

# 激动人心的HTML5之美

■ 文 / 蒋宇捷

从HTML4诞生以来，整个互联网环境、硬件环境都发生了翻天覆地的变化，开发者期望标准统一、用户渴望更好体验的呼声越来越高。二十年磨一剑，HTML5作为下一代Web标准，它的新特性正在每个新式浏览器的版本中快速地得到支持和体现。随着HTML5和CSS3的发展和完善，Web应用程序正在逐渐表现出桌面应用的性能和功能，和桌面应用之间的区别将越来越模糊。未来不能用HTML5来实现的功能将越来越少，未来绘图、视频编辑、3D建模等也将从单机软件转到浏览器上通过Web应用程序来实现。

下文我将结合实例，谈谈激动人心的HTML5之美。

## 语义之美

语义化很美，在合适的地方使用合适的标签，把人和机器一视同仁。书写语义化的页面就像建造符合工业标准的建筑，阅读语义化的页面像阅读一本你熟悉的优美著作。

HTML5的语义化标签包括：

<section>-代表文档中的一段或者一节；

<nav>-用于构建导航；

<article>-表示文档、页面、应用程序或网站中一体化的内容；

<aside>-代表与页面内容相关、有别于主要内容的部分；

<hgroup>-代表段或者节的标题；

<header>-页面的页眉；

<footer>-页面的页脚；

<time>-表示日期和时间；

<mark>-文档中需要突出的文字。

和采用css+div进行页面布局的方式相比，这些新标签明确的表现了页面元素的结构和含义。下面的代码展示了这样的示例：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf8" >
  <title>HTML5</title>
  <link rel="stylesheet" href="html5.
css" >
</head>
<header>
  <h1>Title</h1>
  <h2>Subtitle</h2>
</header>
<aside>
  <nav>
    <h2>Nav1</h2>
    <ul>
      <li>Link1</li>
      <li>Link2</li>
    </ul>
  </nav>
  <nav>
    <h2>Nav2</h2>
    <ul>
      <li>Link3</li>
      <li>Link4</li>
    </ul>
  </nav>
</aside>
<article>
  <header>
    <hgroup>
      <h1>HTML5 is beautiful</h1>
      <h2>Semantic, Natural, Simple
and Useful</h2>
    </hgroup>
```



## 作者简介：

蒋宇捷，腾讯研究院浏览器前端负责人，曾担任傲游Web后台主管，西安交通大学硕士。HTML5研究小组成员，2008年、2009年Perl大会演讲者。关注Web前后端架构、移动互联网应用和团队管理。



```

</header>
<section>
  <h2>Semantic</h2>
  <p>HTML5 is Semantic.</p>
</section>
<section>
  <h2>Natural</h2>
  <p>HTML5 is Natural.</p>
</section>
</article>
<footer>
  <a href="http://www.w3.org/TR">W3C</a>
</footer>
</html>

```

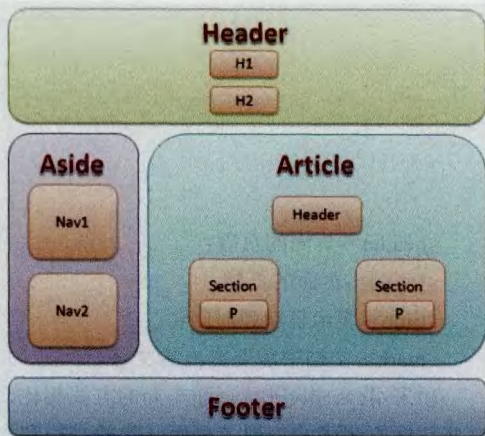


图1 页面元素结构图

相当简单和清晰，我可以不添加任何注释，人和机器都明白它所要构建的结构和内容。

目前主要浏览器的新版本都支持HTML5语义化标签。

### 人性之美

HTML5设计理念中的通用访问体现了人性化之美。通用访问包括可访问性、媒体中立和国际化支持三个概念。可访问性考虑了对残障用户的支持，媒体中立为所有的平台和终端上（例如Android和iPhone的平台上）建立统一标准；而国际化的支持体现在不同的语言和书写习惯上。

就像CSS3中对国际化的支持体现在padding-start, padding-end（在Chrome新版本的设置中心中可以看到它的应用）等新属性一样，HTML5引入了Ruby标签。Ruby标签允许为一个或者多个文本添加附加注释（其命名来源于Ruby字符），例如下面的代码：

```

<p>Hello,
  <ruby>
    <rb>HTML5</rb>

```

```

<rp> (</rp>
  <rt>Hyper Text Mark-up Language
5</rt>
  <rp>) </rp>
</ruby>
</p>

```

显示的文本如下：

Hyper Text Mark-up Language 5  
Hello, HTML5

Ruby标签在中文、韩文和日文教科书和古文中非常有用，读者们可以方便了解文本的详细读音和含义。下面的诗句很美，而HTML5更美。

死生 契 阔，与子成说。执子之手，与子偕老。

有趣的是在IE、Firefox或Opera中复制粘贴上方的文本到编辑器中，会显示下面的文本，这是因为<rp>标签的特殊作用：允许在不支持Ruby标签的浏览器中优雅的降级，同时支持非格式化的复制和粘贴。

死生 契 (读音：qì) 阔 (读音：kuò)，与子成说。执子之手，与子偕老。

目前主要浏览器的新版本对于Ruby标签的支持程度如下：除Firefox和Opera外均提供支持。

### 简单之美

大道至简，化繁为简是许多标准、规范、框架的终极目标。HTML5很好地诠释了这一点，例如以浏览器原生能力支持和代替复杂的Javascript代码、提供简单而强大的新API。

HTML5自身即跨浏览器的JS库，使得开发者的工作更容易：例如新的表单标签和属性、内置拖拽事件使得完全不需要Javascript的支持就能实现许多常用的功能。

我们再来看HTML5表单，它实现了许多简单易用的原生属性和控件，可以轻松的构建强大的表单，帮我们节约大量的Javascript代码。下面是一个HTML5表单的示例：

```

<form>
  <p>
    <label for="email">电子邮箱: </label>
    <input type="email" required autofocus name="email" id="email" placeholder="您的电子邮箱">
  </p>
  <p>
    <label for="username">用户名: </label>

```



```

<input type="text" pattern
="^\\w{6,12}$" required name="username"
placeholder="6到12位的英文字符">
<input type="search" placeholder="
用户名搜索" autosave="www.yujie.com"
results="5" name="username-search"
id="username-search">
</p>
<p>
<label for="username-search">生日: </label>
<input type="date" min="1980-
01-01" max="2011-3-16" name="birthday"
id="birthday" value="1982-10-10">
</p>
<p>
<label for="blog">博客地址: </
label>
<input type="url" name="blog"
placeholder="您的博客地址" id="blog">
</p>
<p>
<label for="mobile">手机: </label>
<input type="number" name="mobile"
pattern="^[0-9]{10}$" id="mobile"
placeholder="您的手机号">
</p>
<p>
<label id="label-working-year"
for="working-year">工作年限: </label>
<input type="range" min="1"
step="1" max="20" name="slider"
name="working-year" id="working-year"
placeholder="您的工作年限" value="3">
</p>
<p>
<label for="age">年龄: </label>
<input type="number" name="age"
id="age" value="20" autocomplete="off"
placeholder="您的年龄">
</p>
<p>
<label for="avatar">头像: </
label>
<input type="image" src="user.
png" name="avatar" id="avatar"
placeholder="点击选择头像">
</p>
</form>

```

图2 HTML5表单

示例中包含了email、search、date、url、range、number、image类型的输入框和required、autofocus、placeholder、pattern、autosave、results、min、max、step、autocomplete等新属性，它们简单到根据名字就可以揣测出用途。效果如图2所示。

HTML5表单在各浏览器里的支持程度和表现并不一致，例如Search输入框目前只被Webkit内核浏览器支持，而日期弹出框和拖动条刻度仅被Opera支持。以上的标签类型和属性目前没有一个浏览器完美支持。

## 实用之美

HTML5的Web Worker、Web Socket、Web Storage等新API让很多后台的工作可以放到前端来处理，Web Worker解决Javascript单线程和阻塞的问题，相当于提供了分布式处理的框架；Web Socket提供了全双工的长连接通信，利用它，我们可以实现微博消息推送、新邮件推送、实时游戏和聊天，减少了不必要的数据传输，提高了信息的实时性；Web Storage相当于前端的Memcached和数据库。

而HTML5的Canvas是最强大的API之一，可以动态生成图形、图像和动画，在HTML5游戏中使用的非常普遍。在下面的例子中，我将展示HTML5 Canvas之美：实现图像颜色渐变效果。

以前，我们在网站上显示灰度图像到彩色图像的渐变动画有两种实现方式：1、基于IE滤镜的方案，缺点是无法实现浏览器兼容；2、手动创建彩色图像的灰度版本。现在，利用HTML5的Canvas画布，我们可以高效简单的实现此动画效果。



图3 利用canvas实现动画效果

将鼠标放在左边的图片上，图片颜色将从灰度渐变到彩色，一切就在眼前鲜活起来。

核心的Javascript代码如下：



```

// 加载时就进行处理
$(window).load(function(){
    var img = $('#color-img');
    // 复制图像
    img.clone().addClass('gray-
img').css({ "position": "absolute",
    "z-index": "2", "opacity": "0" }).
insertBefore(img);
    img.attr('src', grayscale(img.
attr('src')));
    // 图像的淡入
    $('#color-img').mouseover(function()
    {
        $(this).stop().animate({opacity:
1},1000);
    })
    // 图像的淡出
    $('#gray-img').mouseout(function(){
    $(this).stop().animate({opacity: 0},
1000);});
    });
    // 创建灰度版的图像
    function grayscale(src) {
        // 取得canvas元素及其绘图上下文
        var canvas = document
createElement('canvas');
        var ctx = canvas.getContext('2d');
        var imgObj = new Image();
        imgObj.src = src;
        canvas.width = imgObj.width;
        canvas.height = imgObj.height;
        ctx.drawImage(imgObj, 0, 0);
        // 绘制一副图像
        var imgPixels = ctx.getImageData(0, 0,
canvas.width, canvas.height);
        // 获取之前的数据
        for(var x = 0; x < imgPixels.height;
x++){
            for(var y = 0; y < imgPixels.
width; y++){
                var i = (x * 4) * imgPixels.width + y
* 4; var avg = (imgPixels.data[i] +
imgPixels.data[i + 1] + imgPixels.data[i
+ 2]) / 3;
                // 计算灰度值
                imgPixels.data[i] = avg;
                // rgb中的r
                imgPixels.data[i + 1] = avg;
                // rgb中的g
                imgPixels.data[i + 2] = avg;
                // rgb中的b
                // i + 3是alpha通道,我们现在不需要
            }
            ctx.putImageData(imgPixels, 0, 0, 0,
0, imgPixels.width, imgPixels.height);
            return canvas.toDataURL();
        }
    }
}

```

针对不支持的旧浏览器,我们可以使用Modernizr Javascript库或者原生Javascript检测当前浏览器是否支持,并提供替代性的解决方案:

```

if(!Modernizr.canvas) { // 或者使用
!document.createElement('canvas').
getContext
    $(document).ready(backupFunc);
}

```

目前主要浏览器的新版本都提供Canvas标

签的支持。

## HTML5的不足

目前HTML5还有哪些不足之处?

**安全:** Firefox 4的WebSocket和透明代理的实现存在严重安全问题,同时WebStorage、Web Socket这样的功能很容易被黑客利用,来盗取用户的信息和资料,另外HTTP的机制导致了Web应用安全性有所欠缺,这将在很长的时间内成为问题。

**性能:** 某些平台上的引擎问题导致HTML5性能低下。同时在不加入GPU加速的情况下,HTML5处理复杂音视频、动画的性能不尽如人意。

**完善性:** HTML5还在成熟和发展中,像之前曾经支持的Web SQL Database, W3C已经决定不再维护。各浏览器对许多特性的支持程度也不一样,而地理定位API在国内的前途还不清晰。

**技术门槛:** HTML5简化开发者工作的同时代表了有许多新的属性和API需要开发者学习,像Web Worker、Web Socket、Web Storage等新特性要求对于后台的技术需要有一定的了解,甚至需要深入了解其后面原理和逻辑,而Canvas要求视觉和图像算法的一些知识。传统的前端开发者需要掌握更多算法、视觉、后台甚至浏览器原理的知识,机遇的同时也是巨大挑战。

## 总结

上面的大多数不足是HTML5发展过程的中间状态导致的,从长期来看,浏览器厂商的支持、移动互联网的蓬勃发展使得HTML5的前景一片光明,开发者对此必须要有清楚的认识和长远的目光,提前做好技术储备。

对于HTML5,我认为不仅要了解技术细节,还要从根本上理解HTML5的架构,从更高的层面上理解它的深层次含义,它的设计理念,你会发现它的架构之美,对于个人架构设计能力和境界提升有很大帮助。

Web应用程序和移动互联网是大势所趋,IE6终将消亡,Web和浏览器标准终将统一,HTML5就是道之所藏、美之所在,让我们一起迎接和拥抱HTML5的到来! 