Tutorial 9 Notes

JavaScript is a client-side interpreted programming language

Client-side programs run on the user’s computer using downloaded scripts with HTML and CSS files

Methods for including JavaScript code (slide 9)

* External file with .js extension <script src=”url”></script>
* Embedded script anywhere in HTML program <script>code</script>

Modifying process sequence (slides 10 and 11)

* When a browser encounters a script, it immediately stops loading the page and begins loading and then processing the script commands
* async attribute tells a browser to parse the HTML and JavaScript code together
* defer attribute defers script processing until after the page has been completely parsed and loaded
* async and defer attributes are ignored for embedded scripts

Comments (slide 14)

* one-line comment use // comment
* multiple line comments use /\* comments \*/

Applying strict usage (slide 23)

* Strict mode enables all lapses in syntax to result in load-time or run-time errors
* To run a script in strict mode, add the following the first line of the file: “use strict”;

Objects and JavaScript statement format

* JavaScript is an object-oriented programming language
* Objects have properties and methods
* document.write(“text”); is interpreted as
* object.method(argument);

JavaScript object types (slide 25)

* Built-in objects – intrinsic to JavaScript language
* Browser objects – part of browser
* Document objects – part of web document
* Customized objects – created by a programmer to use in an application

Browser object model (BOM) and document object model (DOM)

* Organize browser and document objects in hierarchical structures, respectively
* Object hierarchy shown in slide 26
* Document object collections (slides 28 and 29)

Referencing an object by ID or name

* document.getElementById(id)

Object properties are accessed using statements in the format object.property

Object methods are applied using the expression object.method(argument)

Methods may require more than one parameter argument/value

Object reference to write HTML code

* HTML code stored within a page element is referenced using element.innerHTML
* To return the HTML code within element as well as the HTML code of element itself, use element.outerHTML
* properties and methods to insert content (slide 36)

Variables (slides 37-41)

* Variable declaration statement var variable = value;
* Rules for valid variable names (slide 38)
* Data types – numeric value, text string, Boolean (true/false) value, object, null value

Date objects

* var Today = new Date();
* Empty Date() argument stores current date, can include data to declare a specific date value
* A date include all components of the date and the time
* Date object methods (slide 43)
* Example using toLocaleDateString() and toLocaleTimeString() (slide 44)
* Setting date and time values (slide 45)

Assignment operators (slide 48)

Math objects (slides 49 and 50)

JavaScript functions (slides 54 to 57)

function function\_name(parameters){

 commands

}

* function\_name is the name of the function
* parameters are a comma-separated list of variables used in the function
* commands are the set of statements run by the function

Running timed commands (slides 59 and 60)

* setTimeout(“command”, delay);
* setInterval(“command”, interval);

Converting between numbers and strings (slides 66 to 68)

* testNumber = 123; // numeric value
* testString = testNumber + “”; // text string

**Tulsa’s New Year’s Eve Bash** (tny\_clock.html and tny\_script.js programs)