

第十六课 自定义指令

学习目录

- 自定义指令介绍与应用

一. 自定义指令介绍与应用

在写 vue 项目的时候，我们经常需要对后台返回的数据进行大量的渲染操作，其中就包含了大量的对特殊数据的进一步处理，比如说时间戳、图片地址、特殊数据显示等等特殊数据处理改进。

其实遇到这种情况，通过之前掌握的计算属性可以解决我们的问题，不过我们要说的是自定义指令，前面两种方式有他们好处，自定义指令也有自己独特的优点，比如说自定义指令自身就包含一系列生命周期钩子函数，能够在不同的生命周期函数中传递参数，添加修饰符等操作，因此自定义指令也能够处理更复杂的数据。

首先我们要先了解一下自定义指令的基本结构，这里首先是全局定义一个自定义指令，这个自定义指令的名称是 demo。

```
Vue.directive('demo', { //自定义指令钩子函数  
  
  bind: function() {}, //被绑定钩子函数  
  
  inserted: function() {},//绑定到节点钩子函数  
  
  update: function() { }, //组件更新钩子函数  
  
  componentUpdated: function() {}, //组件更新完成钩子函数  
  
  unbind: function() {}//解绑钩子函数  
  
  })
```

每个自定义指令的生命周期都可以接受一下以下参数，即 `el`、`binding`、`vnode` 和 `oldVnode`，其中主要的参数是 `el` 和 `binding` 两个参数。

el：指令所绑定的元素，可以用来直接操作 DOM 。

binding：一个对象，其中主要属性为：

name：指令名，不包括 `v-` 前缀。

value：指令的绑定值，例如：`v-my-directive="1 + 1"` 中，绑定值为 `2`。

arg：传给指令的参数，可选。例如 `v-my-directive:foo` 中，参数为 `"foo"`。

modifiers：一个包含修饰符的对象。例如：`v-my-directive.foo.bar` 中，修饰符对象为 `{ foo: true, bar: true }`。

在了解了自定义指令的生命周期函数和函数参数之后，下面我们就来自己写一个自定义指令的例子来进一步了解自定义指令的用法，我们来写一个格式化时间戳的自定义指令 `v-timeformat` 来练习一下自定义指令知识。

```
Vue.directive('timeformat', {
```

```
  bind: function (el, binding, vnode) { //这里只在 bind 生命周期函数中进行定
```

义和操作，大家也可以在其他自定义指令的生命周期函数中尝试进行扩展和修改。

```
    function initTime(num){ //判断数字是否大于 10
```

```
      return num < 10 ? '0'+num : num;
```

```
    }
```

```
    function checkTime(timestamp){ //格式化时间戳函数，大家也可以根据
```

实际需求进行扩展和改写

```
var time = new Date(timestamp);

var year = time.getFullYear();

var month = time.getMonth()+1;

var day = time.getDate();

var hour = time.getHours();

var minute = time.getMinutes();

var second = time.getSeconds();

if(binding.modifiers.small){//判断修饰符是否存在，存在即返回指定
```

时间格式

```
return year+'-'+initTime(month)+'-'+initTime(day);

}else if(binding.modifiers.big){

return      year+'-'+initTime(month)+'-'+initTime(day)+'

'+initTime(hour)+':' +initTime(minute)+':' +initTime(second);

}else{

return      year+'-'+initTime(month)+'-'+initTime(day)+'

'+initTime(hour)+':' +initTime(minute)+':' +initTime(second);

}

}

el.innerHTML = checkTime(binding.value);//格式化之后的时间信息插
```

入到当前元素中

```
binding.arg == '' ? el.style.color = '#333' : el.style.color =
```



学习前端，最快的进步是持续！

binding.arg//通过传递参数来改变元素字体颜色

```
    }  
  })  
  
  <div v-timeformat:orange.small="timeStamp"> </div>
```

在使用自定义指令时要在我们定义的自定义指令名称前加 v-。

自定义指令和 vue 系统自带指令类似，可以通过传递参数来改变元素状态，这里传入 orange 表示把这个元素的字体颜色改成橙黄色。

这里紧跟在生命周期函数参数之后的是修饰符，传递不同的修饰符来选择要格式化的不同的时间戳的样式和格式。

谢谢观看！如果觉得课程还不错的话，记得给个好评！

我是星星课堂老师：周小周