

Vue简介



1. JavaScript框架
2. 简化Dom操作
3. 响应式数据驱动

第一个Vue程序

- ◆ 导入开发版本的Vue.js
- ◆ 创建Vue实例对象, 设置el属性和data属性
- ◆ 使用简洁的模板语法把数据渲染到页面上

一.Vue基础



2

本地应用

传智播客旗下高端IT教育品牌

1. 内容绑定,事件绑定

v-text

v-html

v-on基础



3. 列表循环,表单元素绑定

v-for

v-on补充

v-model

v-show

v-if

v-bind

2. 显示切换,属性绑定



- ◆ 通过Vue实现常见的网页效果
- ◆ 学习Vue指令,以案例巩固知识点
- ◆ Vue指令指的是,以v-开头的一组特殊语法

1.el:挂载点

2.data:数据对象

3.v-text 设置文本

设置标签的文本值(textContent)

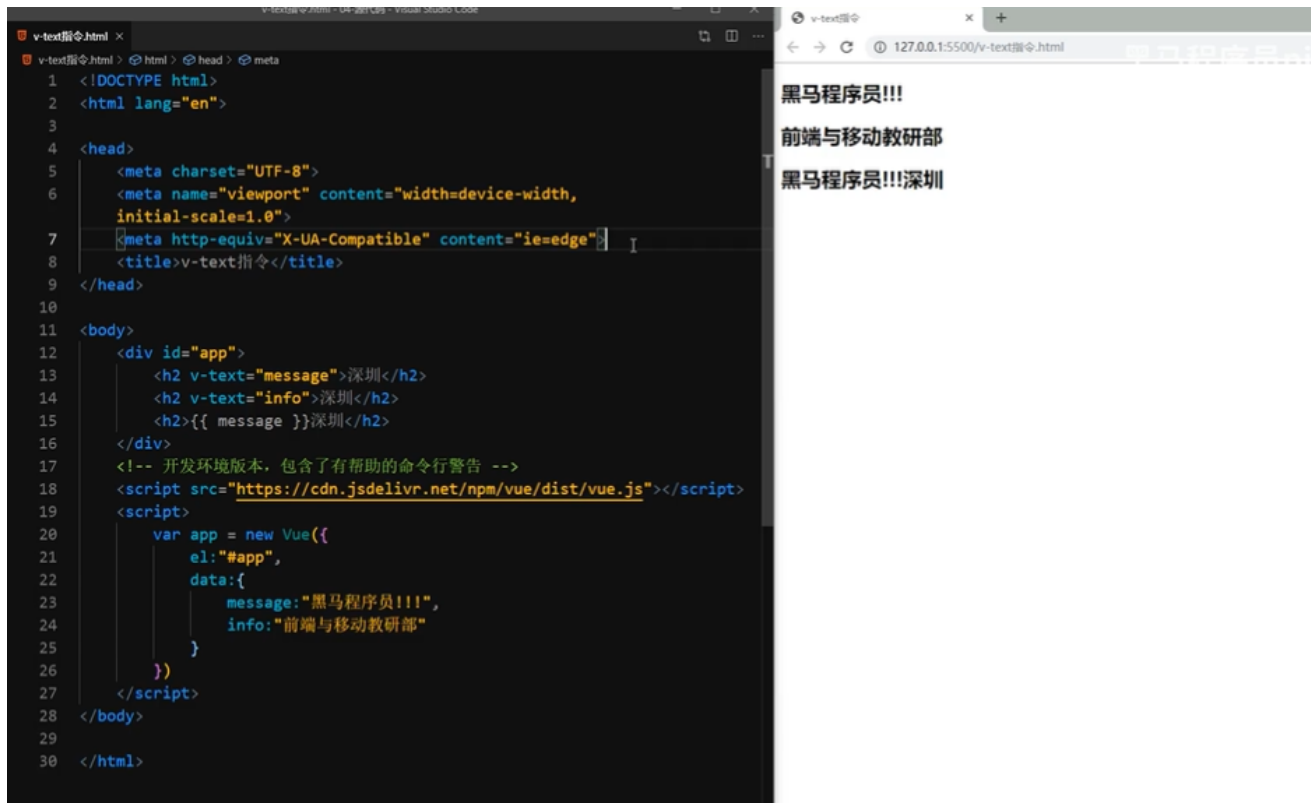
```
<div id="app">
  <h2 v-text="message+'!'"></h2>
  <h2>深圳{{ message + "! "}}</h2>
</div>
```

```
var app = new Vue({
  el:"#app",
  data:{
    message:"黑马程序员"
  }
})
```

设置标签的文本值(textContent)

```
<div id="app">
  <h2 v-text="message+'!'"></h2>
  <h2>深圳{{ message + "! "}}</h2>
</div>
```

```
var app = new Vue({
  el:"#app",
  data:{
    message:"黑马程序员"
  }
})
```

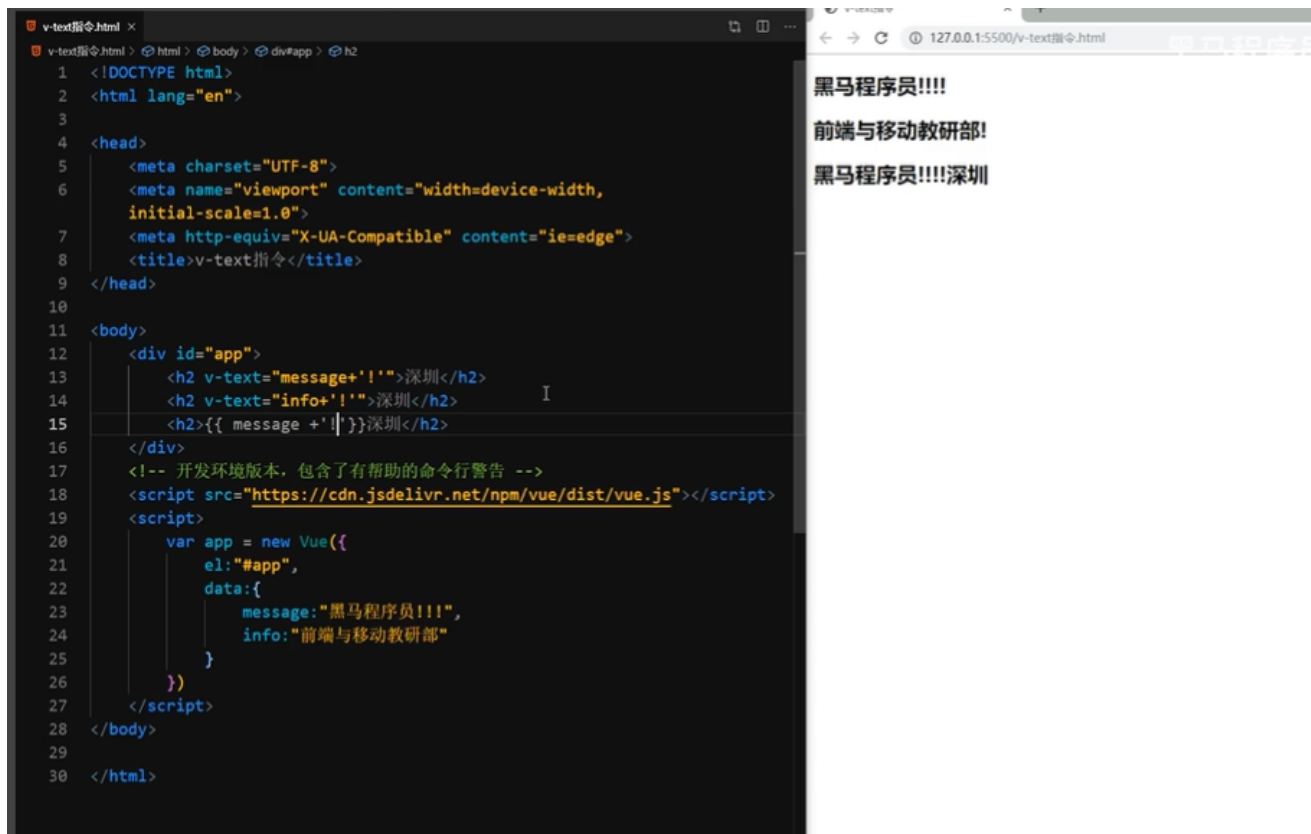


The screenshot shows a web browser window with the URL `127.0.0.1:5500/v-text指令.html`. The browser displays the following text:

- 黑马程序员!!!
- 前端与移动教研部
- 黑马程序员!!!深圳

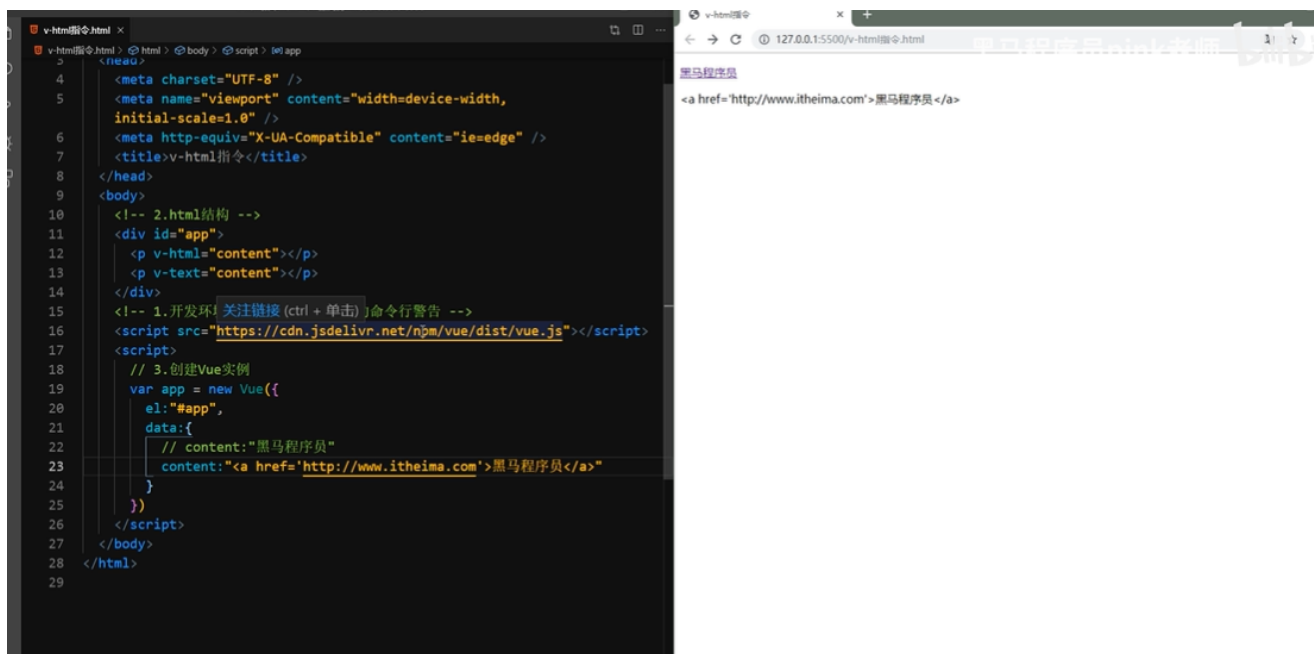
The background shows the source code of the HTML file, which includes the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>v-text指令</title>
</head>
<body>
  <div id="app">
    <h2 v-text="message">深圳</h2>
    <h2 v-text="info">深圳</h2>
    <h2>{{ message }}深圳</h2>
  </div>
  <!-- 开发环境版本，包含了有帮助的命令行警告 -->
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script>
    var app = new Vue({
      el:"#app",
      data:{
        message:"黑马程序员!!!",
        info:"前端与移动教研部"
      }
    })
  </script>
</body>
</html>
```



- ◆ **v-text**指令的作用是:设置标签的内容(textContent)
- ◆ 默认写法会替换全部内容,使用**差值表达式**`{{}}`可以替换指定内容
- ◆ 内部支持写**表达式**

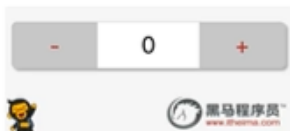
4.v-html 设置标签的innerHTML



- ◆ **v-html**指令的作用是:设置元素的**innerHTML**
- ◆ 内容中有**html**结构会被解析为**标签**
- ◆ **v-text**指令无论内容是什么,只会解析为**文本**
- ◆ 解析文本使用**v-text**,需要解析**html**结构使用**v-html**

5.v-on 为元素绑定事件

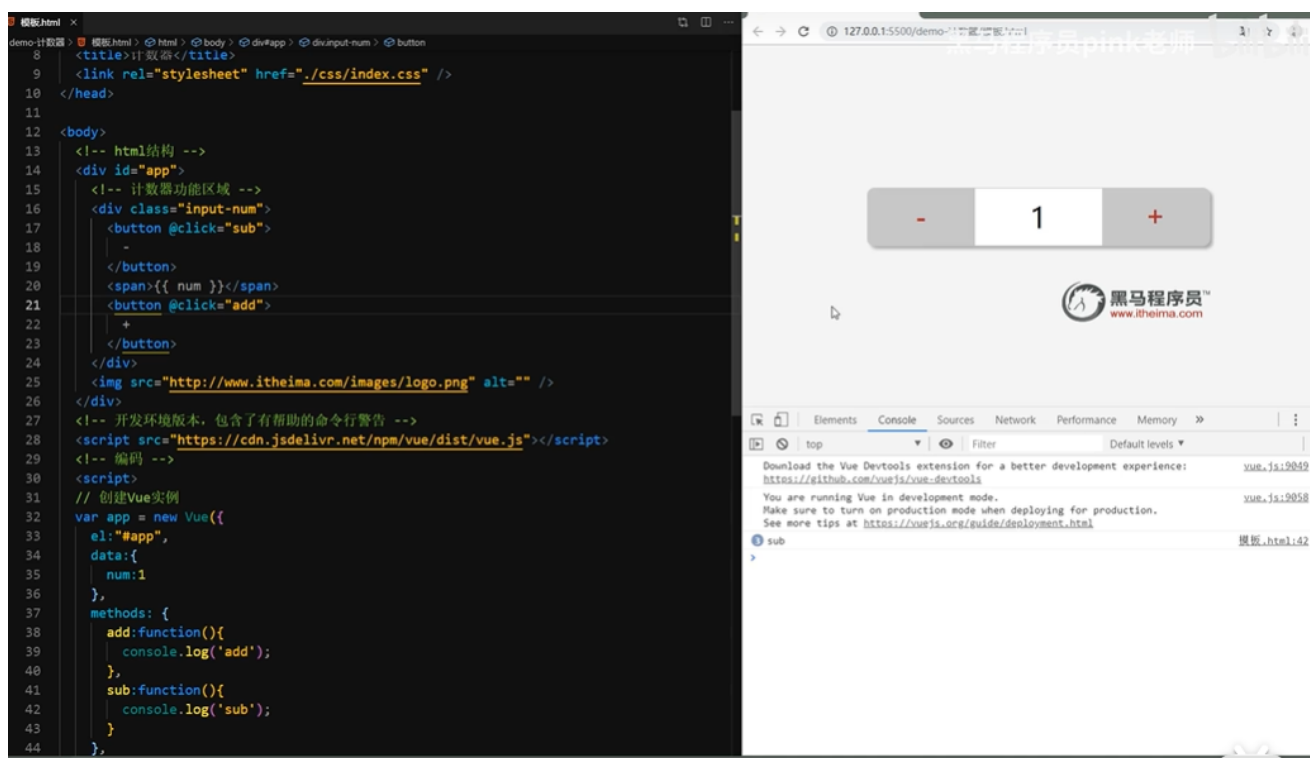
6.计数器



```
var app = new Vue({
  el: "#app",
  data: {
    num: 1
  },
  methods: {
    add: function() {},
    sub: function() {}
  }
});
```

```
<div class="input-num">
  <button @click="sub">-</button>
  <span>{{ num }}</span>
  <button @click="add">+</button>
</div>
```

1. **data**中定义数据:比如**num**
2. **methods**中添加两个方法:比如**add**(递增),**sub**(递减)
3. 使用**v-text**将**num**设置给**span**标签
4. 使用**v-on**将**add**,**sub**分别绑定给**+**,**-**按钮
5. 累加的逻辑:小于**10**累加,否则提示
6. 递减的逻辑:大于**0**递减,否则提示



- ◆ 创建Vue示例时:**el**(挂载点),**data**(数据),**methods**(方法)
- ◆ **v-on**指令的作用是绑定事件,简写为**@**
- ◆ 方法中通过**this**,关键字获取**data**中的数据
- ◆ **v-text**指令的作用是:设置元素的**文本值**,简写为**{{}}**
- ◆ **v-html**指令的作用是:设置元素的**innerHTML**

7.v-show 根据表达式的真假,切换元素的显示和隐藏

8.v-if 根据表达式的真假,切换元素的显示和隐藏(操纵dom元素)

9.v-bind 设置元素的属性

10.图片切换

1.定义图片数组

```
<div id="#app">
  
  <a href="#" @click="prev" v-show="条件">上一张</a>
  <a href="#" @click="next" v-show="条件">下一张</a>
</div>
```

2.添加图片索引

3.绑定src属性

4.图片切换逻辑

5.显示状态切换

```
var app = new Vue({
  el:"#app",
  data:{
    imgArr:[],
    index:0
  },
  methods:{
    prev:function(){},
    next:function(){},
  }
})
```