

微信小程序开发

入门

App使用现状

App使用现状：

一个月用一次也要装App；

弹窗；

全家桶；

“保活”疯狂耗电；

恶意App。

App开发现状



开发成本高，两套平台，不同机器还要适配；

不同应用市场的困难；

H5性能差。

React Native; Weex……

微信小程序；

J前人的故事

微信小程序的战略意义

什么是“微信小程序”：运行在微信中的App，不用单独下载安装，使用起来像公众号一样。又比公众号好：可以调用手机的功能，能做网页做不了的事情；是本地程序，加载快，运行快。

“微信小程序”使用起来的优点：免安装；被腾讯“管控”，更老实；

“微信小程序”开发的优点：开发一次即可，Android、IOS双平台运行；

“微信小程序”的前景：众望所归，人民的期望；Android、IOS开发会受到一定的影响。

微信小程序的技术分析

1、微信小程序的开发技术是什么？

微信小程序是客户端开发技术，和服务器通讯采用Http、WebSocket等方式；使用JavaScript作为开发语言，使用类似Html的界面技术和类似Html5的API，是腾讯自己开发的技术。建立在大家掌握json、http通讯的基础上；

2、微信小程序是网页开发吗？是Html5吗？

不支持javascript dom，也就是document.getElementById、alert等，也就不支持JQuery等。只能说“类似”HTML、JS、CSS。学过HTML、JS、CSS有助于学习，如鹏训练营的课程之前讲过这些，所以这里就建立在大家懂这些的基础上去讲；

3、微信小程序技术特点：迭代比较快，新功能、开发工具一直在升级，也可能会引入bug。 getCurrentPage、 getCurrentPages。

4、微信小程序使用什么开发工具？

腾讯自己搞的开发工具，现在也有了EgretWing等第三方开发工具；

5、学微信小程序开发需要学什么？前后端；

如鹏微信小程序课程安排



项目介绍

课表介绍

开发环境搭建

右边资料区)

重新运行;

更新成功, 当前版本 0.10.102800



a5站长网 2016年09月22日 17:00

微信应用号这个被人们挂念的应用,终于拼着“小程序”的外衣进入内测阶段了,新媒体人、自媒体们、创业者们,又有了忙碌的新方向。与做个APP不同,在移动互联网上... [5条相同新闻](#) - 百度快照

微信小程序成了大新闻,它会改变什么?



网易新闻 2016年09月23日 07:05

“微信小程序”原来的名字叫做“微信应用号”。2016年1月,微信事业部

好课科...

E:\快盘\todo.txt - ...

Foxmail

微信小程序宣传 doc ...

微信web开发者工具 v...

开发环境搭建



微信开发者工具



代理 ▼

◀ 返回 代理 ▼



调试类型

本地小程序项目
用于调试本地小程

公众号网页开发
用于公众号网页开



添加项目

第一次运

< 返回

添加项目

AppID

填写小程序AppID 无 AppID

项目名称

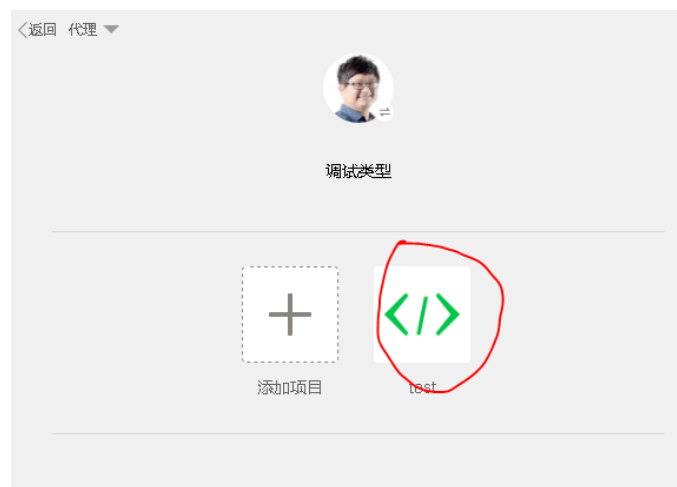
项目目录 选择

空文件夹

取消 添加项目

勾选：在当前目录中创建quick start项目

怎么把项目从启动界面删掉



开发环境搭建



开发工具界面介绍：【编辑】；【调试】，类似浏览器的F12；点击【编译】来编译并且运行项目；

项目结构说明；

修改代码后一定要先ctrl+s；

这是模拟器的运行，是用网页模拟出来的，很真机还是有差别；

修改后模拟器中立即重新加载；

调试面板的使用：在调试模式下进入；console中看编译、运行错误，代码中用console.log等输出的也显示在这里；可以在Source下设置断点，注意source中看到是编译后的js；Network中看网络通讯报文；Storage看本地存储；App_Data看应用数据；Wxml中可以进行页面结构的查看；

初始项目结构讲解



1、文件类型：

wxss是类似于css的样式文件

json是配置文件，和获取服务器端json数据没关系；

js文件是逻辑代码，不支持document等普通浏览器对象，因此也无法使用JQuery等，可以使用标准的js语法，支持ES6部分语法；

wxml是界面文件；借鉴了HTML的一些思想，但是不支持任何HTML元素。

2、文件关系

wxml、wxss、js文件名相同的成为“一组界面”；wxml描绘界面结构、wxss是页面的样式、js是逻辑代码

严格的MVC模式：js中不能直接操作wxml中的控件，js和wxml的交互（给控件属性赋值、把控件的值获取到）都是通过data和表单来进行。

3、app.js是全局脚本，可以处理应用的初始化和生命周期控制；app.json是全局的配置，pages中是app中所有的页面，所有页面都在这里注册（不用后缀），第一个页面是起始页；app.wxss是全局的样式文件，可以被子页面使用。

实现加法计算器

pages下新建一个目录（也可以多个页面放到一个目录下，但是建议一个页面一个目录）test1，再建test1.js和test1.wxml（文件名不一定和目录名一样，但是建议一样；但是js、wxml、wxss、json的文件名必须一致）

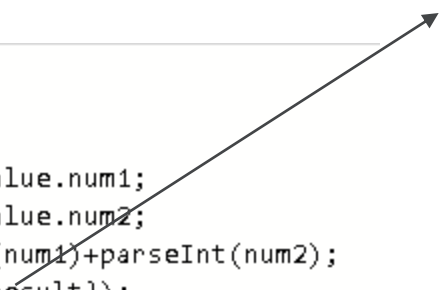
test.wxml
View

```
<view>
  <form bindsubmit="formSubmit">
    <input name="num1"/>
    <text>+</text>
    <input name="num2"/>
    <button form-type="submit">=</button>
    <text>{{result}}</text>
  </form>
</view>
```

test.js
Controller

```
Page({
  formSubmit:function(e)
  {
    var num1 = e.detail.value.num1;
    var num2 = e.detail.value.num2;
    var result = parseInt(num1)+parseInt(num2);
    this.setData({result:result});
  }
});
```

Model



美化加法计算器

新建test1.wxss样式

```
input{  
  background-color:cyan  
}  
  
.result  
{  
  font-size: xx-large;  
  color:red;  
}
```

然后再test1.wxml中给元素设置class="result"

调整app.json调整应用的标题等

初始项目代码讲解：Index

index、logs目录都是官方的例子，不是必须的，可以删掉的，注意保持app.json中页面注册的同步。

<view>是一个横向或者纵向布局子元素的容器；

关于代码风格：官方的例子是没有行结尾的分号；支持let、const等ES6中的严格模式，官方例子还是用var；

index解读：data是js传递数据给View的Model，可以在Page函数中通过data属性赋值初始值，除了在Page中给data初值外，其他地方不要直接给data赋值，而是通过页面的对象setData({motto:"rupeng"})方法这样进行“部分赋值”，setData后界面中就会立即发生变化，setData可以给界面中几乎所有控件的所有属性赋值；使用“page.data.key名字”这种方式取值；

处理点击事件使用bindtap事件；重定向wx.navigateTo({url: '../logs/logs'})；onLoad是页面加载的事件；

需要获取用户输入内容的时候用formsubmit，普通的点击事件处理用bindtap；

和js中的this作用域一样的，在所有Page的事件处理方法中的this都是Page对象，但是在其他function里调用的就不一定了，因此通常的做法是var page = this这样处理；

初始项目代码讲解：logs

var util = require('../utils/util.js')的方式可以引入外部的js文件；外部的js文件通过下面的方式把函数导出（没有导出的函数只能在内部用）：

```
module.exports = {formatTime: formatTime}
```

```
<view class="container log-list">
  <block wx:for="{{logs}}" wx:for-item="log" wx:key="*this">
    <text class="log-item">{{index + 1}}. {{log}}</text>
  </block>
</view>
```

微信小程序开发

20:00开始

昨天视频下载地址 www.rupeng.com

框架

WXML是要求严格遵守XML规范；使用`{{message}}` 这种Mustache 语法格式进行数据绑定；几乎可以用到任何控件的任何属性上；是“单项绑定”（只能把data的数据赋值给wxml的属性，不能由属性的改变反向同步到data中）

还可以“组合使用”：`<text style="color:{{color}}">+</text>`

案例：使用`setInterval`定时器搞一个progress进度条显示；`<progress percent="{{percent}}" active="{{false}}"/>` 注意不能写成`active="false"`，因为`false`是字符串，转成 `boolean` 类型后代表`true`（脑残！）。

```
onLoad:function()
{
    var page = this;
    var intervalId=setInterval(function(){
        var percent=page.data.percent;
        console.log(percent)
        percent++;
        page.setData({percent:percent});
        if(percent==100)
        {
            clearInterval(intervalId);
        }
    },100);
},
```

Mustache 语法

支持三元运算符;

支持算术运算符;

支持<、>等比较逻辑运算符;

支持字符串相加`<view>{{"hello" + name}}</view>`, 但是其实一般`<view>hello{{name}}</view>`更简单;

支持路径运算 (点出来属性, 用数组的索引): `<view>{{object.key}}
{{array[0]}}</view>`

1、`<view wx:if="{{condition}}"> True </view>` 如果condition为false，则这个view根本不会渲染到页面上，不是隐藏；案例：点击按钮切换图片显示；

2、还可以支持else：

```
<view wx:if="{{length > 5}}"> 1 </view>
```

```
<view wx:elif="{{length > 2}}"> 2 </view>
```

```
<view wx:else> 3 </view>
```

3、如果有多个控件需要一起控制是否渲染，那么可以用`<block>`包裹，注意`<block>`不会显示、不是容器，不影响布局。

可以用`hidden="{{true}}"`来隐藏view，隐藏和不渲染是不一样的；

易错：是不是写wx：易错

1、persons:[{name:"范冰冰",imgSrc:"mm1.jpg"},{name:"杨颖",imgSrc:"mm2.jpg"}]

```
<image wx:for="{{persons}}" src="/images/{{item.imgSrc}}"></image>
```

表示循环persons遍历生成image元素，每一项的默认变量名是item，也可以通过给wx:for-item赋值指定数组当前元素的变量名

2、每一项的编号默认值是index，可以通过wx:for-index修改；

3、(*)也可以将 wx:for 用在 <block/> 标签上，以渲染一个包含多节点的结构块。比如同时显示姓名、序号和图片；

4、(*)如果列表的内容会动态的增减，则设定一个wx:key属性，这个属性必须要是数据的唯一标识（思考：index可以吗）。

alt+Z自动换行

案例：上拉加载

如果页面已经到了页面顶端，再下拉刷新就执行onPullDownRefresh事件（需要在config的 [window](#) 选项中开启enablePullDownRefresh。当处理完数据刷新后，[wx.stopPullDownRefresh](#)可以停止当前页面的下拉刷新。）；如果一定到了页面底部，再向下拉就执行onReachBottom；

可以用wx.showToast({title:msg}) 弹出toast消息。不要直接showToast("aaa")；

```
persons:[{id:1,name:"范冰冰",imgSrc:"mm1.jpg"},{id:2,name:"杨颖",imgSrc:"mm2.jpg"},  
  {id:3,name:"aaa",imgSrc:"mm3.jpg"},{id:4,name:"bbbb",imgSrc:"mm4.jpg"},  
  {id:5,name:"cccc",imgSrc:"mm5.jpg"},{id:5,name:"ddd",imgSrc:"mm6.jpg"}],  
meinv2:[]
```

```
onReachBottom:function()  
{  
  var meinv2 = this.data.meinv2;  
  var persons = this.data.persons;  
  if(meinv2.length<persons.length)  
  {  
    var p = persons[meinv2.length];  
    meinv2.push(p);  
  }  
  //不要丢，只有setData才会刷新  
  this.setData({meinv2:meinv2});  
},
```


组件

1、view，横向或者纵向布局，用法和Flex布局一样

设置样式：flex-direction: row 横向（一行）布局；flex-direction:column 纵向（一列）布局。可以嵌套。必须设置display:flex;这个样式才是采用flex布局
几乎所有控件都有bindtap事件

2、(*) scroll-view 带滚动条的区域；

3、swiper：滑块儿容器。

```
<swiper style="height:600rpx" autoplay="{{true}}" interval="1000" indicator-dots="{{false}}">
```

```
  <swiper-item wx:for="{{persons}}">
```

```
    <image src="/images/{{item.imgSrc}}"/>
```

```
  </swiper-item>
```

```
</swiper>
```

属性：

indicator-dots是否显示指示点；默认false

autoplay：是否自动切换；默认false

current：当前页面的序号；

interval自动切换时间（毫秒），默认5000

duration 滑动动画时间（毫秒），默认1000

!!! bindchange：页面改变事件event.detail.current可以获取当前的序号

基础内容

1、icon 内置的图标；<icon type="success" size="45"/>，不够的图标用image

属性名	类型	默认值	说明
type	String		icon的类型，有效值：success, success_no_circle, info, warn, waiting, cancel, download, search, clear
size	Number	23	icon的大小，单位px
color	Color		icon的颜色，同css的color

2、text 文本：支持转义符“\”。显示的文字作为text的内部文本。text支持嵌套

```
<text style="color:red">rup<text style="color:green">en</text>g</text><text>.com</text>
```

3、progress 进度条

属性：percent，百分比（0-100）；show-info：bool，是否在右侧显示百分比；active：是否在进度改变的时候动态从0增长到percent的值，默认为false；stroke-width：线条宽度；color：进度条颜色；

案例：点击一下按钮增长5个值；

表单组件1

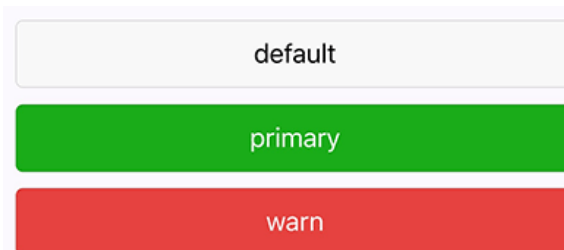
```
<view>
  <form bindreset="formReset" bindsubmit="formSubmit">
    <input name="n1"/>
    <checkbox-group name="c1">
      <checkbox value="vip"/>
    </checkbox-group>
    <button form-type="submit">提交</button>
    <button form-type="reset">重置</button>
  </form>
</view>
```



1、如果把 `<switch/>` `<radio/>` `<picker/>` 放到form中，则可以在bindsubmit事件中获得表单的内容（需要有name）；checkbox、radio必须放到checkbox-group和radio-group中，才能在form中使用（重置、获取选中状态）。

点击form中`<button form-type="submit">`的会提交表单触发bindsubmit；如`<button form-type="reset">`表单会清空，并且触发bindreset；

```
<form>
  <input value="abc"/>
  <button form-type="submit">submit</button>
  <button form-type="reset">reset</button>
</form>
```



2、button 按钮

size: 尺寸，有效值 default, mini;

type: 样式。有效值 primary, default, warn

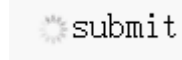
plain: 是否背景色透明；默认是false。如果属性是bool类型，那么plain和plain="{{true}}"是等价的；

disabled: 是否禁用；

loading: 是否显示“加载中”的图标，表示“正在执行中”；

form-type: 可选值submit、reset；

```
<button disabled="{{loading}}" loading="{{loading}}">
```



3、checkbox

属性：value，字符串类型，不是显示的值，而是获得的选中的值；
disabled是否禁用；checked是否选中；

如何显示文字？在旁边放文字。但是点击文字没反应。

应该用label控件，设定控件的for属性，指向checkbox的id；

问题：如何响应点击事件？这种场景用switch组件更合适；

checkbox更适合表单提交的时候做一些事情；而switch适合于立即响应

4、checkbox-group

可以把多个checkbox放到checkbox-group中，这样可以监听checkbox-group的bindchange事件，任何一个发生改变的时候就会触发事件。

通过事件的事件.detail.value获取选中的值(数组)；

```
cb1bindchange:function(event){  
  let msg ="";  
  for(let i=0;i<event.detail.value.length;i++)  
  {  
    msg+=event.detail.value[i];  
  }  
  wx.showToast({title:msg});  
},
```

表单组件3

5、radio

value: 选中的值; checked: 是否选中; disabled: 是否禁用。也可以和label一起用。

6、radio-group

多个radio放到radio-group中成为一组，一组只能有一个处于选中状态；

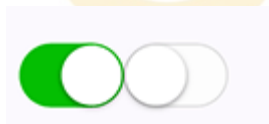
同样监听radio-group的bindchange事件，同样通过事件的事件.detail.value（字符串类型）获取选中的值

wxml中的组件，如果显示的内容是以innerText的方式（而不是属性的方式）设置，那么就可以不仅仅放文本，也可以放image等组件；

```
<label for="nan">男<image style="width:20px;height:20px" src="/images/male.png"/></label>  
<button disabled="{{true}}" type="primary" form-type="reset">重置<image style="width:20px;height:20px" src="/images/male.png"/></button>
```

表单组件4

7、switch



checked 是否选中，bool值；

type：类型，可选值：switch, checkbox。如果想用checkbox的样子，但是不想用checkbox-group，那么可以用type="checkbox"

bindchange：选中发生改变的事件，可以通过event.detail.value获取选中状态；

也通常和label一起使用



8、slider

min：最小值，默认0；max最大值，默认100；

step：步长，可被(max - min)整除

disabled：禁用；

value：当前值；

show-value：是否在右侧显示当前值；

bindchange：值改变事件；event.detail.value获得当前值；

表单组件5

9、input 输入框

value：初始内容；

type：输入的内容。有效值：text, number, idcard, digit, time, date。模拟器不支持，应该在真机上影响软键盘布局；

password：bool，是否是密码框；

placeholder：为空时候的提示文字； placeholder-style、 placeholder-class：placeholder的内联样式和样式类；用input的时候最好都设定placeholder，这样用户知道这是咋回事。

disabled：禁用；

maxlength：最大输入长度；

auto-focus：是否自动聚焦；

bindfocus、 bindblur：获得/失去焦点的事件，可以通过event.detail.value获取输入的值；

案例：做一个登录功能，带“记住密码”的switch；

10、 textarea 多行输入 拥有input所有属性、事件

额外属性 auto-height是否自动高度自适应，设置auto-height时，样式的height属性不生效


```
...  
<form bindsubmit="formSubmit">  
  <input value="{{phoneNum}}" name="phoneNum" place<br>  
  <input value="{{password}}" name="password" place<br>  
  <switch name="remPwd" id="rememberPwd" /><label 1<br>  
  <button form-type="submit">登录</button>  
</form>
```

```
Page({  
  data:{phoneNum:"",password:""},  
  formSubmit:function(e)  
  {  
    let phoneNum = e.detail.value.phoneNum;  
    let password = e.detail.value.password;  
    if(phoneNum=="110"&&password=="123")  
    {  
      wx.showToast({title:"登录成功"});  
    }  
    else  
    {  
      //绑定的单向的，界面中的改变并不会改变data  
      //setData()如果设置的值和现有的值一样，那么就不会更新界面  
      //只能先改成一个和以前不一样的，再改回来  
      this.setData({phoneNum:"1",password:"1"});  
      this.setData({phoneNum:"",password:""});  
      wx.showToast({title:"登录失败"});  
    }  
  }  
});
```

表单组件6



11、picker 选择器，可以选择日期、时间以及列表数据。更符合手机的使用习惯；

1) mode属性有三个可选值： selector（列表）； time（时间）； date（日期）；

2) 通用属性、事件：

value当前选中的值： mode为 selector 时，为数字，表示选择了 range 中的第几个，从0开始。date、time时为日期；

bindchange： 选项发生改变的事件， event.detail.value获取value的值

3) 特有属性：

mode = selector时： range为一个数组，表示“可选数据”

mode=date 时： start、end 为起止时间； fields（有效值 year,month,day）表示选择的粒度；

mode = time时： start、end为起止时间；

注意，picker并不是可见的控件，需要把可见控件作为picker的子元素，可以用按钮、view、文本等几乎任意控件。

案例：选择省、市。选中的省、市会显示到“请选择”上面



取消			确定
2013年	03月		
2014年	04月		
2015年	05月		
2016年	06月	01日	
2017年	07月	02日	
2018年	08月	03日	
2019年	09月	04日	

12、navigator 超链接

可以包含其他组件

属性：

url：应用内的跳转链接

redirect：bool类型，默认false。true为重定向（无法回退回来），false为导航到新页面（可以回退回来，最多5层）

hover-class：点击时候的样式；

url 中可以像普通网址一样带 querystring：

/test1/test1?name=yzk&age=5

然后在目标页面中通过onLoad的参数来获取参数：

```
onLoad: function(options) {  
    this.setData({  
        title: options.name,  
        age:options.age  
    })  
}
```

表单组件8

13、image（讲如何看微信的文档）

src 图片资源地址；

mode: image组件默认宽度300px、高度225px。图片的缩放模式，可选

值: scaleToFill（默认值，不保持纵横比拉伸填满 image 元素）；

aspectFit（保持纵横比，完全显示图片，短的边会有空白）；

aspectFill（保持纵横比，短的边填满，宽的边会截掉）

```
<image src="/images/mm6.jpg" mode="aspectFill" style="width:200px;height:100px;background-color:red"/>
```



原图



scaleToFill



aspectFit



aspectFill

表单组件9

14、audio 音乐播放。一个app安装包最大1M。

id: 标志; 给wx.createAudioContext()创建用

src: 音频文件, 播放网络地址, 因为app大小有限制;

loop: bool, 是否循环播放, 默认false;

controls: bool, 是否显示播放控制条, 默认true; (官方的例子代码我不推荐)

poster: 封面图片地址 (注意有的网址是防盗链了); name: 显示的歌名; author: 作者名;

事件: binderror (错误, event.detail. MediaError.code获得错误码, 和文档中的可选值比较)、bindplay (开始播放、继续播放)、bindpause (暂停)、bindtimeupdate(event.detail.currentTime获取当前播放进度, event.detail.duration获取音乐时长。都是小数类型, 以秒为单位)、bindended (播放结束)

案例: 给src赋值<http://static.rupeng.com/static/wxappdemo/fs.mp3>播放音乐, 并且显示controls。注意监听binderror事件;

表单组件10



AudioContext

使用AudioContext对audio组件进行播放、暂停、定位等控制；

调用`wx.createAudioContext('myAudio')` 根据audio的id创建AudioContext对象；

AudioContext的方法：`play`播放；`pause`暂停；`seek(10)`跳转到第10秒；

案例：播放器和slider双向改变。

```
Page({
  data: {
    audioProgress: 0,
    audioDuration: 0
  },
  onLoad: function() {
    this.player = wx.createAudioContext("player");
  },
  audioTimeUpdate: function(e) //音频播放进度改变slider进度
  {
    let duration = e.detail.duration;
    let progressValue = parseInt(e.detail.currentTime*100/duration);
    this.setData({audioProgress: progressValue, audioDuration: duration});
  },
  sliderChange: function(e) //slider进度改变播放进度
  {
    let duration = this.data.audioDuration;
    let progress = duration*e.detail.value/100;
    this.player.seek(progress);
  }
});
```

```
<view>
  <audio id="player" src="http://static.rupeng.com/static/wxappdemo/fs.mp3"
    controls="{{true}}" bindtimeupdate="audioTimeUpdate"/>
  <slider value="{{audioProgress}}" bindchange="sliderChange"/>
</view>
```

表单组件11

14、video（别写错成vedio）

不支持所有视频格式；只支持开放格式（文档没有详细说明）

video标签认宽度300px、高度225px

id：标识；src 视频地址；controls，是否显示播放控制界面；

danmu-list：弹幕对象数组，对象包含text、color、time三个属性；

danmu-btn：bool是否显示弹幕切换按钮，默认false，只能在初始时赋值

enable-danmu：是否显示弹幕，默认false，只能在初始时赋值；

autoplay：bool，是否自动播放，默认false

事件：bindplay、bindpause、bindended、binderror（可以通过event.detail.errMsg 获取文本类型的报错信息）

测试视频<http://static.rupeng.com/static/wxappdemo/wxapp.mp4>

VideoContext: wx.createVideoContext(videoId)

方法：play、pause、seek(second)、 在当前位置发送弹幕（不能是暂停状态）sendDanmu({text:"如鹏网观光团路过",color:"blue"});

API

- 1、API 接口一般都接受一个OBJECT作为参数。
- 2、api一般都是异步（什么是异步，如何异步）的，同步的以sync结尾。
- 3、OBJECT中可以指定success, fail, complete来接收接口调用结果。 success 接口调用成功的回调函数;fail 接口调用失败的回调函数 ;complete 接口调用结束的回调函数（调用成功、失败都会执行）
- 4、微信小程序文件的三个类型：
 - 1) 微信小程序资源文件，也就是源代码中的文件；
 - 2) 本地临时文件：程序退出后消失，在本次运行期间可用；
 - 3) 本地持久文件，调用wx.saveFile把临时文件持久化保存，以后也可以用。

不同组件、api对着三个类型的支持不一样。video、audio支持本地临时文件和持久文件；image是同时支持这三者；

微信小程序无法直接读写用户手机上的文件，也无法读写其他app的文件，只能读写给自己划定的一个区域的文件。

1、微信小程序和普通app开发一样，手机端只负责数据的展示和收集用户输入，数据、逻辑处理等都放到服务器端。手机端和服务器之间采用https（模拟器支持http，正式环境只支持https，因为更安全）、websocket（高效实时通讯）两种通讯方式。正式环境中只能向小程序服务器端管理中设置的域名发送请求； HTTP。

2、微信小程序支持https请求、websocket通讯，还支持上传、下载。

3、使用wx.request发出https请求（类似ajax），同时只能有5个请求；

url（服务器接口地址，肯定是可以跨域的）

data（请求参数对象）

header（设置请求的报文头，对象，不能设置Referer）

method（请求方式：POST、GET、PUT、DELETE等，支持RESTFUL）

success（返回值的data属性为返回的内容）、fail、complete

如果服务器端返回ContentType为application/json，则success的res.data为反序列化的javascript对象，否则是响应报文体；

测试：向<http://www.rupeng.com/Main/WhoLearning>发请求（只支持POST），并且支持下拉刷新；

1、wx.uploadFile(OBJECT)上传文件

使用场景：app设置用户头像等；

参数：

url 上传服务器地址

filePath 本地文件路径

name 文件的名称，是服务器端文件的请求表单的名称；

header 请求报文头，不能设置Referer

formData 额外的表单数据，可以传递登录用户token等；

success (返回对象的属性{data:"返回报文体文本", statusCode:200})、fail、complete

测试：可以使用wx.chooseImage选择一幅图片；

使用如鹏的测试服务器接口：<http://wwwtest.rupeng.cn:8081/Test/UploadAvatar>

2、wx.downloadFile(OBJECT) 下载文件。app最大1M，有的文件就要运行的时候再下载下来用

url：服务器地址；

header：请求报文头

success：下载成功后的临时文件，res = {tempFilePath: '文件的临时路径'}在小程序本次启动期间可以正常使用，如需持久保存，需再调用 wx.saveFile (后面讲)。文件的路径是 "wxfile://tmp_132aa.png"这样的虚拟地址 (tmp开头是临时文件)。

fail、complete

案例：显示下载的图片到image

如果路径中有中文或者特殊符号，要使用 encodeURIComponent() 进行编码：

<http://static.rupeng.com/static/wxappdemo/微信小程序Day1.ppt>

1、wx.chooseImage(OBJECT)从本地相册选择图片或使用相机拍照

参数(除了success其他都是可选参数)：

count：最多可以选择的图片张数，默认9。可以一次性选多个图片

sizeType：字符串数组，代表选择什么类型的图，可选值" original "(原图)、" compressed "(压缩图)可以两个都选。模拟器不支持。

sourceType：字符串数组，图片来源，可选值" album "(相册)、" camera "(相机)，可以两个都选。模拟器不支持。

success：参数的tempFilePaths属性为图片路径的数组，注意获得的是临时文件路径（因为有可能是相机照片），如果需要永久保存，要用**wx.saveFile**。

也可以把选择的文件显示到image组件中；

2、预览图片，不能是本地图片

```
wx.previewImage({  
  current: 'http://wwwtest.rupeng.cn:8081/imgs/new/logo.png', // 当前显示图片的  
  http链接  
  urls: [  
    "http://wwwtest.rupeng.cn:8081/imgs/new/logo.png",  
    "http://images2015.cnblogs.com/2.png",  
    "http://images2015.cnblogs.com/3.png"] // 需要预览的图片http链接列表  
})
```

3、获得图片信息（宽高）

wx.getImageInfo(OBJECT)，可以是临时、永久路径，也可以是项目中的路径，也可以是网络路径；

1、录音和音频播放控制貌似模拟器还不支持

2、wx.playBackgroundAudio

背景音乐播放。只能有一个正在播放的背景音乐。会在个人中心提供展示控制页面，程序只能控制标题、封面等；

参数：

dataUrl，音乐路径，只支持网络路径，不能下载到本地再播放，应该是基于版权保护考虑；

title：音乐标题

coverImgUrl：封面图片

success、fail、complete

wx.playBackgroundAudio({

dataUrl: 'http://static.rupeng.com/static/wxappdemo/fs.mp3',

title: '',

coverImgUrl: ''

})

3、wx.getBackgroundAudioPlayerState(OBJECT)获得音乐播放信息

success、fail、complete

success的属性：duration（音频的秒数）、currentPosition（当前位置，s）、status（播放状态，2：没有音乐在播放，1：播放中，0：暂停中）、dataUrl（音乐路径）

downloadPercent（下载进度，最大100，因为支持边下载边播放）



正在播放
未知 - 未知



3、 `wx.pauseBackgroundAudio()` 暂停播放背景音乐；

4、 `wx.seekBackgroundAudio(OBJECT)` 改变背景音乐播放进度

OBJECT参数： position、 success、 fail、 complete

5、 `wx.stopBackgroundAudio()` 停止播放背景音乐

6、 事件监听

`wx.onBackgroundAudioPlay(CALLBACK)`

`wx.onBackgroundAudioPause(CALLBACK)`

`wx.onBackgroundAudioStop(CALLBACK)`

如果做音乐播放器、蜻蜓FM之类的应该用背景音乐播放；

wx.chooseImage、wx.startRecord、wx.downloadFile等都返回的是临时文件，微信小程序退出后文件就销毁了，这里介绍的api就可以把临时文件持久保存。不支持访问用户手机上的其他文件、其他app的文件；

1、wx.saveFile(OBJECT) 临时文件保存为持久文件

参数：tempFilePath 临时文件路径； success（参数属性savedFilePath为保存的本地路径）、fail、complete

2、wx.getSavedFileList(OBJECT) 获得本app保存的持久文件列表

success、fail、complete

success返回值：errMsg； fileList（对象数组，每个元素的属性：

filePath 文件本地路径； createTime 文件创建时间戳，从1970/01/01 08:00:00 到当前时间的秒数； size 文件大小，字节为单位）

时间戳转换为字符串的方法见资料区域getLocalTime(nS)

编写一个把字节转换为“**k、**M”的函数getFileSizeName(size)。提示js中使用toFixed(2)格式化小数点位数

缓存用户看的图片，加载用户缓存的图片；

持久化文件管理2

3、 **wx.getSavedFileInfo(OBJECT)**

获得已经保存的文件的信息

filePath: 文件路径; success (参数属性: errMsg、size、createTime)、fail、complete

4、 **wx.removeSavedFile(OBJECT)**

删除文件

filePath; success、fail、complete

5、 **wx.openDocument(OBJECT)**

打开文件, 支持doc, xls, ppt, pdf, docx, xlsx, pptx等格式。不支持app中的文件。

filePath; success、fail、complete

测试: <http://static.rupeng.com/static/wxappdemo/微信小程序Day1.ppt>

注意路径中如果有中文wx.downloadFile中的url要进行encodeURIComponent

案例: 点击一个按钮, 下载ppt (既可以打开临时的, 也可以打开永久的), 然后打开

wx.chooseVideo(OBJECT)

拍一段视频或者选择手机中的视频，返回视频的临时文件路径。模拟器不支持拍视频。

object参数属性（都不是必须的）

sourceType：字符串数组，视频源，可选值：'album', 'camera'

maxDuration：拍摄视频最长拍摄时间，单位秒。最长支持60秒

camera：字符串数组，前后摄像头选择；默认为前后都有，可选值
'front', 'back'

success、**fail**、**complete**

success返回对象的属性：

tempFilePath（临时文件路径）、**duration**（视频秒数）、**size**（文件字节大小）、**height**（视频高度）、**width**（视频宽度）

每个页面的**data**是给每个页面用的
还可以设置全局数据，这个数据在**app**运行期间都存在，**app**退出后消失。
用来实现程序运行期间各个页面数据的共享。类似于**Session**。
app对象的**globalData**属性，可以在**App.js**中通过**this**获得**app**对象，在非**App.js**的其他地方通过全局函数**getApp()**来获得**app**对象。
案例：把当前登录用户信息保存到**globalData**中；

可以把字符串或者js对象保存到本地缓存中，app退出后还存在，下次还可以用。应用场景：获取微博的缓存，再次打开还可以看到；接口分为同步接口(**Sync**)和异步接口。数据量如果不大用同步接口就行；

1、**wx.setStorage(OBJECT)** 异步设置缓存，会覆盖旧的

OBJECT参数说明：

key； data：Object/String类型； success、fail、complete

2、**wx.setStorageSync(KEY,DATA)** 同步设置缓存，如果出错会出现异常

```
try {  
    wx.setStorageSync('key', 'value')  
} catch (e) {  
}
```

3、**wx.getStorage(OBJECT)** 异步获取数据

key、 success、 fail、 complete

success参数属性data

4、**wx.getStorageSync(KEY)** 同步获取数据，通过返回值获取，如果出错会出现异常

- 5、 **wx.getStorageInfo(OBJECT)**异步获取当前storage信息
success、 fail、 complete
success参数属性： keys (key数组) ； currentSize (总空间大小， kb为单位) ； limitSize (限制最大空间大小) ；
- 6、 **wx.getStorageInfoSync()** 同步获取当前storage信息
- 7、 **wx.removeStorage(OBJECT)** 根据key异步移除storage
参数： key、 success、 fail、 complete
- 8、 **wx.removeStorageSync(KEY)**根据key同步移除storage
- 9、 **wx.clearStorage()**异步清理缓存；
- 10、 **wx.clearStorageSync()**同步清理缓存；

1、wx.getLocation(OBJECT) 获取位置

type：默认为 wgs84 返回 gps 坐标，gcj02 返回可用于 wx.openLocation 的坐标。模拟器获取的坐标在同一个城市获得

同一个坐标，不准确。

2、wx.chooseLocation(OBJECT) 地图上选择位置，模拟器貌似有bug。具体看文档

3、wx.openLocation(OBJECT)

使用微信内置地图查看位置（只有经纬度必选）

参数：latitude、longitude、scale（缩放级别、范围1~28，默认为28，越小越详细），name（名字）、address（地址描述）

success、fail、complete

```
wx.getLocation({
  type: 'wgs84',
  success: function(res) {
    var latitude = res.latitude
    var longitude = res.longitude
    var speed = res.speed
    var accuracy = res.accuracy
  }
})
```

1、 **wx.getNetworkType(OBJECT)** 获取网络类型。场景：非wifi下不下载，避免耗费流量；

success、fail、complete

success参数属性：networkType（网络类型2g，3g，4g，wifi）

2、 **wx.getSystemInfo(OBJECT)**。获取设备信息。

场景：显示好友手机类型；组件高度正好等于屏幕高度；

success、fail、complete

success参数属性：model（手机型号）、pixelRatio（设备像素比）、windowWidth（窗口宽度）、windowHeight（窗口高度）、language（设备语言）、version（微信版本号）

wx.getSystemInfoSync() 同步获取设备信息

3、 **wx.onAccelerometerChange()**监听重力加速度变化，模拟器不支持。

应用场景：摇一摇；模拟赛车方向盘；

wx.onCompassChange(CALLBACK) 罗盘、指南针；

4、 **拨打电话**：场景：做一个酒店app，点击前台电话直接拨号

wx.makePhoneCall({ phoneNumber: '1340000' //仅为示例，并非真实的电话号码 })

success、fail、complete需要经过用户确认。



1、 **wx.showToast(OBJECT)** 显示toast消息。用来提示不需要用户确认的消息，不适合报错。

参数：（只有title必填）

title：消息；

icon：可选值"success"、"loading"

duration：提示的延迟时间，单位毫秒，默认：1500, 最大为10000

2、 **wx.showModal(OBJECT)** 显示模态对话框，必须是严重的消息，需要用户确认的消息才能用模态对话框

title 字符串 必填 提示的标题

content 字符串 必填 提示的内容

showCancel bool 否 是否显示取消按钮，默认为 true

cancelText 字符串 否 取消按钮的文字，默认为"取消"

cancelColor HexColor 否 取消按钮的文字颜色，默认为"#000000"

confirmText 字符串 否 确定按钮的文字，默认为"确定"

confirmColor 颜色 否 确定按钮的文字颜色，默认为"#3CC51F"

success 否 接口调用成功的回调函数，返回res.confirm==1 时，表示用户点击确定按钮



交互反馈API 2

3、 **wx.showActionSheet(OBJECT)**

参数属性：

itemList：选项的字符串数组

itemColor：钮的文字颜色，默认为"#000000"

success

success参数： cancel (bool，是否选择了【取消】)
； tapIndex (用户点击的按钮的序号)

```
wx.showActionSheet({  
  itemList: ['A', 'B', 'C'],  
  success: function(res) {  
    if (!res.cancel) {  
      console.log(res.tapIndex)  
    }  
  }  
})
```



1、设置导航条文字

即可通过json文件配置，也可以通过代码动态修改

```
wx.setNavigationBarTitle({  
  title: '用户头像'  
})
```

不能在onLoad、onShow中调用，必须在生命周期的onReady及之后才能

```
wx.setNavigationBarTitle
```

2、**wx.showNavigationBarLoading()**

导航条显示加载动画

3、**wx.hideNavigationBarLoading()**

导航条隐藏加载动画

应用场景：下拉刷新显示正在刷新；

1、页面导航

```
wx.navigateTo({  
  url: 'test?id=1'  
})
```

最多五层页面导航，和navigator组件效果一样

2、页面重定向

```
wx.redirectTo({  
  url: 'test?id=1'  
})
```

无法后退回来

3、**wx.navigateBack(OBJECT)**后退

使用navigator或者wx.navigateTo导航页面的后退

如果用**wx.navigateBack()**则返回上一级；

wx.navigateBack({delta:2}) 返回到上上一页，如果超过了前面页面的个数，则返回首页；

可以使用getCurrentPages() 全局函数 获取当前页面栈的实例，以数组形式按栈的顺序给出，第一个元素为首页，最后一个元素为当前页面。

- 1、下拉刷新、上拉加载分别使用 `onPullDownRefresh`、`onReachBottom` 事件；
- 2、触摸屏操作事件（*唯一的冒泡事件，写成 `bind**` 就是无法阻止冒泡，写成 `catch**` 可以阻止冒泡）
 - `bindtap`：手指触摸后马上离开
 - `bindlongtap`：长按。手指触摸后，超过350ms再离开
 - `bindtouchstart`：手指触摸动作开始
 - `bindtouchmove`：手指触摸后移动
 - `bindtouchcancel`：手指触摸动作被打断，如来电提醒，弹窗
 - `bindtouchend`：手指触摸动作结束

事件参数：

- 1) `e.target`：冒泡事件的源控件；
- 2) `e.currentTarget`：当前事件绑定的控件。
- 3) 在 `touchstart`、`touchmove` 事件中可以通过 `e.touches` 数组获取多个手指（比如放大、缩小、节奏大师）当前的坐标，返回的是一个 **Touch 对象数组**

Touch 对象数组属性

- 1) identifier 数字类型，触摸点的标识符
- 2) pageX, pageY 距离文档左上角的距离，文档的左上角为原点，横向为X轴，纵向为Y轴
- 3) clientX, clientY距离页面可显示区域（屏幕除去导航条）左上角距离，横向为X轴，纵向为Y轴

常用案例 “检查上下左右滑动方向”

- 1、监听几个事件，如果内容铺不满宽高，则无法在整个页面监听事件，因此需要调用wx.getSystemInfo获取窗口高度，然后通过setData设置给view的样式，调整页面高度；
- 2、在touchstart中使用startPos变量记录起始位置，touchmove中一直更新endPos，touchend中判断一下两个点之前的位置关系就知道是哪个方向的滑动了（画图）为了避免轻微的颤动带来的变化，如果x、y变化值绝对值（Math.abs）的和小于一个值则不处理。

自定义属性

支持类似于HTML5的自定义data属性：这些数据将会通过事件传递给SERVICE (js)。 书写方式： 以data-开头，多个单词由连字符-链接，不能有大写如data-element-type，最终在 event.target.dataset 中会将连字符转成驼峰elementType。

```
<view data-alpha-beta="1" data-alphaBeta="2" bindtap="bindViewTap"> DataSet Test </view>
```

```
bindViewTap:function(event){  
    event.target.dataset.alphaBeta === 1 // - 会转为驼峰写法  
    event.target.dataset.alphabeta === 2 // 大写会转为小写  
}
```

太复杂，记得除了data-不要用-、不要写大写就行了，代码获取：

event.target.dataset.alphabeta

案例：点击不同的人的头像显示不同信息，共用一个方法

样式、布局

样式

wxss是样式文件，类似于CSS；目前支持如下样式，不支持级联样式

选择器	样例	样例描述
<code>.class</code>	<code>.intro</code>	选择所有拥有 class="intro" 的组件
<code>#id</code>	<code>#firstname</code>	选择拥有 id="firstname" 的组件
<code>element</code>	<code>view</code>	选择所有 view 组件
<code>element, element</code>	<code>view</code> <code>checkbox</code>	选择所有文档的 view 组件和所有的 checkbox 组件
<code>::after</code>	<code>view::after</code>	在 view 组件后边插入内容
<code>::before</code>	<code>view::before</code>	在 view 组件前边插入内容

可以用内联样式 `<view style="color:{{color}};" />` 但是不建议使用
指定类样式 `<view class="normal_view" />` 也可以同时指定多个样式
微信小程序特有的单位（推荐）rpx，宽度被固定成750rpx，高度不定；

Flex布局是“弹性布局”，是HTML5中的新的替代盒子模型的布局，更适合移动端开发。提供子元素横向或者纵向布局，微信小程序推荐用Flex布局。

使用前，需要把容器设置为flex布局：`display: flex`。设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。也就是不要把盒子模型和flex布局混用。

Flex布局的子元素叫做item(项)。

<http://www.runoob.com/w3cnote/flex-grammar.html>

Flex容器的属性：

1) flex-direction代表排列方向：row（横向，默认值）、column（纵向）。还有两个用的较少：row-reverse（横向反向）、column-reverse（纵向反向）

2) (*) flex-wrap，如果横向排不下，如何换行。nowrap（默认值） | wrap | wrap-reverse。

Flex布局2

3) justify-content: 子元素的横向 (flex-direction=row)/纵向 (flex-direction=column) 对齐方式, 可选值: flex-start (左对齐) | flex-end (右对齐) | center (居中) | space-between (两端对齐, 项目之间的间隔都相等) | space-around (每个项目两侧的间隔相等)

flex-start



flex-end



center



space-between

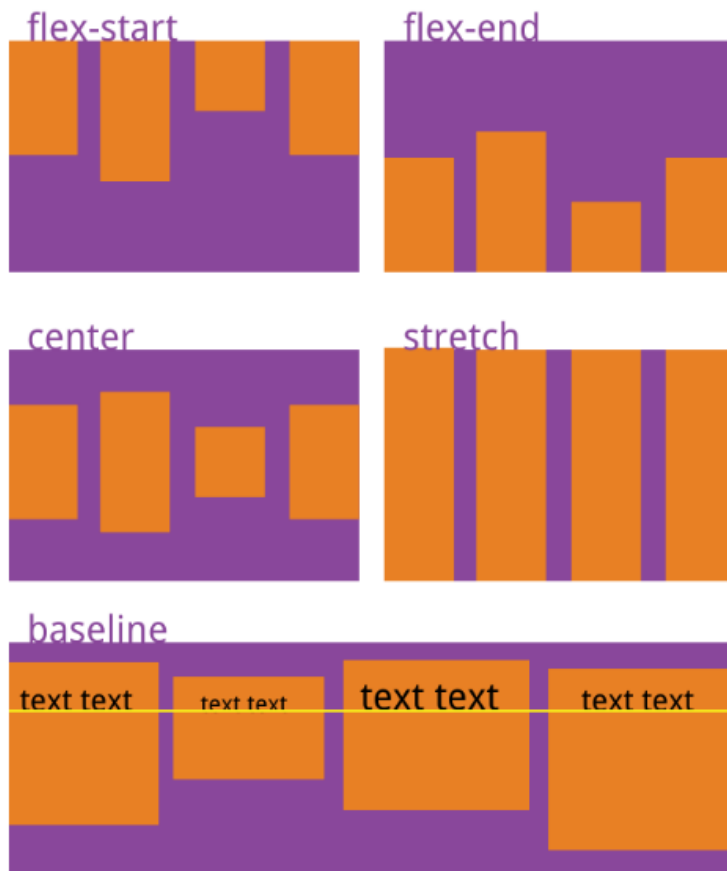


space-around



Flex布局3

4) align-items: 子元素的纵向 (flex-direction=row)/ 横 (flex-direction=column) 对齐方式, 可选值: flex-start (顶部对齐) | flex-end (底部对齐) | center (居中) | baseline (项目的第一行文字的基线对齐) | stretch (如果项目未设置高度或设为auto, 将占满整个容器的高度。默认



Flex布局-子项目样式

子项目的样式（标注在子项目上）

flex-grow：项目的放大比例，默认为0。子项目的尺寸是在**原尺寸基础上再加上**
“**(该子项目的flex-grow占总子项目flex-grow之和的百分比)*剩余空间**”。用
屏幕尺计算验证一下。。常见值：全1（平均拉伸）；一个1其他0（1的拉伸）

```
<view style="display:flex;flex-direction:row;flex-wrap:wrap;">
  <button style="flex-grow:1">111</button>
  <button style="flex-grow:2">222</button>
  <button style="flex-grow:1">333</button>
</view>
```



```
<view style="display:flex;flex-direction:row;flex-wrap:wrap;">
  <button style="flex-grow:0">111</button>
  <button style="flex-grow:0">222</button>
  <button style="flex-grow:0">333</button>
</view>
```



```
<view style="display:flex;flex-direction:row;flex-wrap:wrap;">
  <button style="flex-grow:0">111</button>
  <button style="flex-grow:1">222</button>
  <button style="flex-grow:0">333</button>
</view>
```



```
<view style="display:flex;flex-direction:row;">
  <button style="flex-grow:1">111</button>
  <button style="flex-grow:1">222</button>
  <button style="flex-grow:1">333</button>
</view>
```



Flex布局-子项目样式

(*)**flex-shrink** : 如果空间不足, 子项目的尺寸是在原尺寸基础上再**减去**“(子项目的flex-grow占总子元素flex-grow之和的百分比)*不足的空间空间”。

align-self : 设置与其他项目不一样的对齐方式, 相当于覆盖了父元素的align-items属性, 可选值: auto (默认值)、flex-start、flex-end、center、baseline、stretch;

驾考宝典

需求说明

演示一下已经完成的功能



需求说明



为了避免浪费流量，题库要缓存

接口说明



1、驾照分为c1、c2、a1、b1、a2、b2等类型， c1c2、a1b1、a2b2 各共享一套题库；考试分为科目1、2、3、4，科目1、4是上机考试都是选择题（有单选、有多选），科目2、3是上路考试；

2、获取答案 <http://jiakaoapi.rupeng.cn/api/Question/GetAnswers>
返回下面接口答案的说明。

3、获取题库

<http://jiakaoapi.rupeng.cn/api/Question?subject=1&model=c1c2>

subject可选值1、4

model可选值c1c2、a1b1、a2b2

有的题目有图片，图片位于<http://static.rupeng.com/jiakaoimgs>路径下

开始开发！