Announcements

- Resources
  - https://www.javascript.com
  - https://www.javascript.com/resources
  - Link checker - https://validator.w3.org/checklink
  - http://jsforcats.com/
  - http://www.htmldog.com/
- Tools
  - https://developer.mozilla.org/en-US/docs/Tools

JavaScript

- **JavaScript** → programming language that can appear in html pages
- It allow us to:
  - To dynamically create web pages
  - To control a browser application
    - Open and create new browser windows
    - Download and display contents of any URL
  - To interact with the user
  - Ability to interact with HTML forms
    - Process values provided by checkbox, text, buttons, etc.
- **Example:** SqrTable.html

JavaScript

- JavaScript implements ECMAScript
  - ECMAScript specification
  - http://www.ecma-international.org/ecma-262/6.0/ECMA-262.pdf
- ECMAScript
  - Web browsers are one host environment where it may exist
- ActionScript also implements ECMAScript
- JavaScript is more than ECMAScript
- JavaScript implementation includes
  - ECMAScript
  - DOM (Document Object Model)
  - BOM (Brower Object Model)
- Browser support table at
  - http://en.wikipedia.org/wiki/ECMAScript

JavaScript

- JavaScript engine → Process JavaScript code
  - Safari → JavaScriptCore
  - Chrome → V8
  - Firefox → Spidermonkey
- To write JavaScript programs you need
  - A web browser
  - A text editor
- A JavaScript program can appear
  - In a file by itself typically named with the extension **.js**
  - In html files between a <script> and </script> tags.
- Client-Side JavaScript → the result of embedding a JavaScript engine in a web browser
- Template for JavaScript Programs
- **Example:** TemplateJS.html

Processing HTML Page with JS

- **DOM – Document Object Model**
  - Structured representation of the HTML page
  - Every HTML element is represented as a node
  - Browser uses HTML to build the DOM and can fix problems with the HTML so a valid DOM is generated
- **Lifecycle**
  - **Set the user interface**
    - Parse the HTML and build the DOM
    - Process (execute) JavaScript code
  - **Enter a loop and wait for events to take place**
- When JavaScript is seen in a page, the DOM construction is halted and JavaScript code execution is started.
- JavaScript can modify the DOM (e.g., creating / modifying nodes)
  - One reason why <script></script> elements appear at the bottom of a page is to guarantee elements elements JavaScript manipulates have already been created

Event-Handling

- Relies on a single-threaded execution model
- An **event queue** keeps track of events that have taken place, but have not been processed (event-handler function for the event has not been called)
- All generated events (whether are user-generated or not) are placed in the event queue in the order they were detected by the browser
    - The browser mechanism that detects events and that adds them to the event queue is separate from the thread that is handling the events
- Browser periodically checks the event queue and if any event is found it executes the appropriate handler (if one was defined)

Browser’s Global Objects

- Browser’s provides two global objects: **window** and **document**
- **window** object – represents the window in which a page resides
  - Provides access to other global objects (e.g., document)
  - Keeps track of user’s global variables
  - Provides to JavaScript access to Browser’s APIs
- **document** object
  - Property of the window object that represents the DOM of the current page
  - Via this object you can access / modify the DOM

Types of JavaScript Code

- **Function Code**
  - Code contained in a function
- **Global Code**
  - Code placed outside all functions
  - Automatically executed by JS engine
- As in Java, a stack is used to keep track of function calls. Each function call generates a **function execution context** (stack frame)
- There is one frame called the **global execution context** created when the JS program starts executing. There is only one global execution context (at the bottom of the stack)

JavaScript Comments

- Comments in JavaScript
    - Used to provide information to the programmer
    - Used to identify sections in your code
    - Ignored by the JavaScript interpreter
- Two types of comments
    - Inline comment // This is a comment until the end of the line
    - Block comment

```
/* The following is a comment
 that spans several lines */
```

Variable Declarations

- Variable declaration (no type specification) **var x;**
- Variables names must start with a letter, underscore or dollar sign and can be followed by any number of letters, underscores, dollar signs or digits

JavaScript Data Types

- JavaScript has no class concept
- Two kinds of types:
  - Primitive types – simple data stored as is
  - Reference types – references to locations in memory
- Primitive data types in JavaScript
  - **Null** – has value null
  - **Boolean** – values true or false
  - **Number** – numeric value
  - **String** – character sequence delimited by single or double quotes
  - **Undefined** – has as value undefined (values associated with variables that are not initialized)
- typeof operator
  - Returns string indicating the type of data
  - Note: typeof null → returns "object"

JavaScript Data Types

- Reference types represents objects in JavaScript
- Reference values are instances of reference types and considered objects
- Object – collection of properties
  - Property – string that is associated with a value
  - Value – could be a primitive, object, function
- Object creation

```
var myFirstObject = new Object();
var mySecondObject = {
    id: 789,
    name:  "Rose Smith"
}; // object literal
```

- JavaScript relies on garbage collection
  - When an object is no longer needed set the variable to null

Conversions

- In JavaScript you don’t specify the type of variables
- Most of the time implicit transformations will take care of transforming a value to the expected one
- Example:

```
var age = 10;
var s = “John Age: “ + age;
```

- Mechanism to transform values:
  - **Converting number to string**
    ```
    var stringValue = String(number);
    ```
  - **Converting string to number**
    ```
    var number = Number(stringValue);
    var number = parseInt(stringValue);
    var number = parseFloat(stringValue)
    ```

JavaScript (Comparisons)

- You can compare values by using the following operators
  - === → Return true if the values are equal, false otherwise (e.g., x === y)
  - !== → Returns true if the values are different, false otherwise (e.g., x != y)
  - == and != → Not as strict as previous equality operators
  - Relational Operators
    - < → Less than
    - > → Greater than
    - <= → Less than or equal
    - >= → Greater than or equal

JavaScript (Dialog Boxes)

- We can perform input and output via dialog boxes
- Input via ***prompt***
- **Example:** InputOutput.html
  - Notice we can define several variables at the same time
  - ***prompt*** is a function that displays a dialog box with the specified title. It can be used to read any data
  - You can read numbers and strings via prompt
- **prompt** → returns a string
- If you need to perform some mathematical computation you might need to explicitly convert the value read into a number
- **alert** → used to display a messages in a dialog box
- **Example:** Network.html

JavaScript

- Constructs having same syntax /semantic similar to Java
  - while, do while, for loops
  - if statement
  - cascaded if statements
  - break statement
  - switch statement
- **Example:** SqrTable.html

Strict Mode

- Allows for error checking both globally or within a function
- Use the strict mode pragma
  - “use strict”;
- If pragma used outside of a function it applies to all the script
- It can appear in a function

```
function computeAvg() {
        “use strict”;
}
```

- **Example:** Strict.html
  - We need to use var
  - Cannot use reserved words (interface, package, private, …)

References

- **The Principles of Object-Oriented JavaScript** by Nicholas C. Zakas
  - **ISBN:** 978-1-59327-540-2
- **Secrets of the JavaScript Ninja**, Second Edition, by John Resig, Bear Bibeault, Josip Maras
  - **ISBN-13:** 978-1617292859