|  |  |
| --- | --- |
| |  | | --- | | 讲到[响应式布局](http://www.jiawin.com/tag/%e5%93%8d%e5%ba%94%e5%bc%8f%e5%b8%83%e5%b1%80/)，相信大家都有一定的了解，响应式布局是今年很流行的一个设计理念，随着移动互联网的盛行，为解决如今各式各样的浏览器分辨率以及不同移动设备的显示效果，设计师提出了响应式布局的设计方案。今天就和大家来讲讲响应式布局这件小事，包含什么是响应式布局、响应式布局的优点和缺点以及响应式布局该怎么设计（通过CSS3 [Media Query](http://www.jiawin.com/tag/media-query/)实现响应布局）。  http://www.w3cfuns.com/data/attachment/forum/201211/11/1405119xoyla39l3orro5a.jpg  **一、什么是响应式布局?** 响应式布局是Ethan Marcotte在2010年5月份提出的一个概念，简而言之，就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本。这个概念是为解决移动互联网浏览而诞生的。 响应式布局可以为不同终端的用户提供更加舒适的界面和更好的用户体验，而且随着目前大屏幕移动设备的普及，用大势所趋来形容也不为过。随着越来越多的设计师采用这个技术，我们不仅看到很多的创新，还看到了一些成形的模式。 **二、响应式布局有哪些优点和缺点? 优点：**   1. 面对不同分辨率设备灵活性强 2. 能够快捷解决多设备显示适应问题   **缺点：**   1. 兼容各种设备工作量大，效率低下 2. 代码累赘，会出现隐藏无用的元素，加载时间加长 3. 其实这是一种折衷性质的设计解决方案，多方面因素影响而达不到最佳效果 4. 一定程度上改变了网站原有的布局结构，会出现用户混淆的情况   **三、响应式布局该怎么设计?** 我们在上面了解了什么是响应式布局，那在我们的实际项目中应该怎么去设计呢?在以往我们设计网站的时候都会受到不同浏览器的兼容性的困扰，现在还要来个不同尺寸设备，我们该怎么淡定下来呢?有需求就会有解决方案，呵呵，说到响应式布局，就不得不提起CSS3中的[Media Query](http://www.jiawin.com/tag/media-query/)（媒介查询），这可是个好东西，易用、强大、快捷……Media Query是制作响应式布局的一个利器，使用这个工具，我们可以非常方便快捷的制造出各种丰富的实用性强的界面。接下来就一起来深入的了解Media Query。 **1、CSS中的Media Query（媒介查询）是什么?** 通过不同的媒体类型和条件定义样式表规则。媒体查询让CSS可以更精确作用于不同的媒体类型和同一媒体的不同条件。媒体查询的大部分媒体特性都接受min和max用于表达”大于或等于”和”小与或等于”。如：width会有min-width和max-width媒体查询可以被用在CSS中的@media和@import规则上，也可以被用在HTML和XML中。通过这个标签属性，我们可以很方便的在不同的设备下实现丰富的界面，特别是移动设备，将会运用更加的广泛。 **2、media query能够获取哪些值?**   * 设备的宽和高device-width，device-heigth显示屏幕/触觉设备。 * 渲染窗口的宽和高width，heigth显示屏幕/触觉设备。 * 设备的手持方向，横向还是竖向orientation(portrait|lanscape)和打印机等。 * 画面比例aspect-ratio点阵打印机等。 * 设备比例device-aspect-ratio-点阵打印机等。 * 对象颜色或颜色列表color，color-index显示屏幕。 * 设备的分辨率resolution。   **3、语法结构及用法**   1. @media 设备名 only (选取条件) not (选取条件) and(设备选取条件)，设备二{sRules}   复制代码  **示例一：在link中使用@media：**   1. <link rel="stylesheet" type="text/css" media="only screen and (max-width: 480px),only screen and (max-device-width: 480px)" href="link.css"/>   复制代码  上面使用中only可省略，限定于计算机显示器，第一个条件max-width是指渲染界面最大宽度，第二个条件max-device-width是指设备最大宽度。  **示例二：在样式表中内嵌@media：**   1. @media (min-device-width:1024px) and (max-width:989px),screen and (max-device-width:480px),(max-device-width:480px) and (orientation:landscape),(min-device-width:480px) and (max-device-width:1024px) and (orientation:portrait) {srules}   复制代码  在示例二中，设置了电脑显示器分辨率（宽度）大于或等于1024px（并且最大可见宽度为989px）；屏宽在480px及其以下手持设备；屏宽在480px以及横向（即480尺寸平行于地面）放置的手持设备；屏宽大于或等于480px小于1024px以及垂直放置设备的css样式。 从上面的例子可以看出，字符间以空格相连，选取条件包含在小括号内，srules为兼容设置的样式表，包含在中括号里面。only（限定某种设备，可省略），and（逻辑与），not（排除某种设备）为逻辑关键字，多种设备用逗号分隔，这一点继承了css基本语法。 **4、可用设备名参数：**  http://www.w3cfuns.com/data/attachment/forum/201211/11/1406586aa9zcsttbk635a9.jpg  **5、逻辑关键字：**  http://www.w3cfuns.com/data/attachment/forum/201211/11/140658lp1hjpbjy5c0c0ya.jpg  **6、可用设备名参数：**  http://www.w3cfuns.com/data/attachment/forum/201211/11/140701iv4ycciwzcika4aw.jpg  **7、测试Media Queries** 最后，我们需要对我们刚刚设计的Media Queries进行测试，想要在不同设备上测试Media Queries的效果，可以使用一个浏览工具来检验不同尺寸屏幕下的显示效果，在这里为大家介绍一个不错的在线工具 – [**Responsivator**](http://dfcb.github.com/Responsivator/?site=http://www.jiawin.com)，它可以模拟iPhone等各种不同设备，并且还可以自定义不同尺寸屏幕的显示效果，只需要输入一个url甚至是本地的一个url(如:<http://127.0.0.1/>)，就可以看到网站在不同尺寸屏幕下的显示效果. **8、通过Media Queries实现响应式布局设计** 好了，我们明白了什么是Media Query，那我们一起来运用到响应式布局的设计项目中去。设计思路很简单，首先先定义在标准浏览器下的固定宽度（假如标准浏览器的分辨率为1024px，那么我们设置宽为980px），然后用Media Query来监测浏览器的尺寸变化，当浏览器的分辨率小于1024px的时候，则通过Media Query预设的样式表来将页面的宽度设置为百分比显示，这样子页面的结构元素就会根据浏览器的的尺寸来进行相对应的调整。同理，当浏览器的可视区域改变到某个值（假如为650px）的时候，页面的结构元素根据Media Query预设的层叠样式表来进行相对应的调整。看看我们的例子：   1. /\* 当浏览器的可视区域小于980px \*/ 2. @media screen and (max-width: 980px) { 3. #wrap {width: 90%; margin:0 auto;} 4. #content {width: 60%;padding: 5%;} 5. #sidebar {width: 30%;} 6. #footer {padding: 8% 5%;margin-bottom: 10px;} 7. } 8. /\* 当浏览器的可视区域小于650px \*/ 9. @media screen and (max-width: 650px) { 10. #header {height: auto;} 11. #searchform {position: absolute;top: 5px;right: 0;} 12. #content {width: auto; float: none; margin: 20px 0;} 13. #sidebar {width: 100%; float: none; margin: 0;} 14. }   复制代码  通过上面我们就可以监测浏览器的可视区域变化的是时候我们的页面结构元素也会相对应的变化，当然你可以再多设置几个尺寸的监测层叠样式表，这样子就可以根据不同尺寸设备来进行响应式的布局。为了更好的显示效果，我们往往还要格式化一些CSS属性的初始值：   1. /\* 禁用iPhone中Safari的字号自动调整 \*/ 2. html { 3. -webkit-text-size-adjust: none; 4. } 5. /\* 设置HTML5元素为块 \*/ 6. article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section { 7. display: block; 8. } 9. /\* 设置图片视频等自适应调整 \*/ 10. img { 11. max-width: 100%; 12. height: auto; 13. width: auto\9; /\* ie8 \*/ 14. } 15. .video embed, .video object, .video iframe { 16. width: 100%; 17. height: auto; 18. }   复制代码  最后要注意的是在页面的头部<head></head>之间加上下面这句∶   1. <meta name="viewport" content="width=device-width; initial-scale=1.0">   复制代码  meta viewport这个属性是在移动设备上设置原始大小显示和是否缩放的声明。 **参数设置∶**   * width – viewport的宽度 * height – viewport的高度 * initial-scale – 初始的缩放比例 * minimum-scale – 允许用户缩放到的最小比例 * maximum-scale – 允许用户缩放到的最大比例 * user-scalable – 用户是否可以手动缩放   最后对于在IE浏览器中不支持media query的情况，我们可以使用Media Query JavaScript来解决，只需要在页面头部引用css3-mediaqueries.js即可。示例：   1. <!--[if lt IE 9]> 2. <script src="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-mediaqueries.js"> 3. </script> <![endif]-->   复制代码 | |