上节简单聊到了vue的实例和模板语法 这节简单聊vue的计算属性和侦听器等

计算属性 computed

我们可以在Mustache标签被写入模板表达式，这样虽然在某种情况下是便利的，但是设计他们的初衷是用于简单运算的。

一旦在模板中放入太多的逻辑会让模板过于冗杂且难于维护，所以针对于复杂逻辑vue提出了computed

computed也是vue实例选项中的一个属性

<div id='#app'>

<p>{{index}}</p>

<span>{{comindex}}</span>

</div>

var vm = new Vue({

el:"#app",

data:{

index:"hello vue"

},

computed:{

comindex:function（）{

return "123"+this.index；

}

}

})

上面就是一个简单的vue 计算属性的简单例子，这样看起来代码还是易懂的；

同样的计算属性可以实实现的我们为什么不用自己定义的方法呢？是的 我们还有methods属性

两种方式最后实现的效果是完全一样的，但是

计算属性是基于他们的依赖进行缓存的。只有相关依赖发生改变时才会重新求值，这就意味着只要依赖不改变，计算属性就会直接返回原先的计算结果而不必重新执行函数

而methods则是只要出发重新渲染，调用方法将总会重新渲染函数

二者之下 各有千秋，如果并不需要每次更新就使用计算属性，如果需要每次更新就使用methods

侦听属性 watch

vue提供了一种更通用的方式来观察和响应vue实例上的数据变动：侦听属性

vue通过watch选项提供了一个更通用的方法来响应数据的变化，当需要在数据变化时执行异步或开销较大的操作时，我们应该使用watch

关于Class和Style绑定

操作元素的class列表和内联样式是数据绑定的一个常见需求。因为他们都是属性，所以我们通过v-bind处理他们，只需要通过表达式计算出字符串即可。

但是字符串拼接容易出错，vue在v-bind用于class和style时作了增强，表达式结果除了字符串还可以是对象或数组

绑定class

对象

传给标签一个对象

<a v-bind:class='{wh:whhy}'></a>

data:{

wh:true <a class='whhy'></a>

}

从js角度，对象是一个键值对，所以在此 key：value 此时 wh是一个类名 whhy是一个boolean值

key作为一个变量载体 value是一个变量 只不过 value是在表达式或data中定义的变量

此时 wh是一个类名 whhy是一个boolean值 为true或false 如果为真 wh存在 反之不存在

也可以直接在class后面接一个对象名称 然后在data中定义出来这个对象 为true 或为false与上述效果一样

数组

当然我们也可以在class后传入一个数组

形如<a v-bind:class='[wh]'></a>

然后在data中初始化数据

data{

wh:whhy <a class='whhy'></a>

}

绑定内联样式

对象

v-bind:style的对象语法非常直观，非常像css 但是其实是一个js对象，需要注意的一点那是css属性名用驼峰命名法或者短横线分割

<p v-bind:style='{color:comcolor,backgroundColor:bgc}'></p>

data:{

comcolor:"#444",

bgc:"#888"

}

或者说直接绑定到一个对象上 极为推荐

<p v-bind:style='comstyle'></p>

data:{

comstyle:{

color:'red',

fontSize:'11px'

}

}

需要注意的是样式的属性值必须用引号包裹起来

数组

v-bind：style='[]',可以通过数组把多个样式对象添加到DOM上