

第八课 侦听器 watch 选项

学习目录

- watch 使用

一. watch 使用

之前课程中提到过如果要对 data 中的数据进行较为复杂的计算的时候可以使用 computed 计算属性，不过计算属性只适合于相对简单的计算结果，如果计算过程比较复杂那还可以使用 watch 侦听器来实现业务逻辑。

监听值类型数据变化

```
data:{  
  
  message:'星星课堂',  
  
},  
  
watch: {  
  
  message: function (newVal, oldVal) {  
  
    console.log('数据更新的新值' + newVal);  
  
    console.log('数据更新上一个值' + oldVal);  
  
  },  
  
}
```

监听对象或数组类型数据变化

```
data:{  
  
  obj:{  
  
    num:100  
  
  }  
  
},  
  
watch:{  
  
  arr:{  
  
    handler(newVal, oldVal){  
  
      console.log(newVal);  
  
      console.log(oldVal);  
  
    },  
  
    deep:true //进行深度监听  
  
  }  
  
}
```

Watch 侦听器在默认情况下一开始不会执行监听函数，只有在监听的值发生变化的时候才会调用监听函数，假如一开始就要执行监听函数，那么可以设置 `immediate` 属性为 `true` 来做到。

```
data: {  
  
  message: 'xingxingclassroom'  
  
},  
  
watch: {
```

```
message: {  
  
  handler(newVal,oldVal) {  
  
    console.log(newVal);  
  
    console.log(oldVal);  
  
  },  
  
  immediate: true  
  
}  
  
}
```

在 watch 中还支持异步操作的数据变化监测，在监测过程中还可以设置中间状态，等到异步操作完成在进行数据响应更新。

在 watch 选项中的监听函数尽量不要使用箭头函数，因为箭头函数绑定了父级的上下文对象，也就是 this 是指向的父级，因此假如在 watch 选项的监听函数中调用了 methods 的方法，那么 this 会找不到对应的方法。

谢谢观看！

我是星星课堂老师：周小周