

Client-Side Web Programming: JavaScript (Part 1)

Copyright © 2022 by
Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
 - JavaScript in browsers
 - The browser DOM
 - Browser event handling

Agenda

- **JavaScript in browsers**
- The browser DOM
- Browser events

JavaScript in Browsers

- So far:
 - JavaScript as a language
 - (Appendix) JavaScript for **server-side** web programming
- Now:
 - JavaScript for **client-side** web programming

4

JavaScript in Browsers

So far:

JavaScript as a language

Run using Node.js

JavaScript for server-side web programming (appendix)

Run using Node.js HTTP server and Express

Now:

JavaScript for client-side web programming

Run using browsers

JavaScript in Browsers

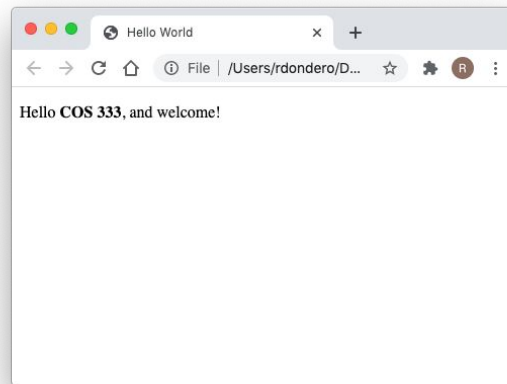
- To run JavaScript code in browsers:
 - Embed it into HTML
- To embed ***someJavaScriptCode*** into HTML:

```
<script>someJavaScriptCode</script>
```

```
<script src="someURL"></script>  
<!-- where someJavaScript code  
      resides at someURL -->
```

JavaScript in Browsers

- See [hello.html](#)



6

JavaScript in Browsers

[see slide]

Code notes

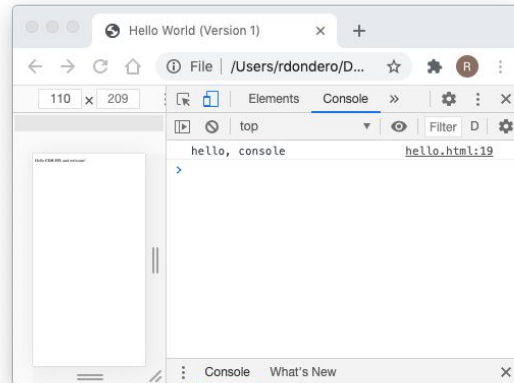
`console.log()` writes to the browser console

JavaScript in Browsers

- See **hello.html** (cont.)
 - The `console.log()` function writes to **browser console**
 - In Chrome
 - More Tools → Developer Tools → Console
 - In Firefox
 - Tools → Web Developer → Web Console

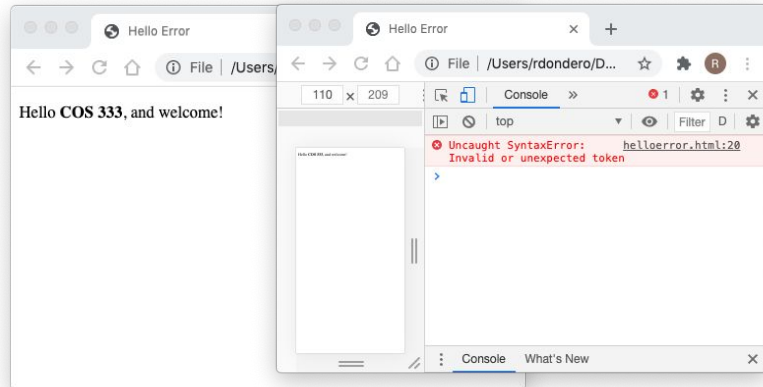
JavaScript in Browsers

- See **hello.html** (cont.)



JavaScript in Browsers

- See **helloerror.html**
 - Errors are reported to console



9

JavaScript in Browsers

[see slide]

Code notes:

The rendering of the HTML document gives no indication of failure
JavaScript writes an error message to the browser console

Agenda

- JavaScript in browsers
- **The browser DOM**
- Browser events

Browser DOM

- **Document Object Model (DOM)**

- A programmatic **model** of the current HTML **document**
- Each HTML element is represented as an object
- Each object is the node of a tree
 - window → document → element
- Standardized by **World Wide Web Consortium (W3C)**
 - Some old browsers don't observe

Browser DOM: Details

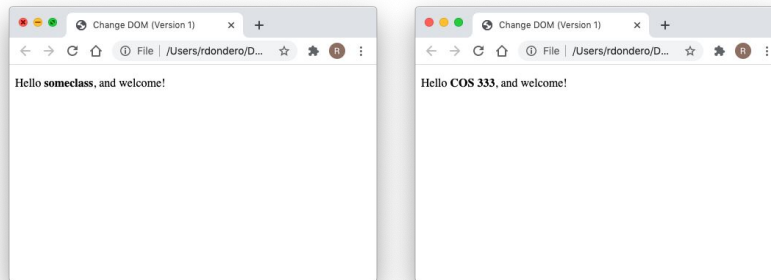
- **window** node
 - Represents the browser window
 - Important property: `document`
- **document** node
 - Represents the current document
 - Typically contains references to *element* nodes
 - Important property: `childNodes[]`
 - Important property: `getElementById()`

Browser DOM: Details

- **element** node
 - Represents an HTML element (<body>, <p>, <hr>, ...)
 - May contain attributes
 - May contain references to child element nodes
 - Important property: `attributes`
 - Important property: `childNodes[]`
 - Important property: `innerHTML`

Browser DOM: Example

- See **changedom1.html**
 - The job:
 - Display message in browser
 - Change part of message after 3 sec



14

Browser DOM: Example

[see slide]

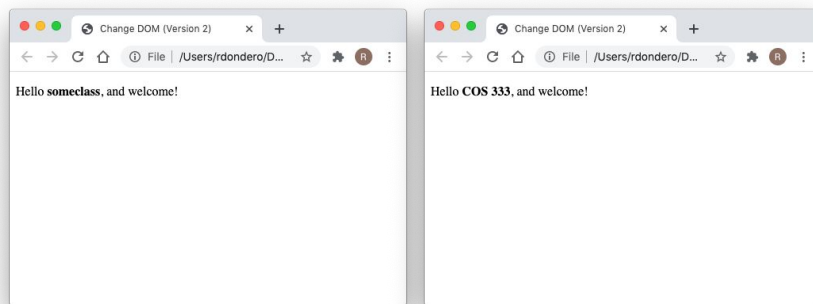
Code notes

JavaScript writes to `` element

Accesses element by traversing the DOM

Browser DOM: Example

- See **changedom2.html**
 - The job:
 - Same as changedom1.html



15

Browser DOM: Example

[see slide]

Code notes

- JavaScript writes to `` element
- Accesses element by id

Browser DOM: Example

- See **writedom.html**
 - The job:
 - Display DOM tree of current page in browser window

16

Browser DOM: Example

[see slide]

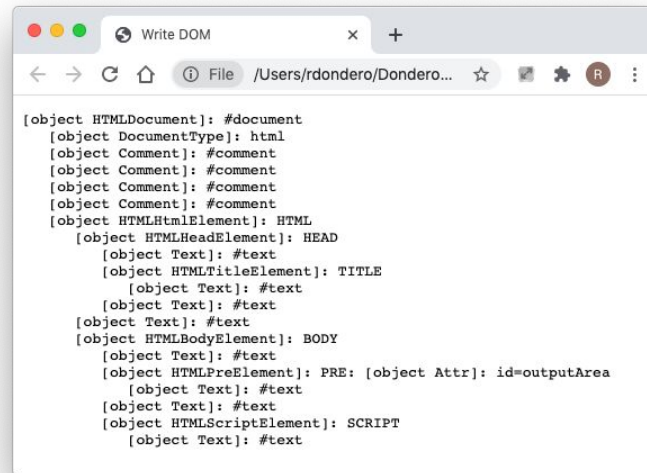
Code notes;

Recursively traverses DOM

Display to browser window

Writes DOM of current document to innerHTML of <pre> element

Browser DOM: Example



```
[object HTMLDocument]: #document
[object DocumentType]: html
[object Comment]: #comment
[object Comment]: #comment
[object Comment]: #comment
[object Comment]: #comment
[object Comment]: #comment
[object HTMLElement]: HTML
[object HTMLHeadElement]: HEAD
  [object Text]: #text
  [object HTMLTitleElement]: TITLE
    [object Text]: #text
  [object Text]: #text
[object Text]: #text
[object HTMLBodyElement]: BODY
  [object Text]: #text
  [object HTMLPreElement]: PRE: [object Attr]: id=outputArea
    [object Text]: #text
  [object Text]: #text
  [object HTMLScriptElement]: SCRIPT
    [object Text]: #text
```

17

Browser DOM: Example

[see slide]

Note:

For each HTML element and attribute there is one node in the DOM tree
There are DOM nodes for HTML comments, white space, etc.

The point:

The DOM tree is more complex than you might expect
So use id attributes to avoid the need to traverse the DOM tree

Agenda

- JavaScript in browsers
- The browser DOM
- **Browser events**

Browser Events

- ***Event***

- An occurrence on a particular element

- ***Event handler***

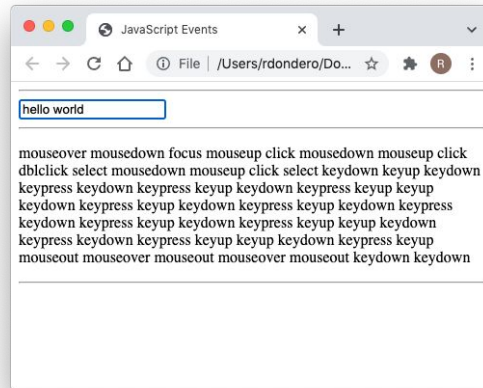
- JavaScript code
- Installed on a particular element for a particular event
- When that particular event occurs on that particular element, event handler is executed

Browser Events: Installing Handlers

- **Question:** How to install event handlers on an element?
- **Answer 1:** In HTML via the element's attributes
- **Answer 2:** In JavaScript via `addEventListener()`

Browser Events: Example

- See **events1.html**
 - The job:
 - Illustrate event handling on an `<input>` element



21

Browser Events: Example

[see slide]

Code notes:

Events on <input> element
Try to make every event occur!

Browser Events: Installing Handlers

- **Question:** How to install event handlers on an element?
- **Answer 1:** In HTML via the element's attributes
- **Answer 2:** In JavaScript via `addEventListener()`

Browser Events: Example

- See **events2.html**
 - The job:
 - Same as events1.html

23

Browser Events: Example

[see slide]

Code notes:

- Overall structure

- Event handlers on input object

Browser Events: Summary

- Event handling in **HTML**
 - Implemented via *onsomeevent* HTML attributes
 - (pro) Succinct
- Event handling in **JavaScript**
 - Implemented via `addEventListener()`
 - (pro) Separates concerns
 - I prefer; I'll use

Browser Events: Summary

- Browser events
 - There are many kinds of events
 - 85 of them
 - https://www.w3schools.com/jsref/dom_obj_event.asp

JavaScript and HTML Summary

- To use JavaScript with HTML, typically:
 - Embed JavaScript code in HTML in `<script>` elements
 - For element X, associate event Y with an event handler (JavaScript code)
 - When event Y occurs on element X, browser executes event handler (JavaScript code)
 - Event handler reads from and/or writes to the DOM
 - Writing to the DOM changes the rendered page

Summary

- We have covered:
 - JavaScript in browsers
 - The browser DOM
 - Browser event handling