

Client-Side Web Programming: JavaScript (Part 1)

Copyright © 2024 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

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

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

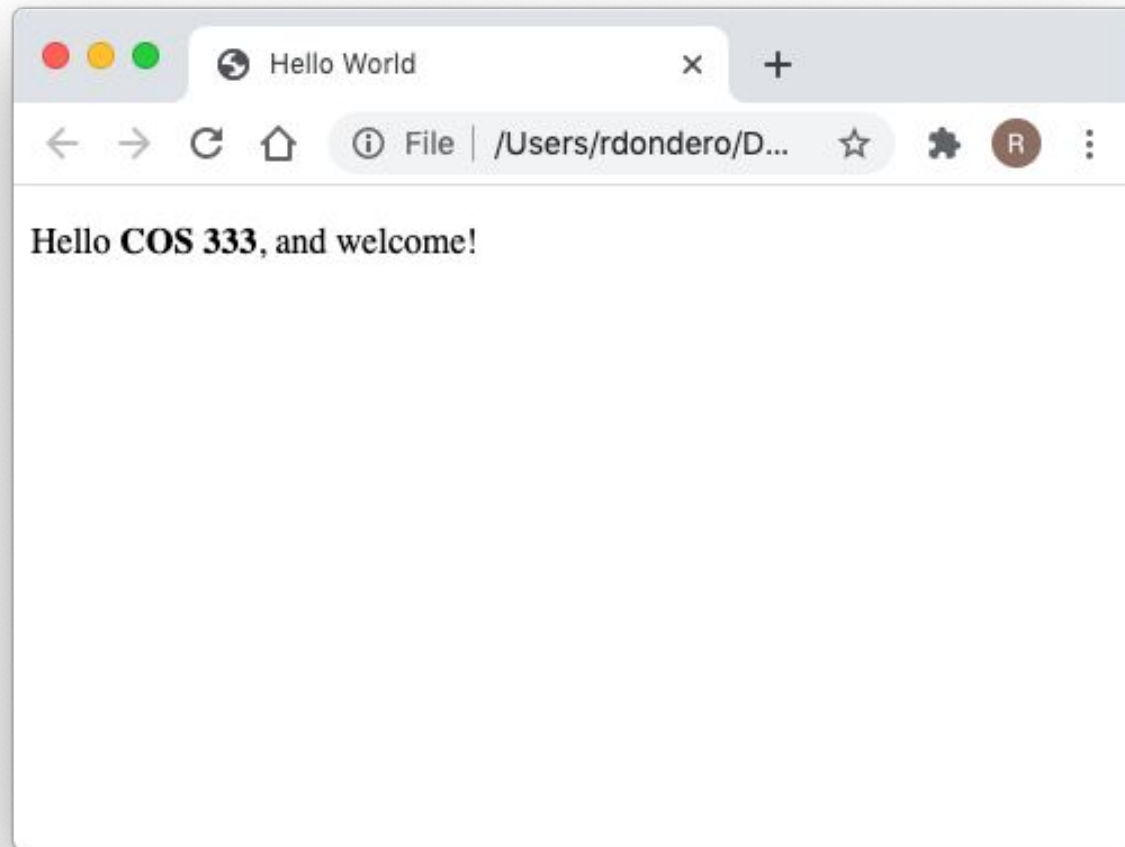
```
<script src="someURL"></script>
```



Where *someJavaScriptCode*
resides at *someURL*

JavaScript in Browsers

- See **hello.html**



hello.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- hello.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:     <title>Hello World</title>
11:   </head>
12:
13:   <body>
14:
15:     <p>Hello <strong>COS 333</strong>, and welcome!</p>
16:
17:     <script>
18:       'use strict';
19:       console.log('hello, console');
20:     </script>
21:
22:   </body>
23:
24: </html>

```

helloerror.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- helloerror.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:     <title>Hello Error</title>
11:   </head>
12:
13:   <body>
14:
15:     <p>Hello <strong>COS 333</strong>, and welcome!</p>
16:
17:     <script>
18:       'use strict';
19:       // Unterminated string literal on next line.
20:       console.log('hello, console);
21:     </script>
22:
23:   </body>
24:
25: </html>
26:

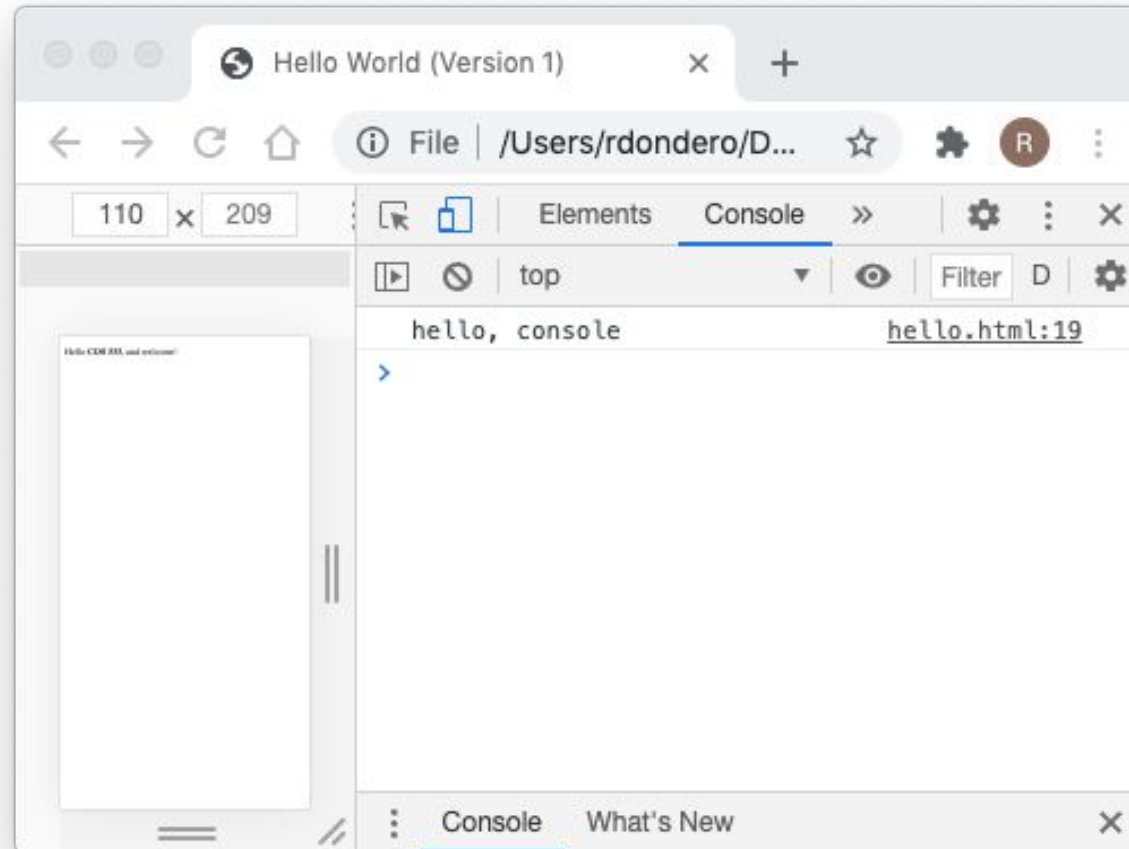
```

JavaScript in Browsers

- Accessing the browser console
 - Menu
 - **Chrome**: More Tools → Developer Tools → Console
 - **Firefox**: More Tools → Web Developer tools → Web Console
 - **Safari**: Develop → Show JavaScript console
 - Keyboard shortcut:
 - **Linux**: Ctrl-Shift-i
 - **MS Windows**: Ctrl-Shift-i
 - **Mac**: Cmd-Option-i

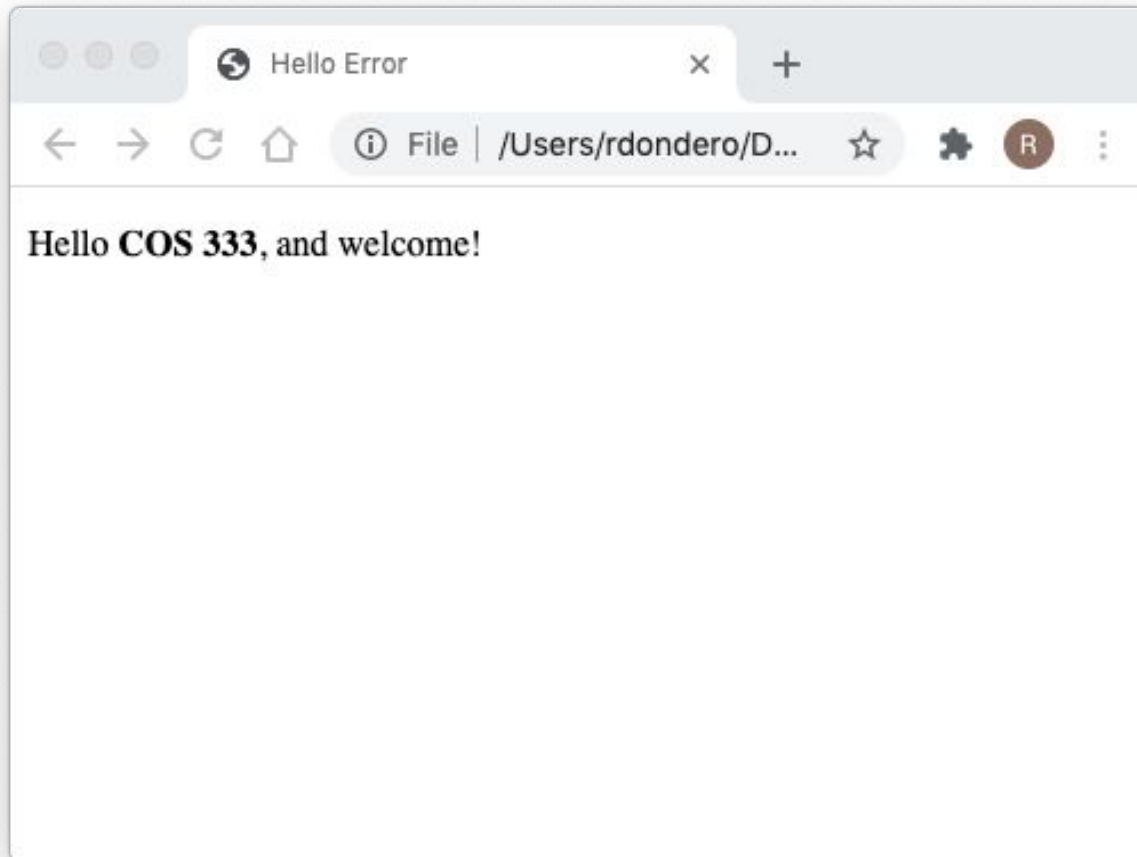
JavaScript in Browsers

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



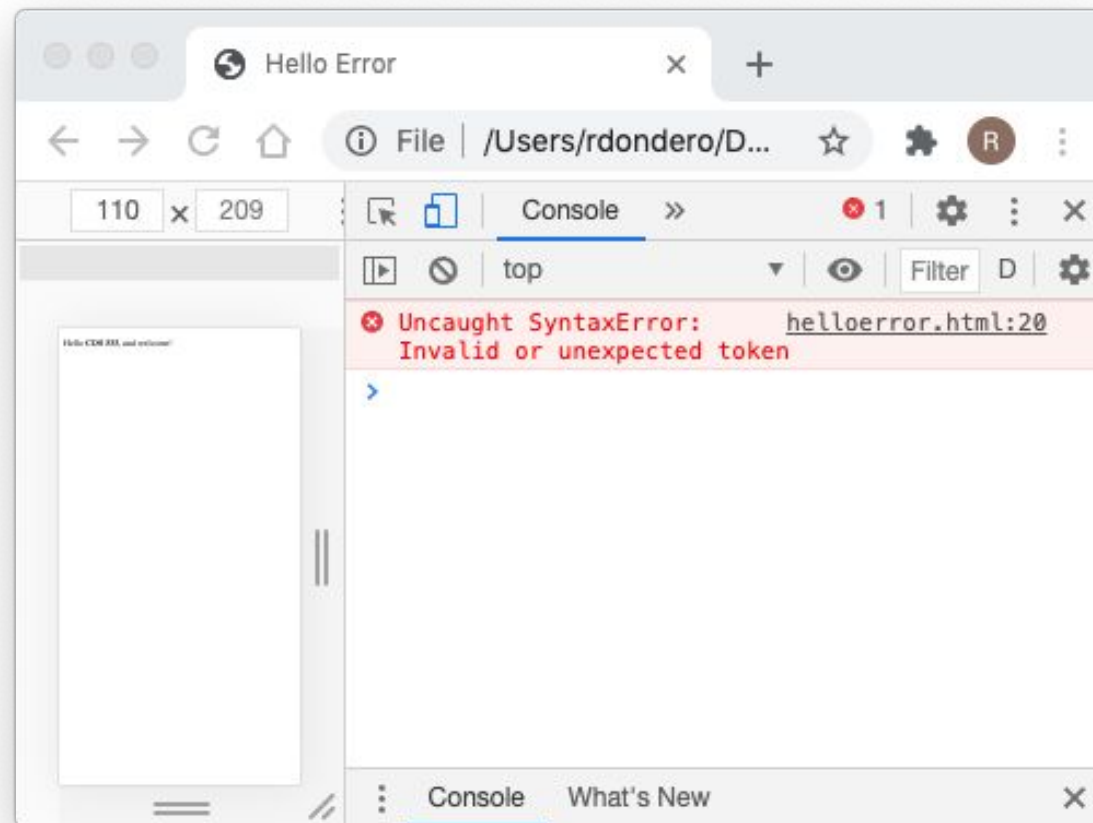
JavaScript in Browsers

- See **helloerror.html**



JavaScript in Browsers

- See [helloerror.html](#) (cont.)



hello.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- hello.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:     <title>Hello World</title>
11:   </head>
12:
13:   <body>
14:
15:     <p>Hello <strong>COS 333</strong>, and welcome!</p>
16:
17:     <script>
18:       'use strict';
19:       console.log('hello, console');
20:     </script>
21:
22:   </body>
23:
24: </html>

```

helloerror.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- helloerror.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:     <title>Hello Error</title>
11:   </head>
12:
13:   <body>
14:
15:     <p>Hello <strong>COS 333</strong>, and welcome!</p>
16:
17:     <script>
18:       'use strict';
19:       // Unterminated string literal on next line.
20:       console.log('hello, console);
21:     </script>
22:
23:   </body>
24:
25: </html>
26:

```

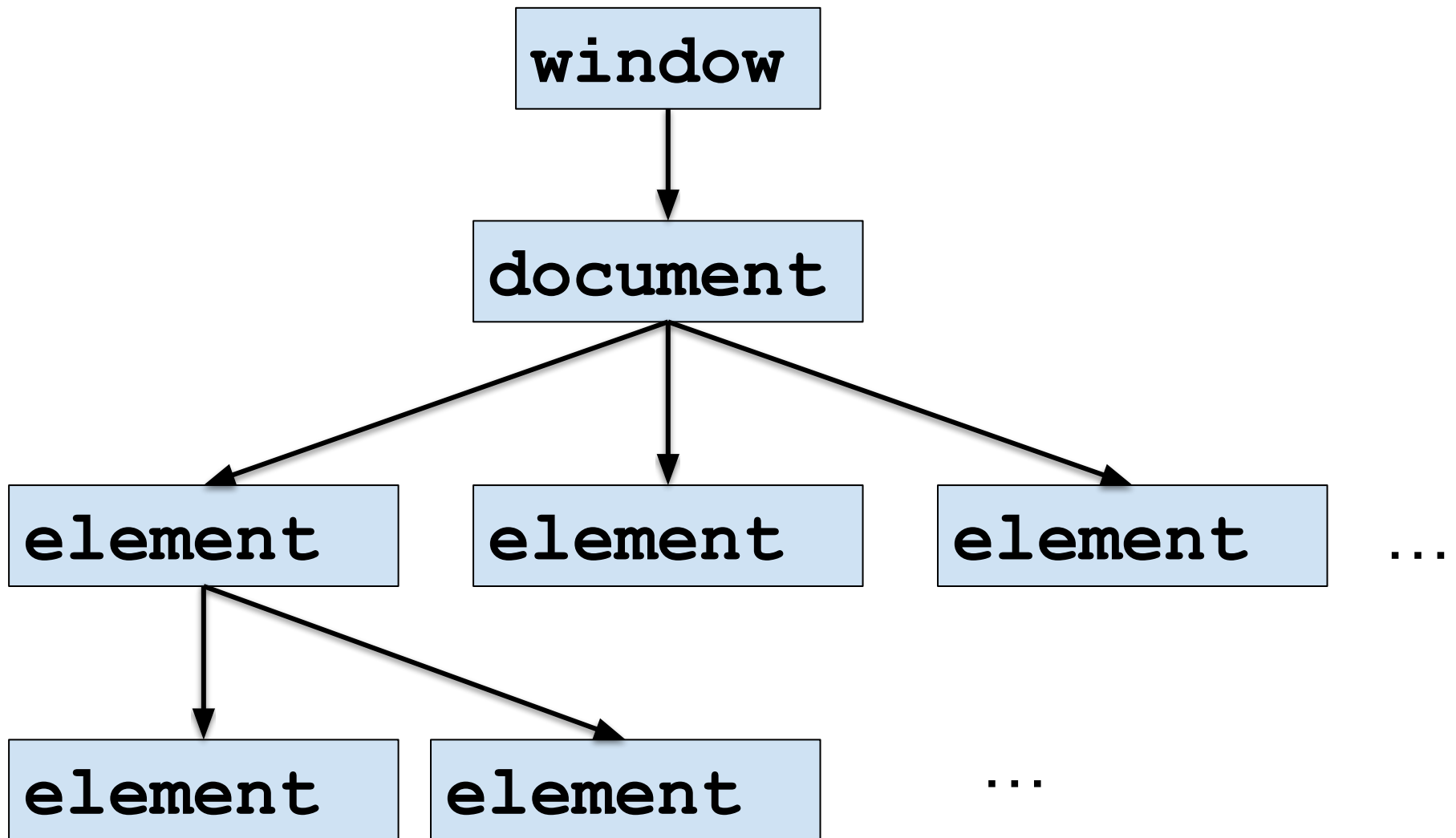
Agenda

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

Browser DOM

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

Browser DOM



Browser DOM

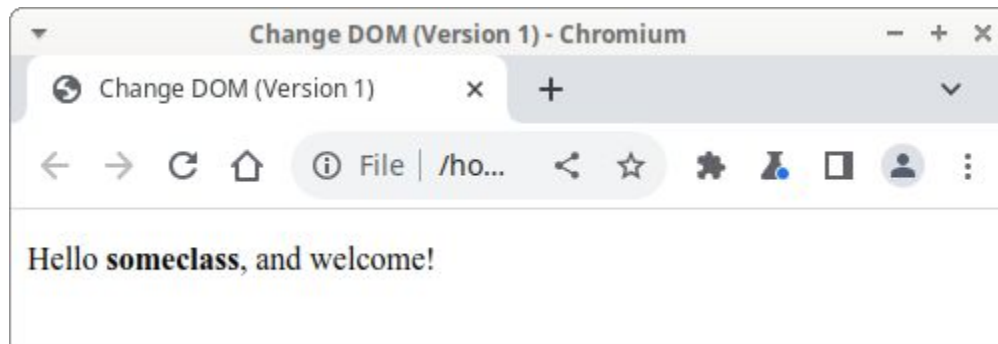
- *window* node
 - Represents the browser window
- *document* node
 - Represents the current document
 - Important property: `childNodes[]`
 - Important property: `getElementById()`

Browser DOM

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

Browser DOM

- See **changedom1.html**



changedom1.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- changedom1.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:    <title>Change DOM (Version 1)</title>
11:   </head>
12:
13:   <body>
14:    <p>Hello <strong>someclass</strong>, and welcome!</p>
15:   </body>
16:
17: </html>

```

changedom2.html (Page 1 of 1)

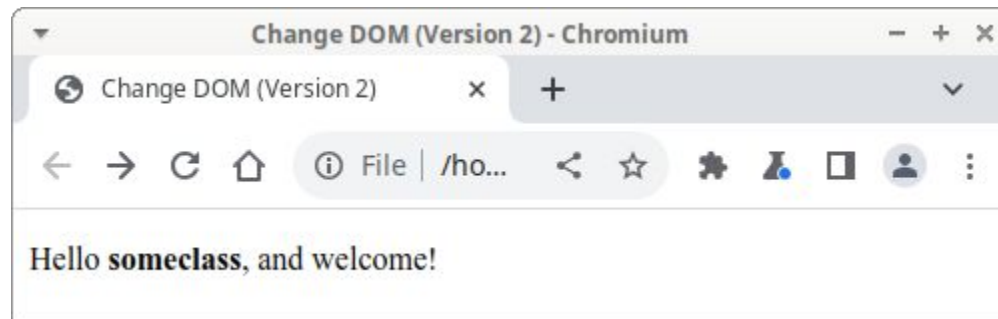
```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- changedom2.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:    <title>Change DOM (Version 2)</title>
11:   </head>
12:
13:   <body>
14:    <p>Hello <strong>someclass</strong>, and welcome!</p>
15:
16:    <script>
17:      'use strict';
18:      function writeDomTree(node, indent) {
19:        console.log(indent + node + ': ' + node.nodeName);
20:        for (let child of node.childNodes)
21:          writeDomTree(child, indent + ' ');
22:      }
23:      writeDomTree(document, '');
24:    </script>
25:   </body>
26:
27: </html>

```

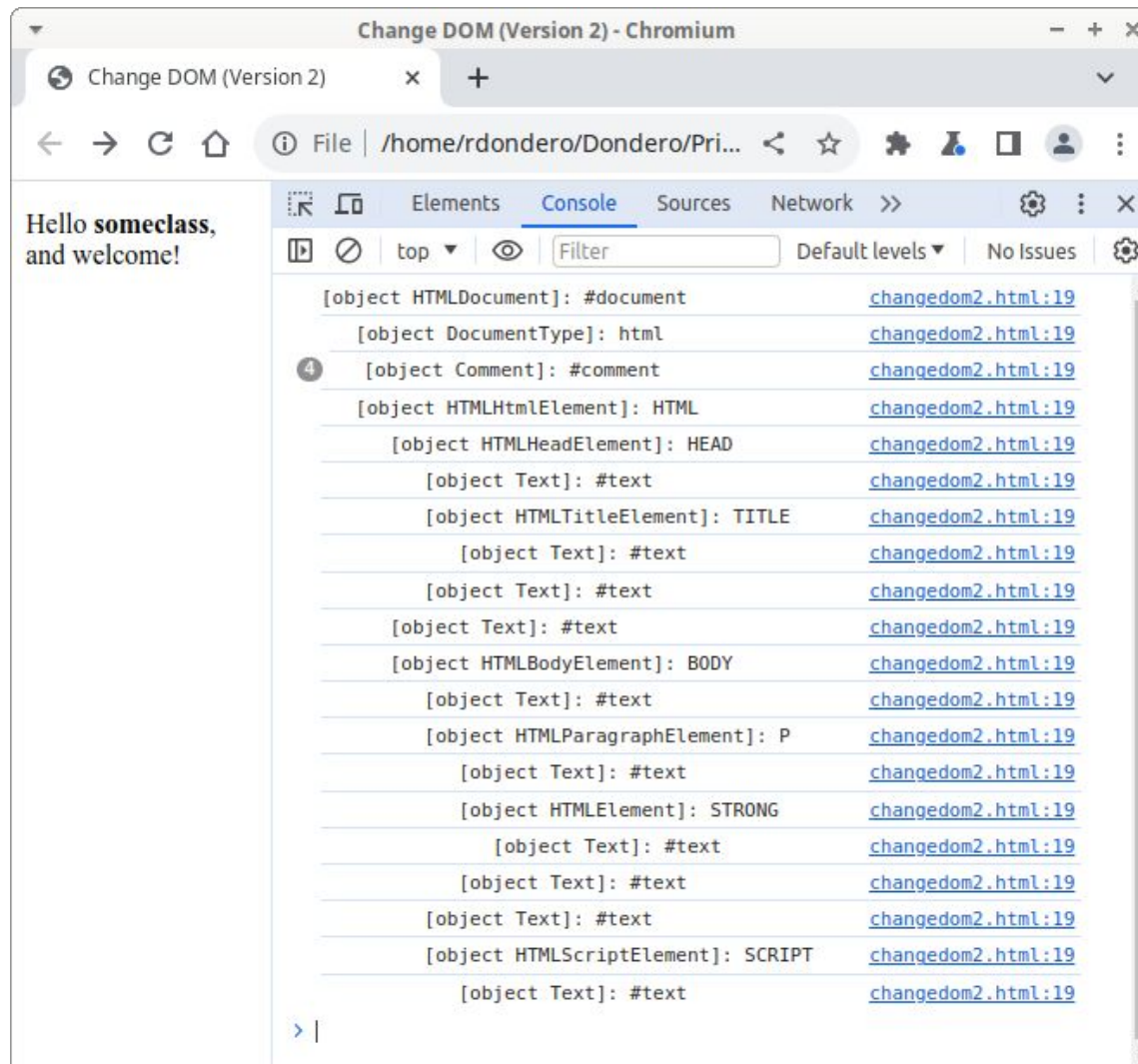
Browser DOM

- See **changedom2.html**



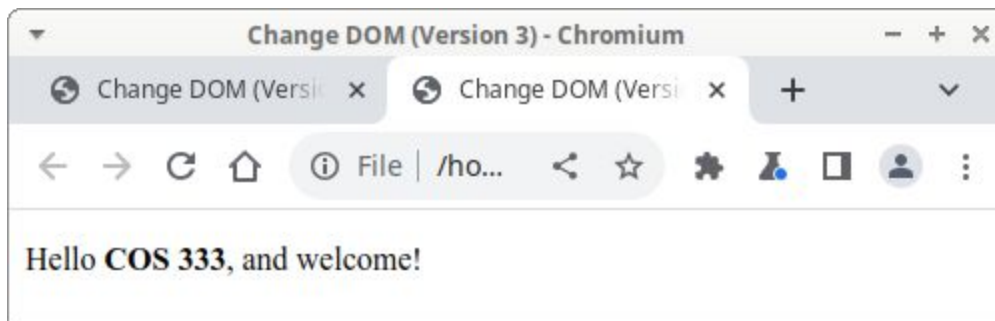
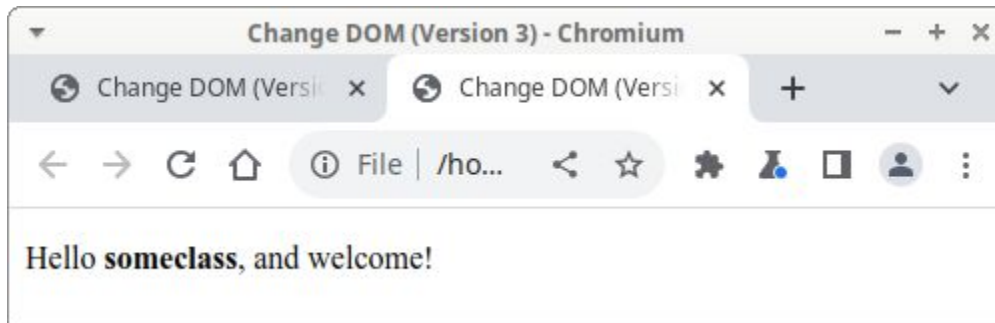
Browser DOM

- See **changedom2.html** (cont.)



Browser DOM

- See **changedom3.html**



changedom3.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- changedom3.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:    <title>Change DOM (Version 3)</title>
11:   </head>
12:
13:   <body>
14:    <p>Hello <strong>someclass</strong>, and welcome!</p>
15:
16:    <script>
17:      'use strict';
18:
19:      function changeName() {
20:        let htmlNode = document.childNodes[5];
21:        let bodyNode = htmlNode.childNodes[2];
22:        let pNode = bodyNode.childNodes[1];
23:        let strongNode = pNode.childNodes[1];
24:        strongNode.innerHTML = 'COS 333';
25:
26:        // Or more succinctly:
27:        // document.childNodes[5].childNodes[2].childNodes[1].
28:        //   childNodes[1].innerHTML = 'COS 333';
29:      }
30:
31:      window.setTimeout(changeName, 3000);
32:    </script>
33:   </body>
34:
35: </html>

```

changedom4.html (Page 1 of 1)

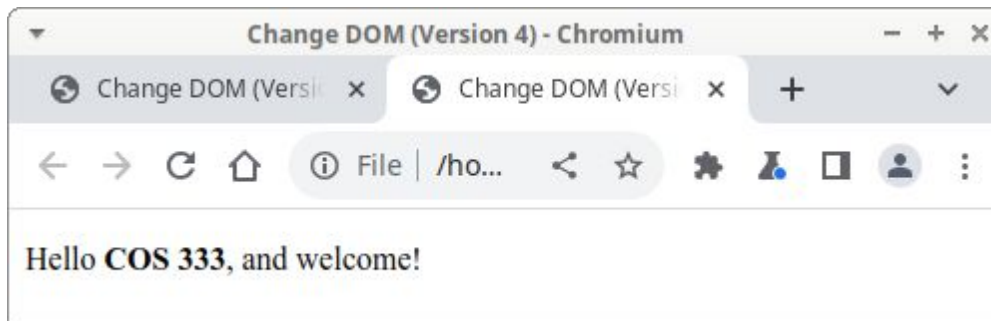
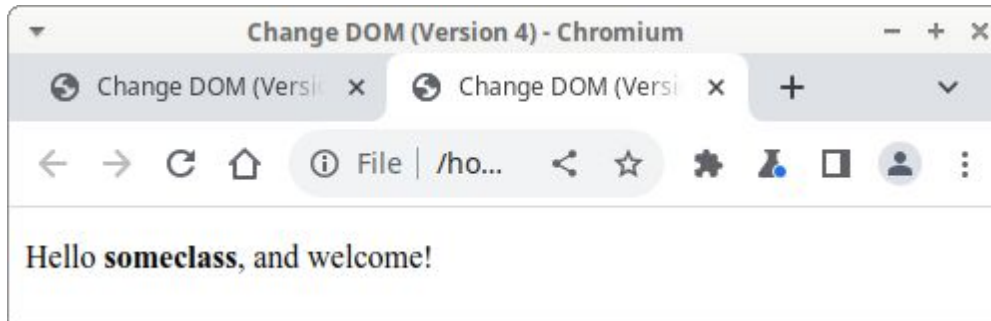
```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- changedom4.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:
9:   <head>
10:    <title>Change DOM (Version 4)</title>
11:   </head>
12:
13:   <body>
14:
15:    <p>Hello <strong id="myStrong">someclass</strong>,
16:    and welcome!</p>
17:
18:    <script>
19:      'use strict';
20:
21:      function changeName() {
22:        // Access an element by element id:
23:        let strongNode = document.getElementById('myStrong');
24:        strongNode.innerHTML = 'COS 333';
25:      }
26:
27:      window.setTimeout(changeName, 3000);
28:    </script>
29:
30:   </body>
31:
32: </html>

```

Browser DOM

- See **changedom4.html**



Agenda

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

Browser Events

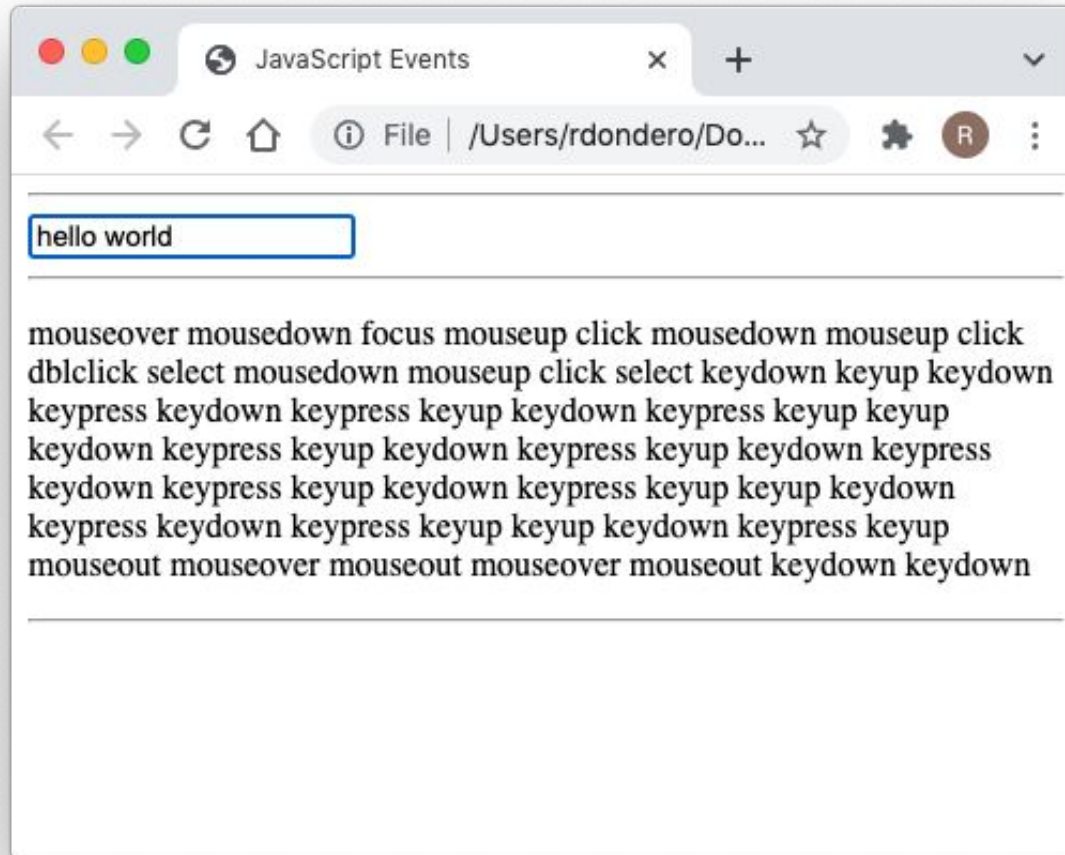
- ***Event***
 - An occurrence on a particular element
- ***Event handler***
 - JavaScript code
 - Installed on a particular element for a particular event

Browser Events

- **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

- See **events1.html**



events1.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- events1.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:   <head>
9:     <title>JavaScript Events</title>
10:   </head>
11:
12:   <body>
13:     <hr>
14:     <input
15:       type="text"
16:       value="Type something here"
17:
18:       onmouseover="notify('mouseover') "
19:       onmouseout="notify('mouseout') "
20:
21:       onfocus="notify('focus') "
22:       onblur="notify('blur') "
23:
24:       onmousedown="notify('mousedown') "
25:       onmouseup="notify('mouseup') "
26:       onclick="notify('click') "
27:       ondblclick="notify('dblclick') "
28:       onselect="notify('select') "
29:
30:       onkeydown="notify('keydown') "
31:       onkeypress="notify('keypress') "
32:       onkeyup="notify('keyup') ">
33:     <hr>
34:     <p id="myNotifyArea"></p>
35:     <hr>
36:
37:     <script>
38:       'use strict';
39:       function notify(eventName) {
40:         let notifyArea = document.getElementById('myNotifyArea');
41:         notifyArea.innerHTML += eventName;
42:         notifyArea.innerHTML += ' ';
43:       }
44:     </script>
45:
46:   </body>
47: </html>

```

blank (Page 1 of 1)

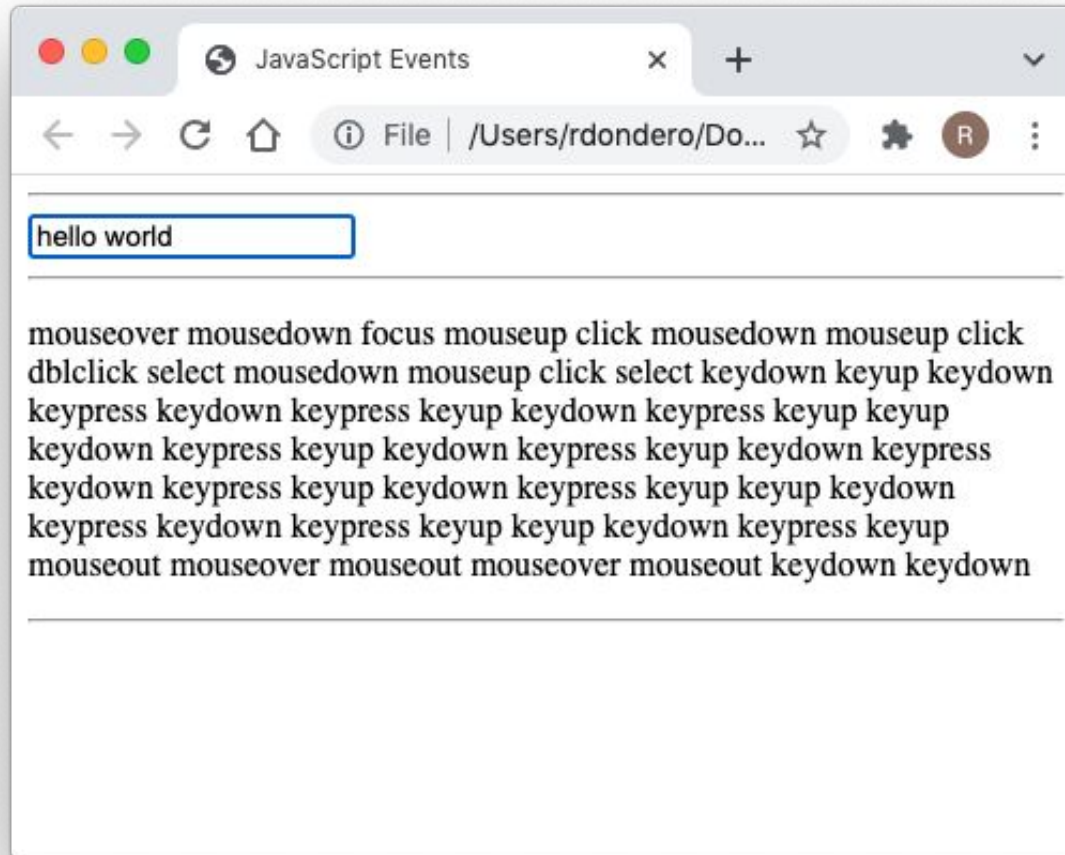
1: This page is intentionally blank.

Browser Events

- **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

- See **events2.html**



events2.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <!-- ===== -->
3: <!-- events2.html -->
4: <!-- Author: Bob Dondero -->
5: <!-- ===== -->
6:
7: <html>
8:   <head>
9:     <title>JavaScript Events</title>
10:   </head>
11:
12:   <body>
13:     <hr>
14:     <input type="text" value="Type something here" id="myInput">
15:     <hr>
16:     <p id="myNotifyArea"></p>
17:     <hr>
18:
19:     <script>
20:
21:       'use strict';
22:
23:       function notify(eventName) {
24:         let notifyArea = document.getElementById('myNotifyArea');
25:         notifyArea.innerHTML += eventName;
26:         notifyArea.innerHTML += ' ';
27:       }
28:
29:       function setup() {
30:         const myInput = document.getElementById('myInput');
31:
32:         myInput.addEventListener('mouseover',
33:           function() {notify('mouseover');}
34:         );
35:         myInput.addEventListener('mouseout',
36:           function() {notify('mouseout');}
37:         );
38:
39:         myInput.addEventListener('focus',
40:           function() {notify('focus');}
41:         );
42:         myInput.addEventListener('blur',
43:           function() {notify('blur');}
44:         );
45:
46:         myInput.addEventListener('mousedown',
47:           function() {notify('mousedown');}
48:         );
49:         myInput.addEventListener('mouseup',
50:           function() {notify('mouseup');}
51:         );
52:         myInput.addEventListener('click',
53:           function() {notify('click');}
54:         );
55:         myInput.addEventListener('dblclick',
56:           function() {notify('dblclick');}
57:         );
58:         myInput.addEventListener('select',
59:           function() {notify('select');}
60:         );
61:
62:         myInput.addEventListener('keydown',
63:           function() {notify('keydown');}
64:         );
65:         myInput.addEventListener('keypress',

```

events2.html (Page 2 of 2)

```

66:           function() {notify('keypress');}
67:         );
68:         myInput.addEventListener('keyup',
69:           function() {notify('keyup');}
70:         );
71:       }
72:
73:       document.addEventListener('DOMContentLoaded', setup);
74:
75:     </script>
76:
77:   </body>
78: </html>

```

Browser Events

- Which is better, events1.html or events2.html?
 - **events1.html**: more succinct
 - **events2.html**: better separation of concerns

Browser Events

- Browser events
 - https://www.w3schools.com/jsref/dom_obj_event.asp

JavaScript in Browsers Summary

- To use JavaScript in browsers:
 - Embed JavaScript code in HTML in `<script>` elements
 - For element ELEMENT1, associate event EVENT1 with an event handler (JavaScript code)
 - Then...

JavaScript in Browsers Summary

- To use JavaScript in browsers (cont.):
 - When EVENT1 occurs on ELEMENT1, browser places event on JS Event Queue to execute event handler
 - JS Engine (when ready) removes event from JS event queue, and executes event handler
 - 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