Syntax and Assignment
- Variable Naming Rules and Conventions
- Variable Types (review)
- Variable Scoping

Variables

- A *variable* is a way to *store data*
- Similar to a mathematic variable: x
- A variable has a *name* ("identifier") and can hold a *value*
- Similar to a key-value pair as in JSON

Declaring & Using Variables

- A variable can be created by using the keyword `var`
- You then provide a variable name
- Optionally, you can *assign* it an initial value using the *assignment operator*, a single equals sign, `=` followed by a value (called a *literal*)
- End the variable declaration with a semicolon `;`

Example:

```
var pi = 3.14;
```

Assignment means that the right-hand-side (an *expression*) is placed into the variable on the left-hand-side.

Printing Variables

- You can "print" messages and variables to the console using: `console.log()`
- In LightTable, the console is at the bottom (View → Console)

Declaring & Using Variables

Demo

```
//a variable declaration without an assignment
var x;
//a variable declaration with an integer assignment
var age = 21;
//a variable declaration with a floating point
// number assignment
var interestRate = 0.05;
//a variable declaration with a string assignment
var firstName = "Chris";
```

Naming Rules

Rules

In JavaScript, variable names must follow several rules:

1. May contain upper and lowercase letters, digits, underscores and dollar signs:
   `a-z, A-Z, 0-9, _, $`
2. May *not* begin with a digit
3. Cannot use *reserved words*[1] such as:
   `var, for, function`
4. Variable names are *case sensitive*, the following are all distinct:
   `total, Total, TOTAL`

[1] Complete List: `http://www.w3schools.com/js/js_reserved.asp`

Naming Conventions

- Avoid using underscore and dollar sign (`_, $`)
- Use lower camel casing: multiword variable names have:
  - First word lowercase
  - First letter of each subsequent word is uppercase
- Use descriptive names (code should be "self-documenting")

Naming Rules & Convention

Demo

Illegal variable names:

```
var interest rate;
var 95theses;
var for;
```

Bad variable names:

```
var variable01;
var $foo;
var myVariable;
var nUmBERofStudeNTs;
```

Good variable names:

```
var average;
var subTotal;
var grandTotal;
var taxRate;
```

Variable Types I

Recall that JavaScript has several variable types:

- Numeric types (integers and floating point)
- Strings
- Booleans
- Objects
- Arrays

Variable Types II

JavaScript is a *dynamically typed* language:

- A variable's type is determined by the value assigned to it
- Can change if the value changes
- Not set for the life of the variable
- The `typeof` keyword can be used to determine a variable's type

Variable Types

Demo

```
var numberOfStudents = 30;
var pi = 3.14;
var lastName = "Bourke";
var isGrad = true;
var student = {
  "lastName": "Smith",
  "firstName": "John",
  "gpa": 4.0
};
var arr = [10, 20, 30];
arr = ["Hello", "World", "!"];
```

Variable Scoping

The *scope* of a variable is the part of a program in which a variable is valid ("in scope").

- A variable declared inside a function (later) is *local* to that function
- A variable declared outside any function is *global*
- Omission of `var` makes a variable global (avoid)

Variable Scoping

Demo

```
function foo() {
  var x = 10;
  console.log(x);
}

//x does not exist here:
console.log(x); //error!
```

Part II: Operators

Topic Overview

Operators

- Assignment Operator (review)
- Arithmetic Operators & The Math Library
- String Concatenation
- Dot Accessor
- Array Indexing

Operators

Operators are symbols that can be combined with variables to produce new results.

- Assignment Operator:
  `var pi = 3.14;`
- Arithmetic Operators:
  - Addition: `+`;
  - Subtraction: `-`;
  - Multiplication: `*`;
  - Division: `/`;
- May be combined with either variables or literals
- Same *order of precedence* as in algebra
- May be combined with parentheses to change order of evaluation

Arithmetic Operators I

Demo

```
var a = 10;
var b = 20;
var c = 2.5;
var x;
x = a + b;
console.log(x); //30
x = a + 50;
console.log(x); //60
```

Arithmetic Operators II

Demo

```
x = a - b;
console.log(x); //-10
x = a * c;
console.log(x); //25
x = a / b;
console.log(x); //0.5
x = a + b + c;
console.log(x); //32.5
x = a * b + c;
console.log(x); //202.5
x = a * (b + c);
console.log(x); //225
```

Math Library

- The Math library provides many useful constants and "calculator" functions for you to use
- Can access the functionality by using `Math.`
- Constants:
  `Math.PI, Math.E`
- Functions:
  `Math.sqrt(), Math.sin(), Math.pow()`
- Complete list: `http://www.w3schools.com/js/js_math.asp`
- Input is provided inside the parentheses
- Output can be used in larger expressions

Math Library

Demo

```
var radius = 1.5;
var area = Math.PI * radius * radius;
//or:
area = Math.PI * Math.pow(radius, 2);
var a = 10;
var b = 20;
var c = 30;
var root = -b + Math.sqrt(b*b - 4*a*c) / (2*a);
var result = Math.sin(Math.PI);
```

String Concatenation

- String *concatenation* is the operation of combining two strings by appending one to the end of another
- Two strings can be concatenated using the concatenation operator: `+`
  `"Java" + "Script"` → `"JavaScript"`
- Same symbol as addition, but when used with strings, it is interpreted as concatenation
- You can mix numbers and strings to convert the numbers to strings

String Concatenation

Demo

```
var lastName = "Bourke";
var firstName = "Chris";
var name = lastName + ", " + firstName;
console.log(name);

var x = 10;
var message = "The value of x is " + x;
console.log(message);

console.log("x = " + x);
```

Dot Accessor

- The *dot accessor* allows you to access or modify an attribute of an object
- Syntax: `a.key`

Dot Accessor

Demo

```
var s = {
  "lastName": "Smith",
  "firstName": "John",
  "gpa": 4.0
};
var name = s.lastName + ", " + s.firstName;
console.log(name);

s.gpa = 3.5;

//you can even "add" new attributes:
s.minor = "Software Development";
```

Array Indexing

- Elements of an array can be accessed or modified by *indexing* them
- Syntax: `arr[i]`,
- The first element in an array is indexed by zero
- The second element is at index 1, etc.
- The last element is at index `arr.length - 1`
- Attempting to access invalid indices will result in `undefined`

Array Indexing

Demo

```
var arr = [5, 9, 42, 2, 8, 4, 4, 3];
console.log(arr[0]);  //5
console.log(arr[1]);  //9
console.log(arr[5]);  //4
console.log(arr[7]);  //3
console.log(arr[arr.length-1]);  //3

console.log(arr[8]); //undefined

arr[7] = 6;
console.log(arr[7]);  //6
arr[8] = 50;
console.log(arr[8]); //50
```

Part III: Exercise

Exercise

Consider a sphere of radius r. The *volume* of the sphere is given by the formula:

$$V = \frac{4}{3}\pi r^3$$

while the surface area of the sphere is given by:

$$A = 4\pi r^2$$

Write code to store a value of a radius into a variable and compute the volume and surface area.