**1.Vue.js的定义:**渐进式的框架.自底向上增量开发.vue只关心图层.

问题:什么叫渐进式?什么叫自底向上?还有什么样的开发?

回答:a.渐进式的意思是:主张最少,即要求最少.框架会根据自己的特点对使用者提出要求,这些要求就是主张.主张的强弱程度会影响在业务开发过程中的方式.

angular主张:必须使用它的模块机制;必须使用它的依赖注入;

react主张:函数式的编程理念.

b.自底向上:先编写每个小代码块 再逐渐增大 层层包裹

自顶向上:先做大框架,每个小代码实现时,大块代码就可以实现.核心本质为分解

**2.框架的作用:**为了帮助我们应对工程的复杂度.因为框架本身也有复杂度,所以要避免杀鸡用牛刀或用太简陋的框架完成复杂工程,一定选择适合的框架去开发

**3.特殊属性-指令:**

**定义**:指令带有前缀v-,代表是vue所特有的属性.

**作用:**当表达式的值改变的时候,响应式地作用于dom;

例如:<p v-if="seen">现在你看到我了</p>中,v-if指令是根据seen的值的真假来插入 或移除<p>元素.

**参数:**某些指令可以接收一个参数.<a v-bind:href="url">...</a> href即为接收到的参数;

**修饰符**:指令后有加修饰符的情况,作用是指出一个指令应该以特殊方式绑定.

**特例**:v-bind缩写为":";v-on缩写为"@";

**4.vue实例**:所有的vue组件都是vue实例.

**5.箭头函数:**

匿名函数,并且简化了函数的定义.两种格式:1.x=>x\*x 只包含表达式 2.包含多条语句 {...}和return无法省略

例如:x => {

if (x > 0) {

return x \* x;

}

else {

return - x \* x;

}

}

箭头函数与传统的JavaScript函数主要区别在于以下几点：

1、对 this 的关联。函数内置 this 的值，取决于箭头函数在哪儿定义，而非箭头函数执行的上下文环境。

2 、new 不可用。箭头函数不能使用 new 关键字来实例化对象，不然会报错。

3、this 不可变。函数内置 this 不可变，在函数体内整个执行环境中为常量

1. 没有arguments对象。更不能通过arguments对象访问传入参数。只能使用显式命名或其他ES6新特性来完成。

**6.插值:**

**插入文本**:数据绑定--“Mustache”语法 (双大括号)msg的属性值为语法包裹内容.属性值改变则插值内容改变

特例:v-once不会随着属性的值的变化而变化.

<span>Message: {{ msg }}</span>

<span v-once>这个将不会改变: {{ msg }}</span>

**插入html:**为了输出真正的HTML,进行转换.

<p>Using mustaches: {{ rawHtml }}</p>

<p>Using v-html directive: <span v-html="rawHtml"></span></p>

数据绑定的内容还可以是js的表达式,但不可以是语句.

**7.JavaScript 表达式:原始表达式和复杂表达式**

原始表达式:this,标识符引用(返回某个变量的值),字面量引用(数据),数组初始化,对象初始化和分组表达式.

复杂表达式:属性访问表达式,函数表达式,对象创建表达式.

**8.条件渲染**

v-if与v-show的区别

v-if:有惰性.如果没有达到它的渲染条件的话 元素不会渲染.如果不经常切换的话 选择if

v-show:成本高.初始状态也会进行渲染.如果要多次切换样式的话,选择show

**9.组件:**

9.1创建组件.(全局和局部)



9.2组件相互包含:

父组件向子组件传递数据时,使用prop;子组件向父组件返回内部发生的事件.

Prop部分:

**注意**:1.字符串模板(js中的内容)需要使用驼峰命名,html中要使用-分隔命名.使用prop时,需要由驼峰切换成分隔线.



1. 动态传递porp的数据,父组件的数据发生改变时,传递的prop数据也发生改变.使用v-bind

<div>  
 <input v-model="parentMsg">  
 <br>  
 <child v-bind:my-message="parentMsg"></child>  
</div>

1. Props在html中传递的是一个字符串 如果需要传递数值的话,使用v-bind

<!-- 传递了一个字符串 "1" -->  
<comp some-prop="1"></comp>

<!-- 传递真正的数值 -->  
<comp v-bind:some-prop="1"></comp>

1. 对prop数据的验证非常有必要.为此,制定prop规则.只能通过对象的方式,不可用字符串数组.

子组件与父组件的通信:

.$on(监听事件)

.$emit(触发事件)

**10.mvvm.**

**mvvm是什么?**

定义:一种开发模式.在框架中可以完成数据和视图的双向交互.只有在数据和用户视图多次进行交换的时候,才会用到.仅限于前端使用.

**mvvm与vue的关系是什么?**

vue运用此种设计模式实现数据和视图的交互.

**mvvm的原理是什么?**

将modle中的数据进行获取,在viewModle中进行逻辑展现,并且绑定view中的相关内容.当视图发生改变时,viewModle创建一个对象,其中的属性描述即为视图的内容,进行逻辑展现.

**mvvm与mvc的区别是什么?**

mvvm是mvc的加强版.mvc中有一个情况无法避免:viewController的代码问题量大 且无法复用.将viewController的代码进行挑选,不可复用的有:接受modle传递给view,接受view传递给modle,连接modle和view.剩下可以复用的逻辑代码由viewModle.

1. **生命周期**

**生命周期是什么?**

**定义:**通俗来说,ViewMoudle即vue实例,从创建到销毁的过程.

**经历什么过程?**

**过程:**创建,初始化数据,编译模板,挂载dom,渲染-更新-渲染,卸载的过程.

**有哪些地方需要注意?**

**注意点:**

1. 含this的函数大多不要使用箭头函数,因为期望this指向的是vue实例.
2. vue实例中包含的内容:el,data,计算属性(computed),methods,watch.
3. computed和method的区别:view中的数据发生变化时才会运行computed,每次进入页面都要执行methods.