# 指令v-\*

## 指令概述

指令：带有 v- 前缀的特殊属性。用于在表达式的值改变时，将某些行为应用到 DOM 上

参数：在指令后以冒号指明。例如， v-bind 指令被用来响应地更新 HTML 属性

修饰符：

v-for v-if v-bind: / : v-on:/ @

v-model : 实现双向数据绑定

## v-bind/:强制数据绑定

1) 功能: 指定变化的属性值

2) 完整写法: v-bind:xxx='yyy' //yyy 会作为表达式解析执行

3) 简洁写法: :xxx='yyy'

## v-on/@绑定事件监听

1) 功能: 绑定指定事件名的回调函数

2) 完整写法:

v-on:keyup='xxx'

v-on:keyup='xxx(参数)'

v-on:keyup.enter='xxx'

3) 简洁写法:

@keyup='xxx'

@keyup.enter='xxx'

# vue内置指令

## 常用

### v-text/v-html: 指定标签体

v-text : 当作纯文本

v-html : 将value作为html标签来解析

### v-if v-else v-show: 显示/隐藏元素

v-if : 如果vlaue为true, 当前标签会输出在页面中

v-else : 与v-if一起使用, 如果value为false, 将当前标签输出到页面中

v-show: 就会在标签中添加display样式, 如果vlaue为true, display=block, 否则是none

### v-for : 遍历

遍历数组 : v-for="(person, index) in persons"

遍历对象 : v-for="value in person" $key

### v-on : 绑定事件监听

v-on:事件名, 可以缩写为: @事件名

监视具体的按键: @keyup.keyCode @keyup.enter

停止事件的冒泡和阻止事件默认行为: @click.stop @click.prevent

隐含对象: $event

### v-bind : 强制绑定解析表达式

html标签属性是不支持表达式的, 就可以使用v-bind

可以缩写为: :id='name'

:class

1.:class="a"

2.:class="{classA : isA, classB : isB}"

3.:class="[classA, classB]"

:style

:style="{color : color}"

### v-model

双向数据绑定

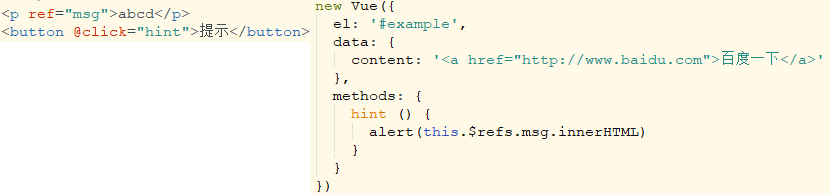
自动收集用户输入数据

### ref : 标识某个标签

ref='xxx'

读取得到标签对象: this.$refs.xxx

## Ref和$refs

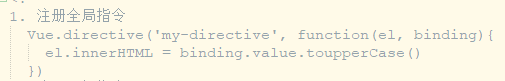


## v-cloak

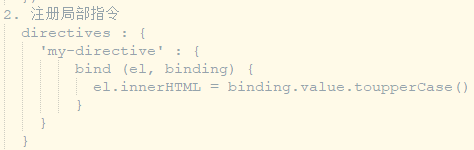
v-cloak : 使用它防止闪现表达式, 与css配合: [v-cloak] { display: none }

# 自定义指令

## 注册全局指令



## 注册局部指令



## 使用指令



