Day1 课件

一.学习目标

1. 了解微信小程序的概念，熟悉微信开发者工具的安装和使用
2. 了解小程序的目录结构及内容
3. 学习阅读/查询微信小程序开发文档
4. 小程序基础知识的学习，编写小程序版的hello world
5. 使用flex布局对小程序进行布局

二. 小程序介绍

1. 小程序是什么?

字面上讲，小程序就是微信里面的应用程序，外部代码通过小程序这种形式，在微信这个手机 App 里面运行。

但是，更准确的说法是， **小程序可以视为只能用微信打开和浏览的网站。** 小程序和网页的技术模型是一样的，用到的 JavaScript 语言和 CSS 样式也是一样的，只是网页的 HTML 标签被稍微修改成了 WXML 标签。所以，小程序页面本质上就是网页。

特点

- 无需安装，用完即走
- 微信小程序的开发成本相对较低，周期也较短。（体验上虽然没法完全媲美原生 APP，但综合考虑却更优于 APP）
- 相较于原生 APP，推广更容易，更简单，也更省成本。

2. 基础知识复习

- JavaScript 语言：懂基本语法，会写简单的 JS 脚本程序。
- CSS 样式：理解如何使用 CSS 控制网页元素的外观。

3. 准备工作

1. 微信开发者工具下载地址
2. 注册小程序账号
3. 小程序文档

4. 微信开发者工具的使用

1. 下载安装

2. 注册微信小程序，申请AppID，也可以使用测试id
3. 认识界面(模拟器,编辑器,调试器)

5. 创建第一个小程序

1. 创建一个微信小程序项目
2. 各个目录的介绍(作用、内容、是否必须)

sitemap.json文件

sitemap定义：Sitemap简称网站地图，就是网站上的网页列表；小程序配置 /sitemap 配置

开发者可以通过 `sitemap.json` 配置其小程序页面是否允许微信索引。

当开发者允许微信索引时，微信会通过爬虫的形式，为小程序的页面内容建立索引。当用户的搜索词条触发该索引时，小程序的页面将可能展示在搜索结果中。

project.config.json文件

小程序项目配置文件,在详情-本地设置中的配置会同步更新到这里

app.wxss 文件

全局的样式文件,在所有页面都可以使用

app.json 文件

`app.json` 采用 JSON 格式。

`app.json` 文件的内容，至少必须有一个 `pages` 属性，指明小程序包含哪些页面。`pages` 属性是一个数组，数组的每一项就是一个页面。这个示例中，小程序只有一个页面，所以数组只有一项 `pages/home/home` 。

app.js 文件

`App()` 由小程序原生提供，它是一个函数，表示新建一个小程序实例。它的参数是一个配置对象，用于设置小程序实例的行为属性。这个例子不需要任何配置，所以使用空对象即可。

utils 工具文件夹

pages 文件夹

`pages/home/home`是一个三层的文件路径。

1. 所有页面都放在`pages`子目录里面。
2. 每个页面有一个自己的目录，这里是`pages`下面的`home`子目录，表示这个页面叫做`home`。页面的名字可以随便起，只要对应的目录确实存在即可。
3. 小程序会加载页面目录`pages/home`里面的`home.js`文件，`.js`后缀名可以省略，所以完整的加载路径为`pages/home/home`。`home.js`这个脚本的文件名也可以随便起，但是习惯上跟页面目录同名。

练习:新建一个项目,创建一个新页面,写一个hello world

6. wxml标签语言

最常用的标签:`和`

- `<view>`标签表示一个区块，用于跟其他区块分隔，类似 HTML 语言的`<div>`标签。
- `<text>`表示一段行内文本，类似于 HTML 语言的`<span>`标签，多个`<text>`标签之间不会产生分行。

注意: 每个标签都是成对使用，需要有闭合标记，即标签名前加斜杠表示闭合，比如`<view>`的闭合标记是`</view>`。如果缺少闭合标记，小程序编译时会报错。

7. 项目配置文件 app.json

- 顶层的`app.json`文件用于整个项目的配置，对于所有页面都有效。
- 除了前面提到的必需的`pages`属性，`app.json`文件还有一个`window`属性，用来设置小程序的窗口。`window`属性的值是一个对象，其中有三个属性很常用。

  > - `navigationBarBackgroundColor`：导航栏的颜色，默认为`#000000`（黑色）。
  > - `navigationBarTextStyle`：导航栏的文字颜色，只支持`black`（黑色）或`white`（白色），默认为`white`。
  > - `navigationBarTitleText`：导航栏的文字，默认为空。

- tabBar属性

如果小程序是一个多 tab 应用（客户端窗口的底部或顶部有 tab 栏可以切换页面），可以通过 tabBar 配置项指定 tab 栏的表现，以及 tab 切换时显示的对应页面。

属性	类型	必填	默认值	描述
color	**HexColor**	**是**		tab 上的文字默认颜色，仅支持十六进制颜色
selectedColor	**HexColor**	**是**		tab 上的文字选中时的颜色，仅支持十六进制颜色
backgroundColor	**HexColor**	**是**		tab 的背景色，仅支持十六进制颜色
borderStyle	string	否	black	tabbar 上边框的颜色， 仅支持 `black` / `white`
list	**Array**	**是**		tab 的列表，详见 `list` 属性说明，最少 2 个、最多 5 个 tab
position	string	否	bottom	tabBar 的位置，仅支持 `bottom` / `top`
custom	boolean	否	false	自定义 tabBar，见详情

其中 list 接受一个数组，**只能配置最少 2 个、最多 5 个 tab**。tab 按数组的顺序排序，每个项都是一个对象，其属性值如下：

属性	类型	必填	说明
pagePath	string	是	页面路径，必须在 pages 中先定义
text	string	是	tab 上按钮文字
iconPath	string	否	图片路径，icon 大小限制为 40kb，建议尺寸为 81px * 81px，不支持网络图片。 **当 `position` 为 `top` 时，不显示 icon。**
selectedIconPath	string	否	选中时的图片路径，icon 大小限制为 40kb，建议尺寸为 81px * 81px，不支持网络图片。 **当 `position` 为 `top` 时，不显示 icon。**

8. 页面样式

以上写了一个最简单的 Hello world 微信小程序。但是，那只是一个裸页面，并不好看。今天接着往下讲，如何为这个页面添加样式，使它看上去更美观，教大家写出实际可以使用的页面。

给hello world添加样式

小程序推荐尺寸单位: rpx与px

- rpx

  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750rpx = 375px = 750物理像素，1rpx = 0.5px = 1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	1rpx = 0.42px	1px = 2.34rpx
iPhone6	1rpx = 0.5px	1px = 2rpx
iPhone6 Plus	1rpx = 0.552px	1px = 1.81rpx

建议： 开发微信小程序时设计师可以用 iPhone6 作为视觉稿的标准。

注意： 在较小的屏幕上不可避免的会有一些毛刺，请在开发时尽量避免这种情况。

flex布局

任务：自行学习flex布局教程后完成3个flex布局效果

9. 微信小程序中的常用组件

1. image

   美观的页面不能光有文字，还必须有图片。小程序的`<image>`组件就用来加载图片。`<image>`组件有很多属性，比如可以通过`style`属性指定样式。当然，图片样式不一定写在`<image>`组件里面，也可以写在 WXSS 样式文件里面。

2. swiper组件

   `<swiper>`组件就是轮播组件，里面放置了三个[`组件](https://developers.weixin.qq.com/miniprogram/dev/component/swiper-item.html)，表示有三个轮播项目，每个项目就是一个`组件。

   `<swiper>`组件的`indicator-dots`属性设置是否显示轮播点，`autoplay`属性设置 是否自动播放轮播。它们的属性值都是一个布尔值，这里要写成`{{true}}`。这种`{{...}}`的语法，表示里面放置的是 JavaScript 代码。

3. scroll-view组件

10.数据绑定

前面的所有示例，小程序的页面都是写死的，也就是页面内容不会变。但是，页面数据其实可以通过脚本传入，通过脚本改变页面，实现动态效果。

小程序提供了一种特别的方法，让页面可以更方便地使用脚本数据，叫做"数据绑定"（data binding）。

所谓"数据绑定"，指的是脚本里面的某些数据，会自动成为页面可以读取的全局变量，两者会同步变动。也就是说，脚本里面修改这个变量的值，页面会随之变化；反过来，页面上修改了这段内容，对应的脚本变量也会随之变化。这也叫做 MVVM 模式。

`name`变量写在`{{...}}`里面。这是小程序特有的语法，两重大括号表示，内部不是文本，而是 JavaScript 代码，它的执行结果会写入页面。因此，`{{name}}`表示读取全局变量`name`的值，将这个值写入网页。

注意，变量名区分大小写，`name`和`Name`是两个不同的变量名。

11.全局数据

数据绑定只对当前页面有效，如果某些数据要在多个页面共享，就需要写到全局配置对象里面。

`App()`方法的参数配置对象有一个`globalData`属性，这个属性就是我们要在多个页面之间分享的值。事实上，配置对象的任何一个属性都可以共享，这里起名为`globalData`只是为了便于识别。

在页面脚本里面获取全局对象的方式如下:

```
const app = getApp();

Page({
data: {
 now: app.globalData.变量名
}
});
```

上面代码中，`getApp()`函数是小程序原生提供的函数方法，用于从页面获取 App 实例对象。拿到实例对象以后，就能从它上面拿到全局配置对象的`globalData`属性，从而把`app.globalData.变量名赋值给页面脚本的`变量名`属性，进而通过数据绑定机制，变成页面的全局变量。

最后，修改一下页面代码。

```
<view>
<text class="title">现在是 {{变量名}}</text>
</view>
```

8. 事件

事件是小程序跟用户互动的主要手段。小程序通过接收各种用户事件，执行回调函数，做出反应。

小程序的常见事件有下面这些。

> - `tap`：触摸后马上离开。
> - `longpress`：触摸后，超过 350ms 再离开。如果指定了该事件的回调函数并触发了该事件，`tap`事件将不被触发。
> - `touchstart`：触摸开始。
> - `touchmove`：触摸后移动。
> - `touchcancel`：触摸动作被打断，如来电提醒，弹窗等。
> - `touchend`：触摸结束。

事件绑定

上面这些事件，在传播上分成两个阶段：先是捕获阶段（由上层元素向下层元素传播），然后是冒泡阶段（由下层元素向上层元素传播）。所以，同一个事件在同一个元素上面其实会触发两次：捕获阶段一次，冒泡阶段一次。详细的介绍，请参考事件模型解释。

小程序允许页面元素，通过属性指定各种事件的回调函数，并且还能够指定是哪个阶段触发回调函数。具体方法是为事件属性名加上不同的前缀。小程序提供四种前缀。

> - `capture-bind`：捕获阶段触发。
> - `capture-catch`：捕获阶段触发，并中断事件，不再向下传播，即中断捕获阶段，并取消随后的冒泡阶段。
> - `bind`：冒泡阶段触发。
> - `catch`：冒泡阶段触发，并取消事件进一步向上冒泡。

下面通过一个例子，来看如何为事件指定回调函数。打开`home.wxml`文件，改成下面的代码。

> ```
> <view>
> <text class="title">hello {{name}}</text>
> <button bind:tap="buttonHandler">点击</button>
> </view>
> ```

上面代码中，我们为页面加上了一个按钮，并为这个按钮指定了触摸事件（`tap`）的回调函数`buttonHandler`，`bind:`前缀表示这个回调函数会在冒泡阶段触发（前缀里面的冒号可以省略，即写成`bindtap`也可以）。

回调函数必须在页面脚本中定义。打开`home.js`文件，改成下面的代码。

```
data: {
name: '张三'
},
buttonHandler(event) {
this.setData({
   name: '李四'
 });
}
});
```

上面代码中，`Page()`方法的参数配置对象里面，定义了`buttonHandler()`，这就是`<button>`元素的回调函数。它有几个地方需要注意。

（1）事件回调函数的参数是事件对象`event`，可以从它上面获取事件信息，比如事件类型、发生时间、发生节点、当前节点等等。

（2）事件回调函数内部的`this`，指向页面实例。

（3）页面实例的`this.setData()`方法，可以更改配置对象的`data`属性，进而通过数据绑定机制，导致页面上的全局变量发生变化。