

基于iconfont、svg sprite 构建公用图标系统

2017/08/04
@maplehongye

起因：

传统的css sprite 技术重复单调的维护图标， 没意思！

移动互联网时代，为了兼容各种高清设备，需额外处理@2x 、@3x图标！



传统的非矢量图标已不能满足日益发展的前端页面

解决方案：

提取复用率比较高的图标，建立矢量图标库

实现方案：

iconfont 或 svg sprite

iconfont如今已经成为了前端必备的工具。

优点：

样式调整非常方便

在多分辨率情况下能够良好的展示和适配

无浏览器兼容问题

缺点：

颜色的填充仅限于单色，无法支持包含多种颜色的图标；

在不同浏览器下展现的效果可能有所不同，在某些浏览器下可能会出现锯齿效果。

需要http请求

iconfont: how to use?

- 视觉同事制作并提供页面图标的svg源文件
- 通过icomoon.io或iconfont.cn等导入svg图标生成项目需要的font文件，生成的font文件放到项目路径下
- 项目css里声明@font-face
- 具体页面使用方法，可参见 iconfont.cn 帮助文档
- 目前已使用并上线的iconfont图标库：<http://page.jd.com/mall/iconfont/daogou/html/daogou-font.html>

iconfont: 使用中细节问题

- 图标大小小于12px时，如何使用iconfont
- 个别双核浏览器，使用webkit极速模式时，图标边缘出现锯齿
- iconfont创建的图标库应该只放复用率高的图标

简单了解下svg的使用方式

svg 在html中的使用方式:

1. svg 作为背景图片使用
2. svg 作为src属性使用
3. 内联svg

内联 svg 直接在文档中包含了所有的XML数据, 并且允许我们通过css 和Javascript来访问它们。如通过fill、stroke、position、opacity和其他各种属性来改变它的外貌和特征。

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" width="24px" height="24px" viewBox="0 0 24 24">  
  <path fill="#E86C60" d="M17,0c-1.9,0-3.7,0.8-5,2.1C10.7,0.8,8.9,0,7,0C3.1,0,0,3.1,0,7c0,6.4,10.9,15.4,11.4,15.8c0.2,0.2,0.4,0.2,0.6,0.2s0.4-0.1,0.6-0.2C13.1,22.4,24,13.4,24,7C24,3.1,20.9,0,17,0z"></path>  
</svg>
```


svg sprite： 两种实现方式

- 实现1：类似css sprite 的原理，把所有svg图标拼在一起，维护复杂，不推荐
- 实现2：symbol + use

svg sprite: how to use?

- <symbol>元素用来对元素进行分组；它不会被直接显示，大概相当于定义一个模板，然后使用<use>元素引用并进行渲染。

```
<svg xmlns="http://www.w3.org/2000/svg">
  <symbol id="icon-aa" viewBox="0 0 72 72"><!-- 路径/图形 --></symbol>
  <symbol id="icon-bb" viewBox="72 0 72 72"><!-- 路径/图形 --></symbol>
</svg>
```

三种引入symbol的方式

1. 页面引入inline svg，inline svg 中存放公用的 symbol

```
<svg><use xlink:href="#icon-aa"/></svg>
```

2. symbol存在于外部引入的svg文件，此时可能存在跨域问题

```
<svg><use xlink:href="sprite.svg#icon-aa"/></svg>
```

3. 通过js动态生成页面所需要的svg symbol，推荐此方式

通常使用一个polyfill：svgxuse.min.js，这是针对IE9+的polyfill；

icomoon打包svg sprite就是使用这个js文件

svg sprite: 注意事项

- 兼容性差，一些浏览器不支持inline SVG特性：

On the browser support front, the danger zones are IE 8 and down, Safari 5 and down, iOS 4.3 and down, and Android 2.3 and down.

- 同时，一些支持inline SVG特性的浏览器，却不支持external svg特性，这使得无法使用浏览器的cache来优化前端性能，例如IE9/10。
- SVG在绘制的性能上比PNG要差



THANK YOU