



链滴

移动前端开发——网页适配并响应不同的分辨率的设备

作者: [limanting](#)

原文链接: <https://ld246.com/article/1628579776718>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>要在移动设备上进行网页开发，只有明白移动设备上的 viewport，并且掌握跟 viewport 有关的 meta 标签的使用，才能更好地让网页适配或响应各种不同分辨率的移动设备。</p>

<h2 id="viewport">viewport</h2>

<p>移动设备上的 viewport 就是设备屏幕上能用来显示网页的区域。</p>

<p>视口当前可见的部分叫做可视视口 (visual viewport)。
可视视口可能会比布局视口 (layout viewport)小，因为当用户缩小浏览器缩放比例时，布局视口不变，而可视视口变小了。</p>

<p>ideal viewport 是最适合移动设备的 viewport，ideal viewport 的宽度于移动设备的屏幕宽度，只要在 css 中把某一元素的宽度设为 ideal viewport 的宽度(单位用 px)，么这个元素的宽度就是设备屏幕的宽度了</p>

<h3 id="css-中的-1px-并不等于设备的-1px">css 中的 1px 并不等于设备的 1px</h3>

<p>css 中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css 中的 1px 所代表的物理像素是不同的。在为桌面浏览器设计的网页中可以忽略这个问题，但在移动设备上必须弄明白这。</p>

<h3 id="利用-meta-标签对-viewport-进行控制">利用 meta 标签对 viewport 进行控制</h3>

<p>移动设备默认的 viewport 是 layout viewport，但在进行移动设备网站的开发时，我们需要是 ideal viewport，最常见的的一个动作就是把下面这个东西复制到我们的 head 标签中</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">&lt;meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0"&gt;</span></span><span class="highlight-line"><span class="highlight-cl">// 让当前viewport的宽度等于设备的宽度，同时不允许用户手动缩放。</span></span><span class="highlight-line"><span class="highlight-cl">// 如果不这样的设的话，默认会使用那个比屏幕宽的默认viewport，即出现横向滚动条。</span></span></code></pre>
```

<table>

<thead>

<tr>

<th>content 中的属性</th>

<th>content 中的值</th>

</tr>

</thead>

<tbody>

<tr>

<td>width</td>

<td>设置 layout viewport 的宽度，为一个正整数，或字符串"width-device"</td>

</tr>

<tr>

<td>initial-scale</td>

<td>设置页面的初始缩放值，为一个数字，可以带小数</td>

</tr>

<tr>

<td>minimum-scale</td>

<td>允许用户的最小缩放值，为一个数字，可以带小数</td>

</tr>

<tr>

<td>maximum-scale</td>

<td>允许用户的最大缩放值，为一个数字，可以带小数</td>

</tr>

<tr>

<td>height</td>

<td>设置 layout viewport 的高度，这个属性对我们并不重要，很少使用</td>

</tr>

```

<tr>
<td>user-scalable</td>
<td>是否允许用户进行缩放, 值为"no"或"yes", no 代表不允许, yes 代表允许</td>
</tr>
</tbody>
</table>

```

<p>缩放是相对于 ideal viewport 来进行缩放的。</p>

<p>visual viewport 宽度 = ideal viewport 宽度 / 当前缩放值 </p>

<p>当前缩放值 = ideal viewport 宽度 / visual viewport 宽度 </p>

<p>如果 width 和 initial-scale=1 同时出现, 并且还出现了冲突, 浏览器会取它们两个中较大的个值。</p>

<p>为什么两个都要写? 因为只写一个的话, iphone、ipad 以及 IE 会横竖屏不分, 通通以竖屏的 ideal viewport 宽度为准, 两者都写上去就 initial-scale=1 解决了 iphone、ipad 的毛病, width=device-width 则解决了 IE 的毛病。</p>

<p>在 iphone 和 ipad 上, 无论你给 viewport 设的宽的是多少, 如果没有指定默认的缩放值, 则 iphone 和 ipad 会自动计算这个缩放值, 以达到当前页面不会出现横向滚动条(或者说 viewport 的宽度是屏幕的宽度)的目的。</p>

动态改变 meta viewport 标签</h3>

<p>方法一: </p>

```

<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">使用document.write来动态输出meta viewport标签
</span> </span> <span class="highlight-line"> <span class="highlight-cl">document.write('&
t;meta name="viewport" content="width=device-width,initial-scale=1"&gt;')
</span> </span> </code> </pre>

```

<p>方法二</p>

```

<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">&lt;meta id="testViewport" name="viewport" content="width = 380"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">&lt;script&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">var mvp = docum
nt.getElementById('testViewport');
</span> </span> <span class="highlight-line"> <span class="highlight-cl">mvp.setAttribute('
ontent','width=480');
</span> </span> <span class="highlight-line"> <span class="highlight-cl">&lt;/script&gt;
</span> </span> </code> </pre>

```

<p>方法三</p>

```

<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">(function (doc, win) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    var docEl =
doc.documentElement,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    resizeEvt =
'orientationchange' in win ? 'orientationchange' : 'resize', // 监听是否有屏幕翻转的动作
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    recalc = fun
tion () {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        var clientW
idth = docEl.clientWidth &gt; 1024 ? 1024 : docEl.clientWidth;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        if (!clientW
idth) return;
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        docEl.style.
fontSize = clientWidth / 3.75 + 'px';
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    };
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    win.addEvent
Listener(resizeEvt, recalc, false);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    recalc();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">})(document, w

```

```
ndow);  
</span> </span> </code> </pre>
```