



# back to the future with web components



IAN BUCHANAN • DEVELOPER PARTISAN • ATLASSIAN • @DEVPARTISAN

# **ANNUAL MEETING OF TIME TRAVELLERS**

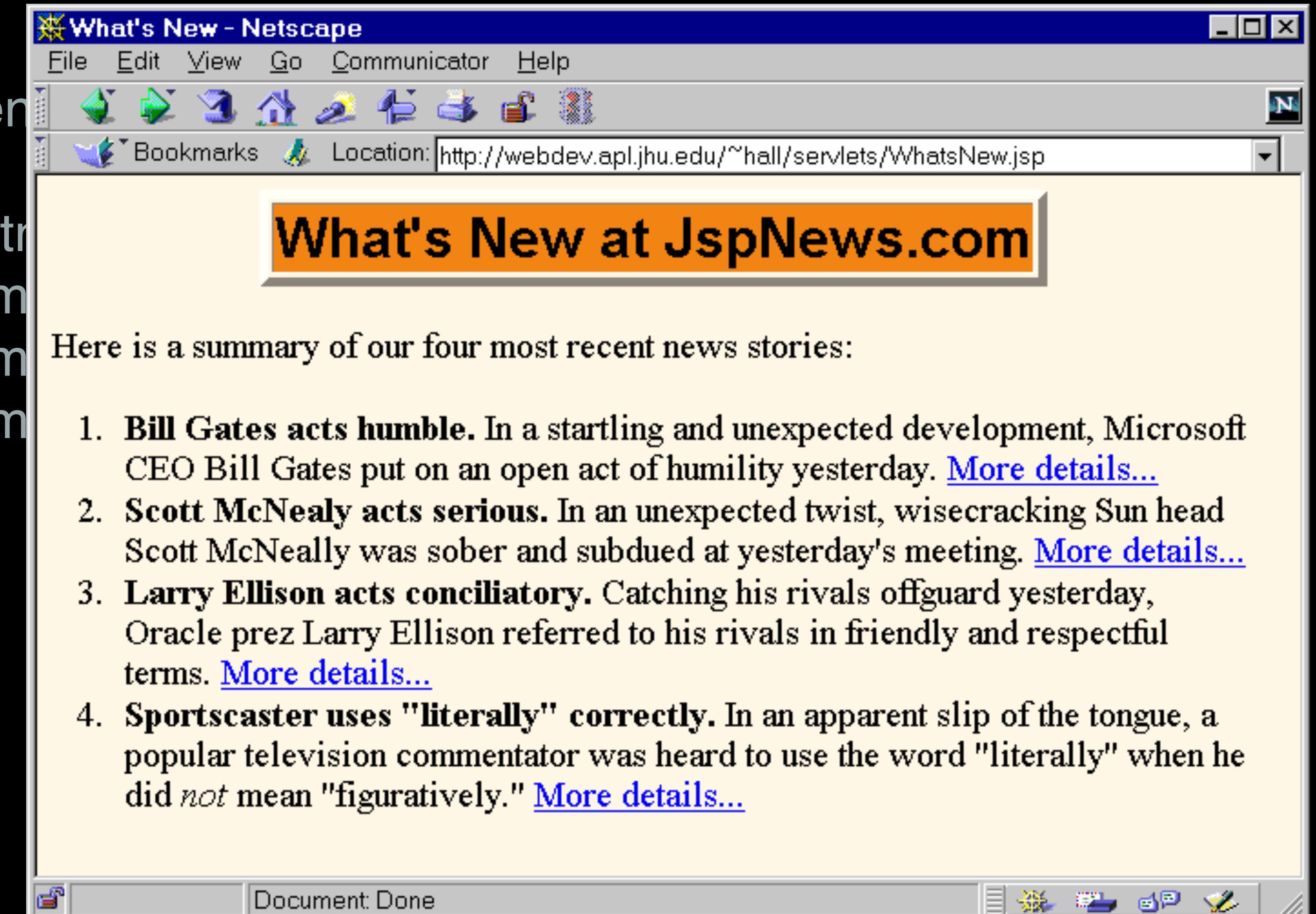


# JSP 1.0 circa 1999

...  
Here is a summary of our four most recent news stories:

```
<OL>
<LI><jsp:include page="news/item1.htm">
<LI><jsp:include page="news/item2.htm">
<LI><jsp:include page="news/item3.htm">
<LI><jsp:include page="news/item4.htm">
</OL>
```

...



# JSP 2.0 circa 2003

```
<%@ taglib prefix="news" tagdir="/WE
```

...

Here is a summary of our four most recent news stories:

```
<OL>
```

```
<LI>
```

```
<news:short
```

```
    title="Bill Gates acts humble"
```

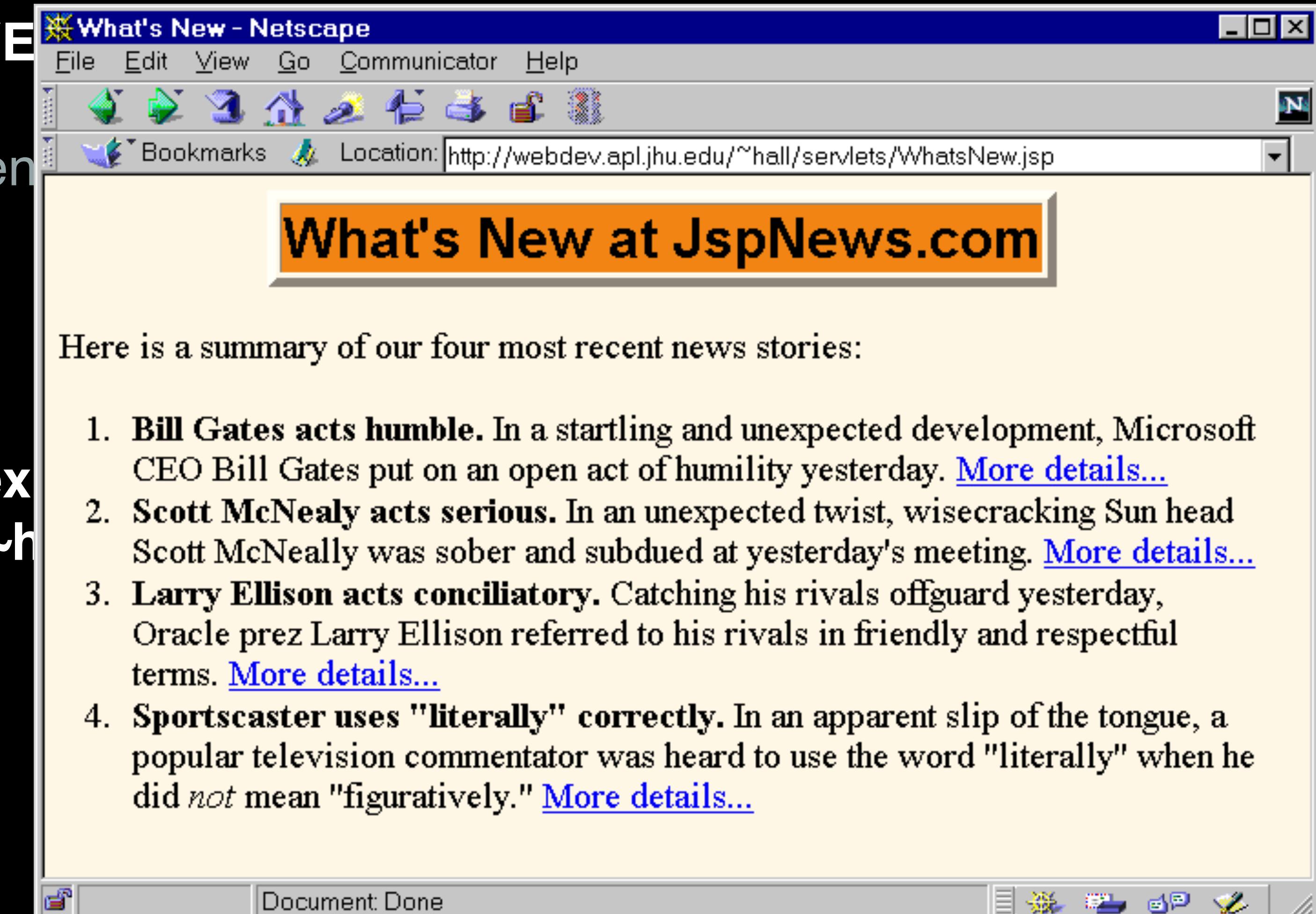
```
    summary="In a startling and unexpected development,
```

```
    href="http://webdav.apl.jhu.edu/~hall/servlets/WhatsNew.jsp
```

```
    />
```

```
</LI>
```

...



# Web Components circa 2012



## Custom Elements

Define and use new types of DOM elements.



## Imports

Include and reuse HTML documents in other HTML documents.



## Templates

Declare inert DOM subtrees in HTML and manipulate them to instantiate document fragments with identical contents.



## Shadow DOM

Enable better functional encapsulation within the DOM.

<http://webcomponents.org/>



# Agenda

EXAMPLE

WHY

HOW

LESSONS LEARNED



# Web Component Example

EXAMPLE

WHY

HOW

LESSONS LEARNED

<https://bitbucket.org/jcreenaune/meme-or-die>

<https://bitbucket.org/jcreenaune/meme-or-die-demo>



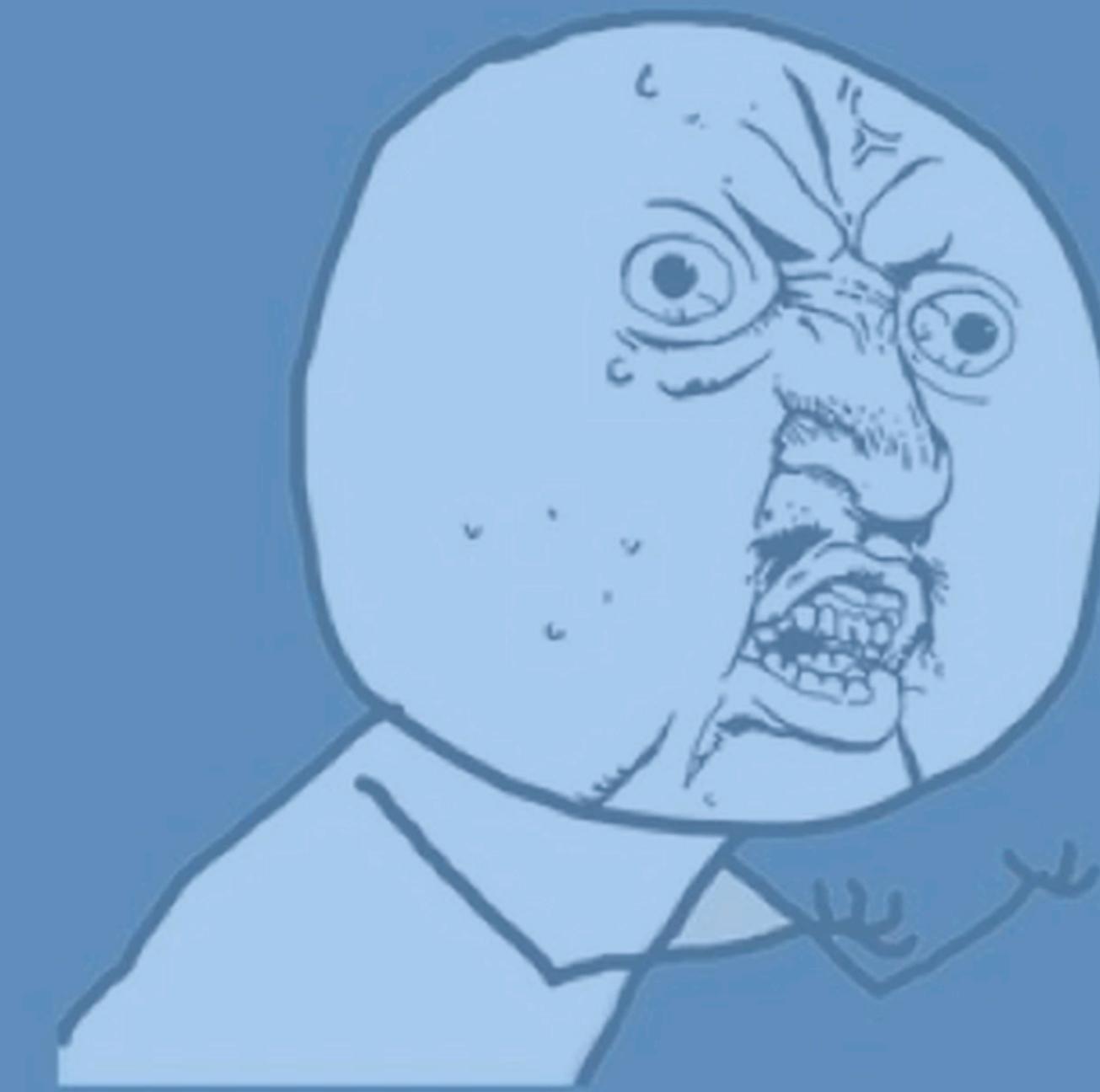
The screenshot shows the 'Elements' tab of a browser's developer tools. The DOM tree is displayed, starting with the root element <!DOCTYPE html>. Below it is the <html> element, which has a <head> child. The <head> element contains several other elements, some of which are partially visible or collapsed. The <body> element is also present under the <html> node. At the bottom of the tree, there is a </html> element. The <body> element is currently selected, indicated by a pink highlight around its opening tag. The status bar at the bottom shows the selected nodes: 'html' and 'body'. Below the status bar, there are tabs for 'Styles', 'Event Listeners', 'DOM Breakpoints', and a 'More' button represented by three dots.

```
<!DOCTYPE html>
▼ <html>
▶ <head>...</head>
<body>
</html>
```

html body

Styles Event Listeners DOM Breakpoints »

meme-image 643px x 484px

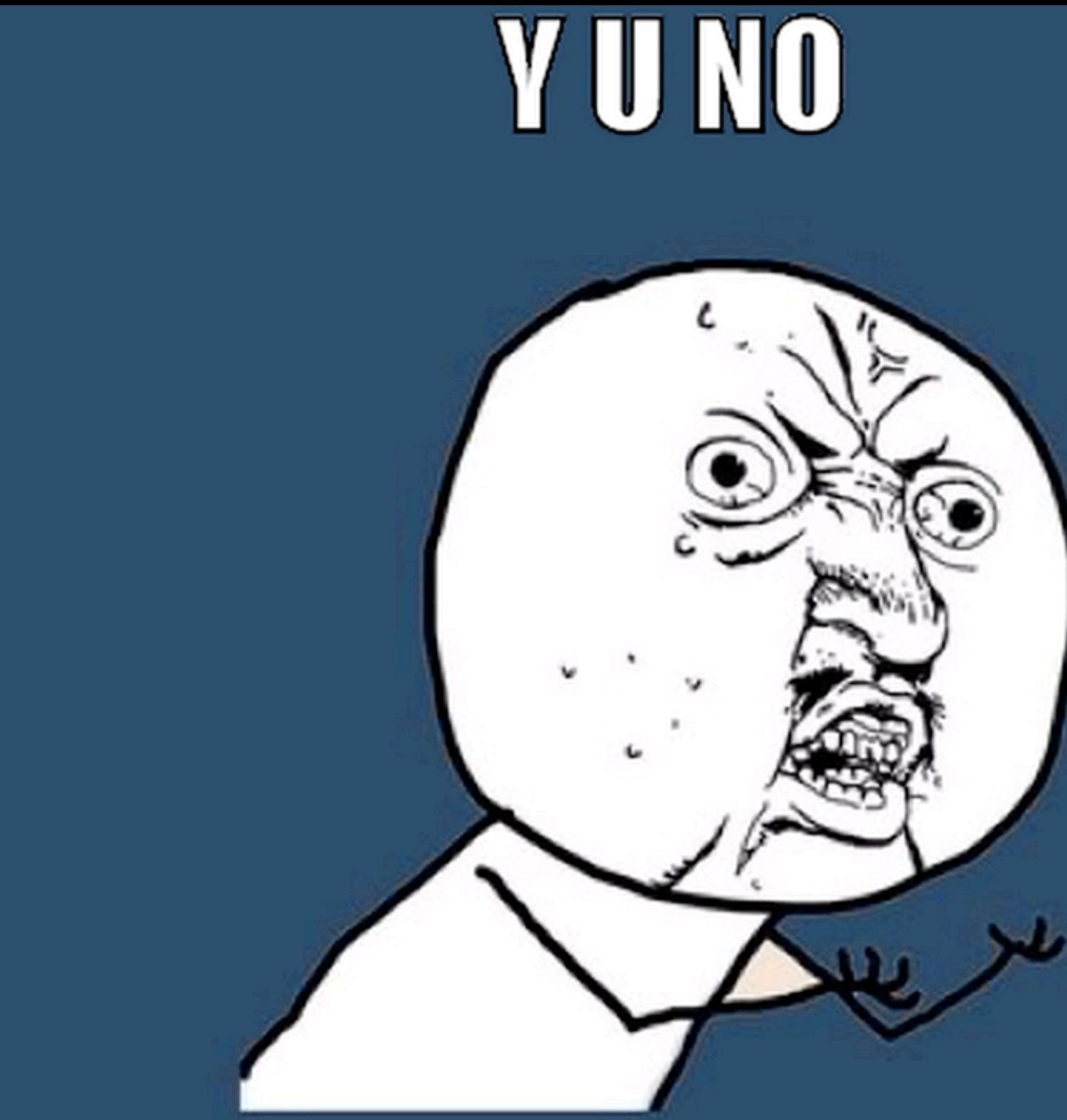


Elements Network > \_ ⚙️ 📁 ×

```
<!DOCTYPE html>
▼ <html>
  ► <head>...</head>
  ▼ <body>
    ► <meme-image src="uno" resolved>...</meme-
      image>
    </body>
  </html>
```

html body meme-image

Styles Event Listeners DOM Breakpoints »



put structure in markup

Elements Network > \_ ⚙️ 📁 ×

<body style>

<meme-image src="yuno" resolved>

<meme-text text="Y U NO" resolved></meme-text>

html body meme-image

Styles Event Listeners DOM Breakpoints >

Console Search Emulation Rendering

✖️ ⚡ <top frame> ▶

meme-or-die.js:122

The screenshot shows a browser's developer tools open over a meme image. The meme features a white circular portrait of a man with a wide-eyed, screaming expression, set against a dark blue background. Above the portrait, the word "YUNO" is displayed in large, bold, white capital letters. The developer tools are focused on the "Elements" tab, which shows the HTML structure of the meme. The "meme-image" element is selected, revealing its source as "yuno" and its resolved state. Inside the "meme-image" element, there is a "meme-text" element with the text "Y U NO" and its resolved state. Below the elements tab, there are tabs for "Styles", "Event Listeners", and "DOM Breakpoints". Further down, there are tabs for "Console", "Search", "Emulation", and "Rendering". At the bottom of the tools, there are icons for "stop", "refresh", and "presenter mode", along with a file path "meme-or-die.js:122" and a blue navigation arrow.

```
var d = document;  
d.addEventListener('meme-generate-start', startSpinning);  
d.addEventListener('meme-generate-end', endSpinning);  
d.addEventListener('meme-generate-error', endSpinning);
```



meme-image 643px x 484px

Y U NO

put structure in markup

Elements Network > \_ X

<body style>

<meme-image src="doge" resolved>

<meme-text text="Y U NO" resolved></meme-text>

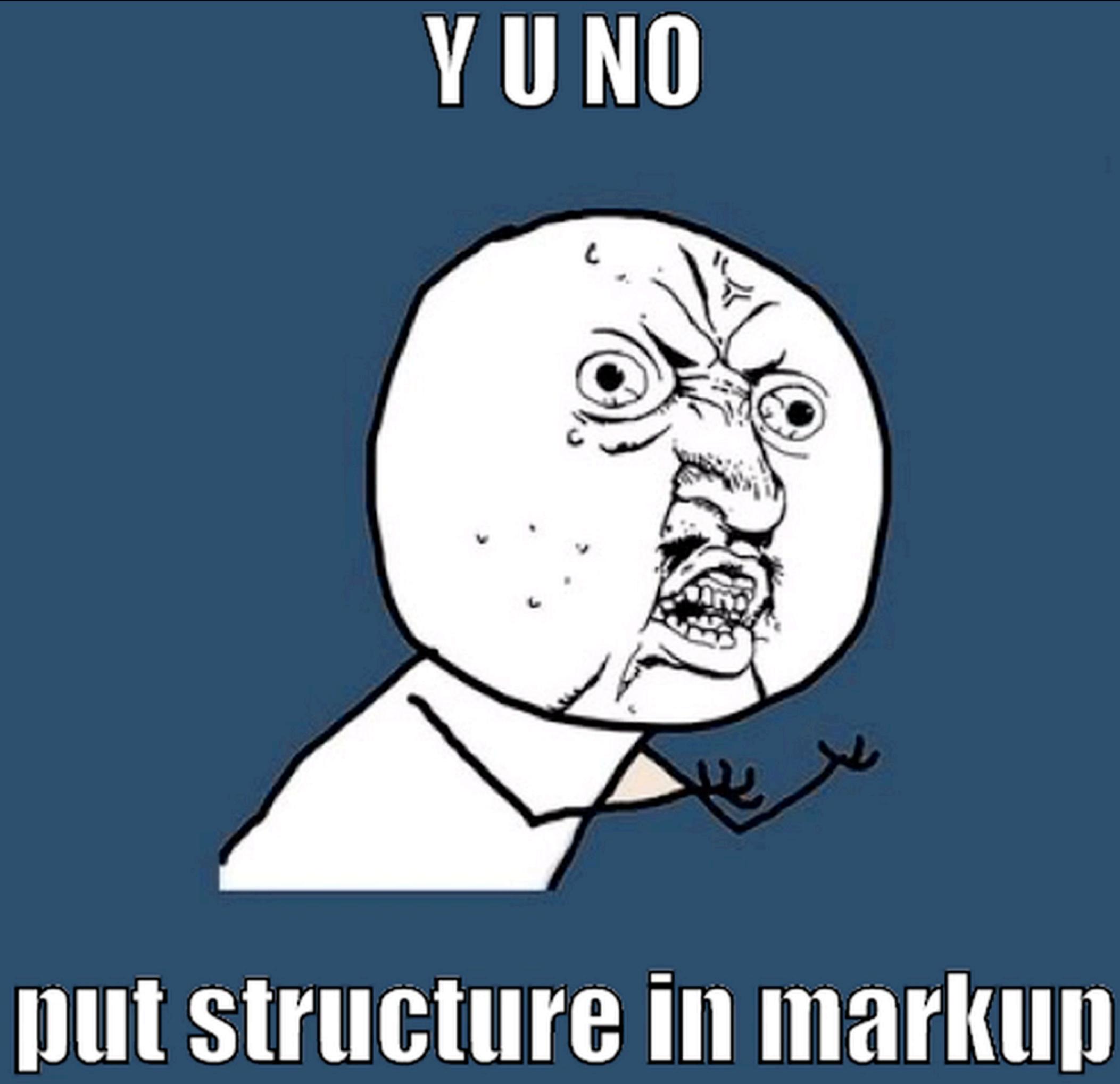
html body meme-image

Styles Event Listeners DOM Breakpoints >

Console Search Emulation Rendering

Prese

meme-or-die.js:122



Elements Network >

<body style>

<meme-image src="yun0" resolved>

<meme-text text="Y U NO" resolved></meme-text>

html body meme-image

Styles Event Listeners DOM Breakpoints >

Console Search Emulation Rendering

<top frame>

meme-or-die.js:122

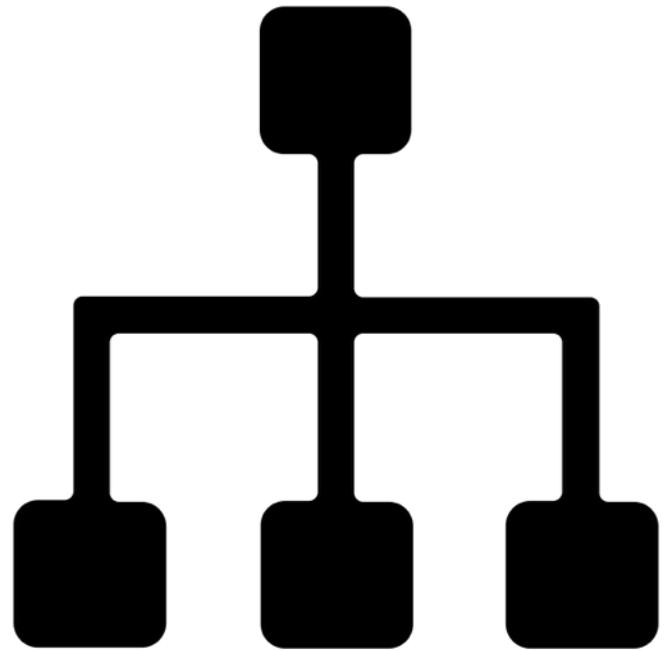
>

A screenshot of a browser's developer tools, specifically the Elements tab. The DOM tree shows the structure of the meme image. The "meme-image" element is selected, highlighted with a blue border. The "text" attribute of the "meme-text" element contains the text "Y U NO". The status bar at the bottom right shows the file path "meme-or-die.js:122".

# Web Components

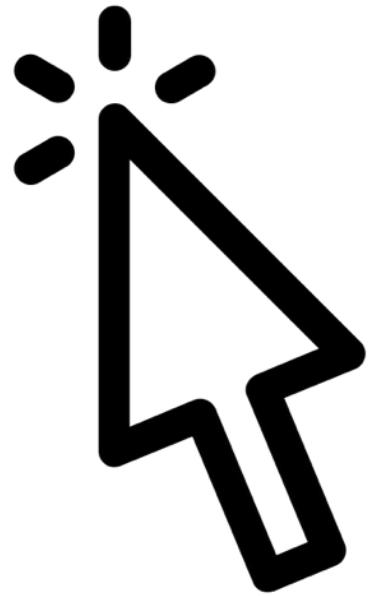


**structure**



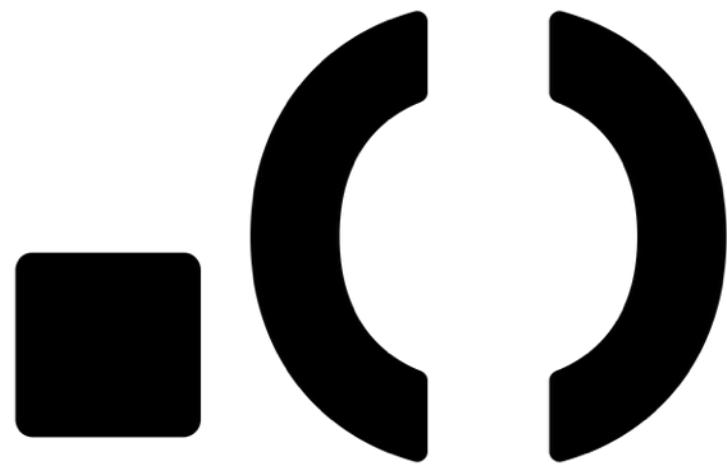
**DOM**

**events**



**DOM events**

**behavior**



**methods on  
DOM elements**



# Why use Web Components

EXAMPLE

WHY

HOW

LESSONS LEARNED





owie!

# AUI

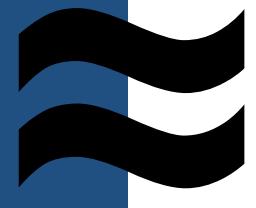
Atlassian User Interface  
is a tailor-made frontend library for  
creating a user interface according to  
the Atlassian Design Guidelines.

<https://docs.atlassian.com/auil/latest/>

<https://design.atlassian.com/>



**Bootstrap**



**AUI**





## Backup stale

This instance was last backed up on Thursday, 18 September 2011.





## Backup stale

This instance was last backed up on Thursday, 18 September 2011.

```
<div class="aui-message aui-message-info">
  <p class="title">
    <strong>Backup stale</strong>
  </p>
  <p>This instance was last backed up on Thursday, 18 September 2011.</p>
</div>
```

i

## Backup stale

This instance was last backed up on Thursday, 18 September 2011.

```
<div class="aui-message aui-message-info">
  <p class="title">
    <strong>Backup stale</strong>
  </p>
  <p>This instance was last backed up on Thursday, 18 September 2011.</p>
</div>
```



# Angry nerds





# Angry nerds

```
<header class="aui-page-header">
  <div class="aui-page-header-inner">
    <div class="aui-page-header-image">
      <span class="aui-avatar aui-avatar-large aui-avatar-project">
        <span class="aui-avatar-inner">
          
        </span>
      </span>
    </div>
    <div class="aui-page-header-main">
      <h1>Angry nerds</h1>
    </div>
  </div>
</header>
```



# Angry nerds

```
<header class="aui-page-header">
  <div class="aui-page-header-inner">
    <div class="aui-page-header-image">
      <span class="aui-avatar aui-avatar-large aui-avatar-project">
        <span class="aui-avatar-inner">
          
        </span>
      </span>
    </div>
    <div class="aui-page-header-main">
      <h1>Angry nerds</h1>
    </div>
  </div>
</header>
```

Tab 1

Tab 2

---

## This is Tab 1

First tab content

```
<div class="aui-tabs horizontal-tabs">
  <ul class="tabs-menu">
    <li class="menu-item active-tab">
      <a href="#tabs-example-first">
        <strong>Tab 1</strong>
      </a>
    </li>
    <li class="menu-item">
      <a href="#tabs-example-second">
        <strong>Tab 2</strong>
      </a>
    </li>
  </ul>
  <div class="tabs-pane active-pane" id="tabs-example-first">
    <h2>This is Tab 1</h2>
    <p>First tab content</p>
  </div>
  <div class="tabs-pane" id="tabs-example-second">
    <h2>This is Tab 2</h2>
    <p>Second tab content.</p>
  </div>
```

Menu ▾

Item 1

Item 2

```
<button class="aui-button aui-dropdown2-trigger" aria-owns="my-menu"  
aria-haspopup="true" aria-expanded="false">  
    Menu  
</button>
```

```
<div id="my-menu" class="aui-dropdown2 aui-style-default" role="menu"  
aria-expanded="false" aria-hidden="true">  
    <ul>  
        <li><a href="#">Item 1</a></li>  
        <li><a href="#">Item 2</a></li>  
    </ul>  
</div>
```

```
<button class="aui-button aui-dropdown2-trigger" aria-owns="my-menu"
aria-haspopup="true" aria-expanded="false">
    Menu
</button>

<div id="my-menu" class="aui-dropdown2 aui-style-default" role="menu"
aria-expanded="false" aria-hidden="true">
    <ul>
        <li><a href="#">Item 1</a></li>
        <li><a href="#">Item 2</a></li>
    </ul>
</div>
```

```
<div class="aui-message aui-message-info">
  <p class="title">
    <strong>Backup stale</strong>
  </p>
  <p>This instance was last backed up on Thursday, 18 September 2011.</p>
</div>
```



Original photo by Ian Buchanan

“Once the product’s task is known, design the interface first; then implement to the interface design.”

JEF RASKIN



## Backup stale

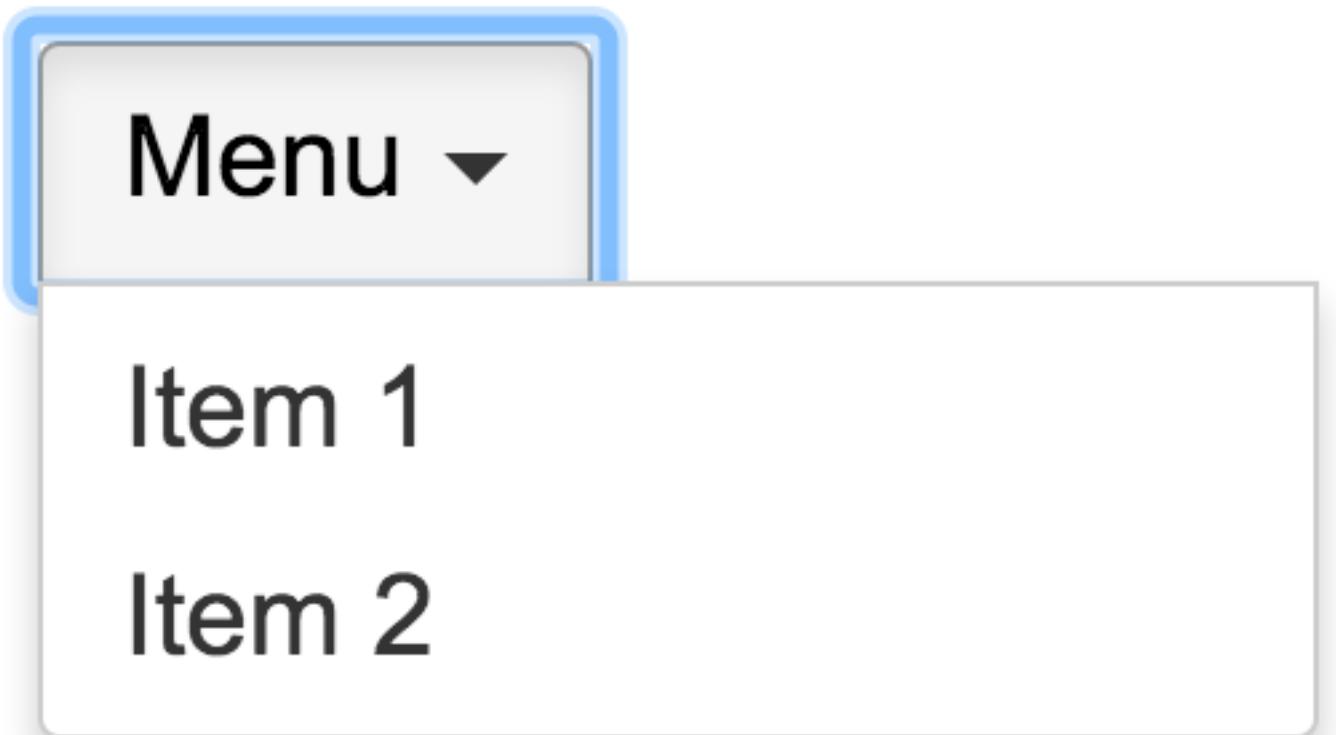
This instance was last backed up on Thursday, 18 September 2011.

```
<aui-message type="info" title="Backup stale">
  <p>This instance was last backed up on Thursday, 18 September 2011.</p>
</aui-message>
```



# Angry nerds

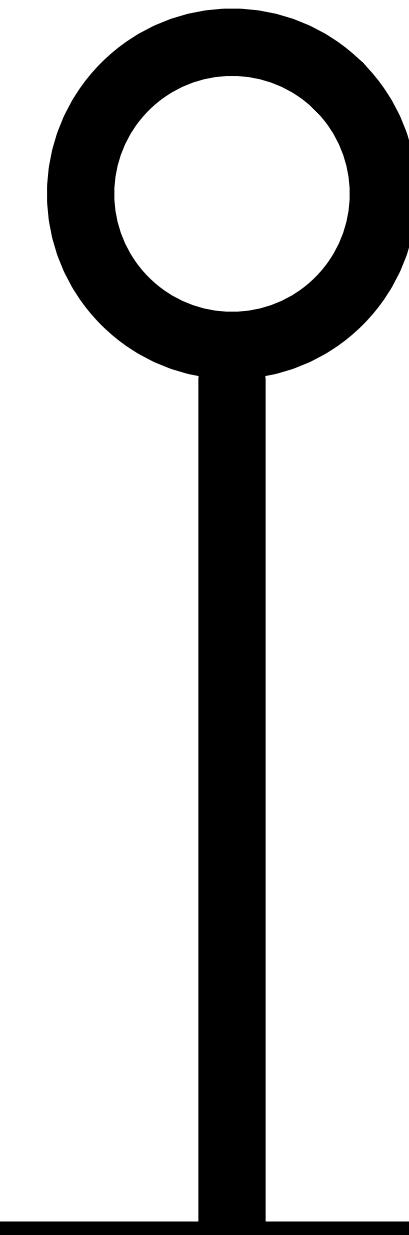
```
<aui-page-header title="Angry nerds">
  <aui-avatar src="images/angry-nerds-64.png" type="project" />
</aui-page-header>
```



```
<aui-trigger for="my-menu">Menu</aui-trigger>
```

```
<aui-menu id="my-menu">
  <aui-menu-item href="#">Item 1</aui-menu-item>
  <aui-menu-item href="#">Item 2</aui-menu-item>
</aui-menu>
```

# Interface



Custom  
Elements



Implementation



# organizing

**view**

**viewmodel**

**model**

# organizing

view

viewmodel  
=  
custom  
elements

model

Backbone.js  
AngularJS  
Ember.js  
KnockoutJS  
Dojo  
YUI  
Knockback.js  
CanJS  
React  
Mithril  
Ampersand  
Flight  
Vue.js

MarionetteJS  
TroopJS  
Spine  
Dart  
GWT  
Closure  
Elm  
Serenade.js  
Reagent  
Thorax  
Chaplin  
CanJS  
Thorax

Lumbar  
soma.js  
Durandal  
Lavaca  
cujoJS  
Montage  
Sammy.js  
Epitome  
soma.js  
DUEL  
Kendo UI  
PureMVC  
Olives

Dijon  
rAppid.js  
DeftJS + ExtJS  
Aria Templates  
SAPUI5  
Exoskeleton  
Atma.js  
Ractive.js  
ComponentJS  
Aurelia  
FOAM  
WebRx  
Angular 2.0





[https://www.flickr.com/photos/usarmyeurope\\_images/12770438685](https://www.flickr.com/photos/usarmyeurope_images/12770438685)

# How to use Web Components

EXAMPLE

WHY

HOW

LESSONS LEARNED





## Custom Elements

Define and use new types of DOM elements.



## Imports

Include and reuse HTML documents in other HTML documents.



## Templates

Declare inert DOM subtrees in HTML and manipulate them to instantiate document fragments with identical contents.



## Shadow DOM

Enable better functional encapsulation within the DOM.





## Custom Elements

Define and use new types of DOM elements.



## Imports

Include and reuse HTML documents in other HTML documents.



## Templates

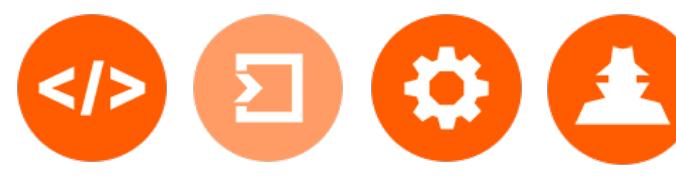
Declare inert DOM subtrees in HTML and manipulate them to instantiate document fragments with identical contents.



## Shadow DOM

Enable better functional encapsulation within the DOM.

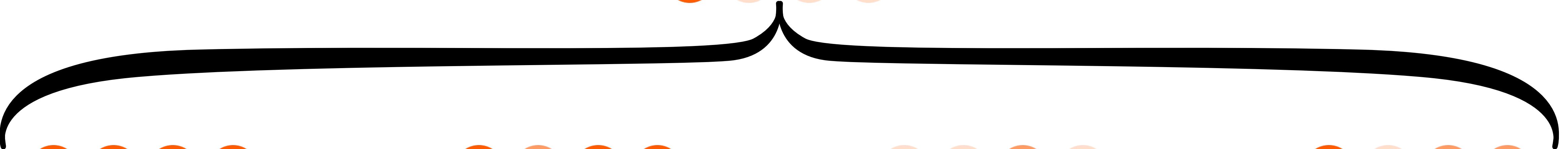




<http://jonrimmer.github.io/are-we-componentized-yet/>



# skate.js



<http://skate.js.org/>

<https://github.com/skatejs/skatejs>



# skate

minimal implementation

custom elements

noConflict(),  
multiple versions on a page

5KB, min+gz

# vs

# polymer

reference implementation

custom elements, imports,  
templates, shadow DOM

single version on page

44KB, min+gz



```
skate('hello-world', {  
  created: function (element) {  
    element.textContent = 'Hello, World!';  
  }  
});
```

```
skate('hello-world', {  
  created: function (element) {  
    element.textContent = 'Hello, World!';  
  }  
});
```

```
skate('hello-world', {  
  created: function (element) {  
    element.textContent = 'Hello, World!';  
  }  
});
```

```
<script href="hello-world.js"></script>
```

```
skate('hello-world', {  
  created: function (element) {  
    element.textContent = 'Hello, World!';  
  }  
});
```

<hello-world>      </hello-world>

```
skate('hello-world', {  
  created: function (element) {  
    element.textContent = 'Hello, World!';  
  }  
});
```

<hello-world> Hello, World! </hello-world>

# Lessons Learned

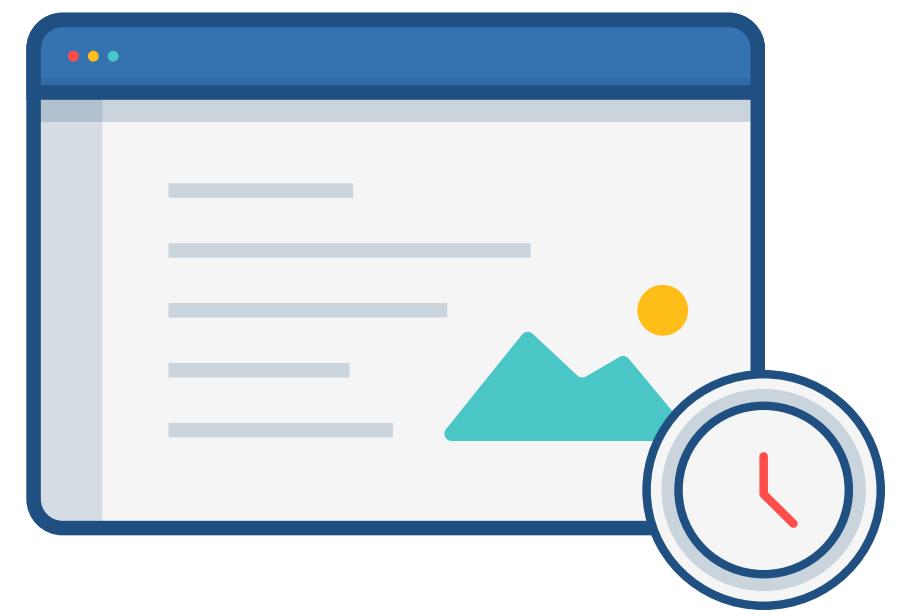
EXAMPLE

WHY

HOW

LESSONS LEARNED

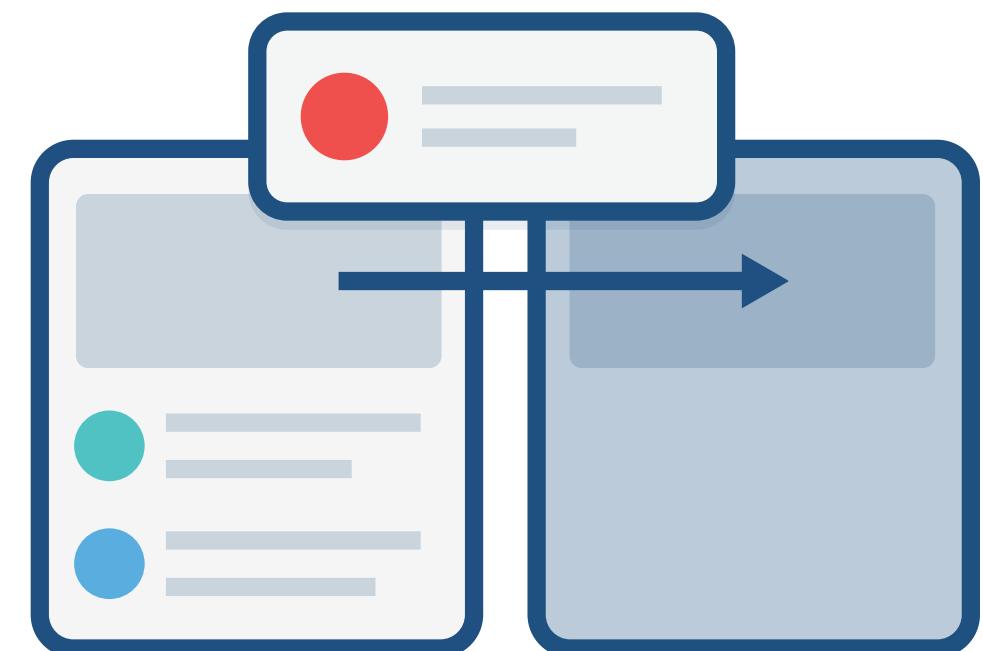




**client-side  
performance**

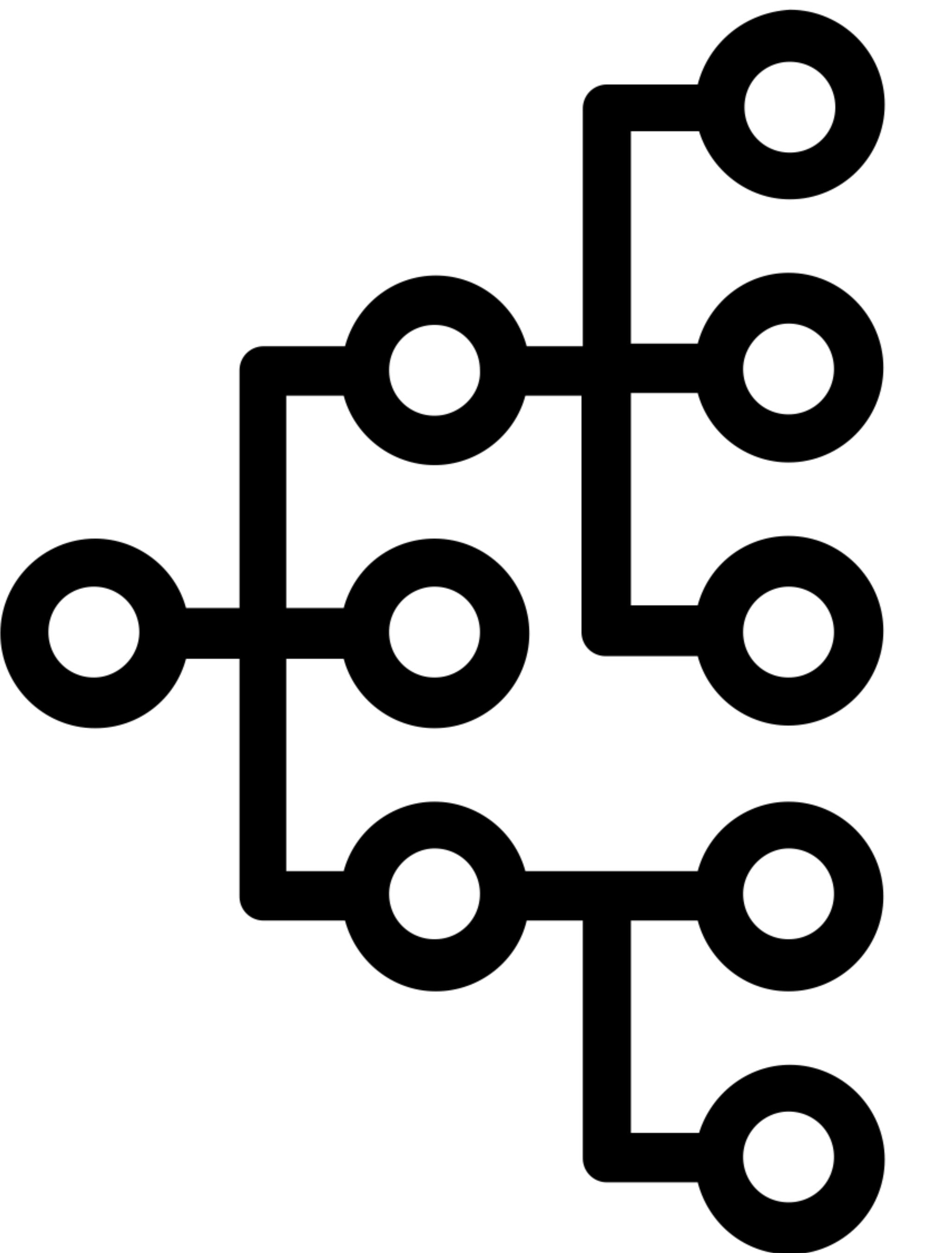


**server-side  
rendering**



**evolving  
components**





**Pull requests**

#171 OPEN [blog/the-err-the-oh-and-the-argh/index.md](#) → [develop](#)

**Merge** **Edit** **Decline** **Approve** 2

**First cut of the new blog post**

**Overview** **Commits** **Activity**

**Author** **Steve Haffenden**

**Reviewers**

**Description** Apologies for the delay in getting this into pull request.

**Comments (4)**

**sarahgoffdupont** Steve, I went ahead and made some edits – revert or change further as you see fit. Two biggest changes are: 1. Moved part of your intro to the bottom to form a conclusion paragraph, and bring your thesis statement ("I wanted to break down the different stages of learning...") closer to the top. 2. Did a light SEO treatment, optimizing for the term "learning techniques". I recommend updating the directory name to include that phrase so it's part of your URL – possibly to just learning-techniques-for-software-developers – which will further boost SEO.  
Nice piece with great reminders for any developer!  
Reply • Create task • 2015-08-26

**Steve Haffenden** AUTHOR Thanks Sarah. That all sounds fine to me, I appreciate the time you took to update :)  
Reply • Create task • 2015-08-31

**Travis Smith** <https://developer.stg.internal.atlassian.com/blog/2015/09/the-err-the-oh-and-the-argh/>  
Reply • Create task • 2015-08-31

**Ralph Whitbeck** I like the post. Unfortunately, the first impression (without reading a word) is that it's a wall of text ... this article might benefit from some simple stock / royalty free photos to break up the major learning techniques. Just a suggestion though. It might break up the sections and make it seem more manageable to read.  
Reply • Create task • 2015-09-14

**What do you want to say?**

**Files changed (1)**

+116 -0 app/posts/2015/09/the-err-the-oh-and-the-argh/index.md 12

**app/posts/2015/09/the-err-the-oh-and-the-argh/index.md** ADDED

```
1 +---  
2 +title: "The err, the oh, and the argh: learning techniques for software developers"  
3 +date: "2015-09-10 10:00"  
  
Travis Smith Update the date & https://extranet.atlassian.com/display/DEVNET/DAC+blog+content+calendar, then merge away.  
Reply • Create task • 2015-09-14
```

```
4 +author: "shaffenden"  
5 +categories: ["development"]  
6 +lede: "Like most people I enjoy things a lot more when I'm good at them but in  
7 +order to get good I'm forced to navigate myself through many of the less pleasing  
8 +states of learning and if I'm honest I find that pretty hard"  
9 +---  
10 +  
11 +I've been taking a little bit of my time to try and improve my Java skills – something which I may  
12 +have mentioned before – it's an experience which I've found equal parts confusing, exhilarating, and
```

**Travis Smith** Do you want to link to you mentioning it before?  
Reply • Create task • 2015-09-14

```
13 +frustrating and unfortunately I'm nowhere near done. I say unfortunately because, like most people,  
14 +I enjoy things a lot more when I'm good at them. But in order to get good, I'm forced to navigate  
15 +myself through many of the less pleasing states of learning. And if I'm honest, I find that pretty  
16 +hard.  
17 +
```

# actual site

Docs Contributing License

Search 



# SkateJS 0.13.5

Skate is a web component library based on the custom element spec. It's focused on being a tiny, performant, syntactic-sugar for binding behaviour to elements, attributes and classes using a consistent, simple and declarative API.

 Start Skating ★ 473

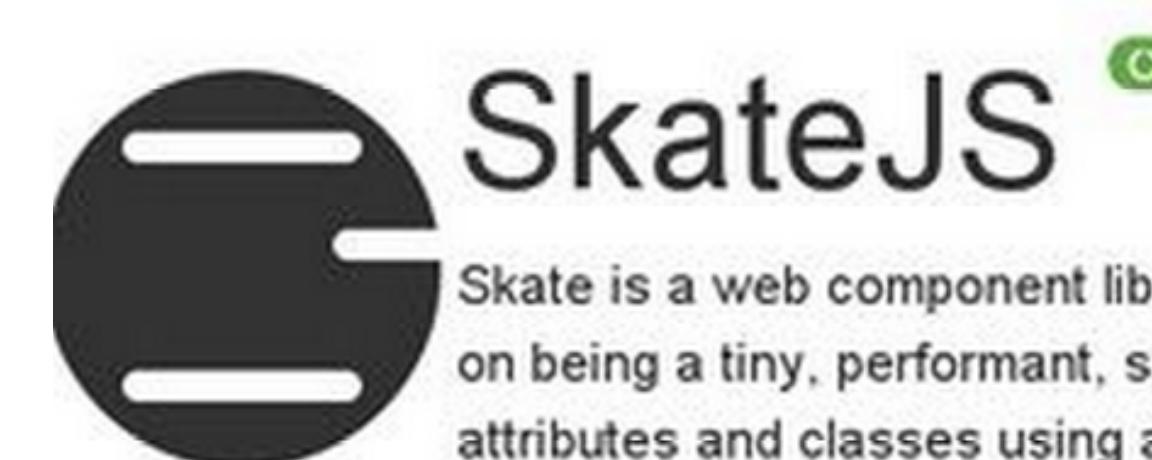
 npm install skatejs

 bower install skatejs

HTML

# this is how Googlebot saw the page

Search 



# SkateJS 0

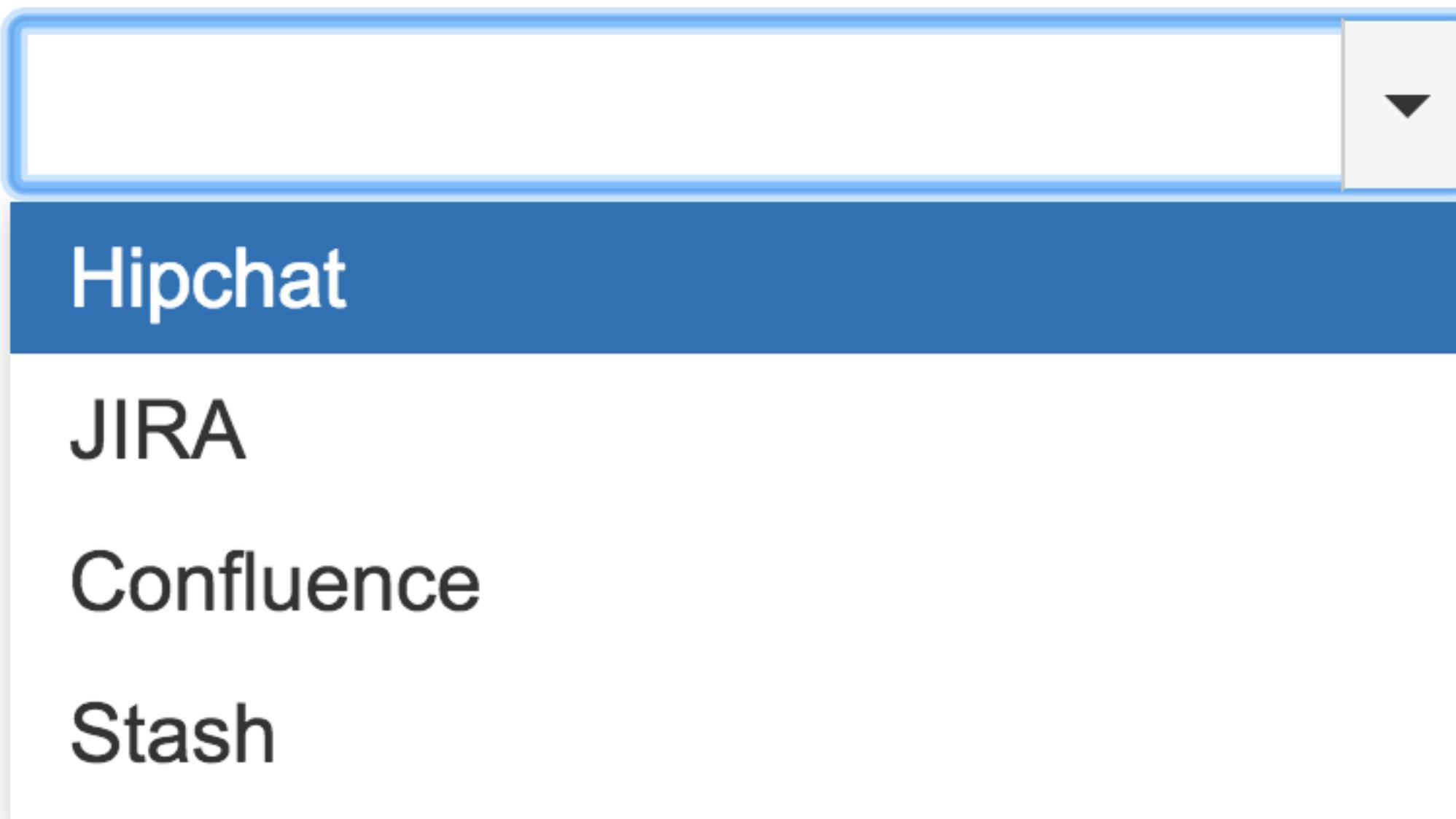
Skate is a web component library based on the custom element spec. It's focused on being a tiny, performant, syntactic-sugar for binding behaviour to elements, attributes and classes using a consistent, simple and declarative API.

Start Skating

npm install skatejs

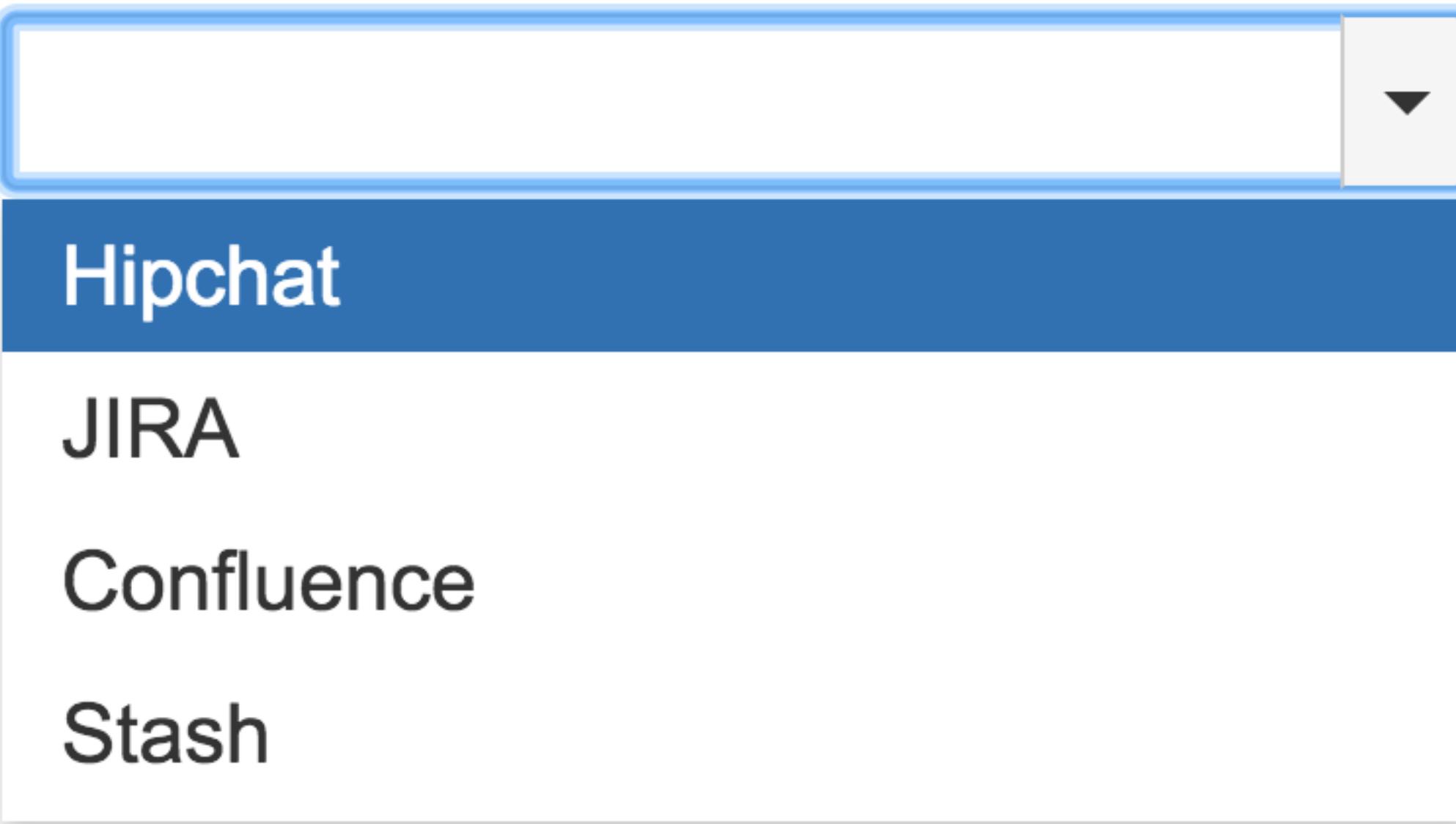
bower install skatejs

HTML



```
<aui-select id="my-picker">
<aui-option>Hipchat</aui-option>
<aui-option>JIRA</aui-option>
<aui-option>Confluence</aui-option>
<aui-option>Stash</aui-option>
</aui-select>
```

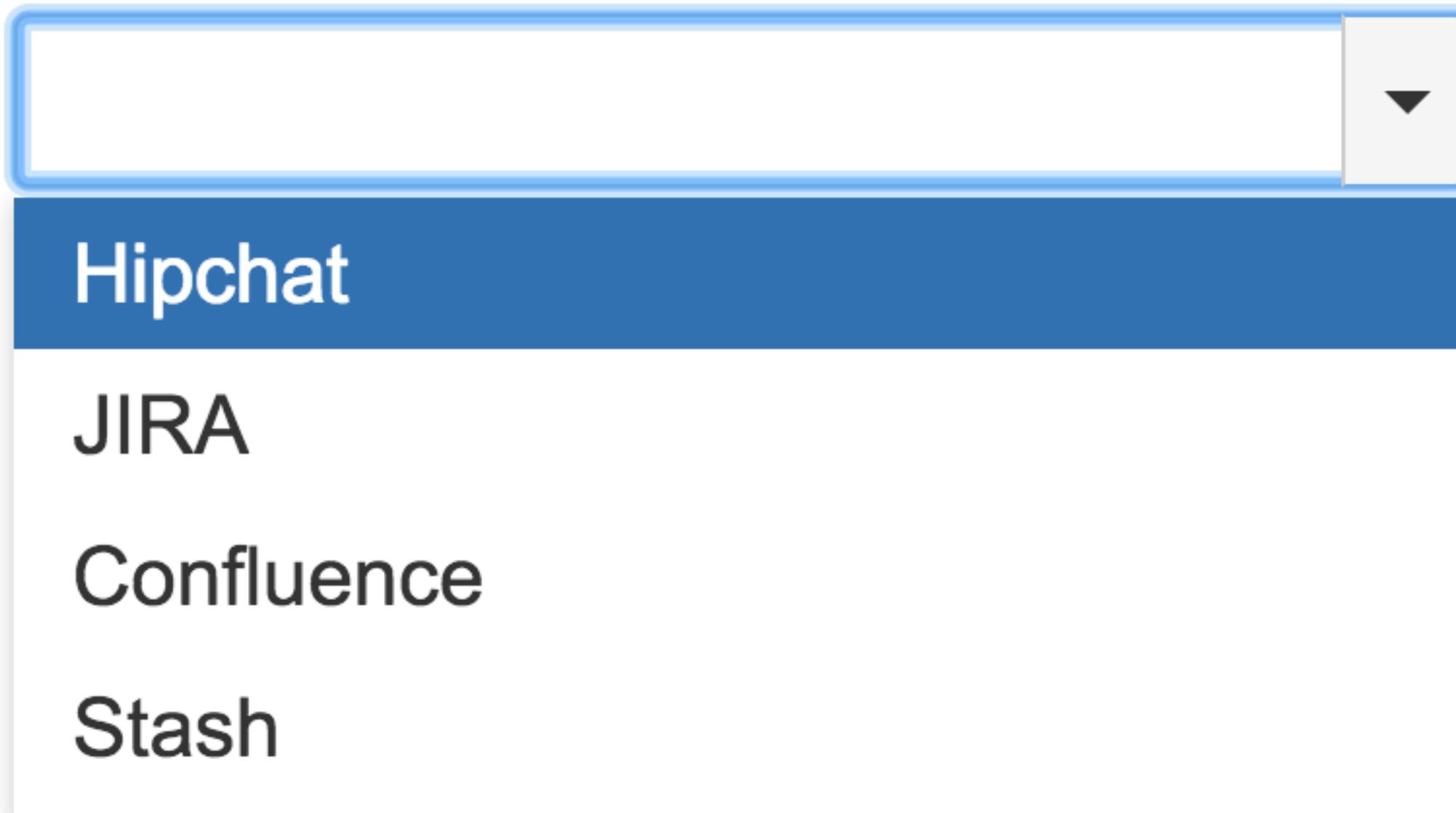
Product:



```
<label for="??????">  
  Product  
</label>
```

```
<aui-select id="my-picker">  
  ...  
</aui-select>
```

Product:



```
<aui-label for="my-picker">  
  Product  
</aui-label>
```

⇒

```
<aui-label for="my-picker">  
  <label for="__aui-my-picker">  
    Product  
  </label>  
</aui-label>
```

```
<aui-select id="my-picker">  
  ...  
</aui-select>
```

⇒

```
<aui-select id="my-picker">  
  <input id="__aui-my-picker">  
  ...  
</aui-select>
```

# What I hope you learned

they are ready to use now

they solve a bunch of problems

you will end up using them





# Thank you!



IAN BUCHANAN • DEVELOPER PARTISAN • ATLASSIAN • @DEVPARTISAN