第三章 UNIAPP与VUE（五）——组件与插件

纲要

一．Uniapp组件分类

二．VUE组件

三．内置组件

四．扩展组件

五．插件

一、Uniapp组件分类

- Uniapp组件分类
  - Vue组件
    - 内置组件
    - 扩展组件
  - 小程序自定义组件

小程序自定义组件支持

- uni-app支持的组件分为vue组件和小程序自定义组件（其规范不是vue规范，而是小程序规范）。
- 日常开发来讲，推荐使用vue组件。uni-app支持小程序组件主要是为了兼容更多生态资源。
- 小程序组件不是vue组件，并且每家小程序都有自己的组件规范，（微信小程序组件：wxml格式）
- 小程序组件不是全端可用。如果需求上只需兼容有限平台，也可以使用小程序组件，否则仍然推荐使用vue组件。
- https://uniapp.dcloud.net.cn/tutorial/miniprogram-subject.html#%E5%B0%8F%E7%A8%8B%E5%BA%8F%E8%87%AA%E5%AE%9A%E4%B9%89%E7%BB%84%E4%BB%B6%E6%94%AF%E6%8C%81

二、Vue组件

```
<component-name property1="value" property2="value">
    content
</component-name>
```

- 组件是视图层的基本组成单元，是一个**单独且可复用的功能模块的封装**。
- 一个组件包括开始标签和结束标签，标签上可以写属性，并对属性赋值。内容写在两个标签之内。
  - 根节点为 <template>，这个 <template> 下在App、H5可以有多个根 <view> 组件，在小程序只能有一个根 <view> 组件。
  - 一个组件的 data 选项必须是一个函数 data()。

属性类型

V指令是特殊的属性

```
<template>
    <view>
        <button size="mini" :disabled="false" hover-start-time=20 >按钮</button>
    </view>
</template> <!-- false是js变量，在组件的属性中使用时，属性名前需增加冒号-->
```

类型	描述	注解
Boolean	布尔值	组件写上该属性，不管该属性等于什么，其值都为 `true` ，只有组件上没有写该属性时，属性值才为 `false` 。如果属性值为变量，变量的值会被转换为 `Boolean` 类型。
Number	数字	1, 2.5
String	字符串	"string"
Array	数组	[1, "string"]
Object	对象	{ key: value }
EventHandler	事件处理函数名	`handlerName` 是 methods 中定义的事件处理函数名
Any	任意属性	

@click="goto('/pages/about/about')"

- 组件允许我们将 UI 划分为独立的、可重用的部分。在实际应用中，组件常常被组织成层层嵌套的树状结构
- **定义组件**：当使用构建步骤时，一般会将 Vue 组件定义在一个单独的 .vue 文件中，即SFC。
- **使用组件**：要使用一个子组件，需要在父组件中导入它。每当使用一个组件，（即使同名）就创建了一个新的**实例。**
- **组件命名**：在单文件组件中，推荐为子组件使用 PascalCase（所有单词的首字母大写,然后直接连接，单词之间没有连接符，如calendarItem ） 的标签名(原生的 HTML 标签名是不区分大小写的），SFC是在编译中区分组件的大小写的。

定义组件名的方式

- 在注册一个组件的时候，需要给它定义一个名字。Uniapp中，定义组件名的方式有两种：
  - kebab-case(短横线分隔命名):必须在引用这个自定义元素时使用 kebab-case，例如 <my-component-name>。
  - 使用 PascalCase (首字母大写命名)
  - 引用这个自定义元素时两种命名法都可以使用，即 <my-component-name> 和 <MyComponentName> 都是可接受的。

深入组件

- https://cn.vuejs.org/guide/components/registration.html

注册组件：全局注册和局部注册

- Vue 组件在使用前需要先被“注册”，这样 Vue 才能在渲染模板时找到其对应的实现。
- 使用单文件组件，可以注册被导入的 .vue 文件。在Uniapp中，全局注册的组件只能在main.js中引入。

  //调用app.component方法全局注册组件

  app.component('my-component', myComponent)

- 全局注册虽然很方便，但有以下几个问题：
  - 全局注册的组件无法在生产打包时被自动移除 (也叫“tree-shaking”)，即使它并没有被实际使用，它仍然会出现在打包后的 JS 文件中。
  - 全局注册在大型项目中使项目的依赖关系变得不明确。父组件使用子组件时，不太容易定位子组件的实现。和使用过多的全局变量一样，可能会影响应用长期的可维护性。
- 局部注册的组件需要在使用它的父组件中显式导入，只能在该父组件中使用，局部注册使组件之间的依赖关系更加明确，并且对 tree-shaking 更加友好。

```
<script>
import ComponentA from './ComponentA.vue'
export default {
  components: {
// 对于每个 components 对象里的属性，它们的 key 名就是注册的组件名，而value就是相应组件的实现。以下是 ComponentA: ComponentA 的缩写
    ComponentA
  }
}
</script>

<template>
    <ComponentA />
</template>
```

props

- 当一个值被传递给 prop 时，它将成为该组件实例上的一个属性。该属性的值可以像其他组件属性一样，在模板和组件的 this 上下文中访问。
- 一个组件可以有任意多的 props，默认情况下，所有 prop 都接受任意类型的值。
- 当一个 prop 被注册后，可以像这样以自定义 attribute 的形式传递数据给它

```
<BlogPost title="Why Vue is so fun" />
```

- Prop**命名**：为了与 HTML属性一致，通常会将其写为 kebab-case 形式
- 所有的 props 都遵循着**单向绑定**原则，props 因父组件的更新而变化，自然地将新的状态向下流往子组件，而不会逆向传递，从而避免了子组件意外修改父组件的状态。每次父组件更新后，所有的子组件中的 props 都会被更新到最新值，这意味着你**不应该**在子组件中去更改一个 prop。若你这么做了，Vue 会在控制台上向你抛出警告。（可以考虑用data局部数据属性或computed计算属性来实现相关想法）。在大多数场景下，子组件应该抛出一个事件来通知父组件做出改变。
- **Prop 校验：**更细致地声明对传入的 props 的校验要求

```
props: ['foo'],
  created() {
    // ✕ 警告！ prop 是只读的!
    this.foo = 'bar
```

监听事件

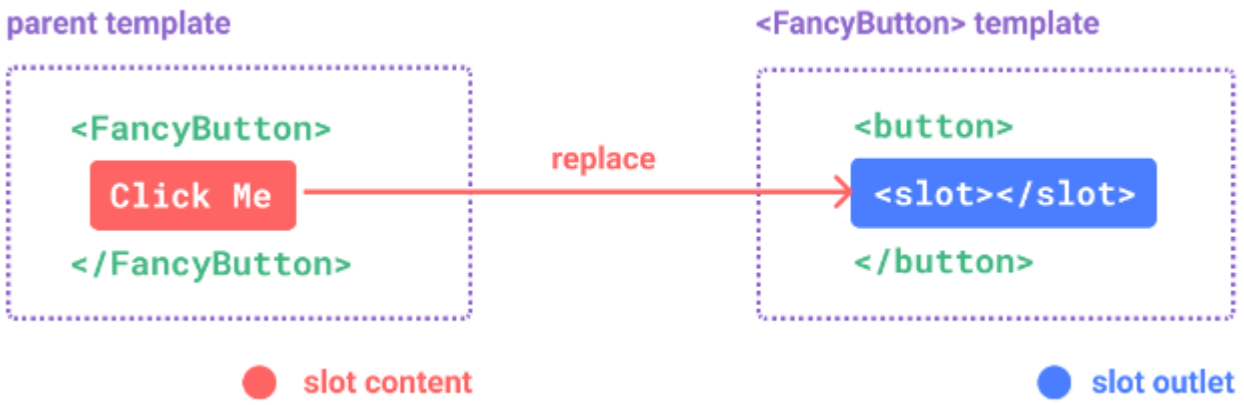
- 子组件需要与父组件交互。子组件可以通过调用内置的 $emit 方法，通过传入事件名称来抛出一个事件 。（emit： 发出 散发光热等）示例
- 事件带参数
- 组件要触发的事件可以显式地通过 emits 选项来声明:

  export default { emits: ['inFocus', 'submit’] }

<slot>插槽、占位符

- 通过插槽来分配内容：使用 <slot> 作为一个占位符，父组件传递进来的内容就会渲染在这里。
- 带 name 的插槽被称为具名插槽 (named slots)。没有提供 name 的 <slot> 出口会隐式地命名为“default”。 <slot name="header"></slot>

```
<div class="alert-box">
  <strong>Error!</strong>
  <br/>
  <slot />
</div>
```



透传属性

- “透传属性”指的是传递给一个组件，却没有被该组件声明为 props 或 emits 的 attribute 或者 v-on 事件监听器；最常见的例子就是 class、style 和 id。

  <button>click me</button> <!-- <子组件MyButton> 的模板 -->

  <MyButton class=“large” /> <!-- <父组件调用了MyButton并加了class -->

  渲染出的 DOM 结果是html：<button class="large">click me</button>

- 如果一个子组件的根元素已经有了 class 或 style attribute，它会和从父组件上继承的值合并，同样的规则也适用于 v-on 事件监听器。如果不想要一个组件自动地继承 attribute，可以在组件选项中设置 inheritAttrs: false。

三、内置组件

- 基础组件 （内置组件）：内置于uni-app框架中，无需再导入项目或注册，随时可以直接使用，比如<view>组件。除了基础组件，都称为扩展组件。
  https://uniapp.dcloud.net.cn/component/view.html
- uni-app的组件与HTML不同，与小程序相同，可更好的满足手机端的使用习惯。编译到非H5平台时编译器会把< div >转换为<view>,转<text><a>转<navigator>等，css里的元素选择器也会转。为了管理方便、策略统一，新写代码时建议直接使用view等组件。
- 小程序组件绑定事件： bindchange="eventName"
- Uniapp（Vue）组件绑定事件： @change="eventName“
- https://uniapp.dcloud.net.cn/component/#%E5%9F%BA%E7%A1%80%E7%BB%84%E4%BB%B6

内置组件

uni-app 组件

- 视图容器
- 基础内容
- 表单组件
- 路由与页面跳转
- 媒体组件
- 地图
- 画布
- webview
- 广告

Navigator页面跳转

基础内容

- icon
- text 包裹文本内容 \n换行
- rich-text 富文本
- progress 进度条

视图容器

- view
- scroll-view 可滚动视图
- swiper 滑块视图
- match-media
- movable-area
- movable-view
- cover-view
- cover-image

表单组件

- button
- checkbox
- editor 富文本编辑器（图片、文字）
- form 表单，uni-ui提供<uni-forms>
- input 单行输入框
- label 标签，使用for属性找到对应的id
- picker 从底部弹起的滚动选择器
- picker-view 嵌入页面的滚动选择器
- radio radio-group
- slider 滑动选择器
- switch 开关选择器
- textarea 多行输入框

四、扩展组件

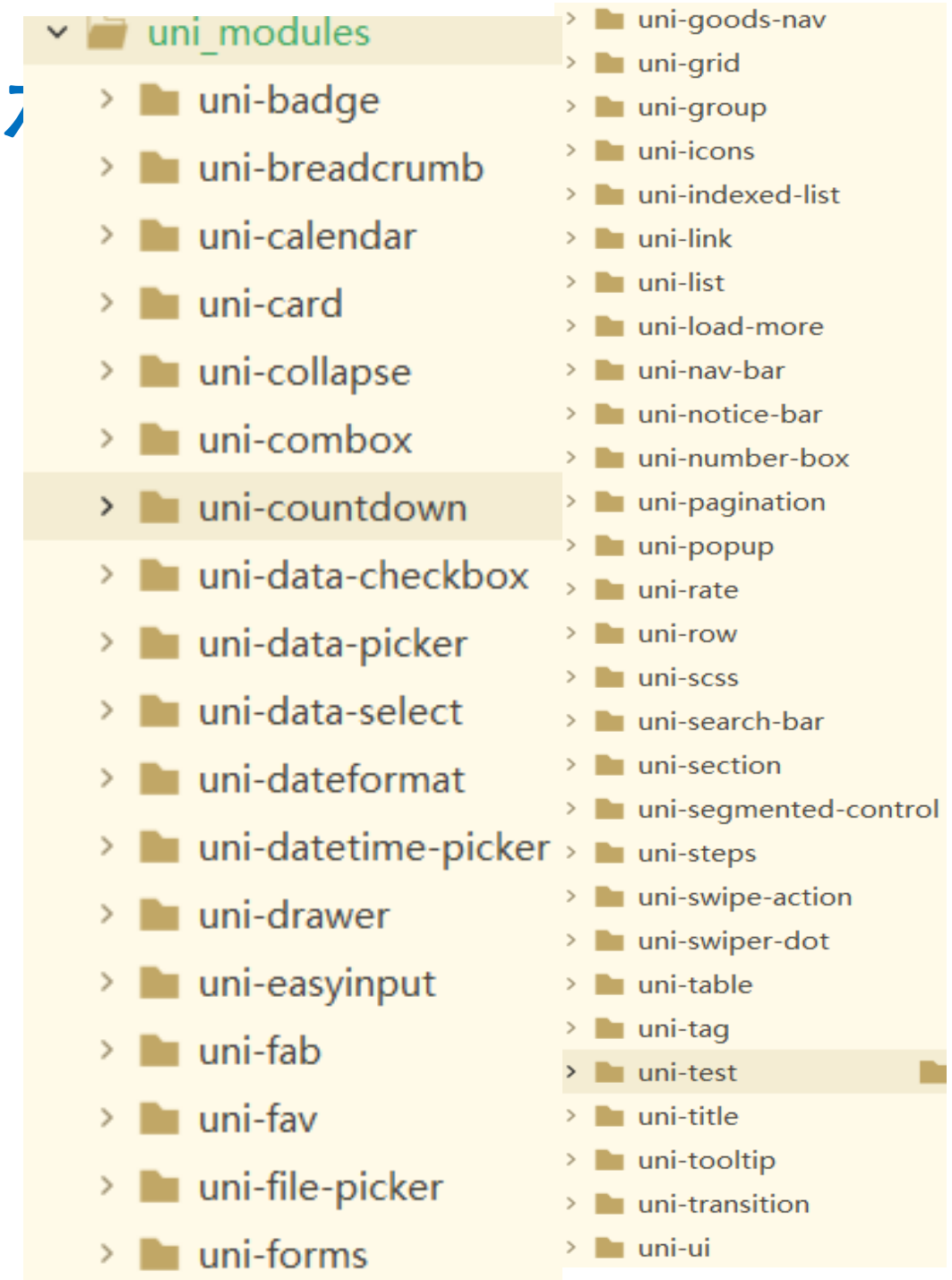
- 尽管业务需求可以通过基础组件满足，但仅有基础组件是低效的，实际开发中会有很多封装的组件。

- 封装扩展组件的优势
  - 可以将组件进行任意次数的复用。
  - 合理的划分组件，有助于提高应用性能。
  - 代码更加方便组织和管理，扩展性更强，便于多人协同开发。
  - 组件化开发能大幅度提高应用开发效率、测试性、复用性等。

uni-ui

- uni-ui是DCloud提供的一个跨端ui库，它是基于vue组件的、flex布局的、无dom的跨全端ui框架，它不包括基础组件，**是基础组件的补充**
- uni-ui支持 HBuilderX直接新建项目模板、npm安装和单独导入个别组件等多种使用方式
- 代码区键入u，拉出各种内置或uni-ui的组件列表，选择其中一个，即可使用该组件。
- uni-scss提供了对应的颜色变量名

uni_modules目录结构



组件名	组件说明
uni-badge	数字角标
uni-calendar	日历
uni-card	卡片
uni-collapse	折叠面板
uni-combox	组合框
uni-countdown	倒计时
uni-data-checkbox	数据选择器
uni-data-picker	数据驱动的picker选择器
uni-dateformat	日期格式化
uni-datetime-picker	日期选择器
uni-drawer	抽屉
uni-easyinput	增强输入框
uni-fab	悬浮按钮
uni-fav	收藏按钮
uni-file-picker	文件选择上传
uni-forms	表单
uni-goods-nav	商品导航
uni-grid	宫格
uni-group	分组
uni-icons	图标
uni-indexed-list	索引列表
uni-link	超链接

uni-link	超链接
uni-list	列表
uni-load-more	加载更多
uni-nav-bar	自定义导航栏
uni-notice-bar	通告栏
uni-number-box	数字输入框
uni-pagination	分页器
uni-popup	弹出层
uni-rate	评分
uni-row	布局-行
uni-search-bar	搜索栏
uni-segmented-control	分段器

uni-steps	步骤条
uni-swipe-action	滑动操作
uni-swiper-dot	轮播图指示点
uni-table	表格
uni-tag	标签
uni-title	章节标题
uni-transition	过渡动画

easycom

- 如果扩展组件符合uni-app的easycom组件规范，可以免注册，直接使用。（如uni-ui扩展组件, Dcloud，新建项目》Uniapp》-Uni-ui项目；HX中敲u就可以列出。）
- easycom将安装、引用、注册，三个步骤精简为一步并直接在页面中使用。只要组件满足以下两个条件：
  - 安装在项目的components目录下或uni_modules目录下，
  - 符合components/组件名称/组件名称.vue目录结构。
- easycom打包后会自动剔除没有使用的组件，对组件库的使用尤为友好。
- 如果组件不符合easycom规范（比如uni_module、datacom、原生组件、uniCloud组件）则需要在代码里手动import和注册后，方可使用。

非easycom规范

- 如果不使用easycom，手动引用和注册vue组件，需以下三步走：
  - import导入组件
  - components里注册组件 （Script>default export>components
  - template中使用组件

五、插件Plugin

- 插件通常用来为 Vue 添加全局功能。插件的功能范围没有严格的限制,一般有下面几种:
  - 添加全局方法或者属性。如: vue-custom-element
  - 添加全局资源：指令/过滤器/过渡等。如 vue-touch
  - 通过全局混入来添加一些组件选项。如vue-router
  - 添加 Vue 实例方法，通过把它们添加到 Vue.prototype 上实现。
  - 一个库，提供自己的 API，同时提供上面提到的一个或多个功能。如vue-router

插件与组件的区别

- 主要表现在以下几个方面：
  - 编写形式
  - 注册形式
  - 使用场景

组件与插件

- Vue组件（component）用来构成你的App的业务模块，它的目标是App.vue。
- Vue插件(plugin) 用来增强你的技术栈的功能模块， 它的目标是Vue本身。（插件是对Vue功能的增强和补充）
- 在某项目中，新建component目录 与 新建 uni-modules目录

模块化：是从代码逻辑的角度进行划分的；方便代码分层开发，保证每个功能模块的职能一致。

组件化：是从UI界面的角度进行规划；前端的组件化，方便UI组件的重用。

Uniapp插件市场

- 插件市场 https://ext.dcloud.net.cn/
- HBuilderX安装目录的plugins目录
- HBuilderX菜单栏“工具”》“插件安装”
  - 已安装插件 安装新插件 前往插件市场安装
- 下载插件并导入HBuilderX
- 用HBuilderX新建IDE插件项目

作业

- 新建项目完成以下三个方面的功能
- 设计页面1，其**至少**包括对**Uniapp内置组件**中以下组件的使用与相关示例
  - 基础内容 icon text rich-text progress
  - 表单组件 form button（或radio） checkbox input label picker-view slider
- 从插件市场下载uni-ui到项目中，设计页面2，选择3个以上uni-ui组件加以应用示例
- 自行创建一个具有一定功能的组件，设计页面3调用该组件，需要包含组件props slot event 等功能，实现父子组件之间的通信。
- 将页面1 2 3以tabBar的方式展示。

*** 作业中，除了提交必要的源代码，还需包含主页和三个页面的运行状态截图

作业截至时间：2023年11月16日