

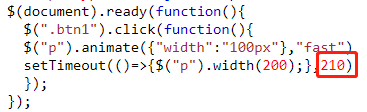
1567146414(1)

Css里面不能用@代替src路径

background-attachment:fixed;

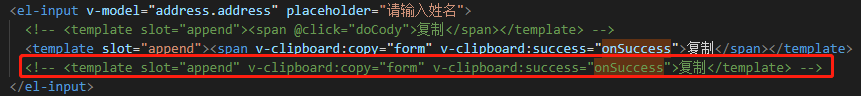
background-position:center;

您需要把 background-attachment 属性设置为 "fixed"，才能保证position在 Firefox 和 Opera 中正常工作。



动画是有一定时间的，所以在动画执行时间内去设置width是没有效果的，通过验证似乎animate是异步的，因为在setTimeout设置时间为0的时候，setTimeout也是在动画执行的时间范围内

Vue-clipboard的坑（start）：



红圈中的代码是无效的，不要在template slot中直接使用v-clipboard，也可能是刚才用cnpm下载vue-clipboard2的原因，刚才换成官网的npm install --save vue-clipboard2就可以了

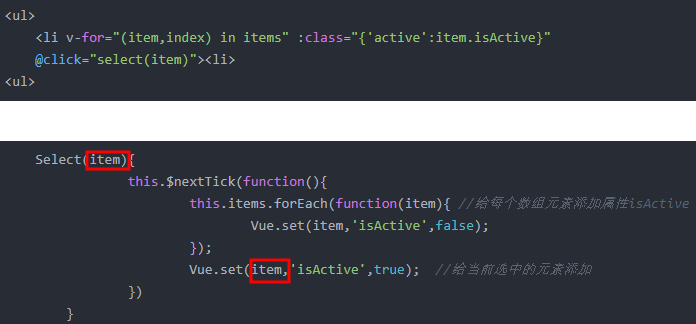
1567587002(1)

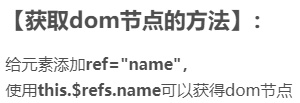
这是elementui组件库的input

1567587223(1)

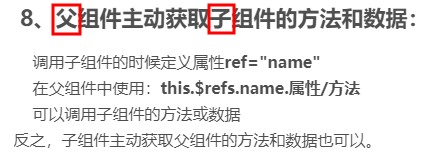
也不能直接复制对象的属性（带有点）

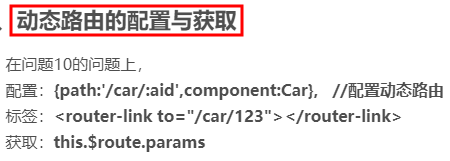
Vue-clipboard的坑（end）

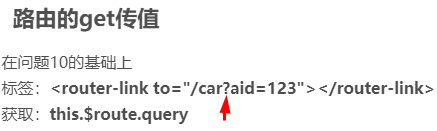




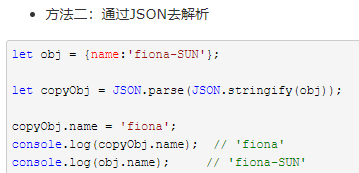
也可以通过自定义指令，其中第一个参数el就是该节点

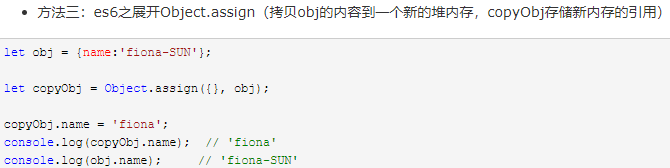


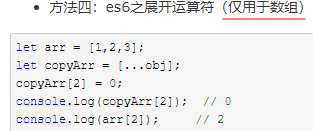


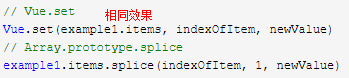


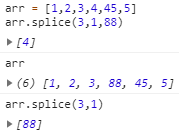






不用concat





不管是删除还是替换，都返回被去除的元素

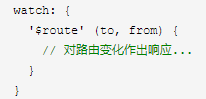
Splice也可以用于截断数组





这是获取视图层的innerHTML

NextTick的作用就是在视图更新之后再执行



Vue.prototype.$vue = Vue

this.$vue.set(this.list,8,{'name':'linxiaosheng'})

Vue.set(list,index,newItem)

index是list的索引，如果索引存在的话是替换，如果不存在是添加

## class 属性绑定

<div v-bind:class="{ active: isActive }"></div>

绑定一个对象



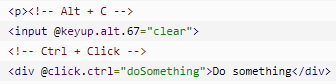






<input v-on:keyup.13="submit">

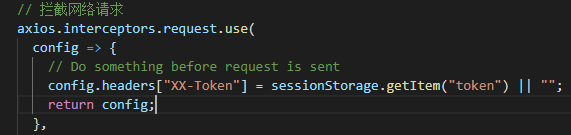




<input v-model.lazy="msg" @change="show">

<span>{{this.msg}}</span>

在失去焦点之后span中的内容才被双向绑定，才执行chang中的方法





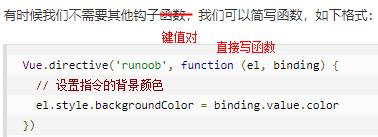
<input v-model.number="age" type="number">

只能输入数字，包括e字母，这是天文数字的e

<input v-model.trim="msg">

自动过滤用户输入的首尾空格







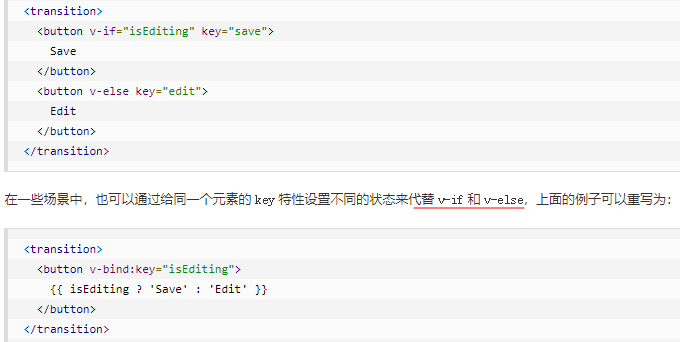




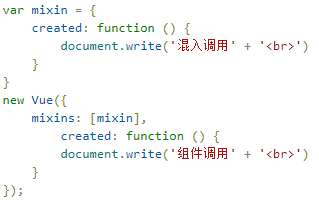
也可以以对象的形式添加binding属性





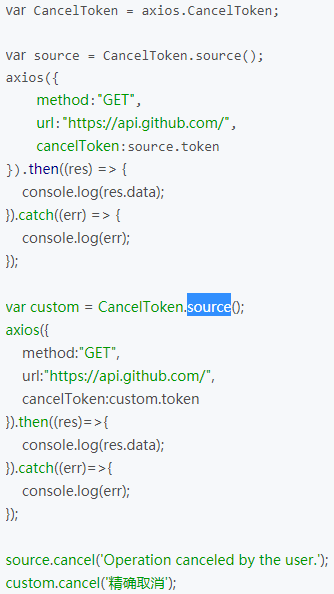












background-repeat:repeat-x;

background-position:left center;



background-attachment:fixed;

text-transform:lowercase;

text-transform:capitalize;

text-indent:50px;第一行的缩进

div.ex1 {direction:rtl;}

<div class="ex1">。你是谁？我是man</div>

文字的居右

letter-spacing:-3px;文字间的距离

text-align:justify两端对齐

text-decoration:overline;

text-decoration:line-through;

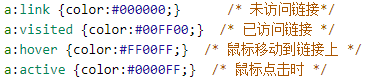
text-decoration:underline wavy red;波浪线

dotted点线

white-space:nowrap;不换行pre保留空白（包括空格和换行）

font-style:italic斜体

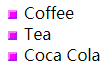
font-variant:small-caps;



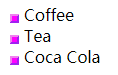
注意： a:hover 必须在 a:link 和 a:visited 之后，需要严格按顺序才能看到效果。

注意： a:active 必须在 a:hover 之后。

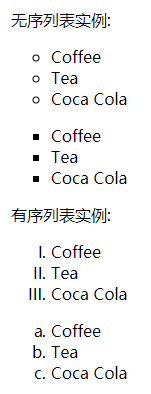
list-style-image:url('sqpurple.gif');







list-style-type:upper-roman;



td

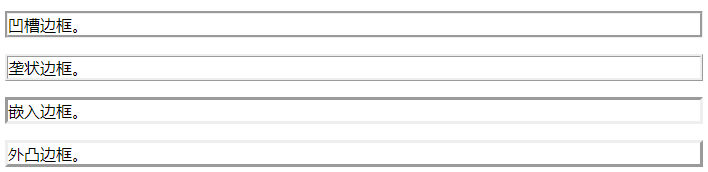
{

height:50px;

vertical-align:bottom;

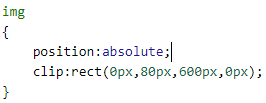
}底部对齐

border-style:groove;



outline:green dotted thick;

position: sticky;粘性定位



剪切

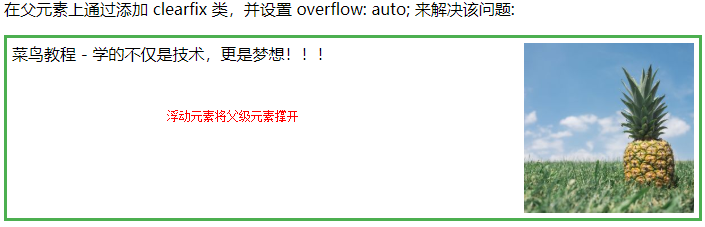
cursor:crosshair

cursor:e-resize

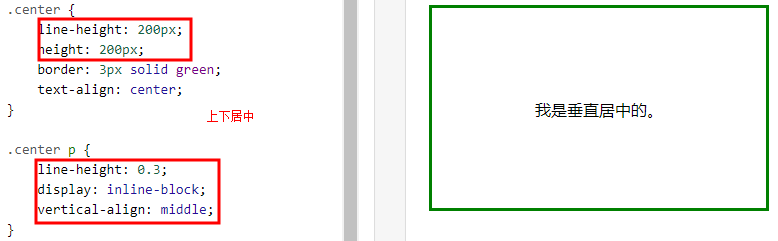
cursor:help

cursor:wait鼠标形状

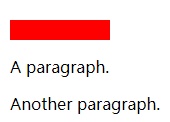
clear 属性指定元素两侧不能出现浮动元素。

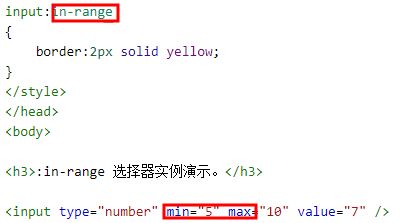


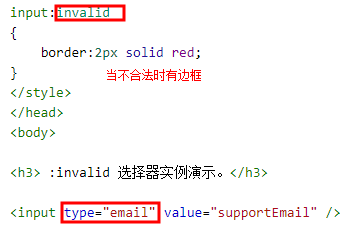




input:checked



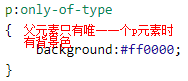


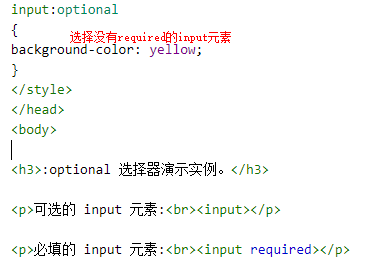


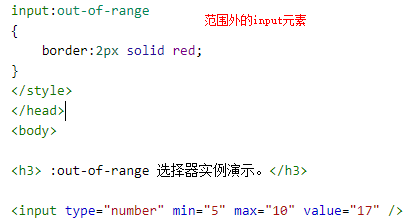
:not(p) {

color: #ff0000;

}







<input readonly>



e.replace(/[^\d]/g,'')

str.replace(/MICROSOFT/i,"W3School")这是替换单词，不是替换字母

var x = 9.656;

x.toFixed(2); 指定多少位小数

var x = 9.656;

x.toPrecision(2); 指定长度

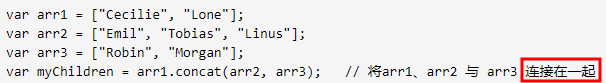
Number()不能有空格parseInt("10 years")可以有空格，返回第一个数字











字符串和数组都可以用slice()

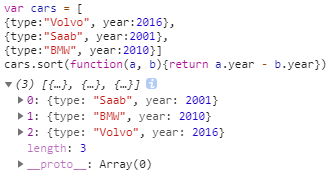
arr.sort(function(a, b){return a - b})该函数计算 40-100，然后返回 -60（负值）。

排序函数将把 40 排序为比 100 更低的值。

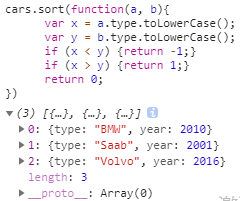
arr.sort(function(a, b){return 0.5 - Math.random()})以随机顺序排序数组，也就是打乱顺序

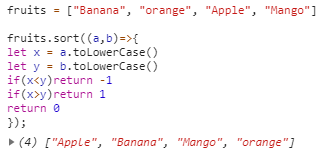
数组中的最大数



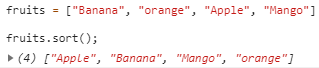
根据对象数组中的某一属性进行排序

对对象数组进行排序

不区分大小写进行排序对象数组

不区分大小写进行排序数字数组

按字母反向排序

区分大小写