

jQuery and Selectors

- 1.** Make a pushbutton that, when pressed, turns all “li” elements blue. Make another pushbutton that returns them to their original state. Use explicit onclick handlers in the HTML.
- 2.** Repeat the above problem, but avoid any explicit JavaScript in the HTML page (i.e., use the “unobtrusive JavaScript” approach that assigns to the click handler programmatically after the DOM is loaded).
- 3.** Make a pushbutton that, when pressed, turns “li” elements blue only if the list is inside a <div> whose CSS class is “colorChangeable”. Make another pushbutton that returns them to their original state. Make sure your HTML page has at least one inside such a div and at least one outside.
- 4.** Make a pushbutton that, when pressed, turns “li” elements blue only if the list is inside a <div> whose CSS class is “colorChangeable”, and only if the li element has a class called “modifiable”. Make another pushbutton that returns them to their original state.
- 5.** Make a pushbutton that, when pressed, fires off an Ajax request. The server should return the name of an HTML element (or CSS selector), and your client-side code should hide all of those elements. Have the server use Math.random() to return different HTML element names (or CSS selector strings). Hit the Reload button in the browser to return it to its original state.