

热门：响应图片(Responsive Images)技术简介

这篇文章发布于 2012年02月15日, 星期三, 22:41, 归类于 [Web综合](#)。阅读 63096 次, 今日 9 次 [5 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=2204>

一、开门见山

响应图片技术可以说是[响应布局](#)衍生出来的一个小分支。说白了, 就是不同显示器宽度下调用不同的图片。

这玩意, 最近在国外讨论很火, 有几种不同的实现方法, 但都并不复杂。

二、Cookie + Server实现

github上有该技术[介绍](#)。该技术需要Javascript以及后台程序的配合。

加载需要的JS文件[responsive-images.js](#), 该JS文件为获取当前用户显示器宽度, 然后根据宽度值大小决定图片是小(small)呢, 中等(medium)呢还是大(large), 然后写入对应的Cookie。

下面就是服务器端事了, 服务器(可以通过配置项)检测Cookie是被设成了small呢, medium呢还是large决定图片的src地址到底是哪个。

github上项目有个demo, 点击[这里](#)访问。

首次进入的时候是个小图(Cookie首次写入, 后台无法获取), 再次刷新就是个大家伙了:





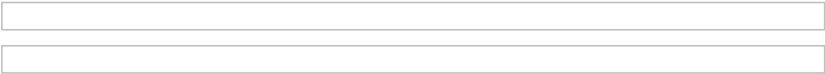
与响应布局差异

虽说该技术有响应布局的影子，但是，实际上实现的原理以及效果等是大相径庭的。差异在于：

1. 非CSS技术，为Cookie + server实现
2. 与浏览器宽度无关，即改变浏览器宽度图片不会有变化
3. 兼容具有考古价值的浏览器

这种响应方式跟京东商城首页几乎如出一辙。页面宽度由用户显示器分辨率决定（非浏览器）。

下图为1024像素显示器宽度和1680像素下(1280像素宽度就足够了)的京东商城页面布局截图。



兼容性

因为使用Cookie，并非CSS3 `media queries`，因此低版本IE浏览器下也是OK的。

如果设备不支持Javascript以及Cookie

如果访问设备不支持Javascript以及Cookie, 则图片会显示mobile手机上尺寸，即小尺寸。

可选项

`responsive-images.js`是一个非常小的脚本，不会其也会有可选项的。可选参数为一个全局的对象字面量。

```
<script>
//这里为配置项:
var responsive_images = {
  "key": "value"
};
</script>
```

对象字面量的名称是固定的，为`responsive_images`，支持的关键字、默认值和含义见下表：

关键字
默认值
释义
cookieName
rwd-screensize
Cookie名
cookieValue
sw > 500 ? (sw > 1000 ? "large" : "medium") : "small"
Cookie值

cookieAge
30000
过期时间, 单位毫秒
cookieDomain
当前访问
Cookie域
cookiePath
/
Cookie路径

三、其他同行的Cookie实现

keithclark在去年夏天的时候就说过响应图片, 不过其文章更多的是展示服务器端的处理(PHP), 原理与上面类似。

先是写Cookie, 很简短的:

```
document.cookie = "device_dimensions=" + screen.width + "x" + screen.height;
```

然后就是PHP语言的绽放了:

```
<?php
$device_width = 0;
$device_height = 0;
$file = $_SERVER['QUERY_STRING'];

if (file_exists($file)) {

    // Read the device viewport dimensions
    if (isset($_COOKIE['device_dimensions'])) {
        $dimensions = explode('x', $_COOKIE['device_dimensions']);
        if (count($dimensions)==2) {
            $device_width = intval($dimensions[0]);
            $device_height = intval($dimensions[1]);
        }
    }
}

if ($device_width > 0) {

    $fileext = pathinfo($file, PATHINFO_EXTENSION);

    // Low resolution image
    if ($device_width <= 800) {
        $output_file = substr_replace($file, '-low', -strlen($fileext)-1, 0);
    }

    // Medium resolution image
    else if ($device_width <= 1024) {
        $output_file = substr_replace($file, '-med', -strlen($fileext)-1, 0);
    }

    // check the file exists
    if (isset($output_file) && file_exists($output_file)) {
        $file = $output_file;
    }
}
```

```
// return the file;
readfile($file);
}

?>
```

然后HTML端的庐山面目：

```
<!doctype html>
<html>
<head>
  <title>Responsive Images Test</title>
  <meta charset="utf-8">
  <script>
    document.cookie = "device_dimensions=" + screen.width + "x" + screen.height;
  </script>
</head>
<body>
  
  <!-- 上面等同于:  -->
</body>
</html>
```

这个嘛希望对喜欢copy代码的同行有所帮助。

四、使用noscript标签创建响应图片

上面例子中的响应图片在两种情况下会直接mobile尺寸图片显示：一是站点首次载入的时候；二是Cookie禁用或是Javascript出去撞劳斯莱斯的时候。

这里的 `noscript` 标签创建法则基本没有这些问题，因为该方法完全借助客户端实现。

同样，github上有该方法的项目页面 – [futurechimp/responsive_image_tag](https://github.com/futurechimp/responsive_image_tag).

这里的响应图片技术貌似不需要Cookie的前后端通讯，而是借助 `noscript` 标签以及JS实现。

下面就是实现的步步展示：

首先是HTML部分：

```
<span class="img-placeholder"></span>
<noscript data-mobilesrc="small.jpg" data-fullsrc="big.jpg" data-alttext="your alt text" class="
  "responsivize">
  
</noscript>
```

上面代码中，类名为 `img-placeholder` 的 `span` 标签是用来占位子和插入图片的。`<noscript>` 中的图片默认调用的是大图地址，在JavaScript被禁用的时候会显示。

下面要做的就是使用JavaScript判别当前显示设备是mobile类别还是desktop类别：

```
var responsiveImageTag = {
  replaceInitialImages:function() {
    var platform = "desktop";
    var responsiveImages = $$(".responsivize");
```

```

var i,
    noOfresponsiveImages = responsiveImages.length;
//当前显示器设备宽度测试
if(screen.width <= 767){
    //767px, 比 ipad 小的都认为是 mobile
    platform = "mobile";
}

//set initial source element on image
for(i = 0; i < noOfresponsiveImages; i = i + 1 ){
    var noScriptElem = $(responsiveImages[i]);

```

既然知道了设备类别，我们就可以根据这个类别分别载入对应尺寸的图片了。对应设备尺寸图片地址以通过HTML5自定义属性 `data-` 定义在了 `<noscript>` 上，因此，我们就有代码（紧接上面）：

```

var img = window.document.createElement("img");
img.alt = noScriptElem.attr("data-alttext");
if(platform === "mobile"){
    img.src = noScriptElem.attr("data-mobilesrc");
}else{
    img.src = noScriptElem.attr("data-fullsrc");
}
img.className = "responsive";

```

然后，把图片以DOM的形式插入到上面用来占位的 `span` 标签元素内，于是就大功告成！

```

        noScriptElem.prev().append(img);
        noScriptElem.hide();
    }
}
};

```

五、另类图片响应

Jeremy Keith提出了另外一种响应图片技术，可以算是一种图片加载优化技术。

如下：

无论是小屏幕的手机，还是大屏幕的桌面设备，默认 `img` 链接的图片都是小图（如下）。



然后页面载入完毕后，如果显示器宽度超过一个限制，则 `src` 地址切换成大图地址。

其中，有个比较关键的就是，如果显示器宽度大于某个值，小图的高宽尺寸是要被放大的，也就是说小图会被拉伸显示。这种尺寸的控制借助于CSS3：

```

@media screen and (min-width: 50em) {
    .photo {
        width: 500px;
        height: 375px;
    }
}

```

于是乎，我们会看到，页面载入后的模糊图片：

但是，这种模糊效果是暂时的，因为页面loaded完毕后，我们会根据宽度范围再次修改 `img` 的 `src` 地址，而载入清晰图片。



我记得以前百度图片浏览就是这种效果滴。

这种图片响应实现的好处在于提高了页面的性能，虽然看上去好像比值加载大尺寸图片多了差不多20K。

根据作者的示例，大图的地址使用HTML5 `data-` 自定义属性藏匿：

```

```

六、参考文章

- [Responsive Design Images](#)
- [Responsive Images](#)
- [Responsive images using cookies](#)
- [Mairead: Creating responsive images using the noscript tag](#)
- [Responsive Images: How they Almost Worked and What We Need](#)
- [Image-y nation](#)

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« 您可能不知道的CSS元素隐藏“失效”以其妙用

了解: 使用CSS namespace进行分隔 »

猜你喜欢

- [响应式图片srcset全新释义sizes属性w描述符](#)

- 纯js无图片无外链CSS图片幻灯片播放插件
- 应运而生的web页面响应布局
- 翻译: web响应设计, 乏味!
- 伪类+js实现CSS3 media queries跨界准确判断
- 小tip: base64:URL背景图片与web页面性能优化
- 梳理: 提高前端性能方面的处理以及不足
- jQuery-马化腾产品设计与用户体验的一些技术实现
- 关于Google圆角高光高宽自适应按钮及其拓展
- 我熟知的三种三栏网页宽度自适应布局方法
- 回流与重绘: CSS性能让JavaScript变慢?

分享到:       0

标签: cookie, javascript, media queries, noscript, 优化, 前端性能, 响应图片, 响应布局, 图片展示, 显示器

发表评论 (目前5条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. D说道:

2015年07月30日 11:14

好强大的功能

回复



2. 竞华说道:

2014年08月13日 10:15

自学真困难, 感觉能者都是语焉不详, 看不明白。

回复



3. 子曰说道:

2014年04月1日 23:52

这个好强大, 赶脚应该不叫响应图片 可以叫响应媒体或超文本 应该这方法还可以针对视频及其它非文本内容的响应优化吧

回复



4. Herrington Darkholme说道：

2014年01月19日 00:41

学习了！

[回复](#)



5. 我需要呢说道：

2013年11月8日 21:12

我需要这个啊

[回复](#)



最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁹³⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹²⁰⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹⁵⁾
- » [HTML5终极备忘大全（图片版+文字版）](#) ⁽⁹³⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸⁶⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸²⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁸⁰⁾
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) ⁽⁷⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁶⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷⁶⁾

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) ⁽⁷⁶⁾
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) ⁽⁶⁴⁾
- » [看，for..in和for..of在那里吵架！](#) ⁽⁶⁰⁾
- » [是时候好好安利下LuLu UI框架了！](#) ⁽⁴⁷⁾
- » [原来浏览器原生支持JS Base64编码解码](#) ⁽³⁵⁾
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) ⁽³³⁾
- » [炫酷H5中序列图片视频化播放的高性能实现](#) ⁽³¹⁾
- » [CSS scroll-behavior和JS scrollToView让页面滚动平滑](#) ⁽³⁰⁾
- » [windows系统下批量删除OS X系统.DS_Store文件](#) ⁽²⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽²⁶⁾

猜你喜欢

- [响应式图片srcset全新释义sizes属性w描述符](#)
- [纯js无图片无外链CSS图片幻灯片播放插件](#)
- [应运而生的web页面响应布局](#)
- [翻译：web响应设计，乏味！](#)
- [伪类+js实现CSS3 media queries跨界准确判断](#)
- [小tip: base64:URL背景图片与web页面性能优化](#)
- [梳理：提高前端性能方面的处理以及不足](#)
- [jQuery-马化腾产品设计与用户体验的一些技术实现](#)
- [关于Google圆角高光高宽自适应按钮及其拓展](#)

- 我熟知的三种三栏网页宽度自适应布局方法
- 回流与重绘：CSS性能让JavaScript变慢？