复习：

见思维导图

今日目标：

(1)Bootstrap第四部分：插件 —— 掌握

(2)Less语言 —— 重点

(3)Bootstrap第五部分：定制 —— 难点

1.Bootstrap插件概述

Bootstrap一共提供了十几个插件函数，可以单个引入，也可以一次性全部引入(bootstrap.js)

使用方法有两种：

**(1)使用JS方式调用**

$('div').modal( );

**(2)使用data-\*方式调用 —— 推荐写法**

<div data-toggle="modal">

2.Bootstrap第四部分：插件 —— 下拉插件

1)JS方式调用：

$('.dropdown a').dropdown( );

2)data-\*方式调用：

<a data-toggle="dropdown">

3.Bootstrap第四部分：插件 —— 警告框

2)data-\*方式调用：

<div class="alert alert-四种颜色">

<span class="close" data-dismiss="alert">×</span>

文字内容

</div>

3.Bootstrap第四部分：插件 ——折叠效果

2)data-\*方式调用：

<a data-toggle="collapse" href="target">触发元素</a>

<div id="target" class="collapse"></div>

|  |
| --- |
| Model： 数据模型  Modal： 模态对话框  Module： 模块 |

4.Bootstrap第四部分：插件 —— 模态框

modal：模态对话框，在父窗口中打开了一个子窗口，只要子窗口不关闭，则父窗口无法获得输入焦点——该子窗口就称为“模态子窗口”。浏览器中默认： window.alert()/confirm()/prompt() 都是模态子窗口，但无法定制。

一般项目中，使用DIV模拟出模态子窗口的效果：

<a data-toggle="modal" href="#modal-login">弹出模态对话框</a>

<div id="modal-login" class="modal"> <!--半透明的遮罩层-->

<div class="modal-dialog"> <!--尺寸和定位-->

<div class="modal-content"> <!--背景/边框/倒角/阴影-->

<div class="modal-header">

<span class="close" data-dismiss="modal">×</span>

头部

</div>

<div class="modal-body">主体</div>

<div class="modal-footer">尾部</div>

</div>

</div>

</div>

午间练习：修改intel项目首页，点击“登录”后弹出模态对话框：



5.Bootstrap第四部分：插件 —— 工具提示(tooltip)

提示：不单要用data-\*，还要调用js

6.Bootstrap第四部分：插件 —— 弹出框(popover)

提示：不单要用data-\*，还要调用js

7.Bootstrap第四部分：插件 —— 轮播广告(Carousel)

Carousel：旋转木马、轮播广告

提示：Bootstrap提供的轮播广告结构复杂，只需要记住最外层容器div.carousel，其余内容全部靠Bootlint提示即可。

练习：

为轮播广告添加“上一张/下一张”轮播控件；

为轮播广告添加“1/2/3/4”序号指示器；

8.自学：Bootstrap第四部分：插件 —— 附加导航(Affix)

9.自学：Bootstrap第四部分：插件 —— 滚动监听(ScrollSpy)

10.样式语言的分类

**(1)静态样式语言**

即CSS，可以直接被浏览器解释执行。作为一门语言，CSS有缺陷的，如缺少：类型、变量、运算、函数、对象和继承等等语言必需的基本概念，导致了CSS样式代码可维护性差。

**(2)动态样式语言**

在CSS基础上添加了动态语言必需的因素，如类型、变量、运算、函数、对象和继承等，极大了提高了样式代码的可维护性。常见的动态样式语言：

1)Sass / SCSS

2)Stylus

3)Less

注意：浏览器只能解析CSS，所以所有的动态样式语言的代码都必须转换为CSS才能执行——此过程称为动态样式代码的“编译(Compile)”

11.动态样式语言——Less概述

官网：http://lesscss.org/

中文网：http://less.bootcss.com/

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文件有两种使用方法：

**(1)在客户端使用Less —— 仅作了解**

创建x.less;

创建x.html，引入x.less，引入less.js（Less编译器）

客户端请求x.html，下载.less和.js两个文件，在浏览器中运行less.js把x.less文件编译为css代码。

**(2)在服务器端使用Less —— 推荐方法**

创建x.less；

程序员在自己的电脑上安装less编译器；

程序员执行less编译器，把x.less 编译为 x.css；

创建x.html，引入x.css；

客户端请求html，下载css，直接运行即可

很容易出错的地方：—— 如何在服务器端搭建Less编译环境

**(1)下载并安装一款独立于浏览器的JS解释器——Node.js**

成功的标志：

node -v 能够显示出版本号

**(2)下载并安装Less编译器程序：lessc.js**

成功的标志： 能够找到

C:\npm\node\_modules\less\bin\lessc —— JS文件

**(3)运行JS解释器，执行lessc.js，编译.less文件得到.css文件**

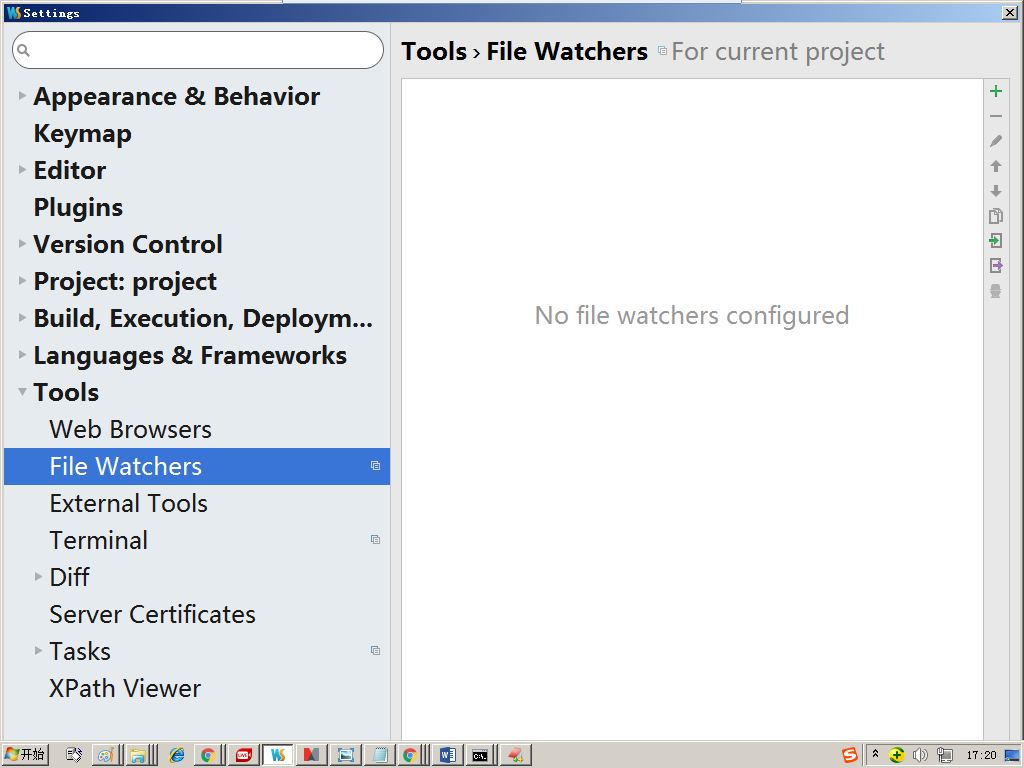
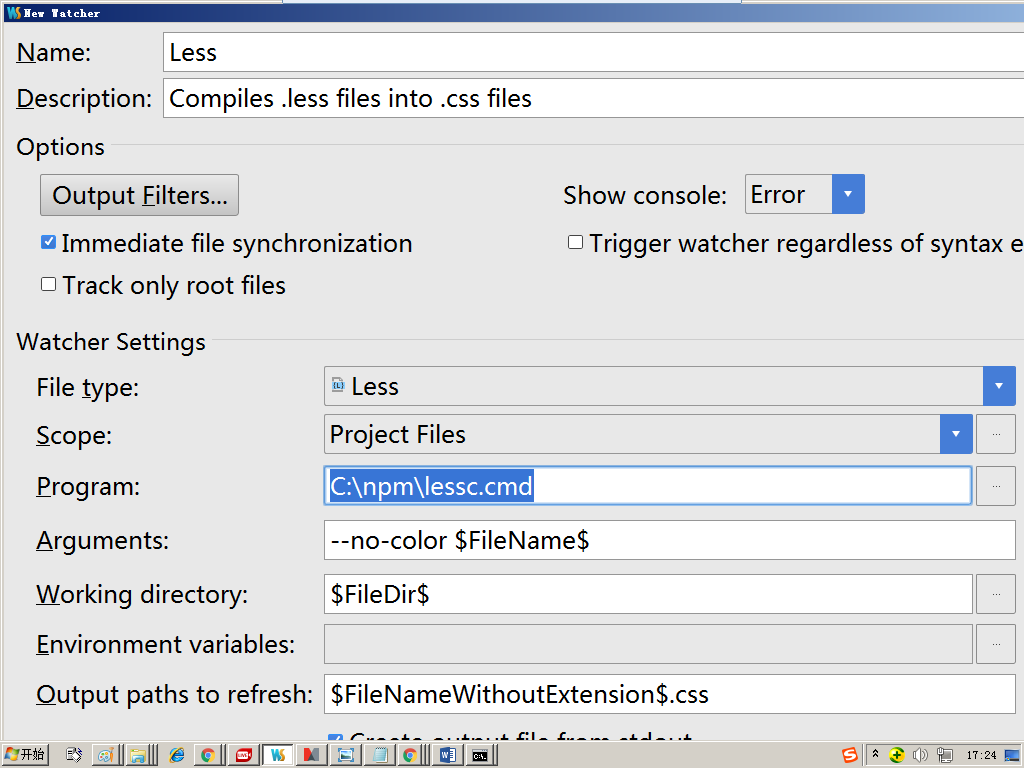
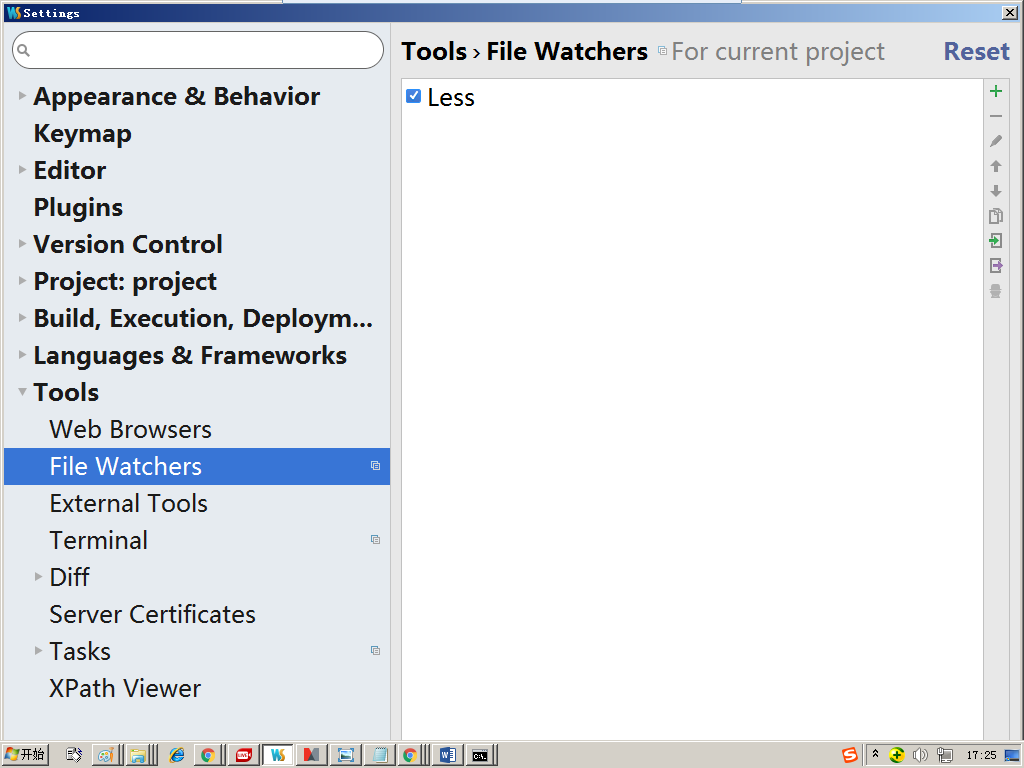
node C:\npm\node\_modules\less\bin\lessc e:\5.less > e:\5.css

成功的标志： 在e盘多出一个5.css文件

此步也可以用WebStorm中的FileWatcher功能来实现

(3.1)配置WebStorm中的FileWatcher，指定lessc位置

(3.2)只要用户新建/修改.less文件，WS会自动调用lessc程序把.less编译为.css

12.Less语法学习

(1)Less支持CSS所有的语法

(2)Less支持多行/单行注释，但CSS只支持多行注释，所以Less中的单行注释不会被编译到CSS文件

(3)Less有变量(Variable)的概念

声明变量： @变量名: 值;

使用变量： 选择器 { 样式: @变量名; }

变量值可以是任意合法的样式值。

(4)Less可以执行样式/变量的计算

加、减、乘、除、取余

(5)Less支持样式的混入(Mixin)

选择器1 { 样式; }

选择器2 {

选择器1;

样式;

}

(6)Less在样式混入时可以指定参数

选择器1(@变量, @变量...) { 样式; }

选择器2 {

选择器1(值, 值...);

样式;

}

(7)Less支持样式嵌套

选择器1 {

样式1;

选择器2 {

样式2;

}

}

会被编译为：

选择器1 { 样式1; }

选择器1 选择器2 { 样式2; }

(8)Less为样式提供了几十个可用的函数

ceil( )

floor( )

percentage( 5/10 ) // 50%

round( )

lighten(@c, 20%) 颜色变淡指定的百分比

darken(@c, 30%) 颜色变暗指定的百分比

....

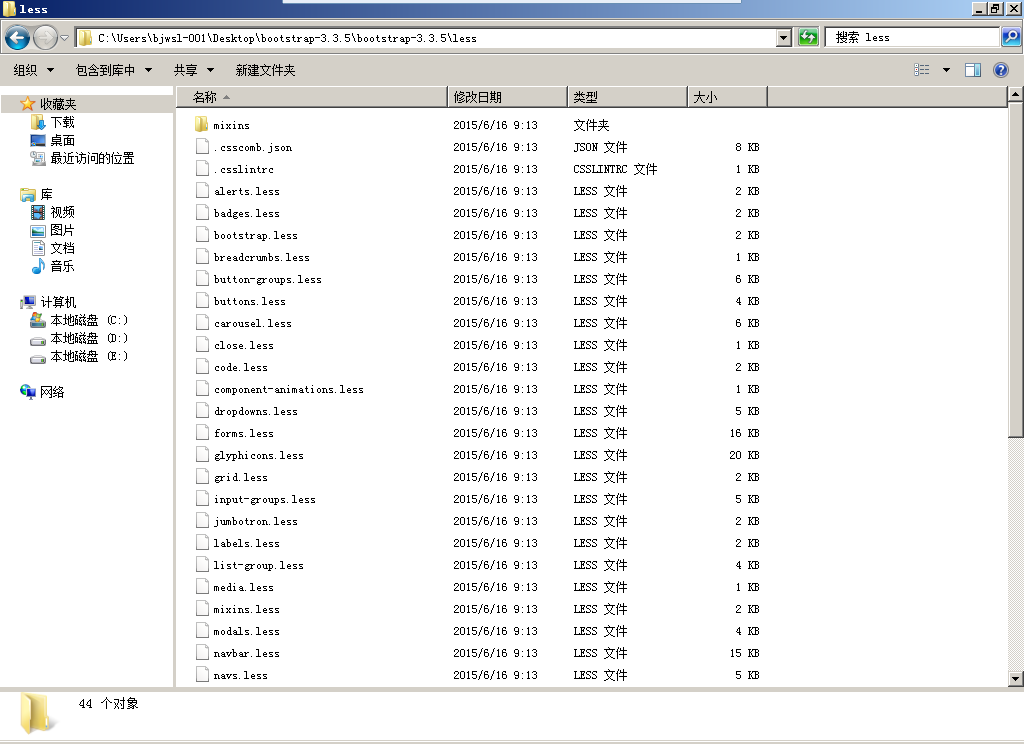
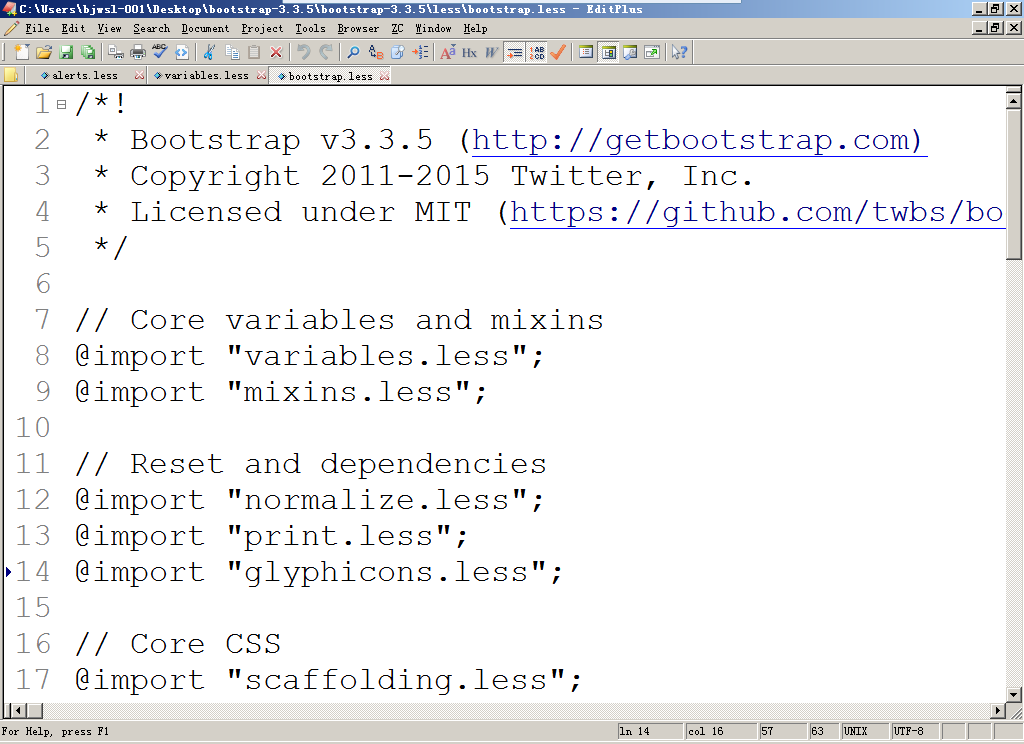
(9)Less支持文件包含指令

CSS提供了@import指令，可用于包含其它的CSS文件，但由于会增加请求次数，不推荐使用；

Less也提供了@import指令，可用于包含其它的Less文件，推荐使用！—— Less的文件包含是在服务器端执行的文件拼合，客户端的一次请求就可以获得所有样式！

@import "xx.less";

练习：观察Bootstrap真正的Less源代码文件的结构

13.通过修改Bootstrap的Less源文件实现定制

定制目标：

**(1)Bootstrap瘦身，删除不必要的样式**

注释掉bootstrap.less中不需要的@import即可

**(2)修改Boostrap默认的样式值，实现粗粒度定制**

修改variables.less中变量的值即可

**(3)修改Boostrap组件的细节样式，实现细粒度定制**

修改特定的组件对应的.less文件，如dropdown.less

课后任务：

(1)仿写Bootstrap起步中的“示例”

(2)实现“我的大项目”响应式版本中样式的定制