EAST WEST UNIVERSITY

CSE479
Web Programming

Nishat Tasnim Niloy

Lecturer

Department of Computer Science and Engineering

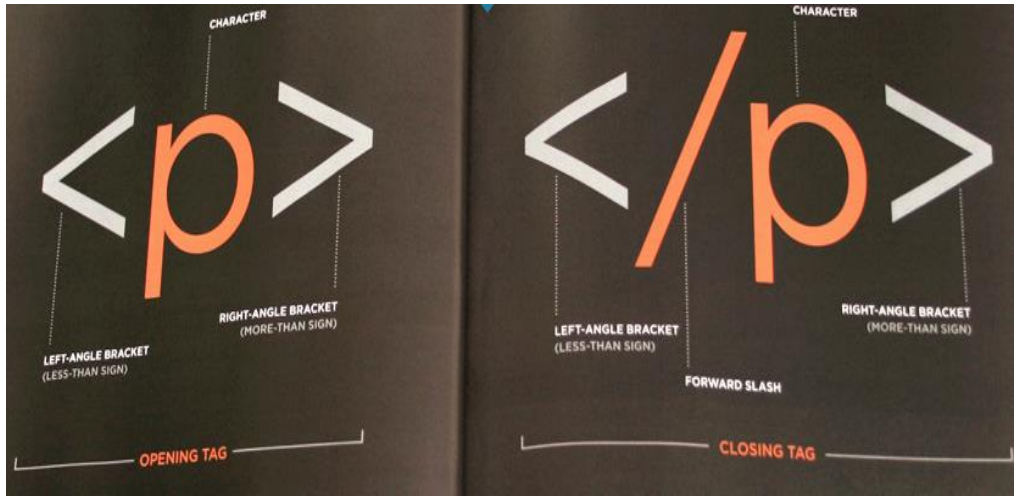
Faculty of Science and Engineering

Topic 3

JavaScript
(Adding behavior to a website)

By the end of this unit you should be able to...

- Explain how JavaScript makes web pages interactive
- Create scripts for a web page with JavaScript
- Discuss how computers fit in the world around them
- Explore various JavaScript instructions like
  - Statements
  - Comments
  - Variables
  - Data types
  - Operators



How JavaScript makes web pages interactive

Access content:

Can use JavaScript to access any element, attribute, or text from an HTML page

Modify content:

Can use JavaScript to add and/or remove elements, attributes, text from an HTML page

Program rules:

Can use JavaScript to specify a set of steps for the browser to follow in order to access or modify content on a page

React to events:

Can use JavaScript to specify that a script should run when a specific event occurs

Writing a script

A series of instructions that a computer can follow to achieve a goal

Like a

- Recipe -- instructions for a cook to follow
- Handbook with procedures to follow in certain situations
- Manual, e.g., car repair manual

Writing a script:

Define the goal:

Define the task you want to achieve

Design the script:

Split goal into series of small tasks involved in achieving it

Code each step:

Implement each step of a task in JavaScript

How computers fit in the world around them

Computers create models of the world using data

Each physical thing or concept can be represented as an object

Each object can have its own:

- **Properties** - name/value pair for each characteristic
- **Methods** - what we can do with an object, code named after verbs
- **Events** - ways we interact with objects, can change property values (using methods).
  - A computer's way of notifying us that something just happened, e.g., ***accelerate event*** when driver presses pedal
  - Event triggers method to respond to what just happened

Together, these create a working model of the object

More on JavaScript overview

- A lightweight programming language ("scripting language")
- Interpreted and run by the browser and Node.js
- Used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - react to events (ex: page load, user click)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)

Not related to Java

A web standard (but not supported identically by all browsers)

NOT related to Java other than by name and some syntactic similarities

From Secrets of the JavaScript Ninja, by John Resig and Bear Bibeault:

Hamburgers and Ham are both foods that are meat products, just as JavaScript and Java are both programming languages with a C-influenced syntax. But other than that, they do not have much in common and are fundamentally different right down to their DNA.

JavaScript different from Java

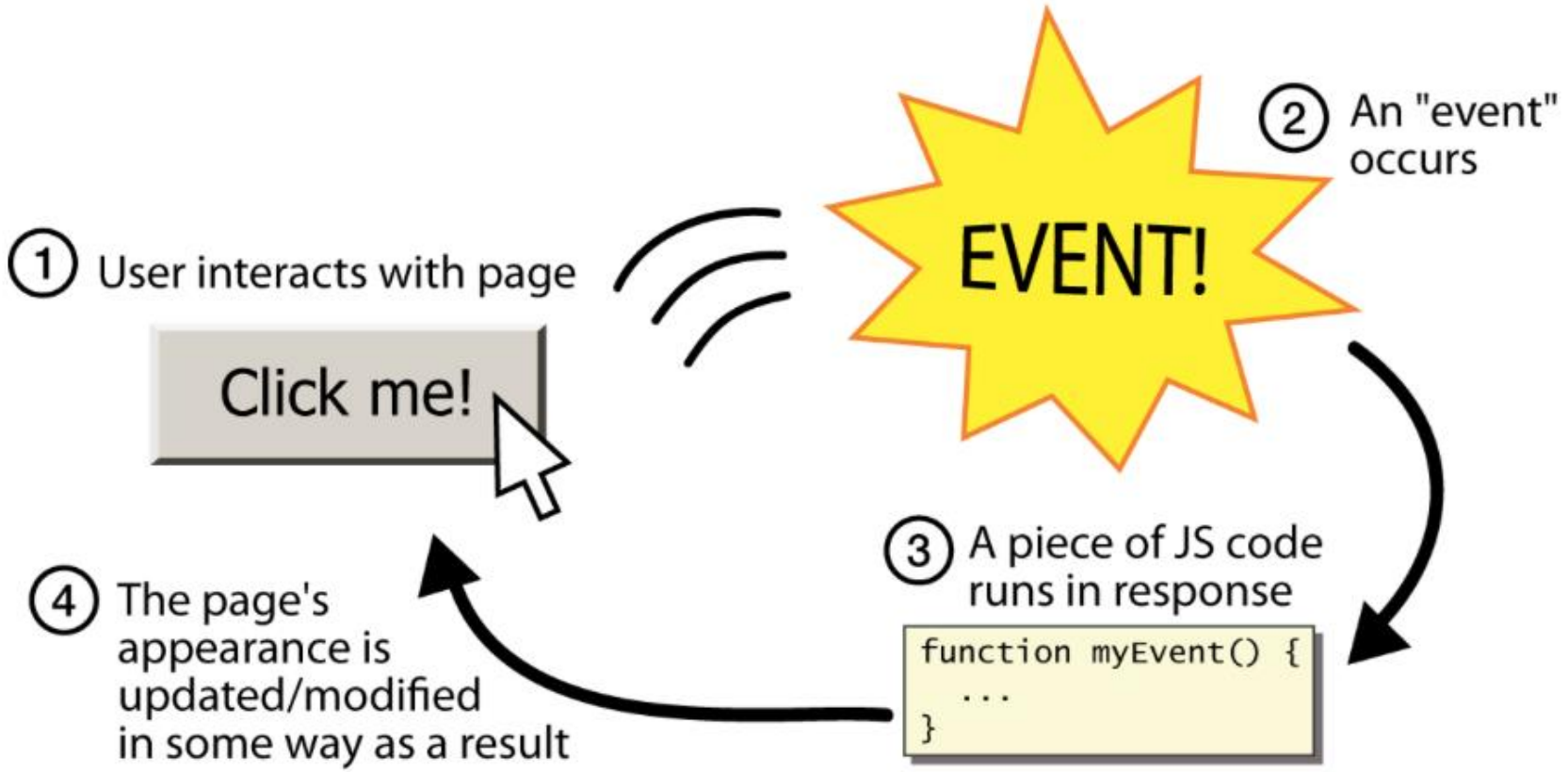
- **Interpreted**, not compiled (Java can be compiled)
- More relaxed syntax and rules
  - Fewer and "looser" data types
  - Errors often silent (few exceptions)
- Key construct is the **function** rather than the class
- "First-class" functions are used in many situations
- Integrates with a page's HTML content and CSS

JavaScript promotes event-driven programming

JavaScript programs have **no main** methods/function

They **trigger events** in response to user actions

Event-driven programming: writing programs that are driven by user events



Web browsers built using objects

Web browser create model of:

- **The HTML page it is showing**
  - ★ The current page is represented as a ***document*** object
  - ★ The ***title*** property is one of several properties contained in the document object.
  - ★ Using the document object, you can:
    - ○ Access and change what content users see on the page
    - ○ Respond to how they interact with the page

- **The window in which the page is displayed**
  - ★ Each window or tab is represented as a ***window*** object
  - ★ A property of the window object is ***location***

Browser’s view of a web page

Need to understand how a browser interprets the HTML code and apply styling to it

Browser receives page as HTML code

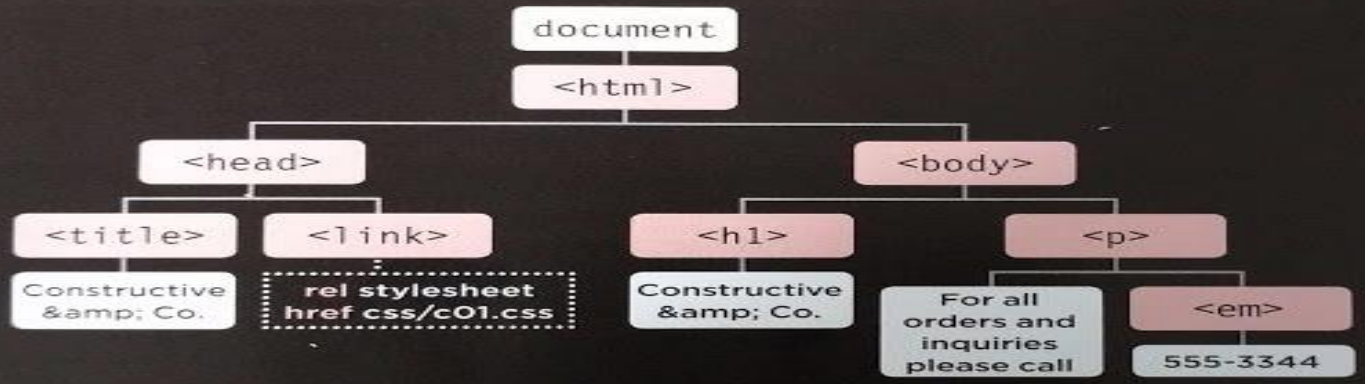
Browser creates a model of the page, stores it in memory -- tree of nodes (object for each element)

Browser uses a rendering engine to show the page on the screen -- engine processes CSS and applies each declaration to corresponding element

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive &amp;amp; Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive &amp;amp; Co.</h1>
    <p>For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>
```

1

The browser receives an HTML page.



2

It creates a model of the page and stores it in memory.

- OBJECT
- ELEMENT
- TEXT
- ATTRIBUTES

3

It shows the page on screen using a rendering engine.

Separation of concerns and progressive enhancements

Three languages that are used to build web pages-build on top of each other

HTML

- ***content layer***: first layer of code that hold content and gives structure and adds semantics

CSS:

- ***presentation layer***: second layer of code that enhances the HTML page with rules that control presentation of content

JavaScript

- ***behavior layer***: third layer of code that can change how the page behaves and add interactivity

Each should be kept in separate files

Including JavaScript in a page

http://javascriptbook.com/code/c01/add-content.html

Use the HTML ***<script>*** element to include JavaScript in a page

Inlining JavaScript:
<script> document.write('<h3>Hello dear! </h3>'); </script>

Should avoid the above: JavaScript code belongs in its own file

Link external JavaScript file:
<script src="js/my-script.js"></script>

This is a better option: separation of concerns.

Put JavaScript code in its own folder.

JavaScript runs where it is found in the HTML

http://javascriptbook.com/code/c01/add-content.html

When the browser finds a ***<script>*** element,

- ❑ it stops to load the script
- ❑ It checks to see if it needs to do anything
- ❑ It may need to run the code in the script

In the add-content example, if the script element is moved below the first <p> element, this affects where the greeting is written into the page.

Comments in JavaScript

// single-line comment

/ multi-line comment */*

Identical to Java's comment syntax

Recall: 3 comment syntaxes

- *HTML:* *<!-- comment -->*
- *CSS/JS/Java:* */* comment */*
- *Java/JS:* *// comment*

Variables and types in JavaScript

*let **name** = **expression**;*

```
let today = new Date();
let hourNow = today.getHours();
let greeting = 'Welcome!';
```

- ★ Declared with the ***let*** keyword (case sensitive)
- ★ Types are **not** specified, but JavaScript has types (loosely typed language)
  - ○ Number, String, Boolean, Array, Object, Function, Null, Undefined
  - ○ Use **typeof** to find a variable's type

Number type

```
let enrollment = 99;
let medianGrade = 2.8;
let credits = 5 + 4 + (2 * 3);
```

- Integers and real numbers are the same type (no ***int*** vs. ***double***)
- same operators: ***+, -, *, /, %, ++, --, =, +=, -=, *=, /=, %=***
- similar precedence to Java
- many operators **auto-convert** types: ***"2" * 3*** is ***6***

String type

```
let s = "Connie Client";
let fName = s.substring(0, s.indexOf(" "));      // "Connie"
let len = s.length;                                          // 13
let s2 = 'Melvin Merchant';                                  // can use "" or ' '
```

Methods: http://www.w3schools.com/jsref/jsref_obj_string.asp

- ***charAt***, ***charCodeAt***, ***fromCharCode***, ***indexOf***, ***lastIndexOf***, ***replace***, ***split***, ***substring***, ***toLowerCase***, ***toUpperCase***

- ***charAt*** returns a one-letter ***String*** (there is no ***char*** type)

- concatenation with **+** e.g., 1 + 1 is 2, but "1" + 1 is "11"

- ***length*** property (not a method as in Java)

More about String

Escape sequences behave as in Java: **\' \" \& \n \t **

Convert between **Number** and **String**:

```
const count = 10;
let s1 = "" + count;                                  // "10"
let s2 = count + " bananas, ah ah ah!";   // "10 bananas, ah ah ah!"
let n1 = parseInt("42 is the answer");            // 42
let n2 = parseFloat("booyah");            // NaN
```

Access characters of a **String**, use [index] or ***charAt***:

```
const firstLetter = s[0];
const firstLetter = s.charAt(0);
const lastLetter = s.charAt(s.length - 1);
```

Null and undefined -- special values

```
const ned = null;
const benson = 9;
let caroline;

//   at this point in the code,
//   ned is null
//   benson is 9
//   caroline is undefined
```

- **null**: variable exists, but was assigned an empty or ***null*** value
- **undefined**: does not exist OR has not been assigned a value

Simple functions in JavaScript

```
function name(params) {
    statement ;
    statement ;
    ...
    statement ;
}

function myFunction(value) {
    console.log("Hello! " + value);
    console.log("How are you?");
}
```

- ★ Statements placed in functions can be evaluated in response to user events
- ★ A way to get output for debugging:

```
console.log("Function started");
```

Boolean type

```
let iLike190M = true;
let ieIsGood = "IE6" > 0;   // false
if ("web dev is great") {  /* true */ }
if (0) {  /* false */ }
```

Any value can be used as a **Boolean**

- **"falsey"** values: false, 0, 0.0, NaN, "", null, and undefined
- **"truthy"** values: anything else

converting a value into a **Boolean** explicitly:

- let boolValue = Boolean(otherValue);
- let boolValue = !!(otherValue);

Array type

```
const name = [];                                    // empty array
const name = [value, value, ..., value];            // pre-filled
name[index] = value;                                // store element
const name = new Array(‘value1’, value2)     // using constructor function


const ducks = ["Huey", "Dewey", "Louie"];

const stooges = [];       // stooges.length is 0
stooges[0] = "Larry";     // stooges.length is 1
stooges[1] = "Moe";       // stooges.length is 2
stooges[4] = "Curly";     // stooges.length is 5
```

More on Array type

- Two ways to initialize an array
- **length** property (grows as needed when elements are added)
- **Array** serves as many data structures: **list**, **queue**, **stack**, ...

Array methods

```
const arr = ["Stef", "Jason"];   // Stef, Jason
arr.push("Brian");               // Stef, Jason, Brian
arr.unshift("Kelly");            // Kelly, Stef, Jason, Brian
arr.pop();                  // Kelly, Stef, Jason
arr.shift();                     // Stef, Jason
arr.sort();                      // Jason, Stef
```

Methods: http://www.w3schools.com/jsref/jsref_obj_array.asp

- ★ **concat**, **join**, **pop**, **push**, **reverse**, **shift**, **slice**, **sort**, **splice**, **toString**, **unshift**, **filter**
- ★ **push** and **pop** add / remove from back
- ★ **unshift** and **shift** add / remove from front
- ★ **shift** and **pop** return the element that is removed

Splitting String: split and join

```
let s = "the quick brown fox";
let a = s.split(" ");                    // ["the", "quick", "brown", "fox"]
a.reverse();                        // ["fox", "brown", "quick", "the"]
s = a.join("!");                         // "fox!brown!quick!the"
```

- **split** breaks apart a string into an array using a delimiter
  - Can also be used with **regular expressions** surrounded by **/**:
  - let a = s.split(**/[\t]+/**);

- **join** merges an array into a single string, placing a delimiter between elements

TODO

Complete Exercises 5 to 8 from
In-class Exercise on Javascript String and Array

Be sure to work with your in-class partner.