JavaScript, Widgets, & Events

Prof. Lydia Chilton

COMS 4170

5 February 2025

Users interact with the system to accomplish a goal.

To buy a book.

The designer must create the subgoals and interactions to help them accomplish it.

Goal: Buy a book

Subgoal:	Find it	Add to cart	Enter payment info	Place order
Interaction:	Type, click	click	Type, click, point	Click

Low-level interactions take time and effort. Minimize them because you do them a lot.

Secure Payment Info
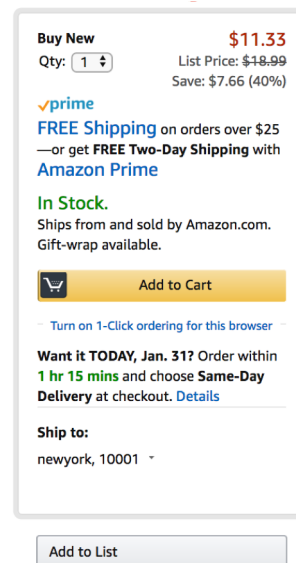
Name (as it appears on your card)
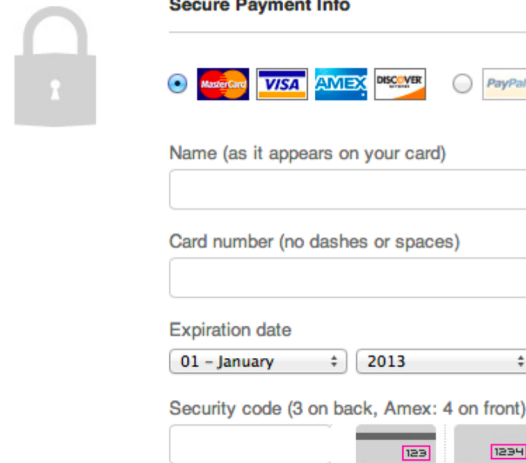
Card number (no dashes or spaces)

Expiration date

01 – January   2013

Security code (3 on back, Amex: 4 on front)

Move
Click
Move
Click
TypeTypeTypeType
Move
Click
TypeTypeTypeType
Move
Tunnel
Click
Move
Tunnel
Click
TypeTypeTypeType

Creating Interactions on the web has two parts:

1. Program the interface and style in HTML & CSS

```
31    <body>
32
33        <button id="counter" class="btn btn-primary">Counter (0)</button>
34
35    </body>
36
```

counter.html

← → C ⓘ file:///Users/lydiach

Counter (0)

2. Program interactions is JavaScript

```
25
26    $(document).ready(function(){
27        $("#counter").click(function(){
28            alert("foo")
29        })
30    })
31
```

counter.html   ×

← → C ⓘ file:///Users/lydiachilton/Documents/hw4/counter.html

This page says:

foo
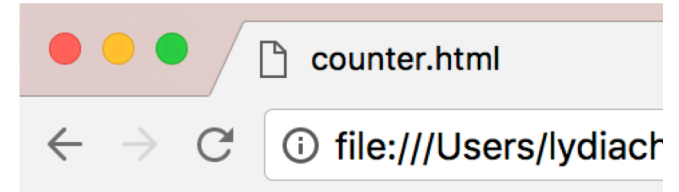
Counter (0)
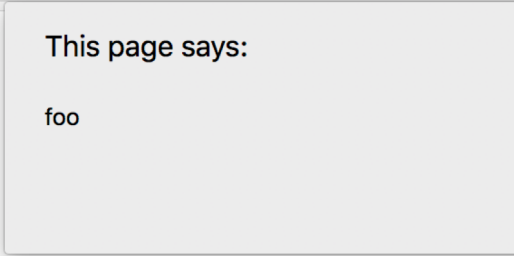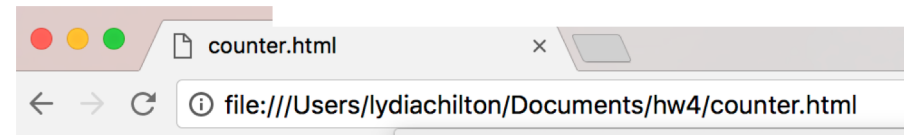
Web Page Execution

Browsers execute an HTML file from top to bottom. What will this execute?

```
1   <html>
2   <head>
3     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bo
4     <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossor
5
6     <script>
7
8       var count = 0
9
10      function incrementCount(c) {
11        return c + 1;
12      }
13
14      console.log("first count: "+count)
15      count = incrementCount(count)
16
17      console.log("second count: "+count)
18
19    </script>
20
21
22  </head>
23
24
25
26  <body>
27    <button id="counter" class="btn btn-primary">Counter (0)</button>
28  </body>
29
30  </html>
31
```
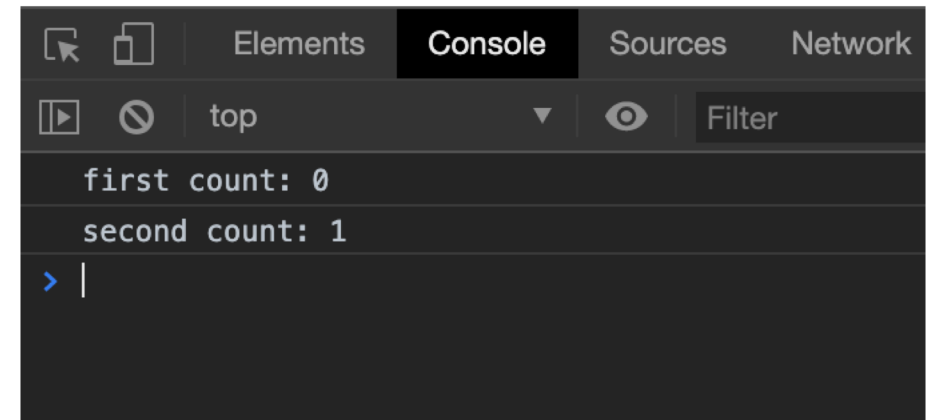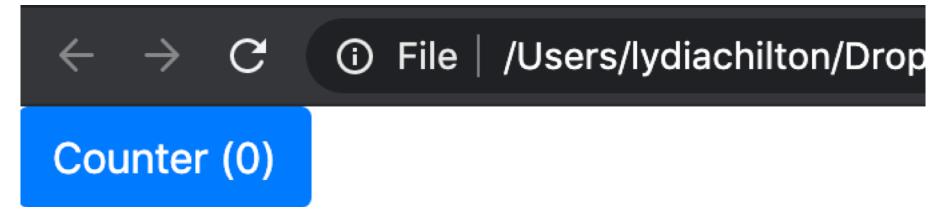
However, JavaScript functions will get "hoisted." Meaning, you can use them anywhere in scope.

```html
1  <html>
2  <head>
3    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bo
4    <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossor
5
6    <script>
7
8      var count = 0
9
10     console.log("first count: "+count)
11     count = incrementCount(count)
12
13     console.log("second count: "+count)
14
15     function incrementCount(c) {
16       return c + 1;
17     }
18
19   </script>
20
21
22  </head>
23
24
25
26  <body>
27    <button id="counter" class="btn btn-primary">Counter (0)</button>
28  </body>
29
30  </html>
31
```

Still works!

Counter (0)

File | /Users/lydiachilton/Drop

Elements **Console** Sources Network

top ▼ Filter

```
first count: 0
second count: 1
>
```
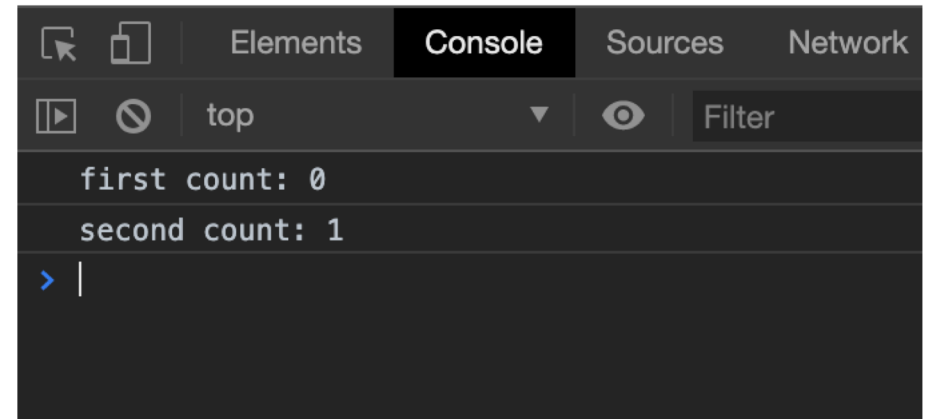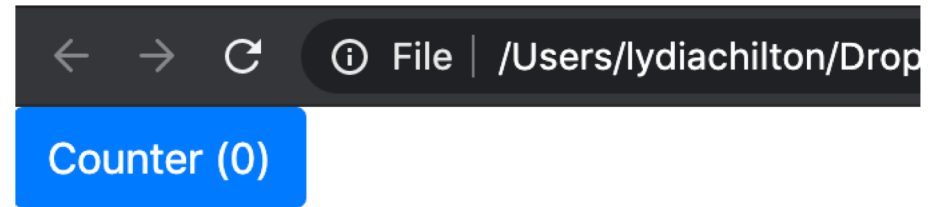
There is another (worse) way to declare functions that will not be hoisted.

```html
<script>

  var count = 0

  count = incrementCount(count)

  // this is a function expression, it will execute whenever it is called
  function incrementCount1(c) {
    return c + 1;
  }

  // This is a variable
  // it will only be available only after it is executed by the browser
  // this will create an error in this case
  var incrementCount2 = function(c) {
    return c + 1;
  }

</script>
```

Do it this way.

Three ways of declaring variables: let, const, var

freeCodeCamp (🔥)

Learn to code — free 3,000-hour curriculum

APRIL 2, 2020 / #JAVASCRIPT

Var, Let, and Const – What's the Difference?

Sarah Chima Atuonwu

Let is the preferred way to declare variables.

```
1
2   let greeting = "hello"
3
4   console.log("0: "+greeting)
5
6 ▾ if(true){
7       greeting ="hi"
8       console.log("1: "+greeting)
9   }
10
11  console.log("2: "+greeting)
12
13
```

>_ Console (beta)   ⊘ 3   ⊘ 0   ⚠ 0   ⊘ 0   Clear console   Minimize

☁ "Running fiddle"

"0: hello"

"1: hi"

"2: hi"

>_

let is block scoped, and can be re-assigned.

let will not be defined outside of scope.

```
1
2  ▾ if(true){
3      let hello ="hello"
4      console.log("1: "+hello)
5  }
6
7  console.log("2: "+hello) // this is undefined
```

>_ Console (beta)   ⊘ 1   ⓘ 0   ⚠ 0   ⊘ 1   Clear console   Minimize
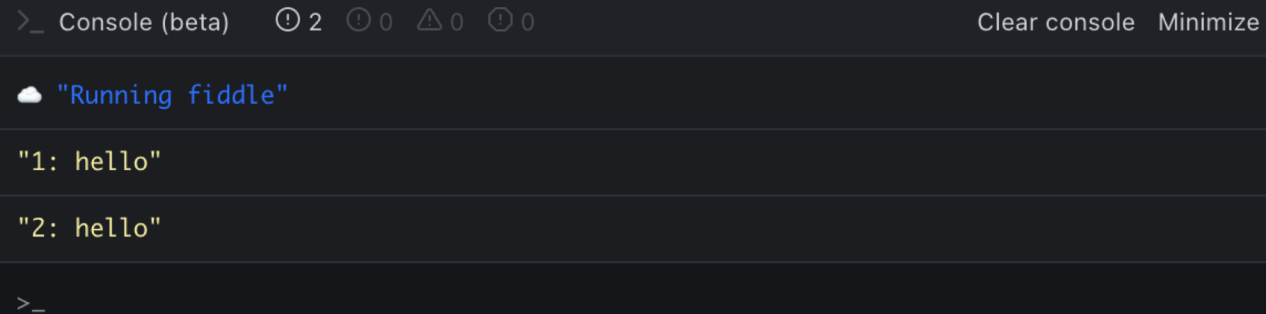
☁ "Running fiddle"

"1: hello"

"47:19 Uncaught ReferenceError: hello is not def

>_

Only use **var** when you really want a global variable

```javascript
1
2  ▾ if(true){
3      var hello ="hello"
4      console.log("1: "+hello)
5    }
6
7    console.log("2: "+hello)
```

>_ Console (beta)   ⊘ 2 ⊙ 0 ⚠ 0 ⊙ 0   Clear console Minimize

☁ "Running fiddle"

"1: hello"

"2: hello"

>_

var is globally scoped (or function scoped)
var variables can be updated and re-declared
"foo = 1" is the same as "var foo = 1"

const is good for constant variables

```
1   const debug_mode = true
2
3 ▼ if(debug_mode){
4     let hello = "hello"
5     console.log("1: "+hello)
6   }
7
8
```

```
>_  Console (beta)    ⊙1  ⊙0  ⚠0  ⊙0                    Clear console   Minimize

☁ "Running fiddle"

"1: hello"

>_
```

const is block scoped, and cannot be re-assigned.

let and **const** are good ways to declare variables.

(var is globally scoped, and can get you into trouble)

```
1    const debug_mode = true
2
3  ▼ if(debug_mode){
4      let hello = "hello"
5      console.log("1: "+hello)
6    }
7
8    console.log("2: "+hello) // not declared
```

```
>_  Console (beta)    ⊘2  ⊙0  △0  ⊙1                                    Clear console   Minimize

"hello"

☁ "Running fiddle"

"1: hello"

"<a class='gotoLine' href='#48:19'>48:19</a> Uncaught ReferenceError: hello is not de
```

let is block scoped, and can be re-assigned.
const is block scoped and cannot be re-assigned.

Adding events

When you click this button, what will it do?

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots

  <script>

    var count = 0

    function incrementCount() {
      return c + 1;
    }
  </script>



</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

counter.html

file:///Users/lydiach
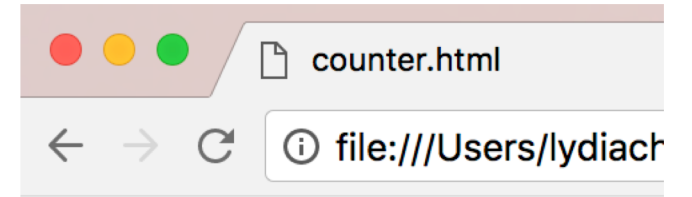
Counter (0)

Nothing

To add click handlers nicely, we're first going to include JQuery (a JS extension)

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigin="anonymous"></script>

  <script>

    var count = 0

    function incrementCount() {
      return c + 1;
    }
  </script>


</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

Syntax is similar to but different from…

Including Bootstrap

If we add an event, what will it do?

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigin

  <script>

    $("#counter").click(function(){
        alert("foo")
    })

  </script>



</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

counter.html

file:///Users/lydiach
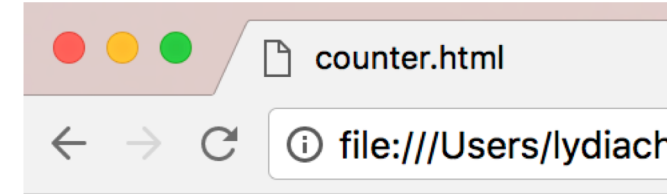
Counter (0)

Nothing

If we add an event after the document is loaded, will it finally work??

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boot
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigi

  <script>

  $(document).ready(function(){
    $("#counter").click(function(){
      alert("foo")
    })
  })

  </script>


</head>


<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

Counter (0)

File | /Users/lydiachilton/Dropbox/Columbia/Teaching/41

This page says

foo

OK

YES!!!!

We added an event. Yay!
How do we increment the counter?

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boot
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigi

  <script>

  $(document).ready(function(){
    $("#counter").click(function(){
        alert("foo")
    })
  })

  </script>



</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```
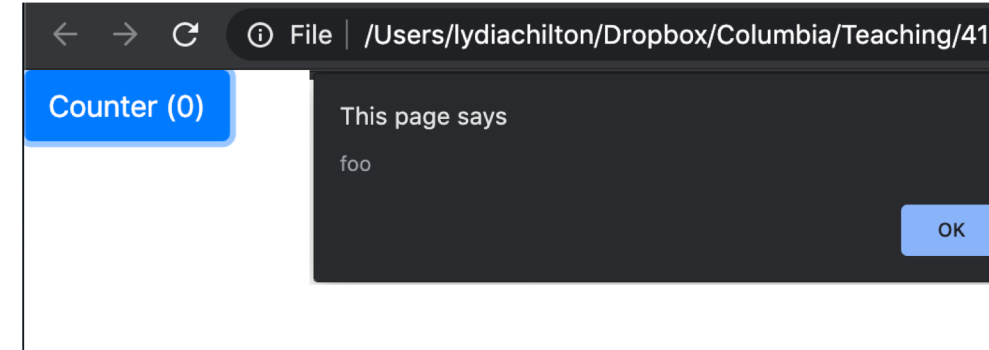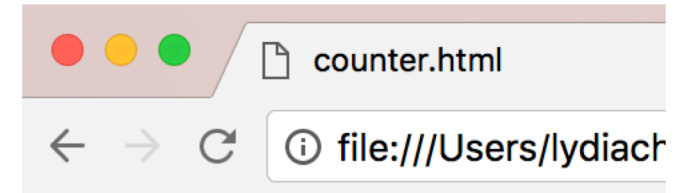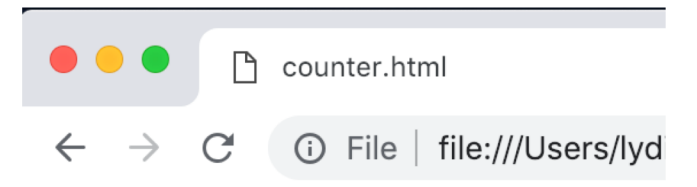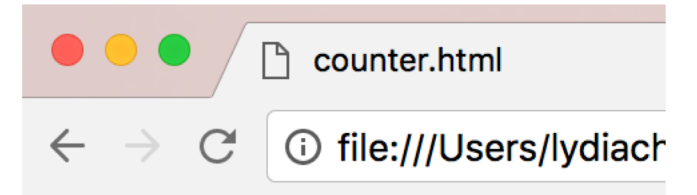
counter.html

file:///Users/lydiach

Counter (0)

How do we increment the count?

```html
1   <html>
2   <head>
3     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstr
4     <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigin="
5
6     <script>
7
8       var count = 0
9
10      function incrementCount(c) {
11        return c + 1;
12      }
13
14
15      $(document).ready(function(){
16          $("#counter").click(function(){
17              count = incrementCount(count)
18              $("#counter").html("Counter ("+count+")")
19          })
20      })
21
22    </script>
23
24
25  </head>
26
27
28
29  <body>
30    <button id="counter" class="btn btn-primary">Counter (0)</button>
31  </body>
32
33  </html>
34
```

Incrementing the count differently.

```html
1   <html>
2   <head>
3     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstra
4     <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorigin="anonymous"></script>
5
6     <script>
7
8       var count = 0
9
10      function incrementCount(c) {
11        return c + 1;
12      }
13
14
15      $(document).ready(function(){
16          $("#counter").click(function(){
17              count = incrementCount(count)
18              // $("#counter").html("Counter ("+count+")")
19              $("#count").html(count)
20          })
21      })
22
23    </script>
24
25
26  </head>
27
28
29
30  <body>
31    <button id="counter" class="btn btn-primary">Counter (<span id="count">0</span>)</button>
32  </body>
33
34  </html>
```

Jquery vs. Pure JavaScript

JQuery is a JavaScript library that make JavaScript easier (and standard across browsers)

JavaScript

```
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

What's the JS equivalent to $(``#counter")$?

JavaScript

```
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

document.getElementById("counter")

JavaScript

```javascript
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```javascript
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

What's the JavaScript equivalent of $(element).click(...)

JavaScript

```
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

We used $("#counter") again…
Is that normal?

JavaScript

```
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

Can use **this** within scope

JavaScript

```javascript
document.getElementById("counter").addEventListener("click", function(){
    this.innerHTML = "Counter (0)";
});
```

JQuery

```javascript
$("#counter").click(function(){
    $(this).html("Counter (0)");
});
```

What's the JavaScript equivalent of setting html?

JavaScript

```javascript
document.getElementById("counter").addEventListener("click", function(){
    this.innerHTML = "Counter (0)";
});
```

JQuery

```javascript
$("#counter").click(function(){
    $(this).html("Counter (0)");
});
```

Will this work?

JavaScript   JQuery

```
document.getElementById("counter").click(function(){
  document.getElementById("counter").innerHTML = "Counter (0)";
});
```

No.
Don't mix pure JavaScript with JQuery in the same line.
For your own sanity. Only use JQuery

Don't do this (even though it will work)

```html
<button onclick="myFunction()">Click me</button>


<button onclick="incrementCount(1)">Counter (1)</button>
```

Good style of attaching events in JQuery

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boo
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorig

  <script>

    var count = 0

    function incrementCount(c) {
      return c + 1;
    }


    $(document).ready(function(){
        $("#counter").click(function(){
            count = incrementCount(count)
            $("#counter").html("Counter ("+count+")")
        })
    })

  </script>


</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

counter.html
file:///Users/lydiach

counter.html
File | file:///Users/lyd

Counter (0)

Counter (1)
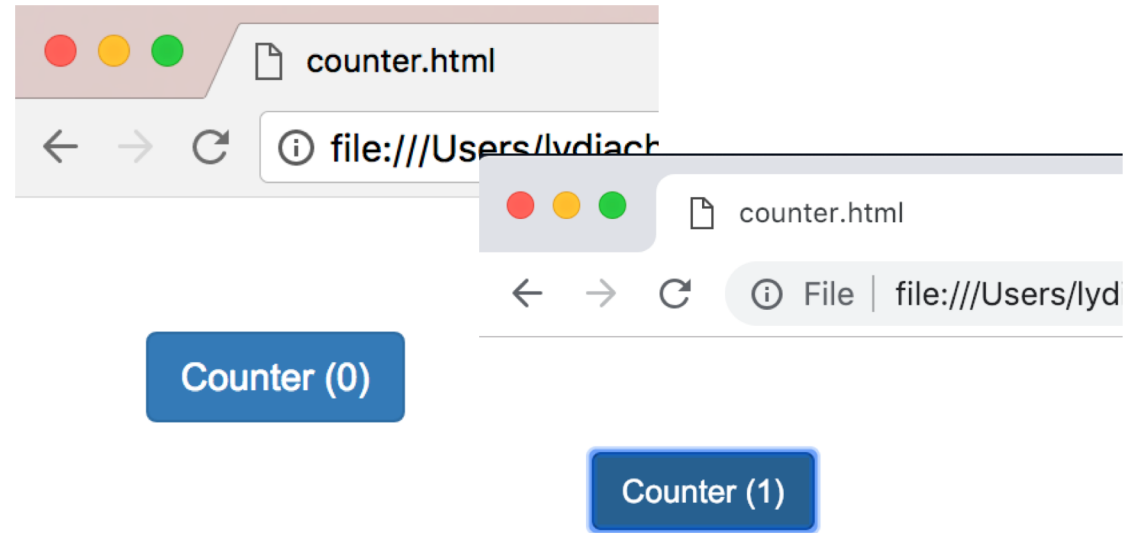
1. Uses Jquery (not pure JavaScript)

2. Attaches click handler as in the <script>
$(element).click(…)
(doesn't attach in HTML)

2. Uses $(document).ready(…)

I Am Devloper
@iamdevloper

It's only jQuery if it's from the jQueré region of France. Otherwise it's just sparkling javascript

5:20 AM · 7/23/19 · Twitter Web App
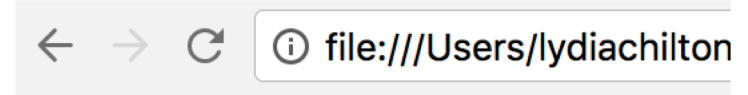
"Lydia, your memes are so... millennial."

- Student who got coal from Santa

Creating Widgets Dynamically

Statically created widget: created on page load.

HTML

```
61    <body>
62
63        <button id="counter" class="btn btn-primary"></button>
64
65    </body>
```

file:///Users/lydiachilton

Static Button (0)

JavaScript

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64        })
65    })
66
```

Dynamically created widget:
created on demand based on user interaction.

HTML

```
61    <body>
62
63        <button id="counter" class="btn btn-primary"></button>
64
65    </body>
66
```

JavaScript

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64        })
65    })
66
```

file:///Users/lydiachilton

Static Button (0)

Create a new button when I am clicked!

Click me

And me!

And me!

And me!
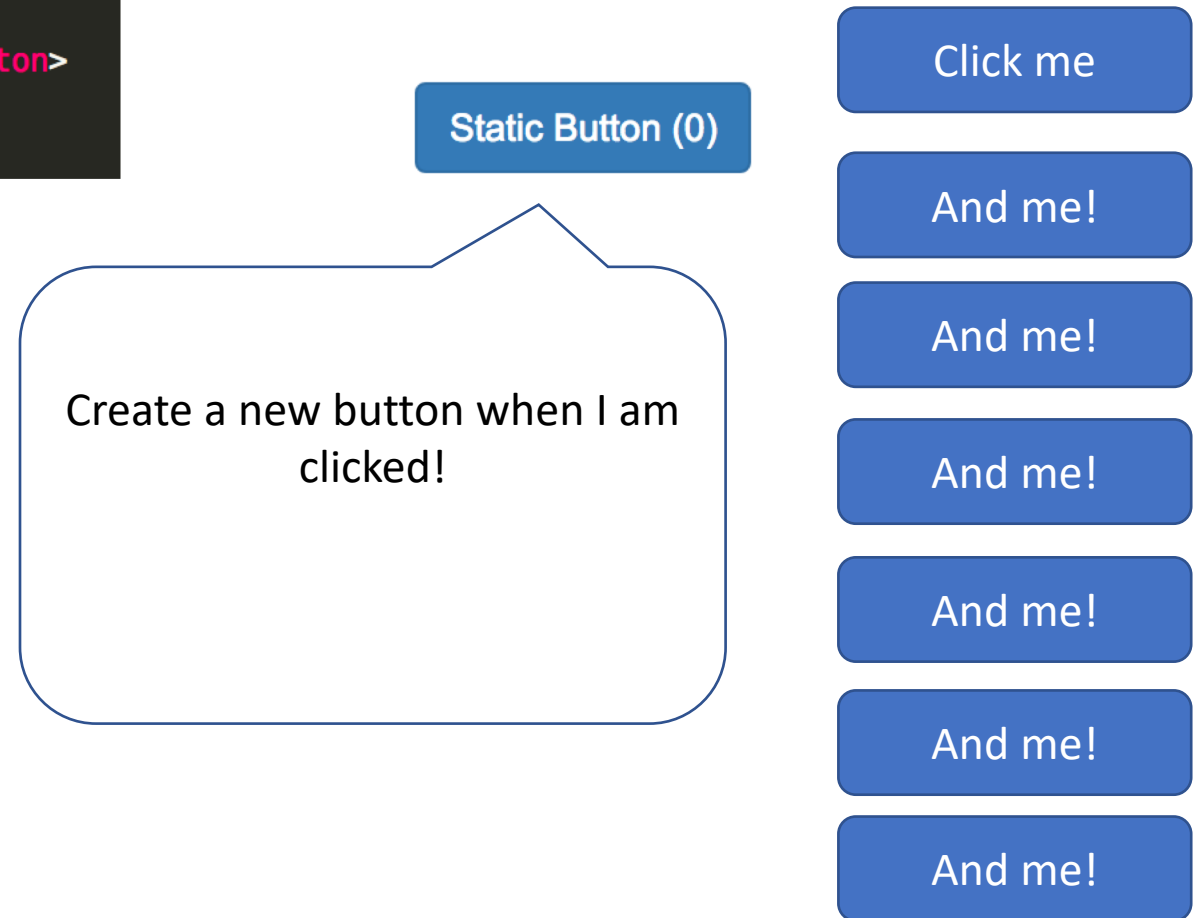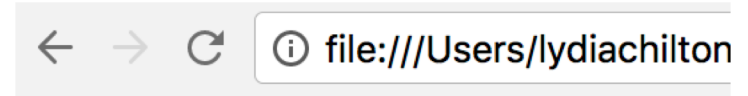
And me!

And me!

And me!

Where in the code should we add the dynamic behavior?

HTML

```
61    <body>
62
63        <button id="counter" class="btn btn-primary"></button>
64
65    </body>
```

JavaScript

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64        })
65    })
66
```
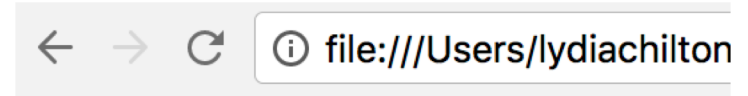
Static Button (0)

Create a new button when I am clicked!

Click me

And me!

And me!

And me!

And me!

And me!

And me!

How did we create the button in JavaScript?

HTML

```
61    <body>
62
63        <button id="counter" class="btn btn-primary"></button>
64
65    </body>
```
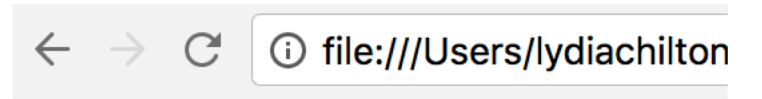
file:///Users/lydiachilton

Static Button (0)

JavaScript

```
42    function createButton(){
43        var new_button = $("<button>")
44        $(new_button).text("dynamic button "+Date.now())
45    }
```

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64            createButton()
65        })
66    })
67
```

First, add a div to HTML to contain the new buttons.

HTML

```
61    <body>
62
63            <button id="counter" class="btn btn-primary"></button>
64            <br><br>
64            <div id="updates"></div>
65    </body>
66
```

file:///Users/lydiachilton

Static Button (0)

JavaScript

```
42    function createButton(){
43        var new_button = $("<button>")
44        $(new_button).text("dynamic button "+Date.now())
45    }
46
```

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64            createButton()
65        })
66    })
67
```

Add widget to UI dynamically

HTML

```html
61  <body>
62
63          <button id="counter" class="btn btn-primary"></button>
            <br><br>
64          <div id="updates"></div>
65  </body>
66
```

Static Button (2)

dynamic button 1519059719092 | dynamic button 1519059720090

JavaScript

```javascript
44  function createButton(){
45      var new_button = $("<button>")
46      $(new_button).text("dynamic button "+Date.now())
47      $("#updates").append(new_button)
48  }
```

```javascript
61  $(document).ready(function(){
62      $("#counter").click(function(){
63          // increment the counter
64          createButton()
65      })
66  })
67
```

How do we create a line break *dynamically*?

HTML

```
61  <body>
62
63          <button id="counter" class="btn btn-primary"></button>
            <br><br>
64          <div id="updates"></div>
65  </body>
```
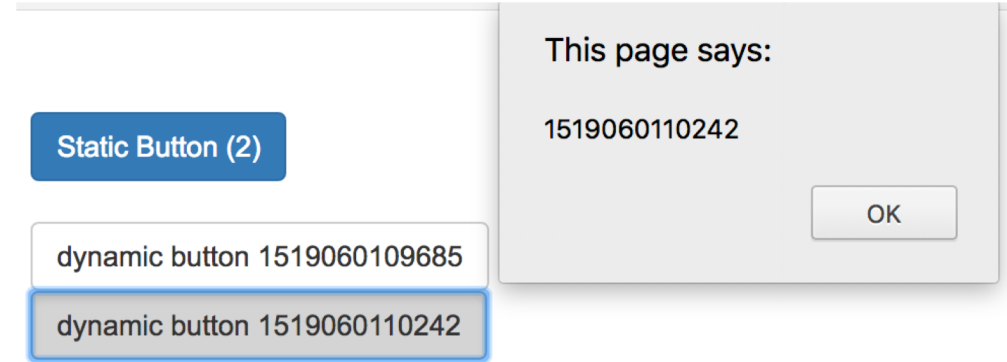
Static Button (2)

dynamic button 1519059891686
dynamic button 1519059892439

JavaScript

```
44  function createButton(){
45      var new_button = $("<button>")
46      $(new_button).text("dynamic button "+Date.now())
47      $("#updates").append(new_button)
48      $("#updates").append("<br>")
49  }
```

```
61  $(document).ready(function(){
62      $("#counter").click(function(){
63          // increment the counter
64          createButton()
65      })
66  })
67
```

How do we create a **bootstrap** button dynamically?

HTML

```
61    <body>
62
63            <button id="counter" class="btn btn-primary"></button>
63            <br><br>
64            <div id="updates"></div>
65    </body>
66
```

Static Button (2)

dynamic button 1519060044460

dynamic button 1519060044905

JavaScript

```
44    function createButton(){
45        var new_button = $("<button class='btn btn-default'>")
46        $(new_button).text("dynamic button "+Date.now())
47        $("#updates").append(new_button)
48        $("#updates").append("<br>")
49    }
```

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64            createButton()
65        })
66    })
67
```

Where do we create a click event *dynamically*?

HTML

```
61  <body>
62
63          <button id="counter" class="btn btn-primary"></button>
            <br><br>
64          <div id="updates"></div>
65  </body>
```

Static Button (2)

dynamic button 1519060044460

dynamic button 1519060044905

JavaScript

```
44  function createButton(){
45      var new_button = $("<button class='btn btn-default'>")
46      $(new_button).text("dynamic button "+Date.now())
47      $("#updates").append(new_button)
48      $("#updates").append("<br>")
49  }
```

```
61  $(document).ready(function(){
62      $("#counter").click(function(){
63          // increment the counter
64          createButton()
65      })
66  })
67
```

How do we create a click event *dynamically*?

HTML

```html
61  <body>
62
63          <button id="counter" class="btn btn-primary"></button>
64          <br><br>
65          <div id="updates"></div>
66  </body>
```

Static Button (2)

This page says:

1519060110242

OK

dynamic button 1519060109685

dynamic button 1519060110242

JavaScript

```javascript
44  function createButton(){
45
46      var new_button = $("<button class='btn btn-default'>")
47      $(new_button).text("dynamic button "+Date.now())
48      $("#updates").append(new_button)
49      $("#updates").append("<br>")
50
51      var d = Date.now()
52      $(new_button).click(function(){ alert(d) })
53  }
```

You can create elements **statically** in HTML Or **dynamically** in JavaScript (JQuery)

Static: HTML, JavaScript onReady

```
61    <body>
62        <button id="counter" class="btn btn-primary"></button>
63        <br><br>
64        <div id="updates"></div>
65    </body>
66
```

```
61    $(document).ready(function(){
62        $("#counter").click(function(){
63            // increment the counter
64            createButton()
65        })
66    })
67
```

Dynamic: All JavaScript

```
44    function createButton(){
45
46        var new_button = $("<button class='btn btn-default'>")
47        $(new_button).text("dynamic button "+Date.now())
48        $("#updates").append(new_button)
49        $("#updates").append("<br>")
50
51        var d = Date.now()
52        $(new_button).click(function(){ alert(d) })
53    }
```

Static Button (2)

dynamic button 1519060109685

dynamic button 1519060110242

This page says:

1519060110242

OK

Widgets and Events

Basic elements for users to interact with your UI

Buttons are one type of **widget**
the main event they can respond to is **clicks**.


```
$("#compose").click(function(){
    //compose new email
});
```

Every time a button is clicked, a click event fires.
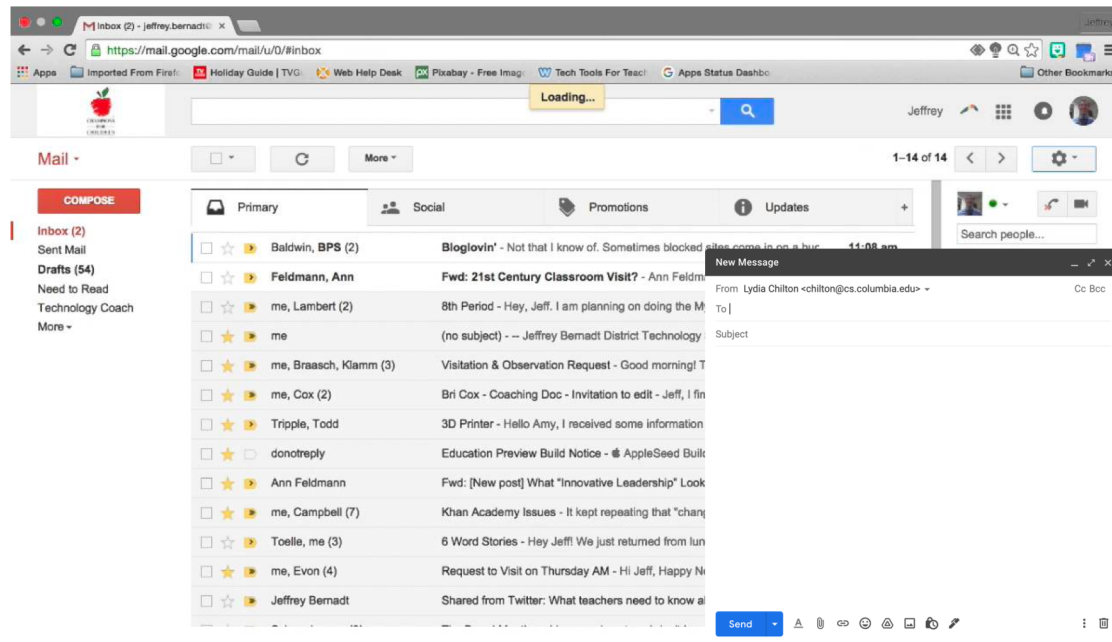

```
$("#compose").click(function(){
    //compose new email
});
```

This code listens for the click event fire and does something is called the "click handler" (more generally: "event handler")

Text Input interaction: What event fires?
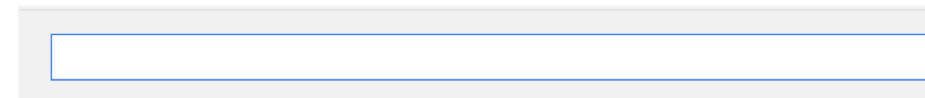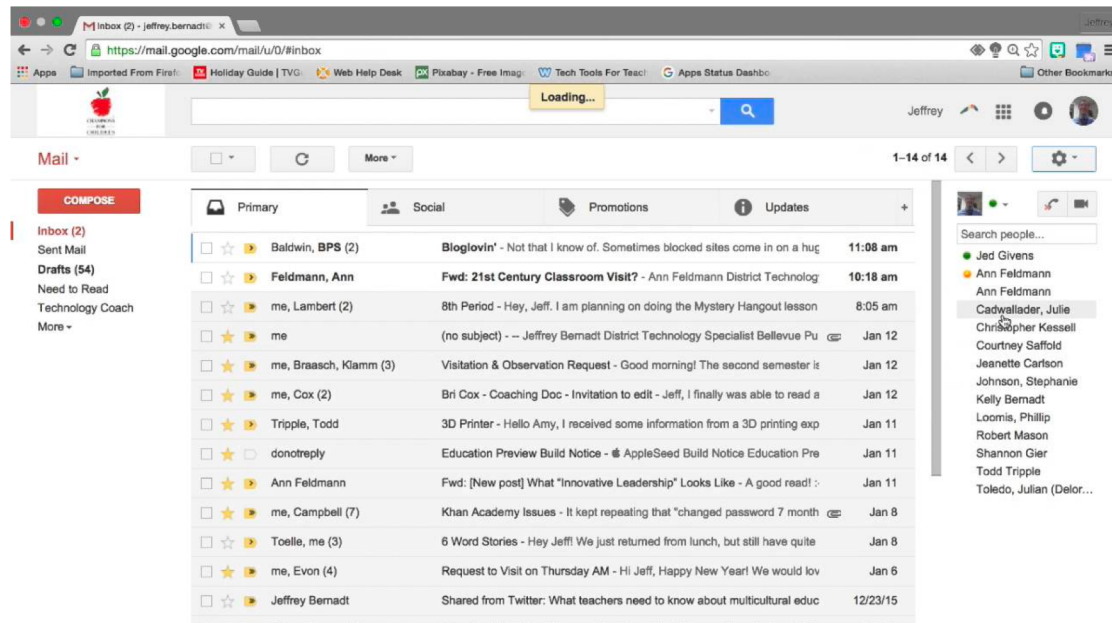
"Keypress" event

Radio Input interaction: What event fires?
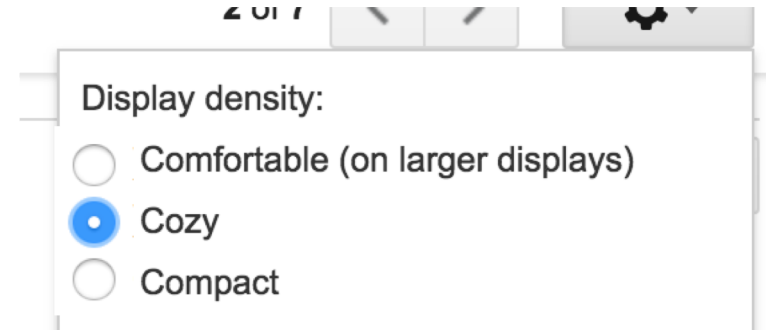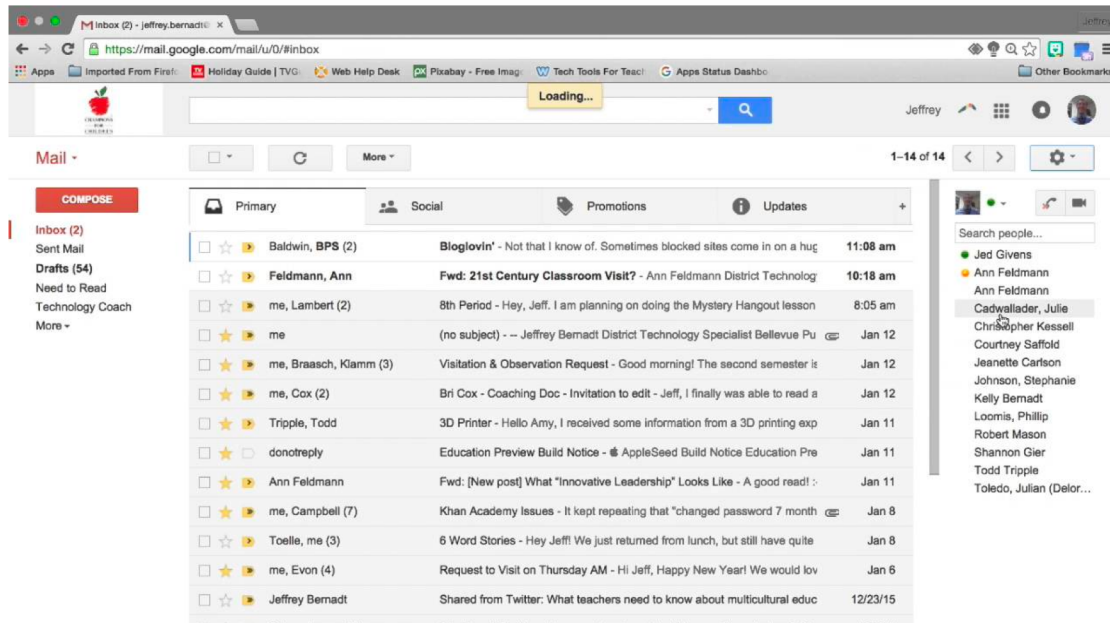
"Change" event

Select Element interaction: What event fires?
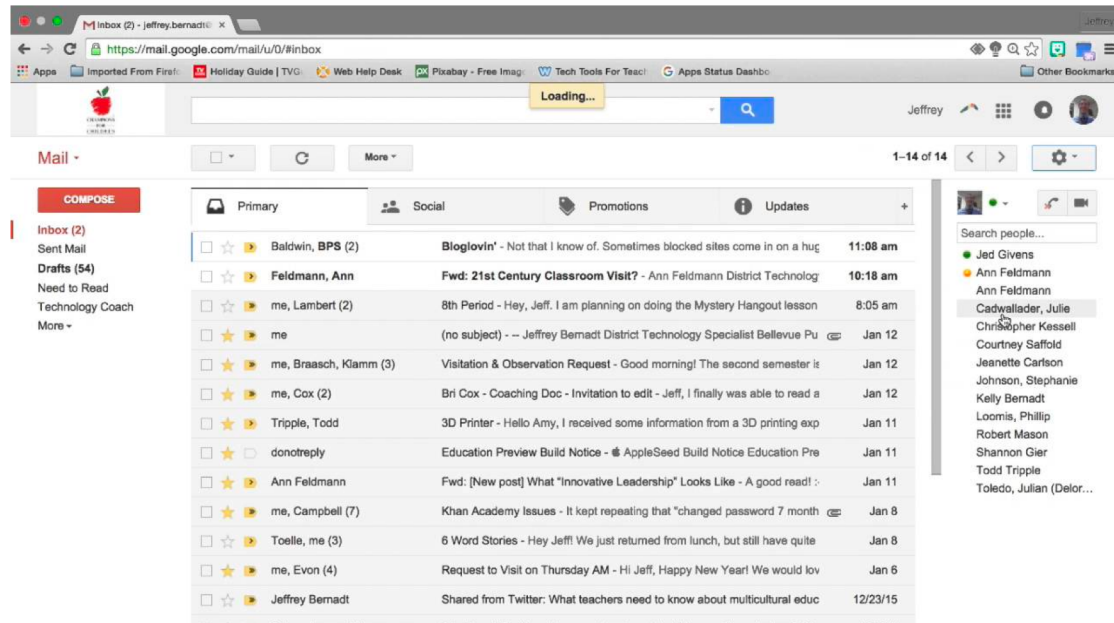
"Change" event

Dropdown interaction: What event fires?
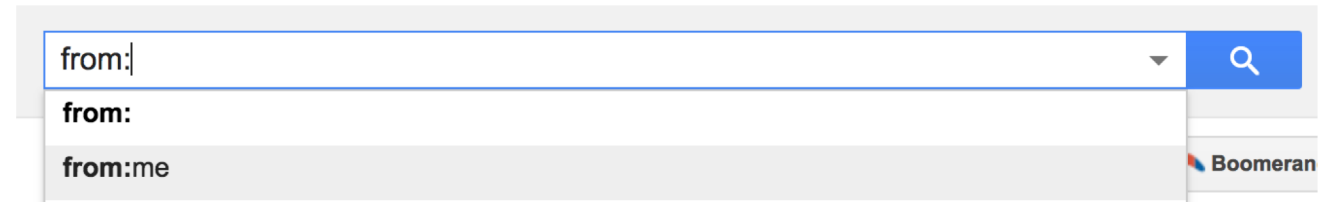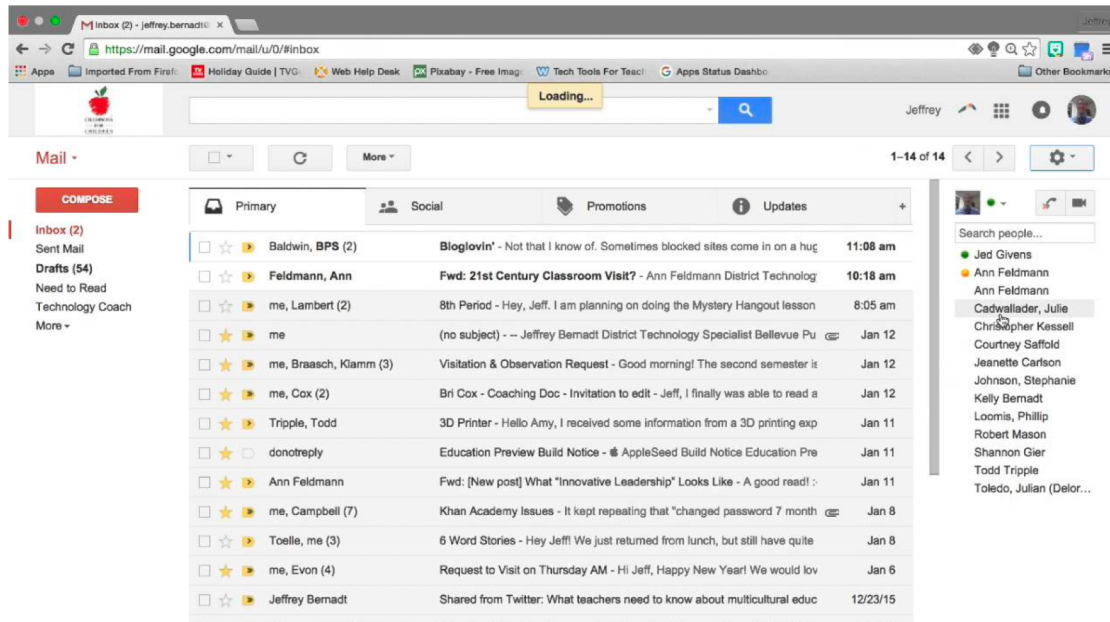
"Select" event

Drag and Drop interaction: What events fire?
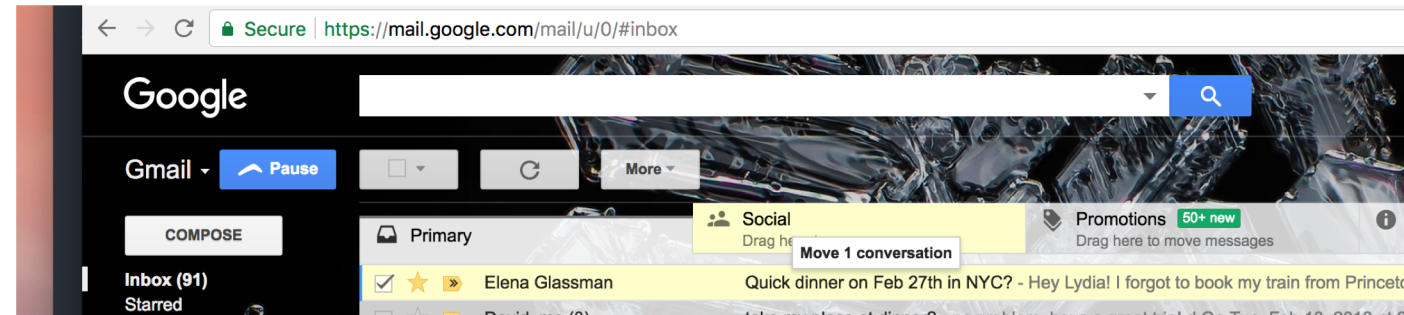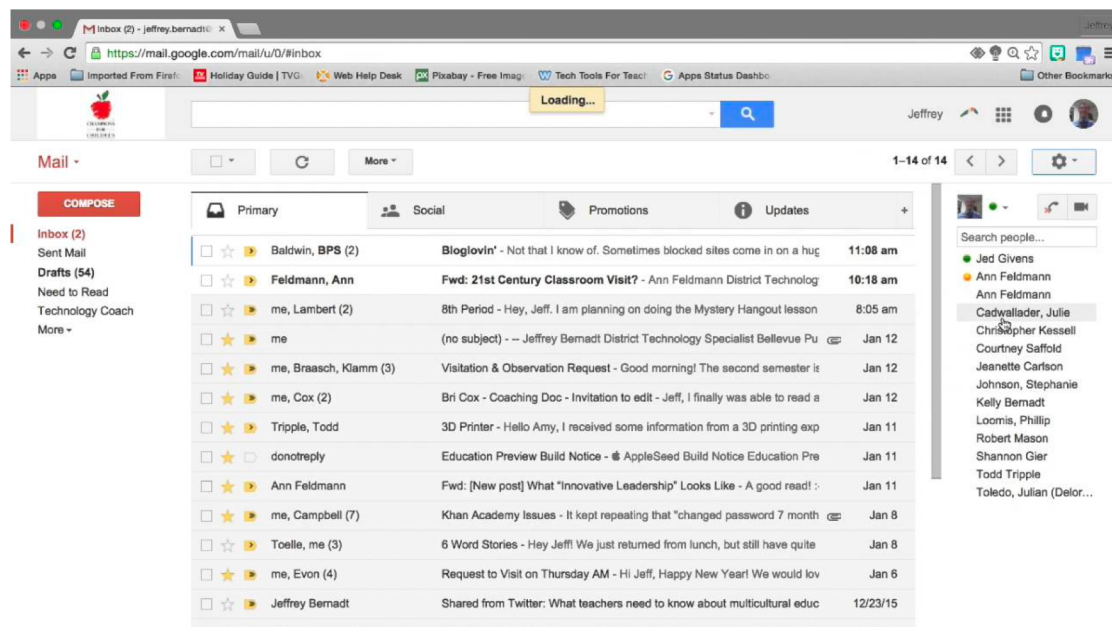
- "Drag" event
- "Drop" event

Widgets are standardized low-level interaction interfaces that trigger events

When you create a widget...

The **appearance** is standardized,
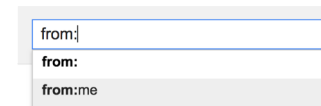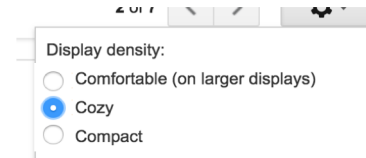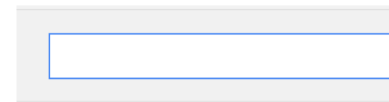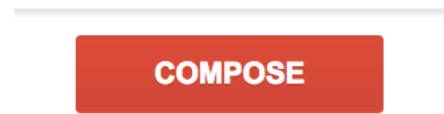
The **types of events** it responds to are standardized

"Click"
"hover"

"Keypress"

"Change"

"Select"
"Search"
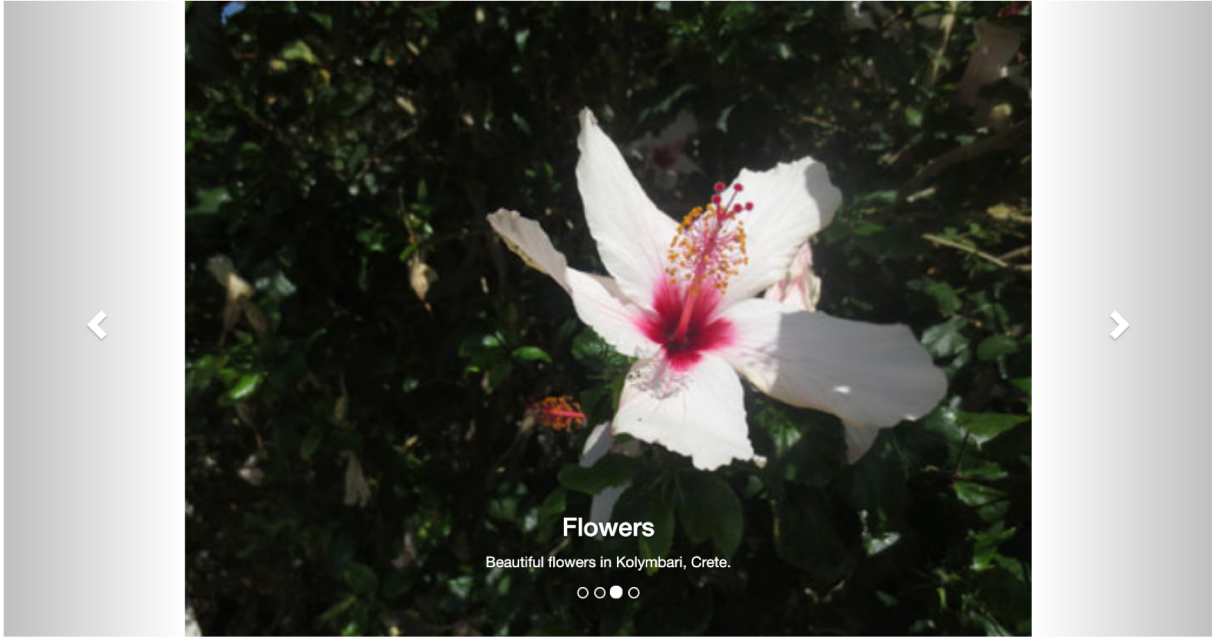
But the actions taken after an event is fired, are not standardized

Widgets can also be big

Because you did not program them yourself, widgets may appear and act differently on different devices
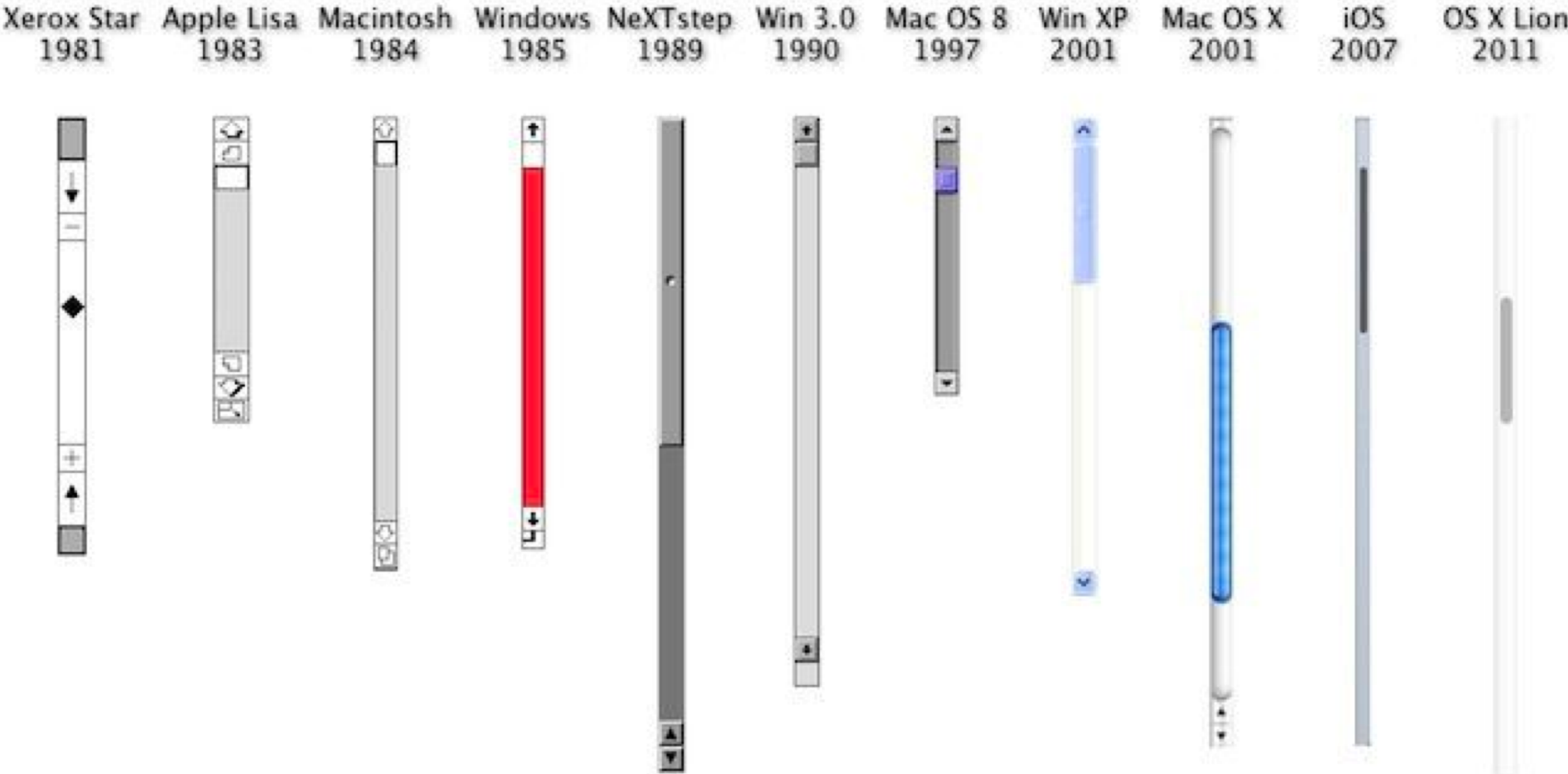
Pros and Cons of Standardization

Things that have become standardized
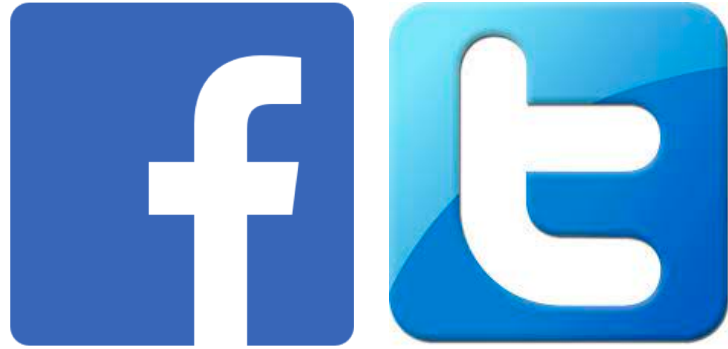
Because people people copy successful designs
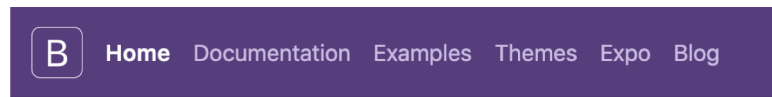
Because one version domains the market

Because people create good, reusable solutions

Things that have not become standardized
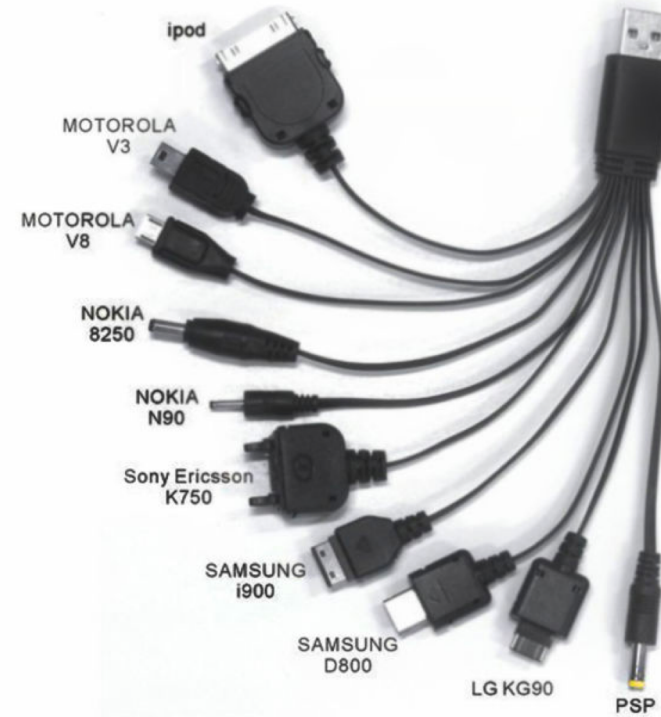
"colour", "honour",
"cheque", "connexion"

Old things that got standardized
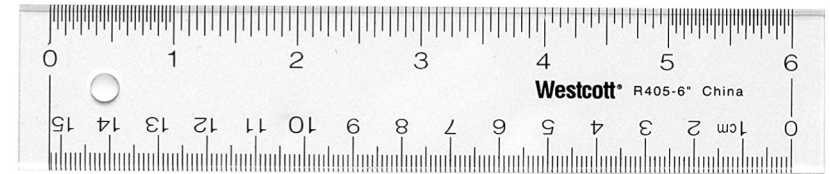
What's good about standardization?

Standardized

Non-Standardized

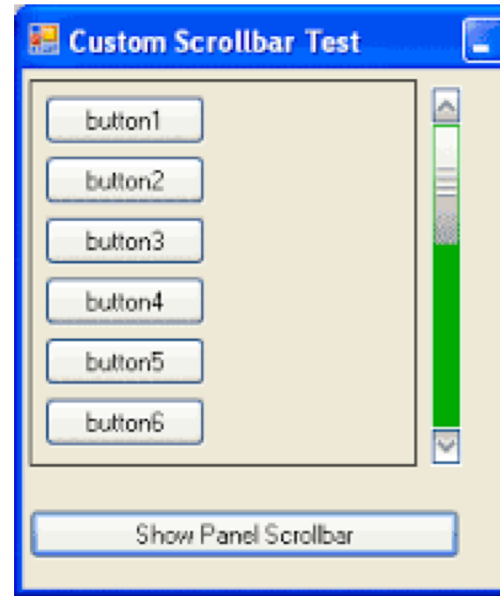# What's bad about standardization?

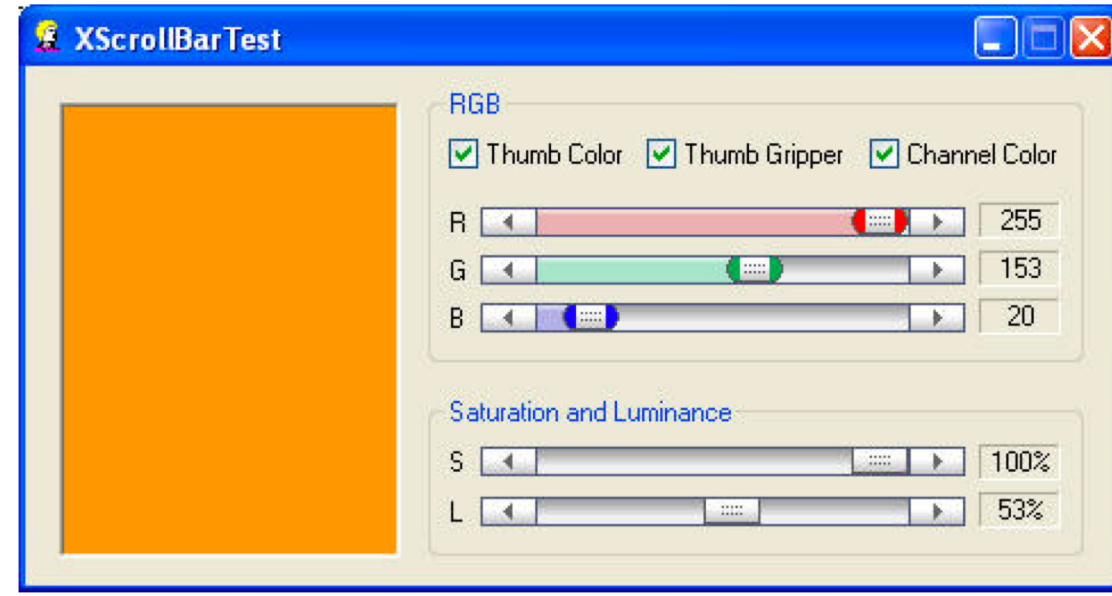Standardized

Non-Standardized
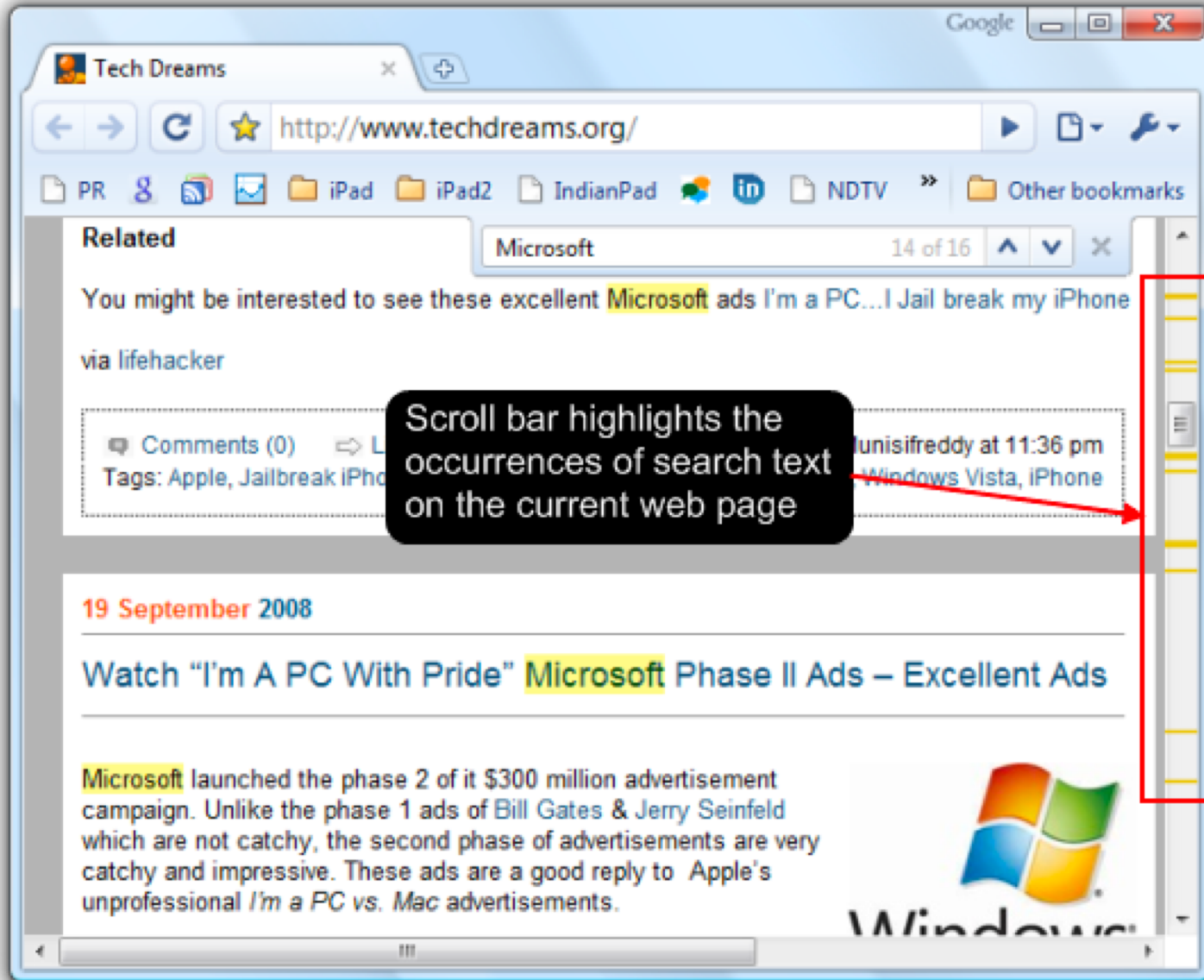
Widgets allow customization

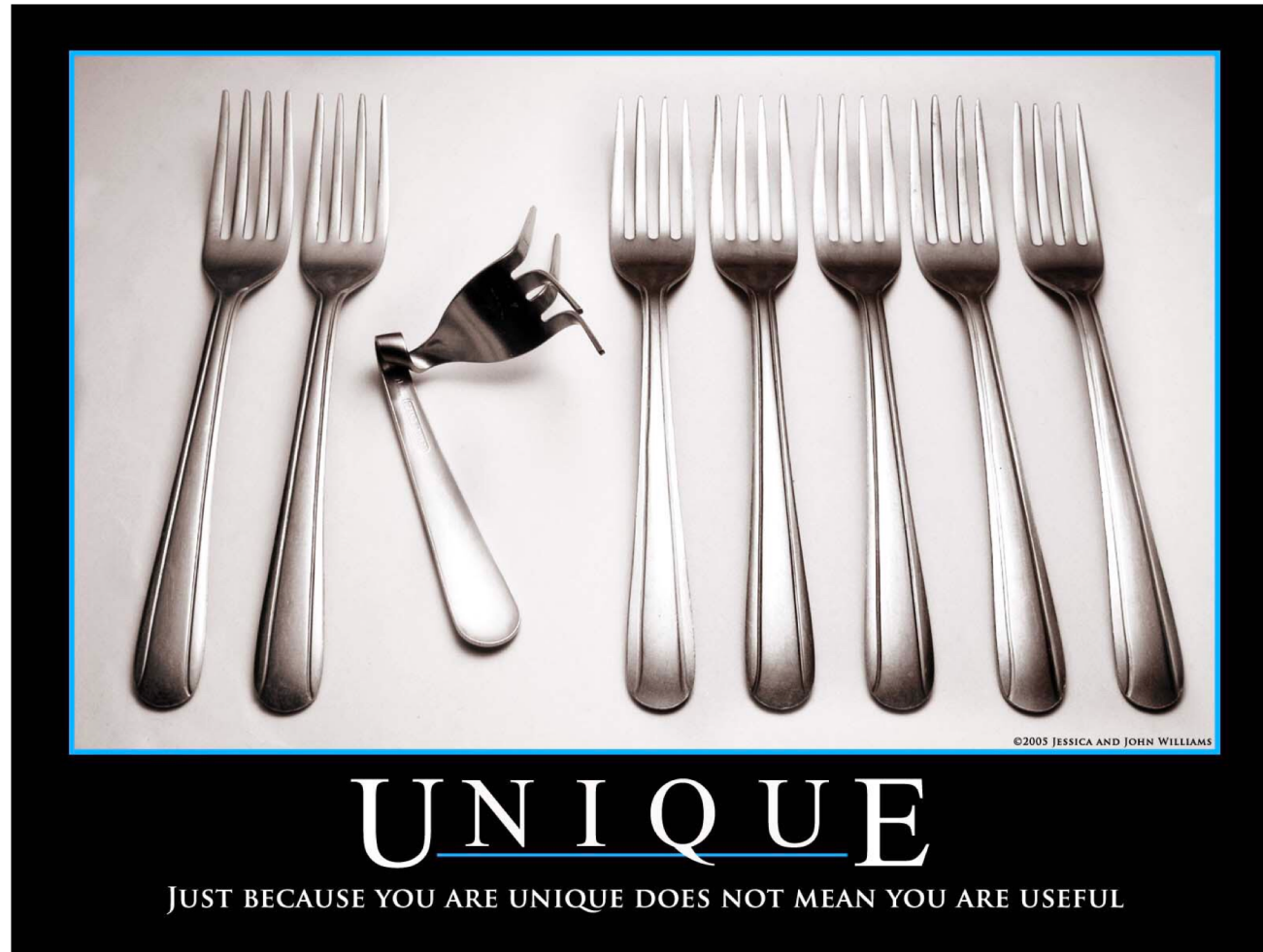Customizable scroll bars

Bad use of customization.

Good use of customization.
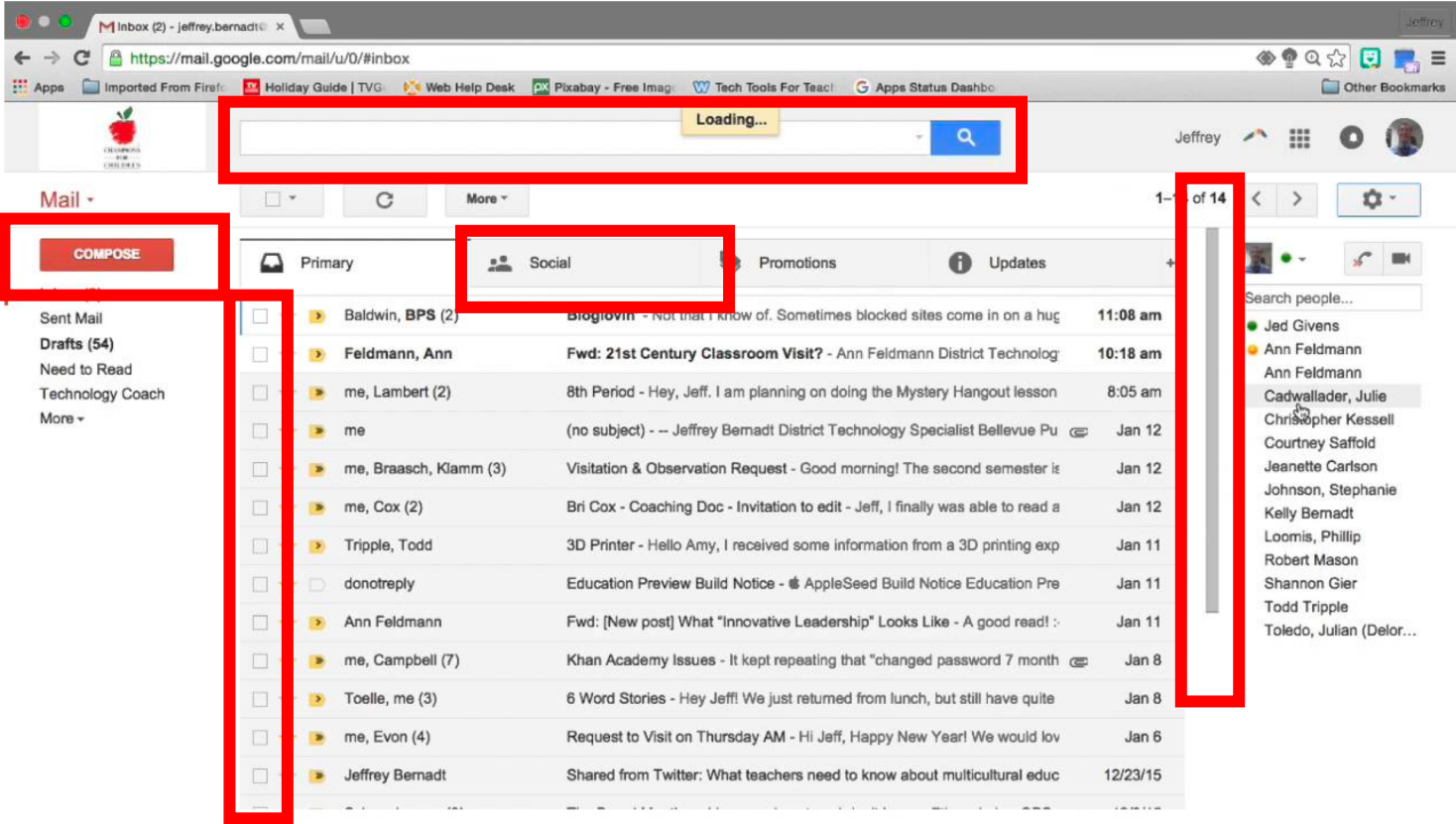
Widgets allow customization
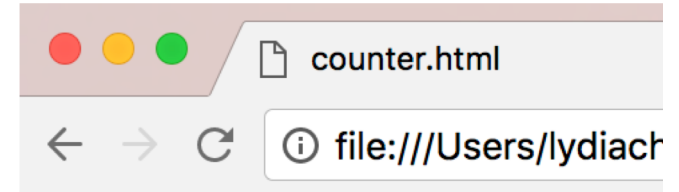
Use your powers of customization wisely.

Summary

We interact with webpages through **widgets**: Elements with standardized appearance and events
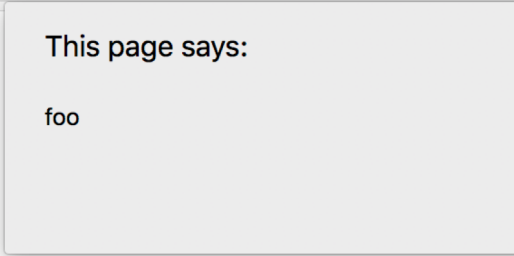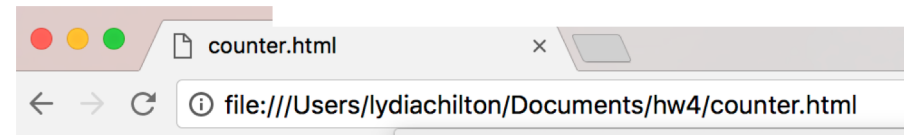
Creating Interactions on the web has two parts:

1. Program the interface and style in HTML & CSS

```
31  <body>
32
33      <button id="counter" class="btn btn-primary">Counter (0)</button>
34
35  </body>
36
```

counter.html

← → C ⓘ file:///Users/lydiach
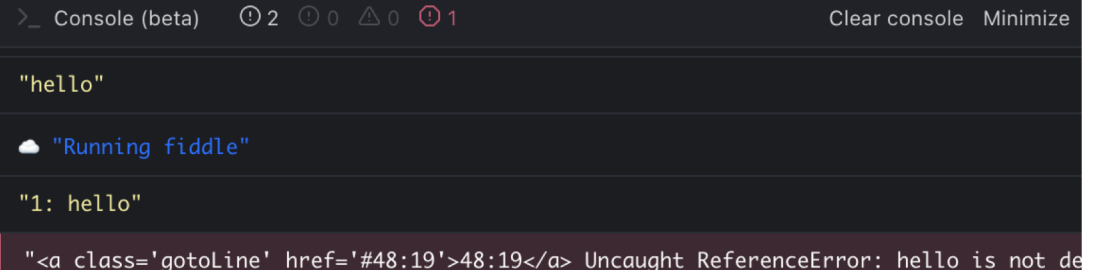
Counter (0)

2. Program interactions is JavaScript

```
25
26  $(document).ready(function(){
27      $("#counter").click(function(){
28          alert("foo")
29      })
30  })
31
```

counter.html   ×

← → C ⓘ file:///Users/lydiachilton/Documents/hw4/counter.html

This page says:

foo

Counter (0)

In JavaScript,
let and **const** are good ways to declare variables.

(var is globally scoped, and can get you into trouble)

```
1    const debug_mode = true
2
3  ▾ if(debug_mode){
4      let hello = "hello"
5      console.log("1: "+hello)
6    }
7
8    console.log("2: "+hello) // not declared
```

```
>_  Console (beta)    ⊘2  ⊘0  ⚠0  ⊘1                    Clear console  Minimize

"hello"

☁ "Running fiddle"

"1: hello"

"<a class='gotoLine' href='#48:19'>48:19</a> Uncaught ReferenceError: hello is not de
```
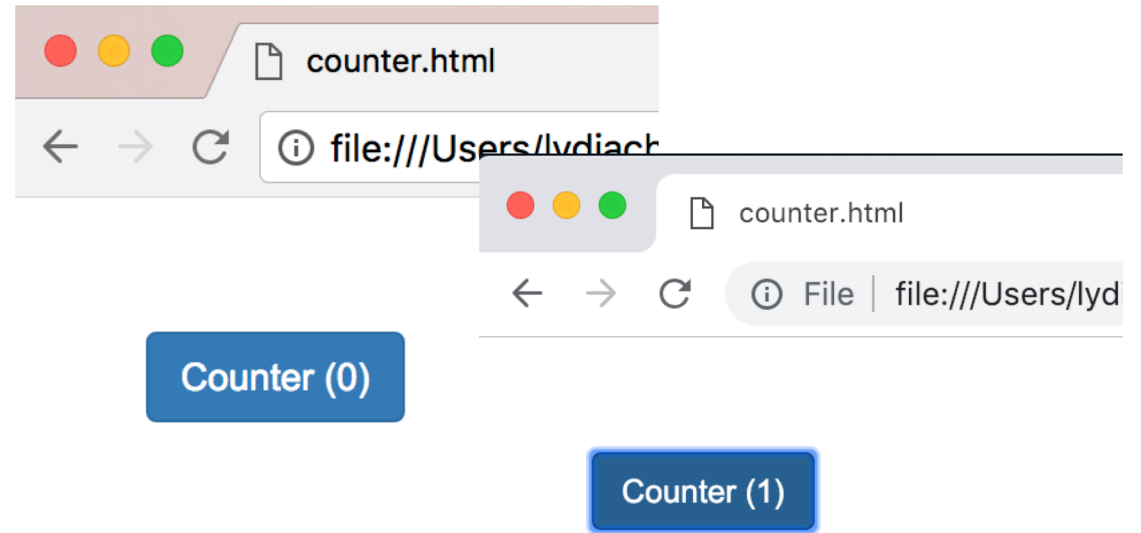
Let is block scoped, and can be re-assigned.
Const is block scoped and cannot be re-assigned.

Good style of attaching events in JQuery

```html
<html>
<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boo
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" crossorig

  <script>
    var count = 0

    function incrementCount(c) {
      return c + 1;
    }


    $(document).ready(function(){
        $("#counter").click(function(){
            count = incrementCount(count)
            $("#counter").html("Counter ("+count+")")
        })
    })

  </script>


</head>



<body>
  <button id="counter" class="btn btn-primary">Counter (0)</button>
</body>

</html>
```

counter.html
file:///Users/lydiach

counter.html
File | file:///Users/lyd

Counter (0)

Counter (1)

1. Uses Jquery (not pure JavaScript)

2. Attaches click handler as in the <script>
$(element).click(...)
(doesn't attach in HTML)

2. Uses $(document).ready(...)

JQuery is a JavaScript Library that make JavaScript easier (and standard across browsers)

JavaScript

```javascript
document.getElementById("counter").addEventListener("click", function(){
    document.getElementById("counter").innerHTML = "Counter (0)";
});
```

JQuery

```javascript
$("#counter").click(function(){
    $("#counter").html("Counter (0)");
});
```

You can create elements **statically** in HTML Or **dynamically** in JavaScript (JQuery)

Static: HTML, JavaScript onReady

```
61  <body>
62
63      <button id="counter" class="btn btn-primary"></button>
64      <br><br>
        <div id="updates"></div>
65  </body>
66
```

```
61  $(document).ready(function(){
62      $("#counter").click(function(){
63          // increment the counter
64          createButton()
65      })
66  })
67
```

Dynamic: All JavaScript

```
44  function createButton(){
45
46      var new_button = $("<button class='btn btn-default'>")
47      $(new_button).text("dynamic button "+Date.now())
48      $("#updates").append(new_button)
49      $("#updates").append("<br>")
50
51      var d = Date.now()
52      $(new_button).click(function(){ alert(d) })
53  }
```

Static Button (2)

This page says:

1519060110242

OK

dynamic button 1519060109685

dynamic button 1519060110242

Widgets are standardized low-level interaction interfaces that trigger events

When you create a widget…

```
61    <body>
62
63          <button id="counter" class="btn btn-primary"></button>
64
65    </body>
```

The **appearance** is standardized,

Counter (0)

The **types of events** it responds to are standardized

```
50
51          $("#counter").click(function(){
52
53
54          })
```
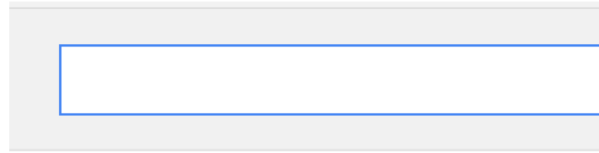
But the actions taken after an event is fired, are not standardized

```
50
51          $("#counter").click(function(){
52              count = count +1
53              setCount(count)
54          })
```
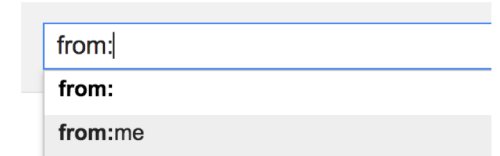
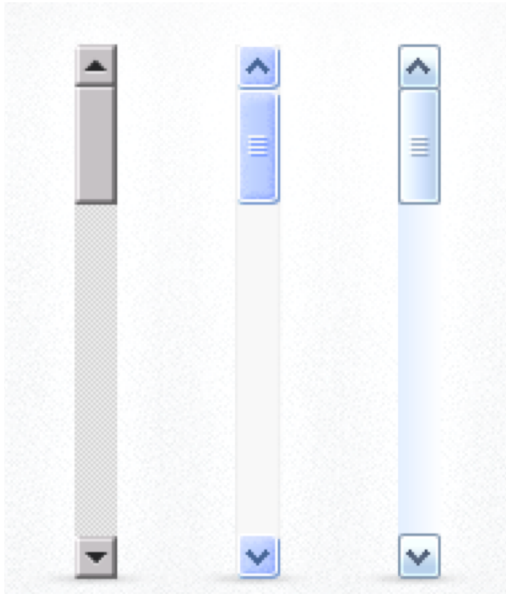# There are many types of widgets and events
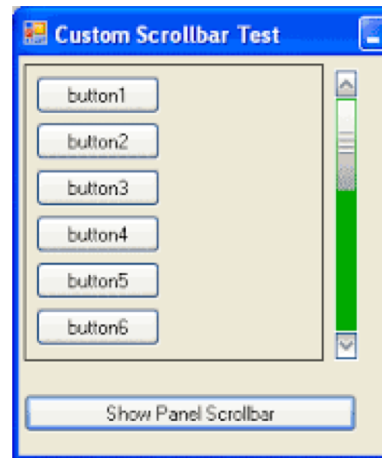

Click

Keypress
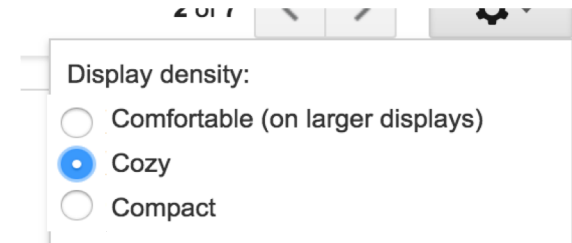
Change

Select

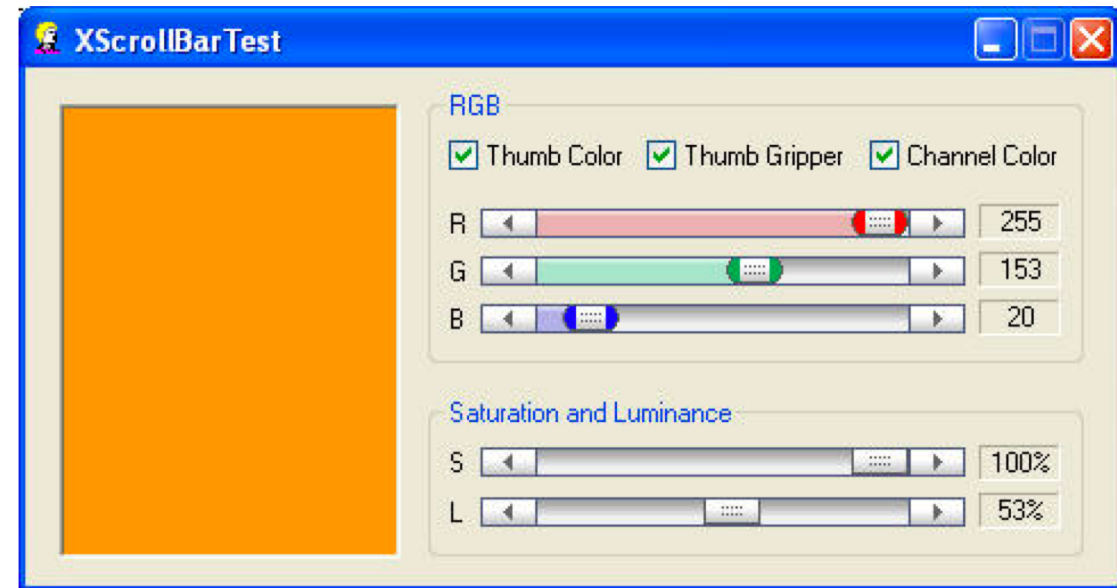# Widgets allow customization. Use it wisely.

Customizable scroll bars

Bad use of customization.
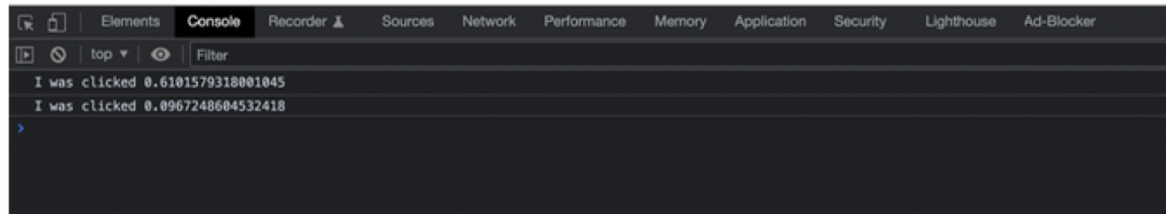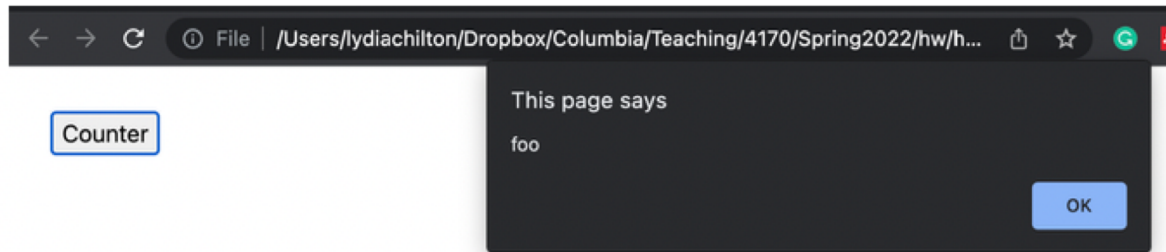
Good use of customization.

Homework 3: User Models and JavaScript

Warm up

Main

Write a tweet

Call me Ishmael. Some years ago- never

-12 Post Tweet

POSTS

chilton	Third post
chilton	Second post
chilton	First post