移动应用交互设计课外学习学习笔记

**31801150-张帅-计算机1803**

**学习内容：视频链接**[**https://www.bilibili.com/video/BV1vE411871g**](https://www.bilibili.com/video/BV1vE411871g)

**学习时长：约30小时（含案例制作）**

**约3.15日**

**学习内容：vue的基础知识**

**学习笔记：**

el:绑定

msg：信息（变量）

{{msg}}：输出信息

与多个绑定，则与视为第一个绑定

v-bind:href绑定属性使用，如href

:href = v-bind:href

v-on:=@=绑定事件

计算属性 computed（多个）

侦听 watch（一个变量/数组）

v-if v-else v-else-if

v-show=”” 满足输出

v-for 循环输出

v-cloak:闪动

1.

[v-cloak]{

Display:none

}

2.在标签添加

<div v-cloak>

v-text:

<div v-text=”msg”>(显示msg)

Div中写v-html：xxxx：插入html片段

仅仅推荐本网站内部，有安全隐患

Div中写v-pre 显示原始信息(跳过编译)

Vue：页面中修改，内容也改变

v-once:只编译一次，如果显示的信息后续不需要修改，节约性能

双向：提交的数据影响之前的数据

数据—页面

内容—模型数据

v-model=xxx

两者双向绑定一个变，另一个变

表单—数据—页面

**学习案例：tab.Html**

**约3.22日**

**学习内容：Vue中methods写方法**

**学习笔记：**

名字：function（）{}  
函数中this指的是函数中实例对象

事件直接绑定名称，默认传递事件对象，作为第一个参数

事件直接绑定调用，事件对象必须作为最后一个参数（显示传递），名称是$event

传参用2

事件修饰符

Event.stopPropagation (阻止冒泡)

Vue:事件.stop与上面相同

Event.prevendefault()(阻止事件);

Vue:事件.prevent与上面相同

可以一起写，但是有顺序问题。

Keyup.enter=xxxx(回车处理)

Keyup.delete=xxx（delete键处理）

自定义：vue.config.keycodes.F1=xx;

把f1定义为xx

v-bind:value=”xxx”,绑定值（单向）

v-bind:class=”{类名，是否开启（t,f）}”

v-bind:class=”{类名，是否开启（t,f）；类名，是否开启（t,f）}”

v-bind:class=”{[类名1，类名2] }”

v-bind:style= 绑定样式

v-bind:style=’[A,B]’ B覆盖A

v-show:是否显示

v-if：是否渲染到页面

for in遍历只推荐对象的遍历

v-for=’（a,b,c）in d’

a:值，b：名称，c：下标

学习案例：输入.Html

约4.2日

学习内容：vue与表单互动和自定义指令，计算属性

学习笔记:

v-model绑定text

按钮:js提交，在按钮上绑定事件，prevent来阻止默认的事件

Radio：绑定同一个属性，绑定的属性是谁的value，默认选中的就是谁

Checkbox：同理绑定是数组

下拉框：在select绑定，

Textarea：同text，可以变为初始值

修饰符：

v-model.number:变为数值

Trim:去掉开始和结尾的空格

Lazy:input切换为xchange事件

自定义指定：

Vue.directive(‘指令名字’,{inserted:function(el(指令所绑定的元素)){执行内容}

})

自定义指令，使用时需要 v-指令名字

Vue.directive(‘指令名字’,{inserted:function(el(指令所绑定的元素)，binding){执行内容}

})

Binding:一个对象（形参，可以改成别的名字）

对象包含多个值，binding.value表示对象的数值

使用时：v-指令名字=”传入参数的名字，即binding”

局部指令

Directives:{

指令名字：xxx

}

只能在组件内部，使用范围有限制

计算属性：直接表达式可能比较复杂，为了表达更加简洁

Computed:计算属性名字：function（）{

Return 返回值} 计算方法

使用：{{计算属性值}}

计算属性有缓存的，方法没有缓存！

Watch:侦听器数据发生变化就会

Watch:{

侦听器的值：function(val（侦听的内容的最新值）){

进行的操作

}

}

使用场景：异步或者开销较大的操作

过滤器：vue.filter(‘过滤器名字’,function(value)

{

//逻辑

Return xxx；  
}

)

使用{{元素 | 过滤器名字}}

使用{{元素 | 过滤器名字1 | 过滤器2}}

从左到右运行

属性绑定也可以使用

Filter:{xxx:function}局部过滤器，同自定义指令的方式

过滤器带参数

{{元素 | 过滤器名字1 （'参数1’，‘参数2’）}}…

数组：push（最后） pop shift（最前） unshift 删除/添加

Splice添加数据 sort排序 reverse翻转

（改变原有的数组）

Filter，concat，slice返回新的数组（独立的）

Vue.set(数组，下标，数值)

**学习案例：图书馆.Html**

**约4月8日**

**学习内容：vue组件的注册和使用**

**学习笔记：**

Vue.component(组建名字,{

Data:数据

Template:模板内容

})

Data:值是一个函数，内部返回一个对象

例子：function（）{

Return{

数据名字：数据数值

}

}

Template：‘内容’

注：外部使用单引号，使用内部使用双引号

Methods：定义内部方法

重用组建的话，当中的数据是独立的

组建的模板应该是确切的根元素，即，两个按钮一起是不行的

两个兄弟元素一起的话应该包括在一个根元素里

模板内容可以是模板字符串

组建可以嵌套组建，驼峰式的命名只能在别的组建中使用

普通的标签模板中只能换成a-b的方式使用组建

局部组建：

Var 组建名字={

内容

}

Vue实例中：

{

Components:已经定义的组建名字

}

这样的组件只能在父对象的作用域中使用

Props:父组件中通过绑定值来传递数据

可以是写死的属性，也可以的v-bind动态绑定的数据

Props：[‘接受的数据名字’，‘数据2’，。。。。。]数组

Props：单向数据流（只允许父组件向子组件传递，子组件不能操作prop中的数据）

子组件向父组件传递信息：$emit(“触发的事件”)；

父组件：监听该事件 @触发的事件

逻辑：子组件操作—》父组件执行

子组件传递值：

$emit(“触发的事件”，传递的值)

$event：接受传递的值

**约4月20日**

**学习内容：非父子组建的传值，slot插槽等**

**学习笔记：**

非父子组建的传值：

不能直接交互，要通过事件中心

事件中心定义var 事件中心名字=new Vue（）；

组建内部：在mounted：生命周期中进行监听

创建的事件中心.$on(‘创建的事件名’，函数)

事件触发

创建的事件中心.$emit(‘调用的事件名’，参数)

销毁事件

创建的事件中心.$off(‘创建的事件名’)

Slot在预留的位置插入插入的信息

父组件：<子组件>内容<\子组件>

等于子组件中：<slot>传入的内容<\slot>

Slot中有默认的内容则显示默认，否则显示传递的内容

有名字的插槽

父组件的内容在标签加上slot=’插槽的名字’

匹配<slot name=’插槽名字’>的插槽

没有slot的则匹配没有那么的插槽

<template>标签加上slot属性可以一次性填充多个标签在插条中

<template slot=‘名字’>

标签1

标签2

标签3

</template>

子组件的增加插槽绑定属性

子组件：<slot :名字：“数据”>（把数据传到了父组件进行操作）

父组件：<template slot-scope=’名字’>

{{名字.绑定的数据的名字}}

</template>(定义子组件的数据的怎么样显示的)

Slot-scope：可以获取子组件中绑定的数据

**学习案例：购物车.Html**

**所有学习案例见tab文件夹**