**一、允许网页宽度自动调整**

"自适应网页设计"到底是怎么做到的？其实并不难。

首先，在网页代码的头部，加入一行[viewport元标签](https://developer.mozilla.org/en/mobile/viewport_meta_tag" \t "_blank)。

　　<meta name="viewport" content="width=device-width, initial-scale=1" />

[viewport](https://developer.apple.com/library/ios/" \l "DOCUMENTATION/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html" \t "_blank)是网页默认的宽度和高度，上面这行代码的意思是，网页宽度默认等于屏幕宽度（width=device-width），原始缩放比例（initial-scale=1）为1.0，即网页初始大小占屏幕面积的100%。

所有主流浏览器都支持这个设置，包括IE9。对于那些老式浏览器（主要是IE6、7、8），需要使用[css3-mediaqueries.js](http://code.google.com/p/css3-mediaqueries-js/" \t "_blank)。

　　<!--[if lt IE 9]>  
　　　　<script src="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-mediaqueries.js"></script>  
　　<![endif]-->

**二、不使用绝对宽度**

由于网页会根据屏幕宽度调整布局，所以不能使用绝对宽度的布局，也不能使用具有绝对宽度的元素。这一条非常重要。

具体说，CSS代码不能指定像素宽度：

　　width:xxx px;

只能指定百分比宽度：

　　width: xx%;

或者

　　width:auto;

**三、相对大小的字体**

字体也不能使用绝对大小（px），而只能使用相对大小（em）。

　　body {  
　　　　font: normal 100% Helvetica, Arial, sans-serif;  
　　}

上面的代码指定，字体大小是页面默认大小的100%，即16像素。

　　h1 {  
　　　　font-size: 1.5em;   
　　}

然后，h1的大小是默认大小的1.5倍，即24像素（24/16=1.5）。

　　small {  
　　　　font-size: 0.875em;  
　　}

small元素的大小是默认大小的0.875倍，即14像素（14/16=0.875）。

**四、流动布局（fluid grid）**

["流动布局"](http://www.alistapart.com/articles/fluidgrids/" \t "_blank)的含义是，各个区块的位置都是浮动的，不是固定不变的。

　　.main {  
　　　　float: right;  
　　　　width: 70%;   
　　}

　　.leftBar {  
　　　　float: left;  
　　　　width: 25%;  
　　}

[float](http://designshack.net/articles/css/everything-you-never-knew-about-css-floats/" \t "_blank)的好处是，如果宽度太小，放不下两个元素，后面的元素会自动滚动到前面元素的下方，不会在水平方向overflow（溢出），避免了水平滚动条的出现。

另外，绝对定位（position: absolute）的使用，也要非常小心。

**五、选择加载CSS**

"自适应网页设计"的核心，就是CSS3引入的[Media Query](http://www.w3.org/TR/CSS21/media.html" \t "_blank)模块。

它的意思就是，自动探测屏幕宽度，然后加载相应的CSS文件。

　　<link rel="stylesheet" type="text/css"  
　　　　media="screen and (max-device-width: 400px)"  
　　　　href="tinyScreen.css" />

上面的代码意思是，如果屏幕宽度小于400像素（max-device-width: 400px），就加载tinyScreen.css文件。

　　<link rel="stylesheet" type="text/css"  
　　　　media="screen and (min-width: 400px) and (max-device-width: 600px)"  
　　　　href="smallScreen.css" />

如果屏幕宽度在400像素到600像素之间，则加载smallScreen.css文件。

除了用html标签加载CSS文件，还可以在现有CSS文件中加载。

　　@import url("tinyScreen.css") screen and (max-device-width: 400px);

**六、CSS的@media规则**

同一个CSS文件中，也可以根据不同的屏幕分辨率，选择应用不同的CSS规则。

　　@media screen and (max-device-width: 400px) {

　　　　.column {  
　　　　　　float: none;  
　　　　　　width:auto;  
　　　　}

　　　　#sidebar {  
　　　　　　display:none;  
　　　　}

　　}

上面的代码意思是，如果屏幕宽度小于400像素，则column块取消浮动（float:none）、宽度自动调节（width:auto），sidebar块不显示（display:none）。

**七、图片的自适应（fluid image）**

除了布局和文本，"自适应网页设计"还必须实现图片的[自动缩放](http://unstoppablerobotninja.com/entry/fluid-images" \t "_blank)。

这只要一行CSS代码：

img { max-width: 100%;}

这行代码对于大多数嵌入网页的视频也有效，所以可以写成：

　　img, object { max-width: 100%;}

老版本的IE不支持max-width，所以只好写成：

　　img { width: 100%; }

此外，windows平台缩放图片时，可能出现图像失真现象。这时，可以尝试使用IE的[专有命令](http://css-tricks.com/ie-fix-bicubic-scaling-for-images/" \t "_blank)：

　　img { -ms-interpolation-mode: bicubic; }

或者，Ethan Marcotte的[imgSizer.js](http://unstoppablerobotninja.com/demos/resize/imgSizer.js" \t "_blank)。

　　addLoadEvent(function() {

　　　　var imgs = document.getElementById("content").getElementsByTagName("img");

　　　　imgSizer.collate(imgs);

　　});