

神奇的Shadow DOM

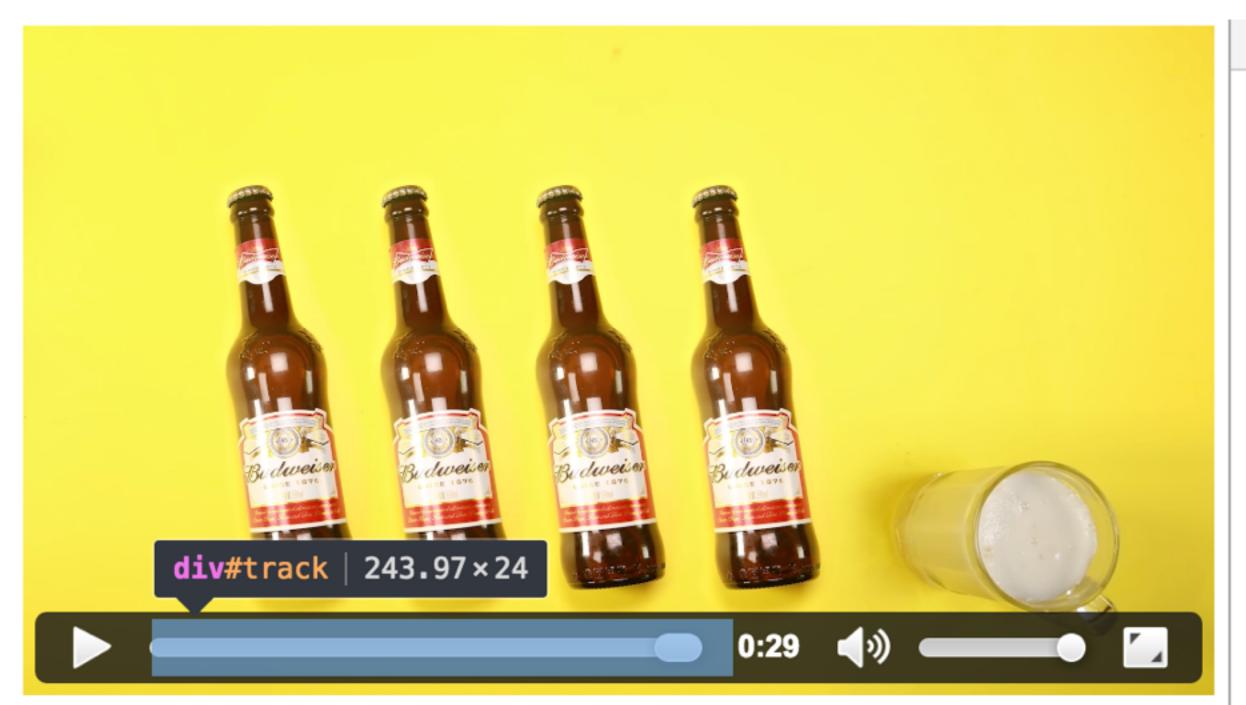
王彩暖



Shadow DOM它允许在文档(document)渲染时插入一棵DOM元素子树,但是这棵子树**不在主DOM树中**。

因此开发者可**利用Shadow DOM 封装**自己的 HTML 标签、CSS 样式和 JavaScript 代码。

Settings	Preferences
Preferences	☐ Disable JavaScript
Workspace	
Blackboxing	Appearance
Devices	Don't show Chrome Data Saver warning
Throttling	Theme: Default ▼
Shortcuts	Panel layout: auto ▼
	Enable Cmd + 1-9 shortcut to switch pa
	Disable paused state overlay
	Elements
	Color format: As authored
	Show user agent shadow DOM
	✓ Word wrap
	Reveal DOM node on hover.
	☐ Show rulers

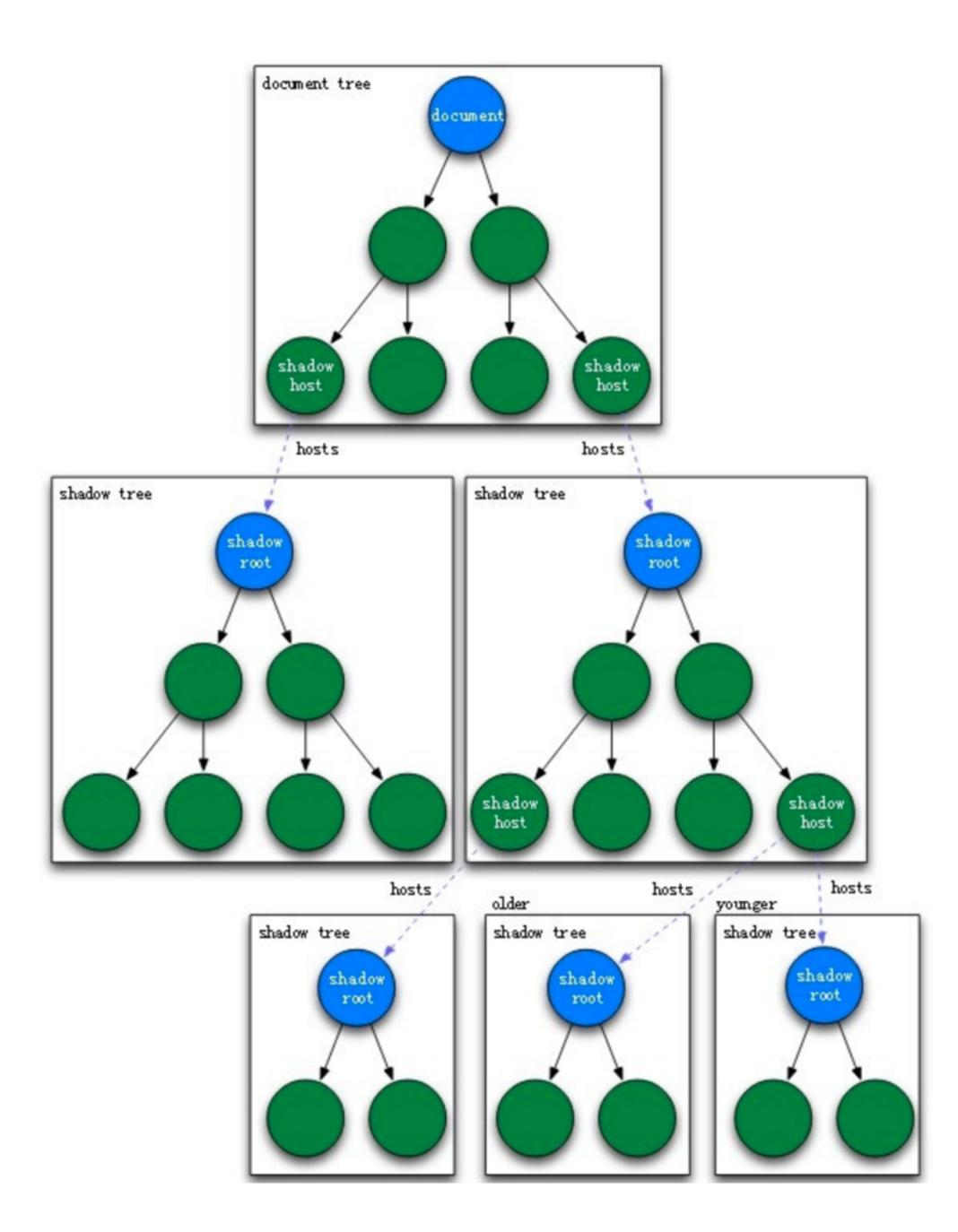


```
Elements
                 Console
                          Sources
                                  Network Timeline Pr
IICau/...\/ IICau/
▼ <body>
 ▼ <video controls autoplay name="media" width=
 "500">
   ▼ #shadow-root (user-agent)
     ▼ <div pseudo="-webkit-media-controls">
      ► <div pseudo="-webkit-media-controls-overlay-</p>
      enclosure">...</div>
      ▼ <div pseudo="-webkit-media-controls-
      enclosure">
        ▼ <div pseudo="-webkit-media-controls-panel"</p>
        style="opacity: 1;">
          ▶ <input type="button" pseudo="-webkit-
          media-controls-play-button">...</input>
          ▼ <input type="range" step="any" pseudo="-
          webkit-media-controls-timeline" max=
          "29.8027">
            ▼ #shadow-root (user-agent)
             ▼ <div>
               ▼ <div pseudo="-webkit-slider-
               runnable-track" id="track"> == $0
                  <div id="thumb"></div>
                </div>
               </div>
           </input>
           <div pseudo="-webkit-media-controls-
           current-time-display" style>0:29</div>
           <div pseudo="-webkit-media-controls-
           time-remaining-display" style="display:
           none;">0:29</div>
```

```
K
         Elements Console Sources
                                   Network Timeline Pr
- \IIcau/...\/ IIcau/
▼ <body>
  ▼ <video controls autoplay name="media" width=
  <u>"500"></u>
    ▼ #shadow-root (user-agent)
     ▼ <div pseudo="-webkit-media-controls">
       ► <div pseudo="-webkit-media-controls-overlay-</p>
       enclosure">...</div>
       ▼ <div pseudo="-webkit-media-controls-
       enclosure">
         ▼ <div pseudo="-webkit-media-controls-panel"
         style="opacity: 1;">
           ▶ <input type="button" pseudo="-webkit-
          media-controls-play-button">...</input>
           ▼ <input type="range" step="any" pseudo="-
          webkit-media-controls-timeline" max=
           "29.8027">
            ▼#shadow-root (user-agent)
              ▼ <div>
                ▼ <div pseudo="-webkit-slider-
                runnable-track" id="track"> == $0
                   <div id="thumb"></div>
                 </div>
               </div>
            </input>
            <div pseudo="-webkit-media-controls-
            current-time-display" style>0:29</div>
            <div pseudo="-webkit-media-controls-
            time-remaining-display" style="display:
            none;">0:29</div>
```

影子宿主: video

影子根: #shadow-root



影子树(shadow trees): #shadow-root可以嵌套 形成的节点树

使用**createShadowRoot()**来创建Shadow DOM, 并赋值给一个变量, 然后添加元素给变量即可。

```
<body>
 <div class="shadowhost">Hello, world!</div>
 <script>
   // 影子宿主 (shadow host)
   var shadowHost = document.querySelector('.shadowhost');
   // 创建影子根(shadow root)
   var shadowRoot = shadowHost.createShadowRoot();
   // 影子根作为影子树的第一个节点,其他的节点比如p节点都是它的子节点。
   shadowRoot.innerHTML = '夏天夏天悄悄过去留下小秘密! </
p>';
</script> </body>
```

夏天夏天悄悄过去留下小秘密!

影子边界:

- * 影子边界保证主 DOM写的 CSS 选择器和 JavaScript 代码都不会影响到Shadow DOM。
- * 保护主文档不受 shadow DOM 样式的侵袭。

需要完成此项任务,需要两个利器: <content>和<template>。

<content>

<content>:

通过 <content> 标签把 来自主文档 并添加到 shadow DOM 的内容 被称为分布节点。

<content>的select属性:

告诉<content>标签有选择性的插入内容。 select 属性**使用 CSS 选择器来选取想要展示的内容**,选择器包 括类选择器、元素选择器等。

<content select=".shadowhost_content1"></content>

<template>

```
<script type="text/template">
// ...
</script>
```

```
<script type="text/template">
// ...
</script>
```

模板HTML做法通常是在<script>中嵌入模板HTML,让内部的HTML标签按照字符串处理的,从而使得内容不显示。

<template>的好处:

- * 在使用前不会被渲染,不会执行加载等操作。
- *实现隐藏标签内容。
- *位置任意性,可以在<head>中,也可以在<body>或者
- <frameset>中。

关于样式

:host 在Shadow DOM中给宿主添加样式

::shadow 主文档给Shadow DOM添加一些样式

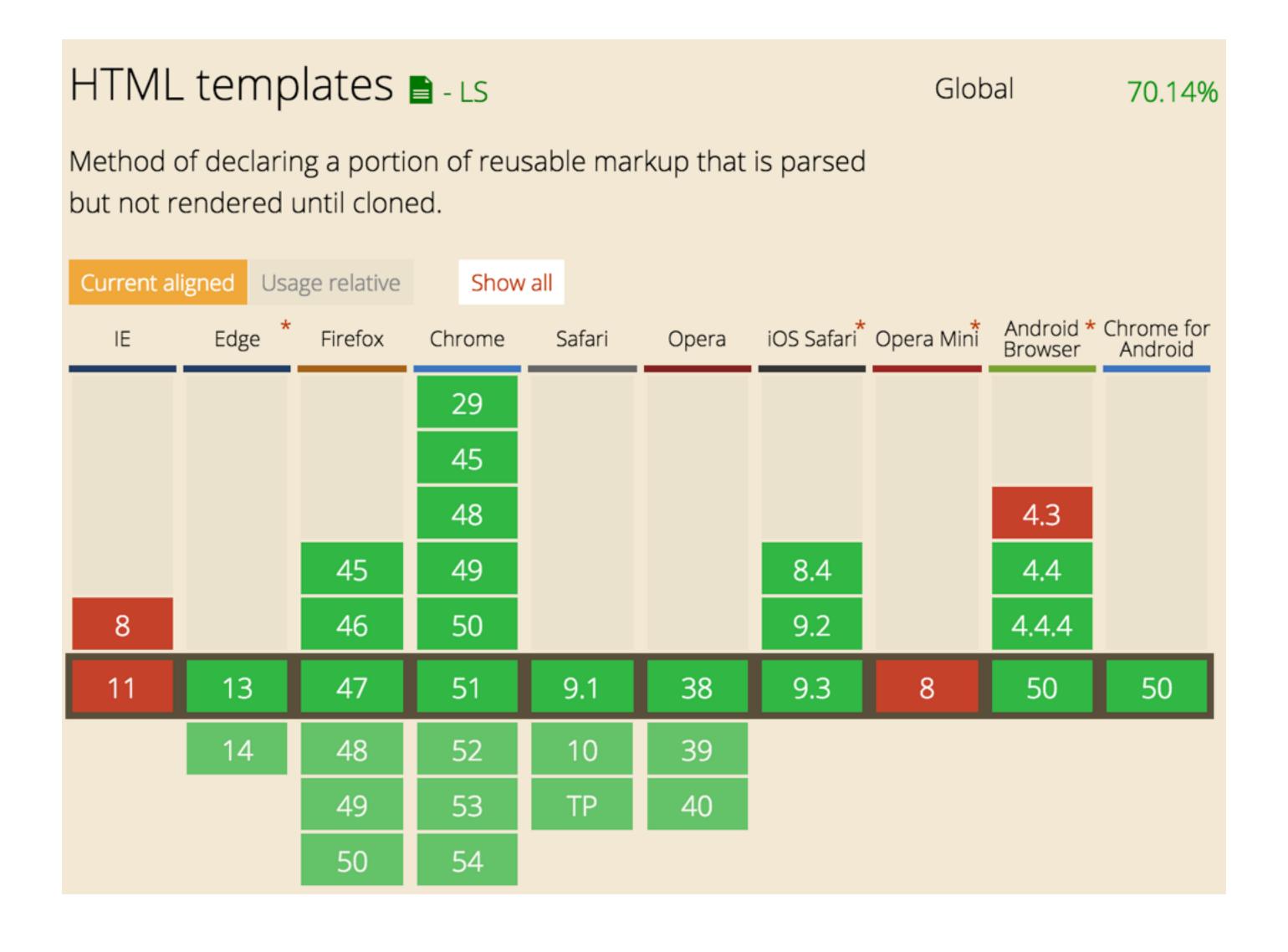
/deep/ 主文档给Shadow DOM添加一些样式

::content 定义分布节点的样式

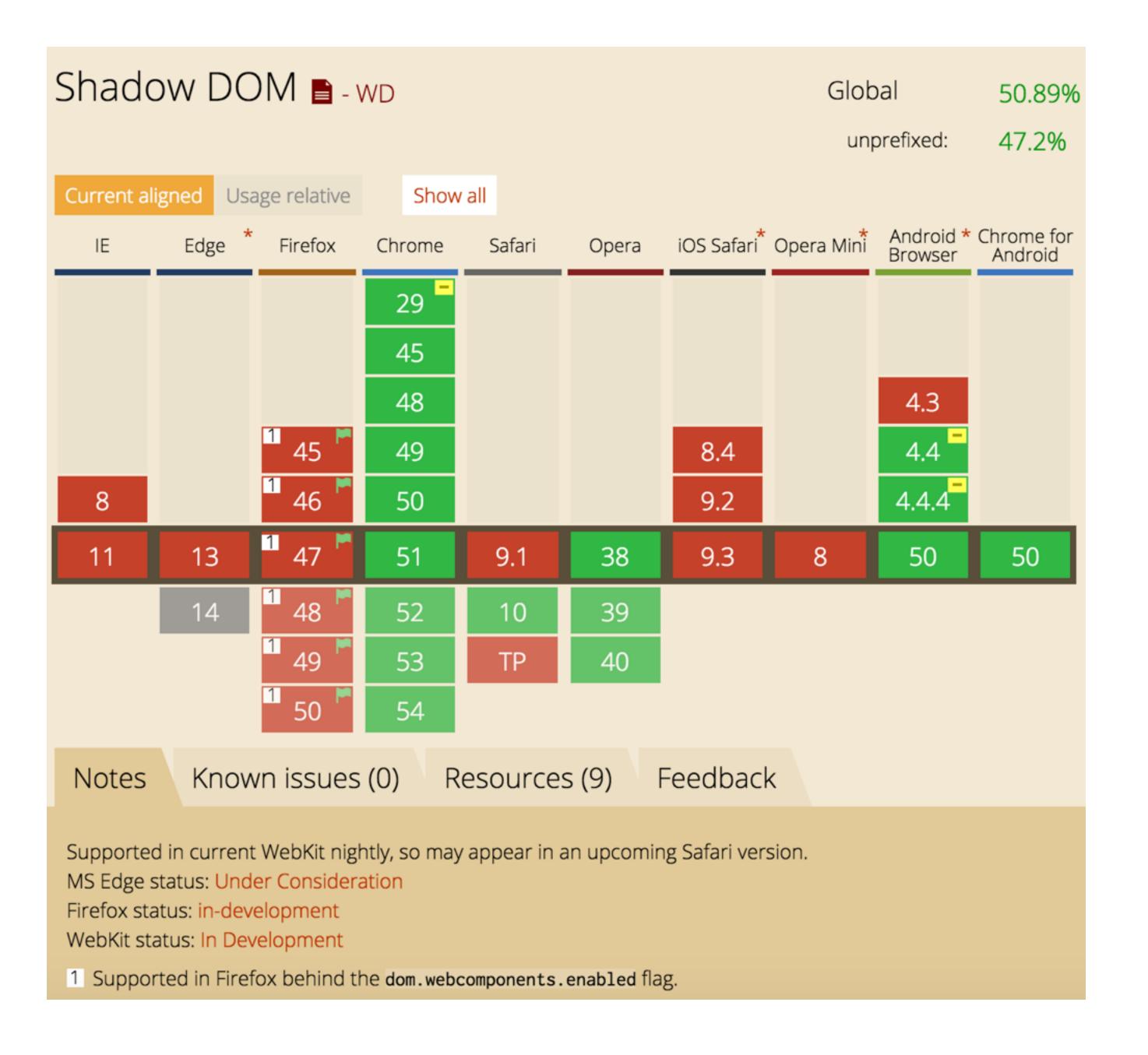
关于JavaScript

- * 原来绑定在 shadow DOM 节点中的事件被重定向了, 看起来像绑定在影子宿主上一样。
- * 事件abort、error、select、change、load、reset、resize、scroll、selectstart不会进行重定向而是被干掉。

兼容性



兼容性



参考文献

http://html5-demos.appspot.com/static/shadowdom-visualizer/index.html

https://css-tricks.com/modular-future-web-components/

http://www.ituring.com.cn/article/177453

http://www.zhangxinxu.com/wordpress/2014/07/hello-html5-template-tag/



THANKS FOR LISTENING

王彩暖