

响应式图片

一、JavaScript方式实现响应式图片

```
$(document).ready(function () {  
    function makeImageResponsive() {  
        var width = $(window).width();  
        var img = $('img');  
        if ( width <= 480 ) {  
            img.attr('src', 'img/480.png');  
        } else if ( width <= 800 ) {  
            img.attr('src', 'img/800.png');  
        } else {  
            img.attr('src', 'img/1600.png');  
        }  
    }  
    $(window).on('resize load', makeImageResponsive);  
});
```

二、利用 srcset 和 sizes 实现响应式图片

兼容：IE完全不兼容，Android Browser 4.7及以上，IOS Safari 8.4及以上

```
<style>  
.content {  
    width: 50%;  
    margin: 0 auto;  
}  
.content img {  
    display: block;  
    width: 100%;  
    max-width: 100%;  
}  
</style>  
</head>  
<body>  
<div class="content">  
      
    <!-- sizes="(min-width:800px) calc(100vw - 30em), 100vw"--> // 注意：calc里减号左右空格没有就无效  
    <!-- sizes="50vw"是因为css里 .content 的 width: 50%; -->  
</div>
```

三、利用 picture 标签实现响应式图片 兼容：IE完全不兼容，Android Browser 4.7及以上，IOS Safari 9.3及以上

```
<style>  
.content {  
    width: 50%;  
    margin: 0 auto;  
}  
.content img {  
    display: block;  
    width: 100%;  
    height: auto;  
    max-width: 100%;  
}  
</style>  
<div class="content">  
    <picture>  
        <source media="(max-width:36em)" srcset="img/tiananmen-s.jpg, img/tiananmen-retina.jpg 2x" /> <!--如果您想为更高密度的显示器提供  
不同分辨率的图像版本，可以通过在 srcset 属性中添加额外的文件名来实现。例如，让我们来看看屏幕像素密度为 2x 的Retina(视网膜) 代码处理  
片断：-->  
        <source media="(orientation:landscape)" srcset="img/tiananmen-s.jpg 768w" /> <!--备注：media="(orientation:landscape)" 指横屏-->  
        <source srcset="img/tiananmen.jpg 1800w" />  
        <source type="image/svg+xml" srcset="logo.svg 480w, logo-m.svg 800w, logo-l.svg 1600w" /> <!--如果浏览器支持svg，则用svg图片 -->  
    <source type="image/webp" srcset="logo.webp 480w, logo-m.webp 800w, logo-l.webp 1600w" /> <!--如果浏览器支持webp，则用webp  
图片 -->
```

```
    
  </picture>
</div>
```

四、使用picturefill解决picture标签的兼容问题

网址<http://scottjehl.github.io/picturefill/>

第一步：引入picturefill.min.js

```
<script src="../js/picturefill.min.js"></script>
```

第二步：引入picturefill.min.js，就可以使用picture标签了，注意：picturefill.min.js已经忽略了DPR，在任意DPR下，图片的切换规则不会改变，省去了浏览器计算，这就是picturefill的好处

```
<div class="item">
  <picture>
    <source media="(min-width:50em)" srcset="img/ad001-l.jpg" />    <!-- 大图片 -->
    <source media="(min-width:30em)" srcset="img/ad001-m.jpg" />    <!-- 中等图片 -->
    <source srcset="img/ad001.jpg" alt="响应式图片" />
  </picture>
</div>
<div class="item">
  <picture>
    <source media="(min-width:50em)" srcset="img/ad002-l.jpg" />    <!-- 大图片 -->
    <source media="(min-width:30em)" srcset="img/ad002-m.jpg" />    <!-- 中等图片 -->
    <source srcset="img/ad002.jpg" alt="响应式图片" />
  </picture>
</div>
```