

# 目 录

## Contents

- ◆ Vue概述
- ◆ Vue基本使用
- ◆ Vue模板语法
- ◆ 基础案例
- ◆ Vue常用特性
- ◆ 综合案例



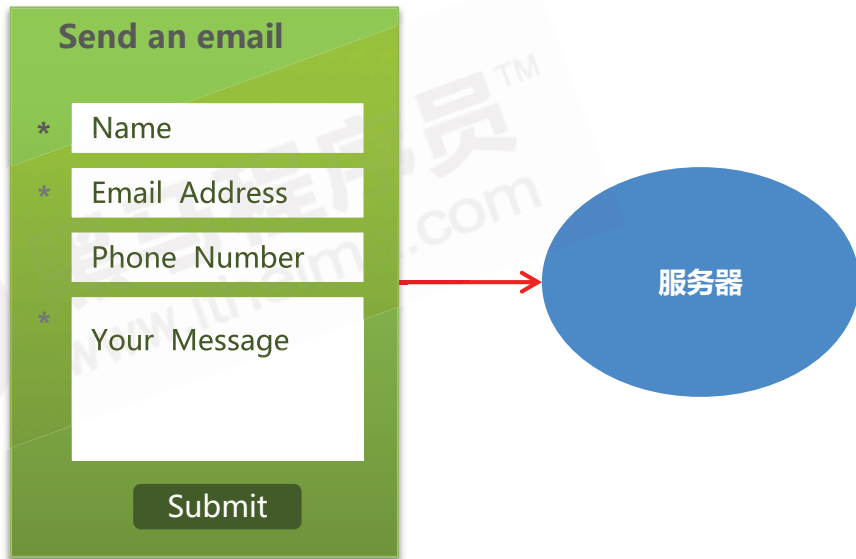
## 5.1 常用特性概览

- 表单操作
- 自定义指令
- 计算属性
- 侦听器
- 过滤器
- 生命周期

## 5.2 表单操作

### 1. 基于Vue的表单操作

- Input 单行文本
- textarea 多行文本
- select 下拉多选
- radio 单选框
- checkbox 多选框



The diagram illustrates a web form titled "Send an email" with a green header and footer. The form contains four input fields: "Name" (marked with an asterisk), "Email Address" (marked with an asterisk), "Phone Number", and "Your Message" (marked with an asterisk). A "Submit" button is located at the bottom of the form. A red arrow points from the form to a blue oval labeled "服务器" (Server). A faint watermark "www.程序员.com" is visible in the background.

Send an email

\* Name

\* Email Address

Phone Number

\* Your Message

Submit

服务器



## 5.2 表单操作

### 3. 表单域修饰符

- number: 转化为数值
- trim: 去掉开始和结尾的空格
- lazy: 将input事件切换为change事件

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



## 5.3 自定义指令

### 1. 为何需要自定义指令?

内置指令不满足需求

### 2. 自定义指令的语法规则 (获取元素焦点)

```
Vue.directive('focus' {  
  inserted: function(el) {  
    // 获取元素的焦点  
    el.focus();  
  }  
})
```

### 3. 自定义指令用法

```
<input type="text" v-focus>
```



## 5.3 自定义指令

### 3. 带参数的自定义指令（改变元素背景色）

```
Vue.directive('color', {  
  inserted: function(el, binding) {  
    el.style.backgroundColor = binding.value.color;  
  }  
})
```

### 4. 指令的用法

```
<input type="text" v-color='{color:"orange"}'>
```



## 5.3 自定义指令

### 5. 局部指令

```
directives: {  
  focus: {  
    // 指令的定义  
    inserted: function (el) {  
      el.focus()  
    }  
  }  
}
```

黑马程序员™  
www.itheima.com



## 5.4 计算属性

### 1. 为何需要计算属性?

表达式的计算逻辑可能会比较复杂, 使用计算属性可以使模板内容更加简洁

### 2. 计算属性的用法

```
computed: {  
  reversedMessage: function () {  
    return this.msg.split('').reverse().join('')  
  }  
}
```





## 5.4 计算属性

### 3. 计算属性与方法的区别

- 计算属性是基于它们的依赖进行缓存的
- 方法不存在缓存



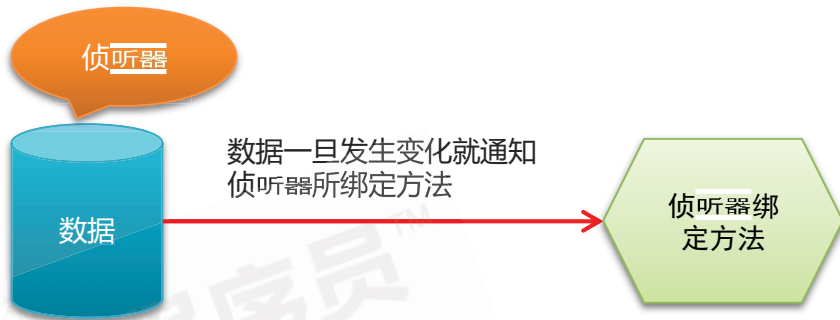
## 5.5 侦听器

### 1. 侦听器的应用场景

数据变化时执行异步或开销较大的操作

### 2. 侦听器的用法

```
watch: {  
  firstName: function(val) {  
    // val表示变化之后的值  
    this.fullName = val + this.lastName;  
  },  
  lastName: function(val) {  
    this.fullName = this.firstName + val;  
  }  
}
```



黑马程序员  
www.itheima.com



## 案例：验证用户名是否可用

用户名：

程序员

需求：输入框中输入姓名，**失去焦点**时验证是否存在，如果已经存在，提示从新输入，**如果不存在**，提示可以使用。



## 案例：需求分析

用户名：

程序员

- ① 通过v-model实现数据绑定
- ② 需要提供提示信息
- ③ **需要侦听器监听输入信息的变化**
- ④ 需要修改触发的事件



## 5.6 过滤器

### 1. 过滤器的作用是什么？

格式化数据，比如将字符串格式化为首字母大写，将日期格式化为指定的格式等



黑马程序员™  
www.itheima.com



## 5.6 过滤器

### 2. 自定义过滤器

```
Vue.filter('过滤器名称', function(value) {  
  // 过滤器业务逻辑  
})
```

### 3. 过滤器的使用

```
<div>{{msg | upper}}</div>  
<div>{{msg | upper | lower}}</div>  
<div v-bind:id="id | formatId"></div>
```

### 4. 局部过滤器

```
filters: {  
  capitalize: function() {}  
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/03504130010011333>