

SHANXI **UNIQUE**  
TECHNOLOGY CO.,LTD.



优逸客

SHANXI UNIQUE TECHNOLOGY

Less 详解

# Less是什么?

- 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、具有逻辑性，能像编程一样编写css代码。

## ➤ 服务器端安装

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
```

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

## ➤ 客户端安装

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>
```

- 导入外部less文件
- 在一个less文件导入另外一个less文件

```
@import "lib.less";
```

```
@import "lib";
```

如果你想导入一个CSS文件而且不想LESS对它进行处理，只需要使用.css后缀就可以：

```
@import "lib.css";
```

这样LESS就会跳过它不去处理它。

## ➤ 注释

CSS 形式的注释在 LESS 中是依然保留的:

```
/* Hello, I'm a CSS-style comment */
```

```
.class { color: black }
```

LESS 同样也支持双斜线的注释, 但是编译成 CSS 的时候自动过滤掉:

## ➤ 免编译

有时候我们需要输出一些不正确的CSS语法或者使用一些 LESS 不认识的专有语法.

要输出这样的值我们可以在字符串前加上一个 ~, 例如:

```
.class {  
  filter: ~"alpha(opacity=100)";  
}
```

```
@color: red;
```

```
#header { color: @color; }
```

输出:

```
#header { color: red }
```

➤ 混合模式

➤ 定义

```
.bordered {  
  border-top: dotted 1px black;  
  border-bottom: solid 2px black;  
}
```

➤ 调用

```
#menu a {  
  color: #111;  
  .bordered;  
}
```

➤ 输出

```
#menu a {  
  color: #111;  
  border-top: dotted 1px black;  
  border-bottom: solid 2px black;  
}
```



# less带参数混合模式

## ➤ 定义

```
.border-radius (@radius: 5px) {  
  border-radius: @radius;  
  -moz-border-radius: @radius;  
  -webkit-border-radius: @radius;  
}
```

## ➤ 调用

```
#header {  
  .border-radius;  
}
```

## ➤ 输出

```
#header{  
border-radius: 5px;  
-moz-border-radius: 5px  
-webkit-border-radius: 5px;  
}
```

# less 不暴露的混合模式

## ➤ 定义

```
.wrap () {  
  text-wrap: wrap;  
  white-space: pre-wrap;  
  white-space: -moz-pre-wrap;  
  word-wrap: break-word;  
}
```

## ➤ 调用

```
pre { .wrap }
```

## ➤ 输出:

```
pre {  
  text-wrap: wrap;  
  white-space: pre-wrap;  
  white-space: -moz-pre-wrap;  
  word-wrap: break-word;  
}
```

# less 模式匹配

## ➤ 定义

```
.mixin (dark, @color) {  
  color: darken(@color, 10%);  
}  
.mixin (light, @color) {  
  color: lighten(@color, 10%);  
}  
.mixin (@_, @color) {  
  display: block;  
}
```

## ➤ 调用

```
@switch: light;
```

```
.class {  
  .mixin(@switch, #888);  
}
```

## ➤ 输出:

```
.class {  
  color: #a2a2a2;  
  display: block;  
}
```

## ➤ 定义

```
.mixin(@color) when (@color=red)
  background:@color;
}

.mixin(@color) when (@color=blue)
  background:@color;
}
```

## ➤ 调用

```
.class {
  .mixin(red);
}
```

## ➤ 输出:

```
.class {
background:red
}
```

# less 嵌套规则

## ➤ 定义

```
.bordered {  
  &.float {  
    float: left;  
  }  
  .top {  
    margin: 5px;  
  }  
}
```

## ➤ 输出:

```
.bordered.float {  
  float: left;  
}  
.bordered .top {  
  margin: 5px;  
}
```

变量可以用类似ruby和php的方式嵌入到字符串中，像  
@{name}这样的结构：

```
@base-url: "http://assets.fnord.com";
```

```
background-image: url("@{base-url}/images/bg.png");
```

LESS 提供了一系列的颜色运算函数.

```
lighten(@color, 10%);    // return a color which is 10% *lighter* than @color
darken(@color, 10%);    // return a color which is 10% *darker* than @color
saturate(@color, 10%);  // return a color 10% *more* saturated than
@color
desaturate(@color, 10%); // return a color 10% *less* saturated than
@color
fadein(@color, 10%);     // return a color 10% *less* transparent than
@color
fadeout(@color, 10%);    // return a color 10% *more* transparent than
@color
fade(@color, 50%);       // return @color with 50% transparency
spin(@color, 10);        // return a color with a 10 degree larger in hue than
@color
spin(@color, -10);       // return a color with a 10 degree smaller hue than
@color
mix(@color1, @color2);   // return a mix of @color1 and @color2
```

LESS提供了一组方便的数学函数，你可以使用它们处理一些数字类型的值：

```
round(1.67); // returns `2`
```

```
ceil(2.4); // returns `3`
```

```
floor(2.6); // returns `2`
```

如果你想将一个值转化为百分比，你可以使用percentage 函数：

```
percentage(0.5); // returns `50%`
```





谢谢观看...

SHANXI **UNIQUE**  
TECHNOLOGY CO.,LTD.