

## 16 字体图标的使用

更新时间：2019-07-03 11:55:18



“

当你做成功一件事，千万不要等待着享受荣誉，应该再做那些需要的事。

—— 巴斯德

”

图标就是一个用来表达信息的图形，在生活中图标随处可见。我们见着箭头就知道往那边走，见着男女两个小人就知道这是卫生间，见着叹号的地方就知道要注意安全。这些图标会很直接的传递给我们信息，要比文字来的更快。图标也是世界通用语言，如果不懂英语的人去国外看见“Rest Room”可能不知道是什么地方，但如果看见了一男一女两个标志，就肯定能知道这是洗手间，并且还能知道应该往哪边走。

我们在页面的开发中也离不开图标，适当的使用图标会帮助用户快速的获取信息并做出反馈。目前图标被应用在各种应用上，下面的图片为例：





上面两张图，一个是微信的个人中心页，一个是淘宝的搜索结果页。我们随便选了两个页面，可以看到在这两个页面里用红框圈出来的全都使用了图标，可见图标在应用的设计中是十分常见而又重要的。

本来这一章按着顺序应该是先讲布局开发的，但是在布局中也会用到字体图标，所以会把字体图标的讲解提到前面来。下面我们就来看一下在移动端Web的开发中，图标是怎么使用的。

## 图标的分类

在使用图标前，我们先来了解一下图标的分类。在Web应用里，用的最多的就是图片图标和字体图标这两种。下面我们来看一下这两种图标各自的特点。

### 一、图片图标

在早期的Web开发中，使用图片来制作图标是一个共识。当时使用透明的png图片把多个小图标合成在一个大的雪碧图里，几乎是所有应用都在使用的方法。

图片图标有很多优点：

1. 图片图标可以使用任意颜色，任意形状。
2. 制作新图标时的成本比较低。
3. 兼容性良好，哪怕在低版本IE上也可以通过一些方法达到兼容的效果。

图片图标的特点也比较明显的：

1. 图片类型的图标一旦制作出来，图标的颜色、形状和透明度等就都固定了，再想改动的话就要重新制作一个。
2. 如果不对图片图标进行合并，页面上会有很多个图标图片的请求，会影响页面的加载。
3. 如果使用雪碧图对图标进行合并，这个合成后的图片尺寸会比较大，另外如果没有工具的帮助，纯手动的来找图标的位置也是很麻烦的。
4. 图片是非矢量的，如果对图标进行缩放，就会造成图标清晰度的变化，图标的显示效果会受到影响。

## 二、字体图标

后来，字体图标就慢慢进入了开发者的视线。实际上早在IE6的时代，很多浏览器就有了自定义字体的能力。但是由于每个浏览器支持的字体格式不同，也没有像样的字体图标库出现，所以当时字体图标并没有被广泛应用。后来随着一些字体图标库的出现，字体图标就开始被广泛的使用了。

字体图标的优点恰好能弥补图片图标的不足：

1. 字体图标本质就是字体，和文字一样可以用CSS指定它的颜色、形状和透明度等样式。
2. 字体图标使用简单，可以直接使用编码进行显示，不需要像图片图标一样再去定位图片的位置。
3. 字体图标就是矢量的，这是字体图标最明显的优点。由于是矢量，所以可以对字体图标进行任意的缩放、旋转和拉伸，这些转换都不会影响图标的清晰度。

当然字体图标也有不足：

1. 字体图标只支持单色或者简单的渐变效果，无法制作多彩的图标样式。
2. 字体图标维护的成本比较大，每制作一个新的图标都要制作几套不同格式的字体。
3. 使用图标库要加载几种格式的字体，且字体的体积一般不会很小。

图片图标和字体图标因为各自的优缺点，都有自己的适用场景，一般具有共性的且颜色比较单一的图标我们用字体图标，对于定制化要求比较高的就可以使用传统的图片图标。我们这节重点要讲的是字体图标的知识，图片图标的使用其实就是图片的使用，这里就不做单独的介绍。

## 字体图标原理

字体图标实际上就是一种字体，和我们平时使用的“宋体”、“楷体”、“苹方”、“微软雅黑”这些是一样的。字体的使用分为两部分，一部分是字体库，用来存放字体的编码和对应的字体图形；另一部分就是字体的调用，通过字符编码去字体库里查找对应的字体图形，然后显示在屏幕上。

有些字体是在操作系统里自带的，像我们使用文本字体的时候，就会默认去系统的字体库里去查询。但是，系统自带的字体库比较有限，为了能使用到更丰富的字体，现代Web中也支持以文件的形式引入外部字体文件。虽然在理论上我们可以在网页中使用各种自定义字体，但是实际上很少在文字上使用自定义字体的。一方面是因为制作一套字体文件需要花费很大的精力，不是个人开发者可以做到的，别人做好的字体也有版权限制不能随便使用；另一方面是字体库文件的体量会比较大，中文字库中大概包含几万个汉字，一个字体库文件就可能达到几十兆，按现在的网络速度还不能在网页上支持这么大文件的下载。但是字体图标就不同了，字体图标的图像相对起文字来还是要简单一些，更重要的是字体图标数量有限，整体字体库文件也不会太大，所以字体图标会先于自定义字体在网页开发中流行起来。

以Font Awesome字体图标库为例，我们看一下Font Awesome的源码：

```
1  /*!
2  * Font Awesome 4.7.0 by @davegandy - http://fontawesome.io - @fontawesome
3  * License - http://fontawesome.io/license (Font: SIL OFL 1.1, CSS: MIT License)
4  */
5  /* FONT PATH
6  * ----- */
7  @font-face {
8    font-family: 'FontAwesome';
9    src: url('../fonts/fontawesome-webfont.eot?v=4.7.0');
10   src: url('../fonts/fontawesome-webfont.woff2?v=4.7.0') format('embedded-otextype'), url('../fonts/fontawesome-webfont.woff?v=4.7.0') format('woff'), url('../fonts/fontawesome-webfont.ttf?v=4.7.0') format('truetype'), url('../fonts/fontawesome-webfont.svg?v=4.7.0') format('svg');
11   font-weight: normal;
12   font-style: normal;
13 }
14 .fa {
15   display: inline-block;
16   font: normal normal normal 14px/1 FontAwesome;
17   font-size: inherit;
18   text-rendering: auto;
19   -webkit-font-smoothing: antialiased;
20   -moz-osx-font-smoothing: grayscale;
21 }
```

在Font Awesome源码的最前面，就是对字体库文件的调用。在@font-face里使用font-family属性定义了一个名为“FontAwesome”的字体名称，这个名称就类似于“宋体”、“微软雅黑”这些，然后用src指定了字体库文件的位置。在下面.fa选择器的样式里用font属性指定了所有.fa元素都使用FontAwesome字体。经过这样的设置，就可以在.fa元素里使用字符编码来显示对应的字体图标了。

## 字体图标的使用

我们这个项目里使用的是一个比较优秀的字体图标库Font Awesome ( <http://fontawesome.dashgame.com/> )。这个字体图标库提供了网页开发中可以用到的大部分图标，并且有提供免费版本。本项目中我们使用最新的免费版本v4.7.0，下面我们就来介绍一下它的用法。

### 一、字体图标的引入方式

Font Awesome提供了多种引入方式，可以在HTML中使用link标签引入，也可以在CSS文件中使用@import引入，还可以通过npm工具以npm插件的方式引入。下面我们来演示下这几种引入方式。

#### 1、使用link标签引入

这种方式是最简单的，只需要使用link引入Font Awesome的地址即可。这个地址可以使用一些线上的CDN提供的文件地址；也可以把文件存到项目本地，然后使用本项目下的地址。用这种方式使用的时候，比较推荐使用线上的CDN文件，这样不占用自己服务器的资源，速度还比较快。唯一要注意的就是要找一个稳定的CDN服务，免得因为别人的服务挂了影响到自己的业务。我们以BootCDN提供的CDN服务为例，只需要在HTML的head标签里加入下面的代码，就可以引入Font Awesome文件了：

```
<link rel="stylesheet" href="//cdn.bootcss.com/font-awesome/4.7.0/css/font-awesome.min.css">
```

有了这个文件，就可以在HTML中按Font Awesome规定的形式使用字体图标了。

#### 2、在CSS文件中引用

在CSS文件中引入的方式和第一种方式很类似，也是可以使用CDN地址，也可以使用项目本地地址。这种引入方式的好处是当把Font Awesome作为依赖时，可以把Font Awesome封闭在框架的内部。但是这种方式也有些问题，就是我们之前提到的@import会影响页面的加载速度，并且通过@import引入网络路径的话不能使用工具进行文件合并。但为了整个框架的闭合，我们会采取这种方式引入Font Awesome。引入的时候直接在CSS文件中按下面方式引用即可。

```
@import '//cdn.bootcss.com/font-awesome/4.7.0/css/font-awesome.min.css';
```

这样引入以后，也可以在页面中使用Font Awesome字体了。

### 3、以npm插件形式引入

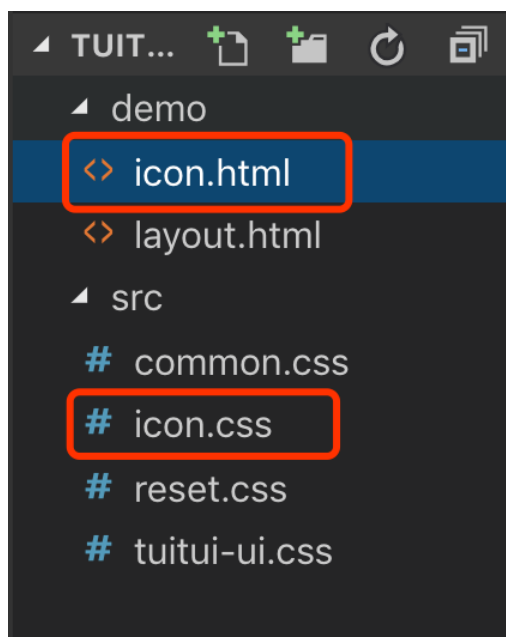
Font Awesome同样提供了npm插件，如果是npm项目，可以直接使用npm安装font-awesome这个插件来使用。我们这个项目是纯文件形式的，所以无法使用npm插件。在做框架类的项目时，以npm方式安装字体图标文件是最好的，它既可以满足项目封闭的要求，也不会像使用@import方式一样会影响页面的加载速度。用npm方式使用的时候要先安装font-awesome插件：

```
npm install font-awesome@4.7.0 --save
```

安装以后，还要结合其他工具才能在页面中引入字体图标文件，比如webpack或者gulp等工具。

### 二、字体图标的使用

刚才说了文件的引入方式，在文件成功引入以后，就可以在html中使用字体图标了。我们先在项目里把需要的文件建出来，包括DEMO下的icon.html文件和src下的icon.css文件：



接下来我们开始完成各个文件的内容。

首先，在 **/src/icon.css** 文件中使用@import方式来引入font-awesome，如下：

```
/*
 * @Author: Rosen
 * @Date: 2019-06-22 20:15:05
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-06-22 20:15:05
 */

/* 字体图标，依赖font-awesome */
@import '//cdn.bootcss.com/font-awesome/4.7.0/css/font-awesome.min.css';
```

然后，在 **/src/tuitui-ui.css** 这个主文件中，把刚才建好的 **/src/icon.css** 文件引入，这样整个项目里就有Font Awesome了。这个文件中已经引入了reset.css和common.css，现在把下面的语句加入到 **/src/tuitui-ui.css** 文件的末尾即可。

```
/* 字体图标 */
@import './icon.css';
```



接下来，在DEMO目录下建一个 **/demo/icon.html** 文件，用来测试一下字体图标的效果。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no">
    <link rel="stylesheet" href="../src/tuitui-ui.css">
    <title>字体图标</title>
  </head>
  <body>
    <!-- 内容区 -->
  </body>
</html>
```

@ Tips:

这是一个标准的移动端HTML文件，这里面要注意：

```
<meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no">
```

这条语句是移动端最常用的一个meta标签，这个标签是用来配置viewport属性的，viewport就是显示器的可视区域。在移动端浏览器中，为了让PC端的网页在手机上能正常显示，不至于把内容缩的太小，移动端浏览器就会让承载页面的区域大于屏幕的尺寸，这样可以让手机屏幕只显示网页的一部分内容，再通过拖拽和缩放来查看其他区域的内容。这种默认的行为虽然照顾了PC端的网页，但如果一个网页本来就是为移动端设计的，那么它的显示就是有问题的。所以浏览器提供了可以手动指定viewport值的方法，让承载页面的区域和屏幕等宽，并且使用user-scalable指定页面不可以缩放，这样原本为移动端设计的网页就可以正常显示了。

建好这几个文件后，我们就可以测试字体图标了。Font Awesome字体库被引入后，只需要在html中加入下面这样一个标签，即可显示出一个标签：

```
<i class="fa fa-address-book"></i>
```

这个标签的class里，fa是必须的，后面的fa-xxx就是用来指定显示哪个标签用的。我们需要使用哪个图标就可以去官方网站提供的图标列表里查询，找到合适的图标后把fa-xxx换成图标对应的名称即可。



4.7.0版新增41个全新的图标

address-book	address-book-o	address-card
address-card-o	bandcamp	bath
bathtub (alias)	drivers-license (alias)	drivers-license-o (alias)
eercast	envelope-open	envelope-open-o
etsy	free-code-camp	grav
handshake-o	id-badge	id-card
id-card-o	imdb	linode
meetup	microchip	podcast
quora	ravely	s15 (alias)
shower	snowflake-o	superpowers

我们以第一个图标来测试，把html中内容区换成如下内容：

```
<i class="fa fa-address-book"></i>
```

这样，这个图标就在页面的左上角显示出来了：



这样证明我们的字体图标就可以正常使用了。

### 三、自定义图标样式

经过刚才的努力，字体图标能正常显示，接下来我们就要演示一下修改图标的样式。在演示之前，我们先在 **/src/icon.css** 文件中给图标一个水平和竖直居中的样式，保证图标是一个正方形且图标在正中间。

```
/*
 * @Author: Rosen
 * @Date: 2019-06-22 20:15:05
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-06-22 20:47:03
 */

/* 字体图标，依赖font-awesome */
@import '//cdn.bootcss.com/font-awesome/4.7.0/css/font-awesome.min.css';

.fa{
  display: inline-block;
  text-align: center;
}
```

我们在这个文件里覆盖了Font Awesome自带的.fa样式。Font Awesome里把字体图标按行内元素来处理，我们这里把它变成行内块级元素，然后给他“text-align:center”属性，可以在图标有宽度的时候水平居中。在字体图标引入的地方修改图标的样式，就可以在全局都生效。

然后在 **/demo/icon.html** 文件中，加入几个用来测试样式用的图标：

```
<!-- 演示修改图标样式 -->
<h2>更改图标样式: </h2>
<div>
  <i class="fa fa-code"></i>
  <i class="fa fa-info"></i>
  <i class="fa fa-check"></i>
</div>
```

下来我们在 **/demo/icon.html** 文件的head标签中加入一个style标签来写测试用的样式。

我们先给图标一个默认样式，让图标变成一个正方形且水平竖直居中的：

```
/* 图标默认样式 */
.fa{
  width: 1.5rem;
  height: 1.5rem;
  line-height: 1.5rem;
  font-size: 1.2rem;
}
```



默认的风格如下所示：



然后再改一下fa-code的字体颜色，就可以用：

```
/* 红色图标 */  
.fa-code{  
  color: red;  
}
```

下来再把fa-info改成一个圆形的实心图标：

```
/* 圆形实心图标 */  
.fa-info {  
  background: #666;  
  color: #fff;  
  border-radius: 50%;  
}
```

我们通过把字体设置成白色，再把背景设置成灰色，就可以有一个实心的图形了。然后使用“border-radius: 50%;”这个属性，把图标变成圆形的。

最后把fa-check这个图标改成一个块级的大图标，可以用来做信息提示页的图标。

```
/* 大图标 */  
.fa-check {  
  display: block;  
  margin: 0 auto;  
  width: 6rem;  
  height: 6rem;  
  line-height: 6rem;  
  font-size: 3.5rem;  
  border-radius: 50%;  
  background: #09BB07;  
  color: #fff;  
}
```

这里我们重置了这个图标的长宽和字号，从而把图标放大。还把它变成了纯的块级元素，让它自己占一行显示，更适合做结果提示页里的大图标。

经过上面三个样式的改动，刚才默认的图标就变成了下面这种效果：

## 更改图标样式：



最后，再介绍一个比较特殊的用法，就是字体图标是支持旋转的。我们再来加几个测试用的图标：

```
<!-- 演示图标旋转 -->
<h2>图标旋转： </h2>
<div>
  <i class="fa fa-spinner"></i>
  <i class="fa fa-circle-o-notch"></i>
  <i class="fa fa-cube"></i>
</div>
```

默认的这几个图标都是静态的：

## 图标旋转：



但是如果我们给这几个图标加上“fa-spin”这个class，这几个图标就会转起来：

## 图标旋转：



经过这几个实验，希望让同学们体会到，字体图标就是文本，修改字体图标样式的时候就把它当成一个文本来改就可以了。下面放上最终完整的 `/demo/icon.html` 文件：

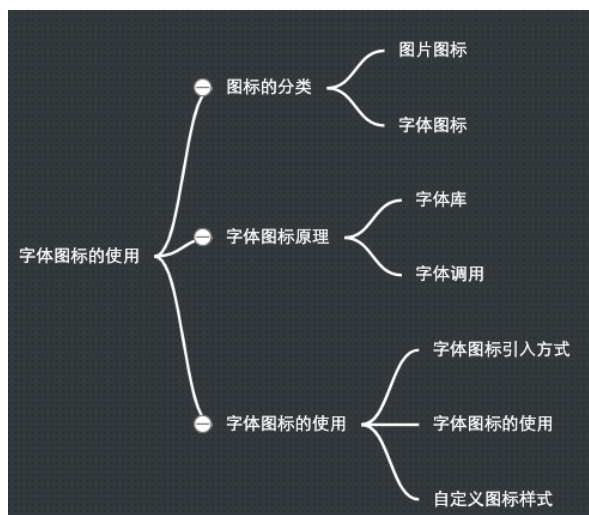
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no">
    <link rel="stylesheet" href="../../src/tuitui-ui.css">
    <title>字体图标</title>
    <style>
      /* 图标默认样式 */
      .fa{
        width: 1.5rem;
        height: 1.5rem;
        line-height: 1.5rem;
        font-size: 1.2rem;
      }
      /* 红色图标 */
      .fa-code{
        color: red;
      }
      /* 圆形实心图标 */
      .fa-info {
        background: #666;
        color: #fff;
        border-radius: 50%;
      }
      /* 大图标 */
      .fa-check {
        display: block;
        margin: 0 auto;
        width: 6rem;
        height: 6rem;
        line-height: 6rem;
        font-size: 3.5rem;
        border-radius: 50%;
        background: #09BB07;
        color: #fff;
      }
    </style>
  </head>
  <body>
    <i class="fa fa-address-book"></i>
    <!-- 演示修改图标样式 -->
    <h2>更改图标样式： </h2>
    <div>
      <i class="fa fa-code"></i>
      <i class="fa fa-info"></i>
      <i class="fa fa-check"></i>
    </div>
    <!-- 演示图标旋转 -->
    <h2>图标旋转： </h2>
    <div>
      <i class="fa fa-spinner fa-spin"></i>
      <i class="fa fa-circle-o-notch fa-spin"></i>
      <i class="fa fa-cube fa-spin"></i>
    </div>
  </body>
</html>

```

## 结语

经过这一节的学习，我们了解字体图标的分类，字体图标的原理和字体图标的使用方法。这一节的内容大纲如下：



这一节里字体图标的分类和图标原理这些内容，同学们能了解就可以，但是应用的部分一定要弄明白。这一节内容比较多也比较繁琐，但难度不大，只要同学们能耐心看下来就没什么问题。我们这一节的内容就到这里，下一节将进入页面布局的开发。