CSS预处理器

> css是标记语言，不是编程语言，所以不具备编程语言的特点（`编程语言的特点：变量存储、判断和循环的逻辑操作、函数的封装继承和多态...`），如果具备以上编程语言的特点，开发css的时候将会非常的方便
>
> css预处理编译语言的宗旨就是`把css变为编程语言来开发，提高开发效率，实现组件化的封装`
>
> - less
> - sass
> - stylus
> - ...
>
> 之所以叫做预编译语言是因为：我们使用上述语言编写完成的样式代码，浏览器不能识别，我们需要把代码编译成为正常能识别的css代码才可以，所以叫做预编译

LESS的编译

导入JS编译

1、首先导入我们需要的less文件，重点记住`rel='stylesheet/less'`

2、导入`less-2.5.3.min.js`,它会自动去识别rel=’stylesheet/less’中的文件，把less编译成正常的css，然后再进行页面的渲染

```
<head>
    <meta charset="UTF-8">
    <title>珠峰培训</title>
    <link rel="stylesheet/less" href="css/temp.less">
    <script src="js/less-2.5.3.min.js"></script>
</head>
```

开发的时候我们可以这样处理，但是项目上线的时候，我们不用这种方式（`使用这种方式耗性能：用户访问页面需要把less现编译成css才可以渲染，耽误页面渲染时间`），项目上线我们引入的肯定是已经编译完成的css文件

编译工具

http://tool.oschina.net/less

http://koala-app.com/ (KOALA)

…

基于NODE编译

1、需要先把node安装上

2、在node全局环境中安装less

`npm install less -g`

`npm install --global less`

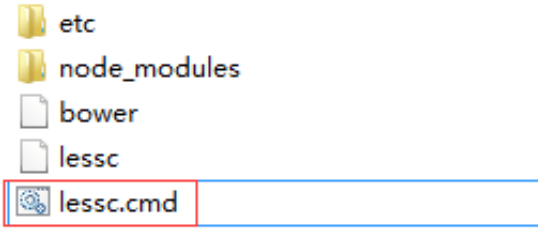
出现了`lessc.cmd`

1)我们可以使用lessc这个命令了

2)命令名称是lessc

```
C:\Users\team>npm install less -g
C:\Users\team\AppData\Roaming\npm\lessc -> C:\Users\team\AppDa
e_modules\less\bin\lessc
- jodid25519@1.0.2 node_modules\less\node_modules\jodid25519
C:\Users\team\AppData\Roaming\npm
`-- less@2.7.2
  +-- image-size@0.5.5
  +-- mime@1.4.0
  +-- promise@7.3.1
  | `-- asap@2.0.6
  +-- request@2.81.0
  | +-- http-signature@1.1.1
```

安装在全局环境中的目录



3、找到less所在的目录，在这个目录中编译less文件即可

1)在当前目录下打开DOS命令

`当前目录下，shift+右键，在此处打开命令窗口`

`在当前目录的地址栏中输入cmd即可`

`通过DOS命令中的cd命令进入`

2)编译文件

`lessc xxx.less xxx.min.css -x` 把less文件代码进行编译，并且把编译后的结果进行压缩

LESS的基础语法

http://lesscss.cn/

变量

```
@color:red;
.box{
    background:@color;
}

@h:100;
@p:20;
.box{
    padding: unit(@p, px);
    width: unit((@h)-(@p*2), px);
    height: unit((@h)-(@p*2), px);
}
```

作用域和层级嵌套

```
@h:100;
.box{
    .progress{
        @h:200;
        a{
            height:unit(@h,px);/*200*/
        }
    }
}

/*output*/
.box .progress a{}
```

```scss
.box {
  a {
    &:hover {
      background: red;
    }
    &.bg {
      background: grey;
    }
    & > span {
      background: orange;
    }
  }
}

/*output*/
.box a:hover {
    background: red;
}
.box a.bg {
    background: grey;
}
.box a > span {
    background: orange;
}
/*&连字符：在当前选择器后面直接加上想要链接的内容*/
```

函数

```
.box{
    width:100px;
    height:100px;
}
.mark{
    .box;
    background:red;
}

/*output*/
.box {
    width: 100px;
    height: 100px;
}
.mark {
    width: 100px;
    height: 100px;
    background: red;
}
```

```
.box{
    width:100px;
    height:100px;
}
.mark:extend(.box){
    background:red;
}
/*output*/
.box,
.mark {
    width: 100px;
    height: 100px;
}
.mark {
    background: red;
}
```

```
.transition(@property:all,@duration:1s,@timing-function:linear,@delay:0s)
{
  -webkit-transition: @arguments;
  -moz-transition: @arguments;
  -ms-transition: @arguments;
  -o-transition: @arguments;
  transition: @arguments;
}
.box {
  .transition;
}
.box {
  .transition(@duration: .5s);
}
.box {
  .transition(@duration: .5s,@delay: 1s);
}
.box {
  .transition(all, 1s, linear, 0s);
}
```

提供一些方法

```less
.box {
  a {
    @c: red;
    background: @c;

    &:hover {
      /*darken:加深  lighten:变浅*/
      background: darken(@c, 15%);
      background: lighten(@c, 15%);
    }
  }
}
```

导入所依赖的文件

```less
/*在当前文件中导入public,编译的时候会把public中的代码也编译*/
@import "public";

/*在当前文件中导入public,编译的时候不会编译public中的代码，只是导入*/
@import (reference) "public";
```