<template>

<div id="app" class="container">

<table class="table">

<thead>

<tr>

<th>产品编号</th>

<th>产品名字</th>

<th>购买数量</th>

<th>产品单价</th>

<th>产品总价</th>

<th>操作</th>

</tr>

</thead>

<tbody>

<tr v-for="(item,index) in message" :key="index">

<td @click="jia(index)">{{item.id}}</td>

<td>{{item.name}}</td>

<td>

<button type="button" class="btn tn-primary" @click="subtract(index)">-</button>

<input type="text" v-model="item.quantity">

<button type="button" class="btn tn-primary" @click="add(index)">+</button>

</td>

<td>{{item.price | filtermoney}}</td>

<!--<td>{{arr[index].one}}</td>-->

<td>{{item.price\*item.quantity | filtermoney}}</td>

<td>

<button type="button" class="btn btn-danger" @click="remove(index)">移除</button>

</td>

</tr>

<tr>

<td>总购买价

</td>

<td>

{{animatenum | filtermoney}}

</td>

<td>总购买数量

</td>

<td>

</td>

<td colspan="2">

<button type="button" class="btn btn-danger" @click="empty()">清空购物车</button>

</td>

</tr>

</tbody>

</table>

<p v-if="message.length===0">您的购物车为空</p>

</div>

</template>

<script>

export default{

data(){

return{

totalPrice:0,

animatenum:0,

message:[

{

id: 7,

name: 'iphone5s',

quantity: 3,

price: 4000

},{

id: 1340,

name: 'iphone5',

quantity: 9,

price: 3000

},{

id: 7758,

name: 'imac',

quantity: 4,

price: 7000

},{

id: 2017,

name: 'ipad',

quantity: 5,

price: 6000

}

]

}

},

watch:{

toComput2(newValue,oldValue){

this.tween(newValue,oldValue);

}

},

computed:{

//计算总金额

toComput2(){

var vm=this;

//每次进来要重置总金额

vm.totalPrice=0;

this.message.forEach(function(mess){

vm.totalPrice+=parseInt(mess.price\*mess.quantity);

})

return this.totalPrice;

}

},

filters:{

filtermoney(value){

return '￥'+value ;

}

},

mounted(){

this.tween('97000','0');

},

methods:{

//计算总数的方法为什么写在methods里面就不行？

toComput(){

var vm=this;

vm.message.forEach(function(mess){

vm.totalPrice+=parseInt(mess.price\*mess.quantity);

})

return vm.totalPrice;

},

add(index){

var vm=this;

vm.message[index].quantity++;

},

subtract(index){

var vm=this;

vm.message[index].quantity--;

if(vm.message[index].quantity<=0){

if (confirm("你确定移除该商品？")) {

vm.message.splice(index,1)

}

}

},

remove(index){

var vm=this;

if (confirm("你确定移除该商品？")) {

vm.message.splice(index,1)

}

},

empty(){

var vm=this;

vm.message.splice(0,vm.message.length);

},

jia(index){

var vm=this;

vm.arr[index].one++;

},

tween(newValue,oldValue){

var vm=this;

var twen=new TWEEN.Tween({animatenum:oldValue});

function animate() {

requestAnimationFrame(animate);

TWEEN.update();

};

twen.to({animatenum:newValue},750);

twen.onUpdate(function(){

//toFixed();保留几位小数

vm.animatenum = this.animatenum.toFixed();

})

twen.start();

animate();

}

}

}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->

<style scoped>

</style>

<template>

<div id="app" class="container">

<table class="table">

<tr>

<th>产品编号</th>

<th>产品名字</th>

<th>购买数量</th>

<th>产品单价</th>

<th>操作</th>

</tr>

<tr v-for="(item,index) in message" :key="index">

<td @click="jia(index)">{{item.id}}</td>

<td>{{item.name}}</td>

<td>

<button type="button" class="btn tn-primaty" @click="subtract(index)">-</button>

<input type="text" v-model="item.quantity">

<button type="button" class="btn tn-primary" @click="add(index)">+</button>

</td>

<td>{{item.price|filtermoney}}</td>

<td>{{item.price\*item.quantity|filtermoney}}</td>

<td>

<button type="button" class="btn btn-danger" @click="remove(index)">移除</button>

</td>

</tr>

<tr>

<td>购买总价

</td>

<td>

{{animatenum|filtermoney}}

</td>

<td>总购买数量

</td>

<td>

</td>

<td colspan="2">

<button type="button" class="btn btn-danger" @click="empty()">清空购物车</button>

</td>

</tr>

</table>

<p v-if="message.length===0">您的购物车为空</p>

</div>

</template>

<script>

export default{

data(){

return{

totolPrice:0,

animatenum:0,

mesage:[

{

id:7,

name:'iphone5s',

quantity:3,

price:4000

},

{

id:1340,

name:'iphone5',

quantity:9,

price:3000

},

{

id:7758,

name:'imac',

quantity:4,

price:7000

},

{

id:2017,

name:'ipad',

quantity:5,

price:6000

}

]

}

},

watch:{

toComput2(newValue,oldValue){

this.tween(newValue,oldValue);

}

},

computed:{

//计算总金额

toComput2(){

var vm=this;

//每次进来要重置总金额

vm.totalPrice=0;

this.message.forEach(function(mess){

vm.totalPrice+=parseInt(mess.price\*mess.quantity);

})

return this.totalPrice;

}

},

filters:{

filtermoney(value){

return '￥'+value;

}

},

mounted(){

this.tween('97000','0');

},

methods:{

toComput(){

var vm=this;

this.message.forEach(function(mess){

vm.totalPrice+=parseInt(mess.price\*mess.quantity);

})

return vm.totalPrice;

},

add(index){

var vm=this;

vm.message[index],quantity++;

},

subtract(index){

var vm=this;

this.message[index].quantity--;

if(vm.message[index].quantity<=0){

if(confirm("你确定移除该商品？")){

vm.message.splice(index,1)

}

}

},

remove(index){

var vm=this;

if(confirm("你确定移除该商品？")){

vm.mesasge.splice(index,1)

}

},

empty(){

var vm=this;

vm.message.splice(0,vm.message.length);

},

jia(index){

var vm=this;

vm.arr[index].one++;

},

tween(newValue,oldValue){

var vm=this;

var twen=new TWEEN.Tween({animatenum:oldValue});

function animate(){

requestAnimationFrame(animate);

TWEEN.update();

};

twen.to({animatenum:newValue},750);

twen.onUpdate(function(){

//toFixed();保留几位小数

vm.animatenum = this.animatenum.toFixed();

})

twen.start();

animate();

}

}

}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->

<style scoped>

[v-cloak]{

display: none;

}

table{

border: 1px solid black;

border-collapse: collapse;

border-spacing: 0;

empty-cells: show;

}

th,td{

padding: 8px 16px;

border:1px solid black;

text-align: center;

}

th{

background-color: gray;

}

</style>