

了解woff2字体及转换

这篇文章发布于 2018年07月25日, 星期三, 00:51, 归类于 [CSS相关](#)。阅读 7875 次, 今日 16 次 3 条评论

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=5474>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

一、woff2字体概览



woff字体在2012年12月被World Wide Web Consortium (W3C)推荐使用, IE9+浏览器支持。

woff2字体最早在2013年7月Chrome Canary版本上可以使用, 发展到现在, 几乎已经成为自定义图标字体使用的标配, 目前浏览器的兼容性已经相当不错了

:

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *
	14+支持		49		10.3
		60	66		11.2
11	17	61	67	11.1	11.4
	18	62	68	12	12
		63	69	TP	
			70		

基本上, 在web应用中, ttf还有svg这种字体就没有出现的必要了。

woff2字体最大的优先在于体积传输小, 借用Google Chrome官方话说:

新的WOFF 2.0 Web 字体压缩格式平均要比WOFF 1.0小30%以上 (某些情况可以达到50%+)

下面是一张woff vs woff2字体大小对比图:

Name	
icons.woff	5 KB
icons.woff2	4 KB
proxima-nova-bold.woff	25 KB
proxima-nova-bold.woff2	18 KB
proxima-nova-light.woff	24 KB
proxima-nova-light.woff2	18 KB
proxima-nova-regular.woff	29 KB
proxima-nova-regular.woff2	18 KB
proxima-nova-semibold.woff	25 KB
proxima-nova-semibold.woff2	19 KB
vjs.woff	4 KB
vjs.woff2	2 KB

二、TTF转WOFF2

Node.js版本

可以参考这个Github项目：<https://github.com/nfroidure/ttf2woff2>

应该是在google官方woff2项目外面包了一层。

使用示意如下：

```
var fs = require('fs');
var ttf2woff2 = require('ttf2woff2');

var input = fs.readFileSync('font.ttf');

fs.writeFileSync('font.woff2', ttf2woff2(input));
```

在线TTF转woff2

下面2个地址都可以在线转换：

- <http://www.fontsquirrel.com/tools/webfont-generator>
- <http://everythingfonts.com/ttf-to-woff2>

我个人都是使用fontsquirrel这个网站的，很多年前我就推荐过，简单三步曲，1. 上传；2. 确认版权；3. 下载。今天还试用了下，很不错，连 `@font-face` 相关CSS代码都一并生成了。

会生成woff和woff2两个字体，如果你的项目无需兼容IE8浏览器，足够了。

对了WOFF2字体貌似是可以还原成TTF字体的，

因为我在google官方项目中看到了下面的示意代码：

```
woff2_compress myfont.ttf
woff2_decompress myfont.woff2
```

很显然，下面一行是“反压缩”，多半就是还原字体的意思，不过具体如何运行我并没有亲自尝试，如果谁有经验，欢迎分享下，我及时补充。

三、WOFF2在CSS中的使用

如下，由于IE13-和部分国产Android浏览器不支持，因此还需要WOFF兜底：

```
@font-face {
  font-family: MyFont;
  src: url('myfont.woff2') format('woff2'),
       url('myfont.woff') format('woff');
}
```

如果你的字体体积不是很大，也可以直接base64内嵌在CSS中，加载体验要比外链字体要好一些，即便你使用了新的CSS3 `font-display` 属性。

语法是类似的：

```
@font-face {
  font-family: MyFont;
  src: url('data:font/woff2;base64,...') format('woff2'),
       url('data:font/woff;base64,...') format('woff');
}
```

如何转base64 encoded字符串呢?

这里有我自己弄的一个任意文件转base64的页面: </sp/base64.html>。任意文件拖进去都可以转成base64,当然也包括woff2文件。

不过,根据我的测试,woff2字体转换后没有显示mime type值,直接就是 `data;base64,...`,我复制到CSS代码中一测试,嘿,效果完全支持。当然,如果你不放心,可以手动加个 `font/woff2`。

四、其他需要知道的知识

- WOFF字体没有必要再开启GZIP,因为这个字体文本本身就是压缩过的。
- WOFF 2.0的mime type值,怎么说呢,有些许小争议。Google使用font/woff2,而W3C则推荐application/font-woff2。我个人的建议是这样的:在CSS base64字体表示时候使用Google的font/woff2,毕竟是web呈现;然后服务器配置走application/font-woff2。

NGINX: WOFF2 mime type

```
types {
    application/font-woff2  woff2;
}
```

APACHE: WOFF2 mime type

```
AddType application/font-woff2 .woff2
```

IIS: WOFF2 mime type

然后,根据我的测试,你设置WOFF2 mime type为application/woff2文件也能正常识别。

五、结语

拥抱WOFF2,无论是字体图标,还是单纯显示具体设计感的文字。

参考文章: [Basics about WOFF 2.0](#)

《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](#)。



猜你喜欢

- 等宽字体在web布局中应用以及CSS3 ch单位嘿嘿
- 真正了解CSS3背景下的@font face规则
- 小tip: base64:URL背景图片与web页面性能优化
- 致设计师: 图标图形制作-路径为王
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- 原来浏览器原生支持JS Base64编码解码
- 学习了, CSS中内联SVG图片有比Base64更好的形式
- 页面可用性之浏览器默认字体与CSS中文字体
- 翻译: CSS中的糟粕
- CSS font关键字属性值的简单研究
- 视网膜New iPad与普通分辨率iPad页面的兼容处理

分享到:        0

标签: base64, font, font-face, mime, woff, woff2

发表评论 (目前3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. 夏樹说道:

2018年11月5日 10:57

第一个网址转换后, 去除了中文

回复



2. Chauncywu说道:

2018年10月12日 10:11

使用了下fontsquirl的转换woff, 貌似把中文的全部去除了, 原来14M的中文字体压缩到30k, 如果不需要中文的话fontsquirl还是很棒的, 刚刚找了个可以在线转换中文的:

<https://www.fontke.com/tool/convfont/>

14M字体文件压缩后有6M, 还是挺可观的。

回复



3. Nfer说道:

2018年08月16日 17:45



新的WOFF 2.0 Web 字体压缩格式平均要比WOFF 1.0小30%以上 (某些情况可以达到50%+)

我在一个项目woff2的大小比woff小了不止50%

```
$ du -sh src/styles/font/Roboto-*  
160K src/styles/font/Roboto-Light.ttf  
68K src/styles/font/Roboto-Light.woff  
12K src/styles/font/Roboto-Light.woff2  
156K src/styles/font/Roboto-Regular.ttf  
76K src/styles/font/Roboto-Regular.woff  
12K src/styles/font/Roboto-Regular.woff2
```

[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](#) ⁽¹⁷⁸⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹²⁾
- » [未来必热: SVG Sprite技术介绍](#) ⁽¹¹¹⁾
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) ⁽⁸⁵⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸³⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸⁰⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁸⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷²⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁶⁹⁾
- » [分享三个纯CSS实现26个英文字母的案例](#) ⁽⁶⁹⁾

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 scrollIntoView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

- [等宽字体在web布局中应用以及CSS3 ch单位嘿嘿](#)
- [真正了解CSS3背景下的@font face规则](#)
- [小tip: base64:URL背景图片与web页面性能优化](#)
- [致设计师: 图标图形制作-路径为王](#)
- [纯前端实现可传图可字幕台词定制的GIF表情生成器](#)
- [原来浏览器原生支持JS Base64编码解码](#)
- [学习了, CSS中内联SVG图片有比Base64更好的形式](#)

- 页面可用性之浏览器默认字体与CSS中文字体
- 翻译：CSS中的糟粕
- CSS font关键字属性值的简单研究
- 视网膜New iPad与普通分辨率iPad页面的兼容处理