

Less 入门与安装

快速入门

Less 是一门 CSS 预处理语言，它扩充了 CSS 语言，增加了诸如变量、混合（mixin）、函数等功能，让 CSS 更易维护、方便制作主题、扩充。

Less 可以运行在 Node、浏览器和 Rhino 平台上。网上有很多第三方工具帮助你编译 Less 源码。

例如：

```
@base: #f938ab;
```

```
.box-shadow(@style, @c) when (iscolor(@c)) {  
  -webkit-box-shadow: @style @c;  
  box-shadow:        @style @c;  
}  
  
.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {  
  .box-shadow(@style, rgba(0, 0, 0, @alpha));  
}  
  
.box {  
  color: saturate(@base, 5%);  
  border-color: lighten(@base, 30%);  
  div { .box-shadow(0 0 5px, 30%) }  
}
```

编译为：

```
.box {  
  color: #fe33ac;  
  border-color: #fdcdea;  
}  
  
.box div {  
  -webkit-box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);  
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);  
}
```

[Improve this section.](#)

使用 Less

Less 可以通过 npm 在命令行上运行；在浏览器上作为脚本文件下载；或者集成在广大的第三方工具内。请参考 [用法](#)。

安装

在服务器端最容易的安装方式就是通过 npm ([node.js](#) 的包管理器)，方法如下：

```
$ npm install -g less
```

命令行用法

安装 Less 后，就可以在命令行上调用 Less 编译器了，如下：

```
$ lessc styles.less
```

这将输出编译之后的 CSS 代码到 stdout，你可以将输出重定向到一个文件：

```
$ lessc styles.less > styles.css
```

若要输出压缩过的 CSS，只需添加 -x 选项。如果希望获得更好的压缩效果，还可以通过 --clean-css 选项启用 [Clean CSS](#) 进行压缩。

执行 lessc 且不带任何参数，就会在命令行上输出所有可用选项的列表。

在代码中调用 Less

可以像下面这样在代码中调用 Less 编译器（Node 平台）。

```
var less = require('less');
```

```
less.render('.class { width: (1 + 1) }', function (e, css) {  
  console.log(css);  
});
```

输出为

```
.class {  
  width: 2;  
}
```

你还可以手动调用分析器（parser）和编译器：

```
var parser = new(less.Parser);
```

```
parser.parse('.class { width: (1 + 1) }', function (err, tree) {  
  if (err) {  
    return console.error(err)  
  }  
  console.log(tree.toCSS());  
});
```

配置

可以给编译器传递多个参数：

```
var parser = new(less.Parser)({  
  paths: ['.'], // Specify search paths for @import  
  directives  
  filename: 'style.less' // Specify a filename, for better error  
  messages  
});
```

```
parser.parse('.class { width: (1 + 1) }', function (e, tree) {  
  tree.toCSS({  
    // Minify CSS output  
    compress: true  
  });  
});
```

Grunt

Less 还和流行的 Grunt 构建工具进行了集成，可以参考 [grunt-contrib-less](#) 插件。

第三方工具

请参考 [用法](#) 一节了解其它工具的细节。

客户端用法

在浏览器上跑 less.js 非常方便开发，但是不推荐用于生产环境。

在客户端使用 Less.js 是最容易的方式，并且在开发阶段很方便，但是，在生产环境中，性能和可靠性非常重要，我们建议最好使用 *node.js* 或其它第三方工具进行预编译。

那就开始吧，在页面中加入 .less 样式表的链接，并将 rel 属性设置为

"stylesheet/less":

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

接下来，[下载 less.js](#) 并通过 `<script></script>` 标签将其引入，放置于页面的 `<head>` 元素内：

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

提示

- 务必确保在 less.js 之前加载你的样式表。
- 如果加载多个 .less 样式表文件，每个文件都会被单独编译。因此，一个文件中所定义的任何变量、mixin 或命名空间都无法在其它文件中访问。

浏览器端设置参数

在 `<script src="less.js"></script>` 之前 定义全局的 less 对象就可以为 Less.js 设置参数：

```
<!-- set options before less.js script -->
```

```
<script>
```

```
  less = {  
    env: "development",  
    async: false,  
    fileAsync: false,  
    poll: 1000,  
    functions: {},  
    dumpLineNumbers: "comments",  
    relativeUrls: false,  
    rootpath: ":/a.com/"  
  };
```

```
</script>
```

```
<script src="less.js"></script>
```

[Improve this section.](#)

浏览器端下载

[下载源码](#)

直接从 GitHub 上下载最新的 Less.js 源码。

[通过 GitHub 克隆或 fork 本项目](#)

通过 [Bower](#) 安装

通过 [Bower](#) 安装 Less.js 项目。

```
bower install less
```

Less CDN 加速

[CDN](#).

```
<script
```

```
src="http://cdn.bootcss.com/less.js/1.7.0/less.min.js"></script>
```