第二阶段

day7.26

一.课程安排（20天）

前言，第一阶段的学习目的

|  |
| --- |
| 准备数据，做好接口  等待ajax来访问  HTML5  css美化HTML5  bootstrap 简化 CSS |

1.HTML5 （2天）

2.ajax异步的数据交互（3天）

3.ajax项目 2天

4.CSS 4+2天

5.bootstrap框架（4天）

6.项目2 （3天） 使用boot完成学子商城的3个页面

7.阶段的特点和问题

单词量大，逻辑少。/笔记只记关键部分， /笔记格式：Markdown

二.web基础知识

早期HTML，专指网页开发，很低级

前端的概念，HTML5 大前端技术，全栈工程师

1.web和internet

www:万维网

BBS：电子公告版，论坛

email：电子邮件服务

telnet 远程登录服务

FTP 上传下载服务器

web，就是运行在Internet上的一种服务

2.Internet上的应用程序

（1）c/s结构

|  |
| --- |
| c:client 客户端  s:server 服务器  LOL，王者荣耀 |

（2）b/s结构

|  |
| --- |
| b:browser 浏览器  s:server 服务器  ‘一刀999级’ 游戏网站 |

（3）c/s和b/s的区别

|  |
| --- |
| c/s 需要升级  b/s 不需要升级（升级过程不影响客户） |

三.web运行原理

|  |
| --- |
| web:运行在Internet的上的一种b/s结构的应用程序，俗称网站  Internet：为web应用提供运行的网络环境  Web的工作原理：基于浏览器和服务器以及通信协议来实现的数据的传输和展示 |

（1）通信协议

|  |
| --- |
| 规范了传输的数据是如何打包和传递的  http:// https:// 是专门访问网站的协议 |

（2）服务器

|  |
| --- |
| 1.功能  存储数据（HTML,CSS，js保存在服务器）  接收用户的请求，并给出响应  提供程序的运行环境  具备一定的安全功能  2.服务器产品  Tomcat(Tom猫), 30年历史  Apache, 美国Apache直升机，形容服务器快取名  IIS, 微软的  3.服务器端的技术  1.Java  2.PHP  3..net  4.python |

（3）浏览器

|  |
| --- |
| 1.功能  代表用户发送请求  把响应解析成图形界面展示给用户（HTML,CSS，js由浏览器解析执行）  2.浏览器的产品 （国产浏览器都是买的国外的浏览器内核）  Chrome  Safari  Firefox  Opera  3.浏览器的技术  HTML5 CSS3 JavaScript |

四.HTML入门

|  |
| --- |
| 1.HTML： hyper Markup language 超文本标记语言  超文本：有特殊能力的文本  标记：超文本的语法，使用关键字（<a>,<center>..），定义一个超文本，具备了功能 |

2.HTML的特点

|  |
| --- |
| 使用尖括号‘标记’来声明超文本 |

知识点：

1.标记又称为标签，元素，节点，对象

2.双标记（封闭类型）

3.缩进 前端开发规定，tab键，一个tab两个空格

4.HTML注释快捷键： Ctrl+？， 注意：（1）注释中不能嵌套其他注释，不然中间会被浏览器解析；（2）注释不能出现在标记中

editplus

|  |
| --- |
| Ctrl+s  Ctrl+c  Ctrl+x  Ctrl+v |

1.key快捷方式

|  |
| --- |
| Ctrl+d 删除当前行  alt+上/下 引动当前行位置  Ctrl+alt+下 复制当前行，问题，与显卡快捷键冲突 |

3.HTML文档的构成

(1)文档类型的声明

|  |
| --- |
| <!doctype html> 告诉浏览器当前网页的版本类型是h5 |

(2)网页结构

<html>:网页整体

<head>：网页的头部，定义全局信息

<body>：网页的主体，展示网页内容

<title>：网页标题

<link />：引入外部CSS文件的标签

<script>:定义或者引入js

<mata>:定义全局信息，mata元数据

<mata charset=”utf-8” />

<mata name=”description” content=”描述内容”>

<mata name=”keyword” content=”关键字内容”>

五.文本标记

1.<h1>~<h5>标题

属性：align:center/left(默认值/缺省值)/right

2.<p>:段落

属性：align:center/left(默认值/缺省值)/right

3.<br>或者<br/> ：换行

浏览器会把html中回车解析成一个空格显示

4.<hr>或者<hr/> :分割线

属性：size/width/align/color

5.<pre>:预格式化

保留输入文本的格式（空格和回车）

6.空格折叠现象

避免方法：&nbsp;

其他符号：

< &lt;

> &gt:

® &reg; 注册商标

© &copy; 版权

¥ &yen; 人民币

× &times;

7.文本样式标签

<del> 删除线

<u> 下划线

<sup> 上标

<sub> 下标

作业：

1.nodejs 6个接口，重新写一遍（登录，注册，list，删除，修改，检索）

2.熟悉HTML标签的编写方式—模拟个人简历

day 7.29

6.url

<url:统一资源定位符，路径>。

1.路径的表现形式：

1）绝对路径：完整的路径，主机名称+文件目录+文件名称

使用网络资源时，要使用绝对路径

优点：不占用本地存储空间

缺点：资源不稳定

使用本地资源，能不能使用绝对路径？

src=”c:\08.png” 项目中不可用

2）相对路径

同级文件直接引用：src=”09.png”

同级文件夹下的文件：src=”image/11.png”

../ 上一级 ../../ 上上级

2.img标签的属性

|  |
| --- |
| src=”” source源 引用资源的路径  alt=”” 图片加载失败时显示的文本  width=””  height=”” 设置图片宽高，单位px  不能同时设置高宽，只设置一个，另一个属性自适应 |

4.链接

1）语法：<a href=”#”></a> 超链接文本

href：链接属性，href的能力比url大

target=””:指定打开链接方式，取值：\_self 当前标签中打开新链接

\_blank 在另一个标签中打开新链接

2）a标签的其他表现形式

|  |
| --- |
| 1.下载  <a href="1.zip">下载</a>  2.发送邮件  <a href="mailto:18156797@qq.com">发邮件</a>  3.调用js脚本  <a href="javascript:alert(1111)">调用js代码</a>  4.返回页面顶部  <a href=””>返回顶部</a> |

3）锚点

|  |
| --- |
| 1.使用a标点定义锚点 ---html4中  <a href="#锚点名称"></a>  2.使用元素的id定义锚点 -- html5中  <any id="锚点名称"></any> any:代表任意标签  3.链接到锚点  <a href="锚点名称"></a>  4.跳转到同级另一个html的锚点  <a href="xx.html#锚点名称"></a> |

7.表格

表格布局的缺陷：浏览器在渲染表格的时候，是先把表格所有的数据都读取到内存中，然后再一次性渲染，这种渲染，效率低。现在项目，表格回归本身的作用，只做数据的显示。学习中，可以学习表格布局。

1.表格的语法

|  |
| --- |
| <table>  <tr> 行 table row  <td></td> 列 table data  </tr>  </table> |

2.表格的属性

|  |
| --- |
| Table  border="2px blod red" 设置边框线属性  width="400px"  height="400px"  bgcolor="pink"  cellspacing="10px" 设置单元格外边距（边框与边框之间的距离）  cellpadding="10px" 设置单元格内边距（内容和边框之间的距离）  align="center" 设置表格本身水平对齐方式  tr  align=”center” 设置文字水平居中  bgcolor=”red” 设置背景颜色  valign=” top/middle/bottom/baseline” 设置表格行内内容垂直对齐方式  td  属性：设置高宽，水平垂直对齐方式，背景颜色，colspan:跨行合并 rowspan:跨列合并 |

练习

|  |
| --- |
| 04\_exam.html  创建一个表格4\*4,400px x 400px,背景色yellow 边框1px 水平居中，内边距5px,外边距10px,表格边框颜色自定 |

3.不规则的表格

|  |
| --- |
| Colspan=”n”:跨行合并 rowspan=”n”:跨列合并  合并要删除多余的列 |

4.可选标记

|  |
| --- |
| <caption>标题</caption> 表格标题  如果设置标题，必须将caption标签紧紧挨着<table>之后  <th>:表头，默认居中，加粗 |

5.复杂的表格应用

1）行分组

|  |
| --- |
| 可以将连续几行，划分到一个组中，进行统一的管理  <thead></thead> 表头  <tbody></tbody> 表身  <tfoot></tfoot> 表脚 |

2）表格的嵌套

|  |
| --- |
| 被嵌套的表格，必须只能放在td中 |

8.列表（重点）

1.列表的作用

|  |
| --- |
| 展示数据，让数据更有条理  经常用无需列表做布局 |

2.列表的组成

|  |
| --- |
| 外层列表类型  1.有序列表 <ol></ol> order list  2.无序列表 <ul></ul> unorder list  列表项  <li></li> list item |

3.属性

1）有序列表的属性

|  |
| --- |
| type:指定标识项类型  取值： 1：默认值  a/A:字母  i/I 罗马数字  start:指定起始编号  取值：无单位数字 |

day7.30

2）无序列表

|  |
| --- |
| 属性  type 设置标识项类型  取值：  none 不显示  disc 实心圆 默认值  circle 空心圆  square 矩形  注意：ul在项目中使用非常多，常用于一组相似的内容进行布局和排版 |

4.列表嵌套

|  |
| --- |
| 所有的嵌套的内容，必须写在li中 |

5.定义标签 ----h5新标签

|  |
| --- |
| 对一个名词进行解释说明，使用定义列表  <dl> 定义列表的起始  <dt></dt> 要解释说明的名词  <dd></dd> 对这个名词的解释内容  …  </dl> |

9.结构标记

|  |
| --- |
| 用于描述整个网页结构的标记，取代div作用布局用的  使用结构标记，增加了代码的可读性，提升百度竞排中排名  1.<header></header>  定义网页的头部，或者某个区域的顶部  2. <footer></footer>  定义网页的脚部，或者某个区域的底部  3.<nav></nav>  定义导航链接  4.<aside></aside>  定义网页侧边栏  5.<section></section>  定义网页的主体内容  6.<article></article>  定义与文字相关的内容（回帖，论坛，评论） |

10.表单（重点&难点…）

1.作用

|  |
| --- |
| 1.提供可视化控件  2.收集用户输入的信息，并提交请求给服务器  总结：form自带收集整理数据功能，自带提交请求功能  使用ajax是不需要form支持  ajax中，input不需要name属性 |

2. 表单的组成

|  |
| --- |
| 前端部分  提供表单控件，与用户交互的可视化控件  后端部分  后台接口对提交的数据进行处理 |

3.表单

|  |
| --- |
| <form></form> |

1）属性

|  |
| --- |
| 1.action=”” 定义表单提交时发送的动作（定义发送的接口）  如果没有值，默认提交给本页  2.method="" 定义表单的提交方法  取值 get 默认值 提交的内容会显示在地址栏上，明文提交，提交的数据有大小限制，最大2KB 向服务器要数据的时候，使用get  post 隐式提交，提交的内容不在地址栏显示，提交的数据没有大小限制，要传递数据给服务器的时候使用post  其他方法 delete put option  3.enctype=”" 指定表单数据的编码方式：允许将什么样的数据格式提交给服务器  取值：1.application/x-www-form-urlencoded 缺省值 允许将任意的字符提交给服务器（不能提交文件）  2.text/plain 允许提交普通字符给服务器  3.multipart/form-data 允许提交文件给服务器 |

4.表单控件：在form标签中，能够与用户进行交汇的可视化元素

1）分类

|  |
| --- |
| 1.input 元素 基础9种，h5新出的10种  2.textarea 多行文本域  3.select和option 下拉选择框，下拉选项  4，其他元素 |

2）input元素

|  |
| --- |
| 文本框，密码框，单选按钮，多选按钮，提交按钮  <input type="控件类型" name=”控件的名称”>  name 为控件定义名称，提供给服务器使用（name 必须写，不然提交不了，name就是提交给服务器的变量的变量名）  value 控件的值，传递给服务器的值  特殊情况：对于所有的按钮来说，value是设置按钮上的文字  disabled 禁用，不能修改值，并且不能提交，无值属性，在标签内使用时，可以不写值 |

3）input详解 基础 9个

|  |
| --- |
| 1.文本框和密码框  <input type=”text” >  <input type="password">  属性：  maxlength="3" 指定输入的最大长度  readonly :无值属性，只读，只能看不能改，可以提交  placeholder 提示占位符，默认显示在控件上的文本，输入信息时文本消失 |

|  |
| --- |
| 2.按钮  提交按钮 type="submit" 将当前表单中的数据，收集整理，发送请求给服务器  重置按钮 type="reset" 将当前表单内容恢复到初始化状态  普通按钮 type="button" 配合事件，调用js代码  H5为了提升代码的可读性，提出了新的按钮元素  <button>H5提供的按钮</button> 此标签功能同submit一样 |

|  |
| --- |
| 3.单选框/复选框  单选按钮 type=”redio”  复选框 type=”checkbox”  属性name，除了定义控件名称之外，还起到分组的作用，必须要写value值，不然传递的数据永远是on  checked 默认选中项 |

|  |
| --- |
| 4.隐藏域  传递不需要给用户看，但是又要传递给服务器的数据  type=”hidden” |

|  |
| --- |
| 5.文件上传  type=”file”  前提：把enctype=”multipart/form-data” 可以提交文件  method=”post” get有大小限制  属性：multiple 无值属性 可以选择多个文件（按住Ctrl） |

|  |
| --- |
| 6.文本域  <textarea name="addr" id="" cols="5" rows="3"></textarea>  根据计算机硬件不同，显示效果不同，不准确 |

|  |
| --- |
| 7.下拉选择框  <select name="city" id="">  <option value="">北京</option>  <option value="">上海</option>  <option value="">广州</option>  </select>  注意：1.如果option没有value属性，那么select的value是选中option的内容  2.如果option有value属性，那么select的value是选中option的value值  属性：size 1 默认值 >1 滚动选择框  multiple 多选 selected 默认选中 |

|  |
| --- |
| 8.其他元素  1.label元素 关联文本与表单控件  需要绑定的空间，加id  把id值写入label的for属性，就可以实现绑定  <input type="radio" name="jd" id="jd">  <label for="jd">京东钱包</label>  2.为控件分组  <fieldset>  <legend>组名</legend>  input...  </fieldset>  3.浮动框架 iframe  在一个页面中，引入其他页面  <iframe src="input\_demo.html" frameborder="0" scrolling="no" width="100%" height="600px"></iframe>  属性  src="input\_demo.html" 要引入的网页资源  frameborder="0" 浮动框架的边框  scrolling="no" 浮动框架的滚动条yes/no/auto  width="100%" 浮动框架的宽度  height="600px" 浮动框架的高度 |

作业：

1.nodejs接口6个，再写一遍

day7.31

4.控件

6.新表单元素 10个

|  |
| --- |
| 1.邮箱  <input type="email" name="um"> 输入的内容有@，@后面有内容  2.搜索类型文本框  <input type="search"> 自带快速清除功能  3.url类型  <input type="url"> 输入必须是网址:http://1...  4.电话号码  <input type="tel" name="tel">  移动设备使用，在获取焦点时，打开虚拟键盘  5.数字类型  <input type="number" name="number" max=””>  max 最大值  min 最小值  step 步长  6.范围类型  <input type="range" name="range" max="50" min="20" step="5">  max 最大值  min 最小值  step 步长  7.颜色类型  <input type="color">  8.日期类型  <input type="date">  9.月份类型  <input type="month">  10.周类型  <input type="week"> |

AJAX

一HTTP协议

1.URL

|  |
| --- |
| http://cdn.tmooc.cn/bsfile//courseImg///23B47E1BC4F9488089AA89F29461A06B.jpg  url的结构：协议+主机名称+目录结构+文件名称  URL的完整结构：  <scheme>://<user>:<pwd>@<host>:< port>/<path>;  <params>?<query>#<frag> |

1.<scheme>：//

|  |
| --- |
| scheme：方案，协议。定义以那种方式获取服务器资源  C:\Users\web\Desktop\lijun\第二阶段\web第二阶段练习效果图示\协议及端口号.png  DNS HTTP HTTPS |

2. <user>:<pwd>

|  |
| --- |
| 后台管理员，偶尔使用，绝大所述程序员不在使用这种方式登录 |

3. host

|  |
| --- |
| 主机名称 localhost/127.0.0.1/www.baidu.com  主机名称：IP/地址/域名  域名通过DNS服务器做域名解析，把IP地址传递给我们，之后，我们的浏览器再去访问这个ip地址 |

域名解析：在命令行输入ping www.baidu.com 回车

4 port

|  |
| --- |
| 端口：就是在主机上，每一个服务对应的柜台号 |

端口号2的16次方-1= 65535，端口1024之前的是系统所占用的端口，用户不能占用。

5. path

|  |
| --- |
| 路径，资源在主机上具体存放的位置  http://127.0.0.1:8080/cl/nodejs |

6. params

|  |
| --- |
| 参数，跟踪状态参数 |

7. ?query

|  |
| --- |
| <http://127.0.0.1:8080/cl/nodejs>？uname=dingding&upwd=123456  Get 方法提交请求是，地址栏出现的查询字符串 |

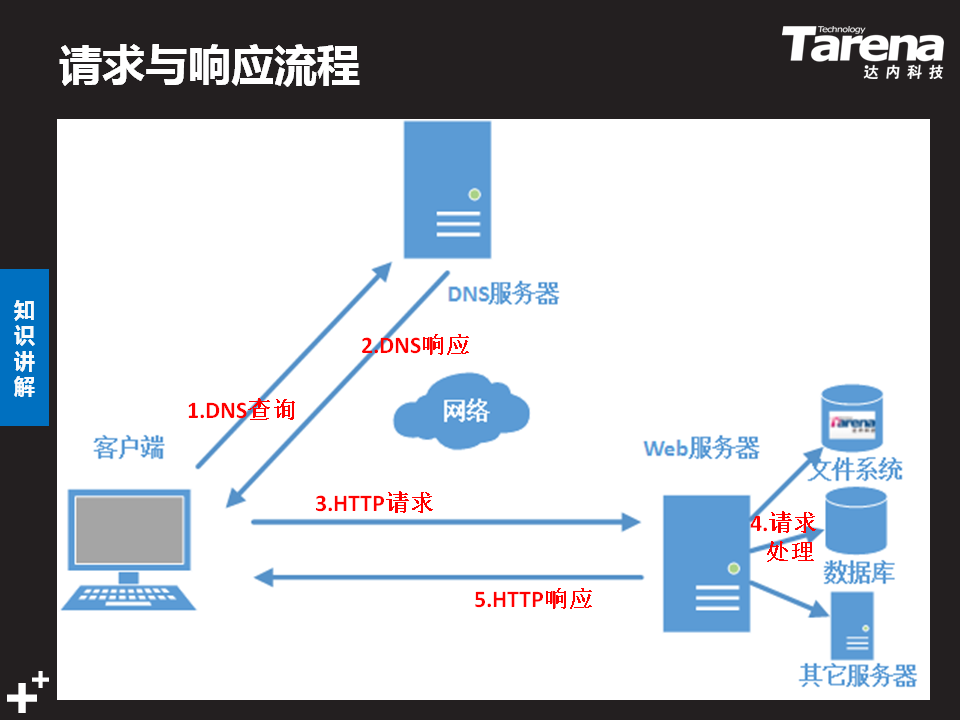
8.# frag

|  |
| --- |
| 锚点 |

2.http协议

|  |
| --- |
| http协议：hyperText Transfer Protocol超文本传输协议  专门用于传输HTML文件，规范了数据是如何打包和传递的  C:\Users\web\Desktop\lijun\第二阶段\web第二阶段练习效果图示\HTTP协议历史.png |

3.web请求响应的原理



4.消息/报文 message

|  |
| --- |
| 请求消息 request message  响应消息 response message  C:\Users\web\Desktop\lijun\第二阶段\web第二阶段练习效果图示\message.png    1. 请求消息 request message  请求起始行 请求头 请求主体  2. 响应消息 response message  响应起始头 响应头 响应主体 |

5.请求消息request message

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 客户端发送给服务器的数据库，请求消息  由三部分组成：请求起始行，请求头，请求主体  1. 请求起始行  1）请求de方法:   |  |  | | --- | --- | | http原生的请求的方法 | Restful API 规则 | | get:客户端向服务器要数据明文，有大小限制2KB  使用查询字符串传递明文  req.query | get：查询 select\*from xz\_user where uid=2  无请求主体  req.params | | post:客户端要传递给服务器数据的时候，使用隐式。没有大小限制  使用请求主体传递参数  req.body  form data就是请求主体 | post:新建 insert  有请求主体。  req.body | | put:客户端网服务器放文件的时候使用，一般被禁用 | put：修改update  有请求主体  req.body | | delete:客户端从服务器删除文件的时候使用，一般被禁用 | delete：删除 delete  无请求主体  req. params |   2）请求的版本协议号：http/1.1  3）请求的url: host: www.tmooc.cn  2. 请求头   |  | | --- | | host: [www.tmooc.cn](http://www.tmooc.cn) 客户端告诉服务器，请求的是哪一个主机  Connection: keep-alive 客户端告诉服务器开启持久连接  User-Agent: 客户端告诉服务器，我现在的浏览器的类型和版本号  Accept-Encoding:客户端告诉服务器，我能够接受的压缩文件的格式  Accept-Language: zh-CN 客户端告诉服务器，我能够接受的自然语言的类型  Referer: <http://www.tmooc.cn/> 客户端告诉服务器，本次请求来自哪个网页 |   3. 请求主体   |  | | --- | | form data  post,put 有请求主体  get,delete 无请求主体 | |

6.响应消息response message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务器发送给客户端的数据块  由三部分组成：响应起始行，响应头，响应主体  1.响应起始行   |  |  | | --- | --- | | 协议版本号 http:/1.1  响应状态码   |  | | --- | | 1xx:正在请求，提示的信息  2xx:响应成功  3xx:重定向  301 永久重定向  302临时重定向  304 命中缓存  4XX:404 请求资源不存在  403 权限不够  405 请求方法不被允许  5XX: 500 服务器代码出错 |   原因短句，就是响应状态码简短的解释说明 |   2.响应头   |  | | --- | | Date: Wed, 31 Jul 2019 07:20:30 GMT服务器告诉浏览器，服务器响应时间  Connection: keep-alive 服务器告诉浏览器，已经开启持久连接  Content-Type: text/html 服务器告诉浏览器，响应主体的内容是什么类型  text/plain 响应回来的数据是普通文本  application/javascript 响应回来的是js文本  text/css 响应回来的是一个css文本  其他：image/jpeg（图片） application/xml(xml的字符串) application/json（json字符串） |   3.响应主体   |  | | --- | | 服务器传递给浏览器的数据 | |

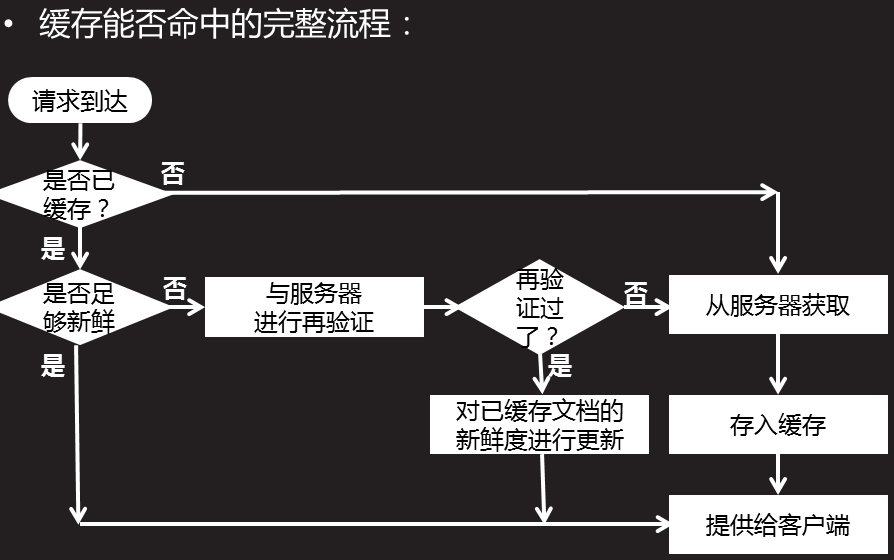
7.缓存

|  |
| --- |
| 客户端将服务器响应回来的数据进行自动的保存  当再次访问的时候，直接使用保存的数据 |

缓存的优点

|  |
| --- |
| 1.减少坑余的数据传输，节省客户端流量  2.节省服务器带宽  3.降低了对服务器资源的消耗，运行的要求  4.降低了由于远距离传输造成的加载延迟 |

缓存的新鲜度和过期



|  |
| --- |
| 1.请求—无缓存—连接服务器—存缓存—客户端运行  2.请求—有缓存—够新鲜—使用缓存---客户端运行  3.请求—有缓存—不新鲜—连接服务器确认是否过期—没过期—更新缓存的的新鲜度—客户端运行  4.请求—有缓存---不新鲜—连接服务器确认是否过期---已过期—连接服务器获取数据—存缓存—客户端运行 |

1.与缓存相关的消息头

|  |
| --- |
| Cache-Control: max-age=0  从服务器将文档传到客户端的那一刻起  此文档处于新鲜的秒数 |

2.设置缓存需要修改的消息头

|  |
| --- |
| <meta http-equiv="cache-control" content="max-age=3600"> |

8.HTTP性能优化

1.http的连接过程

|  |
| --- |
| 发起请求—>建立连接—>服务器处理请求—>访问资源--- >构建响应---->发送响应—>记录日志 |

2.HTTP的性能优化

|  |
| --- |
| 1.减少连接的创建次数（开启持续连接）  2.减少请求的次数（代码的优化，每一次引用外部文件都是一次请求）  3.提高服务器运行速度  4.尽可能的减少响应数据的长度（代码压缩） |

9.HTTPS：安全的http协议

|  |
| --- |
| 为数据通信提供安全支持  1.客户端发送请求---->SSL加密--->服务器接收到加密文件--->SSL层解密，得到请求明文  2.服务器发送响应---->SSL加密--->客户端接收到加密文件--->SSL层解密，得到请求明文 |

二 DOM操作（简单的DOM操作）

|  |
| --- |
| ajax提交请求，不需要form表单  但是，form表单自带收集整理数据的功能  ajax不使用form表单，就没有自动收集整理数据的功能  我们需要使用js的DOM操作，手写代码，收集数据 |

1.完整的Javascript的构成

|  |
| --- |
| 1.js核心 ECMA Script ES6  2.DOM：document object model 文档对象模型  让js动态的操作页码上的元素  3.BOM：browser object model 浏览器对象模型  让js动态的操作浏览器 |

2.使用js获取页码上的元素

|  |
| --- |
| 1.获取元素对象  var $uname=document.getElementById("d1");  2.获取修改值  var u=$uname.value; 获取value  $uname.value=”…”; 修改value  3.获取双标签的内容区域  var u=对象.innerHTML; 获取内容区域  对象.innerHTML=”…”; 修改内容区域  ES6可以通过id获取元素对象，做了简化 |

写DOM的步骤总结：

|  |
| --- |
| 1.写事件 鼠标事件/键盘事件  2.写事件中的方法  3.方法中获取元素对象  4.修改对象属性 |

day8.1

二 DOM操作（简单的DOM操作）

|  |
| --- |
| ajax提交请求，不需要form表单  但是，form表单自带收集整理数据的功能  ajax不使用form表单，就没有自动收集整理数据的功能  我们需要使用js的DOM操作，手写代码，收集数据 |

1.完整的Javascript的构成

|  |
| --- |
| 1.js核心 ECMA Script ES6  2.DOM：document object model 文档对象模型  让js动态的操作页码上的元素  3.BOM：browser object model 浏览器对象模型  让js动态的操作浏览器 |

2.使用js获取页码上的元素

|  |
| --- |
| 1.获取元素对象  var $uname=document.getElementById("d1");  ES6可以通过id获取元素对象，做了简化：在ES6中规定了可以用id值直接代替对象  $开头命名的意义：$uname:定义的变量，uname:对象  2.获取修改值  var u=$uname.value; 获取value  $uname.value=”…”; 修改value(value处理单标签的值)  3.获取双标签的内容区域  var u=对象.innerHTML; 获取内容区域  对象.innerHTML=”…”; 修改内容区域  对象.innerHTML=”<a></a>”来动态的修改页面布局 |

5.事件

|  |
| --- |
| 通过用户的行为来激发的操作  onclick=”fn()”, 鼠标点击事件  onblur=”fn()”, 鼠标失焦事件  onfocus=”fn()”, 鼠标聚焦事件 |

三Ajax

1.同步Synchronous

|  |
| --- |
| 在一个任务执行的过程中，不能执行其他任务  一次只能干一件事  同步访问：浏览器在向服务器发送请求的过程中，浏览器只能等待服务器的响应，不能做别的事 |

2.异步Asynchronous

|  |
| --- |
| 在一个任务执行的过程中，不影响其他任务  异步访问：浏览器在发送请求的时候，用户可以再页面做其他的操作  **出现场合**：  1.用户名重复的验证：codeboy  2.聊天室  3.股票走势图  4.百度的搜索建议 |

3.什么是ajax

|  |
| --- |
| Asynchronous Javascript and XML 异步的js和XML  本质：使用js提供的异步对象，向服务器发送请求，并接受服务器响应回来的数据  异步对象：XMLHttpRequest |

4.使用ajax

|  |
| --- |
| 1.创建异步对象  var xhr=new XMLHttpRequest();  上面版本不兼容IE8以下的版本，如果想支持写以下代码(\*不是重点,不需要记忆，了解即可\*)：  var xhr=””; //如果你的浏览器有XMLHttpRequest属性就说明是IE8以上的版本  if(window. XMLHttpRequest){  var xhr=new XMLHttpRequest();  }else{xhr=new ActiveXObjcet(“Microsoft.XMLHttp”);  }  2.打开连接，创建请求  xhr.open("method","url",isAsyn);  method:请求方法，必须是string类型  url：请求的接口，必须是string类型  isAsyn:是否使用异步访问 true/flase  3.发送请求  xhr.send(请求主体);  如果使用的请求方法，没有请求主体。那么参数就可以不写，或者写null  xhr.send(null); / xhr.send();  4.接收响应（绑定监听，接收响应） 这步要放在第1步创建异步对象之后  xhr.onreadystatechange=function(){  if(xhr.readyState==4&&xhr.status==200){  var result=xhr.responseText; //在status: 0->1和1->2时//responseText并未有值  }  }  (1)xhr.onreadystatechange 监听值是否发生改变  readyState：监听值  0:请求尚未初始化  1：已经打开服务器连接，正在发送请求  2：开始接收响应头消息  3：开始接收响应主体  4：响应数据接收完成  只要请求—创建，readyState的值就自动开始改变  2.xhr.status属性，服务器响应状态码  当值为200的时候，表示服务器响应传输成功  3.xhr. onreadystatechange=function(){}  绑定监听，监听xhr.readyState的值的改变  Function会执行4次  xhr. readyState==4使我们需要的  同时，还要求xhr.status==200  4.xhr.responseText属性  响应数据存在此属性中 |

常见错误

|  |
| --- |
| 1.跨域错误  捕获  2.方法未定义  捕获1  时间中条用的方法名和你定义的方法名是否一致  3.出乎意料的结尾  捕获3  少括号  4.数据库没有开  捕获4  有时候也报错：result.length未定义 |

5.使用http的get方法带参数发送请求

|  |
| --- |
|  |

day8.2

AJAX

(5)使用http的get方法带参数发送请求

带参数的resful的get方法(客户端验证非空)

|  |
| --- |
| 带参数的http原生get方法  xhr.open(“get”,”url”,true)  url中拼接查询字符串  ex:”/ajax/reg?uname=”+$uname+”&upwd=”+$upwd |

(6)带参数restful的get方法（客户端非空）

|  |
| --- |
| resful要求，接口必须如下设计  /版本号/动作 /v1/login  ex:”/v1/login/:uname&:upwd” 看到冒号，肯定是restful接口  带冒号的参数怎么接收 req.params接收 |

(7)post方法

|  |
| --- |
| get不带参数的请求，http和restful的操作一样  get代参数的请求，http使用？传参，使用req.query接收  restful使用：传参，使用req.params接收  post请求，http和restful的操作一样  注意：两点  1.post/put 的请求中数据，要放在请求主体中，然后send（请求主体）  Var formdata=”…”;  Xhr.send(formdata);  2.需要设置请求消息头，让请求可以传递特殊符号的字符  xhr.setRequestHeader("content-Type","application/x-www-form-urlencoded");  请求消息头的设置的位置，在xhr.open和xhr.send()之间 |

练习：

|  |
| --- |
| 08\_post\_login.html  /v3/login 使用post方法完成登录模块  作业：  提高题：09\_post\_regsiter.html post接口完成注册模块 |

四.json数据格式

1.什么是json

|  |
| --- |
| Javascript object notation js对象表现法  以类似js对象的格式表现出来的字符串  Json字符串表示法  1.json中用一对{}来表示应对向  2.json的属性，使用””括起来（写单引号不报错，但是不推荐）  3.如果属性的值是字符串，也用引号括起来  4.json整体是一个字符串，所以做外层添加”” |

2.为什么要使用json

|  |
| --- |
| 当客户端请求服务器，服务器根据请求查询数据库，得到结果  此时，结果类型obj，通过响应发送回客户端  客户端接收到的结果是str（json串）  我们需要把这个str转换成js对象，再继续操作  arr=JSON.parse(result); 解析json字符串为js对象 |

五.xml数据格式

|  |
| --- |
| XML:eXtensible Markup language 可拓展标记语言  Xml是HTML的一个变种，专门用来承载数据  我们需要使用DOM操作，来获取xml中的信息 |

1.xml的语法

|  |
| --- |
| 1.第一行做版本声明  <?xml version="1.0" encoding="utf-8"?>  2.其他语法  (1)一个XML文件，有且只有一个根元素  (2)xml中只有双标签，标签名称自己定义  (3)xml标签严格区分大小写 |

项目

restful api 接口风格

|  |
| --- |
| 登录接口 get req.params  /api/v1/login  查询所有用户 get req.params  /api/v1/userlist  根据uid删除用户 delete  /api/v1/deluser  根据uid查询某一个用户 get  /api/v1/searchuser  注册 post  /api/v1/register  修改用户信息 put  /api/v1/updateuser 灰色有请求主体 |

day 8.3

1.restful的get要在前台做非空验证，有请求主体的可以在后台做非空验证。

2.立即执行事件 onload

|  |
| --- |
| 页面加载完毕，马上执行的事件  Onload,只能写在body里面 <body onload=”onc()”> |

day08

**CSS 基础**

一.CSS3概述

1.什么是css

|  |
| --- |
| css:Cascading Style Sheets  层叠样式表，级联样式表，简称样式表 |

2.作用

|  |
| --- |
| 修饰html元素，让元素变得好看 |

3.HTML和CSS的关系

|  |
| --- |
| html：负责搭建网页结构，显示网页内容  css:负责网页的修饰 |

4.css与html属性之间的使用原则

|  |
| --- |
| w3c建议，尽量使用css的方式，来取代html属性  css样式比属性更好维护，提高了代码的可维护性和可重用性 |

二.CSS的语法规范

1.css的使用方式

①内联样式

|  |
| --- |
| 也叫行内样式  将css样式定义在元素的style属性中  <ANY style="样式声明"> ANY：代表任意元素  样式声明： 样式属性:样式值  样式属性和值之间，用冒号  多个样式声明之间，用；隔开 <div style="color:red;font-size:24px">内联样式</div>  内联样式在项目中基本不用  原因：1.没有代码的重用性和可维护性  2.内联样式优先级最高  只有在学习和测试的时候使用 |

②内部样式

|  |
| --- |
| 在网页的头部<head>标签中添加<style></style>  在style标签中，定义该网页的所有样式  语法 <head>  <style>  选择器{样式声明;样式声明；....}  </style>  </head>  选择器:规范了，页面中哪些元素可以使用定义好的样式  div{}  span{}  a{}  内部样式，在项目中使用不多  可重用性只对当前页面有效，局限性很大  学习和测试使用较多 |

③外部样式

|  |
| --- |
| 单独创建一个.css文件，在此文件写样式  在.html文件的head标签中，使用link引入  <link rel="stylesheet" href="my.css">  href属性，引入的css资源路径  rel属性，引入的文件是什么，必须写  外部样式，是项目中使用最频繁的 |

④3个常用样式

|  |
| --- |
| 1.color 设置文本颜色  取值，合法的颜色值  2.font-size 设置字号大小  取值，px为单位的数字  3.background-color 设置背景颜色  取值,合法的颜色值 |

练习

|  |
| --- |
| 01\_ex.html  h1 lorem 内联，背景为pink，字体为yellow  h2 lorem 内部 背景yellow 字体为red  h3 lorem 外部 背景为red 字体blue |

2.css样式的特性

①继承性

|  |
| --- |
| 大部分的css效果是可以直接被继承的  父元素的效果，作用在后代元素上 |

②层叠性

|  |
| --- |
| 可以为一个元素定义多个样式规则  每个样式规则中，样式属性不冲突的时候，可以同时作用在当前元素上 |

③优先级

|  |
| --- |
| 如果样式声明发生冲突，按照样式规则默认优先级去应用样式  默认优先级，由高到低：  1.最高，内联样式  2.内部样式和外部样式，就近原则(谁里目标标签近，应用谁)  3.最低，浏览器默认样式  ps:f12中的样式顺序，是按照默认优先级显示的 |

④调整优先级

|  |
| --- |
| !important 放在属性值之后，与值之间用空格分开  作用，调整样式的优先级  注意，!important的不能写在内联样式中  ex: h4{background-color:pink !important;} |

练习

|  |
| --- |
| 02\_ex.html  p lorem  内部样式设置文字颜色blue,字号24px  外部样式设置文字颜色red,字号40px  将外部样式引入，f12查看页面效果  改变外部样式引入位置，f12查看效果  尝试使用!important调整样式优先级 |

三.基础选择器(重点)

1.选择器的作用

|  |
| --- |
| 规范了页面中哪些元素能够使用定义好的样式  选择器就是为了匹配元素(选择器就是一个条件，符合这个条件的元素就能使用定义的样式)  学习选择器，就是学习写条件的格式 |

2.选择器详解

①通用选择器

|  |
| --- |
| \*{样式声明} 项目中基本不用  唯一一种可能使用\*的方式  \*{margin:0;padding:0} 所有元素内外边距清0 |

②元素选择器，标签选择器

|  |
| --- |
| 页面中，所有对应元素/标签，都使用这个样式  设置页面中某种元素的公共样式 |

③id选择器，专属定制

|  |
| --- |
| 只对当前页中，一个标签生效  <any id="id值"></any>  #id值{样式的声明}  项目中很少单独使用id选择器  通常会作为子代选择器和后代选择器的一部分  #content ul li a{} |

练习

|  |
| --- |
| 03\_ex.html  h2标签，lorem. id为text1  使用id选择器为text1设置背景为黄色，字体为purple  字体为斜体 font-style:italic;  再使用元素选择器，设置h2 背景为红色，字体为white  坑 |

④类选择器

|  |
| --- |
| 作用：定义页面上某个或者某类样式(公共样式，谁想用都可以用)  通过元素的class属性来引用  .d1{font-size:50px}  <div class="d1"></div><span class="d1"></span>  语法：  1.声明类选择器样式  .类名{样式声明}  注意:类名 1.点不能省略  2.类名不能以数字开头  3.类名只能包含 - \_  2.引用  <any class="类名"></any> |
| 多类选择器---一个元素引用多个类  <any class="类1 类2 类3 ...."></any> |
| 分类选择器  1.将元素选择器和类选择器一起使用  更精准的确定元素的样式  语法：元素选择器.类选择器{样式声明}  引用此类名的这个元素，才能使用这个样式  p.text-color{}  2.将类选择器和另外要给类选择器一起使用  语法 .类选择器.类选择器{样式声明}  同时引用了这个两个类选择器的元素，才能使用此样式  .dropdown-menu.show{} |

练习

|  |
| --- |
| 04\_ex  div和p标签，内容假文  用类选择器为所有设置字体颜色为红色  再创建一个类选择器设置背景颜色为黄色  使用分类选择器设置p元素的字号位40px，同是背景色位黑色 |

⑤群组选择器

|  |
| --- |
| 将多个选择器放在一起定义公共样式  语法：选择器1,选择器2,选择器3,.......{样式声明} |

⑥后代选择器

|  |
| --- |
| 后代：一级或多级的嵌套关系  通过元素的后代关系匹配元素  语法：选择器1 选择器2 选择器3 .....{样式声明}  div p span{color:red;} |

⑦子代选择器

|  |
| --- |
| 通过元素的子代关系匹配元素  子代：一级嵌套关系  语法：选择器1>选择器2>...{样式声明} |

⑧伪类选择器

|  |
| --- |
| 匹配元素不同状态的选择器  语法 选择器:伪类选择器{样式声明}  love hate  :link 匹配未访问过的状态  :visited 匹配已访问过的状态  :hover 匹配鼠标悬停在元素上的状态  :active 匹配元素被激活时的状态(鼠标点住元素不放开) |

练习：

|  |
| --- |
| 06\_ex.html  一个a标签，内容随意，链接到http://www.sina.com  1.访问过后，文本orange  2.被激活时，文本颜色green  3.鼠标悬停时，文本颜色red  4.未被访问 文本颜色pink 没有下划线text-decoration:none;  坑：注意，4个伪类同时作用到1个元素上时，会发生冲突  解决方案，4个伪类需要按照一定顺序编写，就不会冲突  顺序：爱恨原则 love&hate  :link :visited :hover :active |
| :focus 匹配元素获取焦点的状态 |

⑨选择器的权值

|  |
| --- |
| 权值：表示当前选择器的重要程度，权值越大优先级越高  !important >1000  内联样式 =1000  id选择器 =100  类选择器 =10  伪类选择器 =10  元素选择器 =1  通用选择器\* =0  继承的样式 无  权值的特点  1.当一个选择器中含有多个选择器时，需要把所有的选择器的权值进行相加，然后比较，权值大的优先显示  2.权值相同，使用就近原则  3.群组选择器的权值单独计算，不能相加  4.样式后面添加!important，直接获取最高优先级  但是，内联样式不能添加!important  5.选择器的权值计算不会超过自己的最大数量级  （100个1加在一起，也不会大于10  100个村民也打不过一个村长） |

四.尺寸和边框

1.尺寸属性

|  |
| --- |
| 设置元素的高度和宽度 |

附加知识点，尺寸单位

|  |
| --- |
| 1.px像素  2.in 英寸 1in=2.54cm  3.pt 磅值，多用于设置字号大小 1pt=1/72in  4.cm 厘米  5.mm 毫米  6.em 相对父元素的单位乘以的倍数  7.rem  8.% |

作业：

1.Ajax 登录模块和注册模块

day09

ajax注册模块

|  |
| --- |
| 根据用户名称，查询用户 get  /api/v1/checkuname/:uname |
| 注册 post 有请求主体  /api/v1/register |

四.尺寸和边框

1.尺寸

②页面中允许设置尺寸的元素

|  |
| --- |
| 行内元素，设置宽高无效，尺寸是靠内容撑起来  自带宽高属性的元素，可以设置css尺寸，img table  块级元素，可以设置宽高  块级元素不设置宽，默认宽，宽度是父元素宽度的100%  不设置高，默认高，高度靠内容撑开，没内容高就是0  行内块，可以设置宽高  行内块不设置宽高，自带一个宽高尺寸，不同浏览器给的默认宽高不同 |

附加知识点

|  |
| --- |
| 由于不同浏览器，对元素默认提供的样式，有差异  在开之前，我们要把这些差异清除掉。  这个行为，叫做css reset，样式重置  css reset最经典的代码 \*{margin:0;padding:0} |

2.溢出处理

|  |
| --- |
| 当内容较大，元素内容区域小的时候，就会发生溢出效果  默认，缺省情况，溢出部分显示，默认垂直溢出  overflow:  取值： visible 默认值，溢出部分显示  hidden 溢出部分隐藏  scroll 不管是否溢出，水平和垂直都添加滚动条  auto 溢出的时候，在溢出方向添加滚动条  不溢出没有滚动条  overflow-x 设置水平滚动条  overflow-y 设置垂直滚动条  如果写水平溢出 |

练习

|  |
| --- |
| div#list>p>img\*6  横向溢出，有滚动条显示6张小兵  11:05~11:20休息 |

附加知识点，合法颜色值

|  |
| --- |
| 1.颜色的英文单词 red yellow.....  2.#rrggbb 红绿蓝 6位16进制数字 #000000 黑 #ffffff 白  3.#aabbcc---->#abc #000 #fff #f00 #0f0 #00f #ccc #eee #666  4.rgb(0~255,0~255,0~255)  5.rgb(r%,g%,b%)  6.rgba(r,g,b,a) alpha:透明度 0~1 0全透明，1不透明 |

3.边框

①简写方式

|  |
| --- |
| border:width style color;  最简方式border:style; 默认1px 黑色，必须写style  width:边框的宽度  style:边框样式  solid 实线  dotted 虚线，圆点  dashed 虚线，短线  double 双实线  color:边框颜色，合法的颜色值和transparent（等同于rgba(,,,0)）  border:0/none；清除边框 |

练习

|  |
| --- |
| 03\_ex  创建一个div#d1,200px\*200px  设置4个方向的边框为1px 实线 orange  border:1px solid orange; |

②单边设置

|  |
| --- |
| border-top/right/bottom/left:width style color; |

③单属性的设置

|  |
| --- |
| 同时设置4个方向的某一个属性  border-style:  border-width  border-color |

④单边单属性的设置

|  |
| --- |
| border-top-width  border-top-style  border-top-color  border-right-width  border-right-style  border-right-color  border-bottom-width  border-bottom-style  border-bottom-color  border-left-width  border-left-style  border-left-color |

⑤边框的倒角

|  |
| --- |
| 将直角倒成圆角  border-radius:  取值 1.px为单位的数字  2.% 50%就是圆 |

⑥圆角的单角定义

|  |
| --- |
| border-top-left-radius  border-top-right-radius  border-bottom-left-radius  border-bottom-right-radius |

⑦边框的阴影

|  |
| --- |
| box-shadow：h-shadow v-shadow blur spread color  最简方式 box-shadow:h-shadow v-shadow;  h-shadow 水平方向阴影的偏移量  v-shadow 垂直方向阴影的偏移量  blur 阴影的模糊距离，值越大，越模糊  spread 阴影的尺寸  color 阴影的颜色  inset 将默认的外部阴影，变为内部阴影 |

练习

|  |
| --- |
| 03\_ex 写一个发光的球体 |

⑧轮廓

|  |
| --- |
| outline:width style color;  outline:0/none;清除轮廓  一般input需要清除轮廓 |

五.框模型---盒子模型

|  |
| --- |
| 盒子模型---元素在页面上实际的占地空间的计算方式  浏览器默认的元素实际占地宽度：  左外边距+左边框+左内边距+内容区域宽度+右内边距+右边框+右外边距  浏览器默认的元素实际占地高度：  上外边距+上边框+上内边距+内容区域高度+下内边距+下边框+下外边距 |

1.外边距margin

|  |
| --- |
| 改变元素外边距，会产生位移效果  外边距margin:边框意外的距离(元素与元素之间的距离)  margin:value; 同时设置4个方向的外边距  margin-top: 上  margin-right 右  margin-bottom 下  margin-left 左  取值：1.以px为单位的数字  2.% 父元素宽度的%  3.正负值 margin-left:正值，元素向右移动  负值，元素向左移动  margin-top:正值，元素向下移动  负值，元素向上移动  4.auto 自动计算元素的左右外边距  让块级元素水平居中  auto对上下外边距无效  auto只对设置width的块级元素生效  margin取值的数量问题  margin:v1; 设置4个方向外边距  margin:v1 v2; v1:上下 v2:左右  margin:0 auto; margin:auto; 水平居中显示  margin:v1 v2 v3; 上 左右 下  margin:v1 v2 v3 v4; 上右下左 |

练习

|  |
| --- |
| 04\_ex 坑  创建两个div，尺寸都是300px\*300px  背景颜色随意  设置两个div之间的间距为50px  1.单独设置一个div 50  2.每个div都设置25  3.一个div设置30，一个div设置20 |

2.外边距的特殊效果

①外边距合并

|  |
| --- |
| 两个垂直外边距相遇时，他们将合并成一个  值以大的为准  解决方案：页面布局设计的时候，尽量的去规避 |

练习

|  |
| --- |
| 05\_ex  两个div，宽高200px.设置不同背景颜色  为两个div设置4个方向的外边距，f12观察效果  两个span，内容1111 2222.设置不同背景颜色  为两个span设置4个方向的外边距，f12观察效果  两个input  为两个input设置4个方向的外边距，f12观察效果 |

②关于块级元素，行内元素，行内块的总结

|  |  |  |
| --- | --- | --- |
| 块级 | 行内元素 | 行内块 |
| 设置宽高有效。如果不设置宽，宽度是父元素宽度的100%。如果不设置高，高度靠内容撑开 | 设置宽高无效。宽高靠内容撑开。 | 设置宽高有效。不设置，自带默认宽高。 |
| 独占一行 | 与其他行内元素/行内块共用一行 | 与其他行内元素/行内块共用一行 |
| 上下外边距有效 | 上下外边距无效 | 上下外边距有效。  但是，一行中一个行内块的上下外边距发生改变，整行元素，都会跟着一起移动 |

③默认自带外边距的元素

|  |
| --- |
| body h1~h6 ul ol dd p 不同浏览器，给这些元素的默认外边距不同  所以一般开发之前，我们要做css reset.  \*{margin:0;padding:0} |

练习

|  |
| --- |
| 06\_ex 坑 外边距溢出  两个div，id-->#d1,#d2  尺寸300px\*300px,设置不同的背景色  在#d2中添加div#d3  尺寸 100px\*100px,添加不同的背景色  给#d3设置上外边距50px.查看效果 |

作业：

1.ajax的注册模块

2.块级元素，行内块，行内元素总结

day10

五.盒子模型

2.外边距的特殊效果

④外边距溢出

|  |
| --- |
| 在特殊情况下，为子元素添加上外边距，会作用到父元素  特殊情况：1.父元素没有上边框  2.设置上外边距的子元素的内容区域的上沿与父元素内容区域的上沿重合  （为第一个子元素设置上外边距时，这种说法不严谨）  解决方案：1.给父元素添加上边框，  弊端，会影响父元素实际占地高度  2.给父元素添加上内边距  弊端，会影响父元素实际占地高度  3.给父元素设置overflow:auto/hidden  弊端，为了解决外边距溢出，而增加了新的限制  4.在父元素中，第一个子元素位置，添加一个空的table标签  <div id="d2">  <table></table>  <div id="d3"></div>  </div> |

⑤外边距使用时机

|  |
| --- |
| 1.元素位置发生改变，可以使用外边距  2.元素与元素之间有空隙，这个空隙用外边距 |

3.内边距padding

|  |
| --- |
| 边框与内容区域之间的距离  改变内边距，感觉上是改变了元素的大小，  内边距有颜色，颜色同元素背景色  padding:value; 同时设置4个方向的内边距  padding-top  padding-right  padding-bottom  padding-left  取值数量  padding:v1;  padding:v1 v2; 上下 左右 但是padding没有auto  padding:v1 v2 v3; 上 左右 下  padding:v1 v2 v3 v4; 上右下左 |

4.box-sizing(重要)

|  |
| --- |
| 浏览器默认的盒子模型，元素实际占地宽度  左外边距+左边框+左内边距+内容区域宽度+右内边距+右边框+右外边距  box-sizing：设置盒子模型的计算公式  1.默认公式 content-box  2.border-box  设置width是border+padding+内容区域的总宽度  实际占地宽度：左外边距+width+右外边距  如果元素的尺寸，使用%设置，你要留心，是不是需要设置border-box |

六.背景相关样式

1.背景颜色

|  |
| --- |
| background-color:#ff0; |

2.背景图片

|  |
| --- |
| background-image:url(image/07.png); |

3.背景图片的平铺

|  |
| --- |
| background-repeat:  取值 1.repeat 默认值 平铺  2.no-repeat 不平铺  3.repeat-x 水平平铺  4.repeat-y 垂直平铺 |

4.背景图片的定位

|  |
| --- |
| background-position:  取值 1.x y 以px为单位的数字，定义x轴和y轴的位置  2.x% y%  3.x 或者 x%,不写y，默认y为50%  4.关键字 x：left/center/right y:top/center/bottom |

5.设置背景图片尺寸

|  |
| --- |
| background-size:  取值：1.x y 以px为单位的数字  2.x% y%  3.只取一个值，同时设置宽高  4.cover 要求图片把容器全部覆盖，图片会显示不完整  5.contain 要求容器包含完整的图片，图片显示完整，  但是容器会有空白区域 |

6.背景图片的固定

|  |
| --- |
| background-attachment:  取值：1.scroll 默认值，背景图定位以容器为准  背景图会随着页面滚动条滚动而变化  2. fixed 固定，背景图定位以body为准  页面的滚动条，滚动，背景图不会随着发生位置变化  但是，只能在本容器区域内显示 |

7.背景的简写方式

|  |
| --- |
| background:color image repeat attachment position;  最简方式 background:color/image;  background:#f00;  background:url(..png);  注意：如果想在已有的背景设置中，添加或修改属性  不能使用简写方式修改，不然会覆盖之前的样式 |

七.渐变 gradient

1.什么是渐变

|  |
| --- |
| 渐变，就是指多种颜色，平缓变化一种显示效果 |

2.渐变的主要因素

|  |
| --- |
| 色标：一种颜色，及其出现的位置  一个渐变，至少有两个色标 |

3.渐变的分类

①线性渐变 以直线的方式来填充渐变色

②径向渐变 以圆形的方式来填充渐变色

③重复渐变 将线性渐变，或者径向渐变，重复几次实现填充

4.线性渐变

|  |
| --- |
| background-image  background-image:linear-gradient(方向,色标1,色标2,........);  方向：取值 1. to top ↑  to right →  to bottom ↓  to left ←  2.角度 0deg == to top  90deg== to right  180deg== to bottom  270deg== to left  色标： 颜色 位置  位置的写法1. #f00 10px px为单位的数字  2. #f00 10% %  3.只有两种颜色，分别在0%和100%，可以省略位置 |

5.径向渐变

|  |
| --- |
| background-image:radial-gradient(半径 at 圆心,色标1..........);  半径：以px为单位的数字  圆心：x y 以px为单位的数字  x% y%  关键字 x:left/center/right y:top/center/bottom  关于色标，注意，如果色标的位置，使用px为单位的具体数字。  那么这个径向渐变的半径，就失去作用了 |

6.重复渐变

|  |
| --- |
| 重复的线性渐变  background-image:repeating-linear-gradient(方向,色标.....);  重复的径向渐变  background-image:repeating-radial-gradient(半径 at 圆心,色标.....); |

7.渐变的浏览器兼容性问题

|  |
| --- |
| 如果想兼容ie8以下，写兼容代码，写兼容的过程，css hack  第一步，添加浏览器内核  -webkit- chrome/safari  -moz- firefox  -ms- IE  -o- opera  注意：添加了浏览器内核的线性渐变，他的方向，只能写起点，不能加to  -webkit-linear-gradient(left,#f00 0%,#00f 100%);  -ms-linear-gradient(left,#f00 0%,#00f 100%);  -moz-linear-gradient(left,#f00 0%,#00f 100%);  -o-linear-gradient(left,#f00 0%,#00f 100%); |

八.文本格式化（重点）

1.字体属性

①字号大小

|  |
| --- |
| font-size: 以px/pt/em/rem为单位的数字 |

②字体系列

|  |
| --- |
| font-family：系列1，系列2，系列3......  取值：设备中字体库里有的字体，如果字体名称有空格，需要加""  用,隔开多个字体类型 |

③字体加粗

|  |
| --- |
| font-weight:  取值 1.关键字 lighter normal bold bolder  2.无单位的数字，必须是100的整倍数，最大值1000 |

④字体样式

|  |
| --- |
| font-style:  取值：1.normal 默认值 正常 无样式  2.italic 斜体 |

⑤小型大写字母

|  |
| --- |
| font-variant:small-caps; |

⑥字体属性的简写方式

|  |
| --- |
| font:style variant weigth size family;  最简方式：font:size family; |

样式编写步骤

|  |
| --- |
| 找目标元素，从上往下，从左往右， 先写外后写里  1.先写尺寸，大体位置  2.边框和背景相关  3.文本相关  4.微调 |

作业：

1.完成学子登陆页面



day8.9

## 八,文本格式(重点)

### 1.字体属性

#### (1).字号大小

|  |
| --- |
| font-size: 以px/pt/em/rem为单位的数字 |

#### (2)字体系列(字体)

|  |
| --- |
| font-family: 系列1,系列2,系列3........  取值: 设备中字体库里有的字体,如果字体名称中有空格,需要加"",用逗号隔开多个字体类型 |

#### (3)字体加粗

|  |
| --- |
| font-weight:  取值 :1.关键字lighter normal常规(默认值) bold加粗 bolder  2.无单位的数字,必须是100的整倍数 最大值1000 |

#### (4)字体样式

|  |
| --- |
| font-style:  取值: italic斜体  normal常规(默认值,正常,无样式) |

#### (5)小型大写字母

|  |
| --- |
| font-variant:small-caps; |

#### (6)字体属性的简写方式

|  |
| --- |
| font:style variant weight size family  最简方式:**font:size family;** |

2.文本属性

（1）字体颜色：

color

（2）文本的水平对齐方式

text-align:center/left/right/justify(两边对齐)

text-align;设置的是当前元素的子元素的水平对齐（子元素是块级无效，行内元素有效）

margin:0 auto;设置当前的元素居中

（3）行高垂直居中：

|  |
| --- |
| line-height:  特点：如果行高大于字体的大小，那么文字将在行高的中间显示，呈垂直居中显示  取值：1.以px为单位的数字，行高的取值和容器的高度相同  如果文字较多，一行放不下，不建议使用行高居中，文字会溢出。  2.无单位的数字（） |

（4）文本线条修饰

|  |
| --- |
| Text-decoration:  取值：overline 上划线  Underline 下划线  Line-through 删除线  None 无线条 |

（5）首行缩进

|  |
| --- |
| Text-indent:以px为单位 |

(6)文本阴影：

|  |
| --- |
| Text-shadow: h-shadow v-shadow blur color  h-shadow: 水平偏移  v-shadow 垂直偏移  blur 模糊度  color 阴影颜色 |

九.表格相关的样式

1.表格的样式

|  |
| --- |
| 之前我们学习的属性，基本都可以使用  尺寸，边框，背景，字体，文本，内外边距 |

2.td/th

|  |
| --- |
| 尺寸，边框，背景，字体，文本，内边距都可以设置  但是，外边距无效  vertical-align:top/middle/bottom |

3.表格的表现形式

|  |
| --- |
| Table是一种特殊的显示方式、  表格的实际尺寸，是根据内容数据多少决定的  单元格小，内容多，那么撑开  内容少，单元格设置的尺寸大，就按设置的尺寸展示  表格的特点  1.表格每一行，对应的列一样宽，其中一个变宽，其他行的对应列跟着改变  2. 表格每一列，对应的行一样高，其中一个变高，其他列的对应行跟着改变 |

4.表格的专有属性

1）边框合并

|  |
| --- |
| border-collapse:  取值：separate 默认值，边框分离  collapse 边框合并 |

2）边框与边框的距离

|  |
| --- |
| border-spacing: 10px 20px;  一个值表示同时设置4个方向，2个值表示设置水平和垂直的方向的值  注意：如果边框是合并状态，此属性失效 |

3）表格的标题

|  |
| --- |
| <caption></ caption >  caption-side:top/bottom; 该css属性放在table样式中 |

4）设置table的显示规则

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| table-layout:auto(缺省值)/fixed  取值：  1.auto 自动布局的表格  2.fixed 固定布局的表格   |  |  | | --- | --- | | 自动布局 auto | 固定布局 fixed | | 单元格的大小，匹配的内容和设置的尺寸，大的一方 | 单元格大小，取决于设置值 | | 先把所有读取到的内存然后一次性加载 | 按浏览器默认加载方式加载 | | 表格复杂时，加载速度慢 | 在任何情况下，都是加速加载的 | | 自动布局比较灵活 | 固定表格布局不够灵活 | | 适用于不确定每列大小，并且不复杂的表格 | 使用于确定每列尺寸的表格 | |

三.定位

改变元素在页面中位置

1.分类

|  |
| --- |
| 1.普通流定位  2.浮动定位  3.相对定位  4.绝对定位  5.固定定位 |

6.高度坍塌

|  |
| --- |
| 块级元素的高度，如果不设置，默认高度是靠内容撑开  但是，块级元素的内部所有元素，都浮动了（脱离文档流）  块级元素，认为内部，没有元素了，所以高度撑不起来  发生了高度坍塌  解决方案：  1.设置高，很多时候，不能准确计算出高度  2.父元素也浮动，弊端：会影响父元素的同级元素  3.父元素写overflow:hidden。如果父元素需要溢出显示，冲突 |

day8.11

四.显示

1.显示方式

|  |
| --- |
| 显示方式：决定了元素在网页中的表现形式（块级，行内块，行内，table）  Display:  取值：1.none 元素不显示，脱离文档流  2.block 让元素按照块级的方式显示  特点：独占一行。可设置宽高和上下外边距  3.inline 让元素按照行内的方式显示  特点：共用一行，宽高和上下外边距无效  4.inline-block 让元素按照行内块的方式显示  特点：共用一行，可设置宽高和上下外边距  5.table 让元素表现和table一样  特点：尺寸以内内容和设置宽度大的一个为准  独占一行，允许修改尺寸  项目中，修改显示方式，多用于  1.a标签设置按钮  2.显示/隐藏效果，都需要 display |

作业：用a标签写按钮

a

2.显示效果

|  |
| --- |
| 显示/隐藏  visibility  取值： 1.visible 可见的  2.hidden 隐藏  面试题：visibility和display的区别  display：none 隐藏，脱离文档流，不占页面空间  visibility:hidden 隐藏，不脱离文档流，占页面空间 |

3.透明度

|  |
| --- |
| opacity  取值：0~1 1 不透明； 0 全透明  面试题：rgba和opacity的区别  rgba只会改变某一种颜色的透明度  作用于元素，此元素中（包括后代元素），只要跟元素颜色相关的都会变透明 |

4.垂直对齐方式

|  |
| --- |
| Vertical-align  只会在两个地方使用  1.table 中  取值：top/middle/bottom  设置td中内容的垂直对齐方式  2.img中  设置img前后的文字，与img的对齐方式  Vertical-align 取值 top/middle/bottom/baseline基线（默认）  通常会将所有图片的垂直对齐，设置为非基线的方式 |

5.光标的设置

|  |
| --- |
| cursor：  取值：  default 箭头  Pointer 小手  Crosshair +光标  Text 文本  Wait 等待  Help 帮助 |

五.列表的样式

|  |
| --- |
| List-style-type:  取值：none /circle/ disc/square  list-style-image:url(…);  list-style-position:outside(默认值)/inside  简写方式  List-style:type image position  项目中使用最多的写法：list-style:none; |

七.定位--相对，绝对，固定定位

|  |
| --- |
| position  取值：static 默认值，默认文档流，静态  relative 相对定位  absolute 绝对定位  fixed 固定定位  已定位元素：当一个元素被position，并且取值为relative，absolute，fixed其中一种时，这个元素就被称为已定位元素  元素一旦变成已定位元素，会解锁4个偏移属性：top left bottom right  属性值  Top: 正：往下移动，负：往上移动。  Left: 正：往右移动，负：往移动。  bottom: 正：往上移动，负：往下移动。  right: 正：往左移动，负：往右移动。 |

1.相对定位

|  |
| --- |
| position：relative;  配合4个偏移属性，实现位置的移动  特点：  1.相对原始位置偏移某个距离  2.相对定位不脱离文档流，后面的元素不会上前补位  3.如果元素，只定义了position：relative，而没有写偏移属性  不影响任何布局效果，跟没写定位一样。  使用相对定位的场合  1.元素位置微调，类似margin  2.为子元素做绝对定位时，提供参照物 |

2.绝对定位

|  |
| --- |
| Position：absolute; 配合4个偏移属性，实现位置的移动  特点：  1.绝对定位时脱离文档流，后面元素会上前补位  2.绝对定位的偏移（相对已定位的父元素（没有就以body），偏移同相对定位（relative）一样）  1）没有已定位祖先级元素，相对于body的左上角偏移  2）绝对定位的元素，会相对于，离自己最近的“已定位”的祖先级元素  脱离文档流：意味着4件事情：  1.不占据页面空间  2.后续元素上前补位  3.变成块级  4.没写宽高，靠内容撑开 |

定位时出现这种情况：普通文档流盒子间，只要垂直外边距直接接触就会发生合并，合并后外边距视觉高度取两个发生合并外边距中较大者。发生这一现象的有三种情况：相邻兄弟块元素间、父元素及其首子元素和末子元素间（没有内边距或边框把外边距分隔开）、空块级元素自身的合并。

解决方案：1.父元素与子元素间添加内容

2.给父元素设置border

3.父元素与子元素之间添加空table标签

3.固定定位

|  |
| --- |
| position：fixed; 配合偏移属性使用  将元素固定在页面上的某一个位置，不会随着页面 的滚动发生位置的偏移  一直固定在可视区域  特点：脱离文档流 |

4.堆叠顺序

|  |
| --- |
| 默认：HTML后写的标签，会盖在之前的定位元素  后写的堆叠顺序高  使用z-index：调整堆叠顺序  取值，整数，一般1000以下  数值越大，堆叠顺序越高  特点：1.默认，后定位的元素，堆叠顺序高（设置的z-index：1都比默认高）  2.定位的脱离文档流，和浮动的脱离文档刘不是一个体系  3.只有已定位元素，可以设置z-index  4.父子级关系，z-index无效 |

作业：完成学子1楼



**CSS CORE**

一.复杂选择器

1.兄弟选择器（只能往后找，不能往前走）

兄弟：具有相同父元素的，平级元素之间，称为兄弟元素

①相邻兄弟选择器

|  |
| --- |
| 紧紧挨在某元素后面的兄弟元素  选择器1+选择器2 { }  常用于：一组元素，除第一个以外，给其他元素设置样式 |

②通用兄弟选择器

|  |
| --- |
| 获取某元素后面所有符合要求的兄弟元素  选择器1~选择器2 { } |

2.属性选择器 attr属性

|  |
| --- |
| 允许通过元素所附带的属性及值，来匹配元素，非常准确  1. [attr] {} 匹配所有带attr属性的元素  2.[ attr1 ][ attr2].. { } 匹配满足所有attr属性的元素  3.ele[attr]{} 匹配所有带attr属性的ele元素  4.[ attr=value] 匹配所有带attr属性，并且值为value 的元素  5.ele[attr=value] 匹配所有带attr=value属性的ele元素。 |

transition：all 时间;

transform: scale(放大倍数);

1. transition 属性

|  |
| --- |
| transition 属性是一个简写属性，用于设置四个过渡属性：  transition: property duration timing-function delay;  默认值：transition：all 0 ease 0; 请始终设置 transition-duration 属性，否则时长为 0，就不会产生过渡效果  transition-property：规定设置过渡效果的 CSS 属性的名称。  transition-duration：规定完成过渡效果需要多少秒或毫秒。  transition-timing-function：规定速度效果的速度曲线。  transition-delay：定义过渡效果何时开始。 |

2. transform: 元素应用 2D 或 3D 转换。该属性允许我们对元素进行旋转、缩放、移动或倾斜。

day8.13

一.复杂选择器

⑤模糊属性值匹配

|  |
| --- |
| [title^=value] {} 匹配属性值以value开头的元素  [title$=value] {} 匹配属性值以value结尾的元素  [title\*=value] {} 匹配属性值中有value的元素  [title~=value] {} 匹配属性值中有value这个独立单词的元素 |

3.伪类选择器

|  |
| --- |
| ：link :visited :hover :active :focus |

①目标伪类

|  |
| --- |
| 让被激活的锚点，应用的样式  选择器：target{} |

练习：点击菜单，ul>li>a 菜品 img

3张菜品图

②结构伪类

|  |
| --- |
| 通过结构关系找到元素  1.ele.first-child{}  匹配的是ele元素父元素的第一个孩子，还必须是ele标签  2.ele.last-child{}  匹配的是ele元素父元素的最后一个孩子，还必须是ele标签  3.ele.nth-child(n){} n:第几个孩子，从1开始  匹配的是ele元素父元素的第n个孩子，还必须是ele标签 |

练习：

|  |
| --- |
| 03\_ex 创建4\*4的表格，400px\*400px  通过结构式 伪类，设置样式：  第一行背景颜色  最后一行背景颜色  第三行的第二列的背景颜色 |

③：empty

|  |
| --- |
| 匹配内部没有任何元素的标签  连空格和回车都不能有，才叫空 |

④：only-child

|  |
| --- |
| 匹配属于其父元素的唯一子元素 |

⑤否定伪类 :not

|  |
| --- |
| ele:not(selector){}  将满足selector条件的元素都排除 |

⑥伪元素选择器

|  |
| --- |
| 1.::first-letter或者:first-letter  匹配该元素的首字符  2.::first-line 或者:first-line  匹配该元素的首行  首字符和首行，样式冲突，遵循首字符样式  3.::selection  匹配用户选取内容的样式，必须写两个冒号  注意：只能修改文本颜色和背景颜色，其他样式无效 |

4.伪元素选择器—内容生成

|  |
| --- |
| 使用css命令，添加的HTML元素，称为伪元素  1.::before或者:before  在当前元素的内容区域中，最前边添加一个假的元素  我们可以给这个假的元素，定义显示方式display  还有其他样式，但是，这个假元素的内容，使用content添加。  注意：content中，只能写字符串和url(img)  #d1:before{  display:inline;  color:#ff0;  background:#00f;  content:'我是添加的块级';  width:100px;height:100px;  }  2. ::after或者:after  在当前元素的内容区域中，最后边添加一个假的元素  我们可以给这个假的元素，定义显示方式display  还有其他样式，但是，这个假元素的内容，使用content添加。  注意：content中，只能写字符串和url(img) |

①伪元素内容生成，能够解决的问题- - -外边距溢出问题

|  |
| --- |
| #parent:before{  display:table;  content:'';  }  这里的table不是table标签，而是把这个跟元素设置为table的显示方式 |

②伪元素内容生成，能够解决的问题- - -高度坍塌

|  |
| --- |
| .clear:after{  display:block;  content:'';  clear:both;  }  封装一个清除浮动的class样式。 |

作业：webstorm 10的安装

①如果不听的弹出要求注册，需要断网

②使用注册机，生成注册码

二.弹性布局

1.什么是弹性布局？

|  |
| --- |
| 弹性布局，是一种布局方式  主要解决，某个元素中，子元素的布局方式  为布局提供了很大的灵活性 |

2.弹性布局的概念

①容器

|  |
| --- |
| 要发生弹性布局的元素，他们的父元素称之为容器  就是设置了display:flex那个元素，就是容器 |

②项目

|  |
| --- |
| 要发生弹性布局的元素们，叫做项目  就是设置了display元素的子元素们 |

③主轴

|  |
| --- |
| 项目们排列的方向叫做主轴  主轴可以横向排列，也可以纵向排列  项目们排列的方向是指向主轴的起点到终点 |

④交叉轴

|  |
| --- |
| 与主轴垂直的方向，项目们在交叉轴上排列  交叉轴的起点和终点 |

3.语法

|  |
| --- |
| display:  取值：  1.flex 将块级元素设置为弹性容器  2.inline-flex将行内元素设置为容器  将元素设置了弹性容器之后，这个元素所有的子元素，将变成弹性项目  这些项目，都允许按照弹性布局的方式排列  特点：  1.元素设置为容器之后  容器的text-align,vertical-align失效  项目的float，clear失效  2.项目是可以修改尺寸的 |

4.容器的属性

①主轴的方向

|  |
| --- |
| flex-direction:  取值：  1.row默认值，主轴是x轴，主轴的起点左端  2.row-reverse 主轴是x轴，主轴的起点右端  3.column 主轴是y轴，主轴的起点上端  4.column- reverse 主轴是y轴，主轴的起点下端 |

②设置项目的换行

|  |
| --- |
| Flex-wrap:  取值：  1.nowrap 默认值。空间不够，项目进行压缩，不换行  2.wrap 空间不够就换行，往下挤  3.wrap-reverse 空间不够就换行，往上挤 |

③上面两个属性合体

|  |
| --- |
| Flex-flow：direction wrap; |

④定义项目在主轴上的对齐方式

|  |
| --- |
| justify-content  取值：  1.space-between 两端对齐  5.space-around 两端有空白 |

⑤项目在交叉轴的对齐方式（弹性布局项目的垂直对齐）

|  |
| --- |
| align-items:  设置所有项目在交叉轴上的对齐方式  取值：  1.flex-start 默认值，交叉轴起点对齐  2.flex-end 交叉轴终点  3.center 交叉轴中心  4.baseline 交叉轴的基线对齐，与起点一致  5.stretch 前提 项目不能写高度，占满交叉轴上所有的空间 |

5.项目的属性

①order

|  |
| --- |
| 定义项目在主轴上的排列顺序  值越小，越靠近起点，默认值为0；  取值，无单位整数 |

②flex-grow

|  |
| --- |
| 定义项目中放大比例  如果容器有足够大的剩余空间  取值，无单位整数，默认值0，不放大，取值越大，最后占的空间越大 |

③flex-shrink

|  |
| --- |
| 定义项目的缩小比例  如果容器剩余空间不够，项目将缩小  取值，无单位的整数，默认值1，取值越大，最后占的空间越小 |

④某一个项目在交叉轴上的对齐方式

|  |
| --- |
| align-self:  取值：  1.flex-start 交叉轴起点对齐  2.flex-end 交叉轴终点  3.center 交叉轴中心  4.baseline 交叉轴的基线对齐，与起点一致  5.stretch 项目不能写高度，占满交叉轴  6.auto 使用容器定义的align-items的值 |

三.CSS hack

|  |
| --- |
| 由于不同浏览器对css的解析认知不同，  会导致一份css在不同浏览器生成的页面效果不同  针对这种情况，开发人员要写不同代码去适应不同浏览器  编写不同代码的过程就叫做写css hack  -webkit-  -o-  -ms-  -moz-  Tmooc有css hack的拓展视频 |

四.转换（重点\*\*\*\*\*\*）

1.什么是转换？

|  |
| --- |
| 改变元素在页面中的位置，大小，角度，以及形状  2D 只在X轴和Y轴发生的转换效果  3D 在2D的基础上，增加了Z轴的效果，页面上的3D效果都是模拟的 |

2.转换属性

|  |
| --- |
| transform：转换函数  取值：  1.none 默认值，无任何转换  2.transform-function 转换函数，可以有1个或者多个函数 |

3.转换原点

|  |
| --- |
| transform-origin:  取值：  1.以px为单位的数字  1.%  3.关键字x(left/center/right) y(top/center/bottom)  取值数量：  2个值，表示原点在x轴和y轴上  3个值，表示原点在x轴和y轴和z轴上  默认值，原点在元素中心位置（center center）/（50% 50%） |

4.2D转换

位移

缩放

旋转

倾斜

day8.14

四，转换

4.2D转换

①位移

|  |
| --- |
| transform:translate(400px);  取值：  1.translate（x） 元素在x轴上位移的距离  2.translate（x,y） 元素在x轴上位移的距离  X: +:→ -: 向左  Y  3.  translateX(x) 单独设置x轴  translateY(y) 单独设置y轴 |

②缩放

|  |
| --- |
| transform: scale(-1,2);  取值：  1.scale(value);一个值：同时缩放x轴和y轴的比例  Value>1 x轴和y轴都放大  0<value<1 x轴和y轴都缩小  -1<value<0 x轴和y轴都缩小并反转  <-1 x轴和y轴都放大并反转  2. scale(x,y);两个值：分别设置缩放x轴和y轴的比例  3. scaleX(x); 单独设置x轴的缩放  4. scaleY(y); 单独设置y轴的缩放 |

③旋转：改变元素的角度

|  |
| --- |
| transform:rotate(ndeg);  n:+:顺时针; -:逆时针  坑：  d1先旋转45度，然后位移500px  d2先位移500px,再旋转45度  注意：  1.转换原点，回应旋转效果  2.旋转是连同坐标轴一起旋转的，会影响旋转之后的位移方向 |

④倾斜

|  |
| --- |
| transform:skew(ndeg);  取值：  1.skew(ndeg)等同于skewX(ndeg);  让元素的y轴，向着x轴的方向逆时针倾斜n度  2. skewY(ndeg);  让元素的x轴，向着y轴的方向顺时针倾斜n度 |

练习：

|  |
| --- |
| 01\_ex 创建div 200px\*200px 背景随意，鼠标悬停时、  该元素，向右偏移100px，向下偏移100px,旋转135deg，向x轴倾斜45deg,放大1.5倍 |

面试题

|  |
| --- |
| 使用css，让一个div在任何时候，都处于屏幕的正中心 |

5.3D转换-----3D转换是模拟的

①透视距离

|  |
| --- |
| 模拟人的眼睛，到3D转换元素之间的距离  距离不同，看到的效果也不同  Perspective:距离  这个属性，要写在3D转换元素的父元素上 |

②旋转

|  |
| --- |
| Transform:  取值：  1.rotateX(xdeg) 以x轴为中心轴，旋转元素的角度  2. rotateY(ydeg) 以y轴为中心轴，旋转元素的角度  3. rotateZ(zdeg) 以z轴为中心轴，旋转元素的角度  4.rotate3D(x,y,z,ndeg) x,y,z取值为0，代表这个轴不参与旋转  x,y,z取值大于0，代表这个轴参与旋转 ndeg：旋转角度 |

五.过渡

1.什么是过渡

|  |
| --- |
| 让css的值，在一段时间内平缓的变化  只有两个css状态的变化，使用过渡  超过两个css状态，使用动画 |

2.过渡的语法

|  |
| --- |
| ①指定过渡属性  Transition-property:  取值：  1.参与过渡的css属性名称，多个名称之间用空格分开  2.all 所有支持过渡的属性，都参与此次过渡  支持过渡的属性  1.颜色属性  2.大多数取值为具体数字的属性  3.阴影  4.转换  5.visibility  ②设置过渡的时长  Transition-duration  取值：以s/ms为单位  ③设合资时间曲线函数  transition-timing-function:ease;  取值：  1.ease 默认值 慢开，加速，慢关  2.linear 匀速  3.ease-in 一直加速  4.ease-out 一致减速  5.ease-in-out: 先加速再减速  ④过渡的延迟  Transition-delay:  取值：以s/ms为单位  ⑤过渡代码的编写位置  写在原来的选择器里，过渡效果又去优惠  写在hover里，过渡效果有去无回  ⑥过渡的简写  Transition：property duration timing-function delay  默认值：transition：all 0 ease 0; 请始终设置 transition-duration 属性，否则时长为 0，就不会产生过渡效果 |

练习

|  |
| --- |
| 翻滚吧，亮亮 |

六.动画

|  |
| --- |
| 1.超过两个样式之间的渐变，使用动画  2.不使用hover触发，刷新网页时直接开始渐变，使用动画 |

1.关键帧

|  |
| --- |
| 控制动画的每一个状态  关键帧包括：  1.动画的执行时间点  2.在这个时间点上的样式  0%{样式}  50%{样式} |

2.动画的使用步骤

①声明动画、

|  |
| --- |
| @keyframes 动画名称{  0%{样式}  50%{样式}  }  /\*声明一个动画，动画名称animation\*/  @keyframes animation{  0%{background:#00f;border-radius:0%}  25%{background:#ff0;border-radius:50%}  50%{background:#0ff;border-radius:0%}  75%{background:#f00;border-radius:50%}  100%{background:#f0f;border-radius:0%}  } |

②调用动画

|  |
| --- |
| 动画的使用  1.定义执行动画的动画名称  animation-name:animation;  2.设置动画的持续时间  animation-duration:10s;  3.设置动画的时间曲线函数  animation-timing-function:linear;  4.设置动画的延迟  animation-delay:2s; |

③动画的其他属性

|  |
| --- |
| 动画的播放次数  animation-iteration-count  取值：  1.具体的数字，数字  2.infinite 无限 |
| 动画的播放顺序  animation-direction  取值：  1.normal 默认值 0%~100%  2.reverse 100%~0%  3.alternate 轮流播放  奇数次，正向。偶数次，逆向 |
| 动画的简写方式  animation: name duration timing-function delay count delay direction  最简方式 animation: name duration |
| 设置动画的显示状态  在动画延迟时间内，显示的状态  在动画播放完成后，显示的状态  animation-fill-mode:  取值：  1.backwards 动画的延迟时间内，显示第一帧  2.forwards 动画结束后，显示最后一帧  3.both 上面两个都要  4.none 不填充，默认值 |
| 设置动画的暂停和播放  animation-play-state  paused 暂停  running 播放 |

④浏览器对动画的兼容性问题

|  |
| --- |
| IE8.0以下需要写css hack  在定义动画的时候写css hack  @keyframes 动画名称{关键帧}  @-webkit-keyframes 动画名称{关键帧}  @-o-keyframes 动画名称{关键帧}  @-ms-keyframes 动画名称{关键帧}  @-moz-keyframes 动画名称{关键帧} |

3.项目中对动画的使用

|  |
| --- |
| <https://daneden.github.io/animate.css/> 下载网址  使用animate.css,在html中link引用css文件  直接调用animate.css中的动画名称 |

七.CSS优化

1.css优化的目的

|  |
| --- |
| 1.减少服务器端的压力  2.提升用户体验 |

2.css优化的原则

|  |
| --- |
| 尽量减少http的请求个数（使用精灵sprites图）  页面的顶部引入css文件  将css和js放到单独的文件中 |

3.css代码优化

|  |
| --- |
| 合并样式（能写群组，就不单写，能用简写方式，就不单独定义属性）  缩小样式文件的大小（能重用代码就重用）  避免出现空的href和src  代码压缩 |

八.CSS reset 和normalize.css

1.什么是css reset

|  |
| --- |
| HTML标签，在浏览器中有默认样式  不同浏览器内核，对默认样式的解析不同  在开始敲代码之前，把所有浏览器默认样式全部覆盖掉  这种行为叫做css reset  使用css reset  1.自己写  2.使用别人写好的css reset文件  注意：css reset 太霸道，把需要保留的默认样式，也覆盖了 |

2.normalize.css是什么

|  |
| --- |
| normalize.css在默认的html元素的样式上，提供了跨浏览器的高度一致性  相比于传统的css reset，normalize.css是一种现代的，为html5准备，更优质的解决方案  normalize.css是一种css reset的替代方案  保护有用的浏览器默认样式，而不是完全的去覆盖  为大部分html元素提供了样式  修复了浏览器自身的一些bug,并保证不同浏览器的一致性  优化css可用性：添加了一些小技巧  有详细的注释和文档 |

作业：

1.webstorm安装能正常使用，并注册

2.看一遍normalize.css代码

3.动画完成电子时钟

4.学子首页的1L的过渡变大

day8.15

**bootstrap**

一.响应式布局（css 2010）

1.响应式布局是什么

|  |
| --- |
| responsive web page 响应式/自适应网页  可以根据浏览设备不同（pc , pad , phone, print….）  而自动改变布局，图片，文字效果，不会影响用户体验 |

①库：jquery.js, axios,animate.css 等等

库是一个封装好各种功能的函数群的脚本，引用脚本直接使用功能，是加快开发速度的工具。

②框架：Vue.js, angular.js等等

框架：是一个半成品的项目，按照半成品的结果，往里面添加新的内容，做精加工。

2.写响应式布局的要求

|  |
| --- |
| 1.布局:  尽量对元素宽度，不写绝对值  必须是流式布局（默认文档流+浮动） 1rem:16px;  2.文字和图片大小随容器的大小改变  3.使用css3提供的媒体查询技术 |

|  |
| --- |
| 响应式布局存在问题：  1.代码量和复杂程度，几何性的增加  2.复杂的页面不适合使用响应式 |

3.如何测试响应式网页

①使用真实设备测试

|  |
| --- |
| 好处，真实可靠  缺点，成本高  测试任务量巨大 |

②使用第三方的软件测试，虚拟机

|  |
| --- |
| 好处，不需要太多真实设备，测试方便  坏处，测试效果有限，有待进一步验证 phonegap |

③使用谷歌浏览器自带的模拟软件

|  |
| --- |
| 好处，简单方便  缺点，测试效果极其有限 |

4.编写响应式布局

①手机适配

|  |
| --- |
| 代码不需要背，你写的页面，如果不需要在移动端显示，下面代码就不需要写  <meta name=”viewport” content=”width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0”>  width=device-width 设置视口宽度为设备宽度  initial-scale=1.0 设置视口宽度能否缩放，1代表1倍，不缩放  maximum-scale=1.0 允许缩放的最大比例  user-scalable=0 是否允许用户手动缩放，1允许，0不允许  最简方式：  <meta name=”viewport” content=”width=device-width,initial-scale=1”> |

②所有内容/文字/图片尽量使用相对尺寸，尽量不使用绝对值

③使用媒体查询，配合流式布局+弹性布局，完成响应式布局

④媒体查询技术

|  |
| --- |
| css3 media query,做响应式的必备技术  媒体media：浏览器页面的设备  设置：  screen(pc/pad/phone)  TV  print（打印机，包括洗衣机，微波炉等等显示设备）  媒体查询的原理：自动根据当前浏览设备的不同（尺寸，方向，解析度不同），有选择性的，执行一部分css，忽略其他部分  语法：<style>  @media screen and (min-width:992px) and (max-width:1199px){  选择器{样式}  选择器{样式}  …  }  </style> |
| 开发常用屏幕尺寸：  1.超大屏 xl > 1200px 条件：min-width:1200px  2.大屏 lg 992px~1199px 条件：min-width:992px and max-width:1199px  3.中屏md 768~991  4.小屏 sm 576~767  5.超小屏 xs < 575px  响应式布局，基本上所有元素都要改变盒子模型计算方式  使用box-sizing:border-box; |

二.bootstrap

|  |
| --- |
| Boot给我们提供了好多封装好的样式，我们使用类名调用  但是项目中，需求比较繁琐，boot提供的样式不够，很多时候需要我们自己手写样式和手写媒体查询  <https://www.bootcss.com/> boot官网  bootstrap分为5个部分  1.如何使用boot  2.全局css  3.组件和js插件  4.定制scss  5.boot项目 |

1.全局css样式

|  |
| --- |
| .container 定宽容器，不同屏幕尺寸的最大宽度写死了  .container-fluid 变宽容器，容器宽度永远是屏幕宽度的100%  boot支持4个屏幕的响应式  xl-lg-md-sm |

①按钮相关的class

|  |
| --- |
| .btn 基本类  不同颜色按钮  .btn-success 绿色  .btn-danger 红色  .btn-warning 黄色  .btn-info 信息 黄色  .btn-primary 主要的 蓝色  .btn-secondary 次要的 灰色  .btn btn-dark 深色  .btn-light 浅色  只有边框的按钮  .btn-outline-secondary/danger/...  按钮的大小  .btn-sm 小按钮  .btn-lg 大按钮  .btn-block 块级按钮  .btn-link 链接按钮 |

②图片相关class

|  |
| --- |
| .rounded 0.25em的圆角  .rounded-circle 50%的圆角，圆形  .img-thumbnail 有内边距和边框，缩略图  .img-fluid 响应式图片，图片可以缩放，但是不会超过原始尺寸的大小 |

③文字相关

|  |
| --- |
| 文本颜色  .text-danger… 文本字体红色…  .text-muted 文本字体灰色  文本大写，小写，首字母大写  text-uppercase/text-lowercase/text-capitalize  字体的加粗  font-weight-bold/light/normal  字体大小  .h1~.h6 |

④列表相关

|  |
| --- |
| .list-unstyled 去掉点，左边距清0  ul.list-group>li.list-group-item 创建列表组和列表项  列表项颜色  List-group-item-danger/warning…  激活项 .active  禁用项 .disabled |

⑤表格相关 class

|  |
| --- |
| .table 基本类 对table本身和后代写样式  .table-border 带边框的表格，为自己和后代添加边框  .table-danger/warning… 表格的背景颜色  .table-hover 带鼠标悬停效果的表格  .table-striped 奇数行变色  .table-responsive-xl/lg/md/sm 响应式的表格，必须写在table的父元素上 |

作业：

1.手写媒体查询

2.手写课后作业

webstorm的一些设置

1.快捷方式。file—setting—keymaps---eclipse

2.字体设置

day8.16

二.bootstrap

1.全局css样式

2..辅助类

①边框

|  |
| --- |
| . border .border-top/left/… 基本类  . border-danger/… 边框颜色  .border-0 . border-top-0/… 清除各向边框 |

②浮动

|  |
| --- |
| float-\*-left/right/none \*:xl/lg/md/sm  .clearfix 解决高度坍塌 |

③显示

|  |
| --- |
| .visible 显示  .invisible 隐藏 visibility:visible/hidden 另外有一个display:none这个不占文档空间 |

④背景颜色

|  |
| --- |
| bg-danger/… |

⑤圆角

|  |
| --- |
| .rounded/.rounded-0/.rounded-circle  .rounded-top/left/right/bottom 单方向设置圆角，一次设置两个角 |

⑥内外边距

|  |
| --- |
| m/ml/mt/mr/mb/mx/my-\*-0/1/2/3/4/5/auto mx:同时设置左右外边距（x轴） my同时设置上下外边距（y轴）  \*:xl/lg/md/sm  0 1:0.25rem 2:0.5rem 3:1rem 4:1.5rem 5:3rem  p/pl/pt/pr/pb/px/py-\*-0/1/2/3/4/5/auto mx:同时设置左右内边距（x轴） my同时设置上下外边距（y轴）  \*:xl/lg/md/sm  0 1:0.25rem 2:0.5rem 3:1rem 4:1.5rem 5:3rem |

⑦尺寸

|  |
| --- |
| w/h-25/50/75/100 25% 50% 75% 100%  boot没有封装min-width |

3.栅格布局（重点\*\*\*\*\*\*）

①web页面的布局方式，一般3种

|  |  |  |
| --- | --- | --- |
| table布局 | div+css | Boot中栅格 |
| 简单，容易控制 | 语义正确  渲染效率高 | 简单，容易控制  语义正确  效率高  支持响应式 |
| 语义错误  渲染效率低 | 控制起来很麻烦 | 复杂页面不适用 |

②语法

|  |
| --- |
| div.row>div.col-n n:1-12  .row 弹性，换行，主轴是x轴 **带左右-15px外边距**  .col-n **带左右+15px内边距**  注意：  1.必须要放在.container或者.container-fluid  2.每一行都是一个row（弹性，主轴x，左右-15px外边距）  3.在row中声明col占几份（col:左右15px内边距） |

③响应式栅格

|  |
| --- |
| Col-\*-n \*:xl/lg/md/sm n:0~12 |

④不同屏幕有向上兼容的问题

|  |
| --- |
| sm 有效的屏幕 sm/md/lg/xl  md 有效的屏幕 md/lg/xl  lg 有效的屏幕 lg/xl  xl 只有xl有效 |

⑤.col类

|  |
| --- |
| 使用.col，不添加数字，自动的处理布局  每一个.col平局分配，同一行的个数可以超过12个 |

⑥嵌套

|  |
| --- |
| <div class="row">  <div class="col-6">  <div class="row">  <div class="col-1"></div>  </div>  </div>  </div>  .row写直接子类只能是.col，ul的直接子类只能是li |

⑦列偏移

|  |
| --- |
| .offset-\*-n n:1~11 \*:xl/lg/md/sm |

作业：

1.使用栅格布局完成03\_ex.html

2.完成再往前一个响应式布局

2.使用栅格布局完成学子商城1楼（图片用浮动）

4.弹性布局

|  |
| --- |
| d-\*-block/inline/inline-block/flex/table  响应式设置display  使用d-\*-flex/inline-flex 设置一个弹性布局容器 |

①设置主轴方向

|  |
| --- |
| flex-\*-row/column/row-reverse/column-reverse |

②项目在主轴上的对齐方式

|  |
| --- |
| Justify-content-\*-start/center/end/between/around  Boot对弹性布局，进行了封装 |

day8.19

二.bootstrap

5.表单

①表单元素的排列方向

|  |
| --- |
| .form-inline 内联表单 表单元素横向显示  .form-group 堆叠表单，表单元素纵向显示 |

②表单元素的样式

|  |
| --- |
| 关于文本框和密码框  .form-control 是文本框丶密码框的基本类  块级，w100，字体，背景，边框，过渡  .col-form-label 设置内边距和字号大小  .col-form-label-lg 大  .col-form-label-sm 小  关于checkbox,需要父子级，两个类配合使用  div.form-check>input.form-check-input  父级相对定位，子级绝对定位，偏移微量  .form-text 块级，上外边距 |

注意：

|  |
| --- |
| 使用栅格的原理，要求父容器  1.设置成弹性容器，display:flex;  2.主轴可换行，flex-wrap:wrap;  满足这两个条件，容器中就可以使用col所有的类 |

三.组件

|  |
| --- |
| angular 模块开发  boot封装好了一个一个的功能，不需要我们单独写js代码  可以直接实现 |

1.按钮组

|  |
| --- |
| 外层添加一个div，类名如下  .btn-group 横向按钮组 d-flex row  . btn-group-vertical 垂直按钮组 d-flex column  调整按钮组大小  btn-group-lg/-sm |

2.信息提示框

|  |
| --- |
| .alert 基本类  .alert-danger/… 颜色  div.alert-dismisssible右边距4rem  >span.close 右浮动 配合使用控制小叉子的位置 |
| 事件 boot对事件的封装之后，我们调用的方式，发生了改变  跳用boot封装事件的步骤  1.使用自定义的属性和值，调用事件（调用的事件是什么操作）  给span添加自定义属性  <span data-dismiss=”alert”>点我隐藏我亲爹</span>  2.确定事件发生后，改变的目标是谁  点击事件之后，改变目标是直系的父级，不需要写确认目标的代码 |

3.下拉菜单

|  |
| --- |
| 1.基本结构  div.dropdown relative  >button.dropdown-toggle 添加向下的小箭头  +ul.dropdown-menu display:none  2.事件  Button添加data-toggle=”dropdown”  事件目标，不需要写 |

4.导航

①水平导航

|  |
| --- |
| ul.nav x轴的弹性,所有弹性相关的样式，都可以设置  >li.nav-item 本身没有任何样式，需要父元素添加.nav-justified  . nav-justified与.nav-item配合  让每一个li等宽显示  >a.nav-link 块级，内边距和伪类的设置 |

②选项卡

|  |
| --- |
| 1.导航部分基本结构  ul.nav 添加.nav-tabs变成选项卡导航  li.nav-item  a.nav-link 添加a.active或者li.show让选项卡激活（两者二选一）  内容部分的基本结构  div.tab-content>div.tab-pane  tab-content与tab-pane配合，让子div的display:none  让子div添加active可以让当前内容显示  2.事件  1.事件的激活方式  给a标签添加自定义属性data-toggle=”tab”  2.事件的目标  Div.tab-pane添加id ex：id=”tab1”  在a标签的href中绑定对应的id  <a href=”#tab1”></a> |

练习：

|  |
| --- |
| 选项卡导航 中国 美国 泰国 印度  内容自己写 |

③胶囊导航

|  |
| --- |
| 1.导航部分基本结构  ul.nav 添加.nav-pills变成胶囊导航  li.nav-item  a.nav-link 添加a.active或者li.show让选项卡激活（两者二选一）  内容部分的基本结构  div.tab-content>div.tab-pane  tab-content与tab-pane配合，让子div的display:none  让子div添加active可以让当前内容显示  2.事件  1.事件的激活方式  给a标签添加自定义属性data-toggle=”pill”  2.事件的目标  Div.tab-pane添加id ex：id=”tab1”  在a标签的href中绑定对应的id  <a href=”#tab1”></a> |

5.导航栏

|  |
| --- |
| div.navbar 水平的弹性  添加. navbar-expand-\* 变成响应式导航栏  .navbar-expand-md-\* 对父级div没有太多影响  >ul. navbar-nav 垂直的弹性  配合父级. navbar-expand-\*  在某个屏幕尺寸下把主轴变成水平，没有到达这个屏幕尺寸，主轴默认是垂直，形成响应式导航栏  >li. nav-item  a.nav-link |

作业：

1栅格完成学子1L，不用定位



2.boot组件的复习，boot使用事件的方式

①按钮组

②信息提示框

③下拉菜单

④导航（水平导航，选项卡导航，胶囊导航）

⑤响应式导航栏

day8.20

三.组件

6.折叠

|  |
| --- |
| Button data-toggle=”collapse” data-target=”#d1”  div#d1.collapse 内容部分，隐藏 |

7.卡片

|  |
| --- |
| 一个区域，内部内容是垂直排列，都可以使用卡片  div.card  >div.card-header  +div.card-body  + div.card-footer  a.card-link  注意事项  1..collapse 不能和.card-body在同一个元素  不然折叠的时候会卡顿，我们需要分层完成  2.几个折叠部分，可以同时打开，而不能开启一个时候，其他自动隐藏  解决方案，在3个card的外部，添加div.parent  在所有的div.collpase 上添加事件  Data-parent=”#parent”  这样就可以保证，在div#parent中只有一个打开状态 |

语义 基本语言 HTML css js java php 🡪 类比：中文 英文 阿拉伯文 （注重语义）

bootstrap 第三方 --🡪 类比：河南话 广东话 北京土话（不注重语义）

8.手风琴

|  |
| --- |
| 捕获 |

9.折叠导航栏

|  |
| --- |
| 结构  div.navbar .navbar-expand-\* .navbar-dark .bg-dark  >a.nav.bar-brand 不折叠的菜单首项  +button.navbar-toggler \*以上屏幕不显示，以下显示  >span.navbar-toggler-icon 按钮三条线  +div.collapse .navbar-collapse 折叠的内容  >ul.navbar-nav>li.nav-item>a.nav.link  ①.navbar-expand-\*  +. navbar-toggler  让按钮在\*以上的屏幕，不显示，在\*以下的屏幕显示  +. navbar-nav  让ul中的里在\*以上的屏幕，横向显示  在\*以下的屏幕，垂直显示  ②.navbar-dark 对当前div没有任何样式  告诉子元素，这个导航栏是深色背景  .nav-dark .nav-brand  .nav-dark .navbar-toggler  .nav-dark .navbar-nav  这些文字颜色是浅色  ③.navbar-collapse 让折叠导航，打开的位置为导航栏的下方 |

flex-basis: 100%; 当设置100%，会换行独占一行

flex-basis 属性用于设置或检索弹性盒伸缩基准值。

**注意：**如果元素不是弹性盒对象的元素，则 flex-basis 属性不起作用。

10.媒体对象

|  |
| --- |
| div.media>img+div.media-body |

11.轮播图

|  |
| --- |
| 1.轮播的基本结构  div.carousel data-ride=”carousel”  >div.carousel-inner  >div.carousel-item.active 其中一个元素添加.active（不写active所有图片不显示）  >img 坑 --🡪w-100 |
| 2.左右箭头  div.carousel #demo  >a.carousel-control-prev  >span.carousel-control-prev-icon  + a.carousel-control-next  >span.carousel-control-next-icon  事件  a data-slide=”prev/next” href=”#demo” |
| 3.轮播指示器  div.carousel#demo  >ul.carousel-indicators 对后代li添加样式 宽高，背景，位置  >li data-slide-to=”0/1/2/3” data-target=”#demo” |

12.模态框

|  |
| --- |
| Modal是覆盖在父窗体上的一个子窗口  <h3>模态框</h3>  <button>打开模态框</button>  <div class="modal">  <div class="modal-dialog">  <div class="modal-content">  <div class="modal-header"></div>  <div class="modal-body"></div>  <div class="modal-footer"></div>  </div>  </div>  </div> |

四.其他组件

1.巨幕

|  |
| --- |
| div.jumbotron 自带媒体查询 |

2.徽章（小按钮）

|  |
| --- |
| .badge 基本类  .badge-danger/.. 颜色  .badge-pill 样式（小胶囊） |

3.分页

|  |
| --- |
| ul.pagination  >li.page-item .disabled/.active  >a.page-link |

4.面包屑导航

|  |
| --- |
| ul.breadcrumb>li.breakcrumb-item>a（这里a标签没有link）  需要重写样式，修改连接字符  .breakcrumb-item+.breakcrumb-item::after{  Content:”>”;  } |

5.进度条

|  |
| --- |
| div.progress 进度条的槽  >div.progress-bar 进度条 宽度设置进度，背景色设置进度条颜色  . progress-bar-striped 条纹进度条  . progress-bar-animated 带动画的进度条 |

**bootstrap重点：响应+栅格布局+scss**

**scss(动态的css语言)**

一.scss理论

|  |
| --- |
| scss 在服务器运行  需要转换成.css文件，浏览器才能执行 |

1.css的缺点

|  |
| --- |
| 语法不够强大，没有变量和合理的代码重用机制  导致难以维护  我们需要使用动态的样式语言，赋予css新的特性  常见的动态的css语言  1.scss/sass(scss兼容sass，更接近css基础语法)  2.stylus  3.less |

2.scss是什么

|  |
| --- |
| scss是一款强化css的辅助工具  scss与css有高度相似  添加变量，嵌套，混合，导入，函数等高级功能  这些高级功能让css更加的强大和优雅  浏览器无法解析运行scss文件，需要在服务器吧scss文件转换成css |

二scss的使用

1.转换工具安装

|  |
| --- |
| scss在服务器使用  1.需要有nodejs解释器，必须v8.11以上  64位的scss需要nodejsv. 10.1以上  2.在线安装 npm install –g node-sass;  3.离线安装  scss包，解压后4个文件  找到nodeJS安装路径，把上面4个文件复制进去  cmd输入node-sass -v,弹出版本号，安装成功 |

2.scss文件转换成css文件

①单文件转换

|  |
| --- |
| 在scss文件夹的父文件夹内打开命令行，输入：node-sass scss/1.scss css/1.css 回车  转换前scss文佳夹/文件 转换后css文佳夹/文件 |

②多文件转换

|  |
| --- |
| node-sass（转换前文件夹） -o（转换后文件夹）  ex: node-sass scss -o css |

③单文件监听

|  |
| --- |
| node-sass -w scss/1.scss css/1.css （比单文件转换多一个”-w”） |

④多文件监听

|  |
| --- |
| node-sass -w scss -o css （比多文件转换多一个”-w”） |

作业：

1.手风琴

2.折叠导航栏

3.轮播图

day8.21

二.scss的使用

3.scss基础语法

①变量

|  |
| --- |
| 使用$标识变量，变量名可以包括字母，\_，数字， 不能以数字开头  基本与css选择器相同，尽量做到见名知意  1.声明变量时，变量值可以引用其他变量  2.变量声明在{}外，整个scss文件都可以访问这个变量  声明在{}内，只有这个{}内部才能访问这个变量  3.!default规则  如果此变量在之前已经声明赋值了，那么使用之前的值  如果此变量之前没有声明赋值，使用现在的值 |

②嵌套

|  |
| --- |
| 1.HTML结构的嵌套  #content{  font-size:$normal-width;  .top{border:$border-size $border-style $my-color1;  h1{font-style:$font-style; }  p{color:$my-color1;}  }  .bottom{border:$border-size $border-style $my-color2;  p{color:$my-color2;  a{text-decoration:$text-decora; }  }  }  } |
| 2.伪类的嵌套  需要在:之前，添加占位符&  如果不添加，会产生一个空格，变成后代选择器  a{  display: block;  width: 200px;height: 200px;  &:hover{  width: 300px;height: 300px;  }  }  3.群组的嵌套  .nav,.footer,.header{  a{background: #c82333;}  }  4.属性的嵌套（使用较少）  div{  border: {width: 10px;style: solid;color: #007bff};  } |

③导入

|  |
| --- |
| Scss语法中，一个scss文件，如果用\_开头，说明这个scss是一个局部文件  局部scss的特点，不会被转换成对应的css文件  一般在全局scss导入局部scss。  @import “局部名称”; 不带”\_”和后缀(.scss) |

④混合器

|  |
| --- |
| 把多个选择器都会使用的样式，封装进一个选择器  需要使用的时候，直接调用，实现代码的重用  混合器可以带参数  (1)声明混合器  @mixin 混合器名称（参数1，参数2…）{样式}  (2)调用混合器  选择器{  @include 混合器名称(参数1，参数2….);  }  混合器使用最多的场合是css hack  @mixin my\_bg($fx,$color1,$color2){  background: -webkit-linear-gradient($fx,$color1,$color2);  background: -ms-linear-gradient($fx,$color1,$color2);  background: -o-linear-gradient($fx,$color1,$color2);  background: -moz-linear-gradient($fx,$color1,$color2);  } |

⑤继承

|  |
| --- |
| 一个选择器，可以传承另一个选择器的所有样式  使用@extend 选择器名称, 继承这个选择器的样式  选择器{@extend 选择器名称;}  Css的代码解析为群组选择器（假继承，可以继承样式，不属于继承） |

4.运算 + - /

|  |
| --- |
| 运算的时候，scss会自动的转换单位，进行运算  但是前提是几个单位可以转换 in pt cm px,  % em rem 这些相对单位无法转换 |

①加法 +

|  |
| --- |
| 字符串拼接的时候  如果用有引号的字符串去拼接无引号的字符串，结果是带引号  如果用无引号的字符串去拼接有引号的字符串，结果是无引号 |

②减法

|  |
| --- |
| 由于变量名称中可以使用-  当减法前后有变量的时候，系统会分不清楚减号是变量的一部分还是单纯的做减法  所以减号前后要加空格，进行区分  width:$my-w - $my-h; |

③除法

|  |
| --- |
| 在scss中， / 号作用，是除法和分割符  在以上情况 /会被解析成除法  1.如果计算式两边，有变量或者方法的返回值，视为除法  width: $width/2;  2.被（）包裹的计算式，视为除法  height:(500px/2);  3.除法运算式，是其他算术运算式的一部分  margin-left: 5px+8px/2px; |

④scss中的插值操作

|  |
| --- |
| content:"liangliang ate #{50+50}baozis";  #{插值操作} |

5.颜色的运算

|  |
| --- |
| #rrggbb和rgb(r,g,b)都是分段运算  红+红 绿+绿 蓝+蓝  rgb(10,10,10,)+rgb(10,10,10,)=rgb(20,20,20)  rgba的运算，要求参与运算的颜色的透明度必须相同 |

三.函数

1.可以直接在css语法中使用的函数

|  |
| --- |
| rgba(reb,green,blue,alpha)  hsl(hue,saturation,lightness)  hue:色调 0~360 ( 0~120 121~240 241~360)  saturation:饱和度 0~100%  lightness:亮度 0~100% |

2.数学函数

|  |
| --- |
| round($v) 四舍五入  ceil($v) 向上取整  floor($v) 向下取整  min($v1,$v2....)  max($v1,$v2....)  random() 随机数 |

3.字符串函数

|  |
| --- |
| unquote($str) 去掉$str的双引号  quote($str) 给$str加上双引号  to\_lower\_case($str) 变小写  to\_upper\_case($str) 变大写 |

4.自定义函数

|  |
| --- |
| @function add($a,$b){  @return $a+$b;  }  div{  width:add(100px,200px);  } |

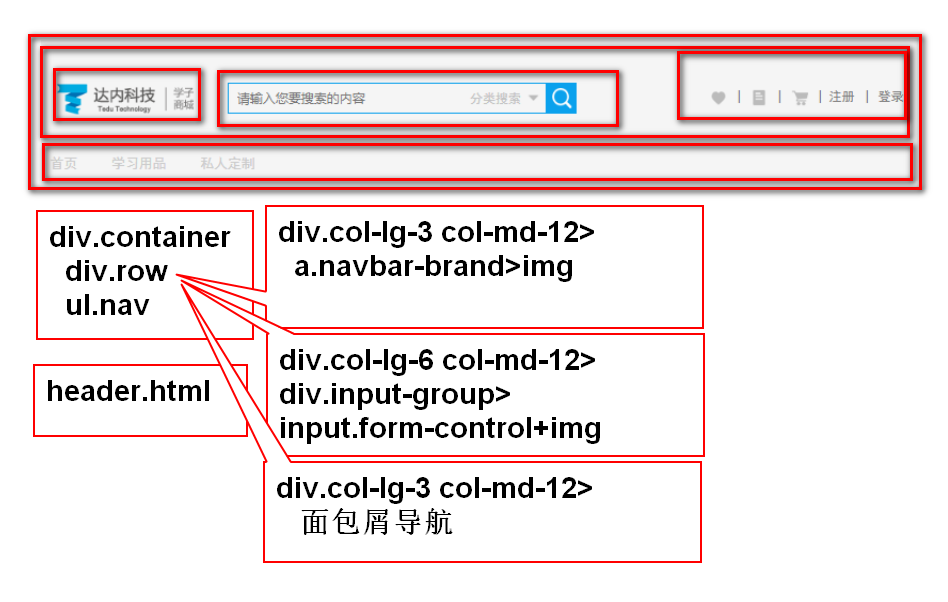
四.指令

|  |
| --- |
| @if（）{ }  @else if（）==moon{ }  @else{ }  注意，指令中条件部分，小括号可以去掉 |

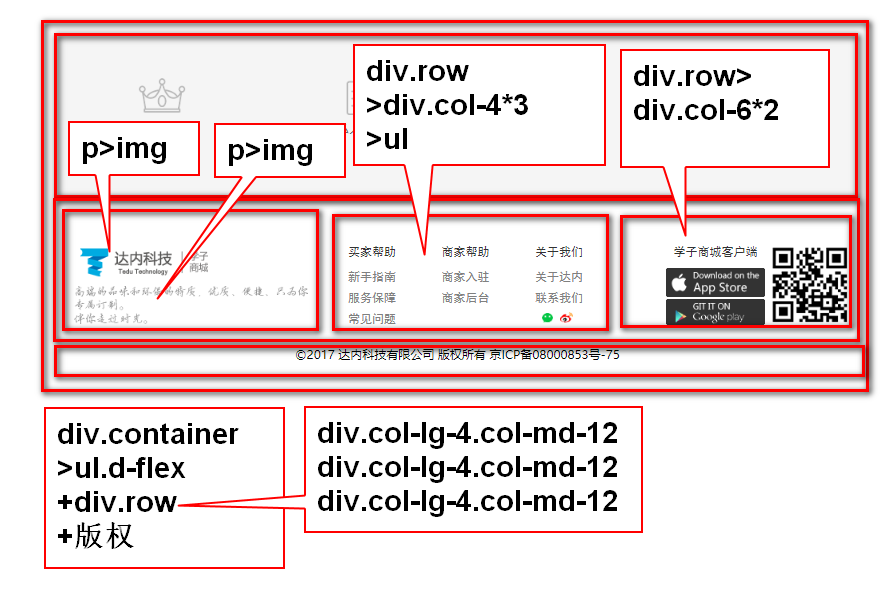
**使用boot完成学子商城项目，要求响应式**

1.使用scss编写comm.css----css reset

2.写header.html



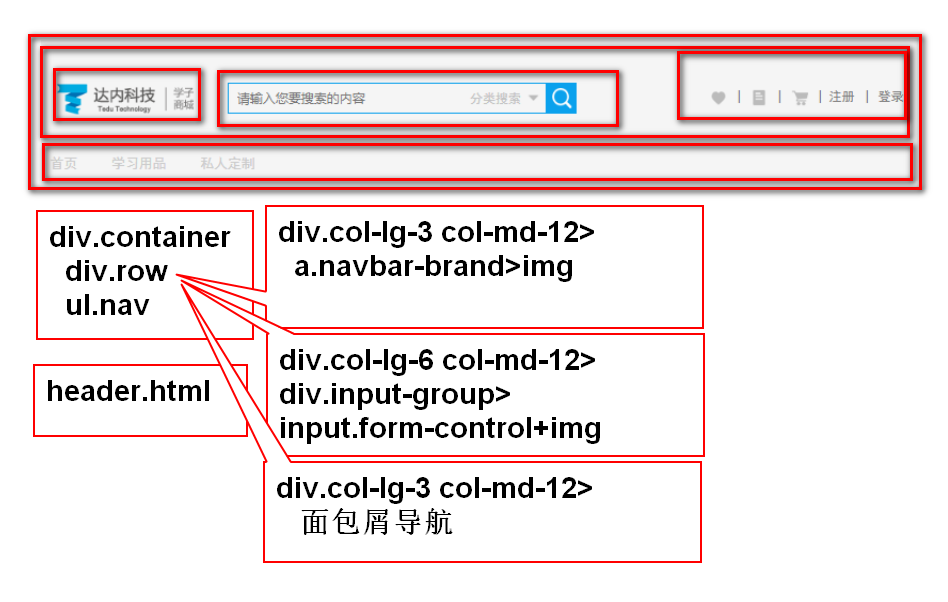
3.写footer.html



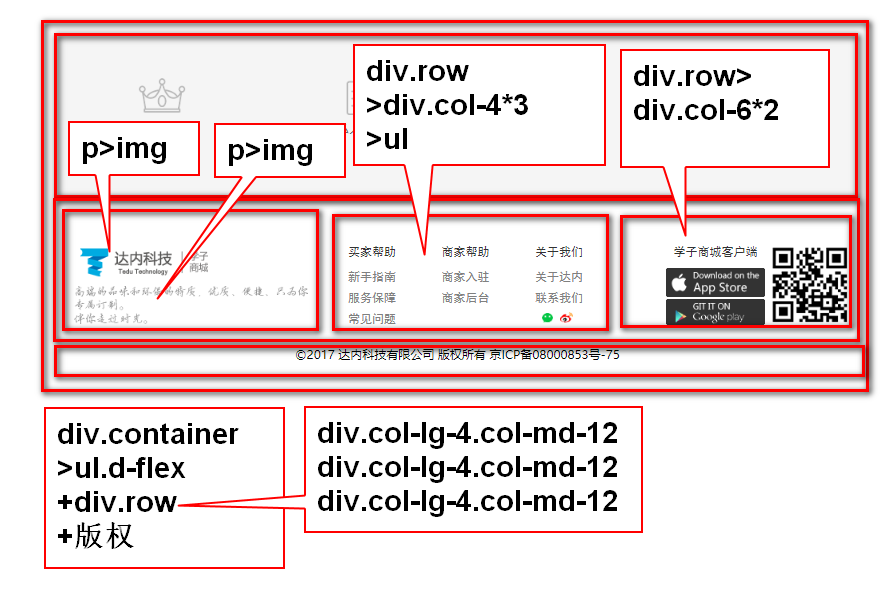
4.在主页使用iframe导入index.html

作业

1.写header.html



2.写footer.html



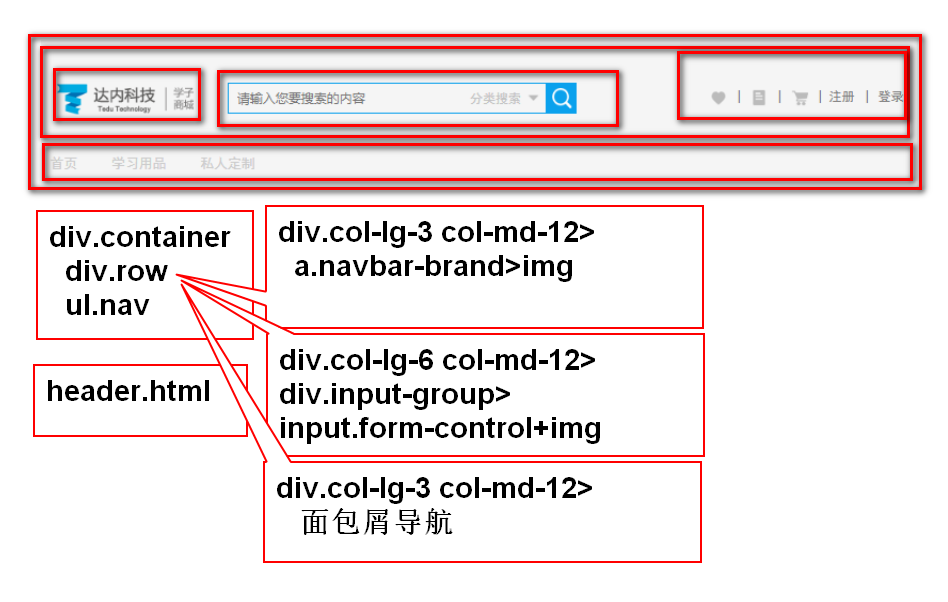
3.在主页使用iframe导入index.html(提高) 宽高 高度有大坑

day8.22

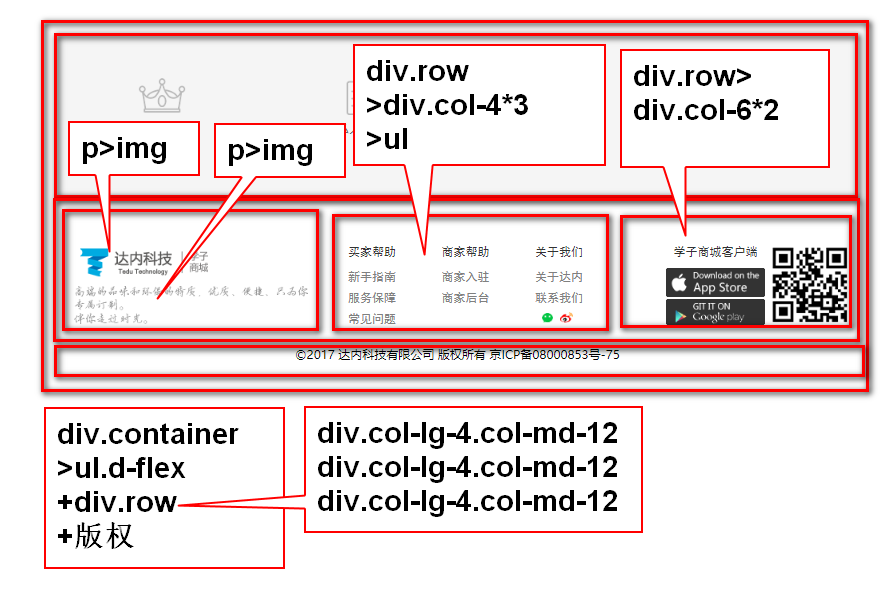
**使用boot完成学子商城项目，要求响应式**

1.使用scss编写comm.css----css reset

2.写header.html



3.写footer.html



4.在首页使用iframe导入index.html

①使用iframe导入header.html和footer.html

|  |
| --- |
| *手写媒体查询编写iframe的高度* |

作业：

1.准备新浪微博的账号

2.复习git

3.完成学子首页

4.学子商品列表页，左侧

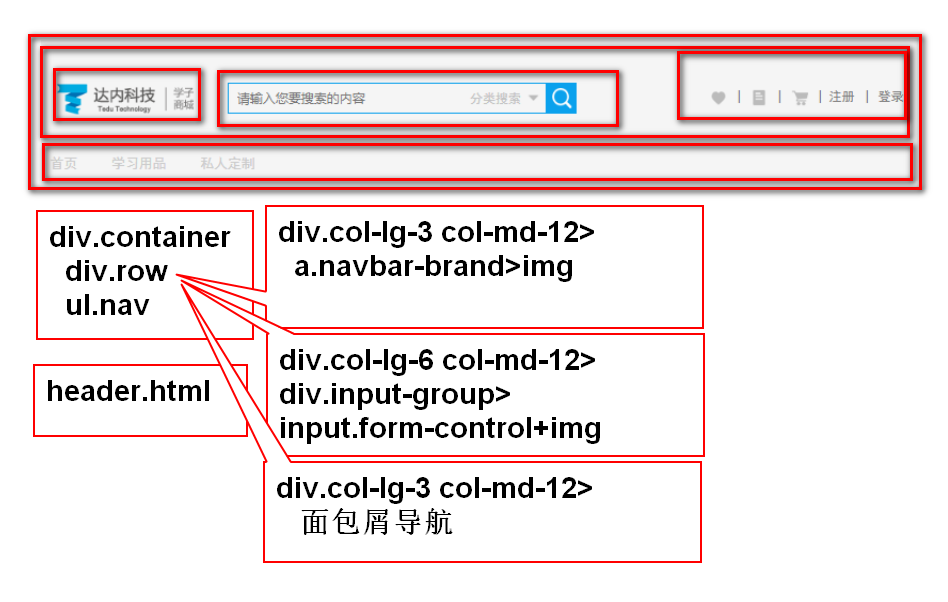
5.复习一下ajax项目

day8.23

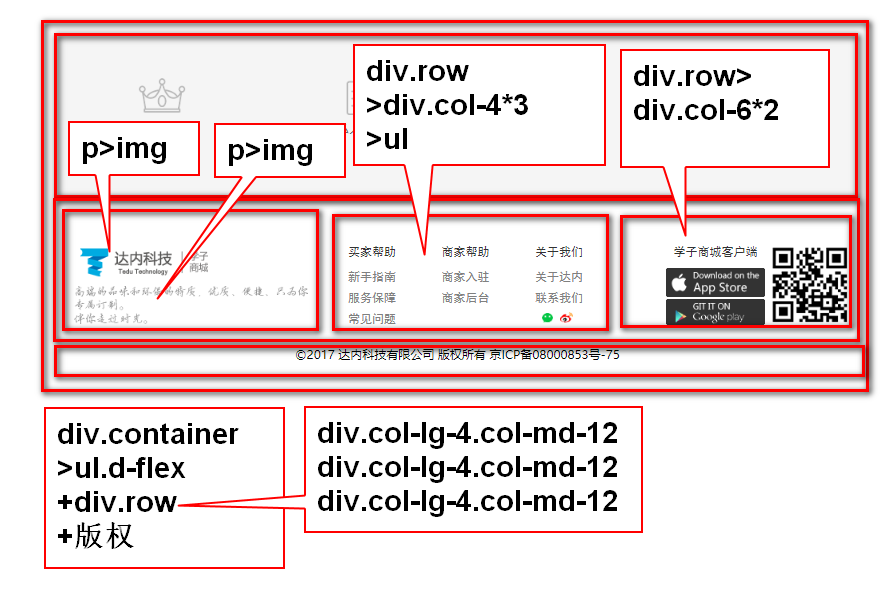
**使用boot完成学子商城项目，要求响应式**

1.使用scss编写comm.css----css reset

2.写header.html



3.写footer.html



4.在首页使用iframe导入index.html

①使用iframe导入header.html和footer.html

|  |
| --- |
| *手写媒体查询编写iframe的高度* |

5.完成首页轮播

6.完成楼层代码

二.完成商品列表页

1.header和footer导入

2.banner图片

3.主体内容左侧+分页

三.完成商品详情页

四.新浪云的使用

1.标准的nodejs代码

<https://www.cnblogs.com/zhangym118/p/8874239.html>

2.注册新浪云服务器---绝大多数服务免费---赠送三级域名

3.创建新应用

①php免费

|  |
| --- |
| 代码上传：  在线编辑代码，运行环境管理-->代码管理-->在线编辑 |

②nodejs 3分/小时

|  |
| --- |
| 代码管理只能使用git  1.如何创建一个标准结构的nodejs项目  2.如何使用git把项目部署到新浪云  ①运行环境管理-->代码管理  ②如何部署  a-创建本地文件夹作为git仓库  b-进入这个文件夹，打开powershell  c-在你应用代码目录里，克隆git远程仓库  git clone https://git.sinacloud.com/rangehaoshuai  #第一次操作的时候，输入您的安全邮箱和密码。  用户名，密码输入错误    出错的解决方案，控制面板--用户--修改windows凭据  把新浪云相关的删除  d- 把你要部署的代码，复制进rangehaoshuai文件夹  e-cd rangehaoshuai  f-添加本地的文件改动  git add .  g-添加本地修改的备注信息  git commit -m 'Init my first app'  i-将改动推送到远程仓库  git push origin master |

③在新云中，部署数据库操作

|  |
| --- |
| 1.生成sql脚本 \*.sql  2.在新浪云共享数据库中导入\*.sql，这样就生成云数据库  3.注意，云数据的用户名，密码，主机名，库名，都发生了改变  连接池的配置一定要修改  4.一个应用中数据库，只能被本应用的代码访问，外部访问不了 |

作业：

1.完成学子商品详情页的编写

2.新浪云使用php应用，把学子boot系项部署上

周末在家完成http1在新浪云上的部署，正确可以登录成功(提高题)

栅格，媒体查询，scss，css基础，html基础，ajax