



Чем полезен Polymer

Рассказывает Сергей Новиков,
UI/UX-разработчик в Grid Dynamics



О чём пойдет речь



О чём пойдет речь

- О легкой клиентской библиотеке
- Передовой
- Из которой можно сделать UI-“фреймворк”
- Обойденной большим вниманием
- Но применяющейся и развивающейся

О чём пойдет речь

- О легкой клиентской библиотеке

```
<script src="/bower_components/webcomponentsjs/webcomponents-loader.js"/>
<link rel="import" href="/bower_components/polymer/polymer.html">
```

Сама библиотека

- Но применяющейся и развивающейся

Полифил



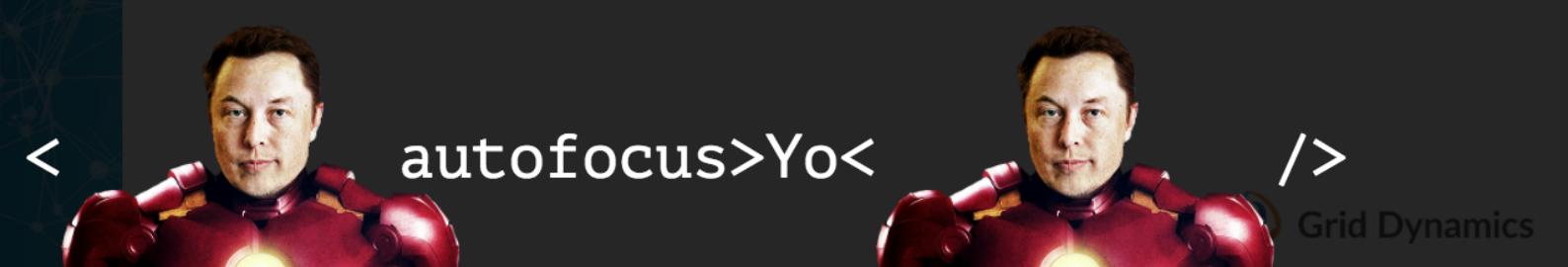
О чём пойдет речь

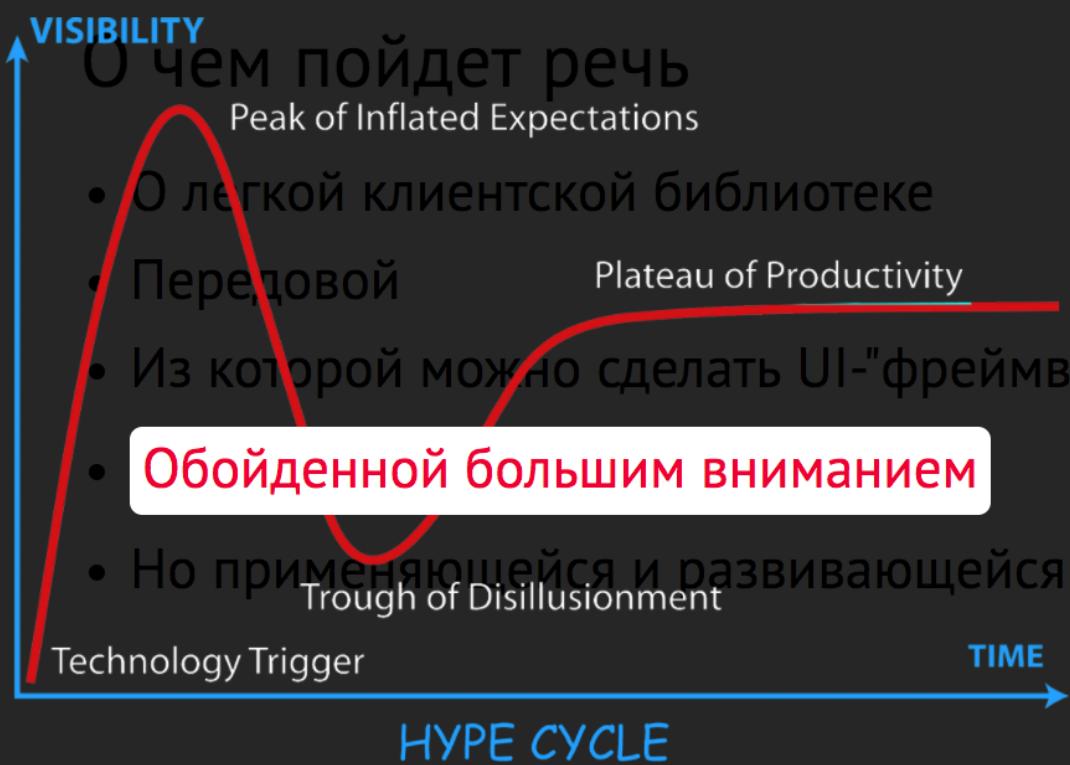
- О легкой клиентской библиотеке
- **Передовой**
- Из которой можно сделать UI-“фреймворк”
- Обойденной большим вниманием
- Но применяющейся и развивающейся



О чём пойдет речь

- О легкой клиентской библиотеке
- Передовой
- **Из которой можно сделать UI-“фреймворк”**
- Обойденной большим вниманием
- Но применяющейся и развивающейся





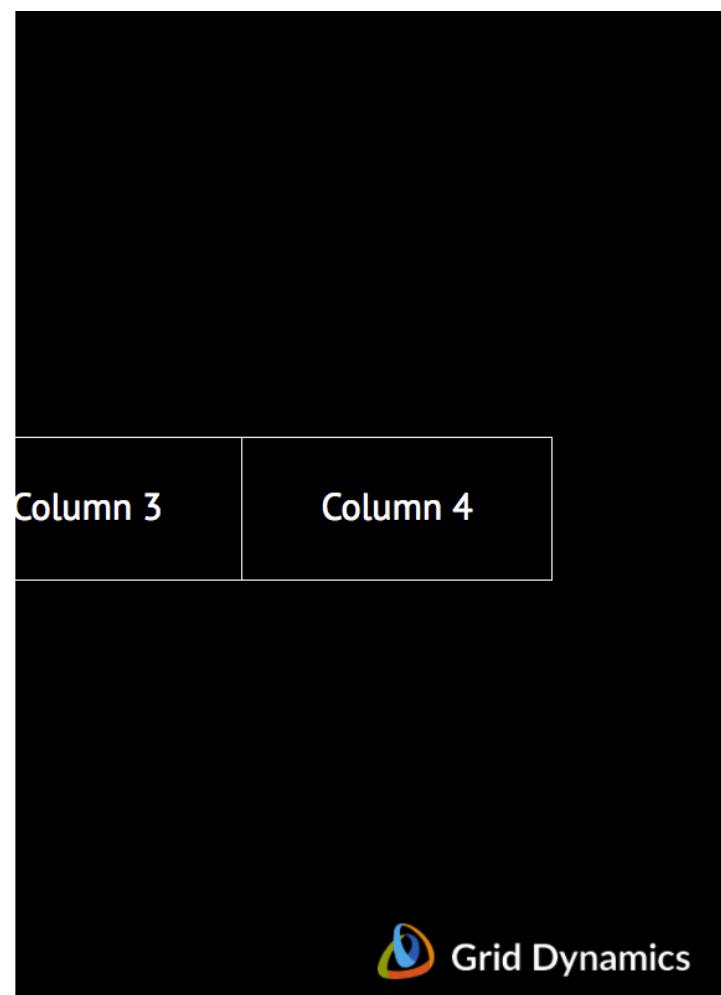
О чём пойдёт речь

- О логотипах
- Перевод логотипов
- Из каких языков можно сделать логотипы
- Обойленной большим вниманием

The Google logo, featuring its characteristic multi-colored letters.

В чём смысл Polymer?

```
<table>
  <tbody>
    <tr>
      <td>Column 1</td>
      <td>Column 2</td>
      <td>Column 3</td>
      <td>Column 4</td>
    </tr>
  </tbody>
</table>
```

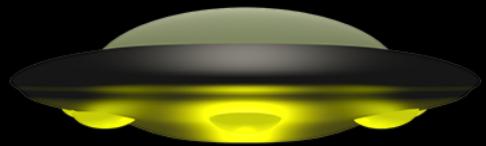


Column 3	Column 4

 Grid Dynamics

```
<ufo easter-egg="aliens"></ufo>
```



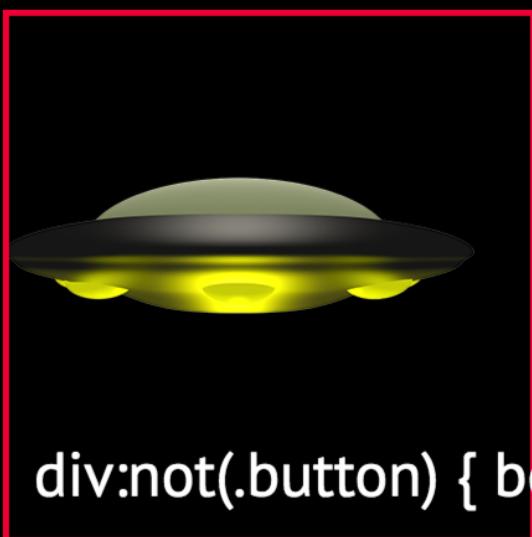


```
<div id="ufo"></div>

<script>
    $ufo = document.getElementById('ufo');
    $ufo.innerHTML = '';
    $ufo.onclick = showEasterEgg;
    spinUfo();

    function spinUfo() {
        // logic
    }

    function showEasterEgg () {
        // logic
    }
</script>
```



`div:not(.button) { border: 5px solid red; }`

```
<div class="ufo"></div>

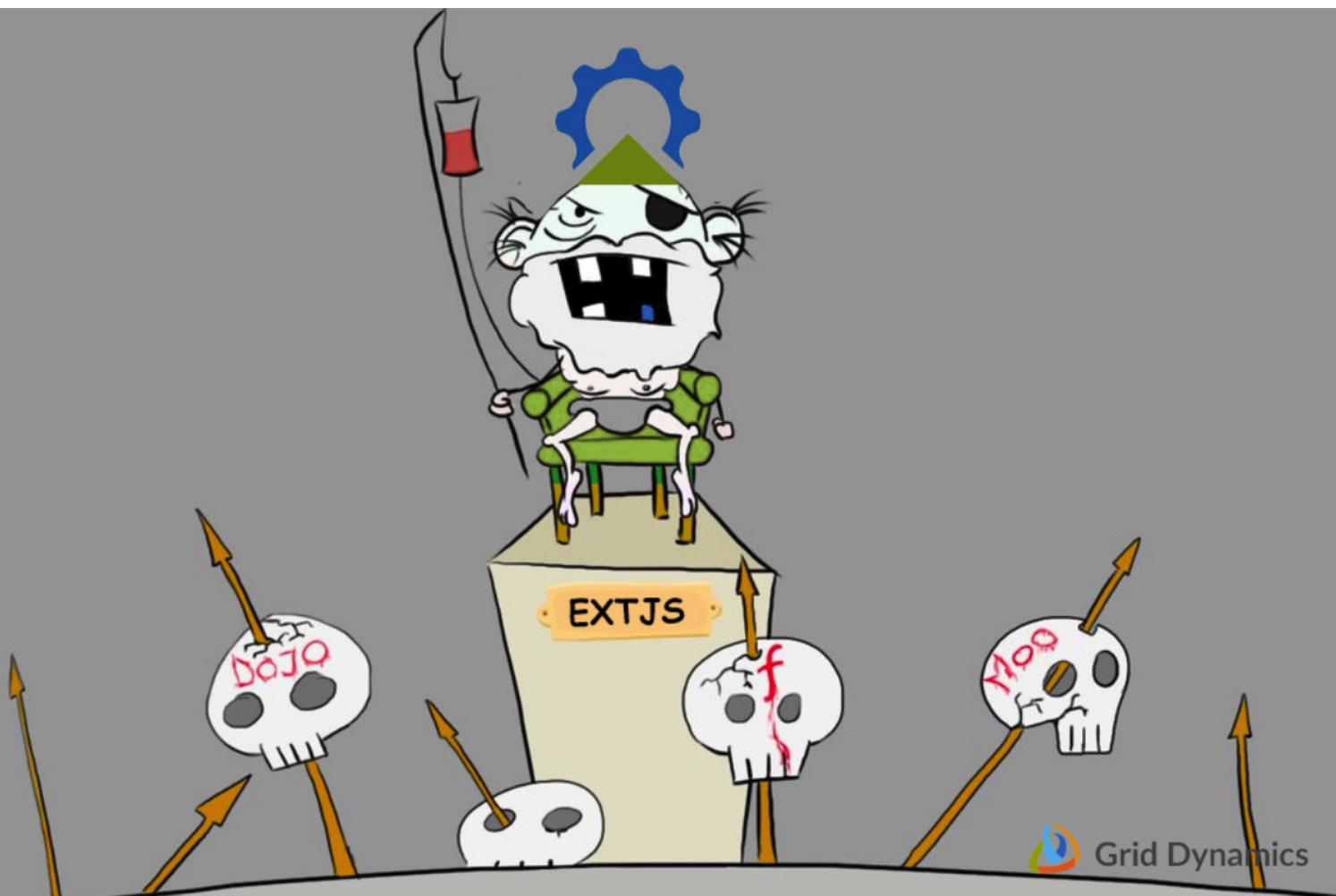
<script>
    jQuery('.ufo').initialize({
        easterEgg: 'aliens'
    });
</script>
```

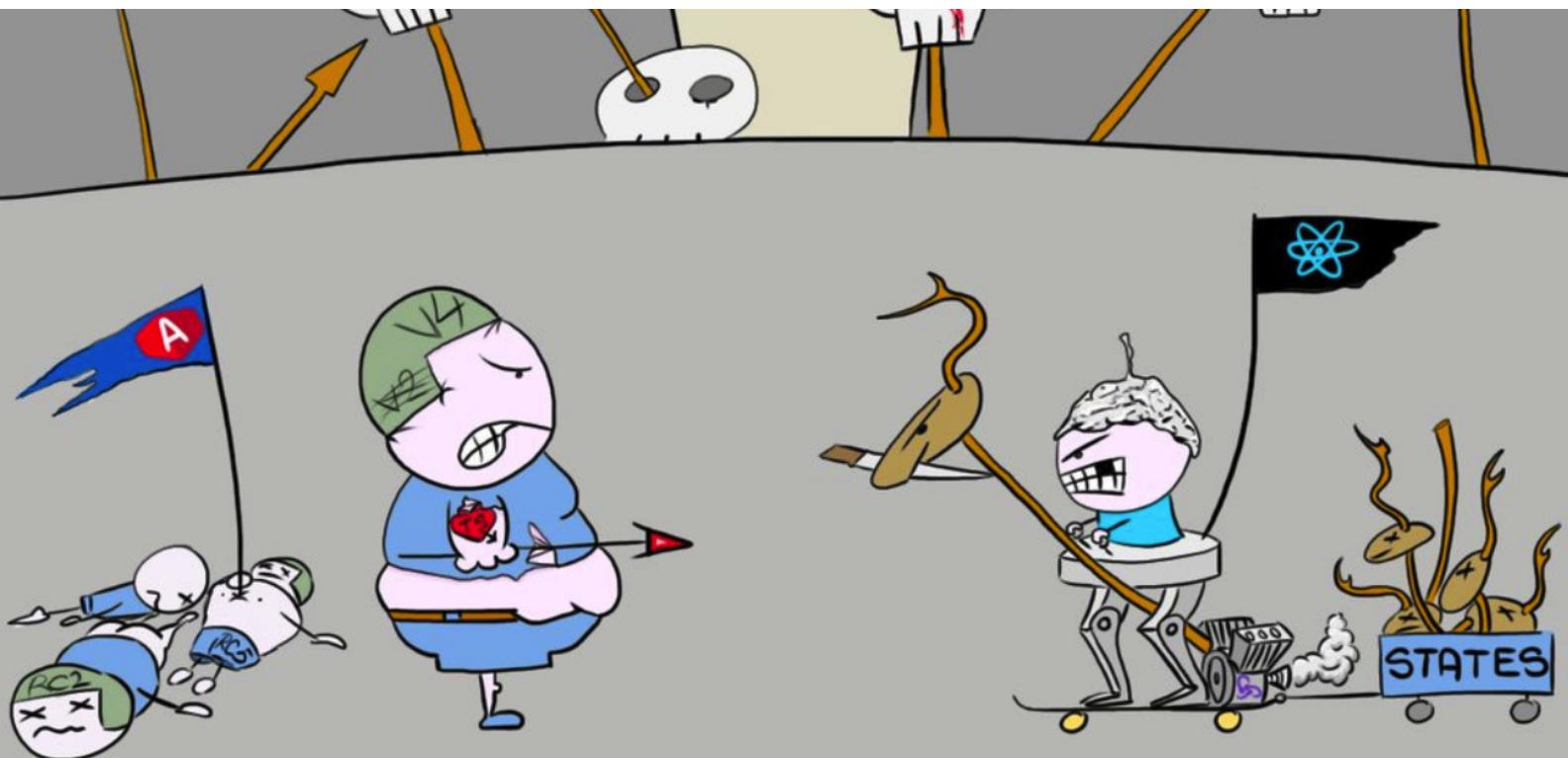


```
<div class="ufo" data-easter-egg="lama"></div>

<script>
  jQuery(".ufo").initialize();
</script>
```







MONKEYUSER.COM

(modified by Sergey Novikov)

 Grid Dynamics

```
<ufo easter-egg="aliens"></ufo>

<script>
    angular.
    module('ufoApp').
    component('ufo', {
        template: '',
        controller: function UfoController() {
            // logic
        }
    });
</script>
```



```
// ufo.scss
.ufo {
  &__image { ... }

  &--aliens { ... }
  &--lama { ... }
}

// ufo.css
.ufo { ... }
.ufo__image { ... }
.ufo--aliens { ... }
.ufo--lama { ... }
```



```
<Ufo easter-egg="aliens"/>

<script>
  class Ufo extends React.Component {
    render() {
      return 
    }

    spin(){} // logic
    showEasterEgg(){ // logic }
  }
</script>
```



```

<Ufo easter-egg="aliens"/>

<script>
  class Ufo extends React.Component {
    render() {
      return <img

```

Screenshot of a browser developer tools Elements tab showing the DOM structure and the Styles tab showing CSS rules for the 'Ufo' component.

Elements Tab:

```

<div id="app">
  <div data-reactroot data-reactid="1">
    <!-- react-empty: 2 -->
    <header class="_3sv0GYf3ZyVQXczKZo9Sg2" data-reactid="3">...</header>
    <main>
      <!-- react-empty: 230 -->
      <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
      <div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">
          <div class="_3Mh-MWj_wYU5Vr5A6r0VEK">...</div>
        </div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
        <div class="Bt9Ssj0ExpW6AUUm4sM32">...</div>
      ... div div div div a div._18yy0M3WFHElw91_-oPsPx._23Z3tCPR_9AkJM5s0lhUky

```

Styles Tab:

Filter: :hov .cls +

Style	Value
element.style	background-image: url(/dist/sunoco-banner-ee7c529...jpg);
@media print, screen and (min-width: 40em)	.18yy0M3WF main-66fc4768a...799269f6.css:1 HEIw91_-oPsPx { height: 500px; }
	.18yy0M3WF main-66fc4768a...799269f6.css:1 HEIw91_-oPsPx { height: 250px; }

```
<ufo easter-egg="aliens"></ufo>

<script>
    import {Component, OnInit} from '@angular/core';

    @Component({
        selector: 'ufo',
        template: '
```



```
<ufo easter-egg="aliens"></ufo>

<script>
  import {Component, OnInit} from '@angular/core';

  @Component({
    selector: 'ufo',
    template: '<img alt="ufo.png" (click)=>showEasterEgg()'
```

The screenshot shows a browser's developer tools open. The top navigation bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, Audits, Augury, and a menu icon. The Elements tab is active, displaying the DOM structure of the page. A red circle highlights the body element, which contains several script tags. The Styles panel is open on the right, showing the CSS styles applied to the body element. The styles listed are:

```
element.style {
}
body {
  font-family: 'Open Sans', sans-serif;
  font-size: 1.6rem;
  color: #444444;
  background-color: #f6f6f6;
}
html * {
  box-sizing: border-box;
}
body {
  line-height: 1;
```

```

<ufo easter-egg="aliens"></ufo>

<script>
  import {Component, OnInit} from '@angular/core';

```

```
@Component({
```

```
  selector: 'ufo',
```

```
  template: '<img alt="ufo.png" (click)=showEasterEgg()>
```

Elements Console Sources Network Performance Memory Application Security Audits Augury

main[_ngcontent-txf-7]

```

<!--template bindings={}-->


<nav _ngcontent-txf-7>...</nav>
  <div _ngcontent-txf-7 class="items-container">
    <!--template bindings={}-->
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6> == $0
      <div _ngcontent-txf-6 class="card">
        <div _ngcontent-txf-6 class="image">...</div>
        <div _ngcontent-txf-6 class="description">...</div>
      </div>
    </ml-item-card>
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6>...</ml-item-card>
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6>...</ml-item-card>
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6>...</ml-item-card>
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6>...</ml-item-card>
    <ml-item-card _ngcontent-txf-7 _nghost-txf-6>...</ml-item-card>
  </div>


```

... div ml-client all-items main div ml-item-card div.card div.image a.overlay

Styles Computed Event Listeners »

Filter :hov .cls +

```

element.style {
}

main[_ngcontent-txf-7]      <style>...</style>
.items-
.container[_ngcontent-txf-7] ml-item-
card[_ngcontent-txf-7] {
  -webkit-box-flex: 1;
  -ms-flex-positive: 1;
  flex-grow: 1;
  -ms-flex-preferred-size: 19rem;
  flex-basis: 19rem;
  width: 19rem;
  max-width: 30rem;
  font-size: 1.6rem;
  margin: 0 0 2.3rem 2.3rem;
}

```





npm



 Grid Dynamics



Нам нужны компоненты:

- Максимально модульные
- Не требующие этапа компиляции
- С нормальной изоляцией
- И стандартными API для всего

Web Components



```
<!doctype html>
<html>
  <head>
    <title>App</title>
    <link rel="import" href="ufo.html">
  </head>

  <body>
    <my-ufo easter-egg="aliens"></my-ufo>
  </body>
</html>
```

```
<template id="my-ufo">
  <style> ... </style>
  
</template>

<script>
  class MyUfo extends HTMLElement {
    connectedCallback() {
      const shadowRoot = this.attachShadow({ mode: 'open' });

      const tpl = document.querySelector('#my-ufo');
      const instance = tpl.content.cloneNode(true);
      shadowRoot.appendChild(instance);

      document.addEventListener('click', this.showEasterEgg);
    }
  }

  customElements.define('my-ufo', MyUfo);
</script>
```



```
<!doctype html>
<html>
  <head>
    <title>App</title>
```

The screenshot shows the browser's developer tools with the "Elements" tab selected. The main pane displays a hierarchical tree of HTML elements, including a nested shadow root. The "Styles" tab in the panel on the right is active, showing a list of styles applied to the selected element. A blue bar at the bottom highlights the "#shadow-root" node in the element tree.

Elements Console Sources Network Performance Memory Application Security »

... #shadow-root (open) == \$0

- ▶ <style>...</style>
- ▼ <meet-slides id="slides">
- ▼ #shadow-root (open)
- ▼ <meet-slide-17>
- ▼ #shadow-root (open)
- ▶ <style>...</style>
- ▶ <div id="rendered" style="width: 1010px;">...</div>
- ▶ <div id="markup">...</div>
- </meet-slide-17>
- </meet-slides>
- ▶ <meet-nav>...</meet-nav>
- </meet-app>
- </body>

Styles Computed Event Listeners »

Filter :hov .cls +

html body meet-app #shadow-root

Что еще улучшить?

- Нестабильный API
- Слабая поддержка браузерами
- Отсутствуют:
 - Биндинги
 - Наблюдаемые и вычисляемые свойства
 - Логика в шаблонах
 - Минимальный Data flow



Polymer

Нет системы сборки

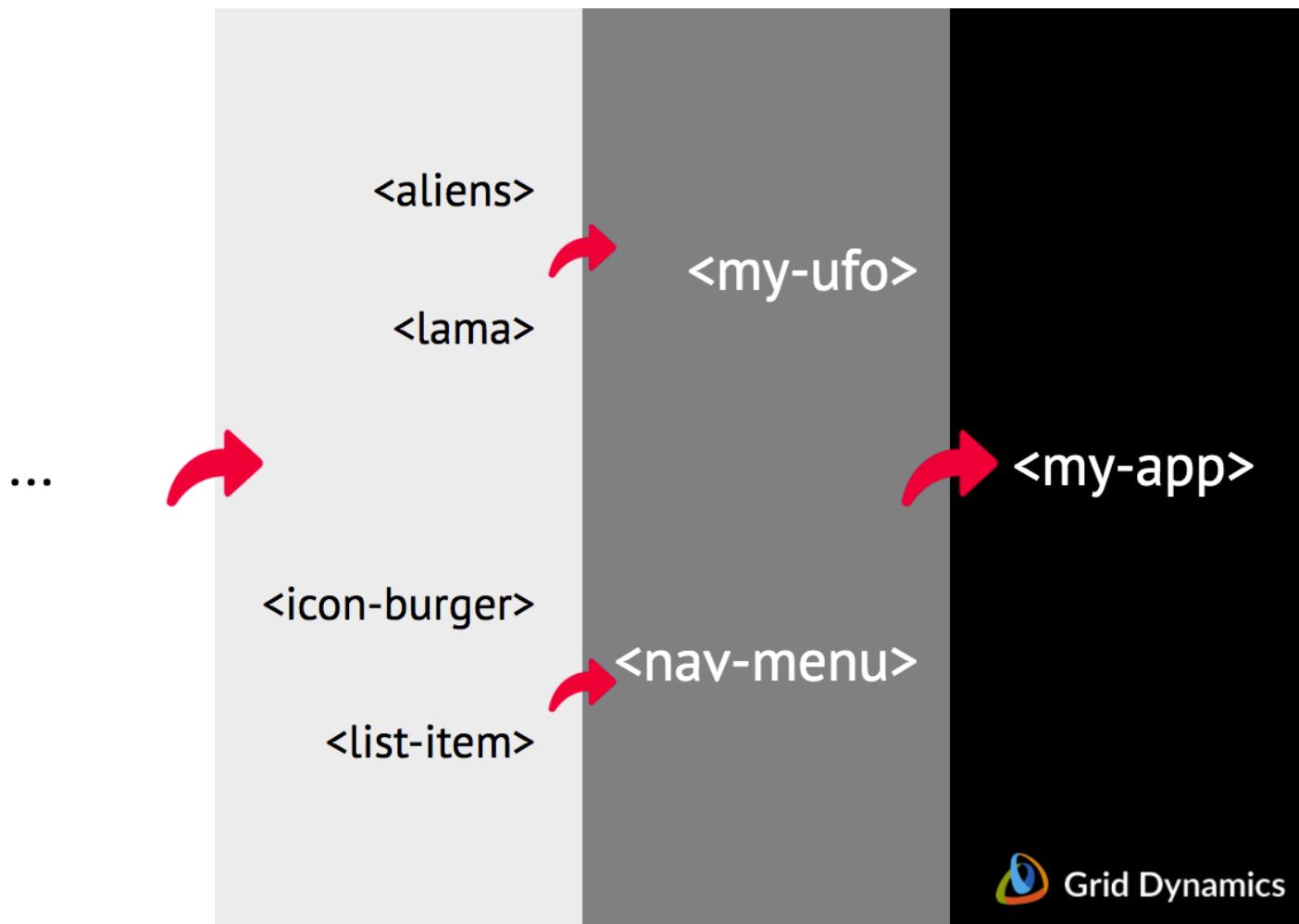
Sources > (index) meet-slide-20.html meet-slide-1.js >

```
8      type: boolean,
9      value: false
10 }
11 }
12 }
13 ready() {
14   super.ready();
15   setTimeout(() => {
16     this.set('disabled', true);
17   }, 1000);
18 }
19 }
20 window.customElements.define(MeetSlide1.is, MeetSlide1);
21
```

{ } Line 14, Column 9

Paused on breakpoint

- Watch
- Call Stack
- Scope
 - Local
 - this: meet-slide-1
 - Script
 - MeetApp: class MeetApp
 - MeetNav: class MeetNav
 - MeetSlide1: class MeetSlide1
 - MeetSlide2: class MeetSlide2
 - MeetSlide3: class MeetSlide3



<karate>

<lama>

<my-ufo>

```
<link rel="import" href="app.bundle.html">
```

<icon-burger>

<list-item>

<nav-menu>

 Grid Dynamics

 Grid Dynamics

```
snovikov@snowikov-ws (dev)*$ ng serve --env=local

** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200
Hash: 5182365a0f8ffa2e416c
Time: 25112ms
chunk {0} main.bundle.js, main.bundle.js.map (main) 174 kB {3} [initial] [rendered]
chunk {1} scripts.bundle.js, scripts.bundle.js.map (scripts) 479 kB {4} [initial] [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 29.7 kB {4} [initial] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 3.8 MB [initial] [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]

webpack: Compiled successfully.
```



```
snovikov@snowikov-ws client $ polymer serve  
info: Files in this directory are available under the following URLs  
applications: http://127.0.0.1:8081  
reusable components: http://127.0.0.1:8081/components/Polymer Meet-up/
```



Нативный шаблонизатор



```
import UfoTemplate from './ufo.html';
UfoTemplate.querySelector('img').doSomething();
```

OR

```
const UfoTemplate = document.querySelector('template#ufo-template');
UfoTemplate.querySelector('img').doSomething();
```

OR

```
export `<img src='ufo.png' class='ufo-animation'>
...
import UfoTemplateString from './ufo.template.js';
...
UfoTemplate.innerHTML = UfoTemplateString;
UfoTemplate.querySelector('img').doSomething();
```

```
// ufo.component.html
<link rel="import" href="../polymer/polymer-element.html">

<dom-module id="ufo">
  <template>
    <img src='ufo.png' class='ufo-animation'>
  </template>

  <script>
    class Ufo extends Polymer.Element {
      ...
      this.shadowRoot.querySelector('img').doSomething();
    }
  </script>
</dom-module>
```



Однофайловые компоненты



```
<link rel="stylesheet" href="../../shared/shared.css">

<dom-module id="my-ufo">
  <template>
    <style>
      img { ... }

      @keyframes ufo-animation {
        ...
      }
    </style>

    <img src='ufo.png' class='ufo-animation'>
  </template>

  <script>
    class MyUfo extends Polymer.Element {
      static get is() { return 'my-ufo'; }
      static get properties() { ... }
      ready() { ... }
      showEasterEgg() { ... }
    }

    window.customElements.define(MyUfo.is, MyUfo);
  </script>
```



```
<dom-module id= my-ufo >
  <template>
    <style>
      img { ... }

      @keyframes ufo-animation {
        ...
      }
    </style>
    <dom-if if="{user.isAdmin}">
      <img src='ufo.png' class='ufo-animation'>
    </dom-if>
  </template>
  <script>
    class MyUfo extends Polymer.Element {
      static get is() { return 'my-ufo'; }
      static get properties() { ... }
      ready() { ... }
      showEasterEgg() { ... }
    }

    window.customElements.define(MyUfo.is, MyUfo);
  </script>
</dom-module>
```



```
class MyUfo extends Polymer.Element {  
    static get is() { return 'my-ufo'; }  
  
    created() { ... }  
    ready() { ... }  
    connectedCallback() { ... }  
    disconnectedCallback() { ... }  
    attributeChangedCallback() { ... }  
  
    static get properties() { ... }  
  
    userDefinedMethod() { ... }  
}  
  
window.customElements.define(MyUfo.is, MyUfo);
```



```
class MyUfo extends Polymer.Element {  
    static get is() { return 'my-ufo'; }  
  
    created() { ... }  
    ready() { ... }  
    connectedCallback() { ... }  
    disconnectedCallback() { ... }  
    attributeChangedCallback() { ... }  
  
    static get properties() { ... }  
  
    userDefinedMethod() { ... }  
}  
  
window.customElements.define(MyUfo.is, MyUfo);
```



```
class MyUfo extends Polymer.Element {  
    static get is() { return 'my-ufo'; }  
  
    created() { ... }  
    ready() { ... }  
    connectedCallback() { ... }  
    disconnectedCallback() { ... }  
    attributeChangedCallback() { ... }  
  
    static get properties() { ... }  
  
    userDefinedMethod() { ... }  
}  
  
window.customElements.define(MyUfo.is, MyUfo);
```



```
class MyUfo extends Polymer.Element {  
    static get is() { return 'my-ufo'; }  
  
    created() { ... }  
    ready() { ... }  
    connectedCallback() { ... }  
    disconnectedCallback() { ... }  
    attributeChangedCallback() { ... }  
  
    static get properties() { ... }  
  
    userDefinedMethod() { ... }  
}  
  
window.customElements.define(MyUfo.is, MyUfo);
```



```
class MyUfo extends Polymer.Element {  
    static get is() { return 'my-ufo'; }  
  
    created() { ... }  
    ready() { ... }  
    connectedCallback() { ... }  
    disconnectedCallback() { ... }  
    attributeChangedCallback() { ... }  
  
    static get properties() { ... }  
  
    userDefinedMethod() { ... }  
}  
  
window.customElements.define(MyUfo.is, MyUfo);
```



```
class MyUfo extends Polymer.Element {  
  static get properties() {  
    return {  
      pilot: {  
        type: Object  
      },  
  
      easterEgg: {  
        type: String,  
        value: 'aliens',  
        observer: '_renderNewEasterEgg'  
      },  
  
      launchMissileAt: {  
        type: Date,  
        computed: 'getLaunchMissileTime(currentPos, targetPos)'  
      }  
    };  
  }  
}
```



```
class MyUfo extends Polymer.Element {  
  static get properties() {  
    return {  
      pilot: {  
        type: Object  
      },  
  
      easterEgg: {  
        type: String,  
        value: 'aliens',  
        observer: '_renderNewEasterEgg'  
      },  
  
      launchMissileAt: {  
        type: Date,  
        computed: 'getLaunchMissileTime(currentPos, targetPos)'  
      }  
    };  
  }  
}
```



```
class MyUfo extends Polymer.Element {
  static get properties() {
    return {
      pilot: {
        type: Object
      },
      easterEgg: {
        type: String,
        value: 'aliens',
        observer: '_renderNewEasterEgg'
      },
      launchMissileAt: {
        type: Date,
        computed: 'getLaunchMissileTime(currentPos, targetPos)'
      }
    };
  }
}
```



Настоящая изоляция



```
<image-slider>
    
    
    
</image-slider>
```



```
<image-slider>
  
  
  
</image-slider>
```



```
<div class="image-slider">
  <slot name="image"></slot>
  <div class="image-title">[[currentImage.alt]]</div>

  <div class="arrow-left"></div>
  <div class="arrow-right"></div>

  <div class="zoomed-area"></div></div>
</div>
```

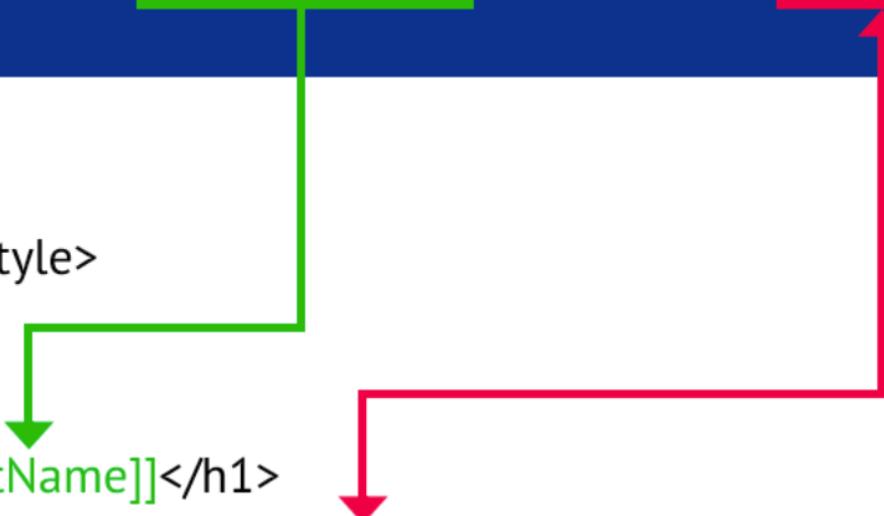
Data Flow & Change Detection

```
<ufo pilot-name=[[pilot.name]] easter-egg={{lama}}></ufo>
```

...

```
<style> ... </style>
```

```
<div>
  <h1>[[pilotName]]</h1>
  <easter-egg type={{easterEggType}}></easter-egg>
  <img src='ufo.png'>
</div>
```



```
// под капотом Polymer:  
set pilotName(value) {  
    if (this._pilotName === value) return;  
  
    this._pilotName = value;  
    // обновляем все биндинги  
    // триггерим observers, если есть  
}
```

```
// под капотом Polymer:  
set pilotName(value) {  
    if (this._pilotName === value) return;  
  
    this._pilotName = value;  
    // обновляем все биндинги  
    // триггерим observers, если есть  
    // триггерим событие 'pilot-name-changed'  
}
```

Что в итоге?

Очень легкая и быстрая библиотека

Позволяет делать сложные вещи проще,
чем в популярных фреймворках

Недостаточно стабильная

Уже сейчас полезна для быстрой разработки,
прототипирования
и создания высокопроизводительных приложений и модулей



Polymer