# 1:html5新特性-简介

1.1：html5新特性-简介

html5十个新特性

-新的语义标签[header;footer...]

-增型表单

-视频与音频

-绘图 canvas(统计图;分析图;特效;小游戏)

-绘图 svg

-地理定位

-拖放API

-Web Worker

-Web Storage

-Web Socket

## 1.2：增型表单

**(1)新的input type** <input type=?>

h4:text;password;radio;checkbox;button;submit

h5:number;email;url;color;date;month;week

建议大家**谨慎使用h5属性，兼容性**

**(2)新的元素** element

**-datalist** 建议列表

配合input创建建议列表，当用户不清楚如果输入内容

提供建议

<datalist id="list3"> 默认情况datalist不可见

<option>xx</option>

<option>yy</option>

</datalist>

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

**-progress** 进度条

显示一个进度条两种形式

<progress></progress> 左右晃动进度条

<progress value="0.7" /> 具有指定进度值进度条

练习:使用定时器+进度条实现一个可以动态

前进的前度条到 100%停止

定时器 1s 修改value+0.1 到1结束

**-meter** 刻度尺

<meter min="最小值" max="最大值" low="下限" high="上限"

optimum="最佳值" value="当前值" />

当前值离最佳值非常远 (红)危险

当前值离最佳值比较近 (黄)警告

当前值离最佳值非常近 (绿)正常

**-output** 输出:语义标签,没有任何外观样式

外观同span

商品单价:￥3.50

购买数量:<input type="number" value="2" />

小计:<output>￥3.50</output>

**(3)新的属性** attr <input ? />

h4:id;class;name;value;style;readonly;checked

h5:

-autofocus: 自动获取输入焦点

-placeholder: 占位符

-form: 用于把输入域放置在form外部

<form id="f5">

</form>

<input type="text" form="f5"/>

-multiple: 允许输入多值(用逗号分隔) email

a@a.com,b@b.com

-验证相关

required:必填项，内容不能为空

minlength;maxlength: 最小字符串长度

min;max 数值最大值与最小值

pattern 输入正则表达式

## 1.3：多媒体视频

Flash

Flash绘图(AS/Flex) -> Canvas+SVG

https://echarts.baidu.com/ echarts第三方绘图库

https://d3js.org.cn/ d3绘图库

Flash动画(游动) -> Canvas+第三方游戏平台

https://www.cocos.com 微信小游戏

Flash视频(音频) -> video/audio (90%)

Flash存储 -> h5 WebStorage

## 1.4：多媒体视频使用

<video src="x.mp4"></video>

知识补充:

(1)现在常用视频格式哪些 .mp4 .flv .webm .ogg

(2)如果浏览器需要播放指定格式视频,需要安装一个解码器

软件

x.mp4 -> mp4解码器

x.flv -> flv 解码器

(3)如果低版本浏览器安装解码器

<video src="x.mp4" />

问题:我们希望大多数浏览器都可以播放视频

-x.mp4 使用工具软件 "格式工厂" x.flv x.webm x.ogg

<video>

<source src="x.mp4" />

<source src="x.flv" />

<source src="x.webm" />

<source src="x.ogg" />

您的浏览器版本太低，请升级!!!

</video>

## 1.5：多媒体视频-常见属性

controls:false 是否显示播放控件 <video controls>

autoplay:false 是否自动播放 <video autoplay>

#autoplay 兼容性非常差

loop:false 是否循环播放

muted:false 是否静音播放

poster:"" 在播放第一帧画面之前显示海报

preload 视频预加载策略

-auto 预加载一定时长视频和元数据

-metadata 只预加载元数据(时长;尺寸;第一帧画面)

-none 不预加载任何数据

## 1.6：多媒体视频-js属性

volume:1 当前音量

playbackRate:1 回放速度大于1 快放小于1慢播

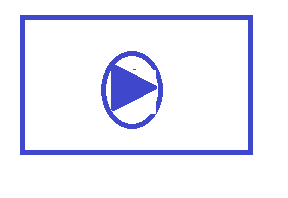
paused:false 当前视频是否处于暂停状态

play() 播放视频

pause() 暂停播放

onplay 当视频开始播放时触发事件

onpause 当视频暂停播放时触发事件



练习1:不使用video自带controls自定义 播放/暂停按钮

鼠标移入视频区域显示按钮，鼠标移出视频区域

隐藏按钮

练习2: 为按钮绑定点击事件 播放/暂停 切换图片

练习3: 创建表单登录表单 背景循环播视频

提示:最外层父元素 相对定位

表单 绝对定位

背景视频 绝对定位 z-index:-1 loop muted

今天学习的内容

1:复习昨天的内容

2:作业:

2.1删除购物车中一项商品(重点)

(1)xz\_cart [id/count/price/pid/uid] 2条记录

(2)node.js

参数:id

sql: DELETE FROM xz\_cart WHERE id = ?

json: {code:1,msg:"删除成功"}

#注意事项:删除是一种比较危险操作

解决方案: xz\_cart [id/count/price/pid/uid/isdel(0;1)]

1 2 9.9 3 1 0

2 1 19.9 4 1 1

SELECT id,count,price,pid,uid FROM xz\_cart WHERE isdel=0

(3)脚手架

-为删除按钮绑定点击事件

-发送ajax请求 node.js

-将组件中购物车数组list对应值删除

2.2 表单

# 2:HTML5新特性--视频高级特性/音频/绘图(重点)

## 2.1:视频高级特性-样式

video元素专用样式 **object-fit**

fill:"填充" 默认值 将视频拉伸操作填满 content box

contain:"包含" 保持原有尺寸比例，容器空白

cover:"覆盖" 保持原有尺寸比例,宽度或高度至少有一个可容一致

注意:cover;contain 使用比例多

如果创建视频与录制视频工程师 750\*1334 iphone6 屏幕(全屏播放)

## 2.2:视频高级特性-特殊属性

#扩展知识: 苹果 IOS/谷歌 Android

(同层播放)

IOS <video webkit-playsinline="true"/>

Android <video playsinline="true"/>

结果

<video webkit-playsinline="true" playsinline="true"/>

(横屏错位)

Android

<video x5-video-player-fullscreen="true"></video>

## 2.3:视频高级特性-特殊事件

-timeupdate 当视频播放时候[时间发生变化]不断触发事件

#currentTime 当视频播放时间

-ended 当视频播放结束时候

-canplaythrough 当视频加载结束可以播放时触发事件

#duration 视频时长

## 2.3:音频

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

#建议:为了兼容性问题使用格式工厂将音频文件转换多种格式

<audio>

<source src="x.mp3" />

<source src="x.wav" />

<source src="x.ogg" />

</audio>

属性方法事件大多数与视频相同

练习:创建复选框控制背景音乐播放

当复选框选中 播放背景音乐 play()

当清空复选框 背景音乐停止 pause()

cb.change cb.checked==true

## 2.4:绘图--(重点)

网页中绘制图形;股票走势图;分析图;网页特效;网页游戏...

都需要使用绘图技术完成

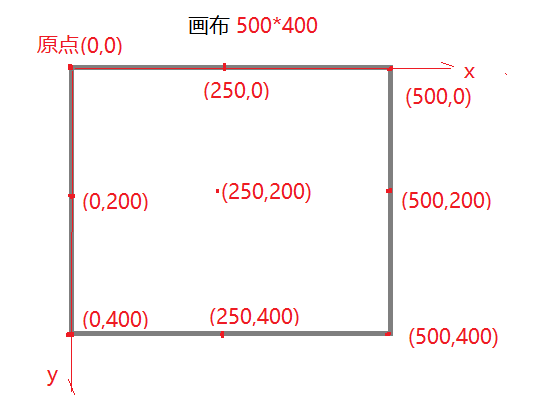
网页中绘制图形三种技术

(1)svg 绘图技术:2d矢量图绘制技术 2000出现纳入h5标准

(2)canvas 绘图技术:2d位图绘制技术

(3)webgl 绘图技术: 3d 位图绘制技术，尚未纳入h5标准

## 2.5:canvas绘图难点二个[坐标系;单词多]



## 2.6:canvas绘图矩形

#创建画布

<canvas id="c3" width="500" height="400"></canvas>

注意事项:画布宽度和高度只能使用属性或js赋值,不能用css

样式来赋值(css缩放，变形)

#通过 js程序获取画布

var c3 = document.getElementById("c3");

#通过 js程序获取画笔[上下文对象]

var ctx = c3.getContext("2d");

#注意事项:**一个画笔对象对应一个画布对象**

-ctx.lineWidth = 1; 描边宽度(空心矩形边框宽度)

-ctx.strokeStyle = "#fff"; 描边样式

-ctx.strokeRect(x,y,w,h); 描边一个矩形

-ctx.fillStyle = "#fff"; 填充样式(实心矩形内容样式

-ctx.fillRect(x,y,w,h); 填充一个矩形

#注意事项:x y 矩形左上角位置 w h 宽度和高度

#注意事项:矩形定位点在自己左上角

-ctx.clearRect(x,y,w,h); 清除一个矩形范围内所有元素

练习:在画布左上角右下角左下角右下角居中位置绘制

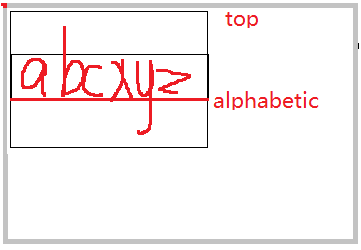
五个填充矩形颜色不同

练习:在画布上绘制可以左右移动100\*80矩形

提示:使用定时器先清除画布己有内容，再重新绘制矩形

(x不停增加)

## 2.7:canvas绘图文本



ctx.textBaseline = "top"; 设置文本基线 alphabetic 第三根

ctx.font = "19px SimHei"; 设置文本大小和字体

var str = "abcxyz"; 创建文本对象

ctx.fillText(str,x,y); 绘制填充文本

ctx.strokeText(str,x,y); 绘制空心文本

ctx.measureText(str); 测量文本宽度 {width:x}

#注意:先绘制文本对象再测量文本宽度

练习一:画布四个角绘制文本

练习二:依据服务器端程序数据绘制销售统计图

(1)创建服务器端程序

(2)发送ajax 请示获取销售数据并且绘制统计图

src/08\_app.js 服务器端程序

public/index.html 客户端程序

作业1:一次性删除购物车中多个商品

明天作业3:视频弹幕

今天学习的内容

1:复习昨天重点

2:作业

2.1:购物车删除多个商品(一)

(1)node.js

-参数 ids "3,4"

#参数需要选中购物id列表 多个商品格式: 3,8,9,4

# 一个商品 3

-sql DELETE FROM xz\_cart WHERE id IN (3,8)

-json {code:1,msg:"删除成功"}

(2)脚手架

[全选] <input type="checkbox" :checked="item.cb"/>

list:[{id;pid;price}]

list:[{id;pid;price;cb:false}]

-将原先数据list中商品对象添加属性 cb:false 表示复选框状态

# 3:销售统计图

(1)node.js

(2)index.html

## 3.2: canvas绘图-路径

path: 由多个坐标点组成任意图形，图形不可见

可以使用"描边" "填充"

ctx.beginPath() 开始一条新路径

ctx.closePath() 闭合当前路径

ctx.moveTo(x,y) 移动到指定点

ctx.lineTo(x,y) 从当前点到指定点画一条直线

ctx.arc(cx,cy,r,start,end); 绘制圆拱形

cx,cy 圆心

r 半径

start 启始角度

end 结束角度

#注意事项:程序不使用角度->弧度

#0 180 360 角度

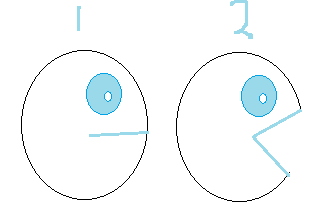
#0 PI 2\*PI 弧度

#通用解决方案:公式 角度转弧度

#角度\*Math.PI/180 = 弧度

ctx.stroke(); 描边

ctx.fill(); 填充



练习:

创建二个函数:openMouth() 张嘴笑脸

closeMouth() 闭嘴笑脸

每隔一秒钟切换另一个笑脸

## 3.3: canvas绘图-图像

canvas属于客户端技术(运在浏览器)但是图片保存服务器中

，所以浏览器必须先下载要绘制图片，且等待下载完成再

绘制.

#图片为什么保存服务器

(1)图片有版权软件项目宝贵资源

(2)图片通常数量巨大

#操作流程

-创建图片对象 var p3 = new Image();

-下载图片 p3.src = "x.png";

-绑定事件[*下载完成*] p3.onload = function(){}

-绘制图片 ctx.drawImage(p3,x,y,w,h);拉伸大小

ctx.drawImage(p3,x,y); 原始大小

练习:在画布左上角 右上角 左下角 右下角

绘制四架小飞机

练习:在画布上绘制一个可以随鼠标移动而移动小飞机

## 3.4: canvas绘图-变形(旋转)

canvas绘图技术可以针对一个图像在绘制过程中进行

变形(旋转)操作

ctx.rotate(弧度); 旋转画笔对象，旋转轴心画布原点

ctx.translate(x,y); 将整个画布原点平移到某处

练习:以250，200为轴心不停旋转小飞机每隔1秒旋转10度

练习:绘制二架小飞机分别是小飞机1(10)和小飞机2(20)

小飞机2旋转速度小飞机1 2倍

注意事件:

通用规则:当一个画布上绘制一个以上的元素画之前先"保存"画之后"恢复"这种操作使不同元素之间不受影响.

ctx.save(); 保存画笔状态

ctx.restore(); 恢复画笔状态

作业1:搜索如何使用canvas + video 弹幕功能

作业2:删除购物车中多个商品

今天学习的内容

1:复习昨天重点

2:作业

2.1:购物车删除多个商品

全选: [ ]

[] mac

[]华为

<input type="checkbox" :checked="item.cb" />

list:[{id:1,lname:,cb:false}]

昨天遇到问题:双向绑定

data(){list:[....]}

- var rows = result.data.data; //原有数组里所有属性双向绑定

- for(var item of rows){

item.cb = false; //后新添加属性不会双向绑定

}

this.list = rows; //双向绑定 ok

2.2:清除某个商品

[\*] <input type="checkbox" value="" :checked="item.cb"

@click="modifyItem" :data-index="i"/>

## 3.5:canvas-渐变对象(过渡效果)

-创建渐变对象

var g = ctx.createLinearGradient(x1,y1,x2,y2);

x1 y1 起点坐标

x2 y2 终点坐标

-添加颜色点

g.addColorStop(offset,color);

示例:

g.addColorStop(0.3,"green");

-将渐变对象赋值样式(描边;填充)

ctx.fillStyle = g;

ctx.strokeStyle = g;

-绘制图形

# 4.1:svg

|  |  |  |
| --- | --- | --- |
|  | canvas | svg |
| 类型 | 2d位图 | 2d矢量 |
| 如何绘图 | 使用js代码绘图 | 使用标签绘图 |
| 事件绑定 | 只能将事件绑定画布 | 每个图形都可以绑定事件 |
| 应用场景 | 特效,小游戏 | 地图 |

W3C html <html> <HTML> <br /> <BR >..灵活

XML 淘汰 java 配置文本 传递

html5 标准 (xml svg绘图)

## 4.2:svg-开发流程

-创建画布

<svg id="s3" width="500" height="400">

创建图形标签

</svg>

-在画布中添加标签(图形)-矩形

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

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

练习:左上角 右上角 左下角 右下角 居位置绘制

五个矩形，颜色不同

svg绘制图形特性

1:所有图形默认只有填充色(黑色)

2:svg图形的样式可以用元素属性声明，也可以使用css

样式声明，但是CSS声明只能使用SVG专用样式,不能

css样式

如边框只能用 stroke 而不能用 border

3:图形可以使用js赋值，但不能使用html dom**只能使用**

**核心dom**

r5.width = 300; r5.height=100; error

r5.setAttribute("width",300); ok

r5.setAttribute("height",100); ok

4:动态添加svg图形

(1)html字符串拼接

var html = "<rect></rect>";

svg.innerHTML = html;

(2)创建对象

var rect = document.createElementNS(

"http://www.w3.org/2000/svg","rect");

svg.appendChild(rect);

## 4.3:svg-开发流程-圆形

<circle r="" cx="" cy=""></circle>

练习:在画布上创建 30个大小随机,位置随机,颜色随机,

透明度随机圆形

练习:当用户点击某一个圆形此圆形慢慢变大

变淡直至消失 从 dom删除

## 4.4:svg-开发流程-椭圆

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

cx,cy 圆心

rx:水平半径 ry:垂直半径

## 4.5:svg-开发流程-直线

<line x1="" y1="" x2="" y2=""></line>

x1,y1 起点坐标

x2,y2 终点坐标

练习:使用直线绘图图标 "汉堡包"

## 4.6:svg-开发流程-折线

一条折线上可以有任意多个连续点

<polyline points="50,50 100,50 ..."

stroke="" fill="transparent"></polyline>

## 4.7:svg-开发流程-文本/图像

svg画布不能使用 span p标签不可以用

<text font-size="" fill="" stroke="">文本内容</text>

<image xline:href=" x.png" x="" y="" width="" height=""/>

## 4.8:svg-开发流程-渐变对象

<defs> //定义特效对象:

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

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

</linearGradient>

</defs>

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

## 4.9:svg-开发流程-滤镜

滤镜(高斯模糊滤镜)

<defs>

<filter id="f3">

<feGaussianBlur stdDeviation="模糊度" />

</filter>

</defs>

<ANY filter="url(#f3)"></ANY>

<https://developer.mozilla.org/zh-CN/docs/Web/SVG/Element/filter>

今天学习的内容

1:复习上周知识重点

2:作业

-完善购物全选按钮

(1)为全选复选框绑定变量

<input type="checkbox" :checked="allcb" />

data(){ allcb=false}

(2) selectAll(e){

var cb = e.target.checked;

this.allcb = cb;//修改双向绑定

}

(3)商品复选框前添加点击事件

累加购物车列表中选中商品数量 3 allcb=true

小于 3 allcb=false

-用户登录:讲解概念

(1)服务器端 node.js 添加第三方模块 express-session

为session配置

(2)脚手架main.js 配置ajax跨域访问保存session

axios.defaults.withCredentials=true;

(3)程序流程(node.js)

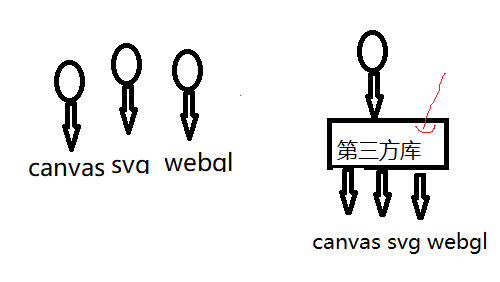
-用户登录成功 req.session.uid = 2;

-其它功能 var uid = req.session.uid;

# 5:html5 第三方绘图库--(重点)

https://d3js.org/ d3开源免费库

https://echarts.baidu.com/ 百度开源免费库



## 5.1:第三方绘图库-echarts

-下载并且在程序引入 echarts.js 文件

-创建图形容器 <div id="main" style="width;height"></div>

-通过js 程序获取容器并且创建echarts对象

var main = document.getElementById("main");

var mychart = echarts.init(main)

-创建配置项

var option = {

title:{text:"echart入门示例"},

xAxis:{data:["衬衫","袜子","雪纺衫"]},

yAxis:{},

series:[{type:"bar",data:[100,1,3]}]

}

-将配置项添加echarts对象中

mychart.setOption(option);

## 5.2:第三方绘图库-折线图

series:[{type:"line"}]

## 5.3:第三方绘图库-仪表图

series:[{

type:"gauge",

detail:{formatter:'{value}%'}

data:[{value:32,name:"任务完成率"}]

}]

## 5.4:第三方绘图库-饼图

series:[{

type:"pie",

radius:"50%", 半径

center:["50%","50%"], 圆心[水平,垂直]

data:[{value:325,name:"邮件营销"},{value:129,name:"搜索"}]

}]

练习1:医疗行业销售统图

-药品名称 1 2 3 4 5 6 ..12 年份 生产日期 保质

-六味地黄丸 2 1 2018 2017 1

功能一:创建药品表

创建药品表 创建销售表

id name ctime rtime id yid m[月] sales[销售量] y

1 六味 2017 1 1 1 1 2 2018

2 五味 2016 2 2 1 2 1 2018

db.sql

# 6:html5新特性--地理定位

geolocation:地理定位使用js获取当前浏览器所处地理坐标

(经度;纬度;海拔;速度)数据,用于实例 LBS Location Base Service

基于位置服务:饿了么,滴滴打车..

手机浏览器如何获取定位信息

(1)首选手机GPS芯片与网络连接定位精度在米

(2)次选手机通基站

HTML5中提供一个新对象，用于获取当前浏览器定位信息

window.navigator.geolocation{

getCurrentPostion:fn 获取当前定位

watchPostion:fn 监听位置变化

clearPostion:fn 取消监听

}

国内三家地理定位服务商

(1)百度地图

(2)高德

(3)腾讯地图

## 6.1:地理定位--百度地图

-注册百度开发者帐户 (手机)

http://lbsyun.baidu.com/

-百度分配 AccessKey 访问密钥

u70A5pnNrRtRT1XAgwM5jL2YIVj1Gv97

-在自己网站中嵌入百度地图 API

<script src="http://api.map.baidu.com/api?v=2.0&ak=你的密钥"></script>

<div id="container"></div>

<script>

1:创建地图对象

var map = new BMap.Map("container")

2:创建坐标点

var point = new BMap.Point(经度,纬度);

3:初始化地图并且指定地图显示级别 1-19

map.centerAndZoom(point,15)

</script>

## 6.2:html5新特性--拖放API

Drag & Drop 拖动和释放

HTML5为拖放操作提供7个事件,分为两组

拖动源对象(会动)

dragstart 拖动开始

drag 拖动中

dragend 拖动结束

整个过程:dragstart\*1+drag\*n+dragend\*1

拖动目标对象(不会动)

dragenter 拖动进入

dragover 拖动悬停

dragleave 拖动离开

drop 拖动释放

整个过程1:dragenter\*1+dragover\*n+dragleave\*1

整个过程2:dragenter\*1+dragover\*n+drop\*1

注意事项:dragover事件有默认行为悬停结束后立即

触发离开事件

解决问题:阻止dragover事件默认行为

练习:实现一个拖动删除的效果

今天学习的内容

1:复习昨天的内容

2:作业

2.1:完善用户登录

用户登录操作:

-在脚手架登录陆组件表单,让用户输入用户名和密码

点击登录按钮后在服务器验证,如果验证成功

登录自动跳转Home组件,在服务器记录用户登录成功

-session 对象生存周期

当用户登录成功后,创建session对象保存当前用户.

登录成功后将uid保存session对象

当我们查询购物车,获取session对象uid

当我们关闭浏览器 session对象消失

2.2:如果实现session存储uid并且其它功能使用uid功能

node.js

(1)验证跨域配置正常

const cors = require("cors");

app.use(cors({

origin:["http://127.0.0.1:8080",

"http://localhost:8080"],

credentials:true

}));

(2)下载第三方模块 express-session并且对其配置

const session = require("express-session");

app.use(session({

secret:"128位随机字符", //自定义安全字符串

resave:false, //每次请求都需要更新session

saveUninitialized:true, //初始化是否保存数据

cookie:{

maxAge:1000\*60\*60\*8 //依靠cookie保存8小时

}

}))

(3)下面操作***顺序***

3.1:先完成用户登录 req.session.uid = 2

3.2:再完成其它功能 "购物车列表".. var uid = req.session.uid;

#所有uid 参数不需从脚手架发送,服务器端己经保存uid

3.3:添加购物车 req.session.uid

脚手架

main.js 配置跨域访问保存session即可

axios.defaults.withCredentials=true;

其它功能一样

--常见错误

experss-session deprecated req.secret ..

secret 结果单词..

--常见错误:顺序错误

2.3:拖动删除小飞机

# 7:html5新特性---web Worker

程序:指可以被CPU执行代码存储磁盘中 1.js 1.html

进程:指程序被OS调用内存中并且分配执行空间

线程:线程是进程内部执行单位

chrome浏览器线程模型

一个chrome进程内至少有6个线程，可以向web服务器发起

请求 -- 资源请求线程

还有一个线程负责将所有资源绘制浏览器上并且执行 js程序

--UI主线程

<button></button>

<script src="1.js"></script>

<button></button>

现象:js扩行过程中，按钮1可见按钮2不可见

原因:浏览器执行js 的绘制按钮同一个线程

解决方法:创建一个新线程由它执行耗时js任务,UI线程

负责网页渲染.

<button></button>

<script>

new Worker("1.js");

#以上代码完成二个任务

#(1)创建Worker对象

#(2)创建新线程执行1.js

</script>

<button></button>

注意事项1:Worker不能双击网页运行,报错

Failed to construct 'Worker'

解决问题:必须在网络情况下运行 http://127..

注意事项2:浏览器不允许Worker执行任何DOM/BOM

对象.

原因:浏览器只允许UI主线程操作DOM/BOM,类似JQUERY

程序不能在Worker执行.

## 7.1: 数据传递

-Worker线程可以发送消息给UI线程

--Worker发送消息

postMessage(stringMsg)

--UI 接收消息

var w = new Worker("1.js");

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

-UI线程可以发送消息发Worker线程

--UI发送消息

var w = new Worker("1.js");

w.postMessage(stringMsg)

--Worker 接收

onmessage = function(e){e.data}

练习:1:在网页创建input 请输入数值 100

2:在 input创建按钮 [开始计算累加]

3:在按钮下 <div>计算结果:5050</div>

4:我们希望通过Worker完成计算工作

# 8:web Storage

在浏览器中存储当前用户专有数据:购物车,定制样式

在客户端存储数据可以使用技术

(1)cookie技术:浏览器兼容性好,不能超过4kb,操作复杂

(2)Flash存储:依赖于Flash播放器

(3)H5 WebStorage:不能超过8MB,操作简单

(4)IndexDB:可能存储大量数据，还不是标准技术

|  |
| --- |
| Session 会话  “操作过程"  浏览器打开某一个网站*第一个页面*(会话开始)中间可能打开多个网页(会话中)直到关闭浏览器(会话结束)  整个过程称为"浏览器与 web服务器一次会话" |

WebStorage技术提供二个对象存储数据

-sessionStorage

将数据保存sessionStorage会话开始保存数据,同一个会话中

其它网页可以获取sessionStorage数据,一旦关闭浏览器(会话结束)

sessionStorage数据清空对象失效

作用:同一个会话中所有页面共享数据

示例:登录用户名,昵称

-sessionStorage[key] = value; 保存数据

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

-var value = sessionStorage[key]; 获取数据value

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

-sessionStorage.removeItem(key); 删除指定key/value

-sessionStorage.clear(); 清除数据

-sessionStorage.length; 数据个数

-var key = sessionStorage.key(i); 获取数据 key

练习1:03.html 将数据保存sessionStorage

04.html 获取sessionStorage

练习2:完成用户登录判断

index.html 右上角提示 "请登录" 超链接

login.html 输入用户名,密码 提交按钮

"登录成功,3s后自动跳转首页"

setTimeout+location.href

index.html 右上角 "欢迎回来 xxx 退出登录"

logout.html 提示 "您己退出 3s自动跳转首页"

index.html 右上角 "请登录" 超链接

# 9:localStorage

本地存储(跨会话级别存储)--永久保存

作用:购物车

- localStorage [key] = value; 保存数据

- localStorage.setItem(key,value); 保存/修改数据

-var value = localStorage [key]; 获取数据value

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

- localStorage.removeItem(key); 删除指定key/value

- localStorage.clear(); 清除数据

- localStorage.length; 数据个数

-var key = localStorage.key(i); 获取数据 key

注意:

localStorage如果数据发生修改,触发一次window.onstorage

事件,可以监听此事件,实现监听localStorage数据改变的目标

不能监听 sessionStorage

练习1:保存和获取数据 08.html 09.html

练习2:用户自定义样式

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名称

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

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

.dark{background:#333,color:#eee};

用户选中某个项(select.onchange)当前body元素指定样式

this.value "blue"

document.body.className = "blue"

作业1:购物车数量 vuex

作业2:六味地黄丸

今天学习的内容

1:复习昨天重点

2:作业

-登录功能

-购物车数量完成 vuex

-刷新数据丢失解决sessionStorage保存数据

2.2:Vuex 是一个仓库,Vuex核心对象store就是容器

此容器存储所有组件共享数据

当某一个组件修改共享数据，Vuex 知所有使用数据组件更新数据

保证所有组件使用数据状态统一

2.3:Vuex 使用方式

-在main.js 下载引入并且创建配置vuex对象

(0)npm i vuex/node\_modules/...

(1)import Vuex from "vuex" //将vuex引入当前项目中

(2)Vue.use(Vuex) //将vuex注册Vue实例中

(3)var store = new Vuex.Store({ //创建Vuex实例对象

state:{},

mutations:{}

getter:{}

});

state: 存储全局共享数据

mutations: 修改全局共享数据方法

getter: 获取全局共享数据方法

(4)new Vue({store })

(5)其它组件操作全局共享数据

修改全局共享数据: js this.$store.commit("修改数据方法名")

获取全局共享数据:<template> {{$store.getter.获取数据方法名}}

示例:将购物车中数据添加全局共享数据中

state:{cartCount:0}

mutations:{

increment(state){

state.cartCount++;

}

},

getters:{

getCartCount(state){

return state.cartCount;

}

}

this.$store.commit("increment");

App.vue {{$store.getters. getCartCount}}

#依据购物车中商品数据修改全局共享数据值

#(1)当用户查询购物车 更新全局共享数据值

# 10:html5新特性--WebSocket

http网络协议:

协议网络工作标准依靠软件实例(node.js服务器/浏览器)

http:工作模型:请求-响应,只有客户端先发起请求,服务器才会返回

响应消息,没有请求就没有响应

http作用:传输网页中资源(html;css;js;img;avi;mp3....)

俗称http网页搬运工

http协议在有一些应用中不适合, 股票实时走势图

解决问题:

webSocket协议:属于"广播-收听"

客户端连接到服务器上就不再断开永久连接,双方都可以

随时向对方发送消息.

ws:适合:股票走势图,在线聊天室

## 10.1:WebSocket--实现此功能

-ws 服务器[java/php/node.js]

--02\_ws\_server.js websocket服务器

--需要下载第三方模块 [ws][async-limiter]

指定监听端口 9001

接收客户请求

向对方发送消息

接收对方消息

-ws 客户端

html5中新对象 WebSocket 发请求 发消息 接收消息

--创建webSocket对象[创建对象;发请求建立连接]

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

--向服务器发送消息

socket.send(stringMsg);

--接收服器消息

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

--关闭连接

socket.close();

4:大鱼和小鱼[canvas]

分析游戏:

大鱼游动时可以吃食物，如果吃到蓝色食物 100 分 吃橙色食200分

大鱼喂小鱼

4.1:游戏目录结构

game -- 游戏所有文件都保存在此处

index.html 唯一html

src 保存游戏中的图片

大鱼眼睛: bigEye0.png bigEye1.png

大鱼身体: bigSwim0.png ... bigSwim7.png

大鱼尾巴: bigTail0.png .. bigTail7.png

蓝色食物: blue.png

橙色食物: fruit.png

大海背景: background.jpg

js 保存游戏中的程序

main.js main.js游戏入口

commonFunctions.js 公共函数库文件

background.js 游戏背景

mom.js 大鱼

baby.js 小鱼

ane.js 海葵

friut.js 食物

....

4.2:游戏唯一index.html

创建二个画布大小一相同一个在前一个在后 800\*600

前面画布:大鱼;小鱼;分数... z-index:1

后面面布:背景;海葵;食物... z-index:0

加载所有 commonFunction.js main.js background.js

复制图片 src

4.3:创建游戏程序入口main.js

创建游戏所有需要用到对象(背景;食物;大鱼;..)

将所有对象画在画布上

4.4:大海背景图

src/background.jpg 绘制第二画布

1:main.js 声明全局变量保存图片

var bgPic;

2:init 创建图片对象下载图片

bgPic = new Image();

bgPic.src = "src/background.jpg"

3:将绘制图片放在函数 background.js

drawBackground(){

ctx2.drawImage(bgPic,0,0)

}

4:创建定时器 100 绘制画布上所有元素

定时器:setInterval(fn,100);

智能定时器:requestAnimationFrame(fn)

兼容性差:解决问题 commontFunctions.js

自定义兼容性很好函数:requestAnimFrame(fn)

main.js gameloop

function gameloop(){

requestAnimFrame(gameloop);

drawBackground();

}

常见错误:

(1) background.jpg:1 404 (Not Found)

原因:图片路径不正确

图片不存在

4.5:海葵

一共50根海葵

海葵基本 高度 200

海葵与海葵之间水平间距 16

始点坐标x+16+random

终点坐标

在js目录创建文件 ane.js