

小tips: 使用JS检测用户是否安装某font-family字体

这篇文章发布于 2018年02月24日, 星期六, 00:08, 归类于 JS实例。阅读 12234 次, 今日 5 次 12 条评论

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7387>

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、使用JS判断用户操作系统是否安装某字体

下午突发灵感, 写了一段JavaScript小脚本, 可以用来判断用户的操作系统是否安装了某字体, 代码如下:

```
var isSupportFontFamily=function(f){if(typeof f!="string"){return false}var h="Arial";if(f.toLowerCase()==h.toLowerCase()){return true}var e="a";var d=100;var a=100,i=100;var c=document.createElement("canvas");var b=c.getContext("2d");c.width=a;c.height=i;b.textAlign="center";b.fillStyle="black";b.textBaseline="middle";var g=function(j){b.clearRect(0,0,a,i);b.font=d+"px "+j+", "+h;b.fillText(e,a/2,i/2);var k=b.getImageData(0,0,a,i).data;return [].slice.call(k).filter(function(l){return l!=0})};return g(h).join("")!==g(f).join("")};
```

语法

```
isSupportFontFamily(fontFamily);
```

其中 `fontFamily` 参数是必须的, 为CSS中 `font-family` 设置的web可识别的字体名称, 例如宋体 `'simsun'`, 微软雅黑 `'Microsoft Yahei'` 等。

例如, 我们要判断用户的操作系统是否安装了微软雅黑字体, 可以这么处理:

```
// isSupportMicrosoftYahei为true或者false
var isSupportMicrosoftYahei = isSupportFontFamily('Microsoft Yahei');
```

如果用户的操作系统安装了微软雅黑, 则 `isSupportMicrosoftYahei` 为 `true`, 否则为 `false`。

由于本检测方法本质上是基于web检测, 因此, `fontFamily` 参数必须是web可识别的, 因此, 很多中文字体需要使用其英文名称如“思源黑体”, “兰亭黑体”等, 具体名称是什么可以参见我之前的文章: [“CSS font-family常见中文字体对应的英文名称”](#)。

字体中文名	字体英文名
思源黑体	Source Han Sans CN
思源宋体	Source Han Serif SC
文泉驿微米黑	WenQuanYi Micro Hei

测试demo

为了演示上面JS脚本的准确性, 我特意制作了一个实例页面, 您可以狠狠的点击这里: [一些中文字体系统是否安装检测demo](#)

例如，在我的windows操作系统上，显示效果如下截图所示：



另外，demo页面源代码有未压缩的JavaScript脚本。

实现原理

根据用户设置的字体将某一个字符绘制在 `canvas` 上（`fillText()`），并提取像素信息（`getImageData()`），然后和默认字体进行比对，如果像素不一致，说明字体生效，说明字体不生效。

兼容性

IE9+，以及其他所有现代浏览器。

二、使用JS判断网页是否支持某字体渲染实例

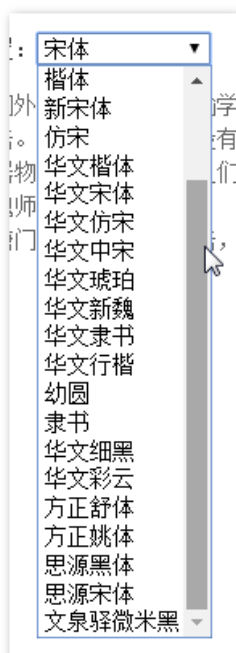
什么场景下需要判断用户的浏览器是否支持某个字体的渲染呢？

例如一些阅读类的场景，希望用户可以自己选择自己喜欢的字体进行设置。对于photoshop，office，keynote等软件，都是直接系统搜刮字体，形成完整而准确的字体下拉。但是web页面并没有这样的权限，因此我们只能曲线救国。例如默认给定50个字体范围列表，遍历检测用户支持那些字体，支持哪个字体就增加哪个下拉选项，于是我们的字体配置选项就大大丰富了，体验也就更进一步了。



您可以狠狠地点击这里：[选择合适的字体进行阅读demo](#)

例如，在我自己windows台式机上，显示了如下图所示的字体列表，这些字体全部都是可以正常渲染的：



例如，我选择一个华文彩云，结果如下图：



三、结束语

经过自己的简单测试，在移动端上效果也是可以的，Android未测。由于此方法是自己突然想到的，目前尚未在实际项目中进行过实践，故并不能保证百分百准确，之后会进一步及时更新信息，也欢迎反馈各类使用问题。

好，就这些，希望本文的内容能够对您的工作学习有所帮助。

感谢阅读！

《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](#)。



« 解决canvas图片getImageData,toDataURL跨域问题

CSS, SVG和canvas分别实现文本文字纹理叠加效果 »

猜你喜欢

- canvas getImageData与任意字符图形点、线动效实现
- canvas文本绘制自动换行、字间距、竖排等实现
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- CSS, SVG和canvas分别实现文本文字纹理叠加效果
- 页面可用性之浏览器默认字体与CSS中文字体
- 翻译 - CSS高峰会议内容精选
- Google字体API使用简单示例
- CSS font关键字属性值的简单研究
- CSS unicode-range特定字符使用font-face自定义字体
- CSS font-family常见中文字体对应的英文名称
- canvas图形绘制之星空、噪点与烟雾效果

分享到: 1

标签: canvas, fillText, getImageData, 中文字体, 字体

发表评论 (目前12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. **Mr.M**说道:
2018年12月6日 09:26
思路很巧妙, 赞
回复



2.

stanJ说道：

2018年04月3日 12:15

android测试没效果

回复


3.

Web前端之家说道：

2018年02月26日 18:28

字体是一门艺术。。。

回复


4.

lonhon说道：

2018年02月26日 16:48

妙

格式化后断点跑了一下，思路清奇。

想到之前看到的用canvas获取GPU型号判断手机型号。

回复


5.

王念一说道：

2018年02月25日 11:18

蛤。。。我还以为是用 RenderingContext2d 的原型方法 measureText() 判断字符宽==

回复

etherdream说道：

2018年03月25日 23:00

measureText 是行不通的，有些字体和未知字体宽度一样，只有高度不同。而测量高度并不是所有浏览器都支持。之前研究 JS 指纹时就遇到这问题。（大部分 JS 指纹库都有枚举字体的功能，比如：fingerprintjs2 — <https://github.com/Valve/fingerprintjs2>）

楼主这个办法虽然可行，但是效率不是最高的，每检测一个字体要一次图形渲染。目前试过最快的办法，是创建一个元素，内容设置几个字符（选择不同字体下宽度差距较大的字符。大家可以自己去试），然后设置各种 font-family，用 offsetWidth/offsetHeight 量出尺寸。这样可以共用一个元素，性能相对较高。

当然不管什么浏览器、操作系统，加载没有缓存的字体是比较费时间的，一般枚举几百个字体都得费大几百 ms。有些浏览器比如 safari 甚至还会假死。。。

回复




6.

ielgnaw说道：

2018年02月24日 17:05

好点子！

回复


7.

心满说道：

2018年02月24日 16:44

妙啊

回复


8.

依韵说道：

2018年02月24日 15:37

把格式化出来学习了下，好奇妙想法啊！

把确定支持的字体和待确认字体以相同的大小都画在canvas上，最后如果画出来的不一样 那就是系统认识这个字体！也就是支持！

真有创造力！



[回复](#)

9. **aaa**说道:

2018年02月24日 09:29

这个真牛啊。但是我还要多看几遍原理。

[回复](#)



mfk说道:

2018年02月27日 17:04

原理是：用一个字体显示字母a。如果这个字体不存在，系统会默认显示成Arial字体。
如果用字体X显示出来的字母的字符点阵和字体Arial不一样，说明字体X存在。
如果用字体X显示出来的字母的字符点阵和字体Arial一样，说明字体X不存在，系统用的默认字体Arial显示的。

[回复](#)



whj说道:

2018年03月8日 13:03

多谢解惑，看了几遍才看明白，这个思路真不错

[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 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

- [canvas getImageData与任意字符图形点、线动效实现](#)
- [canvas文本绘制自动换行、字间距、竖排等实现](#)
- [纯前端实现可传图可字幕台词定制的GIF表情生成器](#)
- [CSS, SVG和canvas分别实现文本文字纹理叠加效果](#)
- [页面可用性之浏览器默认字体与CSS中文字体](#)
- [翻译 - CSS高峰会议内容精选](#)
- [Google字体API使用简单示例](#)
- [CSS font关键字属性值的简单研究](#)
- [CSS unicode-range特定字符使用font-face自定义字体](#)
- [CSS font-family常见中文字体对应的英文名称](#)
- [canvas图形绘制之星空、噪点与烟雾效果](#)