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

3:今天的目标

3.1:html5新特性--localStorage 示例

练习:创建usercenter.html 用户中心

index.html -> usercenter.html 应用刚才定义样式

-创建usercenter.html

--添加一组样式

--获取样式 localStorage 应用当前 body

-如果index.html 修改样式usercenter.html 立即更新样式

3.3: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

3.4:html5新特性--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

作业:

2:时机不正确,登录成功立即更新购物数据

3:刷新全局丢失