****我们可以用Autoprefixer模块来给css自动补全，实现多浏览器的兼容。****

Autoprefixer使用起来很简单，它根据最新的W3C规范，自动补充CSS前缀并编写普通的CSS，生成旧浏览器的代码。

Autoprefixer还可以自动删除过时的、不必要的前缀代码。

### Autoprefixer的安装和配置：

****1、首先当然是安装node.js****

    参考我之前的那篇文章。

****2、安装Autoprefixer****

打开cmd控制台，输入下面一行npm命令，安装Autoprefixer模块：-g是全局安装，如果不加会安装在当前目录。

npm install autoprefixer -g

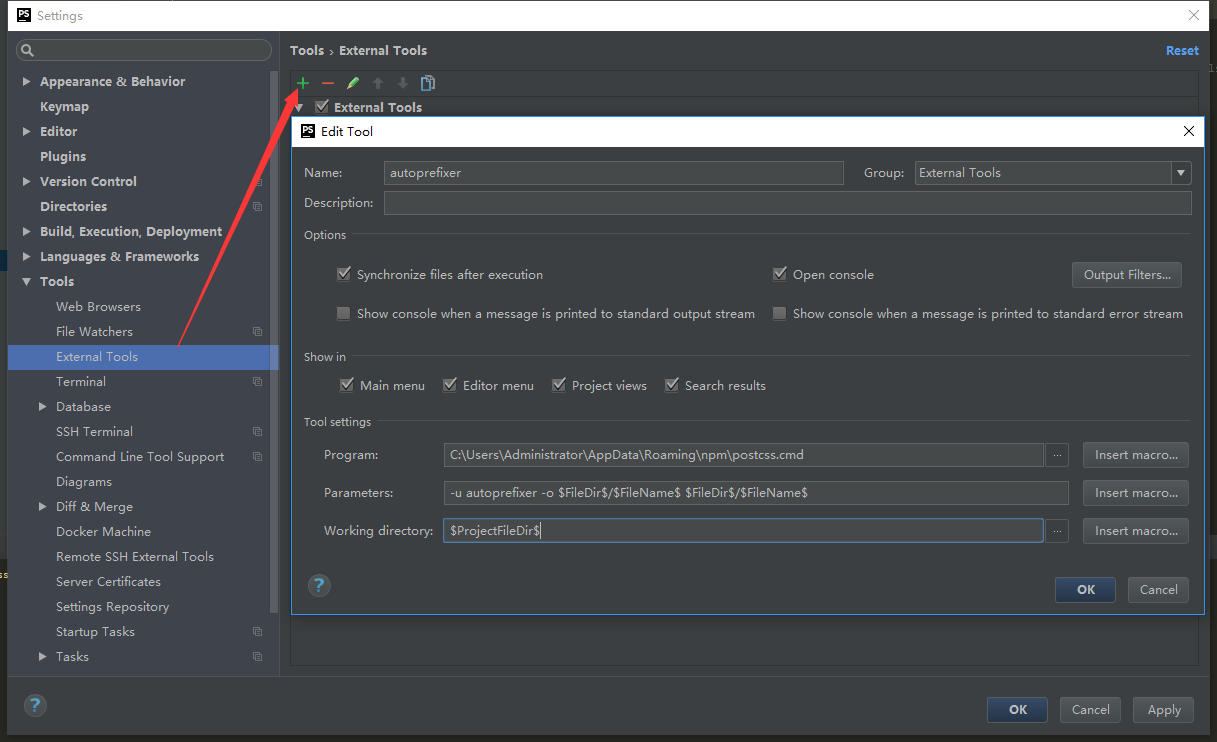
****3、安装postcss-cli****

Autoprefixer其实是postcss的插件，见：<https://github.com/code42day/postcss-cli>

npm install postcss-cli -g

**4、配置External Tools**

打开Webstorm设置，Preferences -> Tools -> External Tools，点击新增按钮，如图：



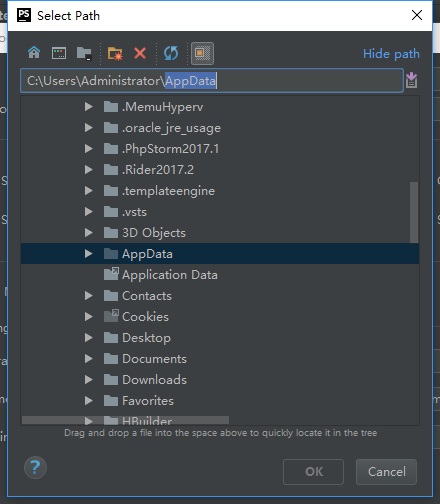
Name：autoprefixer

Tool settings：

       Program： 找到AppData下的文件postcss.cm  若找不到AppData，在地址栏输入%appdata%回车，或者在用户名文件夹后输入AppData即可，如下图

       Parameters：-u autoprefixer -o outputFile inputFile  （-u 插件名称 -o 输出文件 输入文件，注意文件顺序输出文件及输入文件 ，输入文件需要先新建，不能自动生成）

       Working directory：文件目录即可，可根据自己需要调整

****

我的配置：如果您是Windows用户，可以直接复制，注意：Deshun换成您的用户名

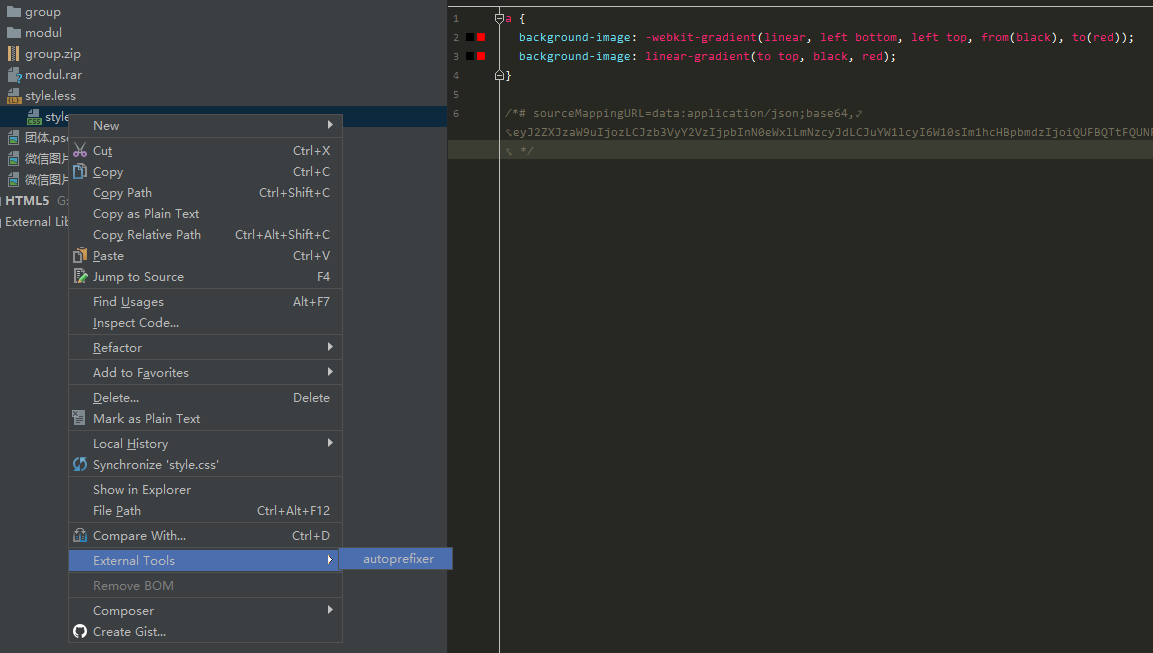
       Program：C:\Users\Deshun\AppData\Roaming\npm\postcss.cmd

       Parameters：-u autoprefixer -o $FileDir$/$FileName$ $FileDir$/$FileName$

       Working directory：$ProjectFileDir$

### 2.使用

**在需要转换的文件点击右键->External Tools->autoprefixer ,即可生成新增兼容前缀的文件**



****注意：Autoprefixer不支持中文目录，并且路径中不能有空格，否则会报错：Input Error: You must pass a valid list of files to parse****

### 3.配置浏览器版本

我们安装的[Autoprefixer](https://link.jianshu.com?t=https://github.com/postcss/autoprefixer" \t "_blank)是在全局目录，找到对应的文件，他又是通过[Browserslist](https://link.jianshu.com?t=https://github.com/ai/browserslist" \t "_blank)来实现版本控制的，最后在

C:\Users\user\AppData\Roaming\npm\node\_modules\autoprefixer\node\_modules\browserslist\index.js找到了这么一段代码，直接修改

# index.js// Default browsers query// browserslist.defaults = [ // 屏蔽默认// '> 1%',// 'last 2 versions',// 'Firefox ESR'// ]

browserslist.defaults = [

"iOS >= 7",

"Android > 4.1",

"Firefox > 20",

"last 2 versions"

]

直接修改我所需要的兼容方式

### 4.**完整清单参考**

您可以通过查询指定浏览器和Node.js版本（不区分大小写）：

* > 5%：全球使用情况统计选择的浏览器版本。 >=，<也<=工作。
* > 5% in US：使用美国使用情况统计。它接受[两个字母的国家/地区代码](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2" \l "Officially_assigned_code_elements)。
* > 5% in alt-AS：使用亚洲地区使用情况统计。可以在以下位置找到所有地区代码的列表[caniuse-lite/data/regions](https://github.com/ben-eb/caniuse-lite/tree/master/data/regions)。
* > 5% in my stats：使用[自定义使用数据](https://github.com/browserslist/browserslist" \l "custom-usage-data)。
* cover 99.5%：提供覆盖的最流行的浏览器。
* cover 99.5% in US：与上面相同，使用[双字母国家代码](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2" \l "Officially_assigned_code_elements)。
* cover 99.5% in my stats：使用[自定义使用数据](https://github.com/browserslist/browserslist" \l "custom-usage-data)。
* maintained node versions：所有Node.js版本，[仍由](https://github.com/nodejs/Release) Node.js Foundation [维护](https://github.com/nodejs/Release)。
* node 10和node 10.4：选择最新的Node.js 10.x.x 或10.4.x发布。
* extends browserslist-config-mycompany：从browserslist-config-mycompanynpm包中获取查询 。
* ie 6-8：选择包含范围的版本。
* Firefox > 20：Firefox的版本比20较新的 >=，<和<=工作了。
* iOS 7：iOS浏览器版本7直接。
* Firefox ESR：最新的[Firefox ESR]版本。
* unreleased versions或unreleased Chrome versions：alpha和beta版本。
* last 2 major versions或last 2 iOS major versions：最近2个主要版本的所有次要/补丁版本。
* since 2015或last 2 years：自2015年以来发布的所有版本（也since 2015-03和since 2015-03-10）。
* dead：来自last 2 version查询的浏览器，但全球使用统计数据少于0.5％，且24个月内没有官方支持或更新。现在是IE 10，IE\_Mob 10，BlackBerry 10， BlackBerry 7，和OperaMobile 12.1。
* last 2 versions：每个浏览器的最后两个版本。
* last 2 Chrome versions：Chrome浏览器的最后两个版本。
* defaults：Browserslist的默认浏览器（> 0.5%, last 2 versions, Firefox ESR, not dead）。
* not ie <= 8：排除先前查询选择的浏览器。

您可以添加not到任何查询。