Announcements

- Resources
  - https://www.javascript.com
  - https://www.javascript.com/resources

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 → language defined in the ECMA-262 standard
  - http://www.ecma-international.org/ecma-262/5.1/
  - Web browsers are one host environment where it may exist
- ActionScript also implements ECMAScript
- Different version of ECMAScript are known as editions
  - Most recent is edition 5
- 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 Interpreter → Process JavaScript code
- Chrome V8 Engine
  - http://code.google.com/p/v8/
- 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 interpreter in a web browser
- Template for JavaScript Programs
- **Example:** TemplateJS.html

Execution of JavaScript Programs

- HTML parser → Takes care of processing an html document
- JavaScript interpreter → Takes care of processing JavaScript code.
- HTML parser → must stop processing an html file when JavaScript code is found (JavaScript interpreter will then be running)
  - This implies a page with JavaScript code that is computationally intensive can take a long time to load

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 */
```

JavaScript Data Types

- Primitive data types in JavaScript
  - Undefined
  - Null
  - Boolean
  - Number (to represent both integer and floating-point)
  - String
- Complex Data Type
  - Object
- 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
- All numbers are represented as floating-point values
- To represent a single character use a string of length 1
- You can use ' ' or " " for strings

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)
    ```

Boolean Type

- We have seen integer, float, and string values
- Another type: boolean type
- Assumes the value *true* or *false*
- Variable declaration and initialization

```
var  found = true;
var  attending = false;
```

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

Functions

- About Functions
  - They are objects
  - There are several ways to define them
  - For now we will use function declarations
- Function Declaration

function name (<comma-separated list of parameters>) {
statements
}

- Functions are invoked by using the () operator
- We don't use ***var*** for parameters (e.g. function print(x, y))
- Parameters are passed by value
- Order of declaration is immaterial
- There is no main function like in other languages
- Returning values via return
- Name of a function is a reference to the function object

Server Side Includes

- File Inclusion (following example includes footer.html)

  <!--#include virtual="footer.html" -->
- Timestamp for file modification

  <!--#echo var="LAST_MODIFIED" -->
- Timestamp for local date/time

  <!--#echo var="DATE_LOCAL" -->
- **Example:** serverSideIncludeExample