**Bootstrap**

**1.1响应式/自适网页：**

**可以兼容不同的版本**

**1.2响应式网页的构成**

**编写响应式网页时，应该在如下方面加以注意：**

1. **流式网格布局（Fluid : 流式，流动式。液态）**
2. **可伸展的图片和文字。**
3. **Css3 Media Query技术，根据浏览设备的高和宽调用不同的CSS**

**1.3响应式网页的特点：**

**(1) 流动布局。**

**(2) 可伸缩的图片和字体。**

**(3) CSS3 Media Query技术，根据浏览设备的高和宽调用不同的css。**

**1.4如何测试响应式网页**

**1.4.1 编写响应式网页后，应该如何测试响应式网页：**

**（1）使用真实物理设备测试---测试效果最好但最麻烦（效果可靠但任务量大）**

**只要手机/平板/PC在同一个局域网/互联网即可测试**

**（2）使用第三方测试/模拟软件测试--效果有待进一步的验证。**

**（3）使用Chrom自带的浏览器模拟器测试**

**优势：可以模拟常见的设备，网速，经纬度，加速度...**

**不足:效果有待进一步验证。**

**（3）使用浏览器提供的设备模拟器测试----最简单但有时测试效果与真实物理设备有所不同，需要进一步验证。**

**作业**

想要让自己设计的网页可以通过互联网来进行访问一般需要以下几个步骤：  
1.如果你用DW做的是静态网页，比如.html/.htm的网页的话，那么使用一个静态虚拟空间即可（无数据库）  
2.在网上找一下免费空间服务商，通过服务商获得空间用FTP工具 把程序上传到空间。（到这一步就可以通过IP地址就可以访问网站）  
3.买域名。（要钱）  
4.把域名解析到你的虚拟主机。  
5.通过ip或者域名让网民访问网站地址。   
  
注意：如果有朋友有服务器，可以让朋友资助一下，分配一个2级域名，那样就可以免费用域名访问咯。

**1.4 编写响应式网页**

**Query : 查询出当前浏览器设备的特向，如 类型，宽度，高度，分辨率。色彩位深，方向Orientation (Landscapo/Portrait),根据这些特性，执行特定的css样式。**

**CSS3MediaQuery 有两种用法:**

1. **根据媒体的特性，执行不同的外部css：**

**<link media=”screen and (min-width:768px) and (max-width:990px)” rel=”stylesheet” href=”xxx.css”>**

**不足：客户端会不管媒体特性，请求所有的css**

**(2) 根据媒体的特性，执行某段CSS中部分内容。**

**@media “screen and (min-width:768px) and (max-width:990)”{**

**h1{**

**.box{**

**}**

**}  
 }**

**二 Bootstrap**

**2.1概念**

**简洁，直观，强悍的前端开发框架，让web开发更迅速，简单。**

**Bootstrap 是最欢迎的html ,css ,js 框架，它用于开发响应式布局，移动设备优先的web项目。**

**注意：jQuery是一个JS框架，提供了很多的函数，简化了DOM操作。**

**jQueryUIP:实现了手风琴效果....**

**是一个框架，是一个html组件库**

**2.2 Bootstrap 分为五部分：**

**（1）起步（Startup）**

**（2）全局CSS样式（Global CSS）**

**（3）组件（Component）**

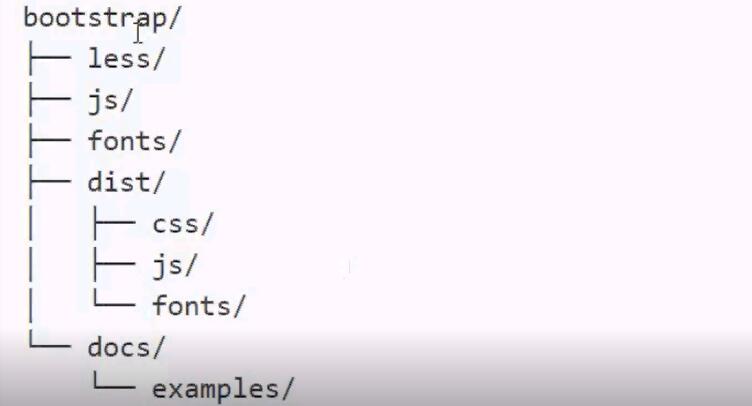
**（4）插件（Plugin）**

**（5）定制（Customine）**

**2.2.1**

**2.1.2 bootstrap 源码**

**Bootstrap源码包含了预先编译的CSS，javascript和图标字体文件，并且还有LESS,Javascript和文档的源码，具体来说，主要文件组织结构如下：**

****

**html5shiv.js**

**由@afarkas @jdalton @jon\_neal @rem 编写的一个JS自调用脚本，用于让老IE支持H5新标签**

**Respond.js**

**由Sott 编写的一个JS自带哦用脚本，用于让老IE浏览器支持CSS3MediaQuery技术，从而实现响应式网页的编写**

**2.2**

**V2：面向PC进行了样式设定，同时兼顾了PADhe PHONE**

**V3: 面向Phone进行样式设定，同时兼顾了PC**

**2.3 主要涉及：**

1. **HTML:为H5已有的标签扩展了一些自定义属性。**
2. **CSS :CSS Reset + 几千个class**
3. **JS :基于jQuery提供了十几个插件函数。**

**2.4内容分为五个部分：**

1. **起步：下载，模板，Bootstrap，禁用响应式布局。**
2. **全局CSS样式**
3. **组件**
4. **插件**
5. **定制**

**2.4.2 全局CSS样式之按钮**

**<button class="btn btn-default">BTN按钮</button>**

**<input class="btn btn-default" type="button" value="INPUT按钮"></input>  
<a class="btn btn-default" href="#">LINK按钮</a>**

**.btn{padding:; border:;}**

**.btn-default{color:; background:; border-color:;}**

**.btn-danger**

**.btn-warning**

**.btn-success**

**.btn-info**

**.btn-primary**

**.btn-lg**

**.btn-sm**

**.btn-xs**

**.btn-block**

**.pull-left{float:left;}**

**.pull-right{float:right;}**

**2.4.3 全局CSS样式之图片**

**.img-rounded**

**.img-circle**

**.img-thumbnail 缩略图片/拇指图片**

**.img-responsive 响应式图片**

**2.4.4 全局CSS样式之排版和代码 仅作了解**

**.text-danger**

**.text-success**

**.text-warning**

**.text-info**

**.text-primary**

**.bg-danger**

**.bg-success**

**.bg-warning**

**.bg-info**

**.bg-primary**

**.text-left**

**.text-right**

**.text-center**

**.text-justify 文本两端调整对齐**

**改变大小写**

**.text-uppercase**

**.text-lowercase**

**.text-capitalize**

**列表相关样式**

**.list-unstyled**

**.list-inline**

**Bootlint工具**

**是一个JS ，由Bootstrap官方提供，用于检测使用Bootstrap的页面中常见的html错误，class使用方面的错误----默认情况下浏览器是检查不出来的。**

**2.4.4 全局CSS样式之表格**

**.table**

**.table-border 带边框的表格**

**.table-responsive 响应式表格 注意：使用在table的父元素上，而不是table**

**.table-striped 隔行变色的表格**

**.table-hover 带悬停效果的表格**

**2.4.5 全局CSS样式之栅格布局系统 -重点&稍难**

**Web开发中页面布局可以采用的方式：**

1. **使用 table布局**

**优势：简单不易出错 不足：加载效率**

**-过时了**

1. **DIV + CSS 布局**

**优势：加载速度快，灵活 不足：不易控制**

1. **使用Bootstrap 提供的栅格(Grid Layout)布局系统**

**优势：加载速度快，灵活，支持响应式功能，容易控制（有行和列的概念，但是用DIV+CSS实现 兼容了DIV+CSS布局的优势还能实现响应式网页的布局）**

**2.4.5.1 栅格布局系统的特点：**

**（1）所有的行必须放在容器中： .container-fluid**

**（2）分为多行（row）,一行平均分为12列（col）**

**（3）网页内容只能放在列 (col) 中，不能直接放在行（row）**

**(4) 可以在col中再嵌套row**

**(5) col 分为四大类 :col-xs col-sm col-md col-lg**

**(6) col-md-\* \*值可以为1-12 ，值就为某个列的宽度（\*/12）**

1. **可以为一个列指定不同屏幕下的不同宽度。**
2. **col-lg-\* 只对大PC屏幕有效**

**col-md-\* 对普通PC,大PC 屏幕有效**

**col-sm-\* 对平板，PC，大PC屏幕都有效**

**col-xs-\* 对手机，平板，PC大PC屏幕都有效**

1. **.hidden-lg 当前列只在大PC屏幕下隐藏**

**.hidden-md 当前列只在普通PC屏幕下隐藏**

**.hidden-sm 当前只在平板屏幕下隐藏**

**.hidden-xs 当前列只在手机屏幕下隐藏**

**.containner 的宽度问题：**

**当屏幕的宽度 > 1200px （超大PC显示器 -lg） : 容器的宽1170px**

**当屏幕的宽度 > 992px （普通PC显示器 -md）: 容器的宽970px**

**当屏幕的宽度 > 768px （平板显示 -sm） : 容器的宽750px**

**当屏幕的宽度 < 768px （手机显示器 -xs） : 容器的宽auto**

**.container-fluid的宽度：width:auto;+before+after;**

**!!! 面试题：Bootstrap布局系统中容器的特点？**

**(1)宽度做了媒体查询**

**(2)添加了前置和后置内容生成，可以防止资源组的越界，浮动造成的影响**

**2.4.5.2全局CSS样式--栅格布局系统 补充：列的偏移问题（offset)**

**.col-xs-offset-1~ .col-xs-offset-12 在lg/md/sm/xs屏幕下偏移**

**.col-sm-offset-1~ .col-sm-offset-12 在lg/md/sm屏幕下偏移**

**.col-md-offset-1~ .col-md-offset-12 在lg/md屏幕下偏移**

**.col-lg-offset-1~ .col-lg-offset-12 在lg屏幕下偏移**

**补充CSS知识点**

1. **如何解决第一个子元素的margin-top越界问题**

**1）为父元素加border-top:1px; ----- 有副作用**

**2） 为父元素指定padding-top:1px; ----- 有副作用**

**3） 为父元素指定overflow:hidden; ---有副作用**

**4） 为父元素添加前置内容生成：---推荐使用 （无副作用）**

**.parent : before {**

**content: ‘’;**

**display:table;**

**}**

1. **如何解决所有的子元素浮动后父元素高度变为0？**

**1）为父元素指定overflow:hidden ; --有副作用**

**2） 为父元素指定高度：height:xxx; --有局限性**

**3）为父元素添加后置内容生成：---推荐使用 （无副作用）**

**.parent : after{**

**content: ‘’;**

**display:table;**

**clear:both;**

**}**

**2.4.6 全局CSS样式 -表单 --次重点&难点**

**bootstrap中的表单分为三种**

**2.4.7全局CSS样式之文本**

**.text-五个颜色**

**.bg- 五个颜色**

**.text-left/right/center/justify**

**.text-upper/lowercase/capitalize**

**.pull-left/right**

**2.5 组件**

**2.5.1 组件--图标字体 --简单**

**Glyphicons.com 网站**

**Glyphicon是一套收费的图标字体，提供了Web/移动开发中常有的小图标**

**bootstrap中可以免费使用这套字体中的250+个。以服务器端字体形式出现的，即客户端若访问了使用Glyphicons字体的网站，会自动从服务器端下载对应的字体文件。**

**提示：**

**1. 体表字体本质不是图片，而是字体；所以凡是可以使用文字的地方都可以使用不同字体。**

**2. glyphicon图标字体只能用于“空元素”---不包含任何内容或子元素！**

**2.5.2 组件-按钮组**

**搜索框中放置的图标，成为“反馈图标”**

**.has-feedback**

**2.5.3 组件-下拉菜单 次重点&简单**

**.dropdown**

**.dropdown**

**data-toggle=”dropdown”**

<**h1**>组件--下拉菜单</**h1**>  
<**div class="container"**>  
 <**div class="dropdown"**>  
 <**a data-toggle="dropdown" class="btn btn-default" href="#"**>产品大全  
 <**span class="caret"**></**span**>  
 </**a**>  
 <**ul class="dropdown-menu"**>  
 <**li**><**a href="#"**>电视</**a**></**li**>  
 <**li**><**a href="#"**>冰箱</**a**></**li**>  
 <**li**><**a href="#"**>空调</**a**></**li**>  
 <**li**><**a href="#"**>书架</**a**></**li**>  
 </**ul**>  
 </**div**>  
</**div**>

**2.5.4组件---导航---小重点**

**提示：此导航不是指导航条！！！**

**bootstrap提供了两种形式的导航：**

1. **标签式导航**

**<ul class=”nav nav-tabs”></ul>**

1. **胶囊式导航**

**<ul class=”nav nav-pills”></ul>**

**此外，还有两种导航变种：**

1. **两端对齐的导航 .nav.nav-tabs/pills.nav-justified**
2. **竖直放置的胶囊导航 .nav.nav-pills.nav-stacked**

**2.5.5 组件-自学：路径导航（面包屑）/分页/标签/徽章/巨幕/水井/页头**

**面包屑： .breadcrumb**

**分页 :**

**标签 : .badge**

**徽章**

**巨幕 : .jumbotron**

**水井 : .well**

**页头 : .page-header**

**2.5.6 组件-响应式导航条 --重点&难点**

**注意：导航条一般不放在容器中**

**响应式导航条：在PC和平板中默认要显示所有的内容；但在手机中默认只显示”LOGO/Brand” ,以及一个”菜单折叠展开按钮”，只有在单击折叠按钮后才显示所有的菜单项。**

**Bootstrap 中的导航条的按位置分：**

**（1）顶部导航条**

**（2）底部导航条**

**Bootstrap 中导航条按颜色分：**

**（1）浅色底深色的字 .navbar navbar-inverse**

**（2）深色底浅色的字**

**Bootstrap 中导航条按定位分：**

**（1）相对定位 position :relative 默认值**

**（2）固定定位 navbar-fixed-top/bottom**

**导航条的结构：**

**<div class=”navbar 颜色 定位”>**

**<div class=”container”>**

**<!-- 导航条的头部：商标+按钮-->**

**<div class=”navbar-header”>**

**<a class=”navbar-brand”>**

**<button class=”navbar-toggle”>**

**</div>**

**<!-- 导航条的折叠菜单：菜单,按钮，搜索框，链接，文本-->**

**<div class=”navbar-collapse”>**

**<ul class=”nav navbar-nav”>**

**<form class=navbar-form”>**

**<button class=”navbar-btn”>**

**<span class=”navbar-text”>**

**<a class=”navbar-link navbar-text”>**

**</div>**

**</div>**

**</div>**

**行业小知识：**

**盲人，视力低下人，行动障碍**

**H5标签中有两类属性与“无障碍阅读”相关：**

1. **<ANY role=””> <b role=”button”></b>**
2. **<ANY aria=””>**

**2.6 插 件**

**2.6.1概述：**

**Bootstrap基于jQuery提供了十几个插件函数（类似于jQueryUI插件库），每个插件对应一个.js文件 ,可以单独引用，也可以整体引用 (bootstrap.js）。**

**调用上述十几个插件可以用两种格式**

1. **传统的JS返航时调用：￥（...）.dropdown(); $().tab(...);**
2. **使用data-\*扩展属性方式调用：<a data-toggle=”dropdown”>**

**2.6.2 插件 - 标签页**

1. **$().tab();**
2. **<a data-toggle=”tab”>**

**2.6.3 插件 - bootstrap提供的弹出框**

**(1) 工具提示框(tooltip) data-toggle=”tooltip”**

**(2) 弹出框（popover） data-toggle=”popover”**

**(3) 警告框（alert） ----小重点**

**为关闭按钮添加data-dismiss=”alert” 属性就可以使其自动为警高框，也就是将其从DOM中删除。**

**<button type=”button” class=”close” data-dismiss=”alert” aria-label=””>**

**<span aria-hidden=”true”>&times;</span>**

**</button**

**为了让警告框在关闭时表现出动画效果，请确保为其添加了 .fade和**

**<div class=”alert alert-四种颜色 alert-dismissible”>**

**<span class=”close” data-dismiss=”alert”>&times;</sapn>**

**XXX**

**</div>**

1. **模态对话框(model) - 小重点**

**模态框定义：在父窗体中弹出一个子窗体，子窗体若不关闭，父窗体无法获得焦点，同事父窗体见还可以传递数据。 window.alert() /confirm/prompt()就是典型例子。**

**模态框必须的结构：**

*<!--模态弹出框推荐卸载body的最后-fixed定位-->* <**div id="mymodal1" class="modal"**> *<!--半透明的黑色遮罩层-->* <**div class="modal-dialog"**>*<!--宽，定位-->*  
 *<!--边框，背景色，阴影-->* <**div class="modal-content alert-dismissable fade in"**>  
 <**span class="close" data-dismiss="alert"**>**&times;**</**span**>  
 <**div class="modal-header"**>头部</**div**>  
 <**div class="modal-body"**>主体</**div**>  
 <**div class="modal-footer"**>尾部</**div**>  
 </**div**>  
 </**div**>  
 </**div**>

**显示一个模态框：**

1. **<a href=”#模态框ID” data-toggle=”modal”>**
2. **<button data-toggle=”modal” data-toggle=”#模态框ID”>**

**若是有超链接的就用：data-toggle=”#modal数字”**

**若是按钮： data-target=”#modal数字”**

**转义字符：&times; 错号**

**2.6插件 - 折叠效果（collapse）**

**触发一个折叠效果：**

1. **<a href=”#折叠组合元素ID” data-toggle=”collapse”>**
2. **<button data-toggle=”collapse” data-target=”#折叠元素ID”>**

**折叠组件的两个扩展效果：**

1. **Accordion手风琴效果**
2. **响应式导航条在手机屏幕中的效果**

**2.7插件-轮播广告（carousel） 小重点**

**.carousel**

**data-ride=”carousel” 定时器**

**本身结构比较复杂，编写时只需要记住根class: .carousel**

**<div class=”carousel”>**

**<div class=”carousel-inner”>**

**<div class=”item active”>**

**<img>**

**</div>**

**</div>**

**</div>**

**2.8插件-附加导航（Affix）**

**2.8.1概念：**

**2.8.2 结构**

**<div data-spy=”affix” data-offset-top=”数值”>**

**<ul class=”nav nav-pills”>**

**<li></li>**

**</ul>**

**</div>**

**2.9插件-滚动监听 了解**

**2.9.1概念**

**随着页面的滚动，某个导航中的项目，会自动的更改.active的位置。**

**2.9.2 实现思路**

**（1）页面中必须首先有一个导航菜单（.nav）--其中可以定义一个菜单项为.active**

**（2）导航菜单中的超链接的href属性值必须和页面中的摸个锚点名字一样。**

**（3）为页面添加滚动时间的监听函数**

**body.onscroll=function(){**

**If(body.滚动的距离===某个锚点距离){**

**此锚点对应的超链接的父元素li添加.active;**

**}**

**}**