Simple JavaScript

Dr. Stephen Blythe
CSC10022, Fall 2020

JavaScript & Statements

- JavaScript code is place inside of a <script> … </script> tag pair
  - outside of an HTML <body> tag
  - will ***not*** be displayed as-is on a web page
- example (don't worry about the <script> contents yet)

```
…
<body>
some <b>HTML</b>
</body>
<script type=text/javascript>
  alert("Hello, World");
</script>
```

- note the *type=text/javascript* option
  - identifies the script as containing JavaScript code.

JavaScript & Statements

JavaScript code is a sequence of statements

- each statement performs a specific task
- each “standalone” statement is ended with a semicolon
- a sequence of statements is performed in the order listed

The alert() statement (see previous slide), when performed:

- “pops up” a new dialog box (details are web browser specific)
- Inside of this box, prints the message “string” found inside of the parentheses
- presents an “OK” button and waits for the user to click that button

Example of alert() Calls

Consider the following example:

```
<script type=text/javascript>
  alert("Hello, World");
  alert("It is a fine day!");
  alert("Goodbye & Goodnight!");
</script>
```

- note that you get three separate dialog boxes
- in the order they are listed
- each alert() "call" ends in a semicolon
- Note that alert is a "built-in" JavaScript **function**:
  - In programming, a function performs some task
  - ***not*** necessarily mathematical in nature!!!!

External Scripts

- Most JavaScript Programmers place code in other file(s)
  - script tag(s) are used to specify the other files
  - use the src option (attribute)
- For example:

```
<script src=otherfile.js type=text/javascript>
</script>
```

- Note that you have to actually create otherfile.js
  - of course, you can name the file anything you like, so long as it ends in .js
- You can have as many script tags as you like …

Output on Browser Console

Consider the following example:

```
<script type=text/javascript>
  console.log("Hello, World");
  alert("It is a fine day!");
  console.log("<center>Goodbye & Goodnight!</center>");
</script>
```

- the `console.log(..)` appends the given "string" to the console
  - for now, think of `console.log()` as another built-in function.
  - technically, it is the "object" `console` …
  - … with the "method" `log` being applied to that object.
- note that you can mix and match statements as you like!

Joining Strings

Consider the following example:

```
<script type=text/javascript>
  alert(“Today is ” + “ a good day ” + “to die”);
  console.log(“Klingons always” + “ were a bit fatalistic<br>”);
  console.log(“2+2 =”  +  2+2  + “, or better yet  = ” + (2+2) );
</script>
```

- you can merge things you want to print with a + sign
  - works with numbers, too
    - if you are careful. *Very careful!!*
    - computers calculate +’s from left to right unless parenthesized!

JavaScript Comments

There are two kinds of comments in JavaScript:

- “in-line” comments: anything from // to the end of a line
- “general” comments: anything found between a /* and */

```
<script type=text/javascript>
  alert(“Today is ” + “ a good day ” + “to die”); // give user a pop-up klingon proverb
  /* this is an interesting point … but maybe we should let the web page reader
     know where this came from; and what we think of it … */
  console.log(“Klingons always” + “ were a bit fatalistic”); //show in console
</script>
```

- Note that you cannot use HTML-style comments in JavaScript!

Variables in JavaScript

A *variable* is essentially a container that you give a permanent name to

- a variable name must:
  - start with a letter *and*
  - be followed by 0 or more letters,digits, or underscores (_)

Valid Name Examples	**Invalid Names**
myValue	my Value
SecondCourse	2ndCourse
Year1994	alert
under_scored	Hyphen-ated
atsign	@sign

Variables in JavaScript ...

Variables can be used to store values:

- `radius = 34; // stores 34 in the variable named radius`
- `piApprox = 4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + 4/13;`
  - note that you can do arithmetic *on the right hand side*
- `circumference = 2*piApprox*radius;`
  - note that you can plug in variables on the right hand side, too

- The above are ***NOT*** mathematical formulas!!!! (see next slide)

- also, remember that these are done in listed order
  - make sure that you give a variable a value before it is used!

It is a good idea (but not required) to declare variables *before* using them

- use the **var** statement, which is followed by a list of variable names

  - `var radius, circumference, piApprox; // note the semicolon!`

The Assignment Statement

The '=' in the previous slide does ***NOT*** define mathematical equality!

- '=' indicates an assignment statement in JavaScript.
- when an assignment statement is encountered, JavaScript:

  1. *ignoring the left hand side*, evaluates the right hand side
  2. stores the result of step 1 into the variable on the left hand side.

- this means the left hand side *must* be a single variable name!!!

Consider this:

```
<script type=text/javascript>
  var piApprox, radius, circumference;
  piApprox = 3.14; // for ease of "hand calculation" below
  radius = 1;
  circumference = 2 * piApprox * radius;
  console.log("For r="   + radius +   ", C="    +   circumference);
  radius= 1000; //do not place commas in numbers !!!!
  console.log("For r="+radius+", C="+circumference); //what did you expect?
</script>
```

Arithmetic Operators

The following may be used in the right side of an assignment statement:

Operator	Type	Example(s)
-	unary	y = -2; z = -x;
-	binary	s=5-2; // 3 x=3-z;
+	binary	a=61+3.1; //64.1 ops = spc + avg;
*	binary	m=3*91; //273 miles=mpg*gallons;
/	binary	d=18/2; //9 mpg=miles/gallons;
%	binary	r = 17%5; //2 after25=change%25

They can be mixed and combined with normal math precedence.

JavaScript That Reads Input

Consider the following example:

```
<script type=text/javascript>
  var answer;
  answer = prompt("Tell Me Something");
  console.log("It is very Interesting that: ");
  console.log(answer);
</script>
```

- the `prompt()` function:
  - pops up a dialog box, *and*
  - lets the user type something in, *and*
  - "returns" what was typed in
- note that you can mix and match statements as you like!

Putting It All Together

```
<script type=text/javascript>
  var piApprox, radius, circumference;
  piApprox = 4*Math.atan(1);                    // reasonable trigonometric estimate of pi
  radius = prompt("Enter a radius value");     // get user input
  circumference = 2 * piApprox * radius;       // calculate circumference
  console.log("For r="+radius+", C="+circumference);  // print circumference
</script>
```

Small subtlety:

- prompt() returns a *string* (sequence of type-able characters)
- the above code uses this string as if it is a *number*!
  - JavaScript converts between strings and numbers when needed
    - sometimes "forcing" this to happen with Number() is required

```
var data = prompt("enter a number"); //saves a string in data
var value = Number(data);       //saves equivalent number in value
```