**Mint-ui**

Day01

今天学习的内容

程涛:taonetwork2018 微信

1：第四阶段课程介绍

(1)vue 移动端组件库和项目(学子商城)

element 团队 mintui 6 day

(2)html5新特性7day

视频处理功能

网页绘制功能

(3)微信公众平台-小程序(云开发) 6day

(4)混编

(web/android/ios/小程序/支付宝/今日头条/..) 1day

2: vue ui 组件库

vue 前端开发框架，刚开时针对象PC用户，移动端

用户数量非常庞大,饿了么

(MinitUI移动端组件库/ElementUI pc端组件库)

,滴滴团队,MUI......

为什么使用组件库开发项目:原因提高开发效率

问题:开发轮播图/底部导航条....

2.1:补充知识点:pc端项目与移动端项目

(1)屏幕宽度

pc端: w>992px

移动端:w<992px

(2)操作方法

pc端:键盘,鼠标

移动:手指

(3)事件类型

pc端:click/keyup/keydown/

移动端:touchstrat/touchmove/touchend/...

2.3:mintui组件的使用方法

下载安装二种方法

(1)方式一:(学习)

mint-ui官方网站下载 css/js/fonts 文件加载html中

<http://mint-ui.github.io/#!/zh-cn>

(2)方式二:(项目 vue cli 脚手架)

-准备脚手架

-脚手架根目录下载

-npm i mint-ui -S

#npm 第三方模块管理工具

#i install 安装参数简写

#mint-ui 组件库名

#-S -save-dev 保存并开发将mint-ui添加配置文件

#验证:node\_modules/mint-ui/lib/...

-启动项目

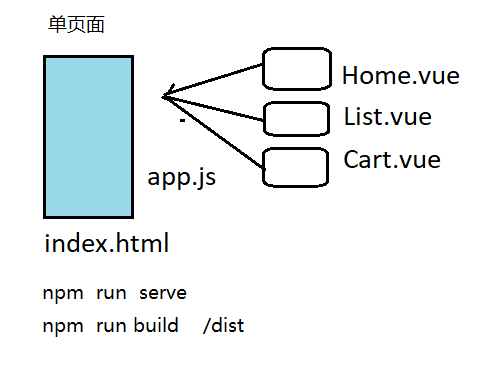
npm run serve

#解决问题:开发web项目时是否一定需要脚手架

(1)使用脚手架开发项目企业必备项目

原因:现在web项目第三方库/资源太多

单页面项目



2.4:mintui组件的使用方法--在项目中引入mintui

(1)引入方式:按需引入

(2)引入方式:完整引入(正确) main.js

import MintUI from "mint-ui" #完整引入组件库(组件)

#MintUI组件对象名称

#mint-ui库名称

import "mint-ui/lib/style.css" #单独引入样式文件

(3)将MintUI注册给Vue实例

Vue.use(MintUI)

解决脚手架问题:

(1)启动指令

npm run dev //dev 启动指令可能修改

npm run serve //serve 启动指令可能修改

脚手架:package.json

(2)启动位置

当你执行启动脚手架，当前目录存在package.json

vue\_app\_00> npm run serve

脚手架启动成功后，打开浏览器访问项目

http://127.0.0.1:8080/#/

2.5:mintui组件的使用方法--第一个组件Toast

目标:显示一个短消息提示框

作用:删除成功,登录成功,用户名有误...

替换pc

基本语法: this.$toast({message:"提示文字"})

参数:

message:提示消息

postion:位置 top,bottom,middle

duration:持续时间(单位毫秒)默认 3000

iconClass:图标类名

2.6:mintui组件的使用方法--创建自定义组件

(1)创建空组件

src/components/exam/Exam01.vue

(2)为空组件指定访问路径

src/router.js 指定自定义组件访问路径'/Exam01'

(3)在浏览器地址栏中访问

http://127.0.0.1:8080/#/Exam01

(4)添加第三方mint-ui组件库中第一个组件toast

2.7提示框图标出现问题

原因:默认提示框图标外观差,不能满足项目要求

解决方案:使专业图标库下载

https://www.iconfont.cn/

#如果在项目使用图标下载纯色图标

#下载文件download.zip 图标字体

在脚手架添加字体图标

(1)在src下创建目录 font 将下载所有内容保存在处

(2)main.js 引入字体图标css文件

import "./font/iconfont.css"

(3)this.$toast({

iconClass:"iconfont icon-home"

});

Day02

今天学习的内容

1:复习上周重点

2:今天的目标

2.1:mint-ui组件库-表单组件

-(1)表单:开关组件:

作用:打开或者关闭程序中某项功能

<mt-switch v-model="val1" @change=""></mt-switch>

属性:

val1值为true 打开状态

val2值为false 关闭状态

事件:

change:修改

-(2)表单:单选框列表组件

<mt-radio

title=""

:options="['a','b','c']"

v-model="val2"

</mt-radio>

属性:title 标题

options 选项:可以使用数组来绑定数据

格式:[{label:"",value:""}]

#单选列表显示时出现 label

#获取值才能获取value内容

#value字符串类型

v-model 绑定变量值

-(3)表单:复选框列表组件

<mt-checklist

title="标题"

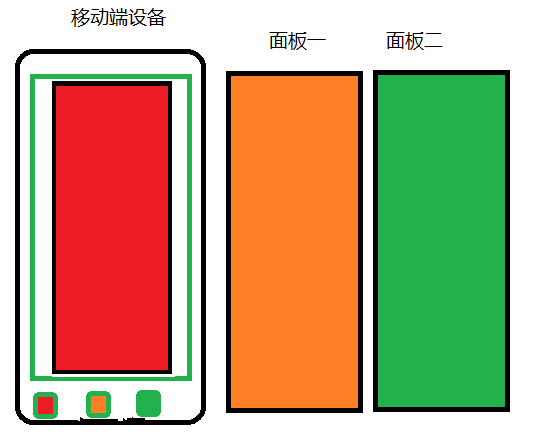
v-model="val3"

options="['选项a','选项b','选项c']">

</mt-checklist>

#注意:val3数组

2.2:mint-ui组件库-面板--(重点)



#多个面板显示不同的内容，同一个时刻只能显示

#一个面板

<mt-tab-container v-model="active"> #父元素

<mt-tab-container-item id="tab1"> #子面板一

</mt-tab-container-item>

<mt-tab-container-item id="tab2"> #子面板二

</mt-tab-container-item>

</mt-tab-container>

#当active变量保存tab1时则显示子面板一  
 #注意:组件id与html 元素id不是一致

#创建组件 src/components/exam/ExamContainer04.vue

#路径 /ExamContainer04

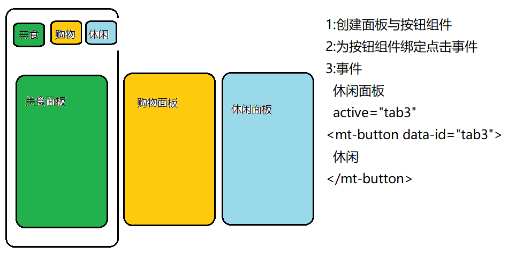
练习:创建三个按钮 "美食" "购物" "休闲"

并且三个按钮对应三个子面板

当用户点击某个按钮时对应子面板显示

#创建组件 src/components/exam/ExamContainer05.vue

#路径 /ExamContainer05



2.3:mint-ui组件库-面板--(底部导航栏)(重点)



底部导航条按钮

(1)图片

(2)文字

(3)当点击按钮时跳转组件

组件tabbar 底部导航条

<mt-tabbar v-model="active" fixed>导航条  
<mt-tab-item id="tab1">按钮1

<img src="" />#路径../../assets/01.png

文字

</mt-tab-item>

<mt-tab-item id="tab2">按钮2

<img src="" />

文字

</mt-tab-item>

</mt-tabbar>

#底部导航条中图标保存在脚手架本地即可

#active:当用户点击按钮时，将按钮id保存至

active变量中

# fixed:将导航条固定在组件底部

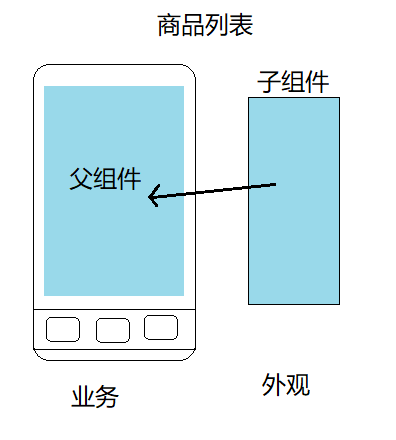
# slot="icon"插槽

#创建组件

src/components/exam/ExamTabbar06.vue

#路径 /ExamTabbar06

2.4:mint-ui组件库--子组件调用与参数传递



示例:子组件调用与参数传递

#子组件:负责显示数据与数据外观

#父组件:负责加载数据并且调用子组件

Sub07.vue子组件

<template>

<h3>{{msg}}</h3> #msg父组件传递数据

</template>

<script>

export default {

props:{ //声明接收父组件数据

msg:{default:""} //数据msg默认值""

}

}

</script>

Fa08.vue 父组件

(1)将子组件引入当前父组件中

import Sub07 from "./Sub07.vue"

(2)注册子组件并且给子组起一个名字

components:{

"字符串":子组件对象名称

"sub07":Sub07

}

(3)在模板中调用子组件

<template>

<sub07msg="测试数据123"></sub07>

</template>

作业1:底部导航条

作业2:顶部导航条



Day03

今天学习的内容

1:复习昨天的内容

2:今天的目标--微信消息列表(重点)



2.1:微信消息列表-分析功能(重点)

(1)第一部分:顶部导航条(完全自定义子组件)

(2)第二部分:主体消息列表(mint-ui面板组件+自定义)

(3)第三部分:底部导航条(mint-ui tabbar+自定义)

2.2:微信消息列表-目录结构(重点)



2.3:功能一:顶部导航条



-创建子组件common/TitleBar.vue 完成以上功能

-左侧文字与右侧两张图片是通过父组件Home.vue

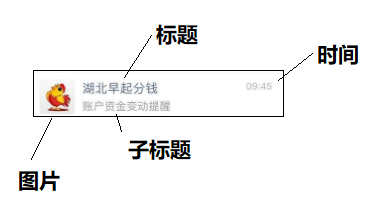
传入

-子组件负责接收父组件数据并且设计外观

-右侧两张图片触发点击事件处理函数也是由父元素

传入

2.4:功能二:一条消息组件(子组件)14:22



-创建子组件common/Message.vue 完成以上功能

-结构:左侧图片/标题/子标题/右侧时间

2.5:功能三:消息列表组件MessageList.vue(子组件)

Day04

今天学习的内容

1:复习昨天的重点

2:今天的目标

2.1:微信消息列表

-注意事项:为mint-ui组件tabbar子元素mt-tab-item

默认点击事件失效

原因:vue对子元素事件绑定有一定要求

如果父元素为子元素添加事件，需要添加修饰

符

native

示例:<mt-tab-item @click.native="事件处函数">

..</mt-tab-item>

-注意事项: 浏览器有内容样式文件,样式文件中内容

有一定差异

解析方式:覆盖方式来解决样式差异

Normalize.css/n.css

静态文件index.html 加载 App.vue 根组件

Home.vue 显示App.vue 组件

2.2:学子商城--mint-ui版本--用户登录

-分析功能

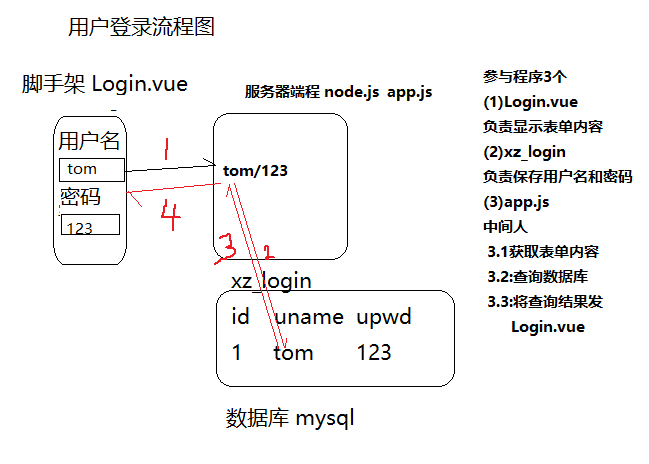
用户打开网页输入用户名和密码点击提交按钮

(1)格式出错提示:用户名格式不正确

(2)用户不存在提示:用户名或密码不正确

(3)正确跳转商品主页

-参与完成此功能程序有些各完成什么任务



2.3:学子商城--mint-ui版本--用户登录-数据库

-使用库 xz

-分析表 xz\_login

id INT PRIMARY KEY AUTO\_INCREMENT,

uname VARCHAR(50),

upwd VARCHAR(32)

-密码是一种特殊数据,此数据需要加密后再保存

-数据库保存加密码结果

-为什么需要加密保存,防止工作人员泄密

-数据库中函数md5()单向加密函数

-提高加密安全级别

(1)多次循环加密md5(md5(123))

(2)指定用户密码长度8以上[小写字母大写字母数字特殊]

合格密码:ab1A!#9

示例:加密文字

-添加二条测试数据

2.4:学子商城--mint-ui版本--用户登录-服务器程序

-功能分析

(1)获取网页中用户输入：用户名和密码

(2)创建sql语句查询数据中是否有匹配数据

(3)将查询结果返回网页

-功能一

获取网页中用户输入：用户名和密码

req.query.uname; 用户名 [tom]

req.query.upwd; 密码 [123]

-功能二

特殊数据:密码将用户输入密码加密

用户数据:md5(123)=>

202cb962ac59075b964b07152d234b70

数据库:202cb962ac59075b964b07152d234b70

SELECT id FROM xz\_login

WHERE uname= ? AND upwd=md5(?)

-功能三

pool.query(sql,["tom","123"],(err,result)=>{

//回调函数

(1)判断是否有错误产生如果有错误抛出

if(err)throw err;

(2)判断用户登录成功select result[]

result.length==0 登录失败

(3)将判断结果发送网页

res.send({code:-1,msg:"用户名或密码有误，请重试"});

res.send({code:1,msg:"登录成功"});

})

2.5:学子商城--mint-ui版本--app.js

(1)下载加载第三方模块

(!!!!此模块与脚手架模块完全不一样不要混用)

express web服务器

mysql mysql驱动

cors 跨域脚手架8080 app.js 3000

express-session session

下载 npm i mysql express express-session cors

(2)前端脚手架完成用户登录

src/components/xz/Login.vue /Login

-创建输入框/用户名/密码/登录按钮

-为登录按钮绑定点击

(1)获取用户输入用户名和密码

(2)创建正则表达式验证用户名和密码

如果验证不成功提示消息

"用户名格式不正确"

或 "密码格式不正确"

(3)发送ajax请求

http://127.0.0.1:3000/login?uname=tom&upwd=123

(4)获取服务器返回值提示用户登录成功

2.6:学子商城--mint-ui版本--axios

项目中经常需要发送ajax请求

(1)jquery.js

(2)axios ajax 库

(3)request ajax 库

问题1:项目中请求服务器路径会经常频繁改动

http://soso.com:3001/login...

解决:使用配置项解决问题(base基础) main.js

axios.defaults.baseURL = "http://127.0.0.1:3000/"

示例: this.axios("login")

问题2:发送请求需要session支持，默认情况

session数据丢失[配置项作用保存session值]

解决:使用配置项

axios.defaults.withCredentials=true

常见错误

(1) 404 (Not Found)请求程序或者资源不存在

Login.vue this.axios.get("login")

app.js server.get("/login")

(2) net::ERR\_CONNECTION\_RESET网络连接出错

服务器程序app.js 出错并且停止工作

查看node app.js 出错信息

(3) Error: connect ECONNREFUSED 127.0.0.1:3306

服务器错误:3306==mysql

1.1 mysql没有启动

1.2 mysql 启动出错(少见)

(4) Unknown column 'uname1' in 'where clause'

未知列 uname1 原因:sql语句拼写错误

(5)代码语法正确结果错误

tom/123 用户名和密码有误

解决方案:F12->控制台->network(记录网络信息)

记录请求次数和请求内容包括参数

标签:[Header/Response]

(6)跨域

No 'Access-Control-Allow-Origin

解决:app.js 跨域配置错误

作业:

(1)完成学子商城登录组件 Login.vue

(2)完成商品首页 Product.vue

商品内容分页显示

Day05

今天学习的内容

1:复习昨天的重点

2:今天的目标

2.1:微信消息列表

-注意事项:为mint-ui组件tabbar子元素mt-tab-item

默认点击事件失效

原因:vue对子元素事件绑定有一定要求

如果父元素为子元素添加事件，需要添加修饰

符

native

示例:<mt-tab-item @click.native="事件处函数">

..</mt-tab-item>

-注意事项: 浏览器有内容样式文件,样式文件中内容

有一定差异

解析方式:覆盖方式来解决样式差异

Normalize.css/n.css

静态文件index.html 加载 App.vue 根组件

Home.vue 显示App.vue 组件

2.2:学子商城--mint-ui版本--用户登录

-分析功能

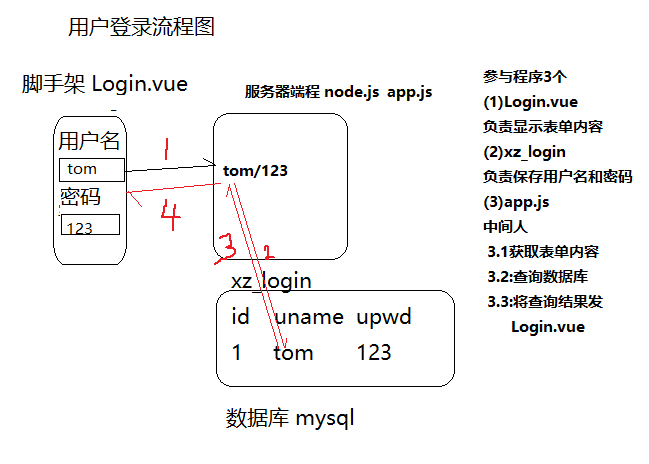
用户打开网页输入用户名和密码点击提交按钮

(1)格式出错提示:用户名格式不正确

(2)用户不存在提示:用户名或密码不正确

(3)正确跳转商品主页

-参与完成此功能程序有些各完成什么任务



2.3:学子商城--mint-ui版本--用户登录-数据库

-使用库 xz

-分析表 xz\_login

id INT PRIMARY KEY AUTO\_INCREMENT,

uname VARCHAR(50),

upwd VARCHAR(32)

-密码是一种特殊数据,此数据需要加密后再保存

-数据库保存加密码结果

-为什么需要加密保存,防止工作人员泄密

-数据库中函数md5()单向加密函数

-提高加密安全级别

(1)多次循环加密md5(md5(123))

(2)指定用户密码长度8以上[小写字母大写字母数字特殊]

合格密码:ab1A!#9

示例:加密文字

-添加二条测试数据

2.4:学子商城--mint-ui版本--用户登录-服务器程序

-功能分析

(1)获取网页中用户输入：用户名和密码

(2)创建sql语句查询数据中是否有匹配数据

(3)将查询结果返回网页

-功能一

获取网页中用户输入：用户名和密码

req.query.uname; 用户名 [tom]

req.query.upwd; 密码 [123]

-功能二

特殊数据:密码将用户输入密码加密

用户数据:md5(123)=>

202cb962ac59075b964b07152d234b70

数据库:202cb962ac59075b964b07152d234b70

SELECT id FROM xz\_login

WHERE uname= ? AND upwd=md5(?)

-功能三

pool.query(sql,["tom","123"],(err,result)=>{

//回调函数

(1)判断是否有错误产生如果有错误抛出

if(err)throw err;

(2)判断用户登录成功select result[]

result.length==0 登录失败

(3)将判断结果发送网页

res.send({code:-1,msg:"用户名或密码有误，请重试"});

res.send({code:1,msg:"登录成功"});

})

2.5:学子商城--mint-ui版本--app.js

(1)下载加载第三方模块

(!!!!此模块与脚手架模块完全不一样不要混用)

express web服务器

mysql mysql驱动

cors 跨域脚手架8080 app.js 3000

express-session session

下载 npm i mysql express express-session cors

(2)前端脚手架完成用户登录

src/components/xz/Login.vue /Login

-创建输入框/用户名/密码/登录按钮

-为登录按钮绑定点击

(1)获取用户输入用户名和密码

(2)创建正则表达式验证用户名和密码

如果验证不成功提示消息

"用户名格式不正确"

或 "密码格式不正确"

(3)发送ajax请求

http://127.0.0.1:3000/login?uname=tom&upwd=123

(4)获取服务器返回值提示用户登录成功

2.6:学子商城--mint-ui版本--axios

项目中经常需要发送ajax请求

(1)jquery.js

(2)axios ajax 库

(3)request ajax 库

问题1:项目中请求服务器路径会经常频繁改动

http://soso.com:3001/login...

解决:使用配置项解决问题(base基础) main.js

axios.defaults.baseURL = "http://127.0.0.1:3000/"

示例: this.axios("login")

问题2:发送请求需要session支持，默认情况

session数据丢失[配置项作用保存session值]

解决:使用配置项

axios.defaults.withCredentials=true

常见错误

(1) 404 (Not Found)请求程序或者资源不存在

Login.vue this.axios.get("login")

app.js server.get("/login")

(2) net::ERR\_CONNECTION\_RESET网络连接出错

服务器程序app.js 出错并且停止工作

查看node app.js 出错信息

(3) Error: connect ECONNREFUSED 127.0.0.1:3306

服务器错误:3306==mysql

1.1 mysql没有启动

1.2 mysql 启动出错(少见)

(4) Unknown column 'uname1' in 'where clause'

未知列 uname1 原因:sql语句拼写错误

(5)代码语法正确结果错误

tom/123 用户名和密码有误

解决方案:F12->控制台->network(记录网络信息)

记录请求次数和请求内容包括参数

标签:[Header/Response]

(6)跨域

No 'Access-Control-Allow-Origin

解决:app.js 跨域配置错误

作业:

(1)完成学子商城登录组件 Login.vue

(2)完成商品首页 Product.vue

商品内容分页显示

Day06

今天学习的内容

1:复习昨天的内容

2:今天的目标

2.1:session(重点)

-session(会话) 用户一次操作过程

(1)会话开始当用户打开某个网站第一个网页

(2)会话中中间打开多个网页

(3)会话结束关闭浏览器

-session对象

session对象在服务器端保存此次会话中多个程序

所共享数据

(1)在当前会话中程序可以获取session对象数据

(2)当会话结束session对象数据丢失

-什么样式数据保存session对象

(1)将用户登录凭证uid保存session对象中

(2)将用户权限保存session对象中

-如何完成session对象保存获取

(1)服务器端程序支持

express-session 下载配置

-将数据保存session对象

server.get("/login",(req,res)=>{

....

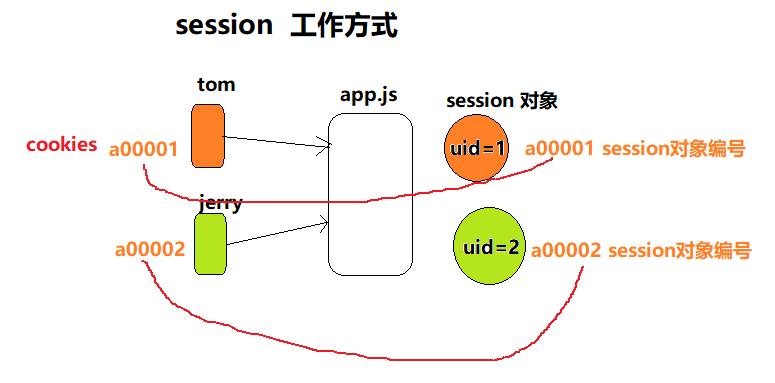
req.session.uid = 1;//登录凭证保存session

})  
 -从session对象中获取数据,添加购物车

var uid = req.session.uid;

(2)客户端脚手架配置axios发送请求保存session数据

-画一张图 session工作流程



2.2:添加购物车

分析功能:操作顺序先用户登录成功后可以

将商品添加至购物车

(1)数据库表xz\_cart(id,lid,price,count,lname,uid)

(2)app.js

-请求方式 GET请求地址 /addcart

-参数 lid,price,lname

-sql

\*查询用户是否购买过此商品

SELECT id FROM xz\_cart WHERE uid = ? AND lid=?

\*添加此商品

INSERT INTO xz\_cart VALUES(null,?,?,?,?)

\*更新此商品购买数量

UPDATE xz\_cart SET count=count+1

WHERE uid = ? AND lid = ?

-json

(3)Product.vue

<mt-button @click="addCart"

:data-lid="item.lid" #自定义属性

:data-price="item.price"

:data-lname="item.lname">

加入购物车</mt-button>

addCart(e){

(1)获取三个商品信息 lid lname price

var lid = e.target.dataset.lid;

var p = e.target.dataset.price;

var n = e.target.dataset.lname;

(2)发送ajax /addcart

(3)接收返回数据

}

常见错误

(1)无法访问此网站服务器程序出错 app.js

解决问题:查看app.js 出错信息

(2) Table 'xz.xz\_cart' doesn't exist购物车表不存在

解决问题:下载db.sql 复制指令在自己数据库执行

(3) Unknown column 'phone' in 'field list'未知列 phone

解决问题:将sql商品名称添加单引号 '${lname}'

(4)Column count doesn't match value count at row

解决问题:sql添加sql表5列= 4列

2.3:登录成功跳转商品列表

vue 当一个线件执行完成需要自动跳转其它组件

this.$router.push("其它组件路径");

示例:

(1)当用户登录成功跳转商品列表组件

this.$router.push("/Product");

2.4:购物车(重点)

分析:

(1)查询当前登录用户购物中所有内容

(2)删除指定商品

(3)删除选中商品(清空购物车)

(4)全选

2.5:购物车功能一:查询当前登录用户购物中所有内容

(1)数据库:xz\_cart

[id/lid/price/lname/count/uid]

添加一列 img\_url 01.jpg/02.jpg

(2)服务器端程序:app.js

-请求方式 GET请求地址 /cart

-参数(无)

-sql SELECT img\_url,id,lid,lname,price,count

FROM xz\_cart WHERE uid = ?

-json {code:1,msg:"查询成功",data:[..]}

(3)脚手架组件Cart.vue

-创建空组件 xz/Cart.vue

-分配访问路径 /Cart

-当组件创建成功后发送ajax请求

获取购物车列表

-将购物车列保存data:list[]

-创建标签添加样式创建循环

2.6:购物车功能二:删除指定商品

-服务器

参数 id

sql DELETE FROM xz\_cart WHERE id = ?

json {code:1,msg:"删除成功"}

-脚手架

-为删除按钮绑定点击事件

<mt-button :data-id="item.id"@click="delItem">

删除</mt-button>

delItem(e){ var id = e.target.dataset.id}

-交互提示用户是否删除指定商品

-同意

-发送ajax 服务器端程序 ?id=id

-返回服务器返回内容

-重新调用 loadMore()最新购物列表查询

HTML5

Day01

今天学习的内容

1:复习上周重点

2:今天的目标

2.1:学子商城--购物车(删除/全选)

-删除用户指定的一个商品流程图



-分析功能

(1)用户点击购物车中某个商品删除按钮

(2)显示交互式提示框(防止用户误操作)

(3)如果用户选择"确定"

(4)发送ajax请求并且获取服务器返回结果

(5)如果删除成功,重新获取新购物车中内容

-特殊注意事件

(1)删除是一种有一定风险操作,原因数据如果

被删除**很难**恢复.

(2)如果有重要备份之后再删除

2.3:学子商城--购物车-删除一个商品(操作)

-数据库表

xz\_cart(id/lid/img\_url/price/count/uid/lname)

-app.js

(1)参数购物车id

(2)sql DELETE FROM xz\_cart WHERE id = ?

(3)json {code:1,msg:"删除成功"}

-Cart.vue

(1)为"删除"按钮绑定点击事件

(2)获取购物车id

<mt-button @click="delItem"

:data-id="item.id">删除</mt-button>

delItem(e){ var id = e.target.dataset.id}

(3)显示交互式提示框,请用户再次确认是否删除

this.$messagebox.confirm("是否删除指定数据")

.then(res=>{//用户选择确定(4)(5)})

.catch(err=>{//用户选择取消})

(4)发送ajax /cart?id=1

(5)删除成功后获取返回结果,重新获取购物车列表

2.4:学子商城--购物车-删除多个商品(操作)



(1)xz\_cart

[id/lid/price/lname/....]

(2)app.js

参数: ids=2,3

sql:DELETE FROM xz\_cart id IN (2,3)

(3)Cart.vue

2.5:学子商城--购物车-全选

<input type="checkbox" @change="selectAll"/>

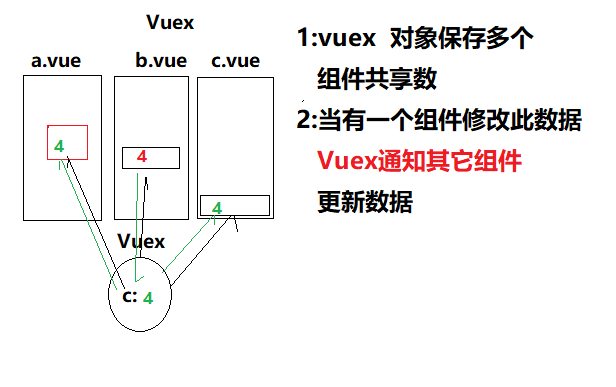
(1)全选按钮'选中'其它商品'选中'

全选按钮'取消'其它商品'取消'

(2)获取全选按钮状态truechecked表示状态

(3)list:[{cb:true}]

2.6:Vuex



Vuex是一个专为Vue应用程序的状态管理模式,

它采用集中式存储管理程序中所有组件状态

并以相应的规则保存状态以一种**同步方式**保证状态

统一

#Vuex 本身容器保存多个组件中需要*共享*的数据(状态)

#默认情况Vue组件自己管理自己数据(状态)

#data(){}

#A.vue data(){} B.vue data(){}

#有时数据需要多个组件之间共享(购物车中商品数量)

示例:

(1)将购物车中数量保存Vuex对象中(集中管理)

(2)Cart.vue 获取购物车数量

(3)Cart.vue 修改购物车中数量

2.7:Vuex 使用方式

(1)下载Vuex第三方模式

-脚手架下载 npm i vuex - S

#下载完成node\_modules/vuex/...

(2)在main.js 对Vuex配置

-引入Vuex 第三方模块

import Vuex from "vuex"

-创建存储实例对象!!!!

var store = new Vuex.Store({

state:{cartCount:0},

mutations:{},

getters:{},

actions:{}

});

#state: 保存多个组件共享数据

#mutations: 添加***修改***共享数据函数

var i = 0; i++; vuex不能使用运算符直修改错

#getters: 添加函数通过函数获取共享数据

#actions: 添加函数通过函数调用使用异步操作?

-注册Vuex

Vue.use(Vuex)

-将实例对象直接添加Vue对象中

new Vue({store}) #store存储实例对象

|  |
| --- |
| 示例:创建存储实例对象,保存购物车数量  var store = new Vuex.Store({  state:{  cartCount:0 //购物车中商品数量  },  mutations:{  //购物车商品数量减1  subItem(state){state.cartCount--},  //购物车商品数量加1  addItem(state){state.cartCount++},  //清空购物车量  clearItem(state){state.cartCount=0}  }  getters:{  //获取购物车数量  getCartCount(state){ return state.cartCount}  }  actions:{}????  }) |

(3)在需要使用Vuex数据组件调用

-修改共享数据调用mutations中添加函数

<mt-button @click="clearCart">清空购物车</mt-button>

clearCart(){

#清空数量

this.$store.commit("clearItem")

}

-获取共享数据

<img></img>{{$store.getters.getCartCount}}

2.x:html5新特性

Day02

今天学习的内容

1:复习昨天重点

2:今天的目标

2.1:html5新特性

新特性:重点(视频/canvas绘图/拖放API/WebSocket)

2.2:html5新特性-视频

基础知识

(1)视频文件格式: .mp4 .flv .webm .ogg ...

(2)解码器:如果浏览器播放指定格式视频需要安装

此文件格式视频对应解码器(软件)

解决方案:"格式工厂" x.mp4 ->x.flv->x.webm

标准语法

简写: <video src="x.mp4">

您的浏览器版本太低，请升级!

</video>

标准:兼容处理

<video>

<source src="x.mp4" />

<source src="x.flv" />

<source src="x.webm" />

浏览器版本太低,请升级!!

</video>

2.3:html5新特性-视频-属性

-常见属性

controls 显示播放原生控件(兼容性非常差)

autoplay 自动播放视频(大多数浏览器此功能禁用)

loop循环播放

muted静音播放

poster在播放视频之前显示图片(广告)

如果视频暂停图片不能显示

preload预加载策略(方式)

none:不预加载任何数据

metadata:只预加载元数据

(视频时长,视频第一帧画面,视频高宽)

auto:预加载元数据和一定时长视频(默认)

-js中常见属性和方法与事件

#必须通过js程序获取"视频对象"才能执行属性方法

volume:1 音量(0~1)

playbackRate: 播放速率(回放速度)

(整型数)大于1 快放小于1慢播

play() 播放视频

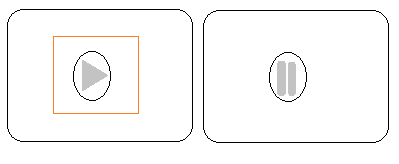
pause() 暂停视频播放

paused 表示当前视频播放状态

true 表示视频暂停

false 表示视频播放

练习:通过自定义播放控件实现视频播放和暂停



按钮资源res/play.png res/pause.png

广告图片 res/p3.png

(1)为视频绑定鼠标移入离开事件

鼠标移入视频区域显示图片res/play.png

鼠标离开视频区域隐藏图片

(2)为图片绑定点击事件

播放视频与暂停播放视频切换

(3)视频暂停播放显示图片广告,如果播放播放隐藏

图片广告

2.4:html5新特性-视频-特殊事件

-canplaythrough 当前视频加载结束可以播放

触发事件

duration:视频时长(秒)

-ended 视频播放结束触发事件

-timeupdate 视频正在播放中(多次)

currentTime 当前播放时间点

2.5:html5新特性-视频-样式(重点)

video 元素有一种专用(img)样式object-fit

此属性:指定视频在区域如何显示

fill: "填充" 默认值将视频拉伸操作填满父元素

contain: "包含" 保持原有视频比例，父元素空白区域

cover: "覆盖" 保持原有视频比例, 宽度和高度至少

有一个与父元素一致(裁切)

3:html5-新特性--绘图canvas---(重点)

-网页中绘图三种技术

(1) svg 用线段组件图形针对2D矢量图

2D=平面图片

特点:矢量可以无限放大和缩小并且不失真,缺点

颜色不够丰富

(2)canvas 用像素点组件图片针对2D位图

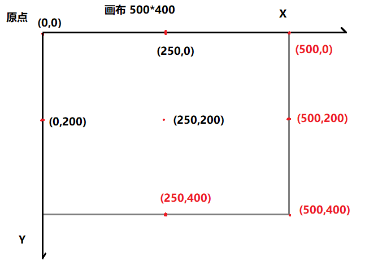
特点:不可以放大或缩小会真,优点颜色非常细腻

(3)web GL 3D位置

尚未纳入H5标准

3.1:html5-新特性--绘图canvas---(坐标系,单词多)

#在完成html5技术路径先不要用jquery.js



-通过标签创建画布

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

#注意事项:画布和宽度和高度只能通过js或者标签

属性来赋值，不能使用css赋值(变形)

-通过js程序获取画布并且

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

-获取画布指定画笔对象(上下文对象)

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

3.2:html5-新特性--绘图canvas---矩形

(1)绘制空心矩形(描边矩形)

ctx.strokeRect(x,y,w,h);

x,y 空心矩形左上角位置

w,h 空心矩形宽度和高度

(2)绘制实心矩形(填充矩形)

ctx.fillRect(x,y,w,h);

(3)设置实心图形样式

ctx.fillStyle = "#f00";

ctx.fillRect(0,0,100,80);生效

ctx.strokeRect(100,100,100,80)不生效

(4)设置空心图形样式

ctx.strokeStyle = "#fff";

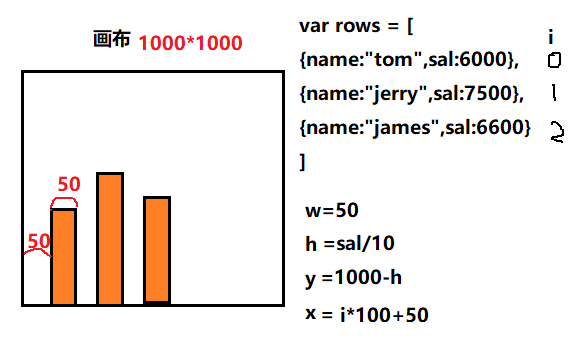
(5)清空矩形范围内所有元素

ctx.clearRect(x,y,w,h);

#清空画布所有元素 ctx.clearRect(0,0,500,400);

练习1:在画布上绘制左右移动矩形

练习2:在画布上绘制薪水统计图(柱状统计图)



作业1:完成薪水统计图--绘制人员名称

作业2:完成薪水统计图--分别给不同柱子添加颜色

方案一:rgb(0,10,100); 随机产生0~255

方案二:["#e4393c","#e43ddd",..]

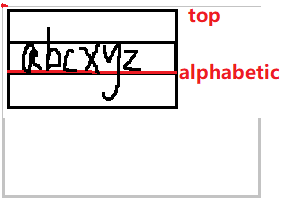
Day03

今天学习的内容

1:复习昨天内容

2:今天的目标

2.1:html5新特性--作业薪水统计图/文字/颜色



ctx.fillText(str,x,y)实心文字(填充文字)

str:文字

x,y:字符串左上位置(为文本基线)

ctx.strokeText(str,x,y)空心文字

ctx.font = "19px SimHei";前面字体大小,字体

ctx.textBaseline = "top";调整文本基线

["top","alphabetic","bottom"]

2.3:html5新特性--canvas--路径

路径:path由多个坐标点组件图形,图形本身不可见

可以使用"描边" "填充"

#比较复杂(不规则)图形使用路径

-ctx.beginPath(); 开始一条新路径(上一条路径结束)

-ctx.moveTo(x,y); 将画笔移动指定点 x,y 点

-ctx.lineTo(x,y); 从当前点到指定点画一条直线

-ctx.stroke(); 描边(空心)

-ctx.fill(); 填充(实心)

-ctx.closePath(); 闭合一条路径

(从结束点到开始点画一条直线)

-ctx.arc(cx,cy,r,start,end); 绘制一个圆拱形

cx,cy 圆心

r 半径

start 起始角度

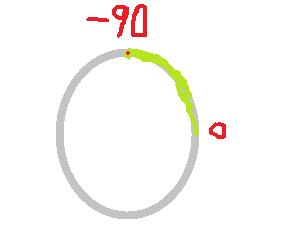
end 结束角度

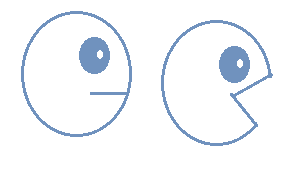
#参数中只能传递以弧度为单位值

#采用角度转弧度公式角度\*Math.PI/180=弧度

练习1:使用路径创建圆环进度条

练习2:张嘴和闭嘴笑脸





2.4:html5新特性--canvas--图片

-思考问题:一个软件项目中图片保存在哪里

所有图片都是保存在服务器端指定目录下

如果需要绘制图片,先下载图片再将下载好图片

绘制canvas画布上

-服务器端保存图片

(1)图片版权

(2)图片数量巨大

-思考问题:为什么要把图片画在画布中

(1)复杂度

(2)效率

2.5:html5新特性--canvas--图片--操作流程

-创建图片对象 var p3 = new Image();

-下载指定图片 p3.src = "x.png";

-为图片绑定*下载成功*事件p3.onload = function(){..}

-将图片绘制画布上

ctx.drawImage(p3,x,y); //原始大小图片

p3 图片对象

x,y 图片左上角位置

ctx.drawImage(p3,x,y,w,h) //拉伸图片

w,h 图片宽度和高度

2.6:html5新特性--canvas--图像变形操作

canvas绘图技术针对图片在绘制过程中进行变形

(旋转)操作

-rotate(deg)旋转

#旋转画笔对象,旋转轴心在画布原点

-translate(x,y)将整个画布原点移动指定位置

#旋转轴心会随画布原点移动

-save()保存画笔状态(原点/角度/颜色/..)

-restore()恢复画笔状态(原点/角度/颜色/..)

作业:仿https://www.bilibili.com/弹幕

Day04

今天学习的内容

1:复习昨天重点

2:今天的目标

2.1:完成视频canvas 弹幕-单机版(重点)

弹幕:专业视频网站常用功能

当用户在观看视频时可以发表自己的想法/情绪/观点

将文字浮动在视频上方

分析:

(1)背景视频

(2)前景弹幕区域,文字通常从右向左移动(速度)

(3)用户输入一段文字点击"发送"按钮，将文字显示

弹幕区域

(4)文字大小

2.2:完成视频canvas 弹幕-创建网页内容

(1)父元素中间视频和canvas画布

-父元素(相对定)

-视频 -后(900\*450) z-index:0

-canvas画布(弹幕区域)-前(900\*450) z-index:1

(2)用户输入区域

-输入框输入文字

-文字大小下拉列表文字大小

-发送

2.3:完成视频canvas 弹幕-js 功能

-main.js 主程序

-msg.js 弹幕程序

2.4:完成视频canvas 弹幕-msg.js

-一串弹幕文字X坐标(水平位置)

-一串弹幕文字 Y坐标(垂直位置)

-创建变量保存一串弹幕文字

-一串弹幕文字 font 文字大小

-一串弹幕文字 spd 速度

-最大100条弹幕文字串

2.5:调用顺序(!!!!!!)

msg.js完成弹幕

-var msgObj = function(){} #创建构造函数

-msgObj.prototype.num = 100 #100条弹幕池内容

-msgObj.prototype.init = function() #初始化

-msgObj.prototype.draw = function(){} #绘制弹幕

main.js调用弹幕对象方法

-function game(){#游戏入口函数

init();

gameloop();

}

-function init(){} #初始化

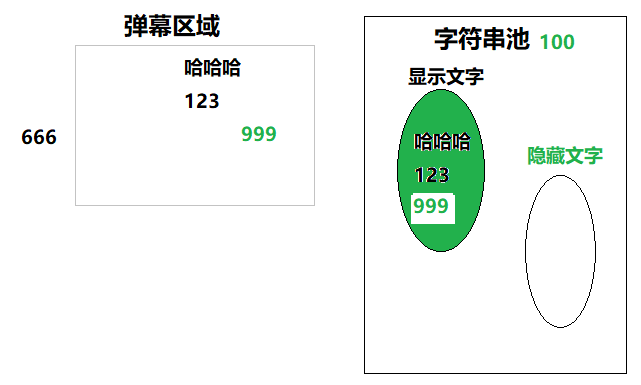
负责创建弹幕对象并且调用初始化方法

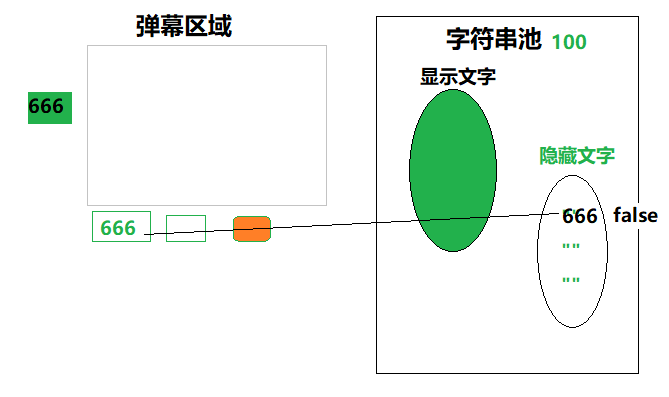
-function gameloop();#创建定时器循环

负责调用弹幕对象draw

-document.body.onload = game; #页面加载成功

2.6:弹幕字符串池原理





2.7:使用智能定时器requestAnimationFrame

#浏览器新对象智能定时器

#此定时器主要使用,动画和游戏场景  
原因:浏览器效率不同如果使用常规定时器小问题

setTimeout(fn,100)一次性

setInterval(fn,100)周期性

二台电脑新15ms旧 130ms

解决问题:requestAnimationFrame新(智能)定时器

requestAnimationFrame(fn)

使用方法特殊:

requestAnimationFrame == setTimeout

如何使用将一次性定时器改周期

function f1(){

requestAnimationFrame(f1)

}

f1();

2.x:/svg/echarts

2.x:拖放API上传文件(重点)

Day05

今天学习的内容

1:复习昨天重点

1.1调用顺序(!!!)

msg.js完成弹幕

-var msgObj = function(){} #创建构造函数

-msgObj.prototype.num = 100 #100条弹幕池内容

-msgObj.prototype.init = function() #初始化

-msgObj.prototype.draw = function(){} #绘制弹幕

main.js调用弹幕对象方法

-function game(){ #游戏入口函数

init();

gameloop();

}

-function init(){} #初始化

负责创建弹幕对象并且调用初始化方法

-function gameloop(); #创建定时器循环

负责调用弹幕对象draw

-document.body.onload = game; #页面加载成功

2:今天的目标

2.1:html5新特性--svg;echart;audio;拖放API(了解)

2.2:html5新特性--svg

(1)创建30个圆形

(2)直线

<line x1="" y1="" x2="" y2="" stroke-width=""

stroke="" stroke-linecap=""></line>

x1 y1 起点坐标 x2 y2 终点坐标

stroke-width 线宽度stroke 线样式

stroke-linecap 线顶端样式 round

(3)折线

<polyline points="50,50 75,100 " stroke=""></polyline>

points 一组坐标点

(4)渐变对象-特效对象一种

<defs>定义特效对象

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

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

</linearGradient>

</defs>

#x1 y1 起始坐标 x2 y2 终点坐标使用%

#stop 颜色点 offset="0%" stop-color="red"

# offset="100%" stop-color="green"

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

(5)滤镜-特效对象一种

#高斯模糊滤镜

#数字模糊级别 1~10

<defs>

<filter id="f3">

<feGaussianBlur stdDevitaion="数字" />

</filter>

</defs>

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

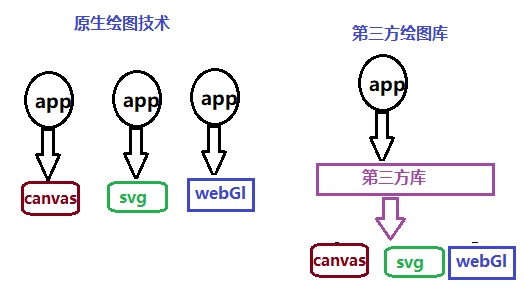
如何查询更多的滤镜效果

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

2.2:html5新特性--echarts

#如果需要快速创建功能简单外观复杂echarts

#第三方绘图库



#何时使用第三方绘图库完成任务

(1)外观要求精美

(2)图形比较复杂

#何时使用自定义图形

(1)网页特效[弹幕]

(2)休闲小游戏

2.2:html5新特性--echarts-百度第三方绘图库(重点)

(1)下载echarts绘图库

https://echarts.baidu.com/download.html

(2)创建存储图形容器

<div id="main" style="width:500px;height:400px"></div>

(3)将echarts.js 文件引入网页中

<script src="echarts.min.js"></script>

(4)获取容器并且创建echart对象

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

var mychart = echarts.init(main)

(5)创建配置对象(重点)柱状统计图

var option = {

title:{text:"入门示例"} #在画布顶部添加标题内容'入门示例'

xAxis:{data:["衬衫","裤子","袜子","鞋"]},#x轴值

yAxis:{}, #y轴值

series:[{type:"bar",data:[120,10,300,66]}] #数据源

};

#type指定图形类型 bar 柱状统计图

#type 指定图形类型 line 折线图

#type 指定图形类型 pie 饼状图统计图

#type 指定图形类型 gauge 仪表图

(6)将配置对象添加mychart完成

mychart.setOption(option);

练习:绘制一家公司销售统计图-柱状

var rows=[

{"city":"北京","sal":1900},

{"city":"上海","sal":2100},

{"city":"杭州","sal":2200},

{"city":"广州","sal":3900}

];

使用echart绘制销售统计图

2.3: html5新特性--拖放API

#pc端项目--(拖放上传图片)

#移动端项目--(上传图片) [自拍][相册]

Drag & Drop 拖动和释放

-拖动源对象(会动)-触发三个事件

dragstart拖动开始

drag 拖动中

dragend 拖动结束

整个过程:dragstart\*1+drag\*n+dragend 拖动结束

-拖动目标对象(不动)-触发四个事件(坑)

dragenter 拖动进入

dragover 拖动悬停 #默认行为悬停结束立即触发离开

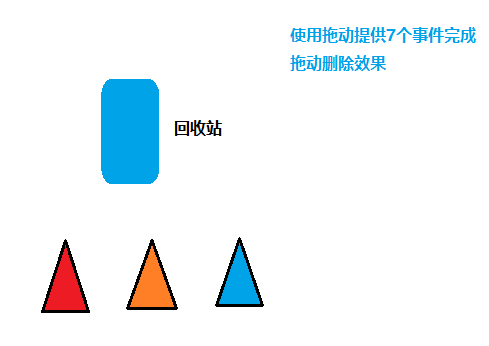
dragleave 拖动离开

drop 拖动释放

通用解决方案:阻止默认行为

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

整个过程:dragenter\*1+dragover\*n+drop\*1



完成练习思路: 17:57

1:为每张小飞机图片添加id属性p1 p2 p3

2:创建全局变量pid 保存准备删除小飞机id

3:当小飞机***拖动开始***将当前小飞机 id保存pid ??

4:回收站绑定释放事件

Day06

今天学习的内容

1:复习上周知识重点

3D AR

2:今天的目标

2.1:完成拖动练习

2.2:web Worker--理论多代码(3行)

程序:指可以被CPU执行代码,程序存储在磁盘上

进程:将程序调用内存中并且分配指定空间,

在内存中运行代码称为"进程"

线程:进程序内部是由多个线程组件

chrome浏览器(程序)

一个chrome进程内部至少有6个线程负责向服务器

发送请求获取资源.(资源请求线程)

一个线程负责所有资源绘制并且负责执行js程序

(UI主线程)

<button></button>

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

<button></button>

问题:如果长时间用户看不到网页内容，用户选择关闭

解决方案:创建新线程帮助UI主线程去执行耗时js任务

UI主线程只负责绘制网内容

新线程负责执行js 任务(Worker)

标准语法:

var w = new Worker("js路径");

(1)创建Worker对象

(2)创建新线程执行js路径中指定js 程序

#注意事项一:Woker线程执行js程序中不能包含

任何DOM/BOM 代码

jquery代码也不能在Worker中执行.

#注意事项二:UI线程与Worker之间可以发送数据

(1)UI线程(html)发送数据Worker线程(js)

UI发数据 (03.html)

var w = new Worker("03.js");

w.postMessage(msg)

Worker收数据(03.js)

onmessage = function(e){e.data}

(2)Worker线程发送数据给UI线程

Worker发数据

postMessage(msg)

UI接收数据

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

2.3:Web Storage

解决问题:可以在*客户端浏览器*存储用户专用数据

用户专用数据:登录用户昵称/uid/购物车/..

|  |  |
| --- | --- |
| 客户端存储方式 | 服务器端存储方式 |
| (不安全/数据量很少/) | (安全/海量/高速查询) |
| (1)cookie  优点:兼容性极好,操作复杂,  存储数据1KB | (1)数据库 mysql |
| (2)Flash  缺点:依据Flash播放 | (2)session 对象(数据量少) |
| (3)WebStorage  缺点:兼容性差,操作简单,  存储8MB | (3)文件 data.txt |
| (4)IndexDB 不标准 | (4)Redis 内存级存储方式 |

WebStorage存储数据方式依赖二个对象

(1)sessionStorage

此对象存储范围有效期也是在当前会话中,在会话中

所有网页都可以保存或者读取sessionStorage中数据

但是如果会话结束,存储数据也丢失.

(2)localStorage

此对象存储有效期永久.

2.4:Web Storage 通用函数与属性

以下函数与属性是sessionStorage和localStorage通用

(1)保存数据

setItem(key,value); key 数据名value 数据值

示例: setItem("uname","james")

(2)获取数据

var value = getItem(key)

(3)删除指定数据

removeItem(key)  
 (4)清空所有数据

clear();

(5)数据个数(长度)

length

练习:05\_save.html 保存数据 06\_get.html 获取数据

05\_save.html 保存昵称通过超链接跳转06 读取数据

2.5:大鱼小鱼--功能分析

-蓝色大海背景

-紫色海葵左右摆动(重点)

-海葵负责创建食物(重点)

-大鱼负责吃食物

-大鱼吃食和得分蓝色100分橙色200分

-小鱼跟着大鱼

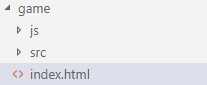
2.6:分析游戏目录结构

game游戏目录保存所有代码与资源

-js 游戏代码

-src 游戏资源(图片)

-index.html 入口文件



2.7:创建基本游戏代码(!!!!!)

-复制游戏图片保存src目录

-index.html 添加画布

画布一:(前)大鱼/小鱼/分数800\*600 z-index:1

画布二:(后)背景/海葵/食物 800\*600 z-index:0

-在js创建文件 main.js 游戏主程序

(1)全局变量: 背景/海葵/食物/大鱼/分数/..

(2)game 函数: 游戏执行第一个函数

document.body.onload = game;

game 负责调用下列二个函数

(3)init 初始化游戏资源

(4)gameloop 将游戏内容绘制画布上

2.8:添加大海功能

分析:将大海图片background.jpg 绘制画布2结束

(1)创建文件js/background.js

(2)在文件添加函数drawBackground[4]

(3)在main.js 添加全局变量 bgPic[1]保存背景图片

(4)在 init 创建图片对象[2]下载图片[3]

(5)在gameloop 调用drawBackground(定时器)

|  |
| --- |
| 图片绘制通常方式  var p3 = new Image();创建图片对象[1][2]  p3.src = "img/p3.png";下载图片[3]  p3.onload = function(){事件下载成功  ctx.drawImage(p3,0,0)绘制图片[4]  } |

2.9:添加海葵功能  
 (1)添加js/ane.js

(2)添加三个函数 aneObj 构造函数 init初始化 draw绘制

(3)在index.html加载ane.js

(4)并且在main.js 创建构造函数对象调用init/draw方法

(5)分析功能添加ane.js 代码

-海葵一共50根(OK)

-起点x,y 坐标/终点 x,y 坐标

this.x = []; 起点x与终点x

this.len = []; 海葵高度

作业:完成海葵绘制任务

Day07

今天学习的内容

1:复习昨天重点

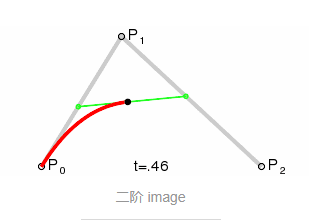
2:今天的目标

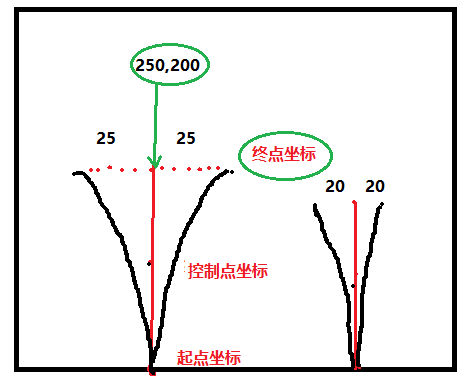
2.1:海葵-第二版本 ane2.js

分析功能:摆动海葵

问题:如何绘制摆动海葵: 曲线

解决方案:贝赛尔曲线(二次)





解决方案:贝赛尔曲线:p0(起点坐标) p1(控制点) p2(终点)

左右摆动修改终点坐标 x值

代码

(1)起点坐标this.rootX = []; #y高度

(2)控制点坐标不需要x因为与起点一致 y=150

(3)终点坐标 this.headX = []

this.headY=[];

(4)摆动幅度 amp = [];

(5)如何绘制贝赛尔曲线

-移动起点坐标 moveTo(起点x,起点y)

-绘制贝赛尔曲线 quadraticCurveTo(控制点x,控制点y,

终点x,终点y);

2.2:食物--fruit.js

(1)30个食物其中蓝色居多橙色很少

fruit.png blue.png

(2)同一个时刻只显示15个食物

[15显示 15隐藏]

(3)食物行为:出生 w->14长大

向上漂浮y-1

(4)食物漂浮出屏幕表示隐藏

(5)从原先隐藏食物中挑一个显示

2.5:大鱼 mom.js

(1)哪个画布

画布1:前大鱼/分数

画布2:后背景/海葵/食物

(2)大鱼组成

眼睛 bigEye0.png bigEye1.png 二张

身体 bigSwim0.png .. bigSwim7.png 八张

尾巴 bigTail0.png .. bigTail7.png 八张

作业:完成html5绘制示例,大鱼眼睛

WEIXIN

Day01

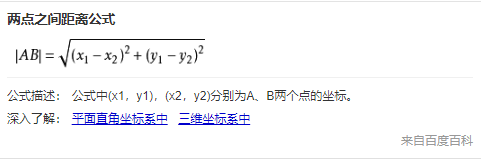
今天学习的内容

1:复习昨天的内容

2:今天的目标

2.1:大鱼分数

2.2:完成经典游戏技巧--碰撞检测(两点之间距离)



js/commonFunction.js 公共游戏专用函数库

2.3:完成经典游戏技巧--

源对象(大鱼)向目标对象(鼠标)慢慢移动

2.4:WebSocket 多对多聊天室-(重点)[上传视频]

(1)HTTP协议

HTTP:超文本传输协议

HTTP作用:传输网页中资源(html;css;js;img;video;..)

HTTP浏览器搬运工

HTTP工作方法:请求与响应

客户端浏览器发送请服务器数据:(请求)

服务器发送客户端浏览器:(响应)

一次请求对应一次响应

问题:在有一些软件环境下此协议不适合

#金融行业:(股票/期货/汇率/..)多长时间发送一次

新解决方案:新协议WebSocket

webSocket工作方式:广播和收听

webSocket适合:金融/***聊天室***/弹幕/....

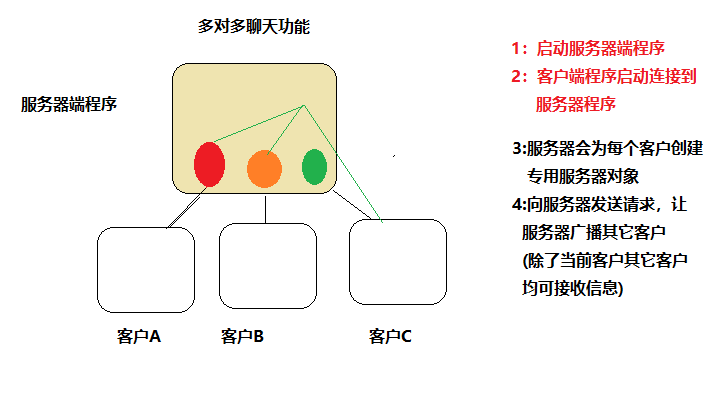
2.5:webSocket 开发流程

#浏览器自带对象WebSocket

#由于 自带对象WebSocket工作复杂/使用第三方

#WebSocket库来实现此功能

#库名 socket.io



(1)下载socket.io服务器端第三方模块

#服务器端下载:npm i socket.io

下载socket.io 客户端文件

#客户端下载文件:socket.io.js

https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.3/socket.io.js

(2)先开发服务器端程序

2.1:创建web服务器对象

var app = require("http").createServer();

2.2:创建websocket.io对象

var io = require("socket.io")(app)

2.3:监听端口 3000

app.listen(3000)

2.4:为io对象绑定事件 connection 客户端连接

io.on("connection",(socket)=>{

//connection事件当有客户连接到服务器

//触发connection

//创建为此客户端专门服务对象socket

//你客户端id

//接收客户端消息向客户发送消息(一对一)

})

(3)开发客户端程序

3.1:创建socket对象并且连接websocket服务器

var clientSocket = io("ws://127.0.0.1:3000");

#clientSocket创建变量

# io() 向指定服务器发出连接请求

#此函数调用触发服务器connection

#ws 协议名称 127.0.0.1 服务器在本地 3000 服务器

端口

#====================================

客户程序发送内容给服务器一对一

绑定message事件

服务器端(收): socket.on("message",(data)=>{data})

触发message事件

客户端(发):clientSocket.emit("message","hello");

#====================================

服务器端给客户端回复消息一对一

绑定server 事件

客户端(收)clientSocket.on("server",(data)=>{})

服务器(发)socket.emit("server","...")

#====================================

广播消息:服务器发送客户

客户端(收) clientSocket.on("list",(data)=>{..})

服务器(广播发送) io.emit("list","明天开始大降价")

#====================================

服务器端绑定事件:disconnect 默认事件

#当客户端浏览器关闭触发disconnect事件

2.6:微信小程序

微信公众平台--(订阅号/小程序/小游戏)

小程序:轻量APP

pc/ios app/android app

2.7:微信小程序

-注册微信公众平台帐户

-下载开发工具

-注册小程序主体类型[个人][企业]

企业:微信支付/卡包/附近小程序

2.8:微信小程序--登录

左侧开发菜单->开发设置->appid #小程序开发钥匙  
 2.9:下载安装最新开发工具

https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html

作业1:使用websocket 完成多对多聊天室

Day02

今天学习的内容

1:复习昨天重点

2:今天目标

2.1:小程序-开发工具

#编译将所有代码重新编译执行看最结果

#预览将代码上传微信服务器,通过手机扫描看结果

#上传将项目上传微信服务器等待审核(开发完成)

#详情 (1)调试基础库 2.2.5[2.2.4]

注意事项:每个小程序项目总体大小不能超过2MB

2.3:小程序-开发工具-项目目录结构



2.4:小程序-开发工具-创建组件

app.json

page:[

"pages/base/base"

]

#自动在pages创建目录base并且在base创建组件

base.wxml

base.wxss

base.js

base.json

2.5:小程序-不同类型文件(wxml)

小程序中没有html替换wxml类似xml

#严格匹配开始与结束标签<text></text>

#常见标签

view 相当于<div></div> #块级元素

text 相当于<span><p> #行内元素

image 相当于<img src="">

form 相当于<form>

..

#常见错误: end tag missing, near `text`没有结束标签

<text selectable='true'></text>

selectable='true'文本内容可以选中复制

<image src="1.jpg" lazy-load="true"></image>

lazy-load="true" 懒加载(提升项目效率方法)

图片默认大小:320\*240

2.6:小程序-不同类型文件(wxss)

wxss 是一套小程序专用样式语言,功能描述样式

(1)单位 rpx(responsive pixel)响应式像素,可以根据

屏幕宽度自适应调整大小

(2)关键字 @import "./style/base02.xss"

?问题移动端像素

-前端设计师: 设计稿(图片)物理像素

-前端开发工程师:将设计稿转网页逻辑像素

示例:iphone6 (dpr 2)/iphone 6plus (3 dpr)

逻辑像素 \* dpr = 物理像素

iphone6 375 \*2=物理像素宽度 750

小程序:rpx

物理像素 750=逻辑像素 750rpx

#小程序项目中会使用 rpx

#@import 引入公共样式文件

-style/common.wxss

-admin/admin.wxss

@import "../../style/common.wxss"

2.7:小程序-第三方组件库

Vant Weapp -有赞团队开发组件库(项目)

https://youzan.github.io/vant-weapp/#/intro

iView

WeUI

2.8:小程序-不同类型文件(json)

(1)全局配置文件app.json

(2)当前组件配置文件 admin.json

(1.1) 全局配置文件影响项目中所有组件

"pages":["pages/base/base"] 组件列表

"windows"全局窗口样式

"tabbar"底部导航栏

"list":[

{

"pagePath":"组件路径" 点击按钮跳转哪个组件

"text":".." 按钮默认文字

"iconPath":"" 默认图标路径

"selectedIconPath":"" 选中图标路径

}

]

#按钮最少二个

#图标路径只能本地图片

#图标大小81\*81px 最大40kb

#json文件不能添加注释

2.9:小程序-不同类型文件(js)

-小程序中js与网页中js区别

(1)ECMA 基本类型:string/number/boolean/null/undefined

(2)ECMA 对象:Object/Date/Math/Array/RegExp/...

(3)DOM/BOM不能使用

(4)小程序顶层以象 wx 相当网页 windows

-app.js 全局共享数据

-admin.js当前js文件

Page({})当前组件对象(对象/事件/方法)

-data:{} 保存模板中显示数据

示例: data:{msg:"Hello"}

<view>{{msg}}</view>

2.10:小程序-数据控制与显示

(1)循环

<view wx:for="{{数组名}}" wx:key="排序条件属性">

</view>

-item 默认表示当前对象

-index 默认数组下标

-wx:for="{{数组名}}"

-wx:key="排序条件" #index id

(2)if

<view wx:if="{{condition}}"></view>

condition:表达式结果true false

true 当前元素正常显示

false 当前元素删除

(3)hidden

<view hidden="{{condition}}">隐藏内容</view>

true 当前元素隐藏

false 当前元素显示

#建议:如果此元素需要频繁切换使用 hidden

#建议:在程序运行时不会再改动 if

(4)if else

<view wx:if="{{condition}}"></view>

<view wx:elif="{{condition}}"></view>

<view wx:else></view>

2.11:小程序-事件

pc端:屏幕宽度> 970px操作:键盘鼠标事件

移动端:屏幕宽度< 970px 操作:手指事件

-touchstart 当手指触碰屏幕时发生事件[不管几个手指]

-touchmove 当手指在屏幕上滑动连续触发

-touchend 当手指离开屏幕时触发

-touchcancel 当手指***操作***被打断

-tap 一次手指触碰

-longtap 手指长按超过350ms再离开(旧)

-longpress 手指长按超过350ms 再离开(新)

小程序将事件绑定元素

-支持事件冒泡<view bindtap="事件处理函数"></view>

-不支持事件冒泡<viewcatchtap="事件处理函数"></view>

2.12:小程序-生命周期

组件生命周期:一个组件从创建开始到使用至最后被

销毁过程

onLoad 组件创建成功并且加载完成一次

(1)当此事件触发发送请求获取数据

(2)获取其它组件传递数据

onReady 组件渲染成功一次

onShow 组件显示(前台) 多次

onHide 组件隐藏(后台)多次

onUnload 组件销毁一次

练习:

-创建二个组件 exam03,exam04

-默认显示 exam03

-在exam03组件添加元素 view并且为其绑定tap

事件当tap事件触发时跳转exam04

-将exam03 数据id传递exam04

#小技巧:如果通过js修改data中数据

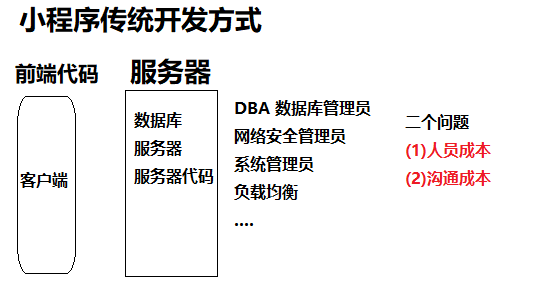
-获取data中数据 this.data.msg

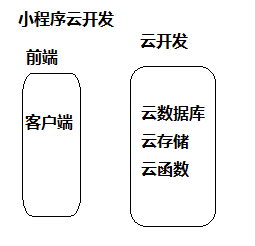
-修改 data中数据 this.setData({msg:"新值"})

2.13:小程序-云开发

小程序+腾讯云合作新方案(云开发)

[云函数+云数据库+云存储]





2.14:小程序-云开发

(1)云数据库: mongodb 增加/删除/更修/查询

(2)云存储:上传图片/下载文件/分享文件/文件管理

(3)云函数:小程序用户登录

2.15:小程序-开通云开发

(1)小程序开发工具-->点击按钮"云开发" 开通

(2)环境名称:[web-test-01]点击"确定"

#不要有空格,不要有中文,字母开头

#自动创建环境id web-test-01-xhsd93

#数据库5GB 文件 2GB 流量 5GB/月

2.16:小程序-云数据库(mongodb NOSQL 数据库)

|  |  |
| --- | --- |
| 关系数据库 | NOSQL数据库 |
| 库 database | 库 database |
| 表 table | 集合 collection |
| 行 row | 记录 record |
| 列 column | 字段 field |

#关系型数据库:适合数据关系复杂企业内部系统

#NOSQL:互联网项目(频繁查询/更新)移动app

|  |  |
| --- | --- |
| 数据类型 | Number 数字 |
| String 字符串 | Object 对象 |
| Array 数组 | Date 日期(客户端) |
| GeoPoint地位位置 | Bool/Null |

\*作业1:查看云开发

\*作业2:创建集合 photo

作业3:创建组件完成向集合添加三条记录

{id:1,name:"tom",age:19}

Day03

今天学习的内容

1:复习昨天重点

2:今天的目标

2.1:小程序云开发--云数据库

#权限

操作方式一:云开发控制台{创建添加修改}

操作方式二:小程序 {创建添加修改}用户相关(有限)

#当用户添加数据可以操作

操作方式三:云函数 {创建添加修改}权限更高全部

2.2:小程序云开发--小程序-添加记录

环境id:web-tao-test-01-5x4hi

(1)在控制台中创建集合 web1904\_01

(2)程序中创建数据库对象

const db = wx.cloud.database({

env:"web-tao-test-01-5x4hi" #环境id

});

(3)向指定集合 web1904\_01中添加一条记录

db.collection("集合名称").add({

data:{ //添加数据(记录)

字段名:值,

字段名:值

}

}).then(res=>{}) //回调函数成功

.catch(err=>{}) //回调函数失败

常见错误

(1) -401003 api parameter type error

参数类型错误:

检查:data:{}拼写错误

(2) Error: errCode: -501005 invalid env//环境

无效环境id修改env环境id修

(3) -502005 database collection not exists

集合不存在

集合名拼写错误

#\_openid

#微信平台:小程序/订阅号/

#获取当前登录用户编号 unionid/\_openid

#以上两个id用于标识用户身份

#\_openid平台唯一(小程序/订阅号)

2.3:小程序云开发--小程序-更新记录

db.collection("集合名称")

.doc("当前记录id") //云数据自动生成\_id条件

.update({ //更新

data:{

sal:12000 //数据

}

}).then(res=>{}) //更新成功

.catch(err=>{}) //更新失败

常见错误:更新成功数据没有改变

2.4:小程序云开发--小程序-删除记录

#删除有一定危险性操作:非常谨慎

#小程序端只能一次删除一条数据

db.collection("集合名称")

.doc("当前记录id") //当前记录id

.remove() //删除

.then(res=>{}) //删除成功回调

.catch(err=>{}) //删除失败回调

2.5:小程序云开发--小程序-查询记录

(1)查询所有记录

db.collection("集合名称") #当前集合中你添加数据查询

.get()

.then().catch()

(2)查询指定记录

db.collection("集合名称")

.where({ //查询条件

sal:10000

}).get().then().catch();

-age:db.command.lt(21), //小于

-age: db.command.eq(30) //等于

-age: db.command.neq(30) //不等于

-age: db.command.lte(30) //小于等于

-age: db.command.gt(30) //大于

-age: db.command.in([30,40])//在

#较为复杂指定查询小程序官方手册

https://developers.weixin.qq.com/miniprogram/dev/wxcloud/reference-client-api/database/database.html

#W3C 小程序手册

https://www.w3cschool.cn/weixinapp/3u9c1qhw.html

综合示例:

(1)在云控制面板创建集合 web1904\_emp

(2)创建组件 exam03

(3)添加view 元素绑定点击事件添加三条记录

uname job sal

(4)添加view 元素绑定点击事件查询添加记录

将查询结果显示exam03 模板

(5)\*为每条记录后添加<view>del</view>

当用户点击 del元素删除指定记录

2.6:小程序云开发--云函数

云函数:运行在腾讯云的程序

开发方式:

(1)你在小程序工具中创建云函数,上传腾讯云中

(2)先在控制面中对云函数测试

(3)在小程序中调用云函数

云函数:开发方式特殊函数

(1)操作云数据库权限高(一次删除多条记录)

(2)网络限制

云函数示例一: sum() 云函数语法

云函数示例二:批量删除数据

2.7:小程序云开发--云函数-sum

#注意事项:云函数要求本地环境安装node.js 8.0 以上

(1)创建云函数

(2)添加代码

package.json云函数描述文件

index.js

|  |
| --- |
| exports.main = async (event,context)=>{  return {  "sum":event.i+event.j  }  }  #exports.main 创建主函数并且导出  #async异步执行  #event 事件对象小程序调用云函数传递参数  依据event获取  #context 上下文对象:用户信息 |

(3)上传和部署(!!!)

(4)云控制面上:测试

(5)使用小程序调用 16:30

wx.cloud.callFunction({

name:"云函数名称",

data:{i:1,j:2}

})

.then(res=>{console.log(res})

.catch(err=>{console.log(err)})

2.8:小程序云开发--云函数-batchDel

1:创建数据库实例对象db

const db = cloud.database({env:"web-tao-test-01-5x4hi"});

2:创建main函数

#删除多条记录

#async 异步 ES7语法

#await 等待 ES7 语法

exports.main = async(event,context)=>{

try{

return await db.collection("web1904\_emp")

.where({sal:20000}).remove();

}catch(e){

console.log(e);

}

}

2.9:小程序云开发--云存储[文件上传云存储]

(1)选中图片

#从手机相册或拍照

#提供开发云临时图片地址

(2)上传图片

#将临时图片上传云存储

1.1:选中图片

wx.chooseImage({})选择图片

count:1 一次选择一张图片

sizeType:[] 图片类型原图/压缩图

sourceType:[] 图片来源相册/相机

success:(res=>{ 选择成功

res.tempFilePaths 数组[最多可以选中9张图片]

})

1.2:上传图片

wx.cloud.uploadFile({})

cloudPath:新文件名称 new Date().getTime()+".jpg"

filePath:临时文件名称

success:(res=>{console.log(res.fileID)})

#fileID你上传图片路径

练习:

(1)云数据库中创建集合 myphoto保存上传文件fileID

(2)在exam04创建 view元素绑定点击事件"上传"

完成上传图片操作(将fileID保存myphoto)

(3)在exam04创建 view 元素绑定点击事件 "显示"

完成显示图片

(4)查询数据库中 myphoto fileID 显示图片

<image src="{{item.fileID}}" >

Day04

今天学习的内容

1:复习上周重点

2:今天的目标

2.1:云存储

(1)云数据库中创建集合 myphoto保存上传文件fileID

(2)在exam04创建 view元素绑定点击事件 "上传"

完成上传图片操作(将fileID保存myphoto)

(3)在exam04创建 view 元素绑定点击事件 "显示"

完成显示图片

(4)查询数据库中 myphoto fileID 显示图片

<image src="{{item.fileID}}" >

2.2:学子商城--云开发版本-娱乐模块(热门电影)

(1)电影列表

(2)电影详情与评论(重点)

(3)位置

2.3:学子商城--云开发版本-*第三方模块 vant****配置***

网址:https://youzan.github.io/vant-weapp/#/intro

(1)创建项目描述文件 package.json

-右击 miniprogram 目录->选中"在终端中打开"

-输入命令 npm init

#一直按回车直到结束

#npm init 命令生成package.json

(2)通过npm安装 vant第三方模块

-右击 miniprogram 目录->选中"在终端中打开"

-npm i vant-weapp -S --production

(3)点击开发工具->工具菜单->构建npm

#创建新目录miniprogram\_npm

#将nodes\_modules复制miniprogram\_npm

#如果报错,重新执行第三步

(4)点击开发工具->详细按钮->[\*] 使用npm模块

(5)在选定组件中配置内应用到当前组件

复制文档内容

"van-button": "path/to/vant-weapp/dist/button/index"

修改

"van-button": "vant-weapp/button/index"

(6)直接复制代码保存当前组件中即可

注意一:以上操作有顺序

注意二:第三步(3)如果构建不成功.多次构建

2.4:学子商城--云开发版本--电影列表

豆瓣网:提供开放接口

接口:http://api.douban.com/v2/movie/in\_theaters?

豆瓣开发密钥:apikey=0df993c66c0c636e29ecbb5344252a4a

参数:&start=0&count=10

start 从第几条记录开始

count 本次查询几条内容

2.5:学子商城--云开发版本--发送请求

|  |  |  |
| --- | --- | --- |
|  | 小程序端 | 云函数 |
| 发送方式 | wx.request({}) | 第三方ajax库request |
| 协议支持 | 只支持https | 根据第三方库决定 |
| 是否备案 | 经过ICP备案 | 可以不用备案 |
| 域名个数限制 | 20个 | 无限 |

2.6:学子商城--云开发版本--创建云函数

(1)创建云函数 movielist4

(2)在当前云函数安装第三方ajax库

#不同云函数之间第三方库不能共享

(3)右键->moielist4->打开终端(下面命令有顺序)

npm install -S request #依赖库

npm install -S request-promise #ajax请求库

(4)开发云函数index.js

4.1:引入request-promise 库

const rp = require("request-promise")

4.2:发送请求

exports.main = (event,context)=>{

return rp(豆瓣网站).then(res=>{

return res;

}).catch(err=>{

console.log(err);

})

}

4.3:参数

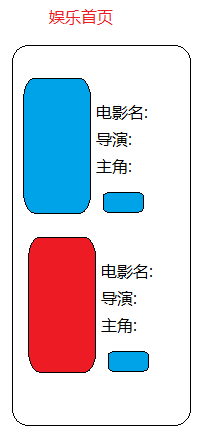
http://...&start=${event.start}&count=${event.count}

4.4:上传部署

4.5:测试

#错误:request-promise不存在

2.7:学子商城--云开发版本--创建组件/home



功能一:当组件创建成功后调用云函数

获取第一页数据保存list

onload加载第一页数据

(1)创建函数 loadMore() start=0 count=4

(2)在onload事件处理函数中调用上述函数

功能二:当用户向上滑动加载下一页数据

onReachBottom上拉触底

start:0,起始行

count:4当前页一共几行

1页:0 2页:4 3页:8...

作业:

详情组件:添加评论评论上传多张图片

Day05

今天学习的内容

1:复习昨天的内容

2:今天的目标

2.1:学子商城

2.2:学子商城->底部导航条

[电影][个人中心][地图]

(1)创建组件 pages/profile/profile pages/map/map

(2)下载图标https://www.iconfont.cn/

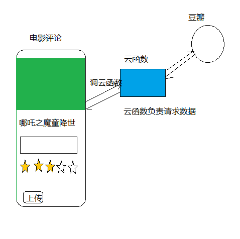
(3)app.json 配置tabbar

2.3:学子商城->电影详情评论(重点)

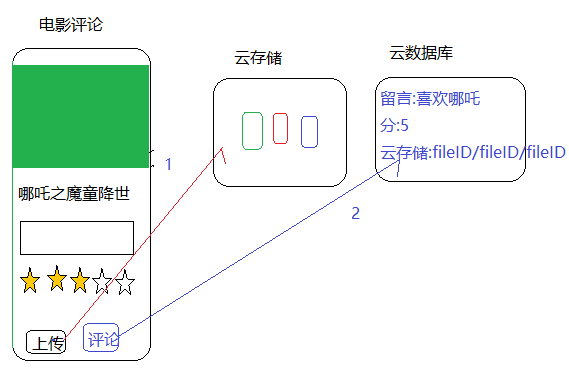
(1)添加组件评论组件pages/comment/comment

(2)完成功能

功能一:显示电影详情



功能二:评论 [留言/打分/上传多张图片]



2.4:学子商城->功能一:显示电影详情-云函数

(1)创建云函数 findDetail1904

(2)下载安装第三方ajax库 request/request-promise

npm install -S request

npm install -S request-promise

(3)添加index.js

http://api.douban.com/v2/movie/subject/26794435?

apikey=0df993c66c0c636e29ecbb5344252a4a

#注意:电影id热门电影会发现变化

(4)鼠标右击->上传部署

(5)云测试

2.5:学子商城->功能一:显示电影详情-组件

#pages/comment/comment

(1)输入框获取用户输入内容

(2)打分获取用户评分值 0~5

(3)上传按钮

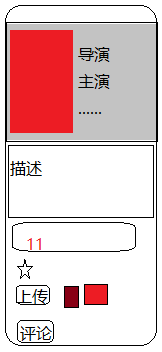
#查看vant手册

https://youzan.github.io/vant-weapp/#/intro小程序

https://youzan.github.io/vant/#/zh-CN/intro vue

#复制对应组件配置文件 comment.json

#在comment.wxml 调用



(4)当组件创建成功后调用云函数 findDetail1904

获取云函数返回结果id:26794435

常见错误:404011 云函数不正确

Function not found: [findDetail19041]

没有找到指定函数

(5)显示组件

(6)显示热门电影列表,点击详细按钮跳转当前comment

组件传递当前电影id

2.6:学子商城->功能二:选中多张图片

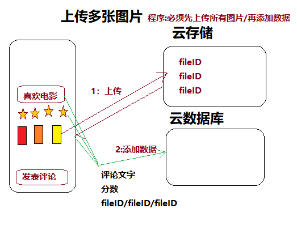
(1)点击上传图片按钮(选中多张图片)

wx.chooseImage({});

(2)预览图片

2.7:学子商城->功能二:上传多张图片并且并图片id保存

云数据库



#以上程序会出现问题:无序

#上传几张图片后开始添加数据

#结果图片没有完成所有上传添加完成,结果错误

解决方案:Promise

(1)将9张图片保存数组 images

(2)创建循环创建对象

new Promise(re,rej)=>{

上传一张图片操作

(1)创建新文件名/上传/上传成功将fileID保存

}

(3)添加数据,必须9上传Promise完成再添加数据

Promise.all(promoise数组).then(res=>{

#当数组所有Promise对象执行完成

#回调then

#获用户输入评论/分数/fileIds 添加云数据库集合中

})

作业1:复习Promise

作业2:一张上传多张图片/再添加数据

Day06

今天学习的内容

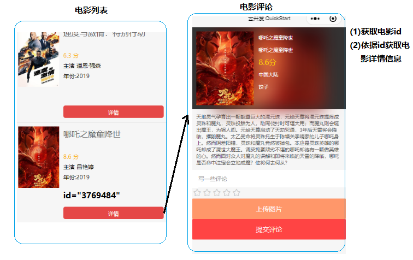
1:复习昨天重点

2:今天的目标

2.1:学子商城--评论

(1)组件之间跳转与参数传递

home 电影列表 comment 评论



(2)为home组件按钮"详情"

<button data-id="{{item.id}}"

bindtap="jumpComment">详情</button>

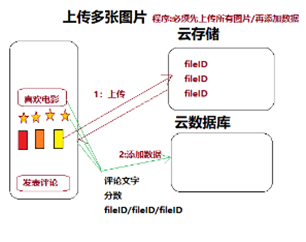
var id = e.target.dataset.id;

wx.navigateTo({url:"组件路径?id="+id})

(3)comment 组件显示内容

onLoad(options){options.id}

|  |
| --- |
| #注意:组件之间跳转  wx.redirectTo({url: url,}) 删除跳转  #将原先组件从内存中卸载,显示新组件  wx.navigatorTo({url:url}) 保留跳转  #保留内存原先组件跳转显示新组件 |



2.2:学子商城--(提交评论)上传图片

(1)将用户选中多张图片一张一张上传云存储中

(2)将上传成功后图片fileID保存 data中

#当上传多张图片时遇到问异步操作

#异步:[效率高/无序]

#遇到问题:上传9张图片与向云数据库添加数据操作,

必须有序(同步)

#解决问题:Promise对象,可以将异步操作以同步操作

来实现.(无序改变有序)

#语法

#new Promise((resolve,reject)=>{})

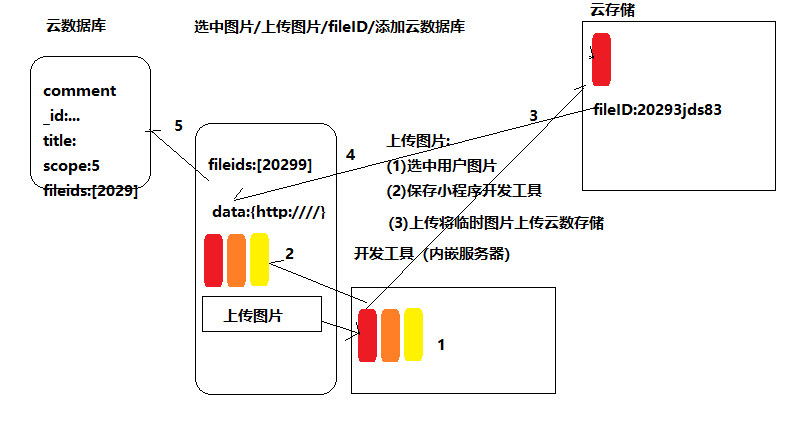
#resolve 解析程序一切正确(成功)调用resolve();

#reject 拒绝程序错误调用reject();

2.3:学子商城--(提交评论)发表评论

(1)将用户评论留言/打分/所有fileID一次添加

云数据库



2.4:学子商城--商城地图

html5 中新特性:geoloction 通过浏览器js获取当前用户

位置信息(经度/纬度/海拨/速度)数据,用于LBS

(Location Base Service)基于定位服务

饿了么,滴滴打车...

手机浏览器如何获取定位

(1)首先手机GPS芯片与网络定位精度在米范围

(2)pc定位IP地址释析

实现定位

(1)window.navigator.geolocation{

getCurrentPosition:fn}

(2)第三方服务器:百度地图/高德/腾讯地图

#小程序--(腾讯地图) map

#小程序提供非常多组件{vant UI组件库}

<map></map>

longitude 中心位置经度116.300901

latitude 中心位置纬度39.916085

scale 缩放级别 3~20

markes 标记点

polyline 路线

show-location 带方向定位点

style="width:100%;height:600rpx"

2.5学子商城--商城地图--个人信息 profile

-当前用户头像

<open-data type="userAvatarUrl"></open-data>

-当前用户昵称

<open-data type="userNickName"></open-data>

-获取当前用户信息

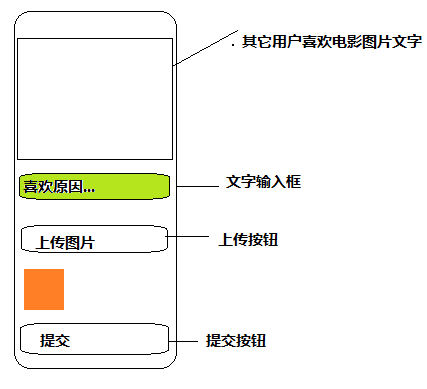
<button open-type="getUserInfo"

bindgetuserinfo="getUser"> #getUser自定义函数

获取信息

</button>

2.6学子商城--商城地图--我喜欢电影



功能一:添加喜欢电影信息(电影喜欢镜头/文字)

(0)创建组件mymovie

(1)上传喜欢电影镜头(图片)描述文字(一次一张)

(2)图片上传云存储

(3)图片fileID和描述文字 ->云数据库

功能二:显示所有用户喜欢电影信息(分页)

(1)分页滑动显示所有图片与文字列表

作业1:完成功能一功能二

三个软件

(1)手机模拟器

BluestacksCnSetup\_6191\_legacy.exe 蓝碟

夜神

(2)phonegap 服务器/手机模拟

PhoneGapSetup-win32.exe

PhoneGap\_Develop.apk

phonegap--混编

今天学习的内容

1:复习昨天重点

2:今天的目标

2.1:phonegap--混编

#安装软件环境 [关闭杀毒软件/防火墙]

混编:势头强技术

混编开发软件一种方式:你开发一套h5项目,由混编工具

生成 android app/ios app/小程序app/支付宝 app/

国外平台:phonegap

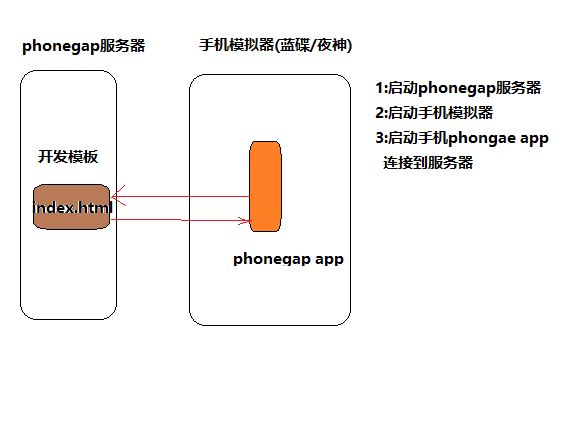
国内平台:uniapphttps://uniapp.dcloud.io/

#phonegap

(1)一键打包

(2)支持手机底层硬件开发(课程内容)

2.3:phonegap--混编-创建开发环境



(1)手机模拟器

BluestacksCnSetup\_6191\_legacy.exe 蓝碟

(2)phonegap 服务器/手机模拟

PhoneGapSetup-win32.exe

PhoneGap\_Develop.apk

2.4:phonegap--混编-调用手机底层硬件

-device

device.platform 获取手机操作系统 IOS Android

device.uuid 获取手机硬件编号

device.version 获取手机操作系统版本

-notification 提示功能

navigator.notification.alert("") 提示信息

navigator.notification.confirm("","") 确认信息

navigator.notification.beep(3) 蜂鸣器 3声

navigator.notification.vibrate(1000) 震动1s

-camera 相机

navigator.camera.getPicture(fn1,fn2,{参数})

fn1 拍摄成功

fn2 拍报失败

{quality:50}拍照延迟时间 50毫秒

-phonegap 提供对象 capture 录音

navigator.device.capture.captureAudio(fn1,fn2,参数);

fn1:录音成功回调函数 array 音频文件位置

fn2:录音失败

{limit:3} 录制几个音频文件

-phonegap 提供一个对象 accelerometer

获取用户位置信息

(1)获取当前用户位置信息

getCurrentAcceleration(fn1,fn2)

fn1:获取位置成功 x y z

fn2:获取位置失败

(2)创建监听器捕获用户动位置信息

var 变量 = watchAcceleration(fn1,fn2,参数)

fn1 获取位置成功 x y z

fn2 获取位置失败

参数监听频率

(3)取消监听器

clearWatch(变量)

2.5:微信公众号(订阅号-聊天机器人)

(1)注册订阅号-(注册新邮箱不能和小程序相同)

订阅号:个人(公司)--媒体平台(报纸/杂志/电台/)

2013

分类:专业(美食/技术/旅游/教育/情感) TOP

编造(历史/明星/国际/...)

2.6:微信订阅号--测试帐户完成聊天机器程序

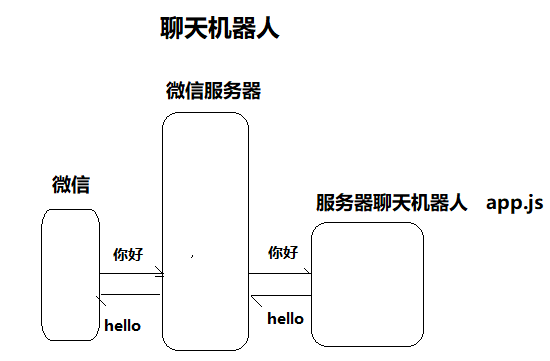
(1)进入你订阅号

(2)左侧菜单->开发者工具

(3)点击：公众平台测试帐户(微信扫描)

(4)appid: wxa56801014e697901

(测试帐户钥匙)



2.7:微信订阅号--程序创建

(1)合法域名/在线服务器

解决方案:内网穿透 (免费提供8小时合法域名和在线服务器)

ngrok工具软件

标准语法:ngrok.exe 协议名端口回车

示例: ngrok.exe http 4000

备选: .\ngrok.exe http 4000

合法域名:http://5fdd856b.ngrok.io

#申请成功后不要关闭窗口

(2)开发聊天机器人程序

-创建express服务器 4000

-创建微信服务器(配置)对象

var config = {

appid:"wxa56801014e697901",//钥匙(第一次验证)

token:"weixin"//令牌(第二次验证)

};

-接收微信服务器发送GET #连接测试

-下载安装第三方模块 wechat 专门处理微信传递文字

server.get("/",wechat(config,(req,res)=>{

-接收微信服务器发送消息

req.weixin.Content

-向微信服务器发送消息

res.reply(msg)

}))

-接收微信服务器发送 POST#聊天内容

server.post("/",wechat(config,(req,res)=>{

--接收 req.weixin.Content

--发送 res.reply(msg)

}))

问题:乱码

解决:保存程序文件使用GBK编号;微信支持 UTF-8

练习

(1)启动mysql数据库

(2)在weixin\_day/目录建文件 db.sql

(3)id q a

你好你好

你是谁我是聊天机器人

再见还来不来

(4)连接数据库

请MIS协助开外网完成下面三个课前任务：

(1)最新版本的Angular需要Node.js版本10.9.0或更高版本

https://npm.taobao.org/mirrors/node/v11.0.0/

node-v11.0.0-x64.msi

node-v11.0.0-x86.msi

可以在命令行输入 node -v 查看已安装的Node.js的最新版本；

如果需要更新Node.js，请到官网nodejs.org或者中文网nodejs.cn下载最新LTS版本并安装。

(2)(可选)Angular项目脚手架工具可用于创建空白的Angular项目，请提前安装：

npm i -g @angular/cli

(3)(可选)运行上面安装的Angular脚手架工具，创建一个测试项目并运行：

ng new myproject01

进入项目目录：

cd myproject01

运行该Angular项目：

npm start

启动成功后，可以打开浏览器访问该Angular项目:

http://127.0.0.1:4200

提示：如果2/3总是失败，等待明天课上解决。