一、响应式网页

Responsive Web Page

1、什么是响应式网页

一个页面可以兼容各种终端，无需编写多个不同的版本，可以根据浏览设备(手机、平板、PC)的不同和特性而呈现出不同的布局方式，不影响用户浏览体验

2、响应式网页的构成

(1)流式(Fluid)页面布局

(2)可伸缩的图片和文字

(3)使用CSS3的Media Query技术

3、如何测试一个响应式网页

(1)使用真实的物理设备来测试

优点：测试结果真实可靠

缺点：需要购置大量的设备；测试工作量太大

(2)使用第三方的虚拟软件来测试

优点：无需购置设备

缺点：运行速度慢；部分功能不易测试(比如屏幕旋转等)；测试结果有待进一步的验证

(3)使用Chrome自带的设备模拟器来测试

优点：方便快捷

缺点：测试结果有待进一步的验证

4、如何编写一个响应式网页

(1)在<head>中声明viewport标签，并进行特殊的设定

即<meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no">

viewport，视口，最早由苹果公司在IOS系统中提出此概念，后来Android也引入，这是一个移动设备浏览器中专有的概念，PC浏览器不支持，会忽略

视口是一个虚拟的概念，用于显示网页内容的一个窗口，其宽和高都可以任意指定

所以有了视口之后，可以在较小的物理屏幕下，浏览较大的网页内容(只需要用户左右滑动)

width=device-width，表示宽度为适应设备宽

initial-scale=1，初始缩放比例为1

user-scalable=no，缩放功能关闭(shift鼠标上下滑动实现缩放)

这样设置后就使得Web页面在呈现上类似于原生的IOS/Android应用

\*移动开发中meta必须放在<head>里面的最前端

(2)所有的尺寸和字体大小都不能用绝对单位(如px)，而要使用相对单位(如%、em、auto)

(3)使用浮动定位float

(4)图片要实现自适应(比如使用%设置图片的style宽度)

img{

max-width:50%;

}

指定在父容器中的占比，同时必须保证不能超过图片的原始大小

(5)有选择性的执行某些CSS样式，忽略另一些 -- CSS3媒体查询技术

5、CSS3 Media Query技术

Media，媒体，指浏览网页的设备，如屏幕(手机、平板、PC)、打印机、投影仪、电视、字符终端(tty)(包括cmd)、盲文设备

Query，查询，查询出当前浏览设备的物理特性，如width、height、resolution(解析度)、orientation(方向)(landscape竖向/portait横向)

CSS3 Media Query，根据当前浏览设备类型，以及物理特性方面的不同而执行不同的CSS代码

\*CSS3媒体查询具体有两种用法：

(1)根据媒体特性执行不同的外部CSS文件

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

<link media="screen and (min-width:992px)" rel="stylesheet" href="css/pc.css">

只有媒体查询结果为真时，指定的样式文件才会被【执行】

缺点： 即使媒体查询失败的外部样式文件，也会被浏览器所【加载】

同时要引入多个外部样式文件

(2)在某段CSS样式声明中，只针对部分样式使用媒体查询

在CSS样式表中：

@media screen and (min-width:992px){... ...}

@media screen and (min-width:768px) and (max-width:991px){... ...}

\*使用了媒体查询技术后，页面中允许使用px等绝对单位了(因为会根据不同的浏览设备去设定不同的值)

二、Bootstrap起步

极大的简化了响应式网页的开发

中文网站：www.bootcss.com

BootstrapV2，PC浏览设备优先的响应式网页开发

BootstrapV3，移动设备优先的响应式网页开发

HTML5：为H5已有的标签扩展了若干个自定义属性

CSS：CSS Reset；近千个class

JS：基于JQuery添加了十几个插件函数

基本模板相关内容

1、html标签的lang属性

<html lang="zh-CN">

zh-CN的两个目的：

(1)为浏览器翻译应用指明网站原始的语言

(2)为屏幕阅读软件指定网页的基础发音

2、X-UA-Compatible元标签的含义

<meta http-equiv="X-UA-Compatible" content="IE=edge">

这段代码的作用是：在新版本的IE中启用旧IE的内核

X，Cross，跨越

UA，UserAgent，用户代理，即浏览器

Compatible，兼容性

edge，边沿、最前端

此元标签是IE浏览器专用的元标签，其它浏览器会直接忽略

3、html5shiv.min.js文件(非bootstrap专有的)

由第三方开发的一个JS自调函数，用于让老版本的IE(<9)支持HTML5新标签

4、respond.min.js文件(非bootstrap专有的)

由第三方开发的一个JS自调函数，用于让老版本的IE(<9)支持响应式网页开发必备的技术 -- CSS3MediaQuery

5、Bootlint，是Bootstrap官方提供的一个HTML、CSS检查工具，可以检查出HTML元素、CSS样式在使用、嵌套方面的【语义错误】

<script>

(function () {

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

s.onload = function () {

bootlint.showLintReportForCurrentDocument([]);

};

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

document.body.appendChild(s)

})();

</script>

Bootstrap把浏览器的屏幕大小分为四种：

超大PC屏幕(lg) >1200px

普通PC屏幕(md) 1200px<w<992px

平板电脑屏幕(sm) 992px<w<768px

手机屏幕(xs) <768px

三、全局CSS样式

1、CSSReset(原有样式重置)

(1) \*{

box-sizing:border-box;

}

使用border-box可以极大的简化盒子模型的宽度计算

\*CSS盒子模型的计算方法：

box-sizing:

content-box，默认值

总宽=margin+border+padding+width

border-box，推荐值

总宽=margin+width(border和padding算在width之内)

(2) html{

font-size:10px;

}

体现了移动设备优先

(3) body{

margin:0;

font-family:"Helvatica Neue";

color:#333;

}

(4) 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;

}

(5) p{

margin-bottom:10px;

}

(6) a{

color:#337ab7;

text-decoration:none;

}

a:hover{

color:#23527c;

text-decoration:underline;

}

(7) img{

border:0;

vertical-align:middle;

}

注：<a><img/></a>，img放在a里面，在旧版本IE里面是有蓝色边框的

2、按钮

.btn

.btn-default

.btn-danger，红色

.btn-success，绿色

.btn-warning，黄色

.btn-info，浅蓝色

.btn-primary，深蓝色

.btn-lg，超大号按钮

Large，lg，一般用于超大PC显示器

.btn-md或不写，普通号按钮

Medium，md(默认)，一般用于普通的PC显示器

.btn-sm，小号按钮

Small，sm，一般用于平板显示器

.btn-xs，超小号按钮

ExtraSmall，xs，一般用于手机显示器

.btn-block，块级按钮

3、图片

.img-rounded

效果相当于border-radius:6px

.img-circle

效果相当于border-radius:50%

.img-thumbnail

缩略图片，通过border、padding、max-width实现

.img-responsive

响应式图片，display:block、max-width:100%

4、文本

文本字体颜色

.text-danger

.text-success

.text-warning

.text-info

.text-primary

文本背景颜色

.bg-danger

.bg-success

.bg-warning

.bg-info

.bg-primary

文本对齐

.text-left，左对齐

text-align:left来实现

.text-right，右对齐

.text-center，居中对齐

.text-justify，两端对齐(只限多行英文)

文本变形

.text-uppercase，全部大写

.text-lowercase，全部小写

.text-capitalize，每个单词首字母大写

行内块的对齐

.pull-left

float:left !important来实现

.pull-right

float:right !important来实现

5、排版和代码

ul

.list-unstyled

去掉前面的序号或者小圆点

.list-inline

排列在同一行

<blockquote>

见6.html

6、表格

.table

.table-bordered，带边框的表格

.table-striped，条状/隔行变色的表格

.table-hover，带悬停效果的表格

.table-responsive，响应式表格

\*此class必须用在table的父元素div上

7、栅格布局系统

(1)页面做布局的几种方式

①Table布局

优势：简单、不易出错

缺点：显示速度慢、语义不符合

②Div+CSS布局

优势：易扩展、渲染速度快

缺点：不易控制、易出错

③Bootstrap提供的响应式布局 - 栅格布局系统

优势：前两种布局的优势集合+响应式布局功能

同样有行/列的概念，但不是通过table来实现，而是通过div来实现

缺点：无

(2)栅格布局系统的使用细则

①系统提出了行(row)和列(col)的概念，一行默认平均分为12列

②行中只能放置列(不能放置内容)，内容只能放置在列中

③列中可以继续嵌套行

④行必须放在容器中

Bootstrap默认提供了两种容器：

.container，固定宽度的容器

lg屏幕下：容器宽1170px

md屏幕下：容器宽970px

sm屏幕下：容器宽750px

xs屏幕下：容器宽auto

margin-left/right:auto;

padding-left/right:15px;

:before，防止子元素的margin越界

:after，防止子元素的margin越界，清除子元素浮动对父元素和后续元素造成的影响

.container-fluid，流式容器，宽度随着父容器的改变而改变

width:auto;

margin-left/right:auto;

padding-left/right:15px;

:before，防止子元素的margin越界

:after，防止子元素的margin越界，清除子元素浮动对父元素和后续元素造成的影响

⑤行必须声明class为.row，列必须声明class为.col-xs/sm/md/lg-\*

\*为一个1-12之间的数字，意为指定某个列的占宽(总共有12列)

⑥四种大小的屏幕中的列

.col-xs-\*对如下屏幕都有效：

xs/sm/md/lg

.col-sm-\*对如下屏幕都有效：

sm/md/lg

.col-md-\*对如下屏幕都有效：

md/lg

.col-lg-\*对如下屏幕都有效：

lg

⑦可以为同一个div指定在不同屏幕下的不同的宽度变化

<div class="col-md-6 col-sm-8 col-xs-12">

灵活变宽的列：PC中6/12；PAD中8/12；PHONE中12/12

</div>

⑧可以将列在某个屏幕下设置为隐藏

.hidden-xs，只在手机屏幕下隐藏

.hidden-sm，只在平板屏幕下隐藏

.hidden-md，只在PC屏幕下隐藏

.hidden-lg，只在超大PC屏幕下隐藏

⑨某个列可以使用偏移实现向后错位的效果

.col-xs/sm/md/lg-offset-\*

对当前屏幕以及更大的屏幕都有效

(3)栅格系统中调整列的位置/顺序

①方式1：使用列的偏移(offset)

\*列偏移会影响后续所有的列

②方式2：使用列顺序的调整

\*列顺序的调整仅影响当前列

.col-xs/sm/md/lg-push-\* 向右推

.col-xs/sm/md/lg-pull-\* 向左拉

推拉操作可以对当前屏幕及更大的屏幕有效，对更小的屏幕无效

8、响应式表单

Bootstrap编码规范要求：

所有的输入控件都必须有相对应的label标签

如果不想在界面中看到输入控件label，可以声明Screen Reader Only(仅屏幕阅读器可见)

<label class="sr-only"></label>

Bootstrap提供了三种样式的表单(见第三天-实例1)

(1)默认样式的表单

每个输入域独立占一整行

单选框、复选框必须使用label包含起来，省略for属性

格式 <form>

<div class="form-group">

<label for=""></label>

<input class="form-control"/>

<span class="help-block"></span>

</div>

</form>

(2)行内(inline)表单

所有输入域都处于一行中

格式 <form class="form-inline">

<div class="form-group">

<label for=""></label>

<input class="form-control"/>

</div>

</form>

(3)水平(horizontal)表单

标签文字与对应的输入域在同一行

水平表单必须与(变形后的)栅格系统组合使用

变形后的栅格系统：

.container变为.form-horizontal

.row变为.form-group

.col-\*-\*依然是.col-\*-\*

\*注： 若想label中的文字左对齐，则col-\*-\*声明在div上

若想label中的文字右对齐，则col-\*-\*声明在label上

格式 <form class="form-horizontal">

<div class="form-group">

<label class="col-\*-\*" for=""></label>

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

<input class="form-control"/>

</div>

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

<span class="help-block"></span>

</div>

</div>

</form>

四、组件

1、图标字体

与普通字体(如黑体、宋体)一样，图标字体【本质也是字体】，不是图片

凡是能使用文字的地方，就可以使用图标字体

可以无限放大、加粗、变斜体、改变颜色

它呈现的内容把每个字符对应为一个图标

Glyphicons Halflings

它是Bootstrap提供的一套专用于Web开发/移动开发常用的图标字体

通常的Web客户端是没有安装此套字体的，Bootstrap把这套字体设置为服务器端字体，若客户端要显示这些图标，会自动从服务器下载这些字体文件

在bootstrap.css文件中：

@font-face { //声明一种服务器端字体

font-family: 'Glyphicons Halflings';

src: url('../fonts/glyphicons-halflings-regular.eot');

}

.glyphicon { //使用服务器端的字体

font-family: 'Glyphicons Halflings';

}

2、按钮组

<div class="btn-group">

.btn

... ...

.btn

</div>

3、下拉框

下拉菜单必需的结构：

<div class="dropdown"> 父元素(相对定位)

<a data-toggle="dropdown"></a> 触发元素

<div/ul class="dropdown-menu"></div/ul> 隐藏元素(绝对定位)

</div>

相关class：

.dropdown-header，字体变形

.divider，横线

4、导航

Bootstrap提供了两种形式的导航：

(1)标签页式导航(tab)

<ul class="nav nav-tabs">

<li><a></a></li>

<li class="active"><a></a></li> //激活状态

<li><a></a></li>

</ul>

(2)胶囊式导航(pill)【横置】

<ul class="nav nav-pills">

<li><a></a></li>

</ul>

栈式导航(胶囊式导航的引申)【纵置】

<ul class="nav nav-pills nav-stacked">

<li><a></a></li>

</ul>

5、警告框

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

<span data-dissmiss="alert" class="close">&times;(叉号)</span>

文本内容

</div>

.close元素必须是alert的第一个子元素

6、媒体对象

常用于显示用户的评论、帖子、商品列表

结构可分为三部分：左(可选)、主体、右(可选)

<div class="media">

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

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

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

</div>

7、列表组

提供整体外边框+每个条目的下边框的一个集合组件

<ul class="list-group">

<li></li>

...

</ul>

或<div class="list-group">

<a></a>

...

</div>

8、响应式导航

响应式导航条的两种状态：

(1)全部展开状态(在lg/md/sm屏幕)

LOGO、菜单、按钮、链接、表单、普通文本都处于同一行中

(2)收缩状态(在xs屏幕)

可见到头部(包括LOGO、折叠菜单触发按钮)和折叠菜单(包括菜单、链接、表单、文本等)两部分

响应式导航条的分类：

按颜色分类

浅色底深色字：.navbar-default

深色底浅色字：.navbar-inverse

按定位方式分类

静态定位：position值为relative(默认值)

固定定位：.navbar-fixed-top .navbar-fixed-bottom

\*导航条包含容器(.container)，而不是容器包含导航条

若LOGO是图片，必须手工控制它的位置(例如通过margin-top)

9、常用小组件

面包屑： .breadcrumb

分页： .pagination .pager

标签： .label

徽章： .badge

巨幕： .jumbotron

页头： .page-header

水井： .well

五、JS插件

Bootstrap提供的JQuery插件函数介绍

共有12个插件函数，它与JQueryUI提供的插件函数的地位是一致的，功能也有一定的重合

调用这12个插件函数有两种方式

(1)编写JS方式调用(传统方式)

$('xxx').插件函数();

(2)声明data-\*扩展属性来调用插件函数

例如 <a data-toggle="dropdown">

1、过渡效果

2、下拉菜单

<div class="dropdown">

<a data-toggle="dropdown"></a>

<div/ul class="dropdown-menu"></div/ul>

</div>

3、警告框

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

<span data-dissmiss="alert" class="close">&times;(叉号)</span>

文本内容

</div>

可以通过为.alert元素添加.fade.in两个class实现关闭时动画效果(渐变消失)

<div class="fade in alert alert-danger alert-dismissible">

4、标签页式导航

为a添加data-toggle="tab"属性，同时href属性指向对应的面板的id即可

通过为面板添加.fade.in两个class实现切换动画效果

见Unit4 练习3

5、弹出提示框

Bootstrap提供了三种形式的提示弹出框：

(1)工具提示(tooltip)

data-toggle="tooltip"+函数调用

(2)弹出框(popover)

data-toggle="popover"+函数调用

(3)模态弹出框(modal)

Web开发中，经常使用模态弹出框代替window.alert() / confirm() / prompt()

模态：在父窗体中弹出一个模态子窗体，则只要子窗体不关闭，父窗体无法获取焦点，且父子窗体间可以传递数据

模态弹出框必需的结构：

<div class="modal"> 固定定位的半透明遮罩层

<div class="modal-dialog modal-lg"> 定位、宽高

<div class="modal-content"> 背景、边框、阴影

<div class="modal-header">头部</div>

<div class="modal-body">主体</div>

<div class="modal-footer">尾部</div>

</div>

</div>

</div>

弹出一个模态框：data-toggle="modal"

关闭一个模态框：data-dismiss="modal"

通过为.modal元素添加.fade实现出现/隐藏动画效果

6、折叠效果(collapse)

触发元素的折叠/展开：data-toggle="collapse"

<a data-toggle="collapse" href="#目标元素ID">控制元素</a>

<div class="collapse [in]" id="目标元素ID">折叠组件的内容</div>

in，可有可无，有表示隐藏，无表示显示

折叠效果有两个重要的扩展应用：

(1)手风琴(Accordion)

多个.panel+.collapse的组合

(2)响应式导航条(Navbar)

响应式导航条中，可以放置多个组件(比如nav、form、link、button、text等)；但默认情况下，由于导航条有高度(50px)，这些内容中的文本在竖直方向都不会居中，解决方法：

添加对应的class：navbar-nav、navbar-form、navbar-link、navbar-button、navbar-text

7、轮播广告 -- Carousel

轮播广告必须有唯一的一个内容元素.carousel-inner

\*前进、后退按钮不随着广告而滚动，不放在carousel-inner里面

\*序号指示器不随着广告而滚动，不放在carousel-inner里面

必须的结构：

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

<div class="carousel-inner">

<div class="item active">

<img src="img/1.jpg" alt="">

</div>

<div class="item">

<img src="img/2.jpg" alt="">

</div>

</div>

</div>

8、附加导航(Affix)

一般是固定在页面左侧/右侧的次要的导航，往往只是在页面滚动到一定位置时才开始启用它的固定定位

data-spy="affix"

9、滚动监听(ScrollSpy)

基于附加导航实现滚动监听

六、Bootstrap定制

1、使用Bootstrap的项目如何定制需要的样式

(1)编写自己的.css文件，覆盖Bootstrap提供的默认样式

缺点： 意味着客户端需要下载大量无用的样式；

自定义的选择器可能不好覆盖Bootstrap编写的选择器

(2)直接修改bootstrap.css文件，编写需要的样式

缺点： 任务量巨大，需要在6800行代码中寻找需要的样式

(3)直接修改Bootstrap框架的源文件：bootstrap.less

2、动态样式语言

静态样式语言：CSS

动态样式语言：在CSS基础上，添加了"变量、运算、函数、继承、选择"等动态语言才具备的能力，从而让样式文件更加容易编写和维护

\*目前浏览器只支持CSS样式语言，其他的动态样式语言必须转换为浏览器能理解的CSS样式

此过程称为“编译”

3、Less使用环境的搭建

Less的语法不能直接被浏览器所理解，必须要转换(编译)

Less文件使用有两种方法：

(1)在客户端使用Less

在HTML页面中引入.less文件，再引入一个Less转换程序(是一个.js文件)，让客户端同时下载.html/.less/.js转换器，在客户端浏览器中运行.js把.less转换为CSS样式

(2)在服务器端使用Less -- 推荐使用

开发人员使用一个命令行工具(Less编译器)，把.less编译为.css，在HTML页面中，直接引入编译好的.css文件即可

4、Less编译器使用步骤

(1)下载并安装一款服务器端的js解释器：Node.js

(2)下载并安装Less编译器程序：lessc

它是一个运行在服务器端的js脚本程序

(3)运行Node，执行lessc转换程序，把一个.less文件编译为一个.css文件即可

具体有两种方法：

①直接在命令行中执行node lessc xx.less>yy.css

②(推荐方法)把lessc编译器配置为WebStorm中的FileWatcher，只要用户在WebStorm中创建/保存了一个.less文件，即会自动编译出对应的.css文件

WebStorm中设置：

File→Settings→Tools→File Watchers→右上角加号→less→Watcher Settings→Program→选择路径... ... lessc.cmd

\*每次新建项目都要重新配置，因为它只对当前项目有效

5、Less的基本语法

官网 http://www.lesscss.net

(1)Less完整的支持所有的CSS语法，即.less文件中可以编写任意合法的css代码

(2)Less支持单行(//...)和多行(/\*...\*/)注释，但只有多行注释会被编译到.css文件中去

(3)特性

①变量(Variable)

语法：

@变量名:值;

使用：

.class{属性:@变量名;}

②混合(Mixin)

在一个样式声明中混入其他的样式

语法及用法：

.class1{...}

.class2{

.class1;

...

}

③带参数的混合

语法：

.class1(@变量1,@变量2, ...){ ... }

.class2{

.class1(实参1,实参2, ...){ ... }

}

④规则嵌套

语法：

.class1{

...

.class2{ ... }

...

}

编译后的结果：class1里面包含一个class2样式

\*在嵌套中，&代替this

⑤运算

任何数字、颜色或者变量都可以参与运算

例如 @a:5%;

@b:@a\*2;

@b的值为10%

@c:@a+@b;

@c的值为15%

color1:#888/4;

color2:color1+#111;

color2的值为#333

@var:1px+5;

它的值为6px

⑥内置函数

Less内置了多种函数用于转换颜色、处理字符串、算术运算等

ceil(参数)，上取整

floor(参数)，下取整

hsl(色相，饱和度，亮度)，把一个HSL颜色转换为rgb颜色

lighten(颜色，亮度)，把指定的颜色变亮指定的百分比

darken(颜色，亮度)，把指定的颜色变暗指定的百分比

例如 @jd-red:#e4393c;

@dark-red:darken(@jd-red,10%);

意为在京东红的基础上变暗10%

⑦导入

在传统的CSS文件中，也可以实现文件的导入：

@import "xxx.css"

注：CSS文件中声明@import会导致浏览器在加载完当前css之后再次发起请求，获取服务器端另一个css文件，增加请求次数，且减慢页面渲染效率，所以不推荐使用

但是，Less文件中的@import "xxx.css"可以实现在服务器端就把被引入的文件内容拼合在当前生成的.css中，多个这样的@import最终可以生成一个大的完整的css文件，供客户端一次性的全部下载，不存在性能问题，推荐使用！

6、样式定制

通过修改Bootstrap的Less源代码实现样式定制

Bootstrap定制主要工作包含如下三个任务：

(1)删除不需要的组件和插件

修改bootstrap.less，把不必要的@import注释掉即可

(2)修改需要的组件默认的样式

修改variables.less，修改特定的Less变量值

(3)深度定制某个组件的实现细节

修改该组件对应的.less文件，比如dropdown.less