



第一章 走进微信小程序的世界

第1节 微信小程序课程介绍

简介：介绍本套课程的大纲

- 学习这套课程需要html5+css3+js基础
- 学后水平：可独立开发小程序项目

第2节 小程序是什么？小程序应用前景

简介：介绍微信小程序以及它的应用前景

第3节 微信小程序的注册流程

简介：讲解微信小程序的注册流程

- 1.去微信公众平台注册小程序账号
- 2.获得appid和appsecret

第4节 微信开发者工具安装及使用讲解

简介：介绍微信小程序工具安装以及界面的使用情况

- 出现合法域名校验错误

Thu Aug 15 2019 12:01:48 GMT+0800 (中国标准时间) request 合法域名校验出错

如若已在管理后台更新域名配置，请刷新项目配置后重新编译项目，操作路径：“详情-域名信息”

✖ ▶ <http://op.juhe.cn> 不在以下 request 合法域名列表中，请参考文档：<https://developers.weixin.qq.com/dev/framework/ability/network.html>

解决：详情里勾选“不校验安全域名、web-view 域名、TLS 版本以及 HTTPS 证书”

第5节 小程序目录结构分析

简介：介绍小程序项目目录的结构

小程序的主体部分：

- app.js（小程序主要逻辑层，用来注册小程序）
- app.json（小程序公共配置）
- app.wxss（小程序公共样式）

第6节 小程序配置文件分析

简介：分析app.json、app.wxss等配置文件



小D课堂

愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第二章 开始第一个小程序之旅

第1节 新建一个小程序及框架组成介绍

简介：如何创建一个小程序及框架组成介绍

- 新建页面的步骤
 - 右键新建page生成页面的框架
 - 在app.json中的pages添加我们新建页面的路径

第2节 小程序生命周期和页面生命周期讲解

简介：详细介绍小程序的生命周期

- 小程序启动时先执行onLaunch>onShow,然后在执行页面onLoad>onShow>onReady
- 小程序生命周期：
 - onLaunch

- onshow
 - onhide
- 页面生命周期：
 - onload
 - onshow
 - onready
 - onhide
 - onunload

第3节 了解小程序自适应尺寸单位rpx

简介：介绍小程序的自适应尺寸单位rpx

rpx单位：是可以根据屏幕宽度进行自适应

尺寸单位

- rpx (responsive pixel)：可以根据屏幕宽度进行自适应。规定屏幕宽为750rpx。如在 iPhone6 上，屏幕宽度为 375px，共有750个物理像素，则 $750\text{rpx} = 375\text{px} = 750\text{物理像素}$ ， $1\text{rpx} = 0.5\text{px} = 1\text{物理像素}$ 。

| 设备 | rpx换算px (屏幕宽度/750) | px换算rpx (750/屏幕宽度) |
|--------------|--------------------------------|-------------------------------|
| iPhone5 | $1\text{rpx} = 0.42\text{px}$ | $1\text{px} = 2.34\text{rpx}$ |
| iPhone6 | $1\text{rpx} = 0.5\text{px}$ | $1\text{px} = 2\text{rpx}$ |
| iPhone6 Plus | $1\text{rpx} = 0.552\text{px}$ | $1\text{px} = 1.81\text{rpx}$ |

第4节 如何使用flex弹性布局

简介：讲解flex弹性布局的使用

- 在父容器上加上display: flex，将我们父容器编程弹性容器

```
.sort_box {  
  display: flex;  
  flex-direction: row; //水平排列  
  margin: 0 30rpx;  
  justify-content: space-around; //每个项目的两端间隔相等  
}
```

flex布局博客: <http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>



小D课堂 愿景: "让编程不在难学, 让技术与生活更加有趣" 更多课程请访问xdclass.net

第三章 核心知识数据绑定与事件交互

第1节 必备知识点之数据绑定的概念和基础

简介: 详细分析小程序数据绑定的基础和概念

- Mustache 语法 (双大括号)

```
{{textstr+2}}
```

- 单向数据绑定

第2节 详细讲解动态数据绑定

简介: 讲解小程序数据的获取和赋值

- 小程序data赋值

```
this.setData(对象)
```

- 获取data里的值

```
this.data.(key值)
```

第3节 深度了解小程序的条件渲染

简介：讲解小程序的条件渲染

- wx:if为true就显示元素，false不显示
- 语法

```
<view wx:if="{{score<60}}">不及格</view>
<view wx:elif="{{score>=80}}">优秀</view>
<view wx:else>良好</view>
```

第4节 深度了解小程序的列表渲染

简介：讲解小程序的列表循环渲染

- wx:for绑定数组，wx:for-item值为当前项目的变量名，wx:for-index值为当前下标的变量名

```
<block wx:for="{{list}}" wx:for-item="course" wx:for-index="idx">
  <view class="item">
    <view class="item_img">
      <image src="{{course.img}}"></image>
    </view>
    <view class="item_name">{{course.name}}</view>
    <view>{{idx}}</view>
  </view>
</block>
```

- 列表渲染文档<https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/list.html#wx:key>

第5节 讲解绑定事件与事件交互

简介：讲解如何在元素上绑定事件以及事件交互

- bind+事件

```
<button bindtouchstart="changeData" bindtouchend="tmove"
bindlongpress="lpress">改变数据</button>
```

- 需要在js里写对应的方法

```
//点击事件
changeData:function(){
  this.setData({
    textStr:"hello 小程序"
  })
},
//触摸移动
tmove:function(){
  console.log("touchend");
},
//长按触发
lpress:function(){
  console.log("longpress")
},
```

- 官方事件文档<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>

第6节 深入了解事件机制--catch与bind

简介：讲解catch与bind事件机制以及捕获阶段和冒泡阶段

- bind会应用到冒泡机制，catch会阻止事件冒泡

```
<view class="box" bindtap="containerTap">
  <text catchtap="txtTap">小程序</text>
</view>
```

- 捕获阶段的bind与catch,绑定事件需要加上冒号

```
capture-bind:tap(绑定事件)
capture-catch:tap(绑定事件)
```

- 官方事件文档<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>



第四章 小程序基础组件及常用组件介绍

第1节 小程序基础组件介绍

简介：介绍小程序基础组件有哪些

- 基础组件

- icon

```
<icon type="warn" size="100rpx" color="red"></icon>
```

- progress

```
<progress percent="30" show-info="true" border-radius="10" stroke-  
width="50rpx" activeColor="red" backgroundColor="#000" active="true">  
</progress>
```

- rich-text

```
<rich-text nodes="{{textStr}}"></rich-text>
```

- text

```
<text selectable="true">小程序</text>
```

- 官方文档：<https://developers.weixin.qq.com/miniprogram/dev/component/text.html>

第2节 小程序常用表单组件介绍（一）

简介：介绍小程序常用的表单组件

- 常用表单组件如下

- button
 - checkbox

- form
- input
- label
- radio
- textarea
- 表单组件<https://developers.weixin.qq.com/miniprogram/dev/component/button.html>

第3节 小程序常用表单组件介绍（二）

简介：介绍小程序常用的表单组件

- 常用表单组件如下
 - picker组件

```
<picker bindchange="bindPickerChange" value="{{index}}" range="{{array}}">
  <view class="picker">
    当前选择: {{array[index]}}
  </view>
</picker>
```

- slider组件

```
<slider min="50" max="200" show-value="true" step="5"></slider>
```

- switch组件

```
<switch bindchange="bindSwitchChange" checked="true"></switch>
```

第4节 小程序媒体组件介绍

简介：介绍小程序媒体组件

- audio组件


```
<audio id="myAudio" poster="{{poster}}" name="{{name}}" author="{{author}}" src="{{src}}" controls></audio>

<button bindtap="audipplay">播放</button>
<button bindtap="audippause">暂停</button>
<button bindtap="audioseek">设置进度条</button>
```

- video组件

```
<video src="https://www.w3school.com.cn/i/movie.ogg" controls></video>
```

- image组件

```
<image lazy-load="true" src="../../../images/sp2.png"></image>
```

第5节 小程序地图组件map

简介：介绍小程序地图组件map的使用

- map文档<https://developers.weixin.qq.com/miniprogram/dev/component/>

第6节 小程序画布组件canvas

简介：讲解小程序canvas的使用

- canvas文档<https://developers.weixin.qq.com/miniprogram/dev/component/canvas.html>

```
<canvas style="width:600rpx;height:400rpx;" canvas-id="canvas"></canvas>
```

```
onReady: function() {
  console.log("ready");
  var context = wx.createCanvasContext('canvas');

  context.setStrokeStyle("#ff0000");
  context.setLineWidth(2);
```

```
context.moveTo(160,100);
context.arc(100,100,60,0,2*Math.PI,true);
context.moveTo(140,100);
context.arc(100,100,40,0,Math.PI,false);
context.moveTo(85,80);
context.arc(80,80,5,0,2*Math.PI,true);
context.moveTo(125,80);
context.arc(120, 80, 5, 0, 2 * Math.PI, true);
context.stroke();
context.draw();
}
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第五章 常用视图容器及路由跳转

第1节 使用视图容器swiper实现轮播图

简介：详细讲解小程序的滑动组件swiper

```
<swiper class="swiper" autoplay="true" interval="3000" duration="1000"
indicator-dots="true" indicator-active-color="#007aff" vertical="true"
circular="true" display-multiple-items="2">
  <block wx:for="{{imgs}}">
    <swiper-item>
      <image src="{{item}}" class="item_img"></image>
    </swiper-item>
  </block>
</swiper>
```

- swiper文档<https://developers.weixin.qq.com/miniprogram/dev/component/swiper.html>

第2节 深度讲解可滚动视图区域scroll-view

简介：讲解可滚动视图容器scroll-view的使用

- 横向滚动

```
<scroll-view scroll-x="true" style="display:flex;white-space:nowrap;">
  <block wx:for="{{imgs}}">
    <image src="{{item}}"></image>
  </block>
</scroll-view>
```

- 纵向滚动

```
<scroll-view scroll-y="true" scroll-top="100" style="height:340rpx;"
scroll-into-view="{{toView}}" bindscrolltoupper="upper"
bindscrolltolower="lower">
  <block wx:for="{{imgs}}">
    <image src="{{item}}" id="{{'img'+index}}"></image>
  </block>
</scroll-view>
<button bindtap="gotoview">定位到第三张</button>
```

第3节 使用scroll-view实现可滚动导航栏

简介：详细讲解可滚动导航栏的实现

Wxml

```
<scroll-view scroll-x="true" class="nav" scroll-left="{{navScrollLeft}}">
  <block wx:for="{{navData}}" wx:key="index">
    <view class="nav_item {{currentNav == index?'active':''}}" data-current="{{index}}" bindtap="switchNav">{{item.text}}</view>
  </block>
</scroll-view>
```

点击移动scroll-view，添加active类

```
switchNav:function(e){
  // console.log(e);
  var cur = e.currentTarget.dataset.current;
  //tab选项居中
  var singleWidth = this.data.windowWidth/5;
  this.setData({
    navScrollLeft: (cur - 2)*singleWidth
```

```

});
console.log(this.data.navScrollLeft);
if(this.data.currentNav == cur) {
    return false;
}else{
    this.setData({
        currentNav: cur
    })
}
},

```

第4节 特殊覆盖组件cover-view及cover-image

简介：讲解特殊覆盖组件cover-view以及cover-image的使用

```

<video src="https://www.w3school.com.cn/i/movie.ogg" controls>
<cover-view class="shuiyin">
小D课堂
<cover-image src="../../../images/sp2.png"></cover-image>
</cover-view>
</video>

```

第5节 小程序页面导航组件

简介：讲解小程序页面导航组件

```

<navigator url="/pages/personal/personal">跳转到个人页面</navigator>

```

- 导航组件文档：<https://developers.weixin.qq.com/miniprogram/dev/component/navigator.html>

第6节 常用路由跳转方法

简介：讲解常用的路由跳转api

- wx.navigateTo()保留当前页面，跳转到应用内的某个页面。但是不能跳到 tabBar 页面
- wx.switchTap()跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面
- wx.reLaunch()关闭所有页面，打开到应用内的某个页面
- wx.redirectTo()关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabBar 页面
- wx.navigateBack()关闭当前页面，返回上一页面或多级页面
- 路由跳转文档<https://developers.weixin.qq.com/miniprogram/dev/api/route/wx.navigateBack.html>

第7节 小程序页面的传参与取参

简介：介绍小程序页面之间的传参和取参

```
<navigator url="/pages/sort/sort?id=1&name=eric">跳转到分类</navigator>
```

第8节 如何实现小程序的底部导航栏

简介：如何编写小程序的tabbar

```
app.json
"tabBar": {
  "color": "#2b333b",
  "selectedColor": "#3bb149",
  "borderStyle": "black",
  "list": [{
    "pagePath": "pages/index/index",
    "text": "首页"
  }, {
    "pagePath": "pages/sort/sort",
    "text": "分类"
  }, {
    "pagePath": "pages/personal/personal",
    "text": "个人"
  }]
},
```



第六章 构建模板化与模块化小程序

第1节 利用require方法加载js模块文件

简介：如何利用require方法加载js模块

- require路径要写成相对路径

```
var util = require('../utils/util.js');
```

第2节 WXML模板编写与引入精讲

简介：讲解wxml模板编写以及引入方式

- 新建模板文件，wxml和wxss

```
<template name="courseList">
  <view class="item">
    <view class="item_img">
      <image src="{{course.img}}"></image>
    </view>
    <view class="item_name">{{course.name}}</view>
    <view>{{idx}}</view>
  </view>
</template>
```

- 在需要用的页面import模板文件

```
<view class="list">
  <block wx:for="{{list}}" wx:for-item="course" wx:for-index="idx">
    <template is="courseList" data="{{course}}"></template>
  </block>
</view>
```

第3节 讲解wxs模块引用

简介：讲解wxs模块的使用

```
<wxs src="../../wxs/common.wxs" module="info"></wxs>

<view>{{info.message}}</view>
<view>{{info.add(20,10)}}</view>
```

wxs文件

```
//JS代码
var msg = "hello world";
//commonjs模块化导出

var num = function (num1, num2) {
  return num1 + num2;
}
module.exports = {
  message: msg,
  add: num
}
```

第4节 wxs案例练习

简介：讲解wxs常用的一些案例

- 价格保留小数

```
var toFixed = function(value,count){
  return value.toFixed(count);
}
```

- 时间戳格式化

```
var formatNumber = function (n) {
  n = n.toString()
  return n[1] ? n : '0' + n
}

var regYear = getRegExp("(y+)", "i");
```

```

var dateFormat = function (timestamp, format) {
    if (!format) {
        format = "yyyy-MM-dd hh:mm:ss";
    }
    timestamp = parseInt(timestamp);
    var realDate = getDate(timestamp);
    function timeFormat(num) {
        return num < 10 ? '0' + num : num;
    }
    var date = [
        ["M+", timeFormat(realDate.getMonth() + 1)],
        ["d+", timeFormat(realDate.getDate())],
        ["h+", timeFormat(realDate.getHours())],
        ["m+", timeFormat(realDate.getMinutes())],
        ["s+", timeFormat(realDate.getSeconds())],
        ["q+", Math.floor((realDate.getMonth() + 3) / 3)],
        ["S+", realDate.getMilliseconds()],
    ];
    var reg1 = regYear.exec(format);
    // console.log(reg1[0]);
    if (reg1) {
        format = format.replace(reg1[1], (realDate.getFullYear() +
''').substring(4 - reg1[1].length));
    }
    for (var i = 0; i < date.length; i++) {
        var k = date[i][0];
        var v = date[i][1];

        var reg2 = getRegExp("(" + k + ")").exec(format);
        if (reg2) {
            format = format.replace(reg2[1], reg2[1].length == 1
                ? v : ("00" + v).substring(("" + v).length));
        }
    }
    return format;
}

```

持续更新。。。。
