

纯前端实现可传图可字幕台词定制的GIF表情生成器

这篇文章发布于 2018年05月4日, 星期五, 00:00, 归类于 [Canvas相关](#), [JS实例](#)。阅读 20800 次, 今日 14 次 [10 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7539>

本文可全文转载, 但需要保留原作者和出处。

一、这两天做了个可传图可写台词GIF生成器

这两天晚上在家做了个可以自己传图, 可以自己写台词的GIF生成器。

您可以狠狠的点击这里: [Gif表情自定义字幕在线生成器](#)

可以用提供的模板生成GIF表情, 例如, 我们选择“今天的风甚为喧闹”, 我们可以自己修改台词, 修改GIF这一帧延迟时间。

☒ 使用现有模板

☐ 有钱为所欲为

☐ 王境泽真香

☒ 今天的风甚为喧闹



要来啰, 帅到爆的一句话

延时: 1500 ms



输入对话信息

延时: 2500 ms



今天的风 甚为喧闹啊

延时: 2500 ms

☐ 自己DIY GIF内容

点击生成

生成GIF图片

<http://www.zhangxinxu.com>

流量开销伤不起, 模板图片都是自己主机做图床, 舍不得提供过多图片序列, 因此, 生成GIF的素材掉帧有些严重, 见谅见谅, 感受精髓, 感受精髓。

然后, 生成GIF效果就是下面图这样:



如果嫌弃提供的模板太简陋，可以自己上传序列图片DIY，例如使用[上一篇文章](#)中用到的的图片序列素材：

重新上传图片

或者将图片拖到此处

1.jpg

输入对话信息

延时：42ms 删除

2.jpg

输入对话信息

延时：42ms 删除

3.jpg

输入对话信息

延时：42ms 删除

4.jpg

输入对话信息

延时：42ms 删除

然后填写台词：“给你一个飞吻”“么啊”，然后生成，就有下面质量不错的GIF效果了：



重点是这个工具纯前端生成的

表情生成器工具很多，简单的如静态图生成器，静态图上面台词可以修改；复杂点的动态GIF表情生成器，例如可能有人已经用过的“表情锅”：<https://sorry.xuty.tk/sorry/>



目前已经有好几十个模板了，访问量应该可以，因为我看到guang gao都出来了。

但是小工具是后端图片服务主导实现的，于是就有这样问题：

1. 服务器开销不小，压力比较大；
2. 只能用对方提供的模板生成gif表情，不能使用自己的自定义素材，例如无法实现一个放飞自我的“飞吻”GIF表情；

实际上，纯前端就能搞定GIF生成，资源开销更小，速度更快，更加灵活！

二、如何JS纯前端web生成GIF图片呢？

本GIF表情生成器核心技术源自这个项目：<https://github.com/jnordberg/gif.js>

借助Web Workers, File API和类型数组在浏览器端进行GIF encode，从而生成GIF图片。

语法相当简单，下面代码是官方示例，我简单翻译下：

```
var gif = new GIF({
  workers: 2,
  quality: 10
});

// 添加图片元素
gif.addFrame(imageElement);

// 或者是canvas元素
gif.addFrame(canvasElement, {delay: 200});
```

```
// 或者从canvas的上下文复制像素信息
gif.addFrame(ctx, {copy: true});

gif.on('finished', function(blob) {
  // 这里的blob就是gif图片blob格式信息
  window.open(URL.createObjectURL(blob));
});

gif.render();
```

三、GIF生成器中的字幕台词是怎么加上去的呢？

GIF生成器中的字幕台词是怎么加上去的呢？答案是使用 `canvas` 绘制的。

`canvas`先使用 `drawImage()` 方法绘制底图，然后把输入框中的对话信息绘制在底图上就好了。

拿其中某一帧举例示意：

```
var gif = new GIF({
  width: width,
  height: height
});
// 画布元素
var canvas = document.createElement('canvas');
var context = canvas.getContext('2d');
// 绘制图片
context.drawImage(img, 0, 0);
// 绘制文字
context.fillText(message, width / 2, height - 32);
// 作为帧插入GIF
gif.addFrame(context, {
  delay: delay,
  copy: true
});
```

四、本地上传GIF帧图片如何处理的？

本地上传素材图片和模板提供的类似，只是素材图片预览时候的 `src` 地址是base64地址。

无论是选择上传还是拖拽上传，都可以get到图片这个文件对象，假设变量名是 `file`，则可以借助 `FileReader` 将图片文件转换成可预览模式：

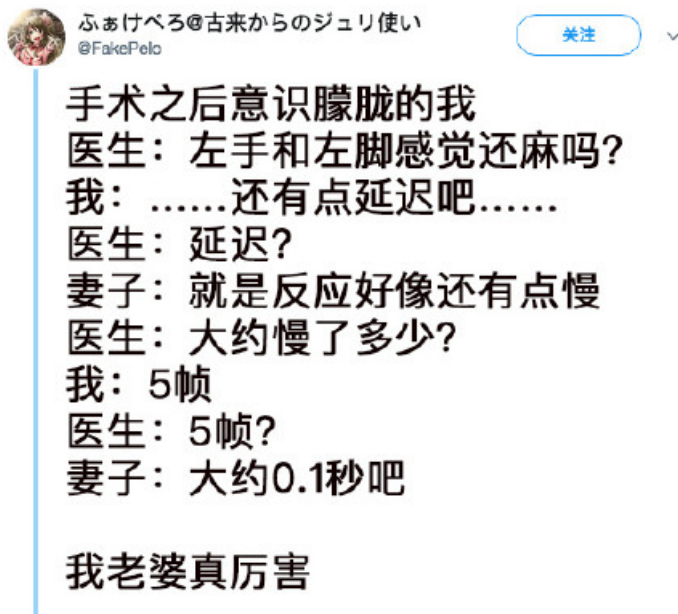
```
var reader = new FileReader();
reader.onload = function(e) {
  // e.target.result就是图片的base64地址信息
  // 让img的src属性值为e.target.result图片即可呈现
}
reader.readAsDataURL(file);
```

接下来的代码逻辑就和模板图片的处理一模一样了。

五、结束语

1. 本工具UI组件由LuLu UI提供支持，LuLu UI —— 你问我有什么不同？只是更自由，更纯粹，和用户走得更近。

2. 欢迎反馈使用中遇到的问题，文章评论，或者[微博私信](#)都可以。
3. 本工具URL地址是：/sp/gif，特别的（special）gif，特别的爱给特别的你~我擦，我居然不自觉唱起来了~
4. 分享看到的一个关于帧和时间挺有意思的推文（日文翻译版）：



按照上图的说法，每帧时间为0.02秒，也就是20毫秒每帧，接近显示器的绘制频率，哈哈哈。对于GIF标签图而言，不需要这么高的帧率，每帧84毫秒用户体验也是可以接受的，频率过高的GIF体积会很大，有的甚至达4~5M，对于web环境，是不太适合的。

以上为本文全部内容，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](#)。



« [炫酷H5中序列图片视频化播放的高性能实现](#)

[windows系统下批量删除OS X系统.DS_Store文件](#) »

猜你喜欢

- HTML5 file API加canvas实现图片前端JS压缩并上传
- 小tips: 纯前端JS读取与解析本地文本类文件
- canvas getImageData与任意字符图形点、线动效实现
- 小tips: 使用canvas在前端实现图片水印合成
- 原来浏览器原生支持JS Base64编码解码
- 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
- canvas文本绘制自动换行、字间距、竖排等实现

- 小tips: 使用JS检测用户是否安装某font-family字体
- iframe和HTML5 blob实现JS,CSS,HTML直接当前页预览

分享到: 1

标签: base64, Blob, canvas, drawImage, File, FileReader, fillText, gif, 工具

发表评论 (目前10条评论)

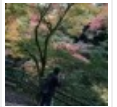
<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<div>提交评论</div>	

1. 小花说道:

2018年05月16日 22:28

在张大大的基础下实现了视频直接截取 <http://www.jackpu.com/qian-duan-zhi-jie-jie-qu-shi-pin-gif-dong-hua>

[回复](#)



张 鑫旭说道:

2018年05月17日 00:23

赞!

[回复](#)



2. 农业微信群大全说道:

2018年05月15日 08:44

css还是皮毛 以后跟着大神学学

[回复](#)

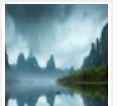


3. mengkun说道:

2018年05月8日 16:23

说道 GIF 加字, 不得不推荐一个相当好用的在线工具: Gifntext (<http://www.gifntext.com/>) 非广告.....

[回复](#)



4. vcxiaohan说道:

2018年05月7日 17:55

我有个问题, 就是一张gif的图片, 在canvas中绘制, 能拿到所有帧的图片吗

[回复](#)



5. 默默学起来说道：

2018年05月7日 15:32

我决定，把大神所有的文章，再看一遍

[回复](#)



6. 低调的小黑衣说道：

2018年05月6日 14:18

太厉害了。什么时候才能像你这么优秀啊！！

[回复](#)



7. 独角兽说道：

2018年05月5日 16:04

发现竟然是昨天更新的，我的大神啊，你竟然坚持到现在还在写。

[回复](#)



8. 鎏金圣手说道：

2018年05月4日 10:52

沙发，晚上学习做一个！

[回复](#)



9. 方小蛋说道：

2018年05月4日 10:35

真棒，马上动手码起来~

[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
- 小tips: 纯前端JS读取与解析本地文本类文件
- canvas getImageData与任意字符图形点、线动效实现
- 小tips:使用canvas在前端实现图片水印合成
- 原来浏览器原生支持JS Base64编码解码
- 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
- canvas文本绘制自动换行、字间距、竖排等实现
- 小tips: 使用JS检测用户是否安装某font-family字体
- iframe和HTML5 blob实现JS,CSS,HTML当前页预览