

小tips: 纯前端JS读取与解析本地文本类文件

这篇文章发布于 2018年03月27日, 星期二, 23:02, 归类于 [JS实例](#)。阅读 15618 次, 今日 5 次 [9 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7463>

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、何为文本类文件?

所谓文本类文件, 指MIME Type为 `text/*` 文件, 例如, CSS文件 (`text/css`), JS文件 (`text/javascript`), HTML文件 (`text/html`), txt文本 (`text/plain`) 等等。

在现代浏览器下, 我们可以直接对这些文件进行处理, 无论是通过 `type=file` 表单元素还是从桌面拖拽到网页中都是可以的。

二、纯前端JS读取与解析本地文本类文件

已知file就是文件对象, 则核心CSS如下:

```
var reader = new FileReader();
reader.onload = function (event) {
    // event.target.result就是文件文本内容
    // 然后你就可以为所欲为了
    // 例如如果是JSON数据可以解析
    // 如果是HTML数据, 可以直接插入到页面中
    // 甚至字幕文件, 各种滤镜, 自定义文件格式, 都可以玩弄于鼓掌之间.....
};
reader.readAsText(file);
```

核心就是 `readAsText()` 这个方法, 我们可能有用过的前端预览本地图片, 则用的是 `FileReader.readAsDataURL()` 方法, 一个转文本, 一个转base64, 性质都是一样的。

目前 `file` 对象通常有下面几种方式获取:

1. `type=file` 表单元素, 假设DOM元素是 `eleFile`, 则 `file` 对象 (假设非多选模式) 为 `eleFile.files[0]`。或者也可以在 `change` 事件中获取, 例如:

```
eleFile.onchange = function (event) {
    var file = event.target.files[0];
};
```

2. 拖拽事件获取。例如:

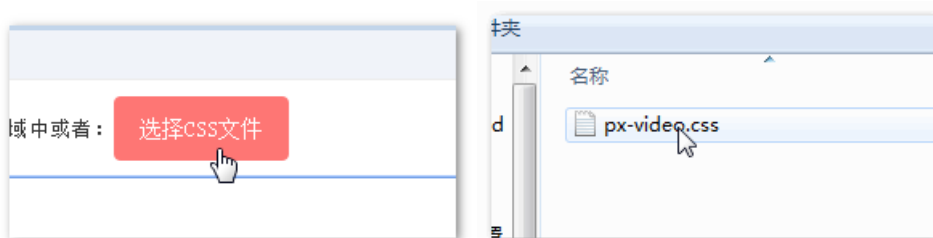
```
this.dragDrop.addEventListener("drop", function(event) {
    var file = event.dataTransfer.files[0];
}, false);
```

三、实践-实现一个纯前端的简易CSS压缩工具

您可以狠狠地点击这里: [CSS在线压缩工具demo](#)

此工具差不多10年了，然后趁此机会加个直接上传CSS压缩功能。操作演示如下：

1. 选择CSS文件，此时会自动触发“开始处理”按钮；



2. 下载CSS文件：



这里的下载功能也是纯前端实现的，有兴趣可以参见之前的文章：[“JS前端创建html或json文件并浏览器导出下载”](#)。

相关代码直接右键->查看页面源代码即可！

四、结束语

`FileReader` 对象除了有 `readAsText()` 和 `readAsDataURL()` 外，还有下面2个标准方法，分别是：`FileReader.abort()` 和 `FileReader.readAsArrayBuffer()`。

`FileReader.abort()` 可以终止读操作。

`FileReader.readAsArrayBuffer()` 作用是把Blob或者file对象以 `ArrayBuffer` 形式读出来。因此，可以罗列下：

- `readAsText()` -> 文本字符数据
- `readAsDataURL()` -> Base64 URL数据
- `readAsArrayBuffer()` -> ArrayBuffer数据

对于非文本类文件，`readAsText()` 方法也是可以用的，但是读出来的东西……呵呵，怕是用不起来，例如我选一个 `.docx` word文档。结果出来的东西鬼都不认识：

所以，标题中特别标明了文本类文件。

以上~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](#)。



« [AMCSS \(CSS属性模式\) 开发简介](#)

[简单了解ES6/ES2015 Symbol\(\) 方法](#) »

猜你喜欢

- [HTML5 file API加canvas实现图片前端JS压缩并上传](#)
- [纯前端实现可传图可字幕台词定制的GIF表情生成器](#)
- [基于HTML5的可预览多图片Ajax上传](#)
- [原来浏览器原生支持JS Base64编码解码](#)
- [基于HTML5 drag/drop模块拖动插入排序删除完整实例](#)
- [拖拽献祭中的黑山羊-DataTransfer对象](#)
- [直接剪切板粘贴上传图片的前端JS实现](#)
- [理解DOMString、Document、FormData、Blob、File、ArrayBuffer数据类型](#)
- [XMLHttpRequest实现HTTP协议下文件上传断点续传](#)
- [小tip: base64:URL背景图片与web页面性能优化](#)
- [JS前端创建html或json文件并浏览器导出下载](#)

分享到: 1

标签: [Blob](#), [DataTransfer](#), [File](#), [FileReader](#), [readAsArrayBuffer](#), [readAsDataURL](#), [readAsText](#)

发表评论 (目前9条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

回复

笼子说道：

2018年03月29日 14:48

Google有这功能已经很多年了

回复



liu说道：

2018年04月3日 16:31

文章里说了,word不行 ,word比较特殊

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](#) ⁽¹⁷⁹⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹³⁾
- » [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](#) ⁽⁷³⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁰⁾
- » [分享三个纯CSS实现26个英文字母的案例](#) ⁽⁷⁰⁾

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

- HTML5 file API加canvas实现图片前端JS压缩并上传
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- 基于HTML5的可预览多图片Ajax上传
- 原来浏览器原生支持JS Base64编解码
- 基于HTML5 drag/drop模块拖动插入排序删除完整实例
- 拖拽献祭中的黑山羊-DataTransfer对象
- 直接剪切板粘贴上传图片的前端JS实现
- 理解DOMString、Document、FormData、Blob、File、ArrayBuffer数据类型
- XMLHttpRequest实现HTTP协议下文件上传断点续传
- 小tip: base64:URL背景图片与web页面性能优化
- JS前端创建html或json文件并浏览器导出下载