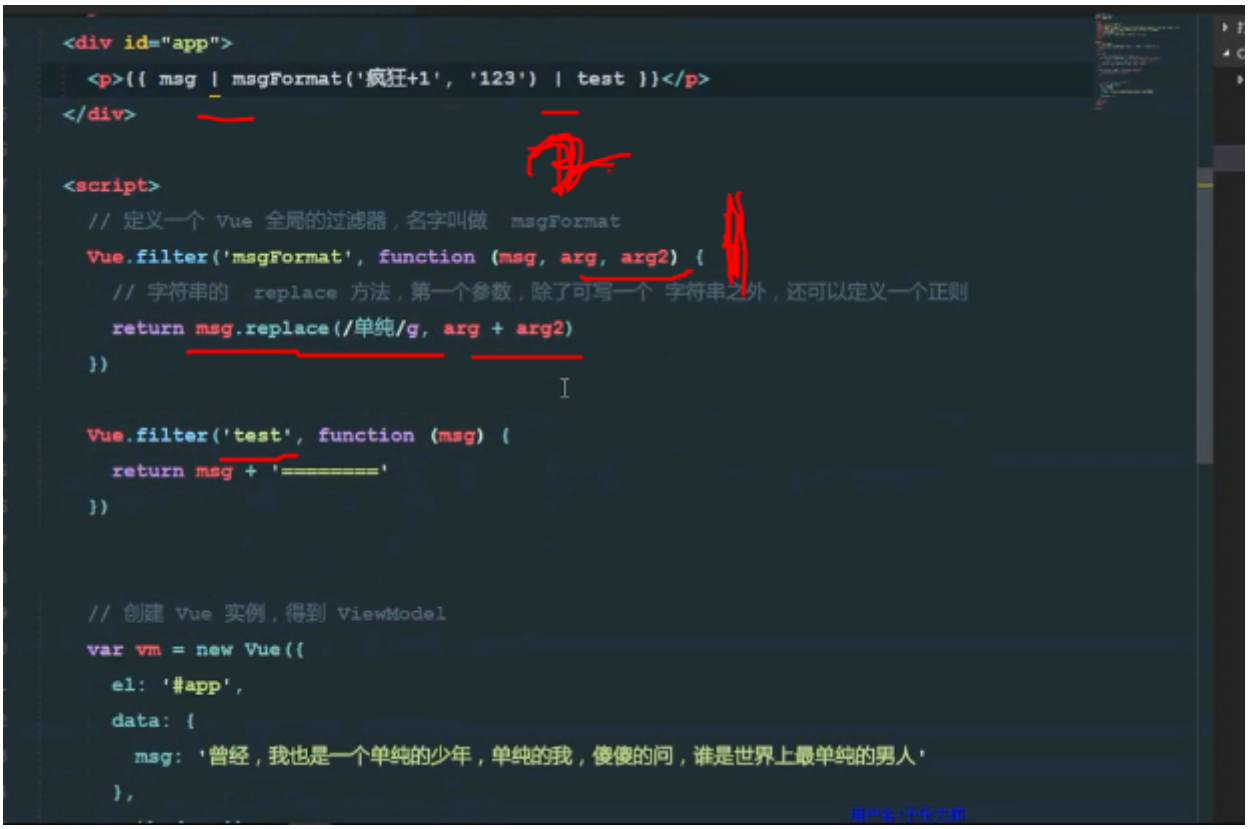
1:定义一个全局的过滤器



过滤器可以定义多个，

```
<p>{{ msg | msgFormat(msg) | test(msg)}}</p>
```

过滤顺序从前到后，

1：过滤器中也可可以穿多个参数

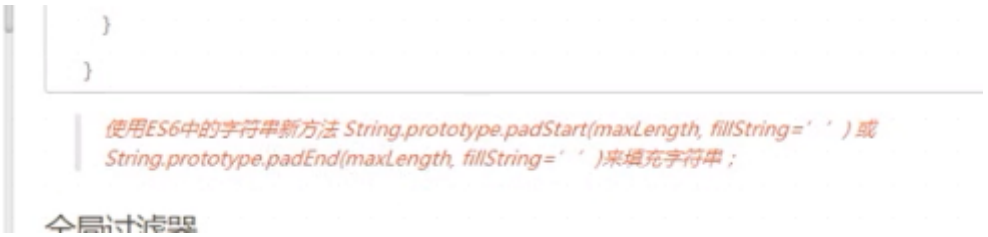
2:也可以使用多个过滤器

二：定义一个私有的过滤器

```
// 如何自定义一个私有的过滤器（局部）
var vm2 = new Vue({
  el: '#app2',
  data: {
    dt: new Date()
  },
  methods: {},
  filters: { // 定义私有过滤器    过滤器有两个 条件   【过滤器名称 和 处理函数】
    // 过滤器调用的时候，采用的是就近原则，如果私有过滤器和全局过滤器名称一致了，这时候 优先调用私
    有过滤器
    dateFormat: function (dateStr, pattern = '') {
      // 根据给定的时间字符串，得到特定的时间
      var dt = new Date(dateStr)

      //   yyyy-mm-dd
      var y = dt.getFullYear()
      var m = dt.getMonth() + 1
      var d = dt.getDate()
```

三：字符串的填充方法：



```
    }
  }
```

使用ES6中的字符串新方法 String.prototype.padStart(maxLength, fillString=' ')或 String.prototype.padEnd(maxLength, fillString=' ')来填充字符串；

String.padStart("填充完的最大长度"，“用什么填充”)；在字符串前面填充

String.padEnd()

四：自定义按键修饰符：

1. 通过 Vue.config.keyCodes.名称 = 按键值 来自定义案件修饰符的别名：

@keyup.f2="方法"；

Vue.config.keyCodes.f2 = 113;

五：自定义一个私有的或者公有 的指令

阶段： bind

inserted

updated

```
Vue.config.keyCodes.f2 = 113

// 使用 Vue.directive() 定义全局的指令 v-focus
// 其中：参数1 ： 指令的名称，注意，在定义的时候，指令的名称前面，不需要加 v- 前缀，
// 但是： 在调用的时候，必须 在指令名称前 加上 v- 前缀来进行调用
//  参数2： 是一个对象，这个对象身上，有一些指令相关的函数，这些函数可以在特定的阶段，执行相关的操作
Vue.directive('focus', {
  bind: function (el) { // 每当指令绑定到元素上的时候，会立即执行这个 bind 函数，只执行一次
    // 注意： 在每个 函数中，第一个参数，永远是 el ，表示 被绑定了指令的那个元素，这个 el 参数，是一个原生的JS对象
    // 在元素 刚绑定了指令的时候，还没有 插入到 DOM中去，这时候，调用 focus 方法没有作用
    //  因为，一个元素，只有插入DOM之后，才能获取焦点
    // el.focus()
  },
  inserted: function (el) {  // inserted 表示元素 插入到DOM中的时候，会执行 inserted 函数【触发1次】
    el.focus()
  },
  updated: function (el) {  // 当VNode更新的时候，会执行 updated， 可能会触发多次

  }
})
```

定义一个私有的过滤器（

el:表示要绑定的元素

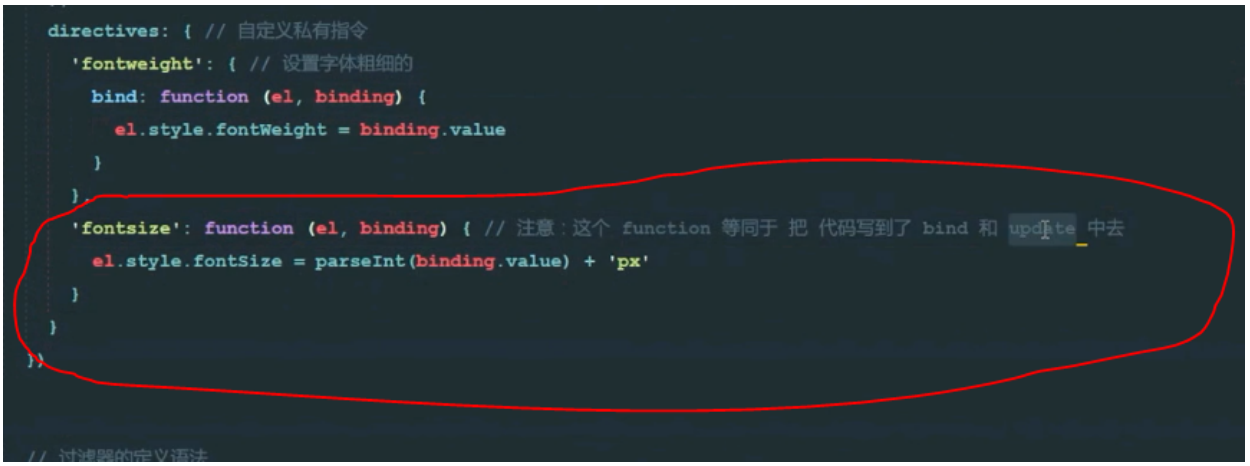
bingding:是一个形参

也可以写成其他的

）

```
// 如何自定义一个私有的过滤器（局部）
var vm2 = new Vue({
  el: '#app2',
  data: {
    dt: new Date()
  },
  methods: {},
  filters: { // 定义私有过滤器  过滤器有两个 条件  【过滤器名称 和 处理函数】 ...
  },
  directives: { // 自定义私有指令
    'fontweight': {
      bind: function (el, binding) {
        el.style.fontWeight = binding.value
      }
    }
  }
```

定义指令的简写形式



// 如何自定义一个私有的过滤器（局部）	
	var vm2 = new Vue({
	el: '#app2',
	data: {
	dt: new Date()
	},
	methods: {},
	filters: { // 定义私有过滤器 过滤器有两个 条件 【过滤器名称 和 处理函数】
	// 过滤器调用的时候，采用的是就近原则，如果私有过滤器和全局过滤器名称一致了，这时候 优先调用私有过滤器
	dateFormat: function (dateStr, pattern = '') {
	// 根据给定的时间字符串，得到特定的时间
	var dt = new Date(dateStr)
	// yyyy-mm-dd
	var y = dt.getFullYear()
	var m = (dt.getMonth() + 1).toString().padStart(2, '0')
	var d = dt.getDate().toString().padStart(2, '0')

```
if (pattern.toLowerCase() === 'yyyy-mm-dd') {
return `${y}-${m}-${d}`
} else {
var hh = dt.getHours().toString().padStart(2, '0')
var mm = dt.getMinutes().toString().padStart(2, '0')
var ss = dt.getSeconds().toString().padStart(2, '0')

return `${y}-${m}-${d} ${hh}:${mm}:${ss} ~~~~~~~~`
}
}
},
directives: { // 自定义私有指令
'fontweight': { // 设置字体粗细的
bind: function (el, binding) {
el.style.fontWeight = binding.value
}
},
'fontsize': function (el, binding) { // 注意：这个 function 等同于 把 代码写到了 bind 和 update 中去
el.style.fontSize = parseInt(binding.value) + 'px'
}
}
})
```

	// 过滤器的定义语法
	// Vue.filter('过滤器的名称', function(){})
	// 过滤器中的 function ， 第一个参数，已经被规定死了，永远都是 过滤器 管道符前面 传递过来的数据
	/* Vue.filter('过滤器的名称', function (data) {
	return data + '123'
	}) */
	// document.getElementById('search').focus()