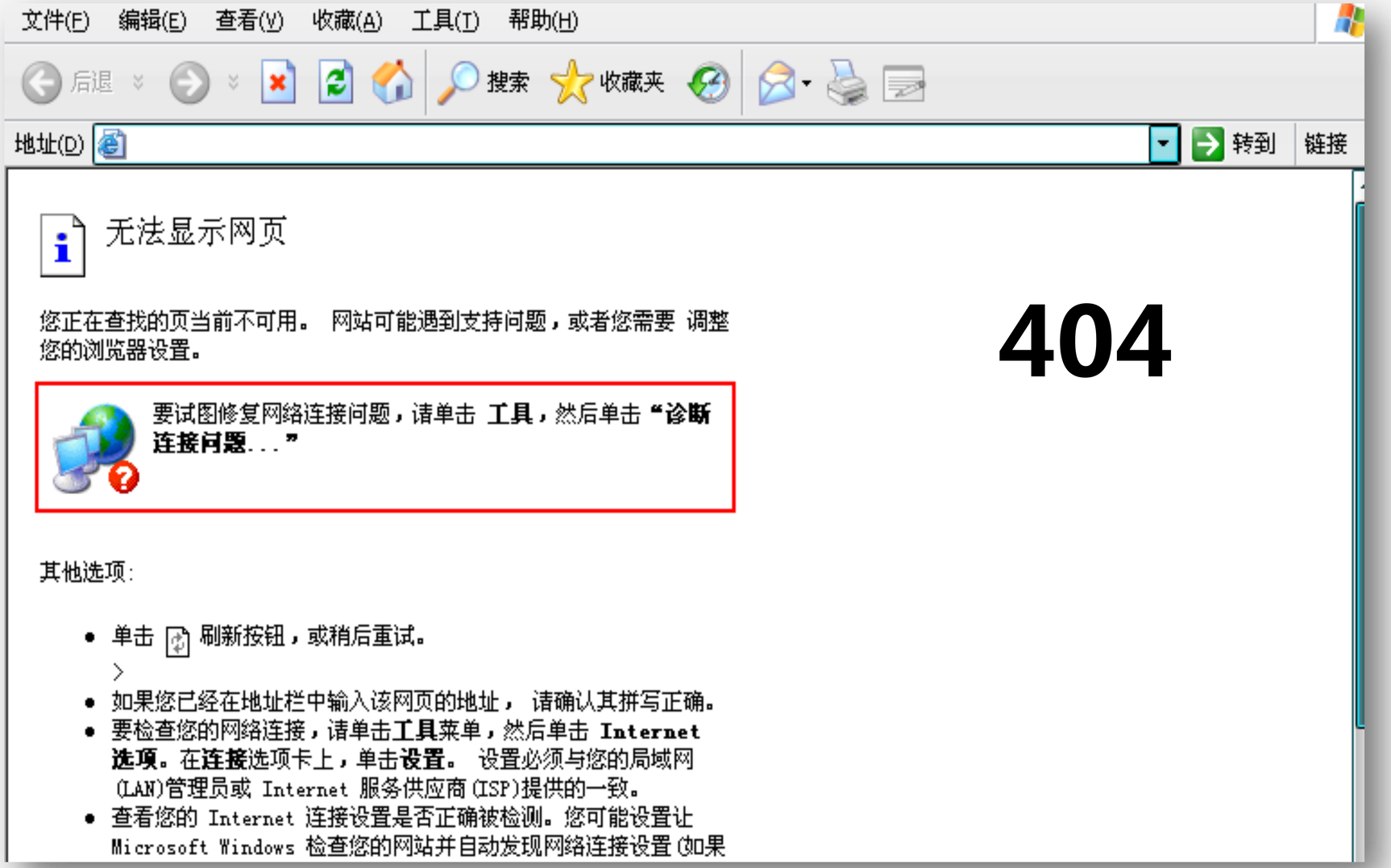
# <Web Components />

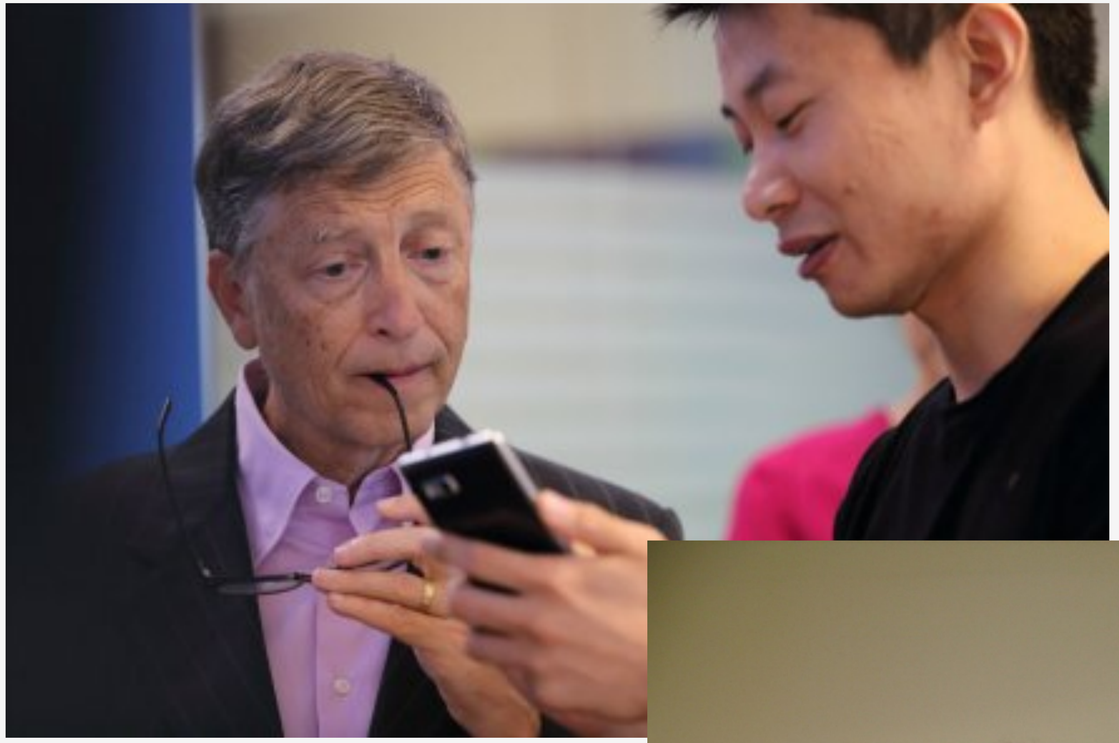
<!– Web 前端开发的未来趋势 --->

陈本峰 2015 - 3

云适配 Allmobilize



# 关于我



# 前端框架

Bootstrap

jQuery
user interface

Semantic UI
UI is the vocabulary of the web.
Semantic empowers designers and developers by creating a language for sharing UI.
VIEW UI
DOWNLOAD

Foundation
ZURB

uikit

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner">
    <div class="item active">
      <img src="..." alt="...">
      <div class="carousel-caption">
        ...
      </div>
    </div>
    ...
  </div>

  <!-- Controls -->
  <a class="left carousel-control" href="#carousel-example-generic" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left"></span>
  </a>
  <a class="right carousel-control" href="#carousel-example-generic" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right"></span>
  </a>
</div>
```

**Bootstrap**

**轮播图组件的引用方式**

# 如果可以这样...

```
<my-slider>
        <img src= “1.jpg”  />
        <img src= “2.jpg”  />
</my-slider>
```

# Web Components

- Template
- Custom Element
- Shadow DOM
- HTML Imports

# 目前的做法

```
<script id="indexTemplate" type="text/template">
        <h1>Inside a template </h1>
        <img src="/image/xyz.png"/>
</script>
```

# 使用<template>元素

```
<template id="indexTemplate”  >
        <h1>Inside a template</h1>
        <img src="/image/xyz.png"/>
</template>

var template = document.getElementById('indexTemplate');
holder.appendChild(template.content.cloneNode(true));
```

# <template>元素特点

- 内容被解析，但是不显示
- 图片资源不会被下载
- 脚本资源不会被下载或者执行

... 直到它被使用

# Custom Element

<my-slider>

</my-slider>

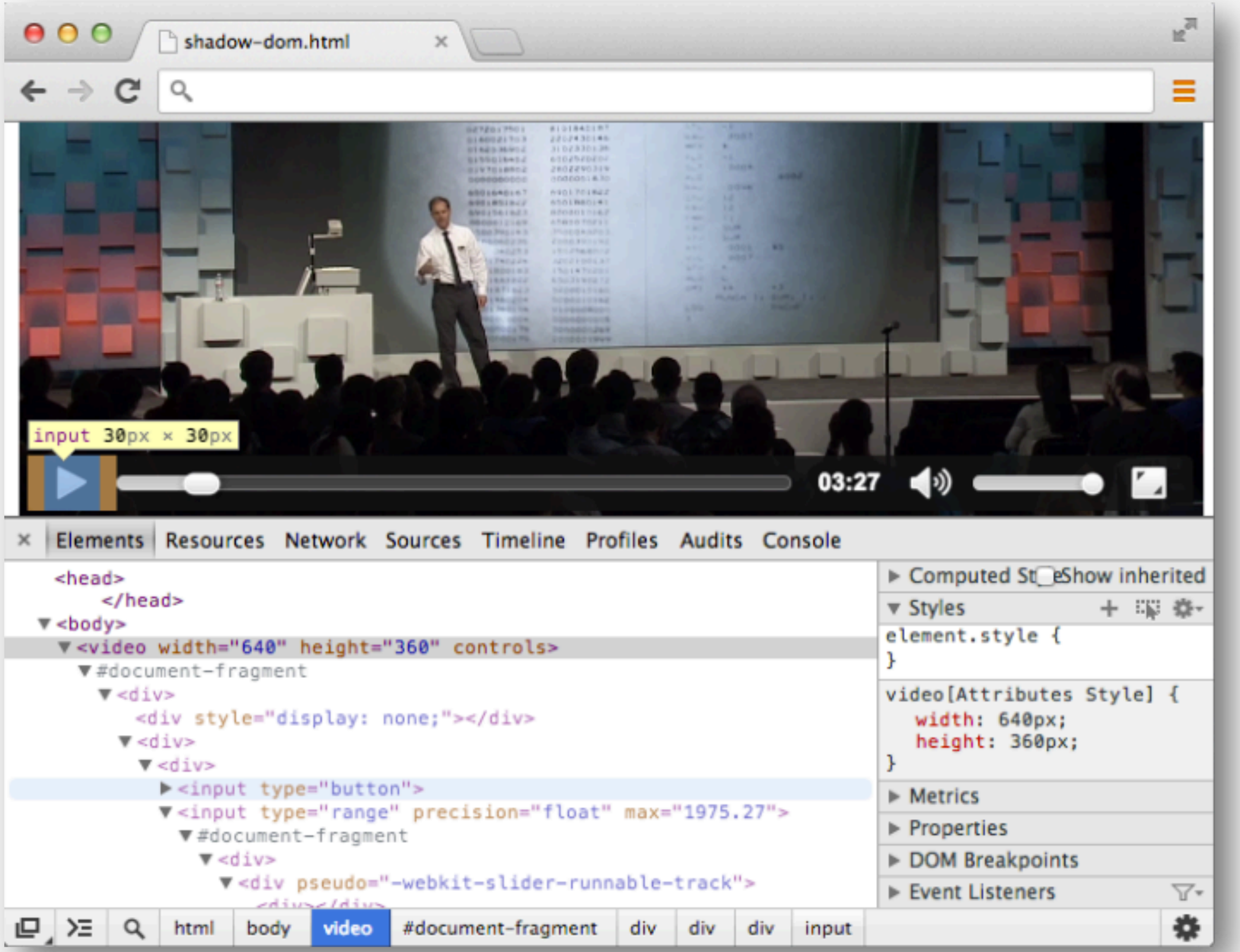
Custom Element

# 利用Custom Element创建<my-slider>元素

```
<template>
  <!– 关于my slider的所有代码 -->
</template>
<script>
var tmpl = document.querySelector('template');
var ImgSliderProto = Object.create(HTMLElement.prototype);
ImgSliderProto.createdCallback = function() {
    var root = this.createShadowRoot();
    root.appendChild(document.importNode(tmpl.content, true));
  };
 var ImgSlider = document.registerElement(‘  my-slider  ‘, { prototype: ImgSliderProto});
</script>
```

PLAYER 1
0
HI SCORE
20000
PLAYER 2
0
影の伝説
THE
LEGEND
OF
KAGE
1 PLAYER
2 PLAYERS
© TAITO CORP. 1986
B
A
WWW.NDUOA.COM N多网

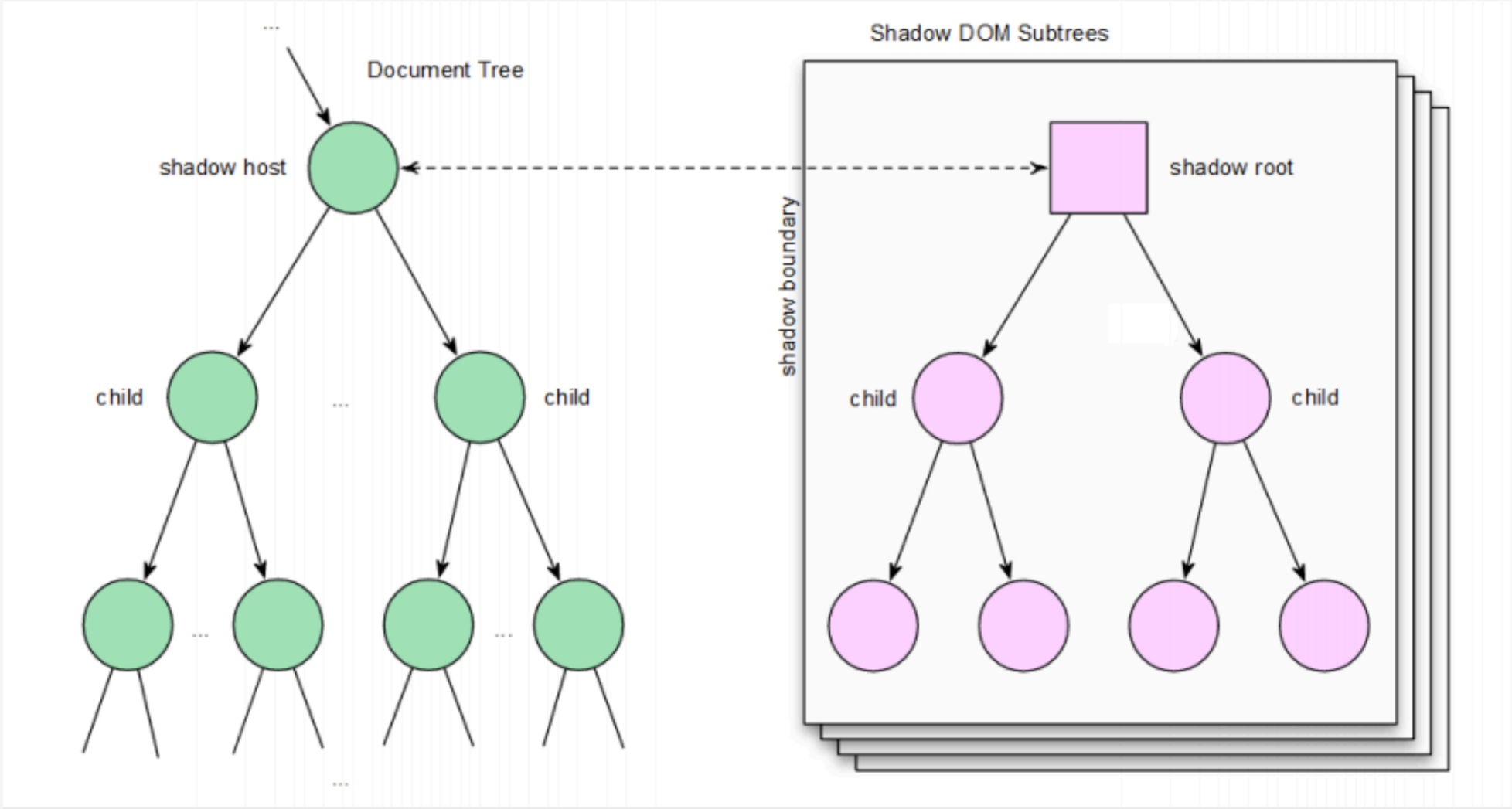
# Shadow DOM


shadow-dom.html
input 30px × 30px
03:27
Elements
Resources
Network
Sources
Timeline
Profiles
Audits
Console
<head>
</head>
<body>
<video width="640" height="360" controls>
#document-fragment

<input type="button">
<input type="range" precision="float" max="1975.27">
#document-fragment

Computed St eShow inherited
Styles
element.style {
}
video[Attributes Style] {
width: 640px;
height: 360px;
}
Metrics
Properties
DOM Breakpoints
Event Listeners
html
body
video
#document-fragment
div
div
div
input

# Shadow DOM


Document Tree
shadow host
child
child
Shadow DOM Subtrees
shadow root
shadow boundary
child
child

# Shadow DOM 特点

- 挂载在某一个元素下面，但是不在DOM里面显示
- 完全独立当前文档，不受当前文档样式的影响

# HTML Imports

```
<link rel="import" href=”my-slider.html">
```

# 完整代码示例

myslider.html 代码

```
<template id="myslider">
    <!-- slider 的完全代码 -->
</template>
<script>
    var sliderProto = Object.create(HTMLElement.prototype);
    sliderProto.createdCallback = function() {
    var template = document.querySelector('#myslider');
    var shadowRoot = this.createShadowRoot();
    var clone = template.content.cloneNode(true);
        shadowRoot.appendChild(clone);
    };
    document.register('myslider', {prototype: sliderProto});
</script>
```

index.html 代码

```
<link rel="import" href=myslider.html">
<myslider></myslider>
```

# 浏览器支持

Templates

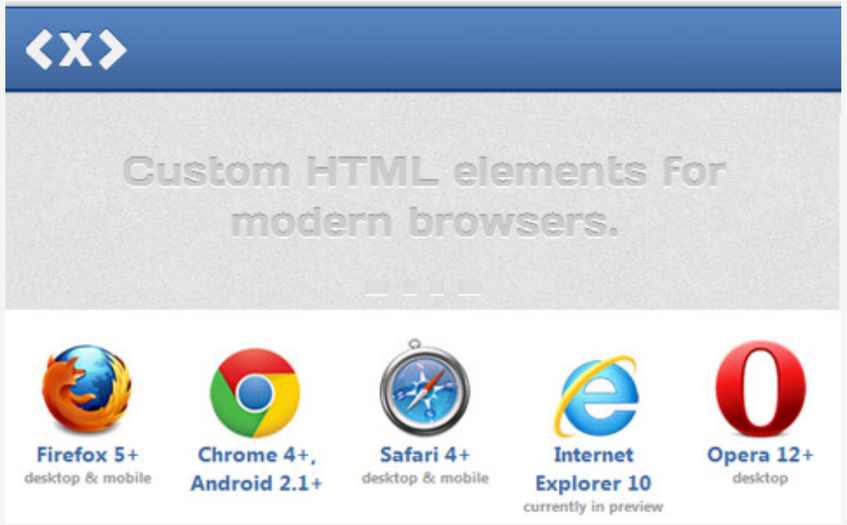
Custom Elements

Shadow DOM

HTML Imports

# Web Components 其他实现

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




http://x-tags.org/

# Web Components 其他实现

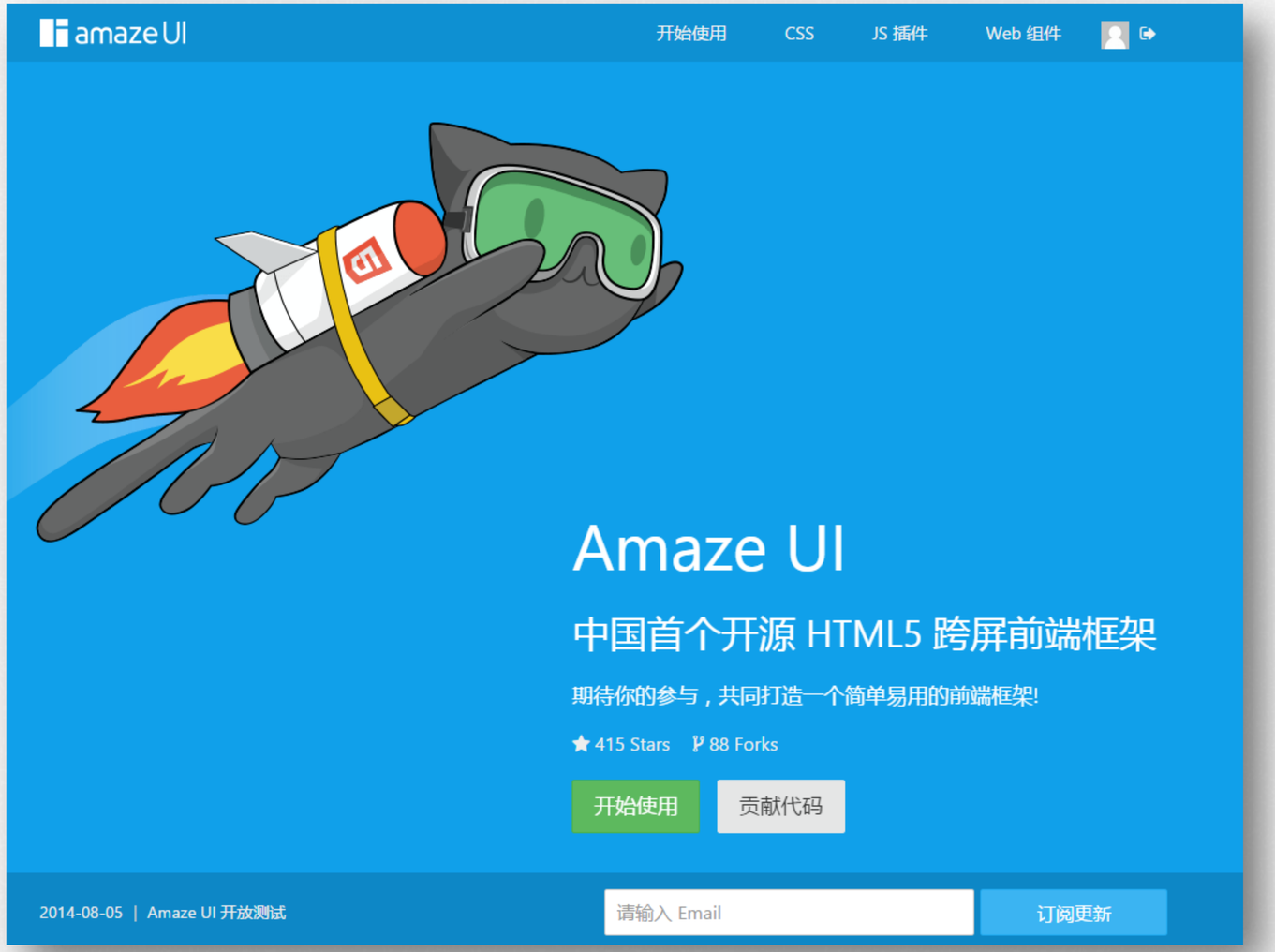
- 一切皆组件！
- 高性能的Virtual DOM

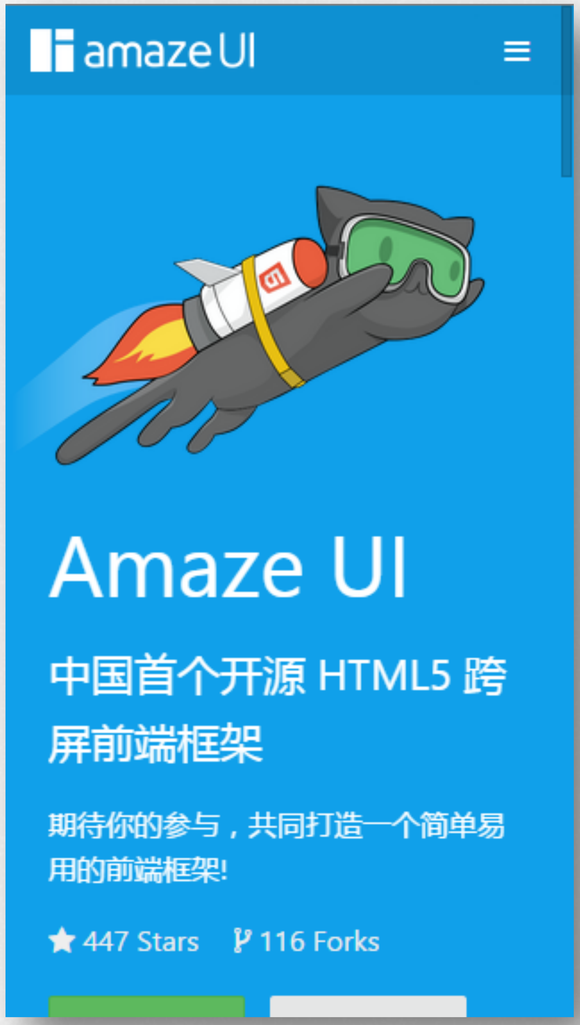
```
var Timer = React.createClass({
  getInitialState: function() {
    return {secondsElapsed: 0};
  },
  tick: function() {
    this.setState({secondsElapsed: this.state.secondsElapsed + 1});
  },
  componentDidMount: function() {
    this.interval = setInterval(this.tick, 1000);
  },
  componentWillUnmount: function() {
    clearInterval(this.interval);
  },
  render: function() {
    return (
      <div>Seconds Elapsed: {this.state.secondsElapsed}</div>
    );
  }
});

React.render(<Timer />, mountNode);
```

Amaze UI

云适配 allmobilize

# Amaze UI –最懂中国程序猿的开源HTML5前端框架



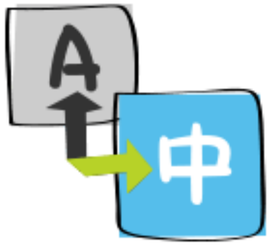
# Amaze UI

## 为移动而生

Amaze UI 采用业内先进的 Mobile first 理念，从小屏逐步扩展到大屏，最终实现所有屏幕适配，适应移动互联潮流。

## 组件丰富，模块化

Amaze UI 含近 20 个 CSS 组件、10 个 JS 组件，更有 17 款包含近 60 个主题的 Web 组件，可快速构建界面出色、体验优秀的跨屏页面，大幅度提升你的开发效率。




## 本地化支持

相比国外的前端框架，Amaze UI 专注解决中文排版优化问题，根据操作系统调整字体，实现最佳中文排版效果；针对国内主流浏览器及 App 内置浏览器提供更好的兼容性支持，为你节省大量兼容性调试时间。

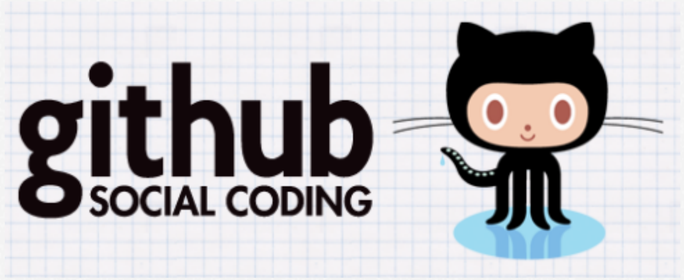
## 轻量级，高性能

Amaze UI 非常注重性能，基于轻量的 Zepto.js 开发，并使用 CSS3 来做动画交互，平滑、高效，更适合移动设备，让你的 Web 应用可以高速载入。

云适配 allmobilize

“妹子UI”最懂中国程序猿的开源HTML5前端框架

- 6个月> **2600** stars
- 超过20万下载




| 公司 | 项目名称 | Stars | 发布时间 |
|---|---|---|---|
| 百度 | echarts | 6246 | 2013.5 |
| 网易 | pomelo | 4449 | 2012.11 |
| 阿里巴巴 | seajs | 3733 | 2010.12 |
| 阿里巴巴 | tengine | 2643 | 2011.11 |
| **云适配** | **Amaze UI** | **2613** | **2014.8** |
| 阿里巴巴 | fastjson | 2045 | 2011.7 |
| 阿里巴巴 | kissy | 1636 | 2012.7 |
| 豆瓣 | dpark | 1267 | 2011.10 |
| 腾讯 | Mars | 1318 | 2013.7 |
| 腾讯 | AlloyImage | 935 | 2013.9 |

云适配
allmobilize

# One Web, Any Device !

万物互联 无缝体验

# Thanks!

allmobilize

诚邀HTML5爱好者加入云适配！

陈本峰 13552800186 bfchen@yunshipei.com QQ: 85436