

# 灵活的 Web 图标

多终端下的图标方案浅析

Ryan @ D.G.Z (GitCafe)

 @ryaneof

## 主要的实现方式

<img>

CSS Sprites

Icon Font

<svg>

Pure CSS Icons

## 需要考虑的点

精确还原

加载速度

兼容性

维护成本

SVG Symbol

&

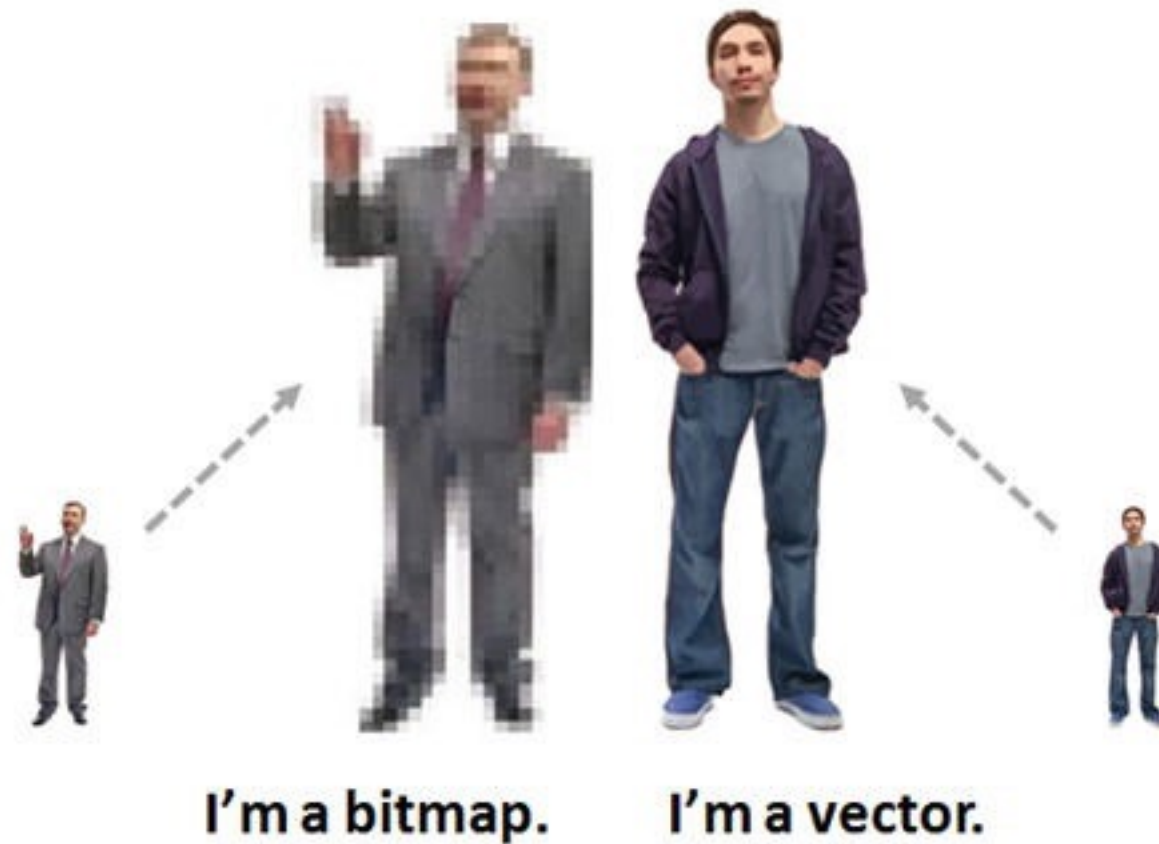
CSS Sprites Fallback

Why SVG

位图 vs 矢量图



## 万恶的视网膜屏幕



“一夜之间，图片全糊掉了，只有矢量图坚挺着”



# 位图作为图标缺陷

图片体积无法控制

2x, 3x 尺寸只是一个大概的估算

不能完美适应所有的像素密度

放大：颜色近似选取，图片模糊

缩小：减少像素取样，资源浪费

设计师：图标本来就应该 是矢量的

矢量图形

Pure CSS Icons

Icon Font

SVG



## Pure CSS Icons

精确还原 √

加载速度 √

兼容性 X

维护成本 X

# Icon Font

精确还原 √

加载速度 √

兼容性 √

维护成本 X

似乎是一个靠谱的方法

# Icon Font 定位问题

## 会受到浏览器对字体渲染的影响

line-height, vertical-align, letter-spacing, word-spacing, text-rendering

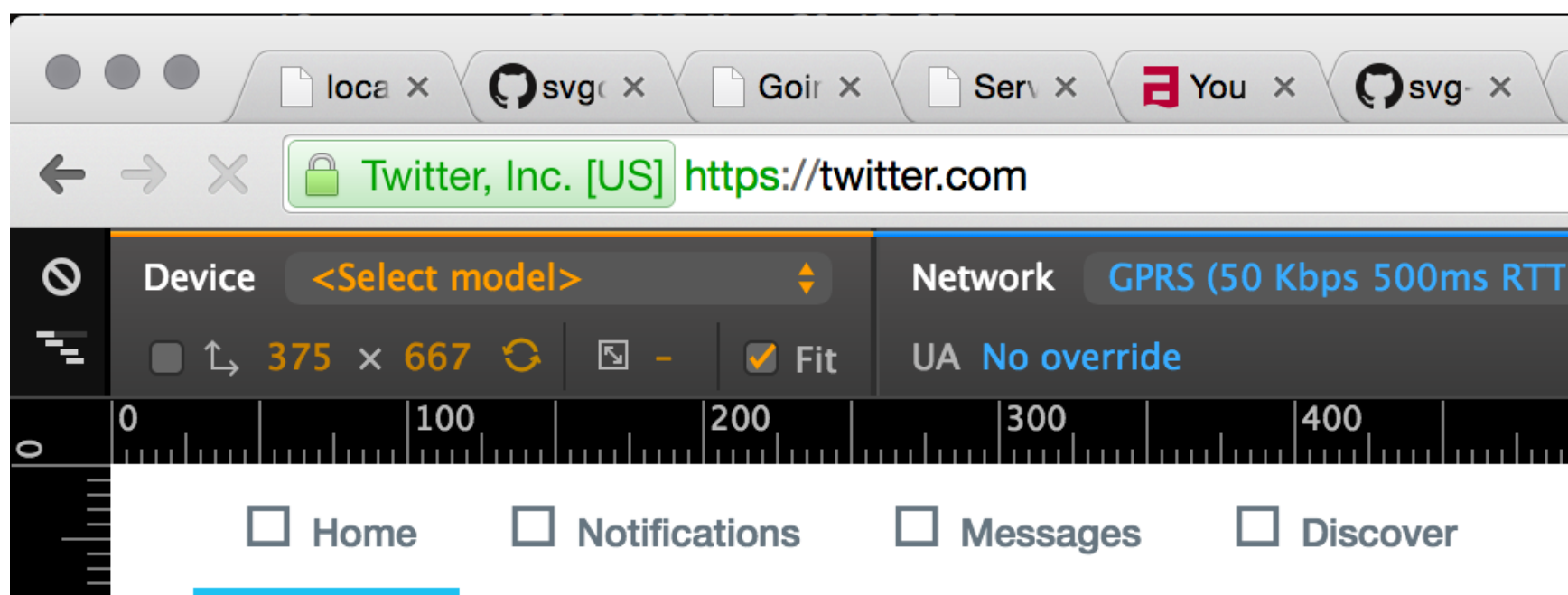
边缘不够锋利，伪元素盒模型不精确，各浏览器渲染的字重不同

## 会受到字符本身设计的影响

留白，字符本身的高度，宽度

# Icon Font 加载问题

会受到网络状况影响，显示 Fallback 字体



# Icon Font 其他问题

Unicode 字符冲突

额外的工作：字体编辑

颜色单一，交互效果有限

不完全的浏览器兼容性

困难的 fallback 检测



## Icon Font



# Scalable Vector Graphics

可缩放的

矢量

图形

# SVG

精确还原 √

加载速度 √

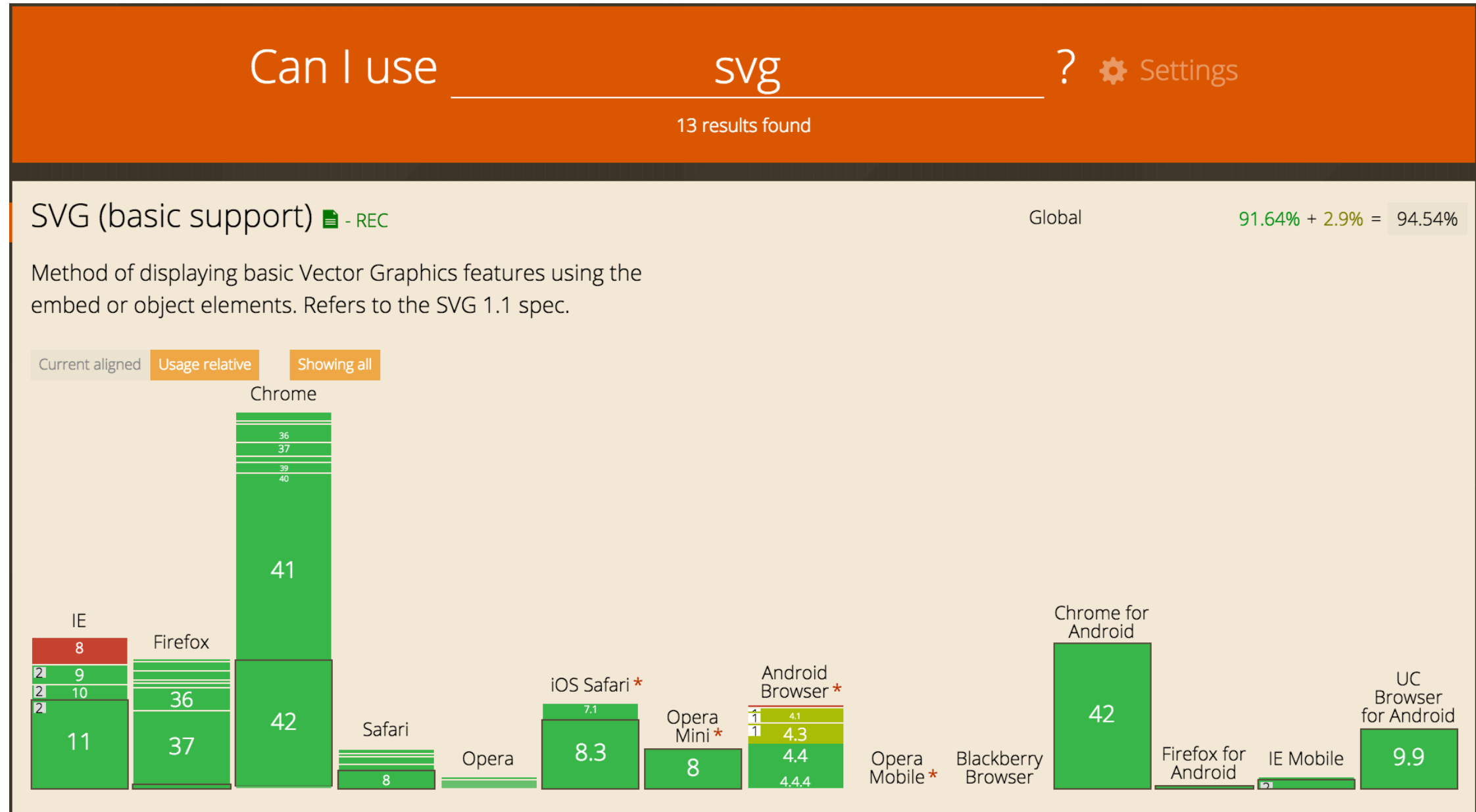
兼容性 X

维护成本 √

## 更易于编辑

```
icon-svg-symbol.svg ×
<?xml version="1.0" encoding="utf-8"?>
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/
xlink">
<symbol viewBox="0 0 26 26" id="icon-svg-symbol--date">
  <path fill="#fff" fill-rule="evenodd" d="M13 2.01c6.07 0 11 4.93 11 11S19.
07 24 13 24 2.01 19.08 2.01 13.01s4.92-11 10.99-11m0-2c-7.17 0-13 5.83-13
13s5.83 13 13 13 13-5.83 13-13-5.83-13-13-13zm-.76
14.7c-.18-.18-.29-.43-.3-.71.01-.55.45-1 1.01-1h7.48c.55 0 1 .45 1 1s-.45
1-1 1h-7.49c-.27 0-.52-.11-.7-.29zm.06
0c-.18-.18-.29-.43-.29-.71V6.34c0-.55.44-1 1-1 .55 0 1 .45 1 1V14c0 .55-.45
1-1 1-.28 0-.53-.11-.71-.29z"/>
</symbol>
<symbol viewBox="0 0 27 24" id="icon-svg-symbol--like">
```

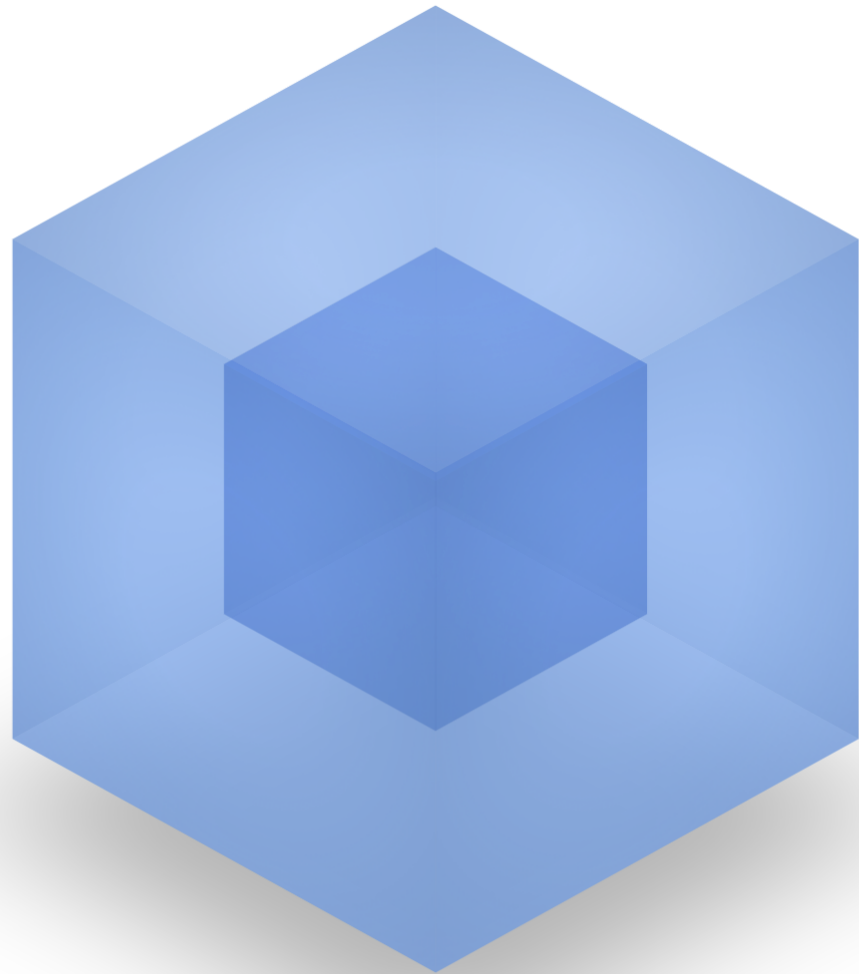
# 兼容性



## 容易的 Fallback 检测

```
1 function supportsSvg() {  
2   var div = document.createElement('div');  
3   div.innerHTML = '<svg/>';  
4   return (div.firstChild && div.firstChild.namespaceURI) == 'http://www.w3.org/2000/svg';  
5 };
```

# 更多的交互可能



SVG Animation

SVG + CSS

SVG DOM + JavaScript

<http://webpack.github.io>

更... 灵活



## 通过图片元素引入

```

```

# 作为背景图片引入（支持 CSS Sprites）

## HTML

```
<a href="/" class="logo">  
  Kiwi Corp  
</a>
```

## CSS

```
.logo {  
  display: block;  
  text-indent: -9999px;  
  width: 100px;  
  height: 82px;  
  background: url(kiwi.svg);  
  background-size: 100px 82px;  
}
```

[buystrip\\_call.svg](#));



# 内联 SVG 元素

```
HTML CSS Result Edit on CODEPEN
<h1>Plain Text Fallback for Inline SVG</h1>
<svg viewBox="-20 -20 40 40">
  <title>SVG OK</title>
  <!--Text Fallback-->
  I'm sorry, your browser does not support
  <circle fill="limegreen" r="19" />
  <path stroke="forestgreen" fill="none" stroke-width="6"
    d="M-12,3 L-3,10 11,-12" />
  <text dy="0.35em" text-anchor="middle" font-weight="bold"
    font-size="18px" font-family="sans-serif"
    fill="indigo">SVG</text>
  <!--Fallback with links-->
  Please upgrade to a <a href="http://browsehappy.com/?
  locale=en">modern browser</a>.
</svg>
```

## SVG Symbol

“概念上就是和 CSS Sprites 差不多的东西”

# .svg

```
icon-svg-symbol.svg ×
icon-svg-symbol.svg
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">

  <symbol viewBox="0 0 32 32" id="svg-symbol-check">
    <title>Check</title>
    <desc>Created with Sketch.</desc>
    <g fill="none" fill-rule="evenodd">
      <path d="M16 1c8.28 0 15 6.72 15 15 0 8.28-6.72 15-15 15-8.28 0-15-6.72-15-15"
        stroke-width="2"/>
    </g>
  </symbol>

  <symbol viewBox="0 0 32 32" id="svg-symbol-checked">
    <title>Checked</title>
    <desc>Created with Sketch.</desc>
    <g fill="none" fill-rule="evenodd">
      <g transform="translate(1 1)">
        <path d="M15 0c8.28 0 15 6.72 15 15 0 8.28-6.72 15-15 15-8.28 0-15-6.72-15-15"
          stroke="#000" stroke-width="2"/>
        <path d="M14 23l-8-6 2-3 5 4 6-9 3 2-8 12z" fill="#333"/>
      </g>
    </g>
  </symbol>
```

.html

```
50 <svg>
51   <use xlink:href="images/icon-svg-symbol.svg#svg-symbol-home"></use>
52 </svg>
```

with fallback

```
48 <span class="icon-svg icon-svg-home">
49   <svg>
50     <a class="icon-svg-fallback"></a>
51     <use xlink:href="images/icon-svg-symbol.svg#svg-symbol-home"></use>
52   </svg>
53 </span>
```

## .css with fallback

```
7  .icon-svg {
8      display: inline-block;
9  }
10
11  .icon-svg-home svg {
12      width: 30px;
13      height: 32px;
14  }
15
16  .icon-svg-home a {
17      display: inline-block;
18      width: 32px;
19      height: 32px;
20      background-image: url(images/home.png);
21      background-size: 32px 32px;
22  }
23
```

span > svg > use

Developer Tools -

Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console »

PE html>

```
pt type="text/javascript">window["_gaUserPrefs"] = { ioo :  
ion() { return true; } }</script>  
>...</head>  
>  
✓ class="pages-container">  
div class="header">  
  <h1 class="title fl">...</h1>  
  <div class="nav-links fr" data-current="home">  
    ▼ <a data-menu="home" href="/" class="active">  
      ▼ <span class="icon-svg icon-svg-home">  
        ▼ <svg>  
          <use xlink:href="https://  
f8ca758cc3344ceb5eec69257ff499d3.svg#exported--  
home"></use>  
        </svg>  
      </span>  
    </div>  
  </div>  
</span>  
<span>首页</span>
```

span.icon-svg.icon-svg-home 21px × 22px

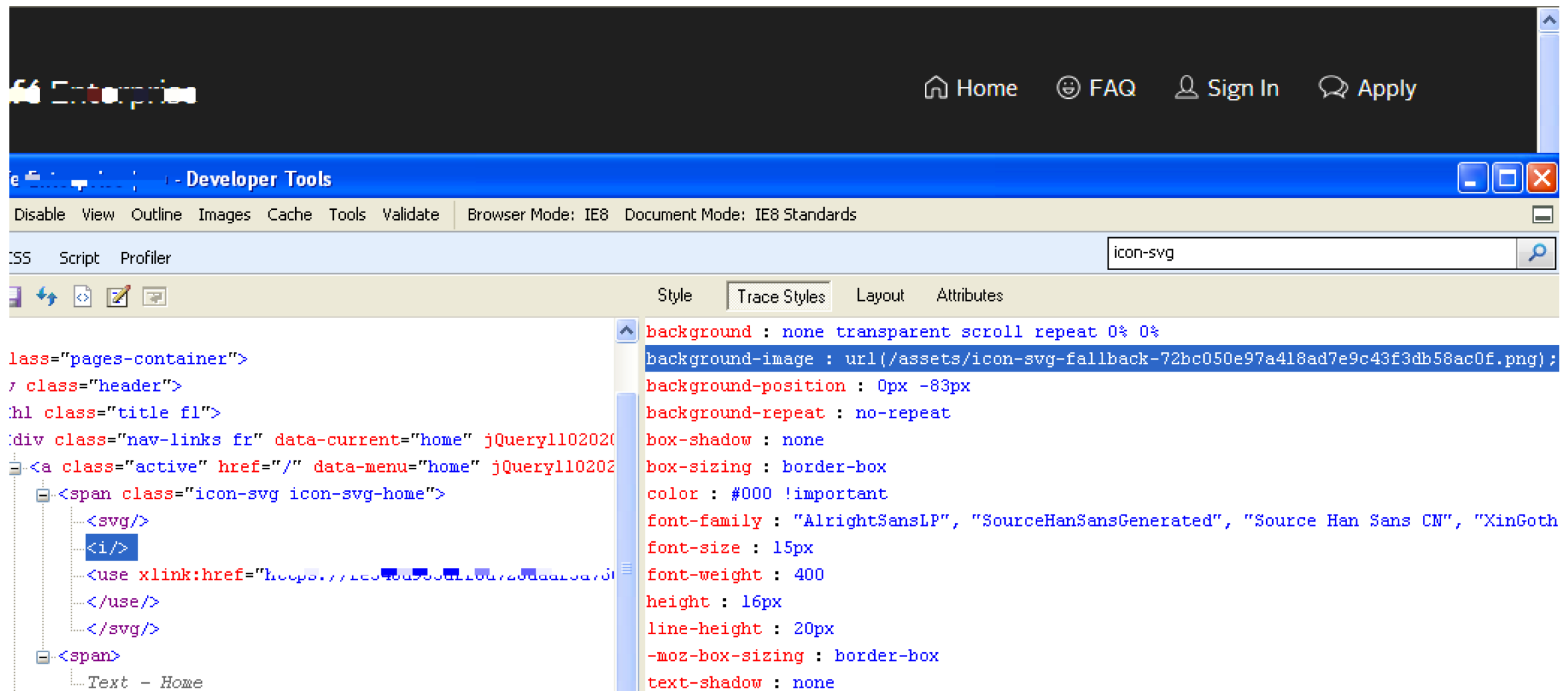
Styles | Computed »

margin -  
border -  
padding -  
6 15 × 22

☐ Show inherited properties  
-webkit-font-smoothing: antialiased;  
box-sizing: border-box;  
color: rgb(255, 255, 255);



span > i



## 更灵活的 Fallback 机制

```
50 <svg>
51   <use xlink:href="images/icon-svg-symbol.svg#svg-symbol-home"></use>
52 </svg>
```

检测 USE 元素的 ownerSVGElement 属性

```
1  (function () {
2    var u = document.getElementsByTagName('use');
3    if (u.length && !(u[0].ownerSVGElement)) {
4      for (var i = 0, n = u.length; i < n; i++) {
5        var use = u[i],
6            el = document.createElement('i');
7        use.parentNode.insertBefore(el, use);
8      }
9    }
10 })();
```

## workflow

设计师：思考怎样透过图标传递情感

设计师：画、画、画、改、改.....

设计师：把 SVG 放在某个目录

工程师：跑个脚本

工程师：复制、粘贴、好了

Everybody's happy.



Q & A

<https://github.com/ryaneof/svg-symbol-demo>

感谢