



{ Javascript }

s.collins@salesforce.com
@theotherstevenc
#LitmusLive



Javascript History

a lot has finally changed



1995

Javascript created and released



1997

ECMAScript **ES 1** created and released



1998

ECMAScript **ES 2** released



1999

no update



2000

no update



2001

no update



2003

no update



2005

no update



2007

...still no update



2008

ES 4

Google, Adobe,
Mozilla, Opera

VS.

ES 3.1

Microsoft, Yahoo



2009

The **3.1** specifications were released as **ES 5**
ES 4 was completely abandoned



2010

no update



2011

no update



2013

no update



2014

still no update



2015

The **ES 4** specifications are finally released

ES Harmony

ES 6

ECMAScript 2015



2019

Current version is **ECMAScript 2019**



*the true power of javascript is generating
markup, whether from scratch, or from the
existing*



Document Object Model

*The DOM originated as a specification for
JavaScript*



objects

html tags

methods

actions applied on
objects and properties

properties

content, attributes

events

interactions with
objects and properties



object.method()



document.getElementById()

```
1<p id="bostonGetId">  
2  access an element based on its ID  
3</p>  
4
```

```
1  
2  
3 document.getElementById(`bostonGetId`)  
4  
5
```

access an element based on its ID



document.querySelector()

```
1 <p id="boston-query">  
2   access <span>element</span> based on a selector  
3 </p>  
4  
5  
6
```

```
1  
2 document.querySelector(`#boston-query span`)  
3  
4
```

access element based on a selector



document.querySelectorAll()

```
1 <p class="boston-all">  
2   returns a <b class="boston-all">list</b> of elements  
3 </p>  
4  
5
```

```
1  
2 console.log(document.querySelectorAll(`.boston-all`))  
3  
4 // NodeList(2) [p.boston-all, b.boston-all]  
5
```

returns a **list** of elements



element.addEventListener()

```
1
2 inputElement.addEventListener(`input`, function () {
3
4   console.log(`I'm your type`)
5
6 });
7
```

sets up a function to be called for a specific event



element.innerHTML

```
1  
2  
3 document.querySelector('p.bostonInner').innerHTML = `New Text`  
4  
5
```

change the HTML content



console.log()

```
1  
2 x = 11  
3  
4 console.log(x + 10)  
5  
6 // 21  
7  
8  
9 console.log(11 + 10)  
10  
11 // 21  
12
```

purely for learning, testing and debugging



{variables}

*“a location in memory
whose data can change
from time to time”*



{data types}

defined in the ECMAScript standards



let (the keyword)

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



string

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



number

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



true/false

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



empty

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



undefined

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



semicolons not required

```
1
2 let str = `abc 12sss3`;
3 //string
4
5 let num = 100
6 //integer
7
8 let x = true;
9 //boolean
10
11 let y = null;
12 //empty
13
14 let z
15 //undefined
16
```



single or a double?

```
1
2 let y = "value for y"
3 //double quotes
4
5 let z = 'value for z'
6 //single quotes
7
8 let x = `value for x`
9 //backtick
10
11 // backticks offer extended functionality
12
```



the other data types

- objects { }
- arrays []
- functions ()



objects {}

collection of data, structured like an unordered list



objects {}

```
1
2 let apple = {
3
4   type: `honeycrisp`,
5   organic: true
6
7 }
8
9 console.log(apple.type)
10 // honeycrisp
11
```



arrays []

collection of data, structured like an ordered list



arrays []

```
1
2 let apples = [
3   {
4     type: `honeycrisp`,
5     organic: true
6   },
7   {
8     type: `gala`,
9     organic: false
10  }
11 ]
12
13 console.log(apples[1].type)
14 // gala
15
```



functions ()

first-class objects



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



the for...of statement

```
1
2 let finders = [
3   {
4     find: `omw`,
5     replace: `on my way`
6   },
7   {
8     find: `lol`,
9     replace: `laughing out loud`
10  }
11 ]
12
13 for(let foo of finders) {
14   console.log(foo.find)
15 }
16 // omw
17 // lol
18
```



Hello Boston

```
1
2 <input type="text" id="myinput" placeholder="Litmus Live Boston..">
3
```

```
1
2 let inputElement = document.querySelector(`#myinput`)
3
4 inputElement.addEventListener(`input`, function () {
5
6   let userInput = inputElement.value
7   document.querySelector(`#myTitle`).innerHTML = userInput
8
9   console.log(userInput)
10
11 });
12
```

Litmus Live Boston 2019

#LitmusLive || @theotherstevenc



a new semantic markup

from the developer point of view

*"writing markup than means something to us,
and rethinking what an email template is"*



how it works

email client targeting example



targeting Gmail

```
1
2 /* Gmail */
3 u + .body .foo {
4   display:none;
5 }
6
7 /* Gmail mobile apps */
8 @media screen and (max-width: 480px) {
9   u + .body .foo {
10     display: block! important;
11   }
12 }
13
```

using standard CSS markup



targeting Gmail

```
1
2 /* Gmail */
3 -gmail .foo {
4   display:none;
5 }
6
7 /* Gmail mobile apps */
8 -mobilegmail {
9   -gmail .foo {
10     display: block! important;
11   }
12 }
13
```

using semantic CSS markup



targeting iOS

```
1
2 /* iOS 13 */
3 .apple-mail-implicit-dark-support .foo {
4   color:contrast;
5 }
6
7 /* iOS 10 */
8 @supports (-webkit-overflow-scrolling:touch) and (color:#fffff) {
9   .foo {
10     display:none;
11   }
12 }
13
```

using standard CSS markup



targeting iOS

```
1
2 /* iOS 13 */
3 -ios13 .foo {
4   color:contrast;
5 }
6
7 /* iOS 10 */
8 -ios10 {
9   .foo {
10     display:none;
11   }
12 }
13
```

using semantic CSS markup



targeting Outlook

```
1  /* outlook (macOS) */
2  _:-webkit-full-screen,
3  _:-webkit-full-page-media,
4
5  _:future, :root .body:not(.Singleton) .foo {
6    display:none;
7  }
8
```

using standard CSS markup



targeting Outlook

```
1
2 /* outlook (macOS) */
3 -outlookmac .foo {
4   display:none;
5 }
6
```

using semantic CSS markup



targeting Outlook

```
1 <div class="ghost-content-container">
2   <!--[if mso 15]>
3   <table role="presentation" width="100%;" bgcolor="green">
4     <tr>
5       <td>
6         <! [endif]-->
7         <div class="ghost-content">content in ghost table</div>
8       <! [endif]-->
9     </td>
10    </tr>
11  </table>
12  <! [endif]-->
13 </div>
14
15
```

using standard HTML markup



targeting Outlook

```
1 <div class="ghost-content-container">
2   <!--mso2013
3   <table role="presentation" width="100%;" bgcolor="green">
4     <tr>
5       <td>
6         mso-->
7         <div class="ghost-content">content in ghost table</div>
8       <!--mso2013
9       </td>
10      </tr>
11      </table>
12      mso-->
13    </div>
14  </div>
15
```

using semantic HTML markup



```
1
2 let targets = [
3   {
4     "find": /-gmail/gi,
5     "replace": `u + .body`
6   },
7   {
8     "find": /-mobilegmail/gi,
9     "replace": `@media screen and (max-width: 480px)`
10   }
11 ]
12
13 let css = document.head.innerHTML
14 for(let target of targets) {
15   css = css.replace(target.find, target.replace)
16 }
17 document.head.innerHTML = css
18
```



```
1
2 let targets = [
3   {
4     "find": /-gmail/gi,
5     "replace": `u + .body`
6   },
7   {
8     "find": /-mobilegmail/gi,
9     "replace": `@media screen and (max-width: 480px)`
10   }
11 ]
12
13 let css = document.head.innerHTML
14 for(let target of targets) {
15   css = css.replace(target.find, target.replace)
16 }
17 document.head.innerHTML = css
18
```



```
1
2 let targets = [
3   {
4     "find": /-gmail/gi,
5     "replace": `u + .body`
6   },
7   {
8     "find": /-mobilegmail/gi,
9     "replace": `@media screen and (max-width: 480px)`
10   }
11 ]
12
13 let css = document.head.innerHTML
14 for(let target of targets) {
15   css = css.replace(target.find, target.replace)
16 }
17 document.head.innerHTML = css
18
```



```
1
2 let targets = [
3   {
4     "find": /-gmail/gi,
5     "replace": `u + .body`
6   },
7   {
8     "find": /-mobilegmail/gi,
9     "replace": `@media screen and (max-width: 480px)`
10   }
11 ]
12
13 let css = document.head.innerHTML
14 for(let target of targets) {
15   css = css.replace(target.find, target.replace)
16 }
17 document.head.innerHTML = css
18
```



```
1
2 let targets = [
3   {
4     "find": /-gmail/gi,
5     "replace": `u + .body`
6   },
7   {
8     "find": /-mobilegmail/gi,
9     "replace": `@media screen and (max-width: 480px)`
10   }
11 ]
12
13 let css = document.head.innerHTML
14 for(let target of targets) {
15   css = css.replace(target.find, target.replace)
16 }
17 document.head.innerHTML = css
18
```



the super list

```
1 // #####  
2 // #### email media css #####  
3 // #####  
4 // #####  
5 let medias = [  
6   {  
7     "find": /-webkit/gi,  
8     "replace": `@media screen and (-webkit-device-pixel-ratio)`  
9   },  
10  {  
11    "find": /-max600/gi,  
12    "replace": `@media screen and (max-width: 600px)`  
13  },  
14  {  
15    "find": /-dark/gi,  
16    "replace": `@media (prefers-color-scheme: dark)`  
17  },  
18  {  
19    "find": /-iphonex/gi,
```



Chrome Extensions

test before you test



what will this extension do?

render the gmail styles

disable animations

unchecked any input



side note: any css not supported in gmail
needs to be in a separate style block

```
1
2 <style advanced="">
3   .styles .not .supported .in .gmail {
4     fancy: things;
5   }
6 </style>
7
```



manifest.json

configuration, properties, information about the
chrome extension



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions":["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions":["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



manifest.json

```
1  {
2      "manifest_version":2,
3      "name":"Gmail Previewer",
4      "version":"1.0",
5      "content_scripts": [
6          {
7              "matches": [
8                  "<all_urls>"
9              ],
10             "js": ["content.js"]
11         }
12     ],
13     "permissions": ["tabs"],
14     "browser_action": {
15         "default_icon": "icon.png",
16         "default_popup": "popup.html",
17         "default_title": "Basic Gmail Preview"
18     }
19 }
```



default_popup

.html and a .js friend



popup.html

```
1 <html>
2   <head>
3     <link rel="stylesheet" href="style.css">
4   </head>
5   <body>
6
7     <div class="container">
8       <button id="gmail">Email Client Preview</button>
9     </div>
10
11    <script src="popup.js" type="text/javascript"></script>
12  </body>
13 </html>
14
15
```



popup.html

```
1 <html>
2   <head>
3     <link rel="stylesheet" href="style.css">
4   </head>
5   <body>
6
7     <div class="container">
8       <button id="gmail">Email Client Preview</button>
9     </div>
10
11    <script src="popup.js" type="text/javascript"></script>
12  </body>
13 </html>
14
15
```



popup.html

```
1 <html>
2   <head>
3     <link rel="stylesheet" href="style.css">
4   </head>
5   <body>
6
7     <div class="container">
8       <button id="gmail">Email Client Preview</button>
9     </div>
10
11    <script src="popup.js" type="text/javascript"></script>
12
13 </body>
14 </html>
15
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10});  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10});  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10 } );  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10 } );  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10 } );  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



popup.js

```
1 var params = {  
2   active: true,  
3   currentWindow: true  
4 }  
5  
6 let button = document.getElementById(`gmail`)  
7  
8 button.addEventListener(`click`, function() {  
9   paintPreview(`gmail`);  
10 } );  
11  
12 function paintPreview(preview) {  
13   chrome.tabs.query(params, gotTab);  
14   function gotTab(tabs) {  
15     chrome.tabs.sendMessage(tabs[0].id, preview)  
16   }  
17 }
```



content.js



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



content.js

```
1 chrome.runtime.onMessage.addListener(gotMessage);  
2  
3 function gotMessage(message, sender, sendResponse) {  
4  
5     let finders = [  
6         {  
7             "find": /u+/gi,  
8             "replace": ``  
9         },  
10        {  
11            "find": /u+/gi,  
12            "replace": ``  
13        }  
14    ];  
15  
16    let markup = document.querySelector(`html[lang]`).innerHTML;  
17    for(let finder of finders){  
18        markup = markup.replace(finder.find, finder.replace);  
19    }  
20}
```



Chrome Extension

github.com/theotherstevenc/test-before-you-test

Markup with Meaning

github.com/theotherstevenc/markup-with-meaning



thank you()

s.collins@salesforce.com
@theotherstevenc
#LitmusLive