响应式网页设计流程及弹性图片研究

11302010003 宋波

摘要：响应式网页设计是当今移动互联网时代重要的网页技术，它能使用户在不同设备上（如智能手机，平板电脑，台式电脑，甚至未来的手表，眼镜等），浏览网页时获得最佳的用户体验。而在响应式网页设计的众多细节技术中，弹性图片是重要的一环。本文将首先介绍响应式网页设计的基本要素及流程，之后对弹性图片进行研究分析。

**简介**

随着科学技术的不断发展，人们浏览网页不仅仅局限在传统的台式电脑上了，用户可以在越来越多的设备上进行网页浏览，例如手机，平板电脑，超级本，台式电脑等。同种设备的分辨率也不尽相同，例如现在台式电脑的屏幕有小有大，分辨率从1024\*768至1920\*1080不等，五花八门的手机、平板电脑的分辨率也基本上不同，再加上新产生的视网膜屏幕越来越多地运用到手机，平板电脑，笔记本电脑上，越来越多的问题由此而生。传统的固定布局的网页设计的局限性越来越明显，因此，一种能够根据不同屏幕，不同分辨率，采取不同的布局，不同的字体，不同的图片等进行适应的响应式网页设计技术应运而生。在这之中，由于图片相对较高的大小，以及特殊的缩放特性，弹性图片成为了响应式网页设计的重点之一。本文将首先通过响应式网页设计的几个关键技术，如媒体查询、流体布局、列布局等来论述该设计的基本流程。之后，从不同角度如载入速度，视网膜显示，客户端缓存等方面分析论述弹性图片技术。

媒体查询

响应式网页设计的基础技术便是媒体查询（Media Queries）。媒体查询的前身是CSS2中的媒体类型(Media Type)，可以通过它对不同的设备指定不同的样式。W3C总共列出了10种媒体类型，常见的有all(全部)，screen（屏幕），print（打印及打印预览）等。可以看出，这些媒体类型的种类较少，局限性非常大，无法适应种类繁多的移动设备。而媒体查询在CSS3中的出现有效地弥补了这一缺点。

在CSS3中，可以通过媒体查询设置不同类型的媒体条件，并根据对应的条件，给相应符合条件的媒体调用相对应的样式表。媒体查询可以通过HTML中的link标签引入，或是在CSS中直接引入，如下所示：

1．link方式引入

<link rel="stylesheet" media="screen and (max-width: 980px)" href="middle.css" />

2．@import方式引入

@media screen and (max-width: 980px) {

选择器 {

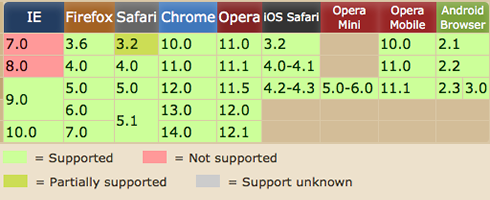
属性：属性值；

}

}

以上的媒体查询的内容由三部分组成：

1. 媒体类型：如screen，与CSS2中的相同。
2. 媒体特性：以(属性：值)的形式指定所选媒体的特性。如(max-width: 980px)指定了最大宽度不超过980px的媒体。常用的属性由图表1所示。
3. 关键字：用于组合不同的媒体特性，有and，not，only三种。and表示需满足两种组合的媒体特性，not表示排除某种制定的媒体类型，only指定某种特定的媒体类型，可以用来排除不支持媒体查询的浏览器。



图表 2 浏览器兼容性

图表 1 常见属性

此外，由于媒体查询是CSS3引入的特性，浏览器的兼容性存在一些问题，具体兼容性情况由图表2所示。

布局

通过媒体查询，我们可以根据不同的屏幕来指定不同的布局来达到优化用户体验的效果。在什么样的屏幕下用什么样的布局是一个很复杂的问题，现有的解决方案有非常多，在这里我们选取一种常见的流体布局加上列布局的方案进行阐述。

当屏幕的分辨率宽度大于980px时，屏幕中间的主容器宽度为980px；当小于980px时，布局不再采用固定布局来而将采用流体宽度布局来替代；如果小于650px，边栏的宽度和主内容的宽度也变成全屏，采用列布局进行显示。典型样例如图表三所示。

图表 3 常见响应式网页布局

为强调重点，简化冗余部分，我们尽将整个pagewarp（页面容器）分为header（包括导航栏，logo，描述等），content(主内容区)，sidebar（边栏）。其简化过的HTML代码如下所示：

<div id="pagewrap">

<header id="header">

<hgroup>

<h1 id="logo">Demo</h1>

<h2 id=" description">Site Description</h2>

</hgroup>

<nav>

<ul id="nav"> … </ul>

</nav>

</header>

<div id="content"> … </div>

<aside id="sidebar"> … </aside>

</div>

大于980px时的简化CSS代码如下所示，pagewrap宽度设为980px，header高度设为160px，content宽度设为600px并将其设置为左浮动，sidebar设为280px并将其设置为右浮动。

#pagewrap {

width: 980px;

margin: 0 auto;

}

#header {

height: 160px;

}

#content {

width: 600px;

float: left;

}

#sidebar {

width: 280px;

float: right;

}

}

当页面宽度小于980px时，显然不能再以px进行固定布局，所以将各宽度以相对百分比来替代，代码如下所示：

@media screen and (max-width: 980px) {

#pagewrap {

width: 95%;

}

#content {

width: 60%;

padding: 3% 4%;

}

#sidebar {

width: 30%;

padding: 8% 7%;

margin-bottom: 10px;

}

}

而当页面宽度小于650px时，需要将

@media screen and (max-width: 650px) {

#header {

height: auto;

}

#nav {

position: static;

}

#logo {

margin: 15px 100px 5px 0;

position: static;

}

#description {

margin: 0 0 15px;

position: static;

}

#content {

width: auto;

float: none;

margin: 20px 0;

}

#sidebar {

width: 100%;

float: none;

margin: 0;

}

}