目录

[3:h5新特性--增强型的表单 1](#_Toc522018712)

[2 4](#_Toc522018713)

[3 9](#_Toc522018714)

[canvas绘图核心知识点--重点 11](#_Toc522018715)

[4 12](#_Toc522018716)

[5 15](#_Toc522018717)

[6 18](#_Toc522018718)

[(1)sessionStorage 5种方法 类数组对象,会话级数据存储 6 4 16:44 21](#_Toc522018719)

[7 21](#_Toc522018720)

[localStorage 在客户端浏览器保存数据 22](#_Toc522018721)

[8 23](#_Toc522018722)

[BOOT 1 25](#_Toc522018723)

[Boot2 26](#_Toc522018724)

[3.2:Bootstrap 下载 27](#_Toc522018725)

[Boot3 29](#_Toc522018726)

[(5)栅格系统针对不同屏幕提供不同列 29](#_Toc522018727)

[3.3:Bootstrap全局样式--组件(css+js)-下拉菜单 31](#_Toc522018728)

[Boot4 32](#_Toc522018729)

[Boot5 34](#_Toc522018730)

[Node1 36](#_Toc522018731)

[Node2 40](#_Toc522018732)

[Node3 43](#_Toc522018733)

[nodejs 访问 mysql数据库 45](#_Toc522018734)

[Node4 46](#_Toc522018735)

[node5 49](#_Toc522018736)

[node6 50](#_Toc522018737)

今天学习的内容

1:课程简介

html5新特性--8 day[游戏/特效 Canvas]

Bootstrap css/js 框架 6 day[响应式布局框架]

Node.js 后端开发平台 6 day

2:html5十个新特性

(1)新的语义标签 footer header..

(2)增强型的表单(表单 2.0)

(3)音频和视频

(4)Canvas绘图(2 day 小鱼和大鱼->低龄儿童)

(5)Svg 绘图(地图)

(6)地理定位

(7)拖放API

(8)Web Worker

(9)Web Storage

(10)Web Socket

## 3:h5新特性--增强型的表单

(1)新的input type 1 1

h4:text/password/checkbox/radio/submit/button/reset

h5:email/url/number/tel/color/date/month/week

(2)新的表单元素

H4:input/button/select/textarea/..

H5:datalist/progress/meter/output

(3)新的表单属性

4: h5新特性--datalist 数据列表 选择输入 1 2

<datalist id="list3"> datalist本身不可见

<option>xx</option>

<option>yy</option>

</datalist>

**<input type="text" value="" list="list3"/>**

datalist为 input 提供输入建议列表，用户可以从中

选取，也可以手工输入.

5: h5新特性--progress 进度条 1 3 1 4

<progress></progress> 左右晃动进度条

<progress value="0.7" /> 具有指定进度值的进度条

value 取值范围 0~1

练习:实现一个可以动态前进的进度条，

使用定时器+进度条实现上述功能

到100%弹出"下载完成"提示

提示:项目中,推荐div+css来代替换process

项目中, 复杂进度条canvas

6: h5新特性--meter 注意属性5+ 刻度尺 1 5

Meter:刻度尺,用于标识一个值所处范围:

不可接受(红色) 可以接受(黄色) 非常优秀(绿色)

**<meter min="可取最小值" max="可取最大值"**

**low="合理下限" high="合理上限值" optimum="最佳值"**

**value="当前值">**

**</meter>**

示例:薪水级

<meter min="0" max="100000" low="20000" high="90000" optimum="99999" value="90000">

练习:动态变化的刻度尺 每隔 0.1秒 修改 value

0~100000 + 500

7: h5新特性--output 语义标签

output:输出，语义标签，没有任何的样式,样式同等span

商品单价:￥3.50

购买数量:<input type="number" value="1" />

小计:<output>￥3.50</output>

8:h5新特性--表单的<input ?>新属性 1 7

h4:type/name/value/id/style/class/disabled/checked

**h5**: 表单的新属性 8个

#placehodler:**占位符**

<input type="text" placehodler="输入用户名称 \"/>

#autofocus:**自动获取焦点**

<input type="text" autofocus/>

#multiple:**允许输入多个值**(用逗号分隔),如果邮箱

#form:用于把输入域**放置 form外部 id连**

#required: **必填**项,内容不能空

#maxlength minlength 指定字符串最大和最小**长度**

#max/min 指定数字**最小值和最大值**

#pattern **正则**表达式

9:学子商城后台管理[管理员登录/商品管理]

10:目录结构[c:/xampp/htdocs/h5/]

admin 项目中所有资源

css 所有css文件

js 所有js 文件

fonts 图标字体文件

data 所有php保存此目录

db 保存所有数据库脚本文本

img 保存所有图片

upload 上传图片

index.html 登录页面

main.html 主页面

product\_list.html 商品管理

建议:所有目录/文件/路径 不要有中文空格,特殊字符串

小写字母：下划线 admin order

11:第一个功能:后台管理员登录 1p照视频写10:53

学子商城:二类用户

(1)注册用户-->购买商品->浏览商品

(2)管理者->上架(下架,删除,更新)

操作流程 4步

(1)创建一张表 管理员数据表 xz\_admin 1p 16: 00

(2)data/01\_admin\_login.php 1p 16: 05 sql加‘’情况16: 46

(3)index.html

(4)index.js 17:45 不发ajax方法：函数里return（视频）、 2a9::27跳转：如code>0,localtion.href="url"

常见错误

(1)php访问时直接下载

原因:header()内容拼写错误

(2) mysqli\_connect(): (HY000/2002

原因:数据库没有成功启动,数据库严重错误

(3)mysql 坑 SET NAMES UTF8 正确

SET NAMES UTF-8 错误

(4) Table 'xz.xz\_admin1' doesn't exist

原因:xz\_admin1 表不存在,通常表名拼写不正确

(5) Unknown column 'uid1' in 'field list

原因:uid1列不存在,通常列名拼写不正确

今天作业:后台管理员登录

[HTTP协议]

# 2

今天学习的内容

1:复习昨天内容

2:作业:json格式

json - > AJAX

json:数据格式,通常是以字符串形式表示

对象 {"name":"james","age":19}

数组 [{"name":"tom"},{"name":"jerry"},{"name","james"}]

所有key 必须添加双引号

所有value 如果字符串类型加上双引号

key和value之间 : 分隔

对象与对象之间 ,

常用数据格式:xml json

<books>

<book>

<name>tom</name>

</book>

<book>

<name>jerry</name>

</book>

</books>

验证码:解决网络攻击手段(机器人攻击)

1:动态创建数字图片/字母数字角度/算术/

一组图片/拖动滑块

2:短信(金融)

项目：如何实现看图输字验证功能: 步骤3 2a 9:45 项目在day01

1:创建php动态生成一张图片

2:用户输入图片中字母或数字

js/index.js

3:01\_admin\_login.php 2a 9:52

先把用户输入字母和数字 a001

再把图片中数字获取 $\_SESSION["code"]

// 网络错误：数据库变了，可以看day01建个表视频

常见错误

1: syntax error,

<b>C:\xampp\htdocs\h5\admin\data\01\_admin\_login.php</b> on line <b>10</b><br />

原因:程序第10 行(附近)有语法错误...

2: network查到已返回{"code":1,"msg":"登录成功"} 看network——response/header

应该对象 header("Conten....");

3:今天的目标

3.1:h5新特性--视频,音频

视频:网站卖点[腾讯/爱奇艺/优酷/tmooc/抖音/直播..]

3.2:Flash被H5 取代替哪些方面?

Flash 绘图 ==>Canvas/SVG

Flash 动画 ==>定时器+Canvas

Flash 视频和视频播放==>video/audio

Flash 客户端存储 ==>WebStorage

3.3:h5新特性--视频播放2 1

h5提供一个新的标签用于播放视频

mp4/mov/webm/ogg

兼容写法：

第1种：<video src="x.mp4">您的浏览器版本太低，请升级<video>

2种：<video>

<source src="x.mp4" />

<source src="x.mov" />

<source src="x.webm" />

您的浏览器版本太低，请升级

</video>

它本身是一个300\*150的inline-block元素

**属性**和成员6+个 2 1

autoplay:false 是否自动播放

controls:false 是否显示控件

muted:true 是否静音

loop 循环

poster:"" 在播放第一帧之前显示海报

preload:"" 视频预加载策略

auto: 预加载视频元数据及缓冲一定时长视频

metadata: 只预加载元数据(尺寸,时长,第一帧内容)

none: 不预加载任何数据

#########js对象属性############ 2 1

currentTime: 当前播放时长

paused:true 当前视频是否处理暂停状态

volume:1 当前音量 0~1

playbackRate:1 回放速率 大于1快放,小于1慢播

成员方法:

play(); 播放视频

pause(); 暂停视频

成员事件:

onplay 播放视频

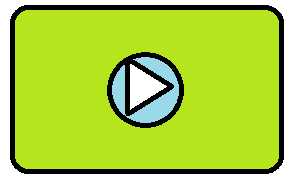
onpause 暂停播放

练习1:不使用video自带controls，自定义播放/暂停按钮2 2 (2p 14:30)

*（提示：如果处于暂停 if(v3.paused) v3.onplay=function() v3触发播放事件：广告隐藏）*

移标移出视频区域，隐藏按钮,鼠标移入视频区域显

示按钮



练习2:只要视频暂停显示一副广告,只要播放，就隐藏广告

周一:弹幕...

3.4:h5新特性--音频播放 2 4 2p15::14

h5提供了一个新的标签用于播放音频

<audio src="bg.mp3"></audio>

<audio>

<source src="bg.wav">

<source src="bg.ogg">

<source src="bg.mp3">

</audio>

它默认是一个300\*30的inline-block,但若没有controls属性,不可见.

成员**属性**:事件/方法/属性

autoplay:false 是否自动播放

controls:false 是否显示控件

muted:true 是否静音

preload:"" 视频预加载策略

auto: 预加载视频元数据及缓冲一定时长视频

metadata: 只预加载元数据(尺寸,时长,第一帧内容)

none: 不预加载任何数据

###js对象属性###

currentTime: 当前播放时长

paused:true 当前视频是否处理暂停状态

volume:1 当前音量 0~1

playbackRate:1 回放速率 大于1快放,小于1慢播

成员方法:

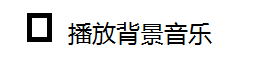
play(); 播放视频

pause(); 暂停视频

成员事件:

onplay 播放视频

onpause 暂停播放

2 4

练习:使用复选框控制网页背景音乐

cb.onchange = function(){

this.checked

}

3.5 h5新特性--canvas绘图(重点)

网页中走势图,统计图,网页游戏,特效,地图应用都要使用绘图技术。

(1)SVG绘图:2D矢量图绘图技术，2000出现，后纳入h5标（地图）

(2)Canvas绘图:2D位图绘图技术,h5 提出绘图技术（范冰冰）

(3)WebGL绘图:3D绘图技术，尚未纳入标准

|  |  |
| --- | --- |
| Canvas绘图难点所在:  (1)坐标系  (2)单词比较多 |  |

3.6 h5新特性--canvas绘图技术 2p 15:43

canvas:画布，是h5提供绘图技术

<canvas width="500" height="400">

您的浏览器版本太低，请升级!!!

</canvas>

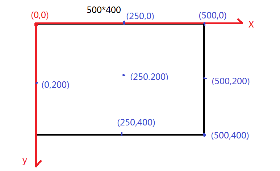
canvas标签在浏览器默认300\*150的inline-block

注意:2点

1.画布宽和高只能html/js属性来赋值，不能用css样式赋值

2.每个画布上有且只有一个"画笔"对象,

var ctx = canvas.getContext("2d");



(1)使用canvas绘制矩形

矩形的定位在自己左上角

画笔的**属性**

ctx.lineWidth = 1; 描边**宽**度

ctx.strokeStyle = "#000"; 描边**颜色**样式

ctx.fillStyle = "#000"; **填充颜色**样式

ctx.fillRect(x,y,w,h); 填充一个矩形(实心矩形) （起点，宽高）

ctx.strokeRect(x,y,w,h); 描边一个矩形

ctx.clearRect(x,y,w,h); 清除一个矩形范围内所有图形（原点，画布宽、高）

练习1:在画布的左上角，右上角，左下角，右下角， 2 5 16:13

居中位置 绘制100\*80描边矩形(颜色不同)

练习2:在画布上描边一个可以左右移动的100\*80描边矩形 2 6 16::43

提示:使用定时器，先清除画布上己有内容,

再重新绘制一个(x不停修改)

上下移动/右角45度..

(2)使用canvas绘制文本 2 7 2 8 17::26

画笔的**属性**

ctx.textBaseline = "alphabetic"; **文本基线**

ctx.font = "12px sans-serif"; 文本**大小和字体**

ctx.fillText(str,x,y); **填充**一段文本

ctx.strokeText(str,x,y); 只**描边**一段文本

ctx.measureText(str); 测量文本宽度{width:w} (放右侧时，找起点要用到2 7)

var w = ctx.measureText(str).width

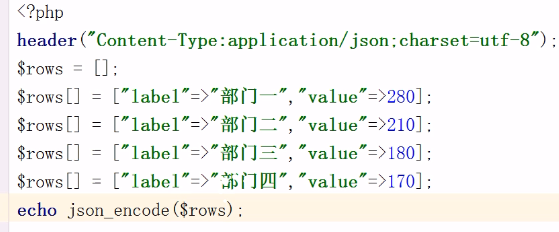
清除...

▲作业:运用：1将返回的数据作宽高画矩形 3 homework+php 2弹幕：画笔.文本,移动

1:创建web应用程序

salesdata.php 2 php 2p 17:55

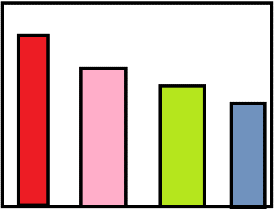
当客户发送请求给php时，返回JSON数据



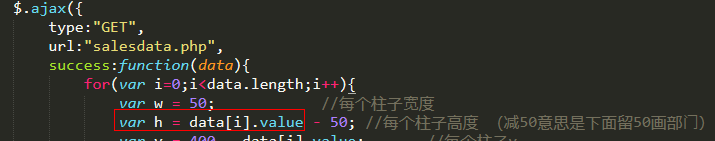
2:创建salesdata.html 3 homework

发送ajax请求salesdata.php

依据返回数据，绘制统计图



*疑问：怎么用高保存数据value？*



拿到data数据(json)——遍历data里设宽高xy——h = data[i].value——x = i \* 100 + 50 , y = 400 - data[i].value(定左上角)

随机生色函数：3 homework

# 3

今天学习的内容

1:复习昨天重点

2:作业

3:今天的目标

3.1:canvas绘图--(重点掌握:渐变对象.路径.图片.变形)

3.2:canvas绘图--渐变对象

**线性**渐变: linearGradient

**径向**渐变:...

创建一个渐变对象赋给填充颜色样式：3 1 3a 10:26 运用：字渐变、矩形渐变

var g = ctx.createLinearGradient(x1,y1,x2,y2); 径向方向（起点，终点）

g.addStopColor(offset,color);

..

ctx.fillStyle = g;

ctx.strokeStyle = g;

3.3:canvas绘图--路径(最复杂) 3 2 3p11:17 3 5

5步：

ctx.beginPath(); **开始新路径**

ctx.moveTo(x,y); 移动指定点

ctx.lineTo(x,y); 指定点**画**直线**到...**

ctx.closePath(); **闭合当前路径**

ctx.stroke(); **描边**

ctx.fill(); **填充**

ctx.arc(cx,cy,r,start,end); 绘制圆拱形 (圆心，半径，起角度，终角度) 3 2

ctx.arc(400,100,50,start\*Math.PI/180,end\*Math.PI/180);

cx cy 圆心

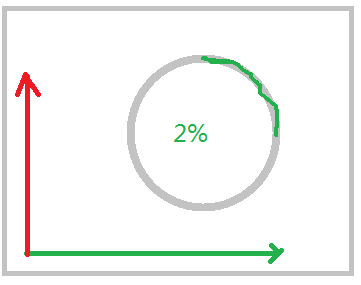
r 半径

start 开始角度 (弧度0~2\*PI)

end 结束角度 角度\*Math.PI/180==弧度

练习:使用描边绘制一个坐标轴 3 3 11:44

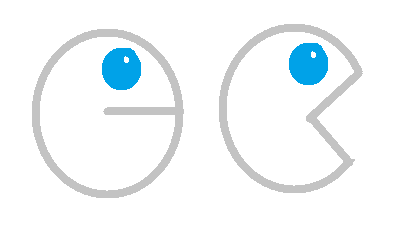
练习:使用圆拱形绘制可以前进的圆环进度条 3 5 14:07 看图想代码



练习:创建一个函数openMouth()/closeMouth(); 3 6 3a14::46

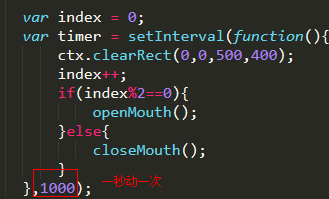
创建定义时器，每隔1s交替调用

openMouth()/closeMouth();



*疑问：怎么1秒交替调用，显示？*

*画好2函数——设num=0，定时器内——清除图形、num++，当num奇时调用1函数，****1秒后变更调用另一个函数就有持续状态了。***



3.4: canvas绘图--把图片画在画布3 7 15:40 3 8

canvas属于客户端技术，图片在服务器中，所以浏览器必须先下载要绘制的图片，且等待图片异步加载完成.

var p3 = new Image(); 创建**图片对象**

p3.src = "x.jpg"; 下载指定**路径**图片

p3.onload = function(){ 图片下载完成触发事件

console.log(p3.width);

ctx.drawImage(p3,x,y); **原始大小**绘图 （起点）

ctx.drawImage(p3,x,y,w,h); **拉伸**绘图

}

练习:在画布四个角各画一架飞机

练习:在画布左上角画一个左右移动飞机（飞机在画布上左右摆：清除图形） 3 8

*难度：定时器内要变的量和掉件判断 ：x xDirection 图片宽w=p0.width*

|  |
| --- |
| canvas绘图核心知识点--重点 绘制矩形:  ctx.fillRect();ctx.strokeRect();ctx.clearRect();  绘制文本  ctx.fillText();ctx.strokeText();ctx.measureText().width  绘制路径  ctx.beginPath();ctx.closePath();  ctx.moveTo();ctx.lineTo();  ctx.arc();ctx.stroke();ctx.fill()  绘制图像  ctx.drawImage(); |

3.4: canvas绘图--变形操作 3 11

canvas绘图可以针对于某一个图像/图形，的绘制过程

进行变形 rotate,translate

ctx.rotate(弧度); 旋转画笔，轴点画布的原点.

ctx.translate(x,y); 平移原点 3p 16:37

ctx.save(); 保存画笔状态(存盘)

ctx.restore(); 恢复画笔状态到上一次保存(读取存盘)

顺序：保存,平移原点,旋转,绘图,恢复

注意：原点、画布、旋转、画笔起点位置要区分

练习:在画布左上角画一个绕自己为中心旋转飞机1 3 10

变量——原点：translate 角度：10\*Math.PI/180——定时器

▲练习:在画布右上角画一个绕自己为中心旋转飞机2 3 11

飞机2旋转速度是飞机1 2倍svg绘图

练习:使用canvas绘制随机改变验证码图片! 4 homework 9:19 看图回忆代码



var str = "ABC..abcdefghim...1234567890";

var char = str[字符串长度随机数] 17:45~17:57

步骤：文本、线、点、随机色:

画布背景随机颜色[浅色],ctx.fillRect();

文字内容随机，大小随机，颜色随机(深色),旋转角度随机

5条随机干扰线(深色), 贝赛尔曲线

100个干扰点(半径为1圆)

作业:弹幕->常见 视频+canvas综合功能.

作业:学子商品-->商品列表分页

# 4

今天学习的内容

1:复习上周知识重点

2:作业:

[验证码;分页] 4a 10:37

分页:商品分页管理

data/01\_productlist.php 查询数据库数据返回

(1)参数 pno页码 pageSize 页大小(一页几行)

1 10 默认值

(2)sql1:查询总记录数

SELECT count(lid) as c FROM xz\_laptop;

sql2:查询当前页内容

SELECT lid,lname,price FROM xz\_laptop

LIMIT (pno-1)\*pageSize,pageSize

(3)返回数据

{pno:3,pageSize:8,pageCount:6,data:[...]

product\_list.html 显示数据

product\_list.js 连接php获取数据，并且显示html

|  |
| --- |
| Product.list分页疑问分析及步骤：4a 10:37  php：查记录总数count——*查当前页数据*select…from..limit (pno-1)\*pageSize,pageSize.  分页变量：发送：页码pno、内容数pageSize  返回：页码pno、内容数 pageSize、总页数：pageCount，当前页内容data  *返回数据：｛pno：3,pageSize:8,pageCount:6,data:[]｝*  *总页数pageCount：count查询抓取——[0]取到数量——$row[0]/#pageSize——得到页数*  *当前页内容data：查 lid lname price title ——limit (pno-1)\*pageSize,pageSize——抓取all（$rs,MYSALI\_ASSOC）*  *拼装：放数组——json\_encode($output)* |

常见错误

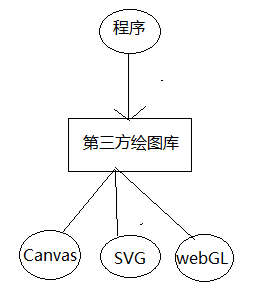
1: mysqli\_fetch\_row() expects parameter 1 to be mysqli\_result

原因:sql语句不正确

解决if(mysqli\_error($conn)){echo mysqli\_error($conn)}

3:今天学习的内容

3.1:h5新特性---第三方绘图工具库 echarts(canvas)



百度 echarts;d3;two.js;....

|  |
| --- |
| **运用echarts** 4 1 4a 40:00  条形图：4 2  实例初始化（id）：var myChart=echart.init(main)  数据：对象——标题｛｝、数据类名｛｝、x轴名｛对应下名｝、y｛｝、填充系列 [{数据类名，图，数据插入},{}]  放数据进实例  *疑问 数据转换的各种格式？Php过来数据放入echart* 4 2  *思路：最终要的是data.label数组和data.value数组*    *php返回json：$rows数组——里面插入两数组*    *请求：（接受到data对象，里面有两数组）*    *将数据放入echart：*  *xAxis:{data:data.label}*  *series:[{name:”销量”，type:”bar”,data:data.value}]*  **饼状**：4 3 |

3.2:h5新特性---SVG绘图 4P 5 130:00

3.2:h5新特性---SVG绘图--矩形

<svg id="" width="500" height="400">

<rect width="" height="" x="" y="" fill="" fill-opacity=""

stroke="" stroke-opacity=""></rect>

</svg>

练习1:在画布中央绘300\*30柱子，初始化淡红色(#faa)  
 填充和深红色(#800)边框，都是半透明  
 鼠标悬停时变为不透明,  
 提示:修改html元素属性setAttribute("",);  
 练习2:绘制一个柱子，高度1，使用定时器，不断修改

高度到300停止

练习3:使用ajax从服务器异步获取一段json数据

根据数据创建统计图

|  |
| --- |
| SVG 图形都是元素可以直接绑定事件监听  SVG 绘图特点  (1)所有图形默认只有填充色(黑色),没有描边色  (2)SVG图形的样式可以用元素属性声明，也可以用css 形式来声明，但是css声明只能使用svg专用样式，不能用css样式，如边框设置 stroke而不border  (3)图形可以用js对属性赋值,但不能使用HTML DOM形式，只能用核心 DOM操作  r3.x r3.width = 无效  r3.setAttribute("x",10);  (4)动态添加svg图形可以有两种方式  #HTML字符中拼接  var html = "<rect></rect>";  svg.innerHTML = html;  #创建元素  var rect = document.createElementNS(  "http://www.w3.org/2000/svg",  标签名);  svg.appendChild(rect); |

http请求方式：4p 16:46

3.3: h5新特性---SVG绘图--圆形

<circle r="" cx="" cy="" fill="" fill-opacity="" ...>

练习1:在svg画布上填充五个圆形

四个角中央一个

练习2:在 svg画布上随机绘制30个实心圆形, 5 1 看图想码 5a 11:33(消失)

位置随机,填充颜色随机,透明度随机

点击某个圆形后，它慢慢变大、变淡直至消至消失

*疑问*：*从DOM树删除：当透明度低时clearInterval(t); —— s3.removeChild(that);*

*防止连续两次点击时第二次删除出错：that.onlick=null*

作业:验证码模仿二遍/分页/HTTP

# 5

今天学习的内容

1:复习昨天内容

2:作业[HTTP协议/大鱼小鱼] 5 a 9:17

http协议复习：5 xmind --6a 9:14 请求头分解

8种请求方式5a 9:24 xmind 5

持久连接：减少握手挥手次数

1.0版本握手3次挥手4次 9:36（每个图片下载都）

1.1版本 共7次 完

game 5a game *9:54*

src 保存所有游戏图片

js 保存所有游戏js

commonFunctions.js 公共函数库

main.js 入口程序

background.js 绘制背景

index.html 显示网页

3:今天的目标

3.1:h5新特性--SVG--椭圆

<ellipse rx="" ry="" cx="" cy=""></ellipse>

**rx:水平半径**

**ry:垂直半径**

3.2:h5新特性--SVG--直线

<line x1 y1 x2 y2 stroke></line>

练习:使用SVG 中的直线绘制如下图标(汉堡包) 5 2 11:59看图想码



3.3:h5新特性--SVG--折线 (一条折线上可以有任意多个点) 5 3

<polyline points="**50,50 10,50 ..**" fill="transparent"

stroke="#000"></polyline>

3.4:h5新特性--SVG--文本

<text font-size="" fill="" stroke="" x= y=>文本内容</text>5 3

练习:使用折线绘制五角星

3.5:h5新特性--SVG--渐变对象 5 4 14:28

id2步：1

<defs> 定义特效对象:渐变对象属于一种特效对象

<linearGradient **id="g3"** x1="" y1="" x2="" y2="">

<stop offset="" stop-color="red" stop-opacity=""/>

<stop offset="" stop-color="red" stop-opacity=""/>

</ linearGradient >

<defs>

2：

<ANY **fill="url(#g3**)" stroke="url(#g3)"></ANY>

练习:创建三个柱子,使用渐变对象，为不同的柱子分配不同 5 5 看图想码

渐变对象

3.6:h5新特性--SVG--滤镜(高斯滤镜--模糊) 5 6 14:50

<defs>

<filter id="f2">

<feGaussianBlur stdDeviation="3" /> 模糊度：越高越模糊

</filter>

</defs>

<ANY filter="url(#f2)" >

滤镜网站：<https://developer.mozilla.org/en-US/docs/Web/SVG/Element/filter> （插入各种滤镜标签） 14:53

3.7:h5新特性--地理定位

Geolocation:地理定位,使用js获取当前浏览器所在地理坐标(**经度,纬度,海拨,速度**)数据，用于实现(Location Based Service)

如饿了么,高德导航...

手机浏览器如何获取定位信息

(1)首选手机中的**GPS芯片通信**，定位精度在米

(2)次选手机**通信基站**进行定位获取

PC浏览器如何获取定位信息

(1)通过**ip地址**解析

h5中提供了一个新对象,用于获取当前**浏览器定位**信息

window.navigator.geolocation{

getCurrentPosition:fn, 获取当前定位信息 （国外ip地址库，访问不了）

watchPostion:fn, 监听

cleaearWatch:fn 取消监听

}

国内第三方服务商(百度地图/腾讯地图) 5 7

#在网页中嵌入百度地图 15:33

(1)注册百度开发者帐户(手机号)

http://lbsyun.baidu.com/

(2)创建一个网站，登录百度地图，为网站申请15:38 密钥

AccessKey

(3)在自己网页中嵌入百度提供API 15:43 代码复制 5 7

坐标拾取器：15:47 (网站中有）

加标记：16:10 5 8

控件：导航控件 5 9

3.8:h5新特性--拖放API

HTML5中为拖放操作提供**7个事件,分两组 16:25**

拖动**源对象**:(会动)

dragstart 拖动开始

drag 拖动中

dragend 拖动结束

拖动**目标对象**:(不动)

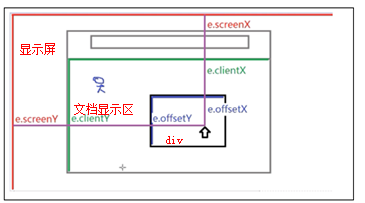
dragenter 拖动进入

dragover 拖动悬停

dragleave 拖动离开

drop 在上方释放

*注意：必须阻止dragover的默认行为，drop才可触发* 17:16



练习:使用拖动源对象提供事件,实现"可以随鼠标拖动而移动小飞机" **5 11**

提示：**p3.style.left = (ex-startX)+"px";**

**p3.style.top = (ey-startY)+"px";**

提示:父元素相对定位，子元素绝对定位，飞机移动，就是修改top/left属性，拖动事件可以取到相对整个页面左上偏移量e.pageX/pageY

▲练习:使用拖动事件，实现“拖动删除效果”5 13 17:09 看图想码

提示:在刚开始拖动时 src.ondragstart记录拖动源对象

的id,释放时target.ondrtop根据此id找到源对象,

执行删除 div.removeChild(c);

作业:学子商城分页/js [1][2][3][4][5][50]

[8][9][10][11][12]

# 6

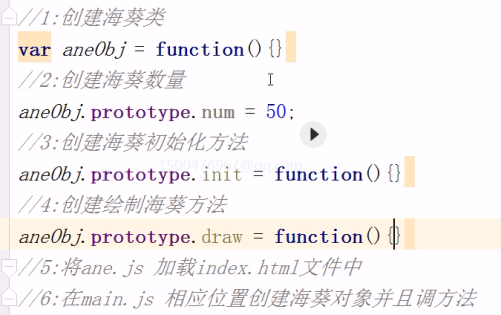
今天学习的内容

1:复习昨天内容

2:作业:[海葵/分页]

海葵：6a 9:50 6 game/ane.js 看图想码 （本次用创建构造函数/类方法做）

创建aneObj类：10:15



分页：10:42 *day6项目代码没有，day4加今天的才完整做出产品列表*

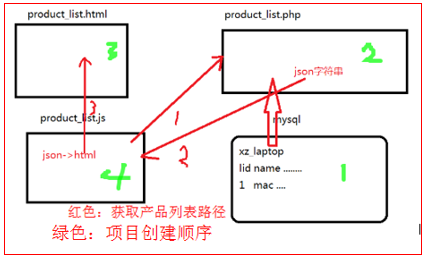
步骤：

1.product\_list.js 发请求：product\_list.php 11:09

2.js中创标签放入dom树

*疑问：全选功能完善 6 admin/js/product\_list.js 14:10*

项目流程图：



常见错误:

错误1：syntax error 01\_productlist.php 41

网络故障-->>F12->NETWORK->Response 响应主体

Headers 响应头

错误2: SQL syntax;

01\_productlist.php</b> on line 42

3:今天的目标

3.1:h5新特性八--Web Worker---代码就3行

 程序:program 存储在外存(磁盘)中代码

进程:Process/Task 将程序调用内存中，分配空间

线程:Thread 线程是进程内部代码基本单位

(1)进程是操作系统分配内存的基本单位

(2)线程是CPU执行代码的基本单位

(3)一个进程内部都多个线程组成，至少一个

(4)多个线程之间"并发执行的",宏观上看同时执行，

微观上看依次循环执行

chorme浏览器

一个 chrome进程内，有多个线程，负责向WEB服务器并发的发起请求,以获取资源: 请求资源线程

还有一个线程将所有内容绘制浏览器中--UI主线程,

还负责执行js/css代码.

线程代码分析：6 1 14:39

观察如下代码

<button>按钮一</button>

<script src="x.js">很耗时js任务</script>

<button>按钮二</button>

现象:js执行代码中，按钮一可见，按钮2不可见

原因:浏览器执行代码和渲染网页同一个线程

解决:创建新线程，由它来执行耗时js任务,

UI主线程负责渲染网页.

代码:

<button>按钮一</button>

<script>

var w = new Worker("x.js"); //不能dom/bom操作

</script>

<button>按钮二</button>

3.2:Worker线程缺陷

浏览器不允许Worker线程执行任何DOM/BOM操作

原因:浏览器只允许UI线程操作DOM/BOM，若多个

线程同时操作DOM/BOM页面，混乱..

所以类似jquery脚本不能使用Worker来执行

3.3:Worker线程可以给UI线程发送数据

worker: 发

**postMessage**(strMessage); 6 2->6 02js 6 15:17

ui: 收

var w = new Worker("x.js");

**w.onmessage** = function(e){e.data}

3.4:UI线程可以给Worker线程发送数据 6 03js->6 2

ui: 发

var w = new Worker("x.js");

w.postMessage(stringmeg);

worker: 收

onmessage = function(e){e.data}

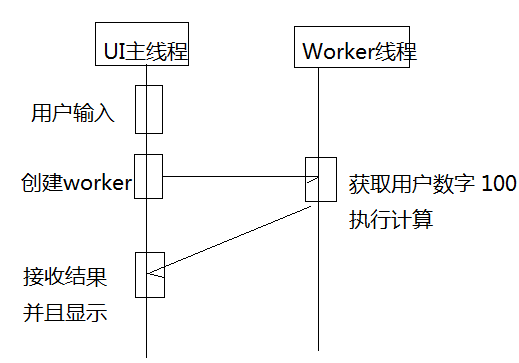
练习: (1)在HTML中有一个INPUT,请用户输入数字, 6 3+6 04js

(2)添加按钮"开始计算累加和"

(3)<div id="rs"></div>

点击此按钮,创建Worker 线程来计算出用户

输入数字累加和，在下方创建div显示计算结果



|  |
| --- |
| 项目中Worker的使用场景  (1)只要js中有DOM/BOM就不能用Worker  (2)Worker适合于执行耗时JS任务, 复杂计算,数据统计.. |

------------------------

3.5:h5新特性九--Web Stroage

在浏览器中存储一些用户专用数据:内容定制，样式定制..

在客户端存储数据可以使用的技术

(1)cookie技术,浏览器兼容性好,不能超4KB,操作复杂 *（f12）application里 6p 16:29*

(2)Flash存储，依赖Flash播放器

(3)**H5 WebStorage**,不能超8MB,操作简单

(4)IndexDB 可存大量数据，还不是标准

|  |
| --- |
| Session:会话（用户操作过程）浏览器从打开某一个网站第一页面开始.(会话开始),中间可能打开多个页面(会话中),直到关闭浏览器(会话结束)+  **欢迎回来例子：**php端的Session方法  16project jq 5p13 data user 三个php  5p 17::20 header.js 结构uname 放进.html()  *下有客户端的sessionStorage方法* |

WebStorage技术中，为用户提供二个对象

## (1)sessionStorage 5种方法 类数组对象,会话级数据存储 6 4 16:44

**保存**数据 sessionStorage[key] = value

保存数据 sessionStorage.setItem(key,value);

**获取**数据 var value = sessionStorage[key];

获取数据 var value = sessionStorage.getItem(key);

**删除**数据 sessionStorage.removeItem(key);

删除数据 sessionStorage.clear();

数据**个数** sessionStorage.length

获**取key** var key = sessionStorage.key(i);

创建login.html 输入用户名，密码，提交，用户点击 6 6+7+8 17:26 *(3个页面转换)*

提交按钮,提示"登录成功,3s后自动跳转 首页"

返回index.html 右上角 欢迎回来xxx 退出登录

创建logout.html 提示"您己退出登录,3s后自动跳转首页"

返回index.html 右上角提示"请登录"超链接

(2)localStroage

作业1:完善全选功能

网上查方法/网上查代码：

作业2:查方法：php上传文件(图片) 6 p 6/upload 18:03





简单问题网址 www.baidu.com

复杂问题网址思狗 https://segmentfault.com/ 6 p 18:07

# 7

今天学习的内容

响应状态码解释：7 9:35 http.xmind

1:复习昨天重点

2:作业[大鱼小鱼食物/上传文件/分页]

7 9:52 6 game/fruit.js 看视频，没写代码

/ 上传文件，举例上传图片

11:28 7 upload\*3 var\_dump($\_FILES); $name=$\_FILES["mypic"]["name"] --14:30前

/分页 7 admin/product\_list.html/js 14:40

[1][2][3]

[1][2][3][4]

[1][2][3][4][5]

3:今天的目标

3.2:h5新特性--WebStorage 7 16:16 7 1+2

## localStorage 在客户端浏览器保存数据

永久保存

保存数据 localStorage [key] = value

保存数据 localStorage.setItem(key,value);

获取数据 var value = localStorage [key];

获取数据 var value = localStorage.getItem(key);

删除数据 localStorage.removeItem(key);

删除数据 localStorage.clear();

数据个数 localStorage.length

获取key var key = localStorage.key(i);

3.1:h5新特性--WebSocket--代码不复杂重点原理

HTTP协议:属于"请求-响应"模型,只有客户端发起请求，服务器才会响应消息，没有请求就没有响应，一个请求，只能得到一个响应。有些场景中HTTP工作模型"不适合"

"股票走势图" 解决方案--长轮询/心跳请求

WebSocket协议:"广播-收听",客户端连接到服务器就不再断开,永久连接，双方随时向对方发送数据，ws在实时数据特别有优势.

ws 协议的应用程序可分为两个部分(服务器/客户端)

服务器:

监听指定端口.接收客户端请求，对向方发消息，并接收消息.可以php/java/node.js 创建服务器

客户端

主动发起请求，保持永久连接，向对方发消息，并接收消息，可以php/java/html5

掌握:使用html5创建ws协议客户端 7 16:49 7 3

启动ws服务器：ws\_server.js--> 7 17:13

1:连接ws服务器

var socket = new WebSocket("ws://127.0.0.1:9001");

2:向服务器发送消息

socket.send(stringMsg);

3:接收服务器消息

socket.onmessage = function(e){e.data}

4:断开ws服务器连接

socket.close();

练习:页面1：主题选项 *(本地存之前选的body的class——每次打开，js读取本地内存并执行) 7 04*

创建index.html，有一个下拉菜单 "请选择您喜欢主题"

"蔚兰天空","芭比公主","暗黑主题",各对应一个class名

.blue{background:#ddf;color:#33a}

.pink{ background:#fdf;color:#a3a }

.drak{ background:#333;color:#eee}

用户选中某个项目(select.onchange)为当前页面(body),

使用指定样式.

创建usercenter.html打开此页面即可应用index.html中一样样式,即使重启浏览器中，直接usercenter.html 仍是之前

曾经选中的主题.

页面2主题：可读取本地存的class名——如果有，js插进body的class 7 04+5 还可以监听本地储存变化

作业1：学子商城 删除指定商品

# 8

今天学习内容

1:复习昨天重点

2:学子商城 -- 后台管理

[管理登录;商品分页;**删除;更新;详细信息;商品搜索;商品添加**]

功能一:删除指定商品 开讲：9:40 分析 8 9:20

1:02\_product\_del.php 8 admin 9:40 *影响的行数mysqli\_affected\_rows($conn)*

参数:lid 商品编号

sql语句: DELETE FROM xz\_laptop WHERE lid = $lid

json格式 {code:1,msg:"删除成功"}

技巧一:由于数据非常重要通常不删除 9:27

lid price lname isdel[0,1] 0正常状态 1己删除

技巧二:保证数据健壮性 为表添加几列

**muid mtime v1 v2 v3 v3**

muid 修改的用户的编号 10

mtime 修改时间 2018-10-19

v1 v2 v3 v4 备用列（生产环境：尽量不要随意添加列）

技巧三：保证数据健壮性

每一张表创建删除历史记录表

xz\_h\_laptop(记录表)

lid lname price ... duid（谁删的） dtime（什么时候删的）

10 2018-10-10

2:product\_list.html

3:product\_list.js

点击事件如果绑定绑定参数

参数

确认框

发送 ajax请求

注意:删除之前要提示,查询之前要排序

功能二:更新指定商品价格 8 11:26 建议直接看product\_list.js 和php

1:依据商品编号获取所有商品信息

03\_product\_find.php

参数:**lid**

sql: SELECT lid,price,lname FROM xz\_laptop WHERE lid=$lid;

2:依据商品编号和价格更新价格

04\_product\_update.php

参数:**lid price**

sql:UPDATE xz\_laptop SET price=$price WHERE lid=$lid

技巧:价格/薪水/存款/余额 钱

xz\_laptop/price decimal(10,2) 商品价格

decimal 高精度浮点数

产品另一种方法:精度更高 使用int保存单位 "分"

12.00 电脑 ---> price(1200) 分

技巧:数据库 库名/名表/列名 严格区别大小写

坑: 开发 windows {乎略大小写}

SELECT LID,Lname FROM xz\_Latop

服务器 linux {严格区别大小写}

# BOOT 1

今天学习的内容

1:复习上周重点

2:作业:大鱼小鱼【食物】/添加商品/搜索

常见错误:

Column count doesn't match value count at row 1

原因:列数不匹配: 示例:20列,你添加19

3:今天学习目标

3.1:Bootstrap -- 响应式布局框架(另一种设计网页技术)

3.2:什么是响应式网页

1G: 只能通话

2G: 可以通话，短信

2.5G: GPRS 可以通话，上网--WML，短信

3G:IOS/Android(Linux) 可以通信,短信,视频,上网--HTML

**京东: www.jd.com m.jd.com**

Responsive Web Page:

响应式网页/自适应网页,一个网页，会自动根据用户浏

览设备不同，自动改变布局，可以被PC/PAD/PHONE

正常浏览.

3.3:如何实现响应式布局网页

a:使用Boostrap框架创建(简单->栅格系统)

b:自定义

3.4:自定义响应式布局网页

(1)添加视口元标签

|  |
| --- |
| View元标签  早期3G手机为了浏览大尺寸网页，只能强行把  网页缩小，导致，图片，文字，超链接都很小    IOS 提出"视口"的概念,视口用于盛放网页内容  大小可以随意指定 |
| <meta name="viewport"  content="width=device-width, initial-scale=1.0">  width=device-width 视口宽度，设备宽度  initial-scale=1.0 初始化缩放比例 1.0 不缩放 |

(2)使用流式布局 float:left;

(3)所有容器/文字/图片使用相对大小不用绝对大小

.container{width:100%}

.main{fonts-size:1em;}

(4)使用css3 媒体查询-查询用户浏览设备

3.5:CSS3 媒体查询 1 16:43

Media:指浏览网页设备,如 screen(pc/pad/phone),print,tv..

Media Query:查询出当前浏览网页设备类型以及特性

(对比度,尺寸,手持方向),有选择的执行某些CSS代码,

而忽略另外一些.

3.6:媒体查询两种使用方法

(1)根据媒体查询不同的执行不同外部css文件 1 1 17:11

<link media="screen" rel="stylesheet" href="1.css">

<link media="screen and (min-width:768px) and (max-width:991px)" rel="stylesheet" href="2.css">

(2)根据媒体查询结果执行css片段的部分选择器

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

......

}

作业1:搜索分页条点击事件

# Boot2

今天学习的内容

1:复习昨天内容

2:作业:大鱼/视频/搜索分页

大鱼:mom.js

数据:位置(x,y)角度(angle)眼睛(2);身体(8);尾巴(8)

行为:

大鱼面向鼠标缓慢游动过去!

大鱼吃食物

大鱼喂小鱼

3:今天的目标

3.1:Bootstrap 下载，配置，全局变量(栅格系统)

Twittler Bootstrap

中文网bootstrap官网网址: http://www.bootcss.com

Bootstrap是一个HTML/css/js 框架，用于开发响应式布局网页.

(1)起步 下载/配置

(2)全局样式 非常多 class

(3)组件

(4)js插件 图片轮播

(5)定制

提示:Bootstrap知识点不难,但难在:1000+

## 3.2:Bootstrap 下载

下载: dist/[css/js/font]

dist/css/bootstrap.css 框架所需所有css样式

dist/js/bootstrap.js 框架所有需所有 js

dist/fonts/\* 所有字体文件

创建模板文本

template.html

|  |
| --- |
| <html lang="zh-CN">  Language:指定当前网页所用自然语言:  zh-CN;en-US;ja  作用有两个:为浏览器翻译功能确定基础语言,  为读屏软件确定基础发音 |
| <meta name="viewport"> 视口 |
| <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta http-equiv="X-UA-Compatible" content="IE=9">  只能IE浏览器识别  告诉IE浏览器请使用最新引擎渲染网页  windows 95 网景(浏览器->FireFox)100%  IE 0.1% 98->IE 100%  IE6  IE7 IE6/IE7  IE8 IE6/IE7/IE8  IE9 IE6/IE7/IE8/IE9  IE11 IE6/IE7/IE8/IE9/IE10/IE11 |
| html5shiv.min.js  是一个自调函数，让老IE(6,7,8)兼容HTML5标签  respond.min.js  是一个自调函数,让老IE(6,7,8)兼容--CSS3 Media |
|  |

3.3:Bootstrap 全局样式--按钮

.btn 按钮基础样式

.btn-default 白底黑字的按钮

.btn-danger/warning/info/primary/success 五种颜色

.btn-lg/sm/xs 按钮四种大小

.btn-block 块级按钮

3.4:Bootstrap 全局样式--图片

.img-rounded 圆角图片

.img-circle 圆形图片

.img-thumbnail 缩略图

.img-responsive 响应式图片

3.5:Bootstrap 全局样式--文本

.text-daner/success/warning/info/primary

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

.text-left/center/right/justify

.text-uppercase/lowercase/captitalize

3.6:Bootstrap 全局样式--列表

.list-unstyled 去除提示符号，左补丁

.list-inline 行内列表

|  |
| --- |
| Bootstrap中屏幕分类 |
| 1:大型PC屏幕(Large-lg) w>=1200px  2:中等PC屏幕(Medium-md) 1200px>w>992px  3:小型PAD屏幕(Small-sm) 992px>w>768px  4:超小型 PHONE 屏幕(ExtraSmall-xs) 768px>w |
|  |

3.7:Bootstrap 全局样式--表格

.table 表格基础样式

.table-bordered 带边框表格

.table-striped 隔行变色

.table-hover 悬停变色

.table-responsive 响应式表格(坑)

3.8:Bootstrap 全局样式--栅格--(栅格布局系统)

网页布局可用方法

(1)TABLE 布局

好处:简单控制

不足:语义错误,页面渲染速度慢

(2)DIV+CSS 布局

好处:语义正确,页面渲染速度快

不足:不容易控制

(3)Bootstrap栅格(网页简单+响应式)

好处:语义正确,页面渲染速度快

不足:不能设备非常复杂网页

3.9:Bootstrap 全局样式--栅格

(1)最外层必须使用容器

div.container 或 div.container-fluid

(2)容器中可以放置任何内容，但是要使用栅格必须用 div.row

.container > .row

(3)一个.row中不能放其它的内容，只能放置 .col 列中可以放置其它内容

.container > .row > .col

(4)bootstrap中行默认12等分,每个列必须指定在行中占比

(5)栅格系统针对不同的屏幕提供了不同的列

.col-lg-1/2/3/4.../12

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

.col-sm-1/2.3///12

.col-xs-12/.../12

3.x:手工创建响应式布局网页(CSS3 Media Query)

作业:使用Bootstrap 栅格系统布局一个网页

# Boot3

今天学习的内容

1:复习昨天重点

2:作业:大鱼:视频

3:今天的目标

3.1:Bootstrap全局样式--栅格

(1)最外层元素 div.container 或 div.container-fluid

(2)容器里面可以放置任何内容,使用栅格

.container > .row

(3)一个row不能放置其它内容 .col

.container > .row > .col

(4)Bootstrap行中默认12等分

## (5)栅格系统针对不同屏幕提供不同列

.col-lg-1

.col-md-1

.col-sm-1

.col-xs-1

(6)要以使用"偏移列"实现指定列及其后的列向右偏移效果

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

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

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

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

(7)需要注意不同屏幕适用性问题

.col-xs-\* 适用于 .col-xs/.col-sm-/.col-md/.col-lg

.col-sm-\* 适用于 .col-sm-/.col-md/.col-lg

.col-md-\* 适用于 .col-md/.col-lg

.col-lg-\* 适用于 .col-lg

(8)一个列可以指定不同屏幕下不同宽度占比

div.col-xs-12 .col-sm-6 .col-md-3

div.col-xs-12 .col-md-6

div.col-xs-12 .col-sm-6 .col-md-6

(9)一个列可以指定在特定的屏幕下隐藏

.hidden-xs 仅在xs屏幕下隐藏

.hidden-sm 仅在sm屏幕下隐藏

.hidden-md 仅在md屏幕下隐藏

.hidden-lg 仅在lg 屏幕下隐藏

(10)栅格系统嵌套

.container >

.row >

.col >

.row>

.col>...

3.2:Bootstrap全局样式--表单

(1)默认表单

<form>

<div class="form-group">

<label class="control-label"></lable>

<input class="form-control">

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

</div>

</form>



(2)行内表单

<form class="form-inline">

<div class="form-group">

<label class="sr-only">用户名</label>

<input class="form-control" />

</div>

</form>



(3)水平表单

水平表单==表单+栅格变形

|  |  |  |
| --- | --- | --- |
|  | 默认栅格 | 水平表单 |
| 最外层元素 | .container | form.form-horizontal |
| 行 | .row | div.form-grouop |
| 列 | .col | div.col-\*-\* |

<form class=" form-horizontal ">

<div class="form-group">

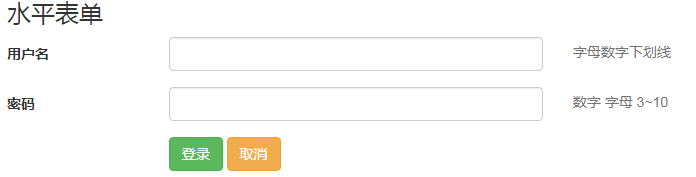
<div class="col-sm-3"> <label></label></div>

<div class="col-sm-6"> <input></div>

<div class="col-sm-3"> <span></span></div>

<div>

</form>



## 3.3:Bootstrap全局样式--组件(css+js)-下拉菜单

下拉菜单必需三级结构

<div class="dropdown">

<a href="" data-toggle="dropdown">触发元素</a>

<ul/div class="dropdown-menu">隐藏元素</ul/div>

</div>

3.4:Bootstrap全局样式--组件-导航

提示:Bootstrap中导航(nav)不同导航条(navbar)

(1)标签页式导航

<ul class="nav nav-tabs">

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

<li class="active"><a href="#" data-toggle="tab">二十元套餐</a></li>

<li><a href="#" data-toggle="tab">三十元套餐</a></li>

</ul>

(2)胶囊式导航

<ul class="nav nav-pills">

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

<li class="active"><a href="#" data-toggle="tab">二十元套餐</a></li>

<li><a href="#" data-toggle="tab">三十元套餐</a></li>

</ul>

3.5:Bootstrap全局样式--组件-字体图标



Web项目常用的图标字体

(1)FontAwesome -- 678 个免费图标

(2)Glyphicons - 800 个收费的图标

服务器端字体的使用方法

(1)web服务器项目目录下有字体文件

(2)css中声明一个新字体，并指定字体文件下载位置

@font-face{

font-family: 'Glyphicons Halflings';

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

}

(3)声明一个基础class,使用到了该字体

.glyphicon {

font-family: 'Glyphicons Halflings';

}

(4)在html页面中输入图标对应unicode

对页面开发者，图标使用方法

<span class="glyphicon glyphicon-pencil"></span>

练习:在按钮上添加下小图标

主页/用户/配置/汉堡包/刷新/打分/我喜欢(心形)

发邮件/拍照/定位/购物车/放大镜/删除/加号/减号

叉号

作业1:商品首页 大小小

# Boot4

今天学习的内容

1:复习昨天的内容

2:作业:小鱼/大鱼吃食物特效/分数/海葵摆动

3:今天的目标

3.1:Bootstrap--组件--进度条

.progress > .progress-bar

动态前进的进度条

3.2:Bootstrap--组件--小组件

面包屑 .bread-crumb

分页条 .pagination

徵章 .badge

巨幕 .jumbotron

水井 .well

3.3:Bootstrap--组件--缩略图

<div class="thumbnail">

<img>

<div class="caption"></div>

</div>

注意:缩略图本身是block组件,需要配合栅格系统控制宽度

3.4:Bootstrap--组件--媒体对象

<div class="media">

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

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

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

</div>

3.5:Bootstrap--组件--面板

<div class="panel panel-颜色">

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

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

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

</div>

面板有一个重要用途:实现手风琴组件

3.6:Bootstrap--js 插件

|  |
| --- |
| Bootstrap基于jQuery提供了十几个函数(bootstrap.js)  调用方法两种  (1)js编码方法  $(".dropdown > a").dropdown();  (2)data扩展方法  <a data-toggle="dropdown"></a> |

3.7:Bootstrap--js 插件--折叠(collapse)

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

<button data-toggle="collpase" data-target="#my-target">展开/收起</button>

<div class="collpase" id="my-target">....</div>

3.8:Bootstrap--js 插件--导航条

.navbar-header[头部==左侧广告+右侧汉堡包图标]

.navbar-brand

.navbar-toggle

.navbar-collapse[折叠=菜单/导航]

#导航条颜色

浅色底深色字: .navbar-default

深色底浅色字:.navbar-inverse

#导航条定位

相对定位: 默认

绝对定位: .navbar-fxied-\*

#导航条位置

顶部: .navbar-fixed-top

底部:.navbar-fixed-bottom

作业:综全示例:导航条实现

3.9:Bootstrap--js 插件--图片轮播

# Boot5

今天学习的内容

1:复习昨天重点

2:作业:大鱼吃食物(特效)/分数/海葵摆动

特效圆环:10个

当大鱼吃到一个食物后显示一个圆环,

圆环从小变大

3:今天学习的目标

3.1:Bootstrap js插件--图片轮播

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

<div class="carousel-inner">滚动元素

<div class="item active">

img + .carousel-caption

</div>

</div>

</div>

3.2: Bootstrap--定制

3.3: Bootstrap--定制-less(重点)--样式语言分类

(1)静态样式语言

css可以直接被浏览器所有解析,但作为一门语言,css并不完整，缺省编程序所必需的概念,数据类型,变量,运算符,函数等特性,导致

样式可维护性很差.

(2)动态样式语言(样式预处理语言)

第三种:Sass/Stylus/Less 在CSS的基础上添加了动态语言必需特性，如数据类型，变量，运算符，提高样式的可维护性.

需要注意:浏览器只能识别CSS,所有动态样式语言代码必须转换

CSS,该过程称为"编译(Compile)"

3.4: Bootstrap--定制-less(重点)

http://lesscss.cn/

less两种使用方法

(1)在客户端使用less---(学习)

编写x.less

在HTML中引入x.less文件，同时再引入一个less.js 编译程序

客户端请求html,下载less.js 文件,下载x.less 在客户端

运行less.js 文件将 x.less编译 x.css 运行

注意:每次客户端请求都会进行编译操作(减慢呈现速度)

(2)在服务器使用less---必须掌握

#下载并安装独立js解释器--Node.JS

在命令行中执行 node -v 可以查看到版本

#下载less编译程 lessc

lessc

#启动js解析器，执行lessc编译命令生成新 x.css文件

示例: lessc e:/01.less e:/01.css

3.5: Bootstrap--定制-less(重点)--语法学习

(1)less支持所有css语法

(2)less支持多行/单行注释,但只有多行注释编译css中

---推荐使用单行注释

(3)less支持"变量(Variable)"的概念

定义: @变量名:值;

使用: color:@变量名;

变量可以取值为任何合法的样式值

(4)less支持变量和常量的算术运算

+ - \* / %

(5)less支持在一个选择器中"混入"另一个选择器的样式

选择器1{....}

选择器2{...选择器1}

(6)less在样式混入时可以指定参数

选择器1(@参数,@参数){...}

选择器2{....选择器1(值1,值2)...}

(7)less支持样式嵌套

选择器1{

....

选择器2{

.....

}

}

上述代码会被编译为

选择器1{...}

选择器1 选择器2 {...}

(8)less提供十几个函数

ceil();

floor();

percentage(num); 所小数转换百分比形式

darken(color,num); 把指定颜色变暗

lighten(color,num); 把指定颜色变亮

image-width(url);

image-height(url);

(9)less中可以使用@import实现文件包含

可以把一个大项目所有需要的css 样式分开保存不同源文件中

有利实现分工协作.

3.6: Bootstrap--定制

Bootstrap的定制达到三个目标

(1)瘦身:删除不需要的样式

只需要注释掉bootstap.less不需 @import

(2)粗粒度定制

只需要修改varables.less 中定义变量值

(3)细粒度定制

修改组件对应的.less 文件, dropdown.less

示例1:瘦身

bootstrap/less/几十个.less

lessc bootstrap.less bootstrap.css

示例2: 粗粒度定制

修改危险按钮

# Node1

今天学习的内容

1:复习上周的内容

2:作业:海葵/英特尔首页/学子商城商品列表

3:今天的目标

3.1:node.js 基本概念/模块

3.2:阿里面试题:用户在浏览器中输入 www.taobao.com 直到看到

网页之间发生了什么?

(1)操作系统(window/linux/macos)访问网络上DNS服务器，把域名转换IP地址

(2)浏览器通过ip向淘宝服务器发送HTTP请求

(3)淘宝服务器接收请求，解析请求，查找资源,

可能访数据库,构建响应消息，发送浏览器中

(4)浏览器接收并解析响应消息

(5)浏览器缓存接收响应主体，解析和渲染响应内容

3.3静态网页和动态网页

静态网页:网页内容任何人在任何时间访问都是不变的

HTML/CSS/JS/Flash/视频/音频/....

动态网页:网页内容不同人在不同时间访问可能是不同

JSP/PHP/ASP/Node.JS

JSP = html+java 功能强大适合大型或超大型项目{卡车}

2003=1个月[php+mysql];5w;50w-->mysql->oracle

500w->php->java[..]->1000w->服务器->小型机

oracle+小型机

集群+java+淘宝(mysql)-> 1亿/1day

PHP = 简单易用;适合互联网项目[论坛] {家用轿车}

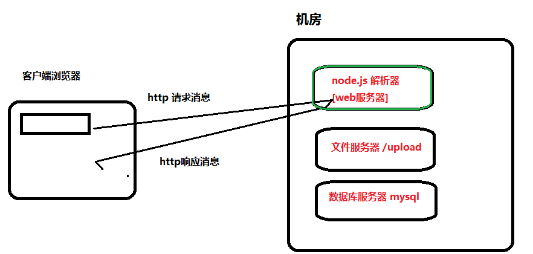
ASP = 微软产品,适全 windows  $$$$ {家用轿车} 300w

数据库 sql server 20w +开发工具+服务器

Node.js = 性能好,适合服务端IO密集型项目，不适合CPU

密集型项目 首页荐node.js {911}

|  |  |
| --- | --- |
| CPU密集型项目 | IO密集型 |
| 滴滴打车;高德导航;天气预报;石油预测分布 | I 输入 O 输出 数据获取，数据输出 |



3.3Node.JS

Node.js不是JS。是一种运行于**服务器端开发平台**，它的开发语言使用 js ,历史上第一个可以通吃前后台语言.

官方网站: www.nodejs.org

淘宝镜像: https://npm.taobao.org/mirrors/node

版本:0.12(16年初) 4.x(16年中)6.x(16年底)

node-v8.10.0-x64.msi windows下64位

node-v8.10.0-x86.msi windows 下32位

node-v8.10.0.pkg 苹果电脑

3.4Node.JS 两种运行模块

(1)交互模式---一般用于临时测试

node 回车

(2)脚本模式--正式项目中使用方式

把要把的要执行语句编写一个文本文件(后缀任意，没有可行)

一次性提交node解释器执行

node x.js 回车

提示:只要安装完node.js 重启一下webstorm，可以自动发现

node.exe 解释程序,新建项目编号一定utf-8

File->Settings->Editor->File Encoding->

Global Encoding->UTF-8

Project Encoding->UTF-8

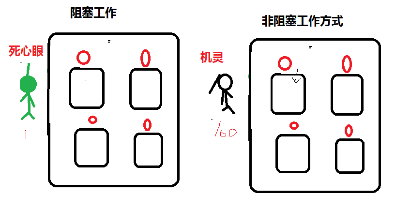
练习:创建02.js 打印九九乘法表, 以脚本模式来运行

3.5 Node.JS 如何自学一门语言-- (node.js)

(1)了解背景

--百度百科 node.js

[Node.js 使用事件驱动， 非阻塞I/O 模型]



(2)搭建开发环境:编写Hello World

(3)数据类型-----重点

|  |
| --- |
| 前端JS数据类型  (1)原始类型  string/number/boolean/undefined/null  (2)引用类型  ES对象类型:Date/RegExp/Object/Error/...  DOM/BOM:window/document/location/event/....  用户自定义:{} |
| 后端Node.js 数据类型  (1) 原始类型  string/number/boolean/undefined/null  (2) 引用类型  ES对象类型:Date/RegExp/Object/Error/...  用户自定义:{}  (3)Node.js 第三方对象 |

(4)变量和常量

var age = 20;

const PI = 3.14;

(5)运算符

算术运算符

比较运算符

逻辑运算符

三目运算符

位运算符

赋值运算符

特效运算符 instanceof typeof .

(6)逻辑结构

循环结构:for(;;);do{}while();while(){};for(in);for(..of..)

练习:03.js 声明一个保存5人学生成绩的数组,

使用三种for循环依次打印出每个成绩值

选择结构:if...else; switch...case

练习04.js 创建一个变量 var path = "/index"

使用二种选择结构,判断path的值为哪种(/index;/search;/login)

调用不同的执行函数.

(7)通用小程序 九九乘法表 100以质数 数组排序

练习:05.js 打印100以所有质数

(8)函数和对象

(9)常用的组件，第三方工具，框架

(10)实际小项目

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

3.5:Node.JS 中特有概念--模块

一个web项目中可以分为多个不同"模块", 用户模块，订单模块 ，支付模块...

Node.js按照不同功能，将函数，对象保存在不同文件或者目录

这些文件或目录称 "模块"

Node.js中每个模块都是一个独立构造函数,解析器为每个.js

添加下代码.

(function(exports,require,module,\_\_dirname,\_\_filename){

exports:{} 用于声明向外部导出自己成员

require:fn 用于导入其它模块，创建指定模块对象

module:{} 指代当前模块对象!

\_\_dirname 二个下划线 当前模块所有属性目录绝对路径

\_\_filename 当前模块绝对路径

//自己编写代码

});

语法:

exports.属性名=值; #导出指定数据

var m = require("./模块名称"); #导入指定模块并创建对象

练习:08.js 创建三个变量(uname/age/ctime)一个函数(fnRun)导出

09\_app.js 引入08.js 模块并且调用三个变量和一个函数

|  |
| --- |
| Node.js 模块中exports和module.exports的区别是什么?  二者都可以用于向外界导出自己的内部成员,但  module变量指代当前模块对象，真导出是 module.exports  node.js 底层 exports = module.exports  所以:若只是给exports添加新成员，则等价于module.exports添加新成员，但若修改了exports指向，则不会产生实质作用. |
| exports.age = 10; 等同于 module.exports.age = 10;  成功导出成员 |
| exports = {uname:"tom"} 导出失败  **module.exports** = {uname:"jerry"} 成功导出对象 |

练习:创建一个文件模块 10\_circle.js 其中定义常量PI，声明一个

方法 getSize(r) ,根据传入圆形的半径，返回其面积,

以及 getPerimiter(r); 返回其周长，再创建一个应用程序

11\_app.js 引入 10\_circle.js 模块，调用其公开二个成员方法

作业1：今写NODE.JS提供一个模块:URL

创建模块文件 myURL,向外导出一个方法resolve(url)

该方法接收的参数形式如下:

http://www.jd.com:443/ad/index?uname=tom&pno=3

返回对象:形如

{

protocol:"http",

servername:"www.jd.com",

port:443,

pathname:"/ad/index",

query:{uname:"tom",pno:3}

}

# Node2

今天学习的内容

1:复习昨天的内容

2:作业:myurl;首页

3:今天的目标

3.1:node.js 模块概念

node.js 中文件或目录统称为"用户自定义模块"

(1)文件模块

创建一个js文件，如:m3.js 导出需要公开的数据.

其它模块可以require("./m3")模块

(2)目录模块

方式一:创建一个目录，m3其中创建名3.js文件，创建,

package.json指定 main属性 3.js导出

数据,其它模块require("./m3");

方式二:创建一个目录,m4其中创建名**index.js** 文件，导出数

据,其它模块 require("./m4");

方式三:创建一个目录 **node\_modules**,其中再创建目录模块,

假设名为m6,其中创建模块描述文件 **package.json**文件,

其中声明 main属性指定默认要执行js文件,比如 6.js,

其中导出数据，其它模块 require("m6")



练习:使用方式3，创建二个目录模块circle;rectangle;都对外公开

两个方法,size(),perimeter()返回指定图形的面积和周长

最后在最外层的主模块中引入上述两个模块

circle.size(r); circle.perimeter(r);

rectangle.size(w,h); rectangle.perimeter(w,h);

|  |
| --- |
| 练习:创建m7.js 文件导出数据  创建m7 目录/index.js 导出数据  创建node\_modules/m7 目录/index.js 导出数据  ? require("./m7"); 引入哪个模块 1  ? require("m7"); 引入哪个模块 3 |
| NPM 包管理器  Node Package Manager 的第三方模块管理器,可用于下载,  更新,删除,维护包依赖关系的工具  npm 默认到 www.npmjs.com 网站下载第三方模块  ***下载命令***  (1)下载安装 node.js mysql驱动程序  npm install mysql 简写 npm i mysql  (2)下载安装 node.js mongodb 驱动程序  npm install mongodb  删除  npm uninstall mysql  为了提升下载速度使用淘宝node.js 镜像 99.99%  (1)下载工具cnpm (淘宝工具下载镜像数据)  (2)cnpm i mysql  npm install -g cnpm --registry=https://registry.npm.taobao.org  cnpm i mysql |

3.2:node.js 原生模块--querystring

原生模块:node.js 解析引擎自带

querystring 模块用于处理HTTP请求URL中查询字符串

var obj = qs.parse(str); #把查询字符串解析为js对象

var str = qs.stringify(obj); #把js对象解析为字符串

使用:

(1)加载模块 const qs = require("querystring");

(2)直接使用函数 var obj = qs.parse(str);

3.3:node.js 原生模块--url

url模块用于解析一个HTTP请求地址，获取其中各个不同的部分

var obj = url.parse(str); 把一个URL字符串解析为一个对象

var obj = url.parse(str,true); 把一个URL字符串解析为一个对象

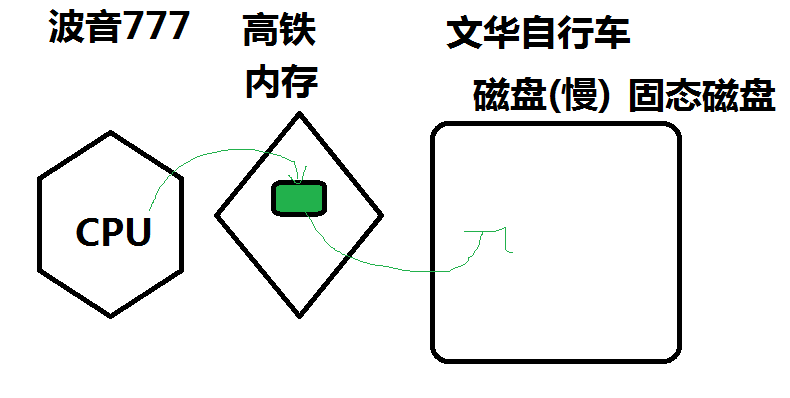
并把其中查询字符串也解析为一个

对象

3.4:node.js 原生模块--Buffer

Buffer：缓冲区,本质是一块内存区域，用于暂时存储数据

(数字,字符串,图片...),该区域称为缓冲区



#分配一个指定大小的缓冲区 var buf1 = Buffer.alloc(1024);

#使用一个数字数组创建一个缓冲区var buf2 = Buffer.from([1,10]);

#使用一个字符串创建一个缓冲区 var buf3 = Buffer.from("abc");

#如果缓冲区是字符串转换字符串形式 buf3.toString();

|  |
| --- |
| 计算机存储单位 字节 byte  1024byte = 1kb  1024kb = 1mb  1024mb = 1gb  1024gb = 1tb  1024tb = 1pb  计算机速度单位 毫秒 1000ms=1s 纳秒 ns 10^9=1s |

3.5:node.js 原生模块--fs--重点

fs模块提供了对文件(文件/目录)系统的操作增删改查,读写的功能

#阻塞(同步)方式操作文件

var data = fs.readFileSync(fileName); 阻塞**读取**文件内容

fs.writeFileSync(fileName,data); 阻塞数据**写到文件**中

注意:

#如果当前写入文件不存在，自动创建

#如果当前写入目录不存在，报错

#写操作之前清空操作

练习:使用上述方法,实现把 4.css文件复制为 4.backup.css

fs.appendFileSync(fileName,data); **追加数据**到文件中

#非阻塞(异步)方式操作文件

非阻塞读取文件

fs.readFile(fileName,(err,data)=>{

#当文件读取完毕后自动触发事件->调用回调用函数

err 如果读取文件发生错误,抛出错误

data 文件内容

});

fs.writeFile(fileName,str/buff,(err)=>{

#当数据写入文件结束触发事件->调用回调用函数

});

常见错误:

Error: ENOENT: no such file or directory

#没有匹配文件或者路径

#原因:目录不存在

fs.appendFile(fileName,str/buff,(err)=>{

});

练习:使用非阻塞方法 4.css 复制 44.backup.css

练习:使用非阻塞方法 向44.backup.css 追加 3.css

17:50~17:55

3.6:node.js 原生模块--http--重点

提示:后续的项目中，文件读取写入使用非阻塞方式--最大限度

发挥node.js 优势.

.

作业1:复制public下所有文件到 backup中

# Node3

***3 5***

，，，用node对数据库进行基本的操作

今天学习的内容

1:复习昨天重点

2:作业:遍历目录中文件;首页后半部分 3 9:21

文档;手册 *工作上问题：查官方文档*

百度 思否 3 9:13

3:今天的目标

3.1:node.js 原生模块--http--重点 3 1 11:50

HTTP模块可用于编写基于HTTP协议(客户端和服务器),

用http模块编写一个web服务器

(1)加载模块创建服务器对象 const http = require("http");

var server = http.createServer();

(2)绑定监听端口 server.listen(3000);

1-65535 建议:1023以上

(3)读取客户端请求数据 server.on("request",(req,res)=>{ 3 1 3 11:00

*request:事件名称*

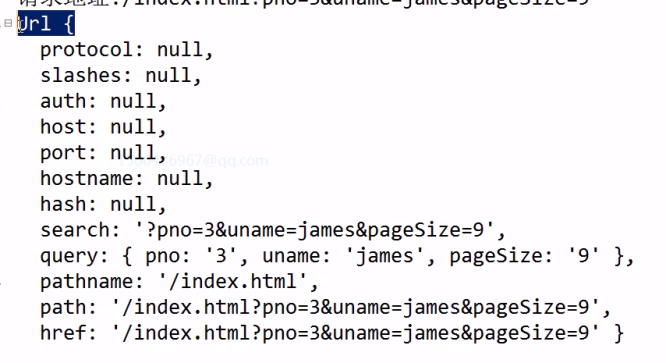
*req:request 请求对象*

*res:response响应对象*

res.write("hello");发送响应数据

res.end(); 响应数据结束

})

req:

(4)向客户端响应数据

练习:解析请求地址

请求地址:/index.html?pno=3&uname=james&pageSize=9

/index.html

pno=3&uname=james&pageSize=9 js obj

常见错误:

Error: listen EADDRINUSE :::3000

端口地址被占用!

解决:停止以前程序

3.2:node.js 第三方模块--mysql--重点 3 14:42

数据库种类

(1):关系型数据库 mysql/oracle/sqlserver/...

(2):非关系型数据库 Redis/memcache/

(3:大数据存储 HBase

SQL种类

DDL: 数据定义语句 CREATE/DROP

DML: 数据操作语句 INSERT/UPDATE/DELETE

DQL: 数据查询语句 SELECT

DCL: 数据控制台

## nodejs 访问 mysql数据库

为了精简node.js解释器,官方没有提供任何数据库相关模块.

,必须使用指定**数据库**(驱动程序)**模块**,下载 [www.npmjs.com](http://www.npmjs.com) 3 15:12

npm i mysql

mysql 驱动模块的使用步骤 3 3 3 15:15

(0):复制node\_modules mysql驱动

(1):加载数据库模块，创建连接池 3 3 15:31

const mysql = require("mysql");

var pool = mysql.createPool({.....});

(2):从连接池中获取连接，发送sql

pool.query(sql,(err,result)=>{result});

(3):获取返回结果

错误:在主键列 fid 添加二个相同值

Error: ER\_DUP\_ENTRY: Duplicate entry '1000' for key 'PRIMARY'

Error: ER\_NO\_DB\_ERROR: No database selected

网络攻击手段:SQL注入 3 16:33

利用sql语句习惯，将危险字符添加sql,对系统造成破坏!

示例:

var sql = ` SELECT count(uid) as c FROM xz\_admin`;

sql += ` WHERE uname = '${uname}' `;

sql += ` AND upwd = md5('${upwd}')`;

SQL注入:var uname = "' or 1=1 #";

解决方案:***使用占位符***,将所有参数替换?,再通变量添加sql语句中

node.js对参数过滤

var uname = "tom";

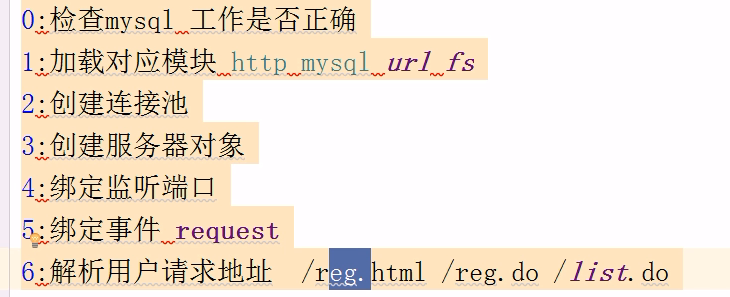
var upwd = "123";

select count(uid) as c FROM xz\_admin

where uname=? AND upwd = md5(?)

pool.query(sql,[uname,upwd]);

练习1:使用http和mysql 模块***3 5 3 17:27***



创建动态web服务器,接收如下请求

/reg.html 服务器端返回 /reg.html 注册{用户名/密码}

/reg.do 接收客户端提交参数 uname,upwd,保存mysql

数据库中 xz\_admin

作业:

/list.do 从数据xz\_admin表中获取所有数据, ***4 1*** (***3 5->) 9:11***

按table方式输出表中的内容

3.3:node.js 第三方模块--express--重点

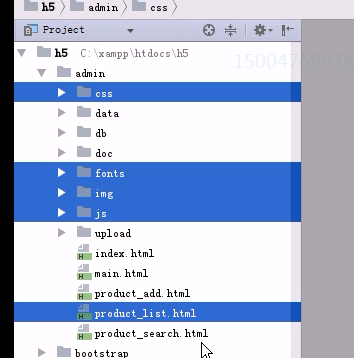
# Node4

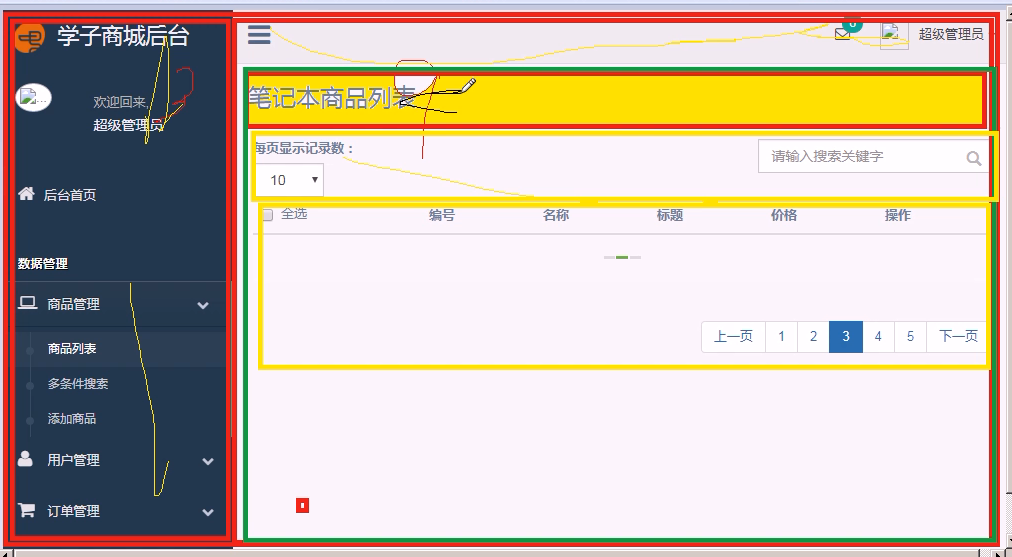
今天学习的内容

1:复习昨天重点

2:作业[list.do; ***4 1*** (***4 5->) 9:11***

用bootstrap做学子商城商品列表 4 homework02  ***9:28***]

复制进4 homework，做product\_list(例子)



Error: connect ECONNREFUSED 127.0.0.1:3306(连接池连接数据库出错，少写了(database：“xz”)

数据库出错

3:今天的目标

3.1:node.js 第三方模块-- mysql-- 驱动程序 4 10:28

a:后台开发流程（省略server创建）（例：后台管理登录、更新商品价格、4 1）

#加载模块

#创建连接池

#创建sql语句

#获取返回结果??

b:常见错误

3.2:node.js 第三方模块-- express 11:28 *（不用解析url和参数了）*

node.js 官方提供HTTP 模块可以创建一个web服务器应用，

但此模块非常底层,要处理问题太多，推荐express 简化http模块

此模块需要单独下载 （node\_module里已有了：express）

[www.expressjs.com](http://www.expressjs.com)

[www.expressjs.com.cn](http://www.expressjs.com.cn)

↓

3.3: node.js 第三方模块--使用 4 2 11:33

#下载 npm i express

#加载指定模块 http express

#创建express对象，创建server对象并绑定端口

#处理各种不同请求

app.get(url,(req,res)=>{}

GET index.html

app.get("/index.html",(req,res)=>{

res.sendFile("/index.html"); //url;fs;write;end;

res.json([]);//

res.send("<html></html>");

});

常见错误:使用sendFile发送文件一定绝对路径

(1)path must be absolute or specify root to res.sendFile

./index.html --> c:/xmapp/htdocs/h5///index.html

(2)Error: Cannot find module 'mysql'

没有mysql模块 node\_modules

(3) Error: Can't set headers after they are sent.

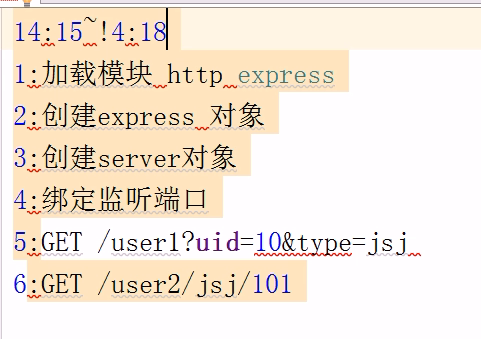
二次调用 res.json(); res.send();

(4) Error: ER\_BAD\_FIELD\_ERROR: Unknown column 'price1' in 'field list'

未知列 price1

原因:列名称错误

3.4: nodejs --express 处理用户请求参数 4 3 4 4 14:14



(1)接收GET请求中查询字符串4 3 14:12

GET /user1?uid=10&loc=bj (获取url中的参.在**query**中)

app.get("/user1",(req,res)=>{

//express为每个req对象添加一个query属性

var uid = req.query.uid; //取参方法

var loc = req.query.loc;

})

(2)接收GET请求中请求参数 4 4 14:12

GET /user2/jsj/60 (获取url中的参.在**params**中)

app.get("/user2/:type/:id",(req,res)=>{

//express为每个req对象添加一个属性 params

req.params.type;

req.params.id;

});

练习:创建express服务器，配置mysql模块

GET /product/3 向客户端输出3号商品所有信息

商品信息json格式;

作业: 6 app.js 参考 4下午视频18:00前

//10:第五功能:查询指定商品详细

//11:第六功能:搜索指定商品 key low high ??

//12:第七功能:添加商品

admin说明版本 项目说明1：

**h5：**

h5 admin项目在boot 2

**node.js 做后台admin：在node 6**

node 6中的nodeadmin项目是用node代替php做后台,node.js 4 15:12  **(6 app.js)**

mapp说明：

node：

# node5

今天学习的内容

1:复习昨天知识重点

2:作业:详情;搜索;添加 6 app.js 5 9:15

3:今天学习的目标

3.1:nodejs---express-- 处理各种不同请求 5 11:23 node的不同请求方式格式

HTTP协议:请求方法:用于标识此次请求目的

GET 表示客户端想"获得"服务器资源

POST 表示客户端想"上传"服务器数据

PUT 表示客户端想"更新"服务器数据

DELETE 表示客户端想"删除"服务器数据

5 11:39 演示post delete put

|  |
| --- |
| GET /user 获取所有用户信息 |
| GET /user?uid=3 获取所有编号为3用户信息 |
| POST /user uname=tom&upwd=123 表示上传数据到服  务器, 在数据库中添加一条记录 |
| DELETE /user uid=3 删除编号为3用户信息 |
| PUT uid=3&upwd=456 修改用户编号为3密码456 |
|  |

3.2:POST uid=3 5 1+index.html 11:43 14:15

exports接收post请求

app.post(接收请求地址,(req,res)=>{

req.on("data",(buf)=>{

var obj = qs.parse(buf.toString());//1.post中取请求主体中的数据：绑data事件——函数中将缓冲区字符串转对象

});

});

用户请求静态资源时：不写：*app.get("/index.html",(req,res)=>{* 5 14:45 =>项目 6 nodeadmin 5 14:55 （用node的express**放静态资源**）

*res.sendFile(\_\_dirname+"/index.html"); })*

直接创静态资源目录：

app.use(express.static("public")); //express 指定目录 public 静态资源目录

3.3:DELETE uid=3

app.delete("接收请求路径",(req,res)=>{

接收参数方法同 post

})

3.4:PUT接收参数方法同 post

app.put("接收请求路径",(req,res)=>{})

|  |
| --- |
| GET 和POST 区别 ，，，记 5 11:28  ?请求目的 GET 获取服务数据(要东西)  POST 发送数据给服务器(给)  ?传数数据量 GET 1023字节 1kb [字母几百个]  POST 没有限(默认2M)(如做视频->50MB  改配置文件php.ini默认值？)  ?请求主体 GET 没有  POST 有请求主体  ?提交方式 GET很多种 ajax、表单、地址栏敲、超链  接a 、location.href src link  POST2种 ajax 、form method="port"  ?安全性 GET 不安全  POST 不安全 |

3.2:nodejs---express-- 中间件

node——admin项目说明：5 15:13 6 nodeadmin

*前端js只改ajax的url，连到对应的nodejs就可以了*

# node6

今天学习的内容

1:复习上周重点

2:作业:

3:今天的目标

3.1:node.js --express-- 中间件(函数) | 5 am2 9:39 9:18

Express是一个自身功能极简框架，提供对外接口,“中间件”

中间件:是一个函数，它可以访问请求对象,响应对象,(req,res)

控制程序流程.

3.2:中间件种类

内置中间件: static 静态资源目录中间件

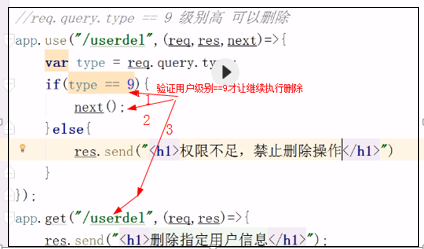
app.use(express.static("public"));

应用级中间件:

路由级中间件:

第三方中间件:cookie session cors 下课了解

3.3:应用级中间件 {公司保安} 6 01\_middle.js | 5 am2 9:20 6 1 9:25



语法: app.use("路径",(req,res,next)=>{})

路径:中间件拦截路径

函数:如果拦截成功，触发函数执行

next:执行后续的代码 next();

3.4: 路由级中间件 6 router/order 5 am2 10:33

路由:请求方式+请求地址+处理函数

app.get("/list",(req,res)=>{..});

解决问题:app.js 代码过长问题 命名重复冲突

app.get("/del") app.get("/del");

user.js /user 所有用户相关请求 /del /list /find /update

product.js /product 所有产品相关请求 /del /list /find /update

order.js /order 所有订单相关请求 /del /list /find /update

最终结果 /order/del /user/del

示例:user.js 5 am2 10:33

1:创建文件 ***router/user.js***

2:创建路由对象 var router = express.Router();

router.get("/list",(req,res)=>{})

...

3:返回路由对象 module.exports = router;

4:主程序加载路由对象 ***app.js***

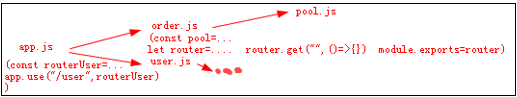
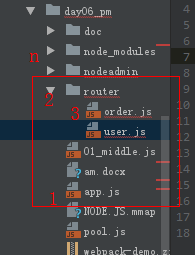
var m = require("./router/user");

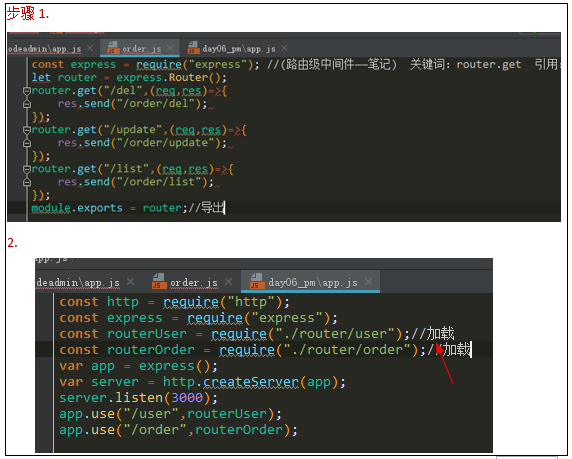
5:为模块创建访问路径,加载路由对象

app.use("/user",m) //

/user/list

路由规律：



常见错误:

Router.use() requires a middleware function but got a Object

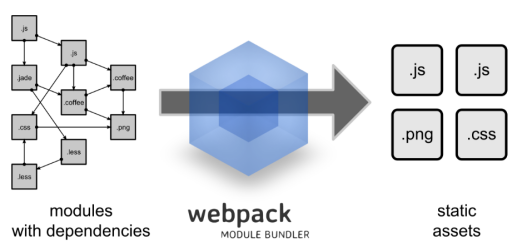
1:拼写出错

2:没导出指定对象

3.5:项目布署(打包)工具 webpack/gulp 5 am2 11:22

|  |
| --- |
| 8种方法：1.js打包 webpack test.js bundle.js  2. 打包一个引用文件 然后html 引用 bundle.js  3. 配置文件webpack.config.js引入导出 总文件夹：webpack  4. webpack --watch 随着数据变动变动  5. 打包多个文件 配置文件webpack.config.js引入导出格式  6. 生成package.json文件 修改命令 再输npm run dev  7. 下载css加载器 改配置文件loader css打包  8. 下载图片加载器 改配置文件loader 图片打包 |





webpack工具(Vue/NG/Rect/...需要打包操作)

webpack布署打包工具(gulp)

webpack 一款打包工具，它能把各种资源(js/jsx/less/css/图片)

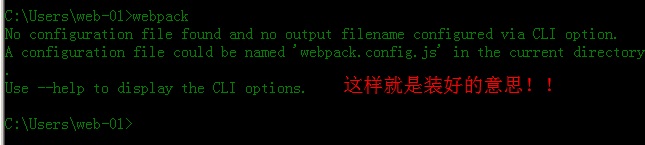
最后打包一个文件或多个文件中.

3.6:webpack 安装和配置 5 am2 11:30

检查是否安装上了:

开始->搜索程序和文件->cmd

webpack

说明里面有webpack

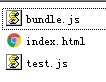
安装步骤：5 am2 11:35

(1)全局下载并安装webpack npm i webpack@3.3.0 -g

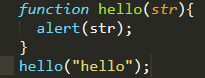
(2)在本地再安装一次 npm i webpack@3.3.0 -save-dev

3.7:案例:

案例一:打包一个js文件



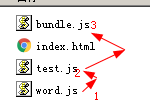
(1)创建目录e:/webpack-demo/01

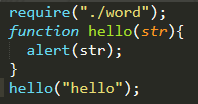
(2)创建文件test.js 

(3)打包语法: webpack test.js bundle.js

(4)index.js 引用 bundle.js

案例二:打包一个引用文件(引用其它js)



test.js

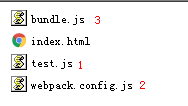
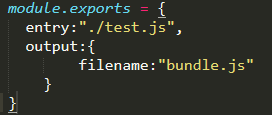
(1)创建目录 e:/webpack-demo/02

(2)test.js word.js

(3)webpack test.js bundle.js

(4)index.html 引用 bundle.js

案例三: 使用webpack配置文件5 pm2 14:03 (6 webpack-demo 03 )

2.

然后在03文件总夹——进命令窗—— 输webpack——系统直接找到配置文件——打包

(1)创建目录 e:/webpack-demo/03

配置文件webpack.config.js

module.exports = {

entry:"./test.js",

output:{

filename:"bundle.js"

}

};

打包命令 在03文件总夹——进命令窗——输 webpack

案例四: 使用webpack 参数：webpack --watch

webpack --progress 打包时显示进度条

webpack --watch 监听代码改动并自动打包



案例五: 使用webpack打包多个文件 5 pm2 14:16 6 webpack-demo 04

关键词：webpack webpack.config.js



在05文件夹——命令窗—— webpack

main1.js bundle1.js

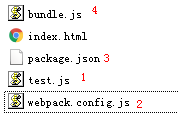
main2.js bundle2.js

webpack.config.js



案例六: 生成package.json文件 修改命令 再输npm run dev

5 pm2 14:25 6 webpack-demo 05



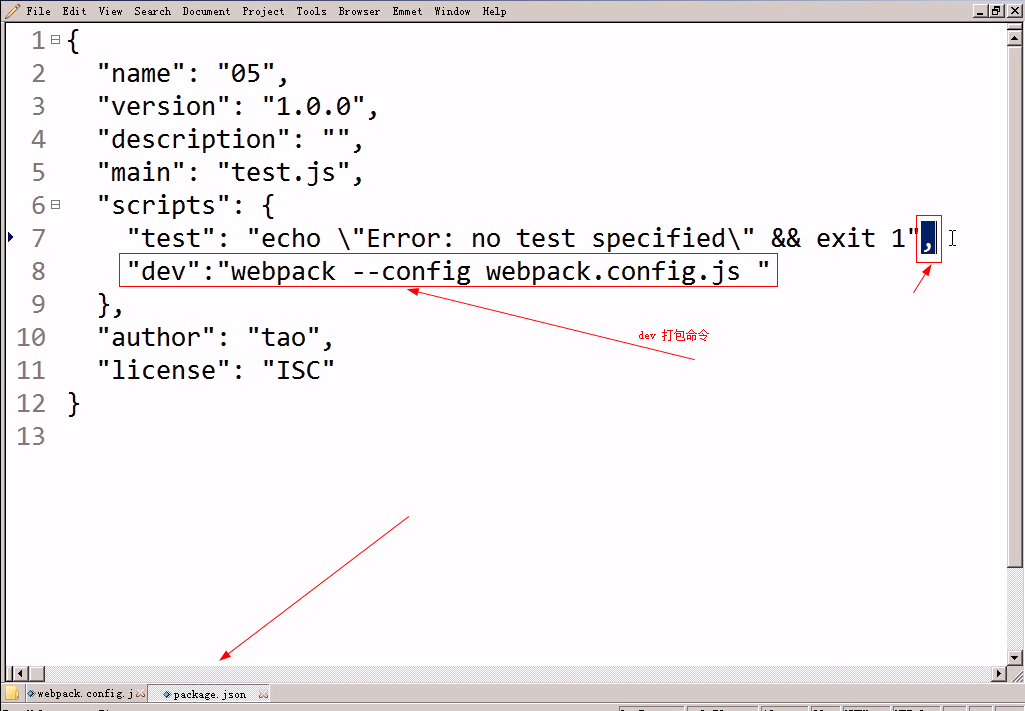
(1)e:/webpack-demo/05

(2)生成package.json文件 （5 pm2 14:28）

npm init 自动生成package.json

(3) "dev":"webpack --watch" （定义打包命令）

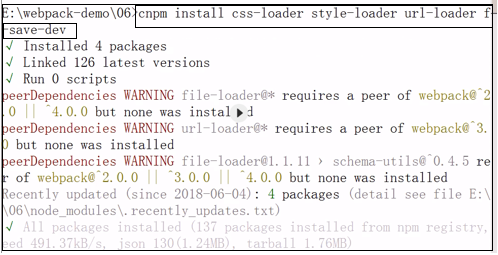
(4) 运行 npm run dev 打包（在cmd敲）

5 pm2 14:30

案例七: 下载css加载器 改配置文件loader css打包 5 pm2 14:40 6 webpack-demo 06



步骤：先下载包：（cnpm install css-loader style-loader --save-dev）



(1)index.css index.html

(2)创建test.js 引入css

(3)修改 webpack.config.js webpack打包

案例八:打包img文件



(1)下载：cnpm install url-loader file-loader --save-dev

(2)修改配置webpack.config.js

(3)打包

glup：6 glup



3.8:项目 5 pm2 15:22

用户管理--更新用户信息(用户密码) 修改用户信息 6 nodeadmin app.js

10 tom [更新]

旧密码[123] 旧密码不正确

新密码[...]

[更新] 5 pm2 15:24 6 nodeadmin app.js

/ufindPwd 用户id/用户旧密码/用户新密码

(1)sql SELECT upwd FROM xz\_user WHERE uid = ?

/uupdate **更新密码** *顺序：服务器端——js*

(2)sql UPDATE xz\_user SET upwd=? WHERE uid = ?

视频说明：（node里的5 pm3 多的）