

SVG精简压缩工具svgo简介和初体验

这篇文章发布于 2016年02月29日, 星期一, 00:42, 归类于 [SVG相关](#)。阅读 38625 次, 今日 29 次 [12 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5278>

补充于**2018-08-19**

周末在家基于SVGO做了个SVG在线压缩合并工具, 您可以狠狠地点击这里: [SVG在线压缩合并工具](#)

支持选择SVG文件上传, 支持拖拽上传, 也支持直接粘贴SVG代码进行压缩。

例如从iconfont.cn网站随便下一个SVG图标, 然后上传下, 还有将近40%的压缩比:



压缩好的SVG我们可以直接下载文件, 也可以直接复制SVG代码, 欢迎使用。

一、事情的起因

佛家讲求因果关系, 种什么因, 得什么果, 比方说种了颗苹果树, 就会结苹果。这里, 扯到SVG的精简压缩也是有原因的, 前年有分享过“[PSD小图标变身SVG Sprites/font-face历险记](#)”, 其中展示了我处理SVG比较重要的一步, 就是对设计师PSD源文件中的图标路径进行重新处理。在没有对设计师进行分享培训之前, 有些细节可能就不会注意, 例如, 图形喜欢用路径叠加实现, 而不是实实在在地勾勒出来; 或者PSD看上去是好的, 但是放大个100倍, 就会发现, 路径的转角和边缘都没对上。此时如果直接SVG, 一是SVG文件大小大, 二是最终的图像可能不是我们想要的。因此, 需要在Adobe Illustrator中重新处理下, 至少我是使用的这个软件。

有小伙伴可能会疑问: 设计师自己挖的坑就让他自己去种苹果树好了, 你佛光普照去帮忙光合作用干什么呢?

佛约: 我不入地狱输入地狱。大家都听过这句话吧, 所以, 我们再看上面的疑问, 怎么样, 是不是听之有理, 实则差矣! 我呸……甩自己一个嘴巴子, “差”它个鬼大头……

- [[cleanupAttrs](#)] 清除换行，结束符以及重复空格
- [[removeDoctype](#)] 删除文档声明
- [[removeXMLProcInst](#)] 删除XML处理指令
- [[removeComments](#)] 删除注释
- [[removeMetadata](#)] 删除 `<metadata>` 源信息
- [[removeTitle](#)] 删除 `<title>` 标题(默认禁用)
- [[removeDesc](#)] 删除 `<desc>` 描述 (默认只有 `desc` 元素无意义的时候)
- [[removeUselessDefs](#)] 删除 `<defs>` 元素如果没有 `id`
- [[removeEditorsNSData](#)] 删除编辑器的命名空间，元素和属性
- [[removeEmptyAttrs](#)] 删除空属性
- [[removeHiddenElems](#)] 删除隐藏元素
- [[removeEmptyText](#)] 删除隐藏文本元素
- [[removeEmptyContainers](#)] 删除空的容器元素
- [[removeViewBox](#)] 如果可以，删除 `viewBox` 属性(默认进行)
- [[cleanUpEnableBackground](#)] 如果可以，删除 `enable-background` 属性
- [[minifyStyles](#)] 使用CSSO最小化元素的 `<style>` 内容
- [[convertStyleToAttrs](#)] 转换样式为属性值
- [[convertColors](#)] 转换颜色(从 `rgb()` 到 `#rrggbb` , 从 `#rrggbb` 到 `#rgb`)
- [[convertPathData](#)] 将路径数据转换为的相对路径和绝对路径中简短的那一个，过滤无用的分隔符，智能四舍五入以及其他很多处理
- [[convertTransform](#)] 合并多个transforms为一个，转换矩阵为短命名，以及其他很多处理
- [[removeUnknownsAndDefaults](#)] 删除未知的元素内容和属性，删除值为默认值的属性/li>
- [[removeNonInheritableGroupAttrs](#)] 删除不可基础组的”presentation”属性
- [[removeUselessStrokeAndFill](#)] 删除无用的 `stroke` 和 `fill` 属性
- [[removeUnusedNS](#)] 删除没有使用的命名空间声明
- [[cleanupIDs](#)] 删除没有使用或者压缩使用的IDs
- [[cleanupNumericValues](#)] 数值四舍五入提高精度，删除默认的’px’单位
- [[moveElemsAttrsToGroup](#)] 移动元素属性们到外面包裹的组元素上
- [[moveGroupAttrsToElems](#)] 移动一些组属性到内容元素上
- [[collapseGroups](#)] 合并无用的组
- [[removeRasterImages](#)] 删除点阵图像(默认禁用)
- [[mergePaths](#)] 合并多个路径为一个
- [[convertShapeToPath](#)] 转换一些基本图形为路径
- [[sortAttrs](#)] 元素属性排序使其像诗歌一样易读(默认禁用)
- [[transformsWithOnePath](#)] 通过里面一条路径实现transforms, 真实宽度剪裁, 垂直居中对齐以及SVG缩放拉伸(默认禁用)
- [[removeDimensions](#)] 如果 `viewBox` 就是当下尺寸限定，删除 `width` / `height` 属性(默认禁用)
- [[removeAttrs](#)] 通过正则删除属性 (默认禁用)
- [[addClassesToSVGElement](#)] 添加类名给外面的 `<svg>` 元素 (默认禁用)
- [[removeStyleElement](#)] 删除元素的 `<style>` (默认禁用)

如何使用？

首先是安装，连我都驾轻就熟了，如下：

```
$ [sudo] npm install -g svgo
```

使用：

```
svgo [OPTIONS] [ARGS]
```

Options:

```
-h, --help : Help 帮助
-v, --version : Version版本
-i INPUT, --input=INPUT : 输入的文件, "-" 为标准输入
-s STRING, --string=STRING : 输入SVG数据字符串
-f FOLDER, --folder=FOLDER : 输入的文件夹, 会优化与重写所有的*.svg文件
-o OUTPUT, --output=OUTPUT : 输入的文件或文件夹 (默认同输入), "-" 标准输出
-p PRECISION, --precision=PRECISION : 设置数字的小数部分, 重写插件参数
--config=CONFIG : 配置文件扩展或替换默认设置
--disable=DISABLE : 根据名字禁用插件
--enable=ENABLE : 根据名字开启插件
--datauri=DATAURI : 输入文件以Data URI字符串形式(base64, URI encoded or unencoded)
-q, --quiet : 仅输出错误信息, 不包括正常状态消息
--pretty : 让SVG漂亮的打印
--show-plugins : 显示可用和存在的插件
```

Arguments:

```
INPUT : 别名 --input
OUTPUT : 别名 --output
```

- 单个文件使用举例:

```
$ svgo test.svg
```

或者:

```
$ svgo test.svg test.min.svg
```

- 使用STDIN / STDOUT(标准输入输出):

```
$ cat test.svg | svgo -i - -o - > test.min.svg
```

- 文件夹举例:

```
$ svgo -f ../path/to/folder/with/svg/files
```

或者:

```
$ svgo -f ../path/to/folder/with/svg/files -o ../path/to/folder/with/svg/output
```

- 使用字符串:

```
$ svgo -s '<svg version="1.1">test</svg>' -o test.min.svg
```

或者甚至是Data URI base64:

```
$ svgo -s 'data:image/svg+xml;base64,...' -o test.min.svg
```

- SVGZ使用:

从 `.svgz` 到 `.svg` :

```
$ gunzip -c test.svgz | svgo -i - -o test.min.svg
```

从.svg到.svgz:

```
$ svgo test.svg -o - | gzip -cfq9 > test.svgz
```

- 带操作界面的GUI – [svgo-gui](#)
- 作为web app – [SVGOMG](#)
- 作为Nodejs模块 – [examples](#)
- 作为Grunt任务 – [grunt-svgmin](#)
- 作为Gulp任务 – [gulp-svgmin](#)
- 作为Mimosa模块 – [mimosa-minify-svg](#)
- 作为OSX文件夹Action – [svgo-osx-folder-action](#)
- 作为webpack loader – [image-webpack-loader](#)

这里出现了一个 `svgz` 格式的文件，这是什么鬼呢？实际上是SVG的gzip压缩文件，Illustrator是可以直接导出的，如下示意：



根据我的测试，Chrome浏览器是可以直接打开的，FireFox浏览器不行（FireFox 44），直接下载，而IE11浏览器是个大白板，一看控制台，报错了：



虽然svgz的压缩比很厉害，是真正意义上的压缩，但是，实际上，应用价值并不大，因为类似JS, CSS字符集类型文本文件，可以通过服务器开启gzip实现传输，SVG也在其中，SO, ...

三、svgo初体验

看上去很厉害，我就先自己试用下，看看是不是用来解决自己的需求的。如下代码截图：



就是所有SVG精简到另外一个平级的名为output的文件夹，于是乎，得到：



从精简效率来看，很惊人，有50%~60%的精简率。

拿其中一个SVG代码（就是svgz示意的钩钩）前后对比下吧：

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 16.0.0, SVG Export Plug-In . SVG Version: 6.00 Build 0)
-->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="图层_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
    width="200px" height="200px" viewBox="0 0 200 200" enable-background="new 0 0 200 200"
    xml:space="preserve">
<path fill="#FFFFFF" d="M147.566,67.575c-3.979-3.24-4.355-3.337-8.9-5.64c-2.043-1.043-5.05
7,1.646-6.529,3.636L92,117.73
    L65.85,83.972c-1.478-1.988-4.205-2.72-6.25-1.782c-4.658,2.408-4.19,2.327-8.168,5.467c-
1.817,1.466-1.932,4.082-0.456,6.065
```

```
c0,0,28.183,36.5,31.592,40.896c5,6.274,14.09,5.965,18.864,0c3.521-4.498,46.59-61.078,46.59-61.078
c149.499,71.55,149.385,68.937,147.566,67.575z"/>
</svg>
```

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="200" viewBox="0 0 200 200"><pa
th fill="#FFF" d="M147.566 67.575c-3.98-3.24-4.355-3.337-8.9-5.64-2.043-1.043-5.057 1.646-
6.53 3.636L92 117.73 65.85 83.972c-1.478-1.988-4.205-2.72-6.25-1.782-4.658 2.408-4.19 2.32
7-8.168 5.467-1.817 1.466-1.932 4.082-.456 6.065 0 0 28.183 36.5 31.592 40.896 5 6.274 14.
09 5.965 18.864 0 3.52-4.498 46.59-61.078 46.59-61.078 1.477-1.99 1.363-4.603-.456-5.965z"
/></svg>
```

恩，有种胖妞变靓妹的即视感：



喔噢，很酷啊，正是自己想要的，而且貌似还可以通过设置让值保留2位小数，又可以进一步精简大小，不过现在这样已经足够了！

好的，不错，这就加入项目的豪华午餐阵营！

恭喜你，svgo, 你得到了评委的一致认可！👍

四、结束语

好像没什么好说的。今天居然是29号，2月29号，4年等一回的日子。还记得上届奥运会似乎就在不久前，时间过得快并不是什么好事，说明每天的都是重复的生活，缺少了值得记忆的东西，不知不觉，像个大人一样了，做事处事，磨砺掉了一般自由随性，还剩下的那点务必要坚守，要从未来的角度看待现在的自己！

哦，突然想起来，MD今天，哦，不对是昨天库里太逆天，最后一个半场绝杀，感动得我眼泪哗哗，大写的服，真是服！

嘛。就这些，晚安~🌙

欢迎交流，欢迎阅读！



《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](#)。



《CSS镂空图片transition过渡初加载背景色块问题解决

腾讯微云黑色遮罩引导蒙版更好的CSS实现方式》

猜你喜欢

- 学习了，CSS中内联SVG图片有比Base64更好的形式
- 网页制作辅助工具-jQuery标尺参考线插件

- style标签下的CSS代码的显示与实时编辑
- 用3D LUT滤镜我做了个在线专业电影级别照片调色工具
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- windows系统下批量删除OS X系统.DS_Store文件
- 高富帅seajs使用示例及spm合并压缩工具露脸
- 小tips: 使用　等空格实现最小成本中文对齐
- 基于active,checked等状态类名的web前端交互开发
- 瞎折腾: 把JS,CSS任意文本文件加密成一张图片
- CSS3/SVG clip-path路径剪裁遮罩属性简介

分享到:         { 0 }

标签: SVG, svgo, svgz, 压缩, 工具

发表评论（目前12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. 一只鱼说道:
2018年12月17日 11:53

大佬

[回复](#)


2. 周大侠啊说道:
2018年09月18日 10:31

单个文件的压缩写错了。
\$ svgo test.svg -o test.min.svg

[回复](#)


3. kaze说道:
2017年05月12日 11:22

遇到了一个蛮奇怪的状况:

 1. 先用 svgo 处理了一堆 svg, 结果体积变小了
 2. 再将处理完的 svg 丢进 icomoon 里, 体积反而比没optimizer增大了...

[回复](#)



4. **arkin**说道：
2017年02月15日 23:29
相问一下博主,像谷哥的这种svg怎么能压缩这么小?是用什么工具做的?
[回复](#)
5. **Lebron**说道：
2016年08月10日 11:59
我喜欢詹姆斯，73胜无冠
[回复](#)
6. **园子大魔王**说道：
2016年03月16日 15:19
自带板凳

[回复](#)
7. **hadaiye**说道：
2016年03月2日 16:09
SVG Cleaner也是个很好的工具，图形化的，直接把SVG文件扔进去，导出来就有了，压缩完成的文件和SVGO出来的是一样的，还有很多可选择的压缩项。。
[回复](#)
- hadaiye**说道：
2016年03月2日 16:12
看过楼主很多博客，跟着学了很多东西，在做SVG文件的时候也参考了楼主的相关文章呢~
[回复](#)


8. **优雅的格调网**说道：
2016年03月2日 15:02
学习经验的
[回复](#)
9. **定世啟**说道：
2016年03月1日 15:24
你很厉害哦，请教javascript:void(0)问题，什么意思？有何用？
[回复](#)
- ca5h**说道：
2016年03月21日 00:49
我不会告诉你在博主以前的篇文章写道了
[回复](#)


10. **张璜**说道：
2016年02月29日 10:28
论库里和svg的联系。
[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 (190)
- » 未来必热: SVG Sprite技术介绍 (119)
- » 粉丝群第1期CSS小测点评与答疑 (115)
- » HTML5终极备忘大全 (图片版+文字版) (93)
- » 让所有浏览器支持HTML5 video视频标签 (86)
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (82)
- » CSS3下的147个颜色名称及对应颜色值 (79)
- » 视区相关单位vw, vh..简介以及可实际应用场景 (76)
- » 写给自己看的display: flex布局教程 (76)
- » 小tips: 纯CSS实现打字动画效果 (76)



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

- 学习了, CSS中内联SVG图片有比Base64更好的形式
- 网页制作辅助工具-jQuery标尺参考线插件
- style标签下的CSS代码的显示与实时编辑
- 用3D LUT滤镜我做了个在线专业电影级别照片调色工具
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- windows系统下批量删除OS X系统.DS_Store文件
- 高富帅seajs使用示例及spm合并压缩工具露脸
- 小tips: 使用　等空格实现最小成本中文对齐
- 基于active,checked等状态类名的web前端交互开发
- 瞎折腾: 把JS,CSS任意文本文件加密成一张图片

