Webcomponents

Everything is AWESOME!

Components

“An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data).”

Component-based software engineering

Wikipedia

EU EDGE
Sharp Minds

Components

- encapsulation
- reusability
- communication via interfaces

EU EDGE
Sharp Minds

Components

- encapsulation
- reusability
- communication via interfaces

Today: components for the web ~= jquery plugins

EU EDGE
Sharp Minds

Components

Example: Lightbox http://lokeshdhakar.com/projects/lightbox2/

```
<script src="js/jquery-1.11.0.min.js"></script>
<script src="js/lightbox.min.js"></script>
<link href="css/lightbox.css" rel="stylesheet" />

<a href="img/image-1.jpg" data-lightbox="i1" data-title="My caption">
    <img src="img/image-1.jpg">
</a>
<a href="img/image-2.jpg" data-lightbox="i1" data-title="My caption2">
    <img src="img/image-2.jpg">
</a>
```

EU EDGE
Sharp Minds

Components

Example: Lightbox http://lokeshdhakar.com/projects/lightbox2/

```
<script src="js/jquery-1.11.0.min.js"></script>
<script src="js/lightbox.min.js"></script>
<link href="css/lightbox.css" rel="stylesheet" />

<a href="img/image-1.jpg" data-lightbox="i1" data-title="My caption">
    <img src="img/image-1.jpg">
</a>
<a href="img/image-2.jpg" data-lightbox="i1" data-title="My caption2">
    <img src="img/image-2.jpg">
</a>
```

But I'm using YUI :(

three request?

HTML Captions?

EU EDGE
Sharp Minds

Components

Example: Lightbox

```
<link rel=”import” href=”//cdn.net/elements/lightbox.html”>

<lightbox-images>
  <a href=”img/image-1.jpg”>
    <figure>
        <img src=”img/image-1.jpg”>
        <figcaption><b>Lorem</b> Ipsum!</figcaption>
    </figure>
  </a>
</lightbox-images>
```

EU EDGE
Sharp Minds

Components

Example: prism.js

```
▼<prism-js language="javascript" linenumbers>
  ▶#shadow-root
    "
    if (true) {
        console.log('foo');
    }
    "
  </prism-js>
```

```
▼<pre class=" language-javascript">
  ▼<code class=" language-javascript" data-language="javascript">
      <span class="token lf">
      </span>
      <span class="token keyword">if</span>
      <span class="token punctuation">(</span>
      <span class="token boolean">true</span>
      <span class="token punctuation">)</span>
      <span class="token punctuation">{</span>
      <span class="token lf">
      </span>
      "    console"
      <span class="token punctuation">.</span>
    ▶<span class="token function">…</span>
      <span class="token string">'foo'</span>
      <span class="token punctuation">)</span>
      <span class="token punctuation">;</span>
      <span class="token lf">
      </span>
      <span class="token punctuation">}</span>
      <span class="token lf">
      </span>
    </code>
  </pre>
```

javascript

```javascript
if (true) {
    console.log('foo');
}
```

EU EDGE
Sharp Minds

Components

Example: prism.js

```
▼<prism-js language="javascript" linenumbers>
  ▼#shadow-root
    <style>
            :host { display: block;};
        </style>
   ▼<pre class="line-numbers language-javascript">
    ▼<code id="codeviewer" class=" language-javascript">
       <span class="token keyword">if</span>
       <span class="token punctuation">(</span>
       <span class="token boolean">true</span>
       <span class="token punctuation">)</span>
       <span class="token punctuation">{</span>
       "
           console"
       <span class="token punctuation">.</span>
      ▶<span class="token function">…</span>
       <span class="token string">'foo'</span>
       <span class="token punctuation">)</span>
       <span class="token punctuation">;</span>
       <span class="token punctuation">}</span>
      ▶<span class="token line-numbers-rows">…</span>
     </code>
    </pre>
  "
  if (true) {
      console.log('foo');
  }
  "
</prism-js>
```

```
▼<pre class=" language-javascript">
  ▼<code class=" language-javascript" data-language="javascript">
     <span class="token lf">
     </span>
     <span class="token keyword">if</span>
     <span class="token punctuation">(</span>
     <span class="token boolean">true</span>
     <span class="token punctuation">)</span>
     <span class="token punctuation">{</span>
     <span class="token lf">
     </span>
     "    console"
     <span class="token punctuation">.</span>
    ▶<span class="token function">…</span>
     <span class="token string">'foo'</span>
     <span class="token punctuation">)</span>
     <span class="token punctuation">;</span>
     <span class="token lf">
     </span>
     <span class="token punctuation">}</span>
     <span class="token lf">
     </span>
   </code>
</pre>
```

EU EDGE
Sharp Minds

Components

Example: prism.js

```
▼<prism-js language="javascript" linenumbers>
  ▼#shadow-root
    <style>
          :host { display: block;};
      </style>
   ▼<pre class="line-numbers language-javascript">
    ▼<code id="codeviewer" class=" language-javascript">
       <span class="token keyword">if</span>
       <span class="token punctuation">(</span>
       <span class="token boolean">true</span>
       <span class="token punctuation">)
       <span class="token punctuation">{
       "
           console"
       <span class="token punctuation">.
      ▶<span class="token function">…</span>
       <span class="token string">'foo'</span>
       <span class="token punctuation">)</span>
       <span class="token punctuation">;</span>
       <span class="token punctuation">}</span>
      ▶<span class="line-numbers-rows">…</span>
     </code>
    </pre>
  "
  if (true) {
      console.log('foo');
  }
  "
</prism-js>
```

span { display:block }

```
▼<pre class=" language-javascript">
  ▼<code class=" language-javascript" data-language="javascript">
     <span class="token lf">
     </span>
     <span class="token keyword">if</span>
     <span class="token punctuation">(</span>
     <span class="token boolean">true</span>
     <span class="token punctuation">)</span>
     <span class="token punctuation">{</span>
     <span class="token lf">
                                    ole"
                                    ss="token punctuation">.</span>
                                    ss="token function">…</span>
                                    ss="token string">'foo'</span>
     <span class="token punctuation">)</span>
     <span class="token punctuation">;</span>
     <span class="token lf">
     </span>
     <span class="token punctuation">}</span>
     <span class="token lf">
     </span>
   </code>
  </pre>
```

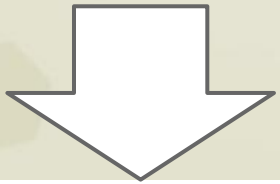
EU EDGE
Sharp Minds

ShadowDOM - encapsulation

Shadow DOM

http://rszaloki.github.io/webcomponents/shadow_dom/index.html

```
<div id=”foo”>Light DOM</div>
<script>
    var foo = document.querySelector(‘#foo’);
    shadow = foo.createShadowRoot();
    shadow.innerHTML = “Shadow DOM”;
</script>
```

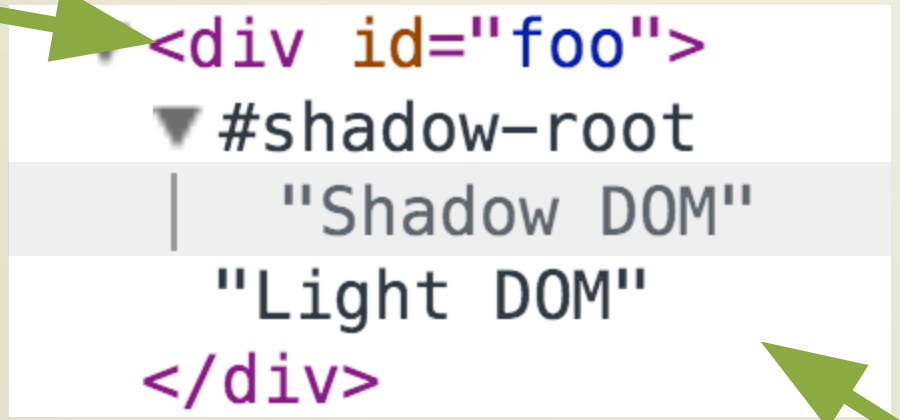


Shadow DOM

EU EDGE
Sharp Minds

Shadow DOM

Shadow host =
has a shadow root



Normal DOM under the
shadow host =
Light DOM

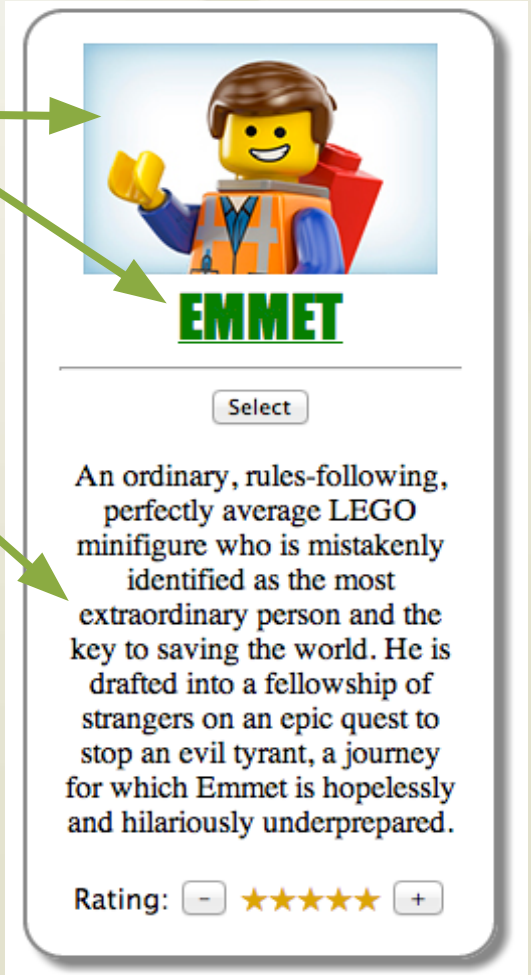
The content of a shadow host isn't rendered; the content of the shadow root is rendered instead.

EU EDGE
Sharp Minds

```
<div id="lego-name">
 <img src="img/emmet.jpg">
 <a href="">EMMET</a>
 <p>An ordinary...</p>
 <input type="text"
        value="5"
        is="x-rating">
</div>
```

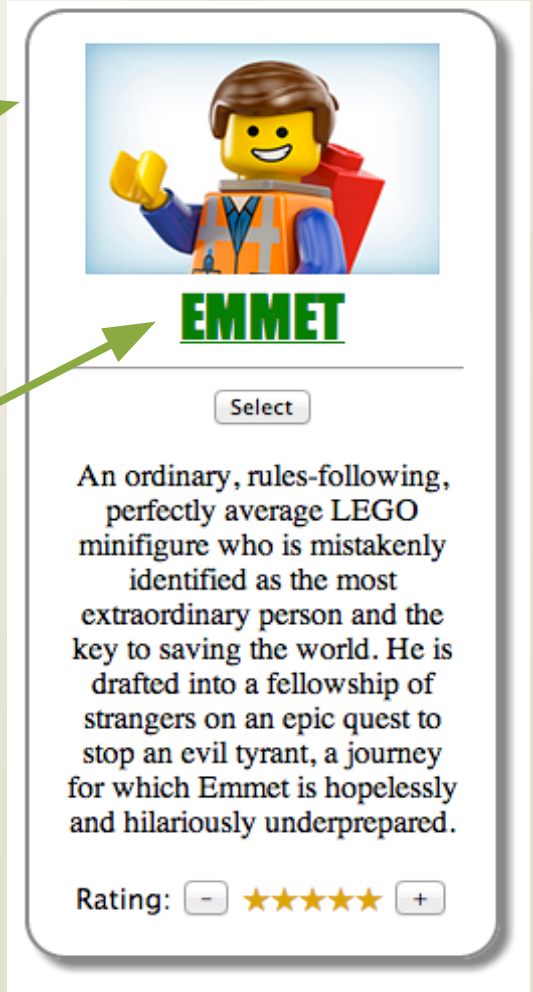
```
<content select="img"></co
<content select="a"></cont
<hr>
<button>Select</button>
<content></content>
```

```
<script>
    var foo = document.querySelector('#lego-name);
    shadow = foo.createShadowRoot();
    shadow.innerHTML = "          ";
</script>
```



```
<div id="lego-name">
 <img src="img/emmet.jpg">
 <a href="">EMMET</a>
 <p>An ordinary...</p>
 <input type="text"
        value="5"
        is="x-rating">
</div>
```

```
<style>
:host {
 text-align:center;
 display:inline-block;
 width:200px;
 ...
}
:host a
::content a {
 font-family: impact;
 font-size:30px;
 color:green;
}
</style>
```



```
<input is="x-rating"
       value="2">

<input is="x-rating"
       value="2"
       readonly>
```

```
<style>
:host {
 border:0;
 color:black;
}

:host([readonly]) button {
 display:none
}
</style>
<label>Rating: </label>
<button id="dec">-</button>
<span id="value"></span>
<button id="inc">+</button>
```

EU EDGE
Sharp Minds

Shadow DOM

http://rszaloki.github.io/webcomponents/shadow_dom/index.html

New selectors in the shadow:

- :host, :host(.selector), :host(:hover)
- ::content
- :host-context(.selector)

EU EDGE
Sharp Minds

Shadow DOM

http://rszaloki.github.io/webcomponents/shadow_dom/index.html

Pierce through the boundary:

- ::shadow
  selects the elements shadow root
- /deep/
  ignores the boundary

EU EDGE
Sharp Minds

Shadow DOM

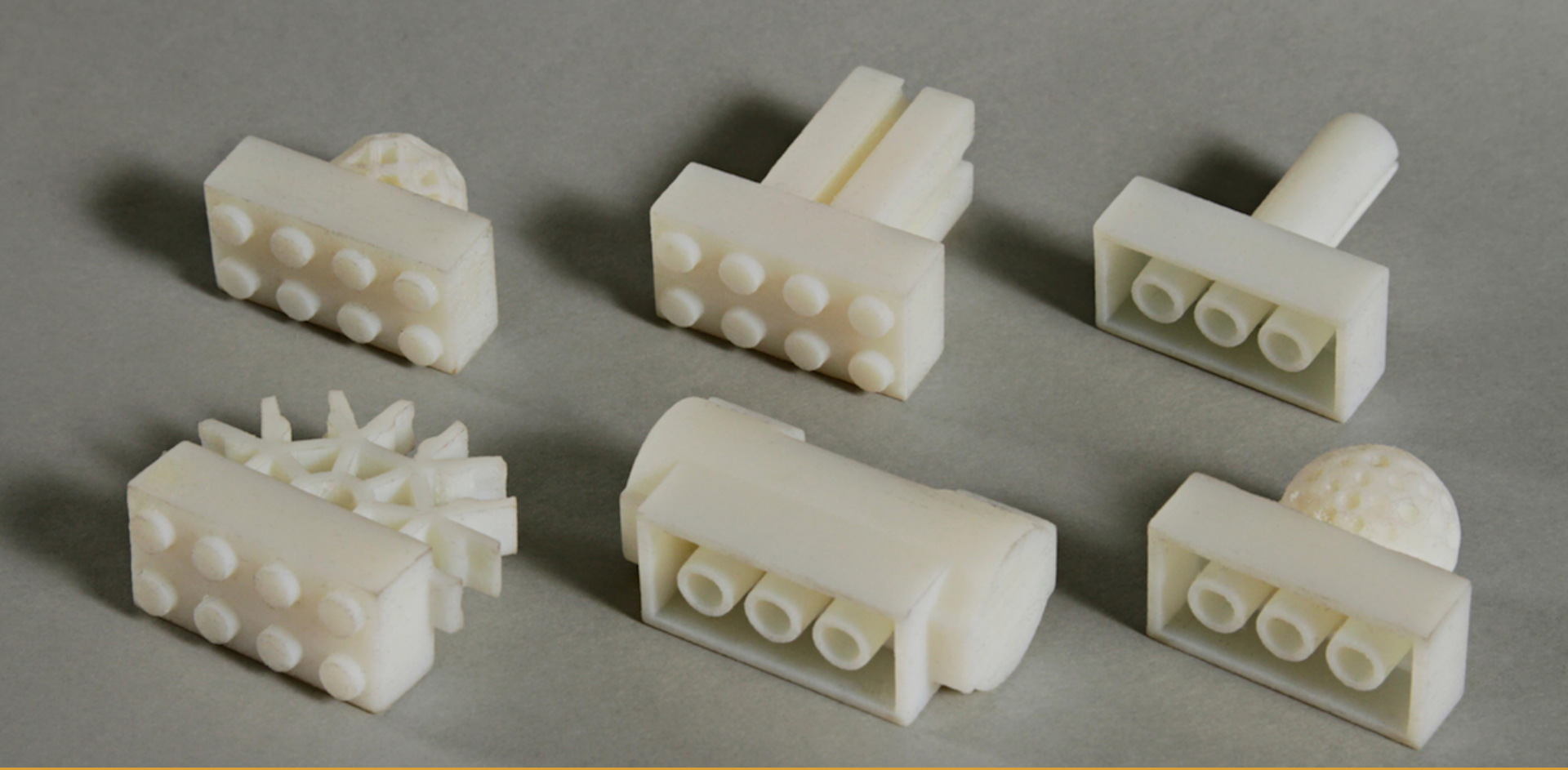
http://rszaloki.github.io/webcomponents/shadow_dom/index.html

Events that are **always stopped** at the boundary:

- abort
- error
- select
- change
- load
- reset
- resize
- scroll
- selectstart

Others are **retargeted**: the event is changed, to look like, they've come from the host element

EU EDGE
Sharp Minds



Custom Elements - reuse

Custom elements

http://rszaloki.github.io/webcomponents/custom_elements/index.html

```
var LegoName = document.registerElement('lego-name');

var e1 = new LegoName();

var e2 = document.createElement('lego-name');

document.body.appendChild(e1);

document.body.appendChild(e2);

<lego-name></lego-name>
```

EU EDGE
Sharp Minds

Custom elements

http://rszaloki.github.io/webcomponents/custom_elements/index.html

Lifecycle events:

- createdCallback
- attachedCallback
- detachedCallback
- attributeChangedCallback

EU EDGE
Sharp Minds

Custom elements

```
var LNameProto = Object.create(HTMLElement.prototype);
LNameProto.createdCallback = function(){
   this.innerHTML = “Hello!”
}
var LName = document.registerElement('lego-name',{
   prototype:LNameProto
});
```

```
     <lego-name></lego-name> - source
<lego-name>Hello</lego-name> - DOM
```

EU EDGE
Sharp Minds

Custom elements

http://rszaloki.github.io/webcomponents/custom_elements/index.html

```
document.registerElement('lego-name', {
    prototype: prototype object,
    "extends":'li'
});

<li is="lego-name">
```

EU EDGE
Sharp Minds

HTML Import

http://rszaloki.github.io/webcomponents/import/index.html

```
<link rel="import" href="assets.html"
   onload="render()" id="assets">
...
<script>
function render(){
 var content = document.getElementById('assets').import;
 document.body.innerHTML = content;
}
</script>
```

EU EDGE
Sharp Minds

HTML Import

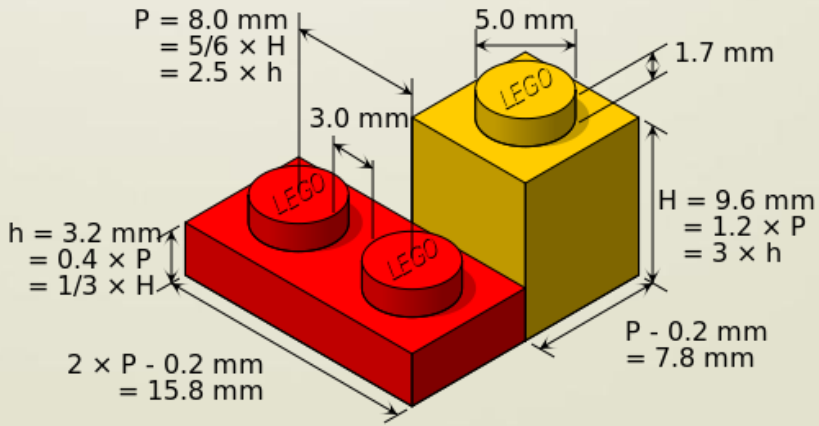
- imports load asynchronous
- same-origin policy!
- javascript in the import
  - global object is the window
  - document is the window.document
  - importDoc = document.currentScript.ownerDocument;
    importDoc.querySelector(‘template’);
- scripts inside the imports are blocks the parsing
- subimports
- the same **import** loads only once

EU EDGE
Sharp Minds

HTML Template

http://rszaloki.github.io/webcomponents/template/index.html

```
<template>
    <div>Everything is AWESOME!</div>
</template>
<script>
…
target.appendChild(
    document.importNode(
        template.content,true));
</script>
```



EU EDGE
Sharp Minds

HTML Template

http://rszaloki.github.io/webcomponents/template/index.html

```
<template>
    <div>Everything is AWESOME!</div>
</template>
<script>
…
target.appendChild(
    document.importNode(
        template.content,true));
</script>
```

EU EDGE
Sharp Minds

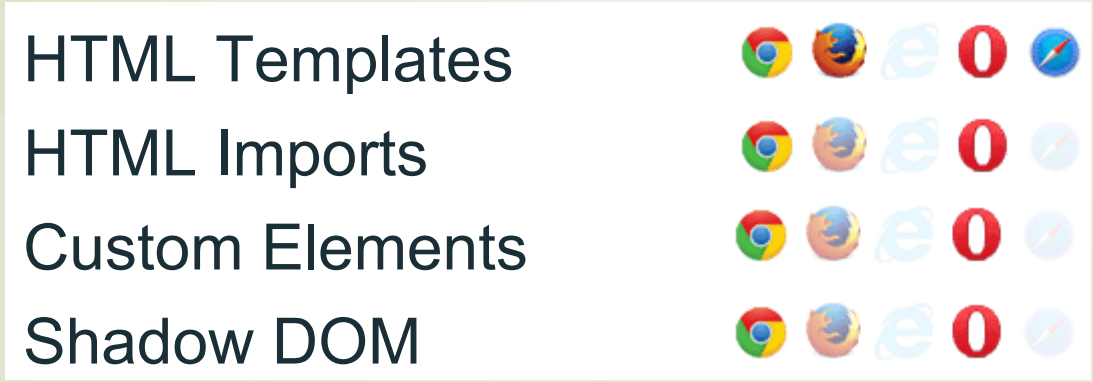
Webcomponents

- encapsulation - Shadow DOM
- reusability - Custom Elements, HTML Import
- communication via interfaces - *Attributes, Properties*
- helpers - HTML Template

EU EDGE
Sharp Minds

Can I use it?

Can I use it?



- too much boilerplate
- no declarative api: <element>
- no easy way to let the users override the default styles: parts, css variables

EU EDGE
Sharp Minds

Polymer Project

Polymer

Platform - polyfills

Web Components

- Shadow DOM
- HTML Imports
- Custom Elements

Other

- Pointer Events
- Pointer Gestures
- Web Animations

DOM

- URL
- WeakMap
- Mutation Observers

EU EDGE
Sharp Minds

Polymer

Platform - polyfills

	Polyfill						
	Chrome Android	Chrome	Canary	Firefox	IE 10+	Safari 6+	Mobile Safari
Template							
MutationObserver					[1]		
HTML Imports							
Custom Elements					[1]		
Shadow DOM							
Object.observe()							
Pointer Events					[2]		
Pointer Gestures							
Web Animations							
Platform							

EU EDGE
Sharp Minds

Polymer

create elements

- declarative syntax
- dynamic templates
- two way data binding
- property observation

EU EDGE
Sharp Minds

Polymer

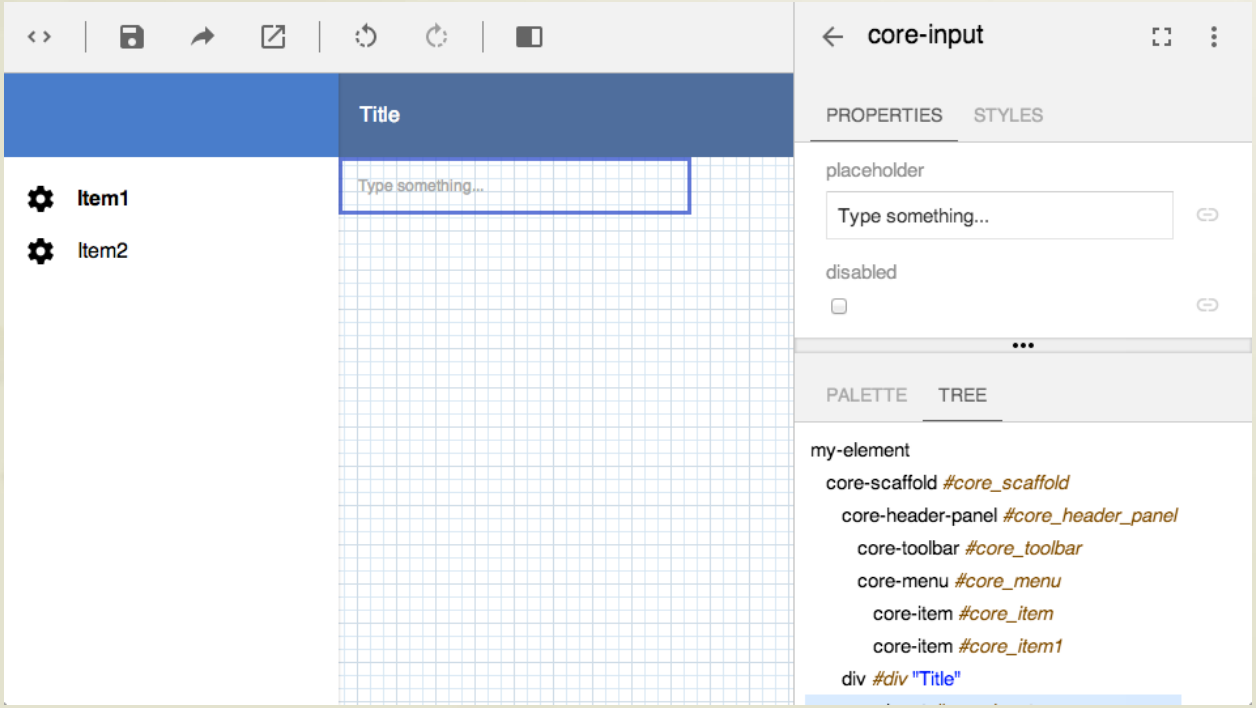
use elements

- widget library
- UI and non-UI elements
- core elements + labs elements

EU EDGE
Sharp Minds

Polymer

tools



Vulcanize: concatenate the imported elements

designer: http://www.polymer-project.org/tools/designer/

EU EDGE
Sharp Minds

Polymer

Thanks! Questions?

robert.szaloki@euedge.com / @rszaloki

http://rszaloki.github.io/webcomponents/polymer-demo/lego.html

https://github.com/rszaloki/webcomponents

https://gist.github.com/ebidel/6314025

EU EDGE
Sharp Minds