

Widgets and JavaScript

No screens



Prof. Lydia Chilton
COMS 4170
19 February 2018

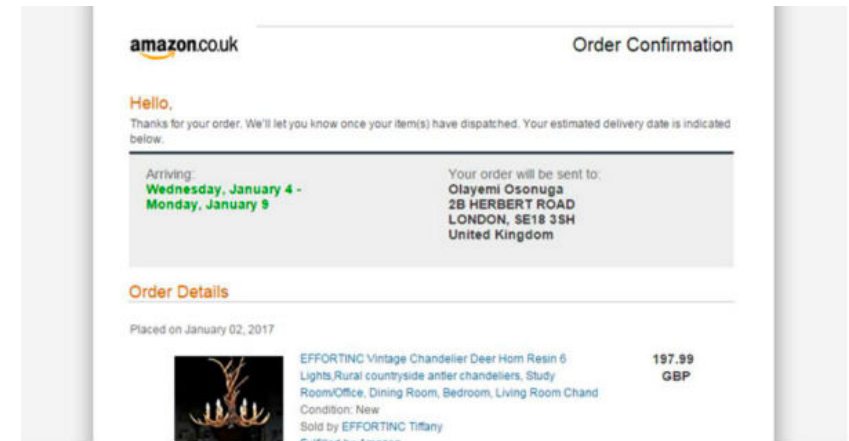
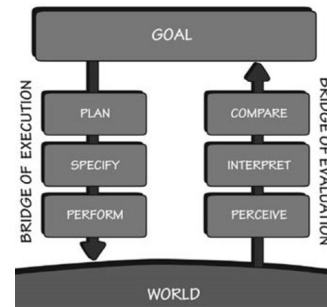
Say your name



Goal 1

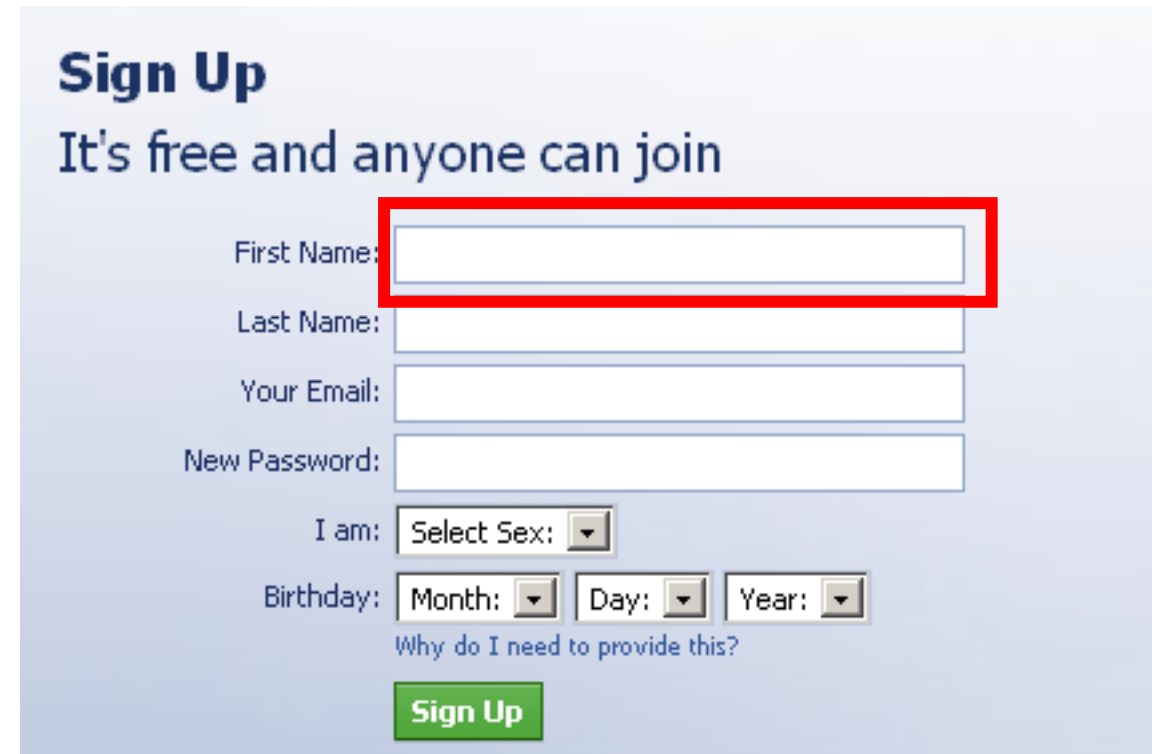
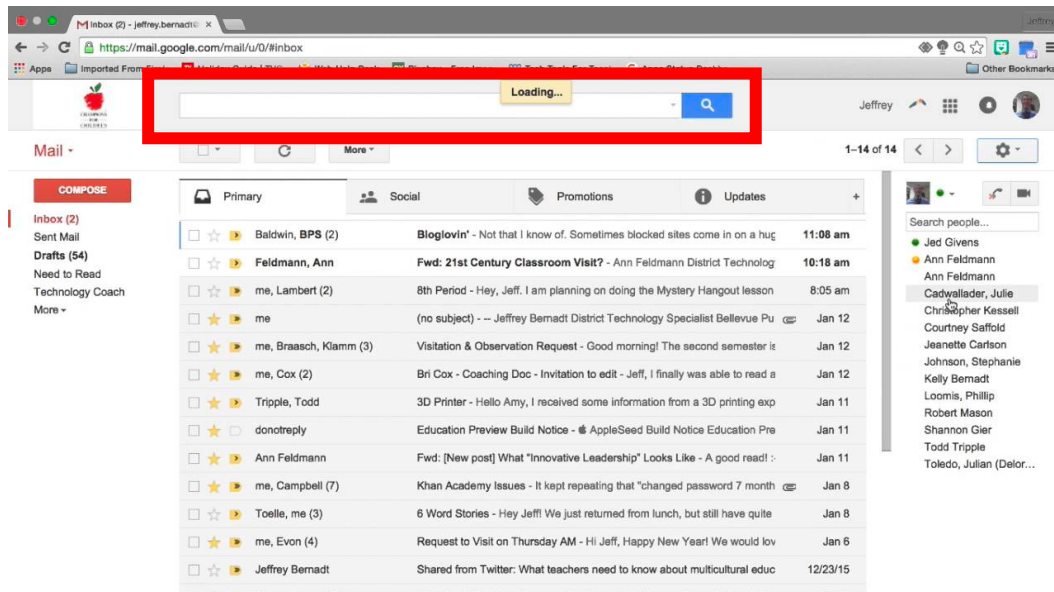
Build websites that suit the needs and abilities of users

To accomplish a goal, users must **execute** an operation and evaluate the result

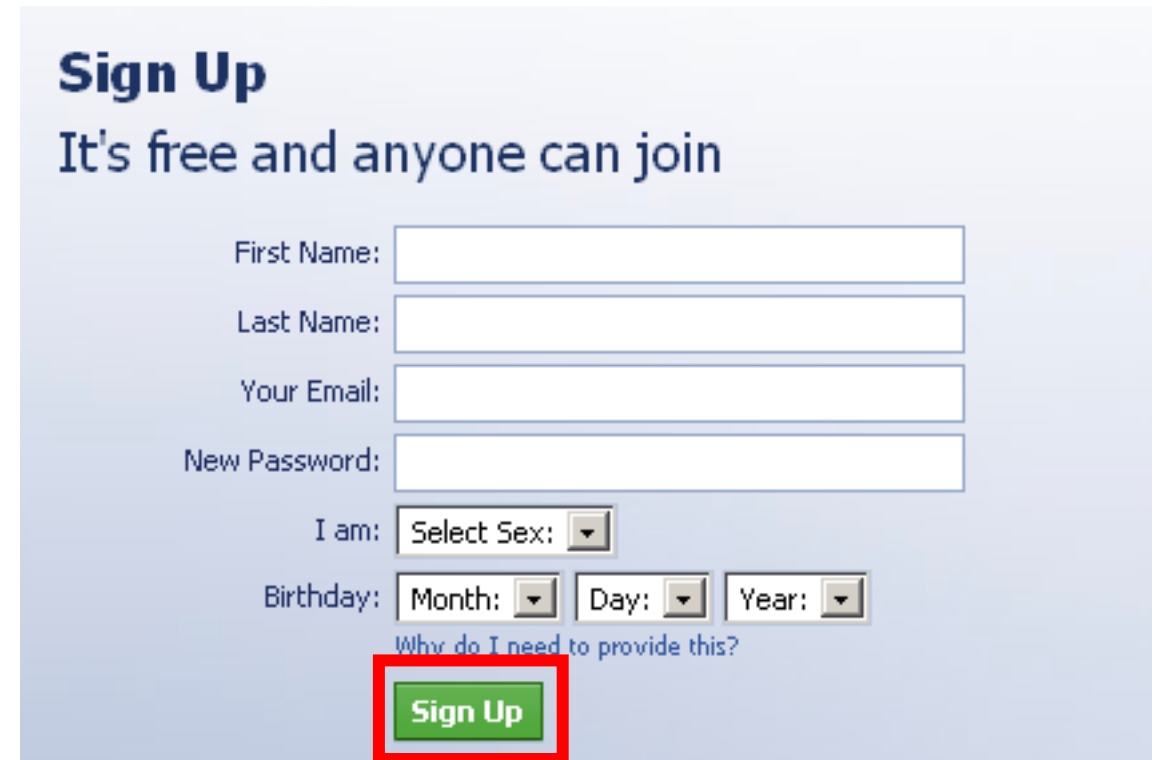
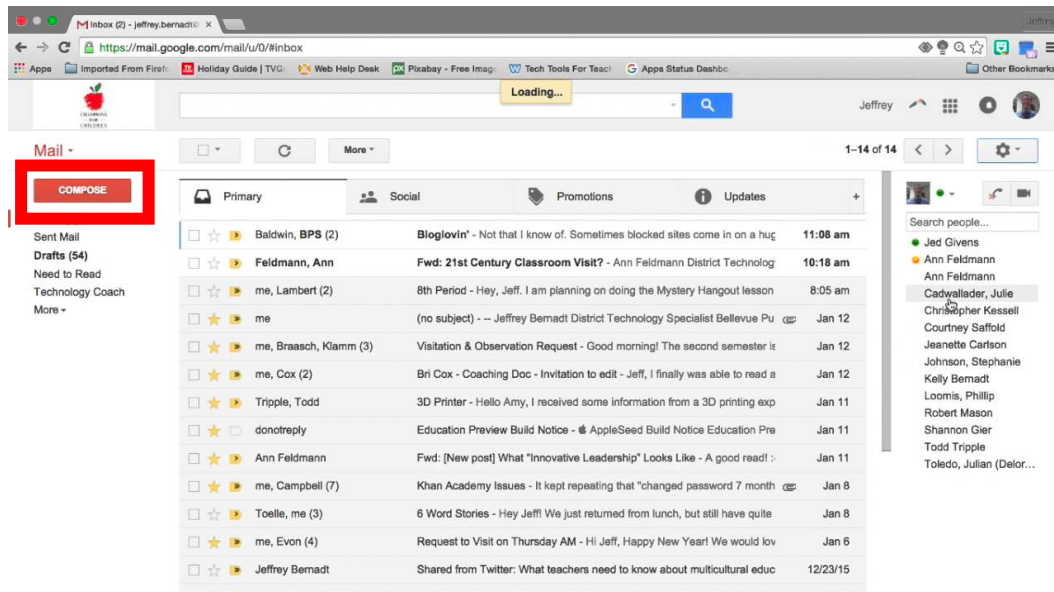


To execute an operation users must **interact with elements on the page**

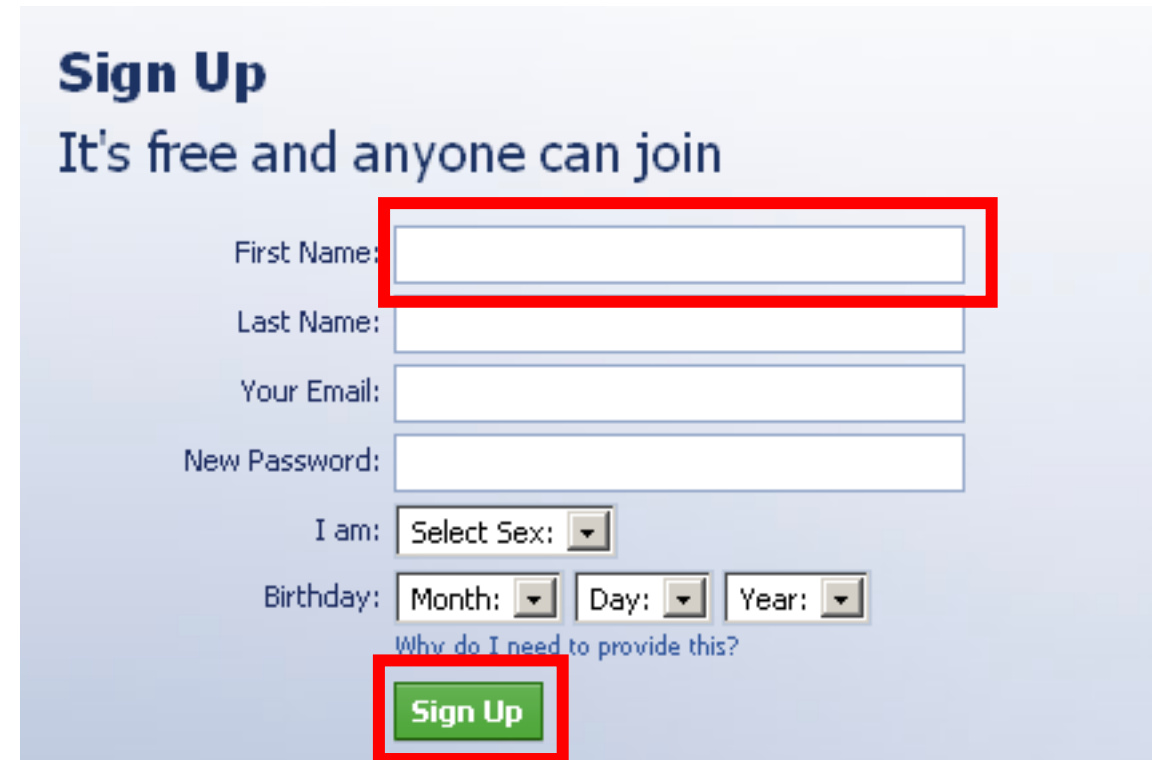
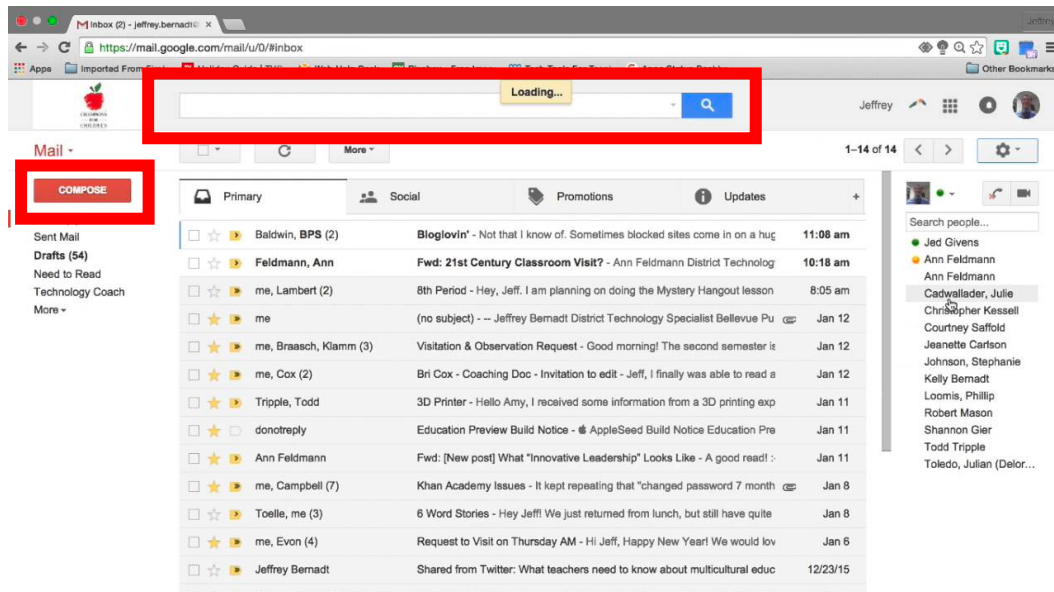
There are some standard ways we interact with elements in an interface:



There are some standard ways we interact with elements in an interface:



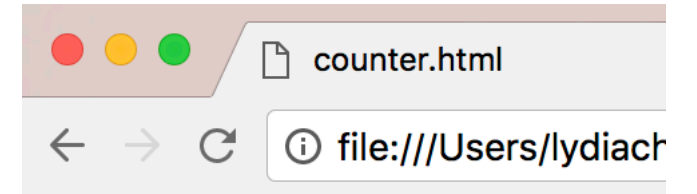
Buttons and Text Inputs are examples of widgets.



Remember the JQuery Button Counter?

HTML

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



Counter (0)

JavaScript

```
41
42 var count = 0
43
44 function setCount(count){
45     $("#counter").html("Counter (" + count + ")")
46 }
47
48 $(document).ready(function(){
49     setCount(count)
50
51     $("#counter").click(function(){
52         count = count + 1
53         setCount(count)
54     })
55 })
56
```

Widgets standardized low-level interaction interfaces

When you create a widget...

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

The **appearance** is standardized,



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

```
50
51   $("#counter").click(function(){
52     [REDACTED]
53   })
54
```

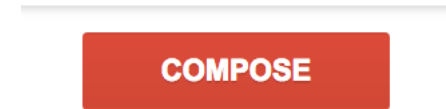
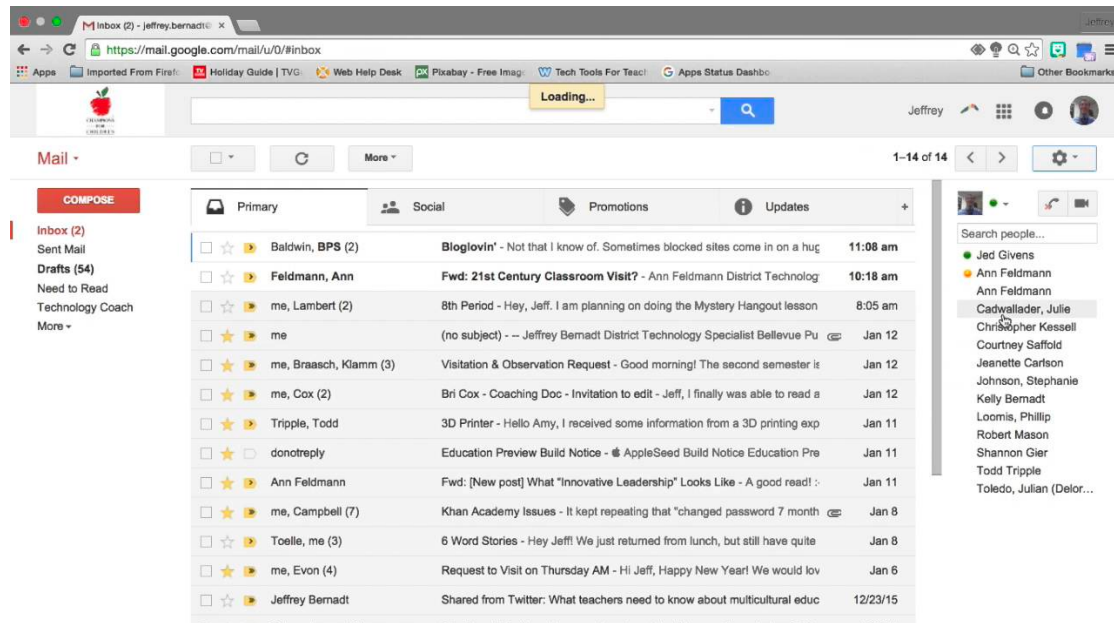
But the actions taken after an event is fired, are not standardized

```
50
51   $("#counter").click(function(){
52     count = count + 1
53     setCount(count)
54   })
```

Widgets

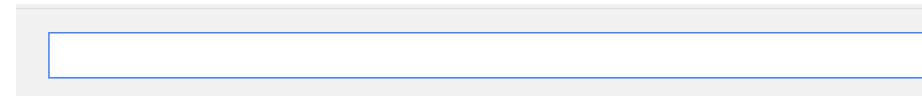
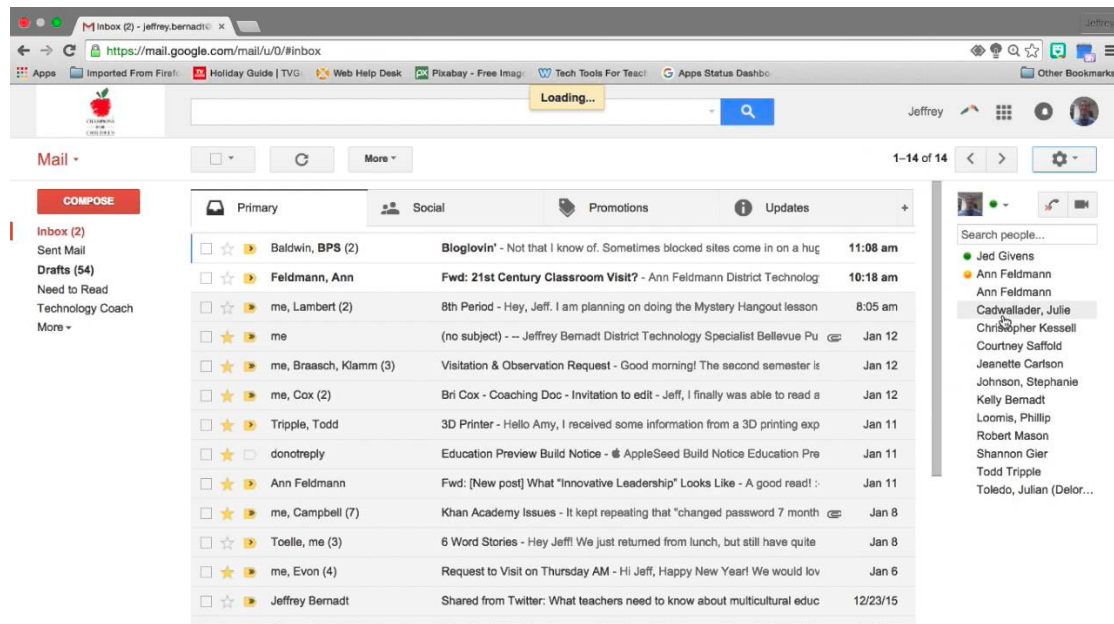
Basic elements for users to interact with your UI

Button Widget interaction: What event fires?



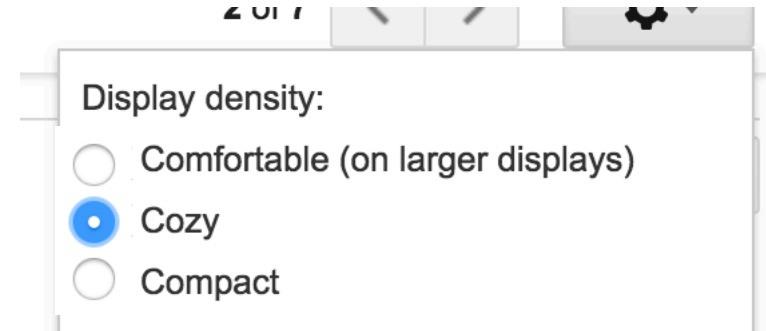
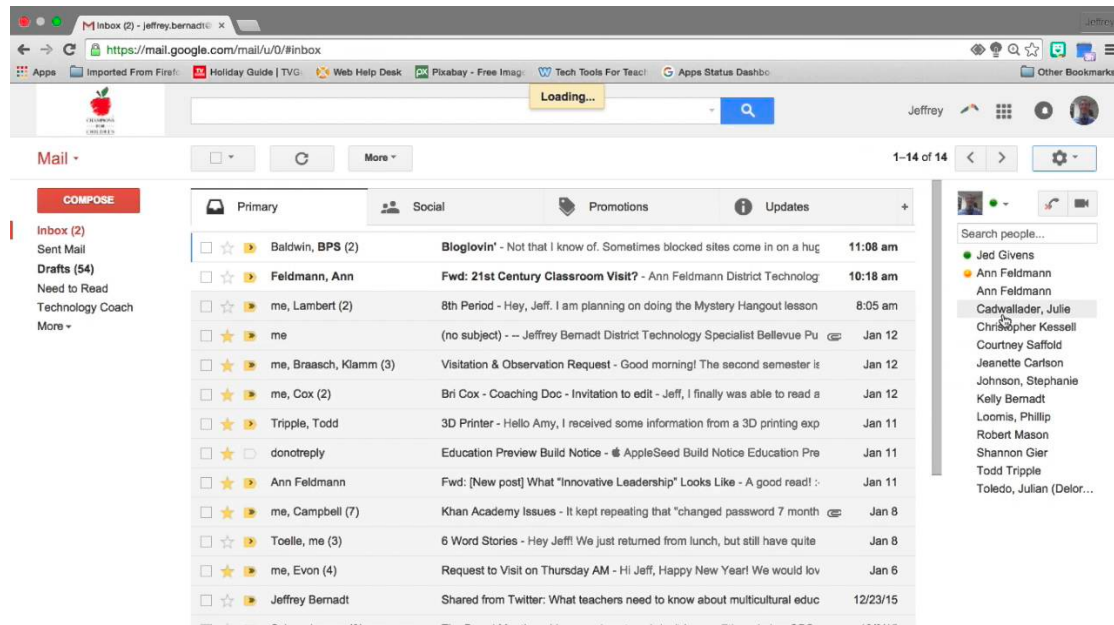
“Click event”

Text Input interaction: What event fires?



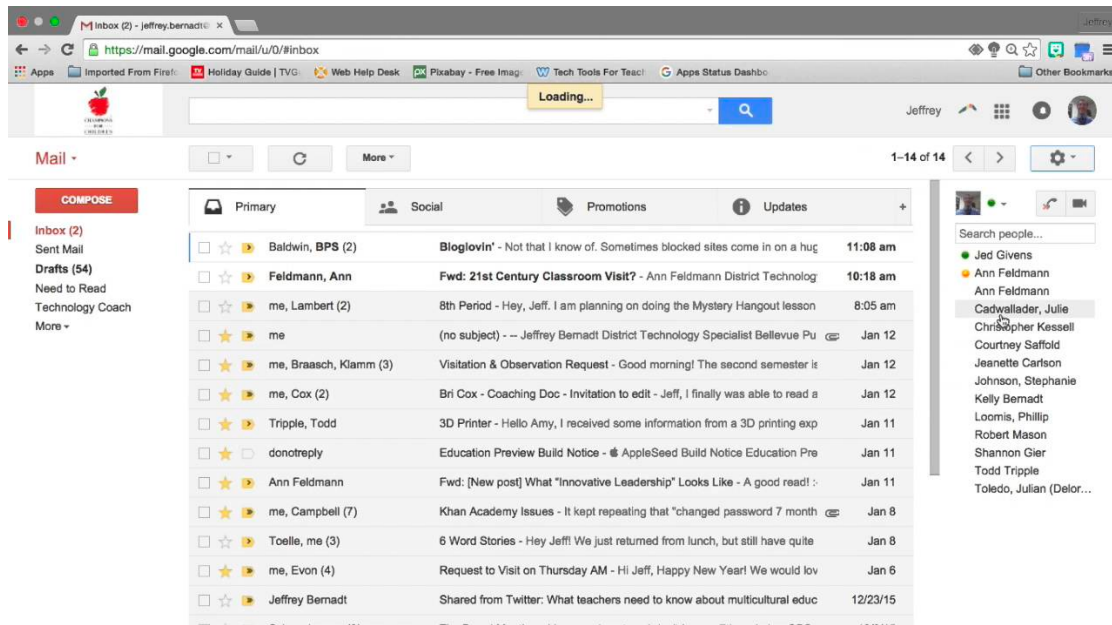
“Keypress” event

Radio Input interaction: What event fires?



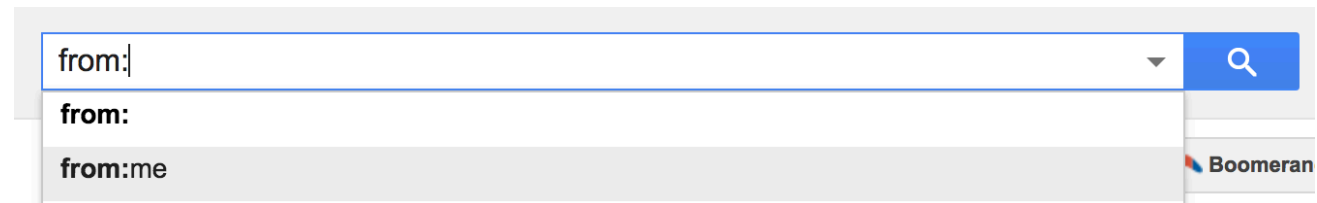
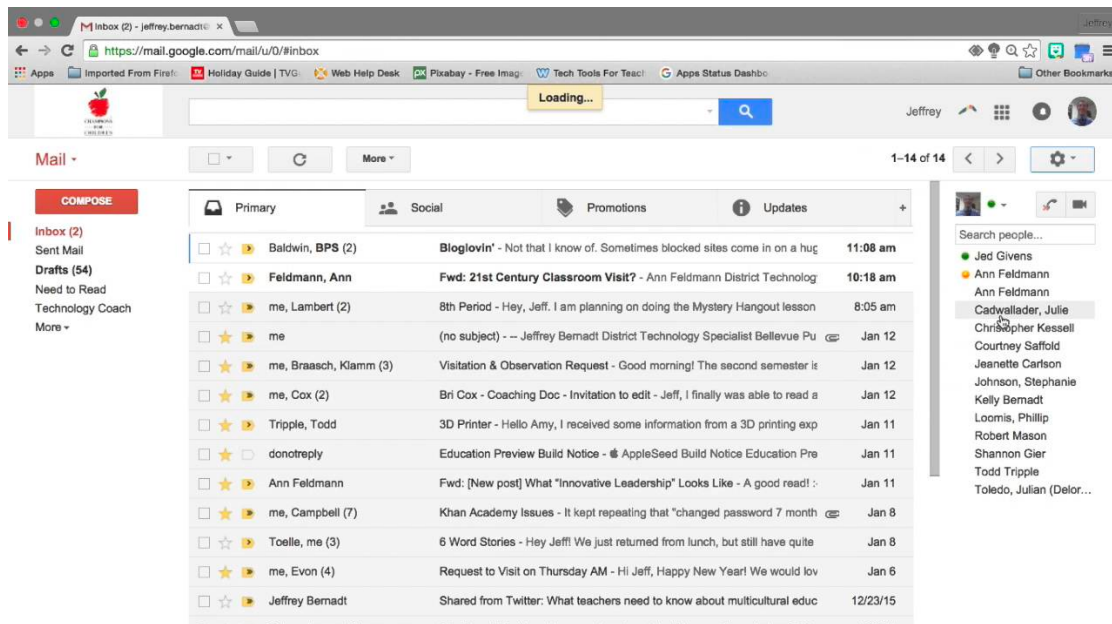
“Change” event

<select> interaction: What event fires?



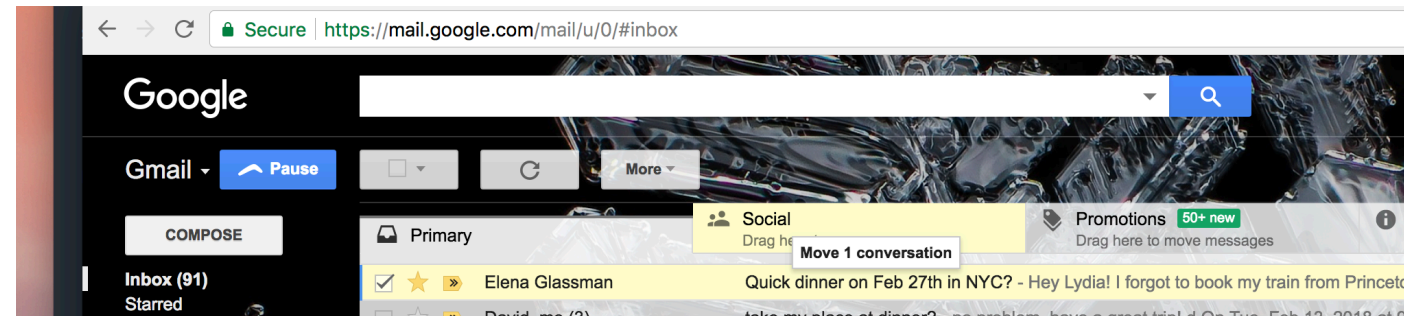
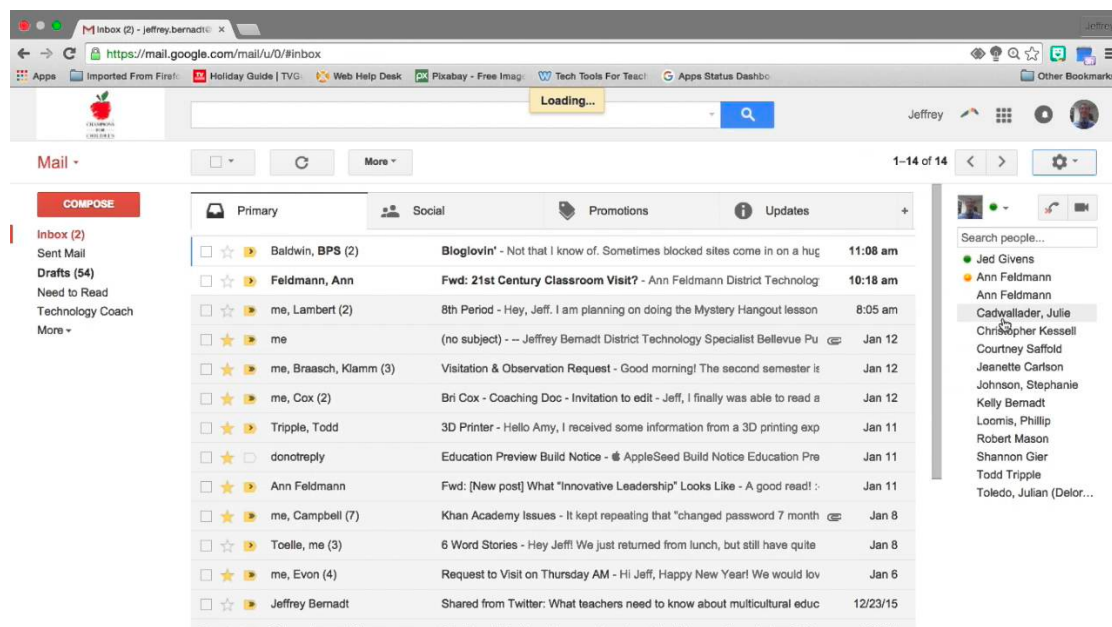
“Change” event

Autocomplete interaction: What event fires?



- “Search” event
- “Select” event

Drag and Drop interaction: What event fires?

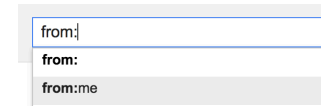
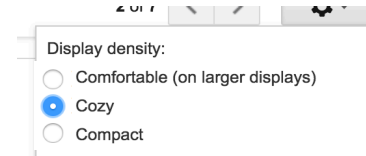
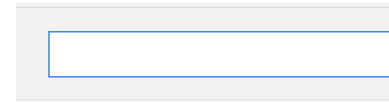
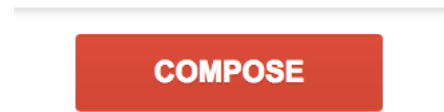


- “Drag” event
- “Drop” event

Widgets standardized low-level interaction interfaces

When you create a widget...

The **appearance** is standardized,



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

“Click”

“Keypress”

“Change”

“Search”

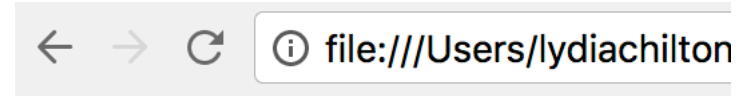
But the actions taken after an event is fired, are not standardized

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>
66
```



Static Button (0)

JavaScript

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

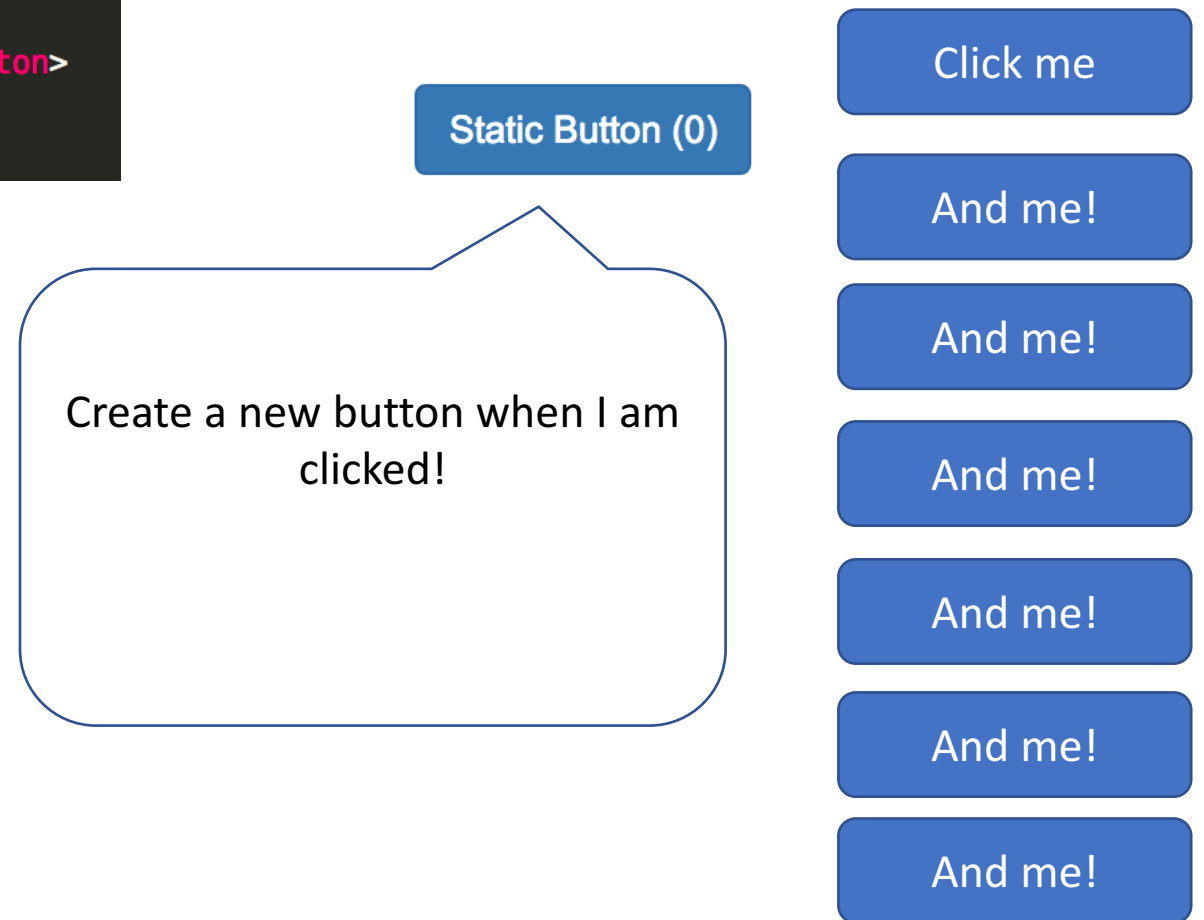
Dynamically created widget: created on demand based on user interaction.

HTML

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

JavaScript

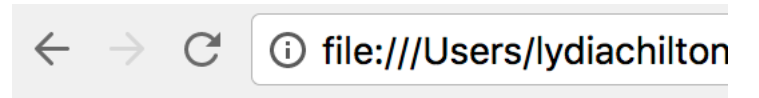
```
60
61 $(document).ready(function(){
62   $("#counter").click(function(){
63     // increment the counter
64   })
65 })
66
```



How did we create the button in JavaScript?

HTML

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



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

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

Static Button (2)

dynamic button 1519059719092

dynamic button 1519059720090

JavaScript

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

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

Where do we create a line break *dynamically*?

HTML

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

Static Button (2)

dynamic button 1519059719092

dynamic button 1519059720090

JavaScript

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

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

How do we create a line break *dynamically*?

HTML

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

Where do we create a bootstrap button *dynamically*?

HTML

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


How do we create a bootstrap button *dynamically*?

HTML

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

Where do we create a click event *dynamically*?

HTML

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

Static Button (2)

dynamic button 1519060044460

dynamic button 1519060044905

JavaScript

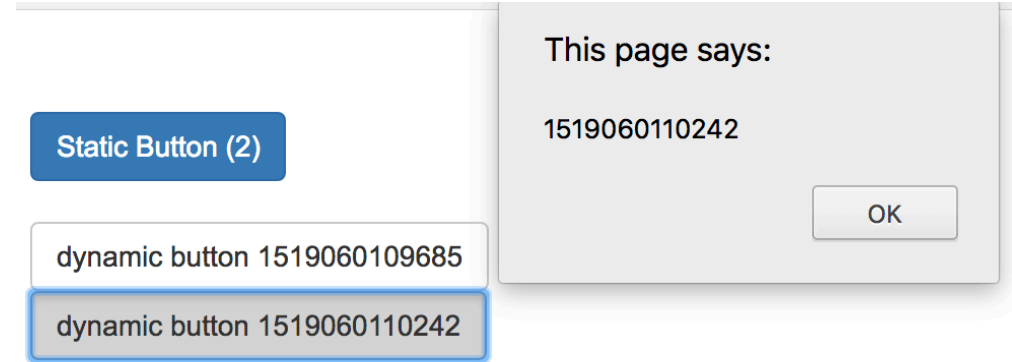
```
44 function createButton(){
45     var new_button = $("
```

```
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

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



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 }
```

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

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

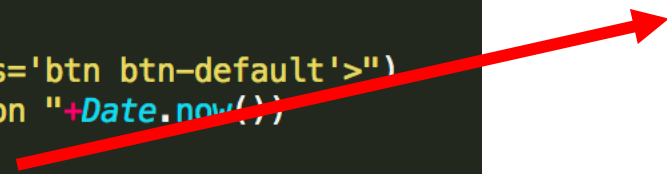
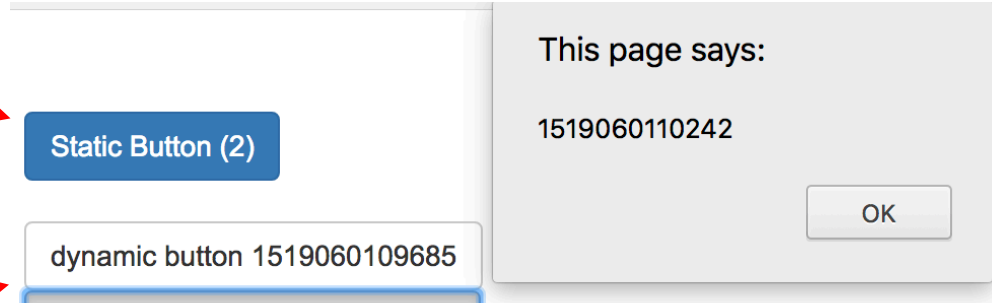
Static: HTML, JavaScript onReady

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

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

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 }
```



Pros and Cons of Standardization

What's good about standardization?

Standardized



Non-Standardized



What's **bad** about standardization?

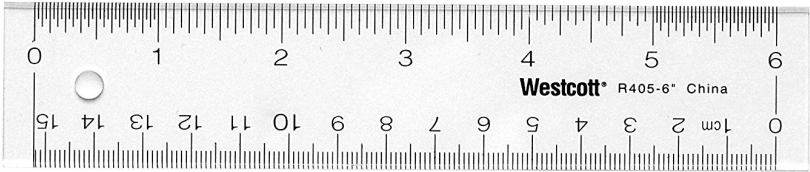
Standardized



Non-Standardized

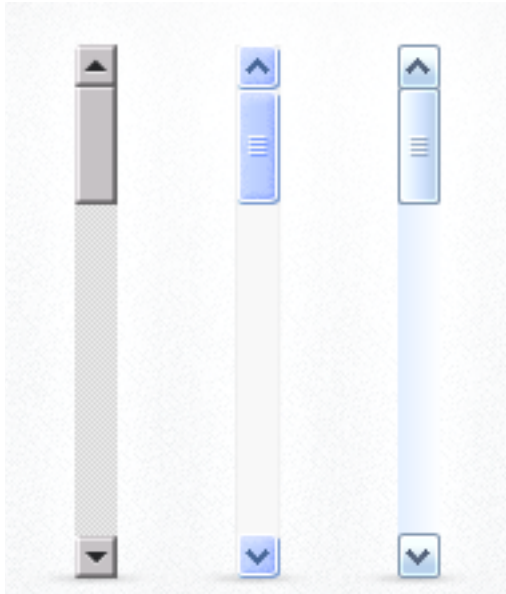


Things that have moved toward standardization

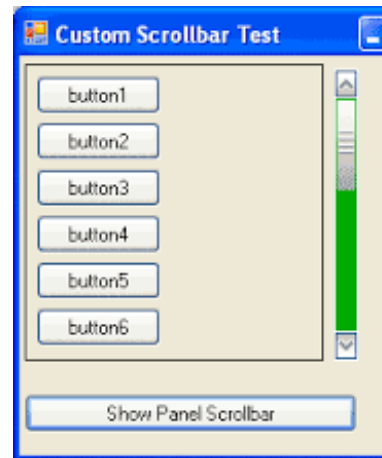


Widgets allow customization

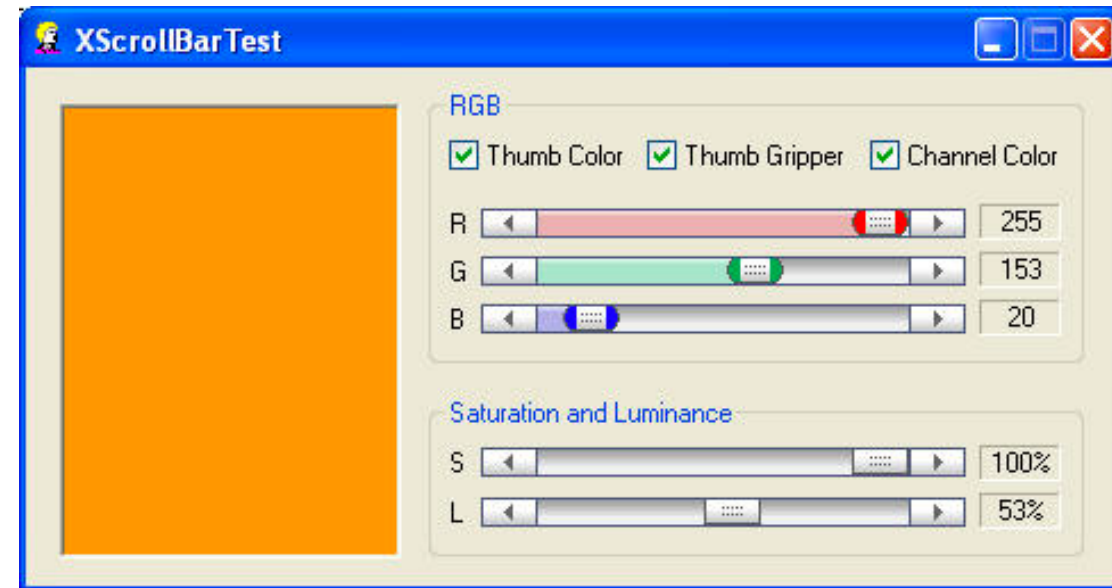
Customizable scroll bars



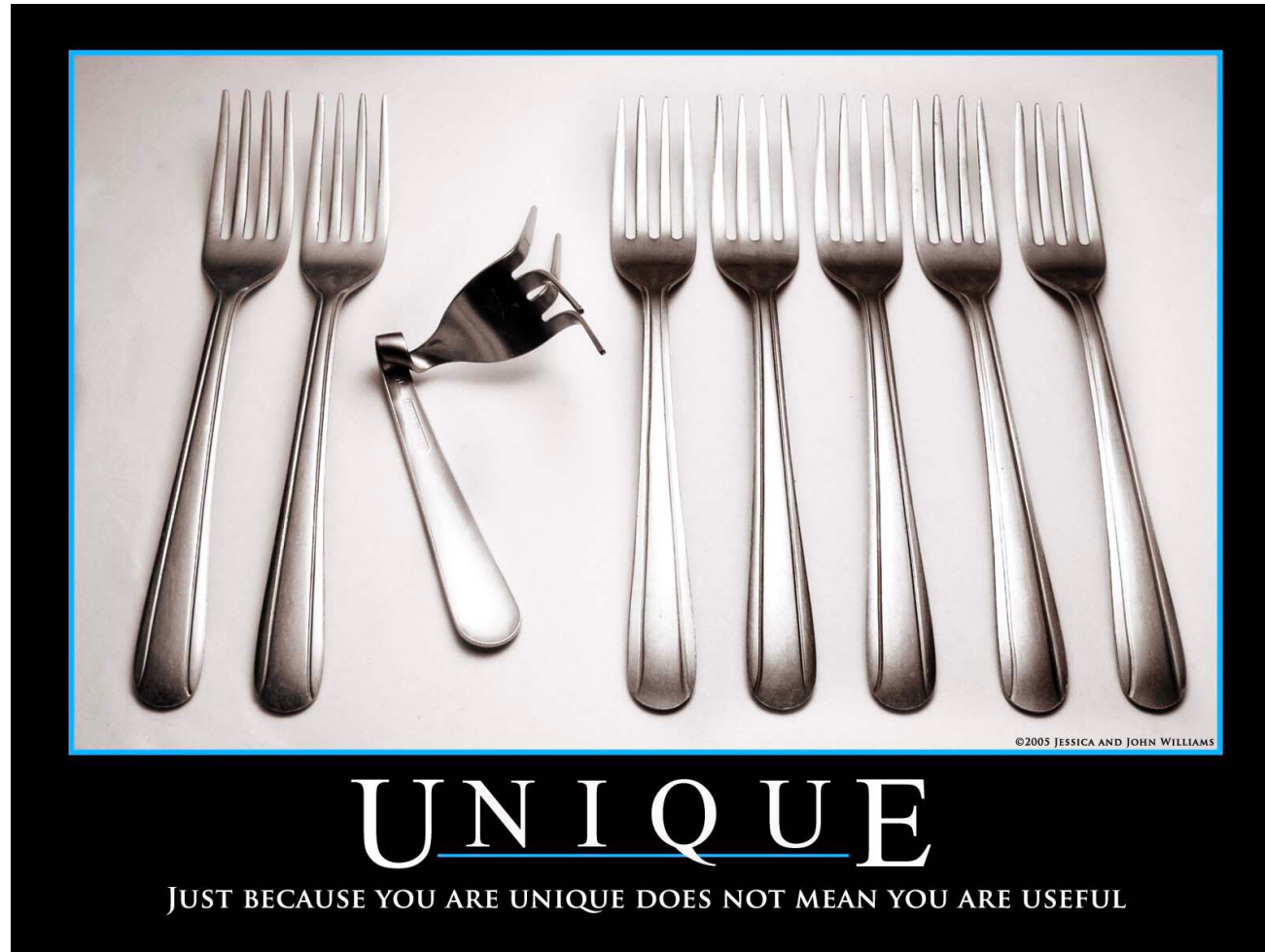
Bad use of customization.



Good use of customization.

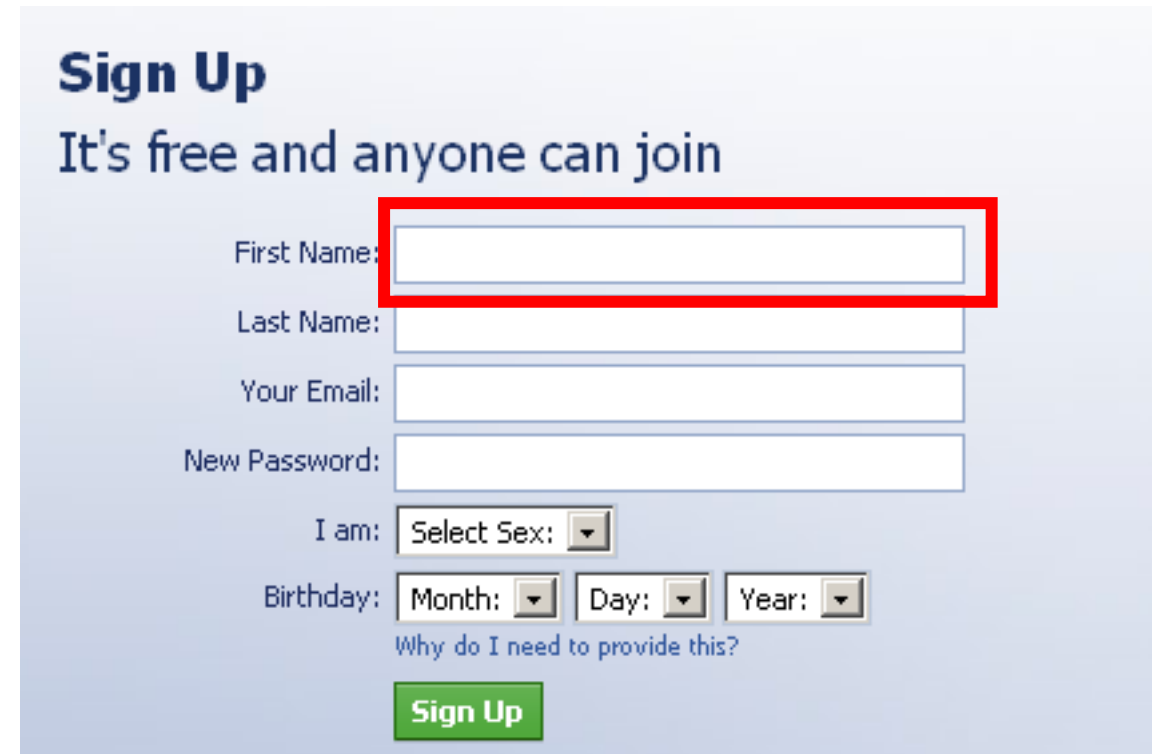
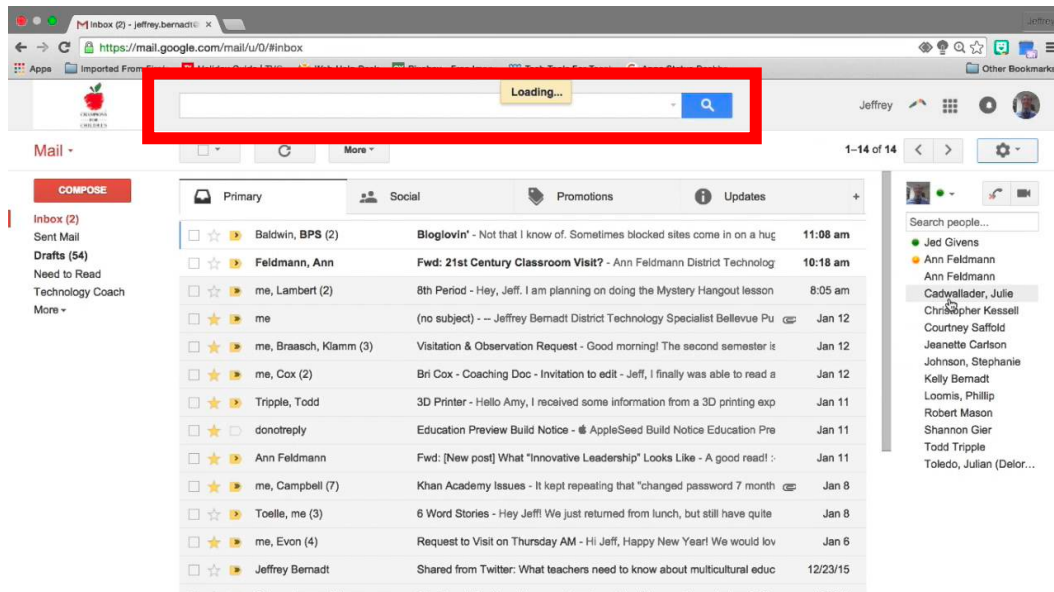


Use your powers of customization wisely.



Summary

There are some standard ways we interact with elements in an interface:



Widgets standardize common, low-level interaction interfaces

When you create a widget...

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

The **appearance** is standardized,



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

```
50
51   $("#counter").click(function(){
52     [REDACTED]
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   })
```

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

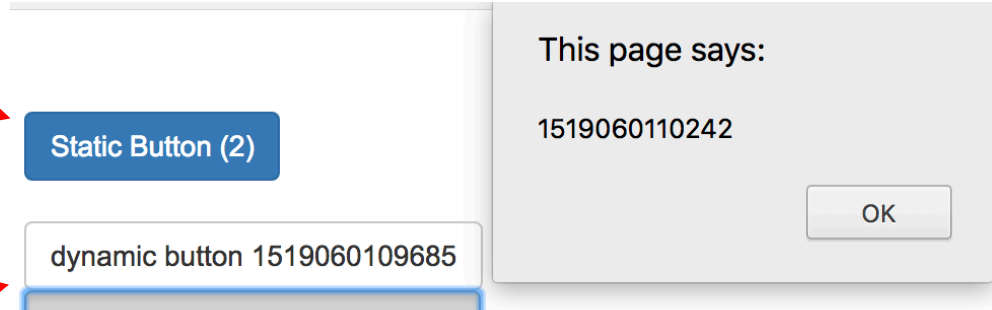
Static: HTML, JavaScript onReady

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

```
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 }
```

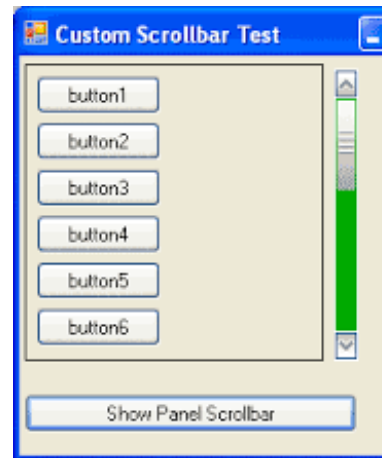


Widgets allow customization. Use it wisely.

Customizable scroll bars



Bad use of customization.



Good use of customization.

