第三阶段课程：

AJAX：

HTML5新特性：

Bootstrap框架：

Bootstrap容易学，不容易掌握！难点两个：

(1)class名称太多——1000+个！

(2)class特定的嵌套关系——需要加强记忆！

今日目标：

(1)手写响应式网页 —— 重点

(2)Bootstrap第一部分——起步——比较简单

1G：模拟信号

2G：数字信息

2.5G：GPRS，浏览网页——WAP/WML

3G：智能手机/Linux系统，浏览真正网页——HTML

1.什么是响应式网页？

一个页面，可以根据浏览设备的不同，以及特性的不同，而自动改变布局、大小等。

优点：可以自动适配PC、PAD、PHONE浏览器屏幕

不足：代码变复杂，需要考虑更多兼容性，并不适合内容非常多网页

2.如何测试响应式网页？

(1)使用真实的物理设备

优点：测试结果真实可靠

不足：测试任务量太大

(2)使用第三方测试软件

优点：无需添置真实设备

不足：软件测试效率低、部分功能无法测试、测试结果有待进一步验证

(3)使用Chrome等浏览器的设备模拟器

优点：测试功能强大，简单易用

不足：测试结果有待进一步验证

|  |
| --- |
| 早期的网页都是比较大，用于在手机屏幕中浏览，只能进行缩小，导致用户的浏览体验不好。    iOS中提出了Viewport（视口）的概念，可以任意的指定大小，显示任意大小的网页：    网页内容不再缩放得很小，可以看清。Android也引用该概念。  为了防止视口宽度超过手机屏幕宽度，推荐设置为width=device-width |

3.如何手写响应式网页——重点——面试题

(1)添加viewport元标签

<meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no">

(2)容器宽度尽量使用相对尺寸，不用绝对单位

.container {

/\*width: 1211px;\*/

width: 100%;

height: auto;

}

(3)文字尽量使用相对尺寸，不用绝对单位

.txt {

/\*font-size: 14px\*/

/\*font-size: 1em;\*/

font-size: 1rem;

}

(4)图片尽量使用相对尺寸，代替绝对单位

img {

/\*width: 400px;\*/

/\*width: 100%;\*/

max-width: 100%; /\*不会超过图片的原始尺寸，防止无限放大产生的失真\*/

}

(5)尽量避免固定布局，使用流式布局

1)float: left;

2)display: inline-block;

(6)使用CSS3 Media Query技术，实现有选择性的执行部门CSS

4.CSS3媒体查询技术—— 重点

Media：媒体、媒介，指浏览网页的设备，如screen(pc/pad/phone/watch)、print、tv、projection、tty(命令行、针式打印机)、braille等

Query：查询出浏览设备的特性，如宽、高、色彩深度、解析度、方向等

CSS3 Media Query：在CSS中，根据浏览设备的类型、特性不同而有选择性的执行不同的CSS代码。

**具体有两种使用方法：**

(1)为不同的设备编写不同的CSS文件，让浏览器有选择的加以执行

<link media="screen and (min-width:768px) and (max-width:991px)" rel="stylesheet" href="css/screen\_pad.css"/>

不足：当前屏幕不满足条件的CSS文件也会被请求，但不会被执行

(2)编写一个CSS文件，其中的部分代码只在特定的设备中加以执行

@media screen and (min-width: 768px) {

.box {

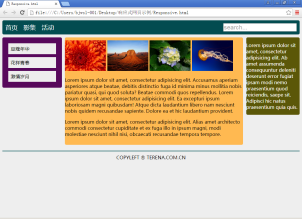
margin: 10px;

padding: 10px;

}

}

午间练习：使用媒体查询，实现如下的响应式网页



>=992 991px ~ 768px <767px

5.Bootstrap概述

Bootstrap：起步，引导程序，是一个很常见的技术词汇。

**Twitter Bootstrap是一个HTML、CSS、JS框架，适用于响应式网页的开发。**

英文官网：http://getbootstrap.com/

中文镜像：http://www.bootcss.com/

主要内容分为五部分：

(1)起步

(2)全局CSS样式

(3)组件

(4)jQuery插件

(5)定制

6.Bootstrap第一部分——起步

下载

基本模板

|  |
| --- |
| <html lang="zh-cn"></html>  lang属性指定当前页面所用的自然语言，如zh-cn、zh-hk、zh-tw、zh-sg、zh、en、en-us、en-uk、de、fr、ja.... 有两个作用：  (1)为浏览器的自动翻译功能提供基础语言  (2)为屏幕阅读软件确定基础发音 |
| <meta http-equiv="X-UA-Compatible" content="IE=edge">  X-UA-Compatible响应消息头部的含义——IE浏览器专用头部：  Cross-UserAgent-Compatible，跨（IE）浏览器兼容性级别。  IE6：  IE7： IE6/7  IE8： IE6/7/8  IE9： IE6/7/8/9  IE10： IE6/7/8/9/10  IE11： IE6/7/8/9/10/11  X-UA-Compatible: IE=6 告诉客户端IE浏览器，启用其中的6版本的内核。  X-UA-Compatible: IE=edge告诉客户端IE浏览器，启用其中的最新版本的内核。 |
| **html5shiv.js**：由@afarkas @jdalton @jon\_neal @rem编写的JS，让老IE兼容H5新标签，如header、section.... |
| **respond.js**，由Scott Jehl编写，让老IE支持响应式网页的必备内容——CSS3媒体查询 |

7.Bootstrap对HTML5标签默认样式的重置（CSS Reset）

提示：Bootstrap底层使用normalize.css对默认样式进行了重置。

常用标签重置的样式：

\* {box-sizing: border-box;}

html { font-size:10px; }

body { font-size:14px; font:'Helvatica Neue'; color:#333; }

h1 { font-size:36px; margin-top: 20px; margin-bottom:10px; }

h2 { font-size:30px; margin-top: 20px; margin-bottom:10px; }

h3 { font-size:24px; margin-top: 20px; margin-bottom:10px; }

h4 { font-size:18px; margin-top: 10px; margin-bottom:10px; }

h5 { font-size:14px; margin-top: 10px; margin-bottom:10px; }

h6 { font-size:12px; margin-top: 10px; margin-bottom:10px; }

p { margin-bottom: 10px; }

a { color:#337ab7; text-decoration:none; }

table { background: transparent }

form { margin-top: 0; }

div { }

span { }

8.Bootstrap第二部分：全局CSS样式 —— 按钮相关样式

.btn 基础class

.btn-default 白底黑字

.btn-danger 按钮的五种常用颜色

.btn-warning

.btn-success

.btn-info

.btn-primary

.btn-lg 按钮的四种大小

.btn-sm

.btn-xs

.btn-block 块级按钮

9.Bootstrap第二部分：全局CSS样式 —— 图片相关样式

.img-rounded 圆角图片

.img-circle 圆形图片

.img-thumbnail 缩略图片

.img-responsive 响应式图片

10.Bootstrap第二部分：全局CSS样式 —— 列表相关样式

.list-unstyled 取消提示符

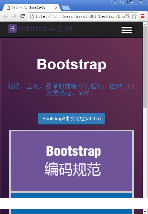
.list-inline 行内列表

课下作业：

(1)em和rem的区别？

(2)自学全局CSS样式——辅助类、文本和排版相关类、表格相关类

(3)使用媒体查询编写响应式网页（注意导航条和四张图片）：



day02

复习：

(1)响应式网页

声明viewport

容器使用相对尺寸

字体使用相对尺寸

图片使用相对尺寸

流式布局

使用CSS3MediaQuery

(2)Bootstrap是一个HTML/**CSS**/JS框架，适用于响应式网页开发。内容分为五部分：

起步

全局CSS样式

组件

jQuery插件

定制

今日目标：

(1)全局CSS样式 —— 重点在栅格布局！

(2)组件 —— 重点&难点：响应式导航条

1.Bootstrap屏幕尺寸划分

**(1)lg: large，超大PC屏幕**

w>=1200px

.container { width: 1170px; }

**(2)md: medium，中等PC屏幕**

1200px>w>=992px

.container { width: 970px; }

**(3)sm: small，平板屏幕**

992px>w>=768px

.container { width: 750px; }

**(4)xs: extra small，手机屏幕**

768px>w

.container { width: 100%; }

2.Bootstrap第二部分：全局CSS样式——按钮

.btn .btn-default

.btn-danger/warning/success/info/primary

.btn-lg/sm/xs

.btn-block

3.Bootstrap第二部分：全局CSS样式——图片

.img-rounded

.img-circle

.img-thumbnail

.img-responsiv

4.Bootstrap第二部分：全局CSS样式——列表

.list-unstyled

.list-inline

5.Bootstrap第二部分：全局CSS样式——辅助类

.pull-left

.pull-right

.clear-fix

.caret

.close

6.Bootstrap第二部分：全局CSS样式——文本和排版

h1 ~ h6

.h1~.h6

.text-五种颜色

.bg-五种颜色

.text-left/right/center/justify

.text-lowercase/uppercase/capitalize

7. Bootstrap第二部分：全局CSS样式——表格——小重点

.table

.table-bordered 带边框的表格

.table-striped 隔行变色的表格

.table-hover 带鼠标悬停效果的表格

.table-responsive 响应式表格，用于TABLE的父元素DIV上！

练习：——查看手册

(1)如何让tr或td显示为红色/绿色背景？ tr.danger td.success

(2) .table-responsive的正确用法？

8.Bootlint

Bootlint 是 Bootstrap 官方所支持的 HTML 检测工具。在使用了 Bootstrap 的页面上，它能自动检查某些常见的 HTML、CSS使用方面的错误。 使用方法：

<script>

(function () {

var s = document.createElement("script");

s.onload = function () {

bootlint.showLintReportForCurrentDocument([]);

};

s.src = "js/bootlint.js";

document.body.appendChild(s)

})();

</script>

9. Bootstrap第二部分：全局CSS样式——栅格布局系统——最重点

Web页面的布局方法有三种：

**(1)TABLE布局**

优点：简单，容易控制

不足：语义错误；渲染效率低

**(2)DIV+CSS布局**

优点：语义正确，渲染效率高

不足：控制起来比较麻烦

**(3)Bootstrap的栅格布局**

优点：简单，容易控制(概念上.row和.col)，语义正确，渲染效率高(实现上是div+css)，支持响应式

不足：没有！

栅格布局系统的使用：

(1)最外层需要使用容器

.container： 定宽容器

.container-fluid：变宽容器 width:100%

需要防止子元素margin越界；防止子元素浮动造成的影响

(2)容器中声明行：div.row，一行均分为12等份

(3)行中声明列：div.col，每个列都需要指定宽度占比 n/12

(4)针对不同的屏幕需要使用不同的列，如

.col-lg-1/2/3/.../12

.col-md-1/2/3/.../12

.col-sm-1/2/3/.../12

.col-xs-1/2/3/.../12

(5)栅格布局系统可以嵌套

<div class="container">

<div class="row">

<div class="col-\*-6">

<div class="row">

<div class="col-\*-2">

</div>

</div>

</div>

</div>

</div>

(6)注意不同屏幕下列的适用性问题

col-lg-\* 只在LG屏幕下有效

col-md-\* 在MD/LG屏幕下有效

col-sm-\* 在SM/MD/LG屏幕下有效

col-xs-\* 在XS/SM/MD/LG屏幕下有效

总结：列可以在当前屏幕已经更大的屏幕下有效

(7)响应式：一个div可以指定在不同的屏幕下的不同宽度占比

<div class="col-xs-9 col-sm-6 col-md-3"></div>

注意：小尺寸可以适配大尺寸，不要出现冗余代码

(8)无需使用空白占位列，使用列偏移即可

offset：偏移，让当前列以及后续的列向后错位

col-xs-offset-1/2/3/.../12

col-sm-offset-1/2/3/.../12

col-md-offset-1/2/3/.../12

col-lg-offset-1/2/3/.../12

注意：列偏移对当前屏幕以及更大的屏幕有效

(9)可以指定某个列在某个屏幕下隐藏

.hidden-lg

.hidden-md

.hidden-sm

.hidden-xs

.hidden

注意：列隐藏只对指定屏幕有效

(10)调整列顺序........

练习：使用Bootstrap，实现如下的响应式网页：

午间练习：

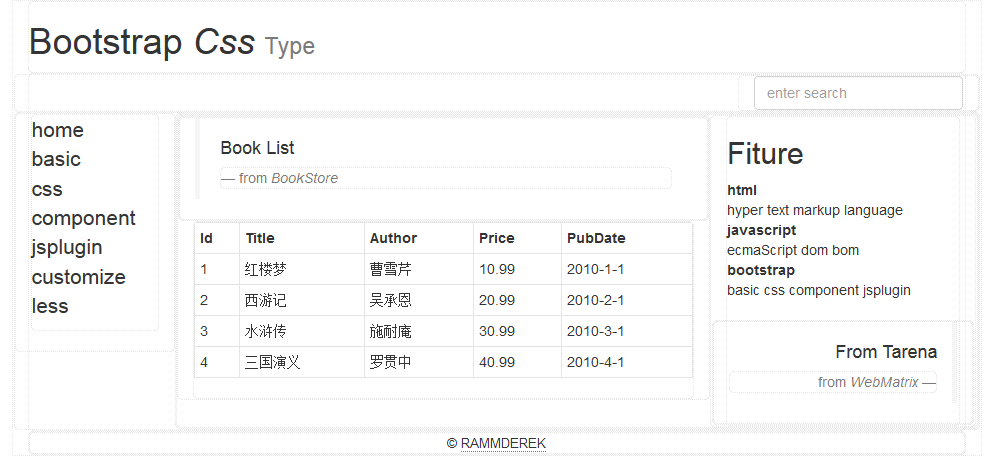
(1)如何把一行分为1:3:1三列？

方法1：定制栅格系统，让一行均分为10列

方法2：把1:3:1修改为1:2:6:2:1，左右各空出1/12

(2)练习：使用栅格布局实现如下的页面布局；

提示：<input class="form-control">， “文本和排版”章节有blockquote和dl>dt+dd的使用介绍。



10.Boostrap第三部分：组件 —— 下拉菜单

下拉菜单必需的三级结构：

<div class="dropdown">

<a data-toggle="dropdown">触发元素</a><span class="caret">x</span>

<ul class="dropdown-menu">隐藏元素

<li class="dropdown-header"></li>

<li class="dropdown-divider"></li> 分隔线

</ul>

</div>

提示：上述data-toggle="dropdown"会被bootstrap.js选中，添加事件监听。

11.Boostrap第三部分：组件 —— 警告框

<div class="alert alert-四种颜色 alert-dismissible">

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

警告消息

</div>

导航条(navbar) 和 导航(nav) 完全不一样！

12.Boostrap第三部分：组件 —— 导航(nav)

Bootstrap提供了三种形式的导航：

**(1)标签页式导航**

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab"></a></li>

<li><a data-toggle="tab"></a></li>

</ul>

**(2)胶囊式导航**

<ul class="nav nav-pills">

<li class="active"><a data-toggle="tab"></a></li>切换

<li><a data-toggle="tab"></a></li>

</ul>

**(3)导航条中的导航**

课后练习：英特尔中国官网首页

提示：只实现响应式布局，无需考虑蓝色的主题颜色。

day03

复习：

Bootstrap是一个HTML/**CSS**/JS框架，适用于响应式项目开发。内容涉及到：

CSS: (1)CSS Reset (2)1000+个class

HTML: 基于H5标签添加了扩展属性data-\*

JS: 十几个插件函数

分为五部分：

(1)起步

下载、模板、Bootlint、实例、兼容性

(2)全局CSS样式

按钮 .btn .btn-default .btn-颜色 .btn-lg/sm/xs .btn-block

图片 .img-rouned/circle/thumbnail/responsive

列表 .list-unstyled .list-inline

文本 .text-颜色 .bg-颜色 .text-uppercase/lowercase/capitalize .text-left/right/center/justify

辅助 .pull-left/right .clear-fix .caret .close

表格 .table .table-bordered .table-striped .table-hover .table-responsive

栅格布局

<div class="container">

<div class="row">

<div class="col-lg/md/sm/xs-1/2/.../12"></div>

</div>

</div>

(3)组件

下拉菜单

<div class="dropdown">

<a data-toggle="dropdown">触发元素</a>

<ul class="dropdown-menu">隐藏元素</ul>

</div>

警告框

<div class="alert alert-四种颜色 alert-dismissible">

<span data-dismiss="alert" class="close">&times;</span>

</div>

导航（标签页、胶囊式、导航条中的导航）

<ul class="nav nav-tabs/pills">

<li class="active"><a data-toggle="tab"></li>

</ul>

(4)插件

(5)定制

今日目标：

(1)Bootstrap第三部分——组件——难点：响应式导航条

(2)Bootstrap第四部分——插件——掌握

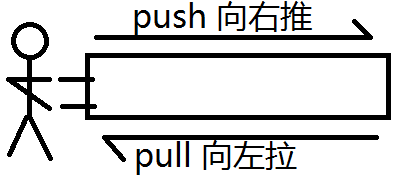
1.补充：栅格布局系统知识点：

(1)列偏移：让当前列及后续的列向右错位

.col-md-offset-\*

效果：实现1:3:1布局、左右分居布局、向右/中央定居布局

(2)列排序：仅向右或向左修改当前列的出现位置，不影响其它列



效果：让某列进入另一列内部、在特定屏幕下调整列的出现顺序

col-md-push-1/2..../12 向右推

col-md-pull-1/2..../12 向左拉

2.Bootstrap第三部分：组件—— 图标字体

******

图标字体不是图片，而是文字，所有对文字可以使用CSS样式都可以用到图标。Web项目常用的图标字体：

(1)FontAwesome - 675个图标，免费

(2)Glyphicons Halflings - 800+个图标，收费

服务器端字体的使用方法：

(1)在服务器上保存字体文件

(2)在CSS声明服务器端字体名称，并指定下载字体文件的URL

(3)定义class，使用刚刚声明服务端字体，浏览器会自动下载对应的字体文件

(4)在需要使用图标的地方声明对应的Unicode码即可——可以使用固定名称的class生成对应的Unicode码。

<span class="glyphicon glyphicon-\*"></span>

提示：每个span放置一个小图标，span中不能包含其它内容。

练习：编写带图标的小按钮，如下：

首页(房子)、配置(齿轮)、发邮件(信封)、用户(小人)、我喜欢打分(心形)、搜索(放大镜)、刷新、定位(向下水滴)、加号、减号、对勾、叉号、上一张、下一张、前进、后退、播放、暂停、上一首、下一首

午间自学组件：

面包屑(路径导航) .breadcrumb、

分页条 .pagination、

分页器、 .pager

标签、 .label

徽章、 .badge

巨幕、 .jumbotron

页头、 .pager-header

水井、 .well

进度条 .progress

3.Bootstrap第三部分：组件——按钮组

<div class="btn-group">

<a class="btn"></a>

.......

<a class="btn"></a>

</div>

4.Bootstrap第三部分：组件——缩略图

<a class="thumbnail"><img></a>

<div class="thumbnail">

<img>

<div></div>

</div>

练习：实现响应式的商品列表 PC：一行四列 PAD：二行二列 PHONE：四行一列，每个商品都需要图片、名称、价格、添加到购物车的按钮

5.Bootstrap第三部分：组件——列表组

ul.list-group > li.list-group-item

div.list-group >a.list-group-item

6.Bootstrap第三部分：组件——媒体对象

一个媒体对象(Media Object)用于介绍一个事物，分为“左中右”三部分，左右一般是图片，中部一般是说明文字

<div class="media">

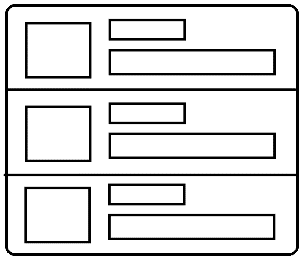
<div class="media-left"></div> 15:47

<div class="media-body"></div>

<div class="media-right"></div>

</div>

练习：实现如下的产品列表 提示：需要使用到“列表组”和“媒体对象”两个组件



7.Bootstrap第四部分：插件—— Collapse（折叠效果）—— 小重点

<adata-toggle="collapse" href="#xid"></a>

<button data-toggle="collapse" data-target="#xid"></button>

<div id="xid" class="collapse in">

<div class="外观相关class">内容</div>

</div>

折叠插件函数本身很简单，它的两个扩展应用比较复杂且重要：

(1)手风琴（Accordion） —— 比较难

(2)响应式导航条（Navbar） —— 很难&很重要！

8.Bootstrap第三部分：组件——面板

一个面板：就是框元素，分为“上中下”三级结构

<div class="panel panel-五种颜色">

<div class="panel-heading"></div>

<div class="panel-body"></div>

<div class="panel-footer"></div>

</div>

9.响应式导航条的写法

PC和PAD中所有的内容都会平展开来：



PHONE中才能体现其真正的结构：



导航条必须的结构：

<div class="navbar">

<div class="container">

<div class="navbar-header">

.navbar-brand + .navbar-toggle

</div>

<div class="navbar-collapse">

.navbar-nav + .navbar-text/link/btn/form

</div>

</div>

</div>

导航条的两种颜色：

(1)浅色底深色的字： .navbar-default

(2)深色底浅色的字： .navbar-inverse

导航条的两种定位：

(1)相对定位（占用布局高度）： 默认

(2)固定定位（不占用布局高度）： .navbar-fixed-\*

导航条的两种位置：

(1)页面顶部：相对定位 或者 .navbar-fixed-top

(2)页面底部：.navbar-fixed-bottom

课下练习：

(1)仿写“起步”->“实例精选”

(2)topspeeder的词库改为Bootstrap中所有的class

(3)难点：自学“组件”- 表单

试着为导航条中添加表单

(4)完成“英特尔首页”中的导航条 —— 完成整个首页效果！

难点：导航条中的导航 可以实现下拉菜单效果——仿手册

day04

复习：

见思维导图

今日目标：

(1)插件 —— 掌握

(2)Less语言 —— 重点

(3)定制 —— 掌握

1.补充全局CSS样式——表单

Bootstrap提供了三种形式的表单：

(1)默认表单



(2)行内表单



(3)水平表单

提示：水平表单是“栅格系统(变种)”+“表单”组合

|  |  |  |
| --- | --- | --- |
|  | 标准栅格系统 | 水平表单中的栅格系统 |
| 容器 | div.container | form.form-horizontal |
| 行 | .row | div.form-group |
| 列 | .col-\*-\* | .col-\*-\* |

2.Bootstrap第四部分——JS插件

Bootstrap基于jQuery提供了十几个插件函数，有两种调用方式

**(1)传统的JS调用**

$('a').dropdown();

**(2)data-\*扩展属性调用——官方推荐**

<a data-toggle="dropdown">

3.Bootstrap第四部分：JS插件 —— 下拉菜单

(1)JS方式调用

$('a').dropdown()

(2)data-\*方式调用

<a data-toggle="dropdown">

4.Bootstrap第四部分：JS插件 —— 警告框

点击close按钮，关闭警告框

<div class="alert">

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

</div>

5.Bootstrap第四部分：JS插件 —— 导航

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab" href="#">十元套餐</a></li>

</ul>

6.Bootstrap第四部分：JS插件 —— 折叠

<a data-toggle="collapse" href="target-id">展开/收起</a>

7.Bootstrap第四部分：JS插件 —— 工具提示

提示：所有插件函数中，只有工具提示和弹出框不单要声明data-\*属性，还要调用js函数！

<ANYdata-toggle="tooltip"title="">

$('[data-toggle="tooltip"]').tooltip()

8.Bootstrap第四部分：JS插件 —— 弹出框

提示：所有插件函数中，只有工具提示和弹出框不单要声明data-\*属性，还要调用js函数！

<ANYdata-toggle="popover"title="" data-content="">

$('[data-toggle="popover"]').popover()

|  |
| --- |
| Modal：模态框  Ｍodel：模型  Module：模块 |

9.Bootstrap第四部分：JS插件 —— 模态框 ——重点

Modal：若父窗口弹出了一个子窗口，子窗口只要不关闭，父窗口就无法获得输入焦点，这样的子窗口称为“模态子窗口”

**模态框必需的HTML结构：**

<div class="modal"><!--半透明的遮罩层-->

<div class="modal-dialog"><!--宽/定位层-->

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

<div class="modal-header">头部</div>

<div class="modal-body">主体</div>

<div class="modal-footer">尾部</div>

</div>

</div>

</div>

弹出模态框：

<a data-toggle="modal" href="#modal-id">

<button data-toggle="modal" data-target="#modal-id">

关闭模态框：

<span data-dismiss="modal">

10.Bootstrap第四部分：JS插件 —— 轮播广告 ——重点

Carousel：旋转木马，引申为轮播广告

提示：Bootstrap轮播广告结构比较复杂，无需记忆，依靠bootlint的提示即可一步步全部写出来！

<div class="carousel" **data-ride="carousel"**>

<div class="carousel-inner">

<div class="item">文字或图片</div>

</div>

</div>

11.样式语言的分类

**(1)静态样式语言**

CSS，可以直接被浏览器解释执行，很简单。作为一门语言，不完整的，缺少语言必需的因素，，导致CSS编写和维护异常麻烦——可维护性差。

**(2)动态样式语言**

在静态样式语言的基础之上添加了动态编程的特性，如数据类型、变量、运算、逻辑结构、函数、对象继承，从而大大的提高了样式代码的可维护性。常见的动态样式语言：

1)Sass / SCSS .sass => .css

2)Stylus .styl => .css

3)Less .less => .css

CSS是行业标准，可以被浏览器直接理解；其它样式语言，浏览器是不能直接解析的，必须使用工具程序转换为CSS代码——此过程称为“编译(Compile)”。

12.动态样式语言——Less使用环境的搭建

官网：http://lesscss.org/

中文教程：http://www.bootcss.com/p/lesscss/

LESS 将 CSS 赋予了动态语言的特性，如 变量， 继承， 运算， 函数. LESS 既可以在 客户端 上运行 (支持IE 6+, Webkit, Firefox)，也可以借助Node.js或者Rhino在服务端运行。

**(1)在客户端运行Less代码——不推荐使用，效率太低**

编写一个.less文件；

编写一个.html文件，引入.less文件；

在.html中再引入一个Less编译程序，less.js；

客户端在浏览html时，会自动下载less编译器和源文件，在客户端执行.js把less代码编译为css

**(2)在服务器端运行Less代码——推荐使用，效率与使用CSS一样**

编写一个.less文件;

开发者电脑上安装一款Less编译器；

开发者调用Less编译器，把.less转换为.css文件；

再编写.html文件，直接引入.css即可；

客户端浏览时下载html和css即可。

如何在开发人员的电脑上安装一款Less编译器：

(1)下载并安装一款JS解释器，如Node.js

node -v 可以显示出版本号

(2)下载并安装Less编译器

C:\npm\node\_modules\less\bin\lessc

(3)运行JS解释器，执行JS文件，编译.less文件得到.css文件

**nodeC:\npm\node\_modules\less\bin\lessc e:/2.less > e:/2.css**

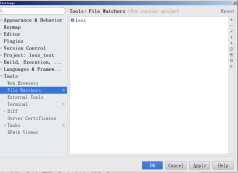
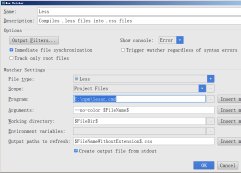
或者

**C:\npm\lessc.cmd e:/2.less > e:/2.css**

或者

把上述命令配置为WS中文件监视器(File Watcher)，只要用户创建或修改了.less文件，WS自动调用上述命令，把.less编译为.css文件——必须保证WS当前项目的编码方式为UTF-8!!!

File -> Settings -> Tools -> FileWatchers



13.Less语法

(1)Less支持所有的CSS语法

(2)CSS只支持多行注释；Less支持多行/单行注释，单行注释不会被编译到CSS中去

(3)Less支持变量（Variable）：

语法： @变量名: 值;

使用： color: @变量名;

变量值可以是任何合法的CSS样式值。

(4)变量和常量可以进行四则算术运算

如：+ - \* / %

(5)Less支持样式混入(Mixin)：

.选择器1 { ... }

.选择器2 {

....

.选择器1;

...

}

(6)样式混入时还可以指定参数

.选择器1(@变量名1, @变量名2) { ... }

.选择器2 {

....

.选择器1(值1， 值2);

...

}

(7)Less支持样式嵌套

选择器1 {

...

选择器2 {

....

}

}

编译后的结果：

选择器1 { ... }

选择器1 选择器2 { .... }

(8)Less提供了几十个实用function，用于编写样式

image-width(url) 返回指定图片的宽

image-height(url) 返回指定图片的高

ceil()

floor()

round()

lighten(color, 30%) 返回一个变浅后的颜色

darken(color, 30%) 返回一个变深后的颜色

(9)Less支持文件包含

CSS中也支持文件包含——不推荐使用——增加文件请求次数！

Less中的文件包含推荐使用——在服务器端把被包含的文件组成一个大的CSS文件，客户端只要请求一次即可。

Bootstrap的源代码分为多个.less文件，最终由bootstrap.less统一包含到一个文件中，生成 bootstrap.css！

12.定制Bootstrap

注意：绝不要手工修改bootstrap.css！！应该修改对应的Less源代码！

定制的目标有三个：

(1)瘦身：裁剪掉不需要的样式——注释掉bootstrap.less不需要的@import即可

(2)粗粒度修改组件的默认样式——修改variables.less中声明的变量的默认值

(3)细粒度修改某个组件的默认样式——修改对应的.less中的样式值，如修改.dropdown>.divider的margin，需要修改dropdown.less

课下任务：

(1)延伸学习Bootstrap插件—— affix、scrollspy

(2)定制英特尔首页需要的bootstrap.css，实现目标2和目标3