

css后处理器

[参考链接](#)

css后处理器是对css进行处理，并最终生成css预处理器，它属于广义上的css预处理器

Sass、LESS和Stylus的对比

安装

1. sass的安装

首先安装ruby, 建议安装到C盘中，安装过程中选择 “Add Ruby executables to your PATH”

ruby安装成功后，启动Ruby的Command控制面板，输入命令

```
gem install sass
```

回车后会自动安装好Sass

2. Less的安装

分为两种：客户端和服务端安装。

a. 客户端

只需要在官网载一个Javascript脚本文件主 “less.js”，然后在我们需要引入LESS源文件的HTML的中加入如下代码：

```
<link rel="stylesheet/less" type="text/css" href="文件路径/styles.less">
```

```
<script src="文件路径/less.js" type="text/javascript"></script>
```

b. 服务器

利用node的包管理器（npm）安装LESS，安装成功后就可以在Node环境中对LESS文件进行编译。

首先安装node, 安装成功后，启动node的Command控制面板，输入命令

```
$ npm install less
```

回车后会自动安装好less

3. Stylus的安装

Stylus的安装和LESS在服务器端的安装很相似，先安装Node 。在Node的Command控制面板输入命令：

```
$ npm install stylus
```

回车后会自动安装好stylus

Sass、LESS和Stylus转换成css文件

1. sass

先在项目中创建一个Sass文件，此例中将其命名为“style.scss”，并且将其放在对应的项目样式中

启动Ruby的Command控制面板，找到需要转译的Sass文件

在对应的目录下输入下面的命令：

```
sass style.scss style.css  
sass --watch style.scss:style.css  
1  
2
```

就会生成自动“style.css”文件，并自动更新对应的CSS样式文件

2. Less (了解)

在安装的Node JS环境下通过其自己的命令来进行转译。

```
$ lessc style.less
```

1

上面的命令会将编译的CSS传递给stdout，你可以将它保存到一个文件中：

```
$ lessc style.less > style.css
```

1

3. Stylus (了解)

Stylus具有可执行性，因此Stylus能将自身转换成CSS。Stylus可以读取自“stdin”输出到“stdout”，因此Stylus可以像下面转译源文件：

```
$ stylus -compress <some.styl> some.css
```

1

Stylus也像Sass一样，同时接受单个文件和整个目录的转译。例如，一个目录名为“css”将在同一个目录编译并输出“.css”文件。

```
$ stylus css
```

1

下面的命令将输出到“./public/stylesheets”：

```
$ stylus css -out public/stylesheets
```

1

还可以同时转译多个文件：

```
$ stylus one.styl two.styl
```

1

如果你的浏览器安装了Firebug，那么可以使用FireStylus扩展。

```
$ stylus -firebug <path>
```

变量的处理

Less：变量处理方式 - 懒加载，所有 Less 变量的计算，都是以这个变量最后一次被定义的值为准。

Less

```
@size: 10px;
.box { width: @size; }
@size: 20px;
.ball { width: @size; }
```

1

2

3

4

输出：

```
.box { width: 20px; }
.ball { width: 20px; }
```

1

2

Stylus

```
size = 10px
.box
  width: size
```

```
size = 20px
.ball
  width: size
```

1

2

3

4

5

6

7

8

输出：

```
.box {  
  width: 10px;  
}
```

```
.ball {  
  width: 20px;  
}
```

1

2

3

4

5

6

7

Sass的变量处理方式和Stylus相同，变量值输出时根据之前最近的一次定义计算。这其实代表了两种理念：Less 更倾向接近 CSS 的声明式，计算过程弱化调用时机；而 Sass 和 Stylus 更倾向于指令式。

如果在Less中引入第三方样式库，可以通过重新配置变量来改变样式

优点：stylus和sass这样的处理会不容易受多个第三方库变量名冲突的影响，因为一个变量不能影响在定义它以前的输出样式

作者：seven-yang

来源：CSDN

原文：<https://blog.csdn.net/yangyixue123/article/details/79133748>

版权声明：本文为博主原创文章，转载请附上博文链接！