day01

一.课程安排：

1.HTML5 basic 2天(搭建网页结构，可以显示数据)

2.Ajax异步数据交互3天(页面和服务器的交互，为了得到数据在页面中的显示)

3.Ajax项目2天

4.CSS3 4+2天(负责给页面化妆，让页面变得好看)

5.BootStrap框架 4天

6.Boot项目 3天

二.web基础知识

html5：大前端技术，泛指前端的所有技术

internet：全球性计算机互联网

BBS：电子公告板，俗称论坛

Telnet：远程登录

ftp：文件的上传下载

2.internet上的应用程序

C/S dient客户端/server服务器

B/S browser浏览器/server服务器

C/S和B/S的区别

C/S需要升级B/S不需要

我们主要做B/S

3.web：运行在Internet上一种B/S结构的应用，俗称网站

web的工作原理基于浏览器和服务器继续通信协议来实现数据的传输和展示

通信协议：规范了数据和打包和传输的

服务器

1.功能： 存储数据，接受用户的请求并给出响应

提供了程序的运行环境

具备一定的安全功能

2.服务器产品

Tomcat(java)

Apache(php)

IIS(.net)

3.服务器端的技术

PHP/JAVA/.NET/python/nodejs

浏览器

1.功能：

代表用户发送请求

作为HTML、CSS、js解析器以图形话的解码展示给用户

2.浏览器产品

Chrome

Firefox

IE

Oprea

Safari

3.浏览器支持的技术

H5,CSS,JS

三.HTML快速入门

1.什么是HTML

HTML:Hyper Text Markup Language超文本标记语言

a 普通文本

<a>a</a>超文本标记

标记：超文本的组成形式<关键字>

具有自己的功能

语言：有语法结构

2.HTML的特点

1.以.html或htm为后缀

2.有浏览器解析执行

3.用带有尖括号的"标记"来标识

4.可以在网页中执行JS脚本

3.HTML的基础语法

标记，标签，元素，节点，对象

主要是表示特殊的功能

标记在使用的时候，必须使用<>包裹起来

1.标签的分类

1.双标记(封闭类型)

<关键字></关键字>

双标记必须成对出现，有开始就有结果 <p></p>

2.单标记(非封闭类型)

<关键字>或者<关键字/>

2.标签的嵌套

1.什么是嵌套

一对标签出现另一个(对)标签，形成了层级

<div>

<p></p>

</div>

2.语法

<关键字1>

<关键字2>、

<关键字3></关键字3>

</关键字2>

</关键字1>

3.属性和值

允许通过属性和值对标记进行修饰

语法：

1.属性和值必须声明在开始标记中

<关键字 属性="值"></关键字>

2.多个属性之间用空格分开

<关键字 属性1="值1" 属性2="值2" ...></关键字>

3.标准属性(通用属性)

所有的元素都支持的属性

id：定义此元素在页面中的唯一标识

title：鼠标悬停在元素上时，显示的文本

style css中，定义行内演示

class css中，应用类型的选择器

学习HTML，就是学习固定关键字，学习此关键字作用，学习固定的标签属性，学习嵌套关系

开发小技巧

1.标签只写完整的第一个，按tab不全

2.属性如果拼写正确，会变色

3.属性值可以不添加""和''，但是不推荐裸写值

4.每次修改要记得保存 ctrl+s

5.页面刷新 F5

6.添加eclipse快捷方式

tools-首选项-keyboard-最右边load加载1.key

退回 ctrl+z

前进 ctrl+y

删除当前行 ctrl+d

整行赋值 ctrl+alt+l

整行位移 alt+l

其他快捷方式

windows + d

alt+F4

4.注释

语法：<!--注释内容-->

注意：注释不能嵌套注释；注释不能写在标签中

四.HTML的文档结构

1.HTML文档的构成

1.文档类型的声明

<!DOCTYPE html>这句话告诉浏览器使用H5规则解析

2.网页的结构

<!DOCTYPE html>

<html>表示网页的开头和结束

<head></head>网页的头部，定义全局的信息

<body></body>网页的主体，展示的内容

</html>

3.head元素是其他头元素的容器

<title></title>网页的标题

<meta/>元数据，定义网页的全部信息

<meta charset="utf-8" />

Chrome 62以上的版本，自动识别中文

<meta name="Keywords" content="关键词的内容">

<meta name="discription" content="描述的内容">

<style></style>定义网页的内部样式

<script></script>定义或者引用js文件

<link/>应用外部样式文件

4.body元素

定义网页的主体

body的属性

bgcolor 背景颜色

text 字体颜色

取值：合法的颜色值--->颜色的英文单词

五.文本标记

1.标题元素

作用：在文本中以醒目的方式显示文本

语法：<h1></h1>

...

<h6></h6>

特点：

1.文本加粗

2.字号发生变化

3.独占一行，上下有垂直间距

属性：align设置标签内容水平方向对其方式

取值：left/center/right

2.段落标记paragraph

语法：<p></p>

作用：突出的表示一段文字

特点：独立成行，上下的垂直的间距

属性：align

3.换行标记

<br>或者<br/>

4.水平线(分割线)

<hr>或者<hr/>

width="50%"表示水平线的宽度，取值为%或者px为单位的数字

size="5px"表示水平线的尺寸(粗细)，取值为%或者px为单位的数字

align="left"对其方式

color="red"设置水平线的颜色

5.预格式话标签

<pre><pre>

标签内部的文本，保留其格式，在页面上显示

空格折叠现象：一个或多个空格，会被浏览器渲染成一个空格显示

一个或多个回车，会被浏览器渲染成一个空格显示

6.特殊字符(实体)

空格 &nbsp;

< &lt;

> &gt;

© &copy;

® &reg;

¥ &yen;

× &times;

7.文本样式

粗体<b></b> <strong></strong>

斜体<i></i> <em></em>语意

下划线<u></u>

删除线<s></s> <del></del>

上标2<sup>3</sup>

下标2<sub>3</sub>

8.分区元素

1.块分区，用于页面中布局

<div></div>

独立成行

2.行分区，处理同一行文本的不同样式

<span></span>

可以与其他的span和文字共用一行

9.行级元素和块级元素(元素的显示形式)

1.块级元素

在网页中，单独成行的元素，就是块级元素

从上往下排列

常见的块级元素

h1~h6

p

div

2.行内元素(行级元素)

在页面中，与其他行内元素/行内块共用一行

从左往右排列

常见的行内元素

i

em

b

strong

u

sup

sub

span

3.行内块元素

显示的方式与行内元素相同，但是具备块级元素的特征

4.table

表格的快高是有表格的内容决定的

内容多，就宽一些，高一些

表格中所有的数据都于都到内存

六.图像和链接

1.图像

1.使用

<img>或者<img/>

属性

src="url" url-图片资源路径

2.URL

Uniform Resource Locator 统一资源定位符 简称：路径

3.URL路径的表现形式

1.绝对路径：一个写完整的路径

通信协议+主机地址+文件目录结构+文件的具体名称

合适使用：

1.使用网络资源的时候，使用绝对路径

优点：不占据自己服务器的存储空间

缺点：不稳定

2.使用本地资源的时候，使用绝对路径(项目中100%用不到)

从最高的盘符开始查找

项目中不用

2.相对路径(参照物是html本身)

1.图片与html文件在同一目录下，src直接引用图片名称

day02

一.图像与链接

1.路径

1.绝对路径

完整的路径

使用网络图片资源

通信协议+主机地址+文件目录结构+文件的具体名称

2.本机资源的绝对路径

从最高盘符开始查找

C:\xxx\xxx\1.jpg

项目中不用

2.相对路径---参照物，本html文件

2.1在同一目录下(兄弟关系)，直接引用兄弟文件

<img src="08.png" alt="">

2.2子集目录，文件在兄弟文件夹的内部，先找兄弟，再找兄弟的孩子

<img src="./image/09.png" alt="">

2.3父级目录，文件时父级的兄弟

<img src="../image/02.png" alt="">

3.属性

src：source源，设置资源路径

alt：图片资源出错，显示的文字

width：设置图片的宽度，以px为单位的数字

height：这只图片的高度，以px为单位的数字

由于宽高比设置的不对，会发生图片失真的情况

解决：宽高只设置一个，另外一个自适应

2.链接

1.语法：<a href="#"></a>

<a href="http://www.tmooc.cn/">tmooc</a>

属性：

href：链接的路径

target：指定打开链接的方式

取值：\_self 默认值，在当前页面标签页中打开新链接

\_blank 在新的标签页打开链接

2.其他的表现形式

<!--a标签的其他表现形式-->

<!--下载-->

<a href="1.zip">下载</a>

<!--新建邮件，前提：本机必须有邮件软件的客户端-->

<a href="mailto:514948303@qq.com">新建邮件</a>

<!--执行js脚本-->

<a href="javascript:alert(111)">执行JS</a>

<!--返回当前页面的顶部-->

<a href="#">返回顶部</a>

3.锚点

锚点就是网页中的一个记号

3.1定义锚点

h4：使用a标签的name属性定义锚点

<a name="锚点名称"></a>

h5：任意标签的id属性定义锚点

<any id="id"></any>

3.2链接到锚点

(1)使用a标签链接锚点，#锚点名称

<a href="#锚点名称">回到锚点</a>

二.表格

1.表格语法

<table>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

......

</table>

写一个4x4的表格：table>tr\*4>td\*4

2.属性

1.table标签的属性

border：设置边框

width：设置宽度

height：设置高度

内容小于宽高的，按宽高尺寸；内容大于宽高，按内容改变大小

align：设置表格排列方式(table自己的水平位置)

取值：left、right、center

bgcolor：背景颜色

cellpadding：设置单元格内间距(边框到内容的距离)

cellspacing：设置单元格外间距(边框与边框的距离)

2.tr的属性

1.align：设置当前行内容的水平对齐方式(设置表格内容的对齐方式)

2.valign：设置当前行内容的锤子对齐方式

取值：top/middle/bottom

3.bgcolor：当前行的背景颜色

3.td的属性

width

height

align

valign

bgcolor

colspan

rowspan

3.不规则表格的使用

1.跨列

colspan=?

从指定的单元格位置处开始，横向向右合并?个单元格(包含自己)，被合并的单元格，要删除

2.跨行

rowspan=?

从指定的单元格位置处开始，纵向向下合并?个单元格(包含自己)，被合并的单元格，要删除

4.table中的可选标记

1.表格的标题

<caption></caption>

如果设置标题，必须将<caption>放在<table>之后

2.行/列的标题

<th></th>代替<td></td>

加粗，居中

5.表格的复杂应用

1.行分组

可以将连续几个行划分到一个组中，进行统一管理

表头行：<thead></thead> 表格最上面一行的标题放在thead中

表主体：<tbody></tbody> 允许将若干行放在tbody中进行统一管理

表尾行：<tfoot></tfoot> 表格中最后一行放在tfoot中

2.表格的嵌套

**一张表格中，所有的嵌套的表格，只能放在td中**

三.列表

默认按照从上到下的顺序来显示数据，并且可以在数据前添加标识

**使用列表布局(去掉标识，从上到下，从左到右排列)**

1.列表的组成

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

2.列表属性

1.有序列表的属性ol

type设置标识符的类型

取值：1默认值，数字

a A 字母

i I 罗马数字

start指定其实编号

取值：无单位数字

2.无序列表的属性ul

type设置标识

取值：disc 默认值，实心圆

circle 空心圆

square 实心方块

none 无

3.列表的嵌套

所有的嵌套必须只能写在li中

day03

一.列表

1定义列表

1.作用

做名词解释使用，常用于图文混排

<dl>

<dt>需要被解释说明的名字</dt>

<dd>对名词的解释内容</dd>

</dl>

二.结构标记

1.结构标签的作用

代替div，用于描述整个网页的结构，提升标记的语义

2.常用的结构标记

<header></header>

定义网页或者某个区域的头部内容

<nav></nav>

定义网页的导航链接部分

<section></section>

定义网页的主体内容

<aside></aside>

定义网页的侧边栏

<footer></footer>

定义网页底部内容

<article></article>

定义与文字相关的内容

三.表单

1.作用

1.提供可视化的输入控件

2.收集用户填写的信息并提交给服务器

2.表单的组成部分

1.前端部分

提供表单的控件，与用户做交互的可视化控件

2.后端部分

对提交的数据进行处理(接口)

3.表单

<form></form>

属性

1.action

定义表单被提交时发送的动作

通常是定义服务器处理数据的地址、接口(url)

默认，提交给本页地址

2.method

指定表单数据提交的方式(方法)

1.get(默认值)

明文提交，提交时内容会显示在浏览器地址栏上

提交的数据有大小限制，2kb

向服务器要数据的时候使用get

2.post

提交数据无大小限制

隐式提交，提交的数据不会显示在地址栏中

把数据传递给服务器的时候，使用post

3.delete

4.put

3.enctype编码方式

设置表单允许将什么样的数据提交给服务器

1.提交普通文字，不能有特殊符号

enctype="text/plain"

2.允许将任意字符提交给服务器

enctype="application/x-www-form-urlencoded"

3.允许将文件提交给服务器

enctype="multipart/form-data"

4.表单控件

能够与用户进行交互的可视化元素

4.1分类

1.input元素

2.textarea多行文本域

3.下拉选择框

4.其他元素

4.2 input元素

在页面中提供各种各样的输入控件，通过type属性来设置输入控件的类型

<input>或者<input/>

属性

type设置输入控件的类型

name为控件定义名称，提供给服务器端使用(必须值)

value控件的值，提交给服务器端使用

disabled禁用控件，不能操作也不能提交，无值属性

type取值

1.文本框和密码框

type="text" 默认值

type="password"

属性

1.maxlength限制输入的最大字符数

2.readonly只读，只能看不能写，但允许被提交，无值属性

3.placeholder占位符，默认显示在控件上的文本，只要得到焦点，文本就无效了，此文本不能提交

2.按钮

type="submit" 提交按钮

将表单中的数据，提交给服务器

type="reset" 重置按钮

让表单内容恢复到初始化状态

type="button" 无功能按钮，可以调用js脚本

3.单选按钮和复选按钮

单选按钮

type="radio"

name属性：除了定义控件名称，还起到分组的效果

必须有value，不然提交的值是on

checked：设置默认选中项，无值属性

多选按钮

type="checkbox"

name属性：除了定义控件名称，还起到分组的效果

必须有value，不然提交的值是on

checked：设置默认选中项，无值属性

checkbox的name要写成数组hobby[]

4.隐藏域

type="hidden"

想提交给服务器，但是不想展示给用户看的数据，使用隐藏域

5.文件选择框

type="file"

注意

method="post"

enctype="multipart/form-data"

multiple 无值属性，可以同时上传多个文件

需要name属性值为数组

4.3textarea文本域

允许录入多行数据的文本框

<textarea></textarea>

name控件名称

readonly只读

cols指定文本域的列数，以英文单词为准，中文减半

rows指定文本域的行数

4.4下拉选择框

<select> 表示一个下拉选

<option></option> 表示一个下拉选中的一个选项

<option></option>

...

<select>

注意：当option没有value属性时，提交的时候select的value为选中option的内容。当option有value，提交时，select的value为选中option的value

select属性

name选项框的名称

size默认为1，定义显示选项的数量。如果值大于1，下拉选表现为滚动列表

multiple无值属性，可以多选，但是name要用数组，设置了multiple之后，就算size="1"也是滚动列表

option属性

value

selected设置默认选中项

5.其他元素

1.label

文本与表单控件的关联

<label></label>

属性

for：要与label中文字发生关联的form控件的id

2.为控件分组

<fieldset></fieldset> 为控件分组

<legend></legend> 分组标题

3.浮动框架

允许在一个网页中，引入另外一个网页

<iframe></iframe>

属性

src="url"

width="100%"

height="" 高度存在问题，需要使用js.dom来解决

scrolling="no" 不显示滚动条

frameborder="0" 不显示边框

6.H5新表单元素

什么是新表单元素

在HTML5版本中新提出来的表单控件

1.电子邮件

<input type="email">

表单提交时，会验证数据是否符合email的规范(@以及@后面有内容)

2.搜索类型

<input type="search">

提供了快速清除文本框的功能

3.url类型

<input type="url">

提交时，验证数据是否符合url的规范，绝对路径的规范

验证http://以及后面的必须有内容

4.电话号码

<input type="tel">

在移动设备中，显示拨号键的效果

5.数字类型

<input type="number" value="10" max="20" min="5" step="2">

只能接受数字，并且提供了箭头灵活的调整数字

属性

value指定数字

min能接收提交的最小值

max能接收提交的最大值

step步数：每次点击箭头，变化的数量

6.范围类型

<input type="range" value="50" min="0" max="100" step="5">

提供一个滑块组件，允许用户指定选取的范围

属性

value指定数字

min能接收提交的最小值

max能接收提交的最大值

step步数

7.颜色拾取器

<input type="color">

提供一个颜色的拾取器

8.日期类型

<input type="date">

提供了一个选择日期的控件，可以选择年月日

9.月份类型

<input type="month">

提供了一个选择月份的控件，可以选择年月

10.周类型

<input type="week">

提供了一个选择周的控件，可以选择年和周

day4

ajax的异步加载

一.HTTP协议

1.url

结构：协议+主机名称+目录结构+文件名称

url完整结构

<scheme>://<user>:<pwd>@<host>:<port>/<path>;<params>?<query>#<frag>

scheme方案、协议，以哪种方式获取浏览器的资源

不区分大小写，创建协议，http/https/ftp

|  |  |  |
| --- | --- | --- |
| 常见的协议 | 默认端口号 | 基本作用 |
| ftp | 21 | 文件的上传下载 |
| SSH | 22 | 安全的远程登录 |
| TELNet | 23 | 远程登录 |
| SMTP | 25 | 邮件传输协议 |
| DNS | 53 | 域名解析 |
| HTTP | 80 | 超文本传输 |
| POP3 | 110 | 邮件接收 |
| HTTPS | 443 | 加密的传输http |

user：用户名

pwd：密码

host：主机名

port：端口号

path：资源在服务器上存放的位置

params：参数，服务器端使用

query：查询字符串 ?uname=lilei&upwd=123456

frag：锚点

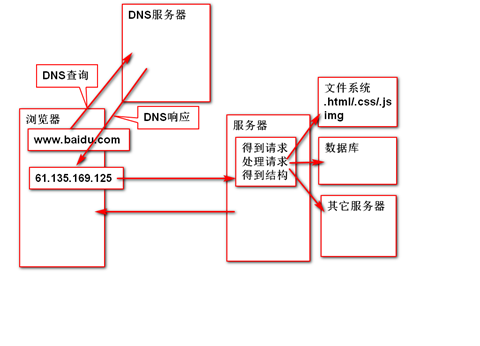
2.HTTP协议

HTTP：Hyper Text Transfer Protocol超文本传输协议

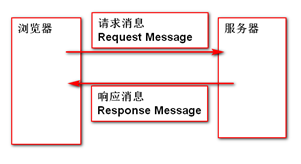
规范了数据是如何打包及传递的

HTTP的历史 现在使用的版本是http/1.1 不需要设置

3.详解



HTTP协议规定，客户端与服务器端传递的数据块Message(消息/报文)。必须符合特定的格式，这个Message才能被客户端和服务器端彼此理解



1.请求消息(请求起始行，请求头，请求主体)

浏览器要发送给服务器的数据

2.响应消息(响应起始行，响应头，响应主体)

服务器发送给浏览器的数据

Request请求消息

客户端要发送给服务器的数据

由三部分组成：请求其实行，请求头，请求主体

1.请求起始行

-请求方法

-请求url

-协议版本

1.请求方法

get 客户端向服务器要数据的时候使用

靠地址栏传递查询字符串，无请求主体

post 客户端向服务器提交数据的时候使用

隐式发送。有请求主体(form data)

delete 表现客户端可以删除服务器上的内容(一般禁用)

put 表示客户端可以往服务器上放数据(一般禁用)

connect 测试连接

trace 追踪请求路径

option 选项

head 表示客户端值回去响应的消息头  
 2.url 访问的地址链接

3.协议版本 http/1.1

2.请求头 Request Headers

host告诉服务器，请求哪一个主机

Connection: keep-alive 告诉服务器进行持久链接

User-Agent告诉服务器自己(浏览器)的类型

Accept-Encoding: gzip, deflate 告诉服务器，自己能接受的压缩文件类型

Accept-Language告诉服务器自己能接受的自然语言类型

Referer引用/推荐人，告诉服务器请求来自哪个网页

3.请求主体Form Data

Response响应消息

服务器端发送给客户端的数据

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http/1.1

响应状态码

原因短句：对状态码的解释说明

响应状态码

告诉客户端浏览器，服务器的响应状态是什么

取值：

1XX：100-199 提示信息

2XX：成功响应 200 ok

3XX：301 永久重定向

302 临时重定向

304 请求未被修改，还是原来的请求，命中缓存

4XX：客户端请求的错误

404 NOT Found请求资源不存在

403 Forbidden权限不够

405 Method not Allowed请求方法不被允许

5XX：服务器运行错误

原因短句，就是对状态码的解释

2.响应头

Date告诉浏览器，服务器的响应时间，格林威治时间

Connection:keep-alive告诉浏览器已经开启了持久链接

Content-Type响应主体的类型

text/html响应回来的数据是html文本

text/plain普通文本

text/css样式文件

application/javascript js文件

image/png(jpeg,gif)图片

application/json响应回来的是json格式的字符串

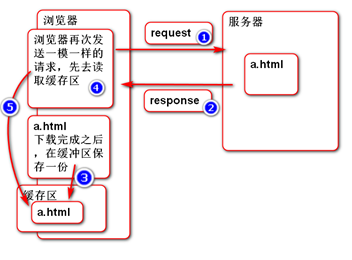
application/xml响应回来的是xml格式的字符串

3.响应主体 响应回来的数据

network中，在header的右边第二个

缓存

1.什么是缓存



2.缓存的优点

1.减少冗余的数据传输，节省客户端的浏览

2.节省服务器带宽

3.降低了对服务资源的消耗和运算需求

4.降低了远距离传输二造成的延迟加载

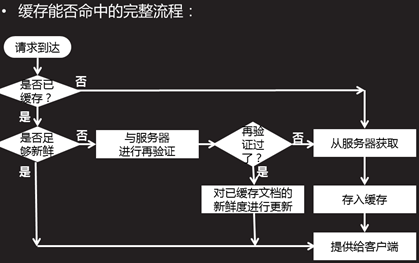
缓存的原理

1.请求--无缓存--访问服务器--存缓存--客户端得到数据

2.请求--有缓存--够新鲜--命中缓存--客户端得到数据

3.请求--有缓存--不新鲜--访问服务器看是否过期--没过期--更新缓存的新鲜度--客户端得到数据

4.请求--有缓存--不新鲜--访问服务器看是否过期--已过期--访问服务器--存缓存--客户端得到数据



3.与缓存相关的消息头

Cache-Control：http1.1的用法

从服务器将文档传到客户端之时起，此文档处于新鲜的秒数，这个秒数是一个相对时间

语法：0 不缓存

max-age=新鲜的秒数

Expires：http1.0用法

指定过期的确切时间，是一个具体的时间点

4.网页如何添加缓存

<meta http-equiv="消息头" content="消息头对应的值">

example:

<meta http-equiv=" Cache-Control " content="max-age=3600">

5.http性能优化

1.http连接的过程

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

2.http连接性能优化的思路

2.1减少请求次数

2.2减少连接创建次数(开启持久连接)

2.3提高服务器端的运行速度

2.4尽可能减少响应数据的长度

3.安全的HTTP协议

HTTPS，安全版本的http

SSL：为数据通信提供了安全支持

1.客户端发送请求消息时，在SSL层加密

服务器接受到加密文件，在SSL层解密

得到请求明文，对请求进行处理

2.服务器发送响应消息时，在SSL层加密

客户端接收到加密文件，在SSL层解密

得到响应明文，解析响应内容

二.DOM操作

1.完整的JavaScript操作的组成

1.js核心：ECMAScript ECMA--欧洲计算机制造联合会

2.DOM：Document Object Model 文档对象模型

让js动态的操作页面上的元素

3.BOM：Browser Object Model 浏览器对象模型

让js动态的操作浏览器

DOM：这个阶段，获取页面上的元素，改变这个元素的内容

2.简单DOM操作

1.要获取的元素一定要有id

<div id="d1">这是测试的文本！！</div>

2.在js中，用id获取对象

document.getElementById('d1')

H5支持简写方式

d1就是div对象

3.获取/设置元素的内容innerHTML

d1.innerHTML=''

var str = d1.innerHTML

innerHTML指的是，双标签，开关标签之间的所有内容。如果开关标签之前有其他元素，会得到该元素的文本字符串，如果设置innerHTML时，添加了新标签，在语法正确的情况下，新标签也会在页面上渲染

4.事件 用户与页面交互行为，调用什么

|  |
| --- |
| ajax错误总结  404：检查url是否正确，要访问的资源是不是在url指向的位置(html位置和路由)  5XX：检查接口中的代码是否正确 |

day05

一.DOM操作

1.获取元素的对象

直接使用这个元素的id值

2.获取/修改元素的内容(只有双标签叫内容)

d1.innerHTML

可以获取/修改内容文本

还可以动态的添加元素

3.获取/设置表单控件的数据

1.使用input控件的id代表控件对象

2.通过对象的value属性，获取/设置控件的值

<input type="text" id="uname">

uname.value=100

4.HTML元素的事件

允许通过用户的行为来激发的操作就是事件

常用的事件 onclick用户点击元素，激发某项操作

文本框/密码框，失去焦点事件 onblur

文本框/密码框，获取焦点事件 onfocus

body的事件onload

当页面加载完成，马上要激发的操作

二.AJAX

同步Synchronous

在一个任务进行过程中，不能开启其他的任务

同步访问：浏览器在想服务器发送请求时，浏览器只能等待服务器的响应，不能做其他事情

出现场合：

1.地址栏输入网址并跳转

2.a标记跳转

3.submit表单的提交

异步Asynchronous

在一个任务开启的同时，可以开启其他任务

异步的访问：浏览器在向服务器发送请求时，不耽误用户在网页上的其他操作

出现的场合：

1.用户名的重复验证

2.聊天室

3.搜索建议(百度，京东，淘宝)

4.股票走势图

1.什么是AJAX

Asynchronous JavaScript And Xml

本质：使用js提供的异步对象(XMLHttpRequest----xhr)

异步的向服务器发送请求

并接受响应回来的数据

ajax中，可以无刷新效果，去更改页面的布局结构和内容

2.使用ajax步骤

//1.创建ajax对象

var xhr = new XMLHttpRequest()

//2.绑定监听事件 监听器---关注响应--->激发一个事件

xhr.onreadystatechange=function(){

if (xhr.readyState == 4 && xhr.status == 200) {

//接受响应

var result = xhr.responseText

}

}

//3.打开链接，创建请求

xhr.open('get','url',true)

//4.发送请求

xhr.send()

url='127.0.0.1:3000/list?uname=...'

3.创建异步对象

标准创建 var xhr = new XMLHttpRequest()

IE8以下的创建 var xhr = new ActiveXObject("Microsoft.XMLHttp")

如何判断用户的浏览器版本？

使用window.XMLHttpRequest属性

如果是高版本，这个属性有值

如果浏览器不支持标准创建，这个属性为null

4.异步对象的常用属性和方法

readystate表示xhr对象的请求状态

取值0~4表示着5个状态

0：请求尚未初始化

1：打开服务器链接，正在发送请求

2：接受响应头

3：接受响应主体

4：接受响应数据成功

当readystate的值为4的时候，才表示所有的数据接收完毕

onreadystatechange当xhr的readystate属性发生改变时触发

status表示服务器的响应状态码

只有在status=200，表示服务器已经正确处理了请求

5.使用get发送带参数的请求

url='demo/login?uname=value1&upwd=value2'

day06

一.ajax

1.post

1.post请求的数据放在请求主体中

var formdata = 'uname='+uname.value+'&upwd='+upwd.value

xhr.send(formdata)

2.发送请求之前，要手动修改请求消息头

xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')

默认的消息头Content-Type:text/plain，传递的数据是普通的，无特殊符号的字符，主体中有特殊符号'&'，必须要把值设置成可以有特殊符号的字符类型application/x-www-form-urlencoded

注意：设置请求消息头，放在xhr.open和xhr.send之间

二.json数据格式

1.对象的数据格式

var computer = {

name:macbook\_air,

price:7000,

left:200

}

2.json的数据格式

1.什么是json

JavaScript Object Notation

js对象方式法，以js对象的数据格式表现出来的字符串

2.json的语法

1.json中用一对{}来表示一个对象

2.对象中的属性名称必须用""引起来(不要用单引号)

如果值是字符串，也必须用""引起来

3.json表现出来是一个字符串，所以最外层添加单引号''

|  |  |
| --- | --- |
| js | json |
| var computer = {  name:macbook\_air,  price:7000,  left:200  } | var computer = '{  "name":macbook\_air,  "price":7000,  "left":200  }' |
| ["tom","lilei","hanmeimei"] | '[  {"name":"tom","age":18}  {"name":"lilei","age":20}  {"name":"hanmeimei","age":19}  ]' |

json是一个以js对象为格式的字符串，不能以 对象.属性 的方式获取值，需要先把json字符串转化成js对象，才能使用。

使用JSON.parse(json串)将json字符串解析为js对象/js数组

可以在前端页面中，使用js语法操作对象/数组，在使用dom把内容动态的加载到html中

三.xml数据格式

什么是XML

eXtensible Markup Language

xml的标记是没有被预定义过得，需要自行定义

xml的宗旨，是做数据的传递，而不是数据展示

语法规范

1.xml的最顶端做声明

<?xml version = "1.0" encoding = "utf-8"?>

2.xml标记的语法

必须成对出现

xml中，标签严格区分大小写

xml允许嵌套，但注意嵌套顺序

每个标签允许自定义属性，格式与html一致，但是值必须带""

每个xml文档，有且只有一个根标签

ajax获取xml数据

使用var result = xhr.responseXML获取xml的对象

var result = xhr.responseXML

var students = result.getElementsByTagName('student')

var sname = students[0].getElementsByTagName('name')

sname[0].innerHTML

这种dom操作获取的数组，叫**类数组**

**类数组可以使用下标查找元素，但是不能使用数组的API**

四.项目阶段

1.登录