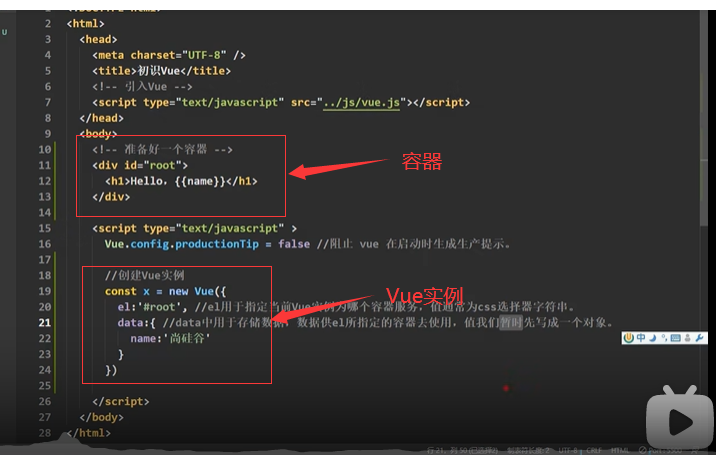
***一、初识Vue***

1. 想让Vue工作就必须创建一个Vue实例，且要传入一个配置对象；
2. 容器里的代码依然符合html规范，只不过混入了一些特殊的Vue语法；
3. 容器里的代码被称为Vue模板。



1. Vue实例和容器一一对应；
2. 真是开发中只有一个Vue实例，并且会配合着组件一起使用；
3. {{xxx}}中的xxx要写js表达式，且xxx可以自动读取到data中所有属性；
4. 一旦data中的数据发生改变，那么页面中用到的该数据的地方也会自动更新；
5. 区分js表达式和js代码（语句）
   1. 表达式：一个表达式会产生一个值，可以放在任何一个需要值得地方；
   2. Js代码(语句)：if(){};for(){}
6. 开发版本得vue.js能在控制台提示一些详细得错误信息，生产版本不会。

***二、模板语法(两大类)***

1、插值语法

a） 功能：用于解析标签体内容；

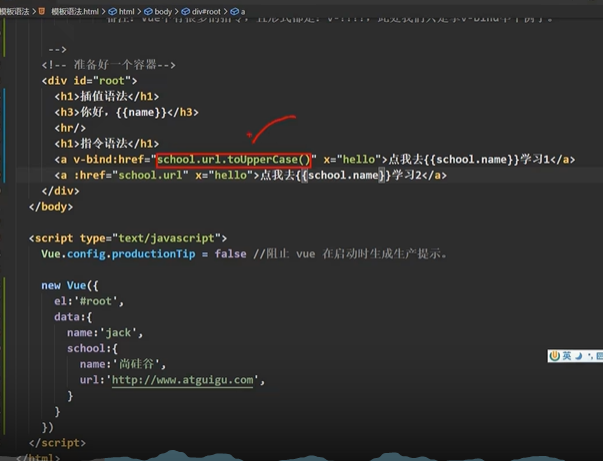
b） 写法：{{xxx}} xxx是js表达式，且可以直接读取到data中得所有属性

2、指令语法

a） 功能：用于解析标签(标签属性、标签体内容、绑定事件……)

b） 举例：v-bind:href=”xxx” 或简写 :href=”xxx” xxx同样要写js表达式

c) 备注：Vue中有很多得指令，且形式都是:v-???, 此处我们只是用v-bind举例



***三、数据绑定(两种方式)***

1、单向数据绑定(v-bind):数据只能从data流向页面

2、双向数据绑定(v-model):数据不仅能从data流向页面，还能从页面流向data。

a） 双向绑定一般都应用在表单类元素上（如：input、select等）

b） v-model:value可以简写为v-model,因为v-model默认收集得就是value值

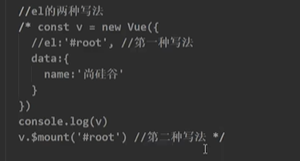


***四、el与data的两种用法***

1、el的两种写法

a） new Vue时候配置el属性

b） 先创建Vue实例，随后再通过vm.$mount(‘#root’)指定el的值

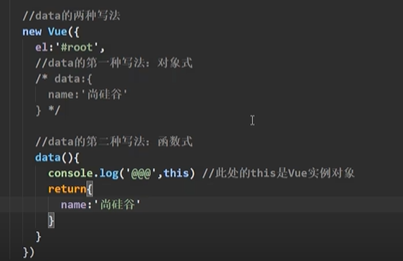


2、data有两种写法

a） 对象式

b） 函数式

如何选择：正常页面用那哪种都可以，组件必须要用函数式



3、一个重要原则

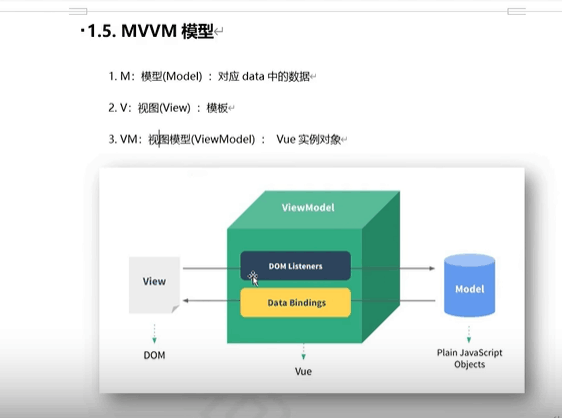
由Vue管理的函数，一定不要写箭头函数，一旦写了箭头函数，this指向的就不是Vue实例，而是window

***五、MVVM***

M：model

V：view

VM：viewmodel



1. data中所有属性，最后都出现在vm身上
2. vm身上所有的属性以及Vue原型上所有的属性，在Vue模板中都可以直接使用。

