**插值**

文本

数据绑定最常见的形式就是使用 {{...}}（双大括号）的文本插值：

    <h1>{{ msg }}</h1>

Html

使用 v-html 指令用于输出 html 代码：

    <div v-html="msg"></div>

export default {

  name: 'app',

  data () {

    return {

      msg: '<h1>菜鸟教程</h1>'

    }

  }

}

属性

HTML 属性中的值应使用 v-bind 指令。

以下实例判断 class1 的值，如果use为 true 使用 class1 类的样式，否则不使用该类：

<template>

  <div id="app">

    <div v-bind:class="{'class1': use}">v-bind</div>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      use: true

    }

  }

};

</script>

表达式

Vue.js 都提供了完全的 JavaScript 表达式支持。

<template>

  <div id="app">

    {{5+5}}

    {{ ok ? 'YES' : 'NO' }}

    {{ message.split('').reverse().join('') }}

    <div v-bind:id="'list-' + id">菜鸟教程</div>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      ok: true,

      message: "RUNOOB",

      id: 1

    };

  }

};

</script>

**指令**

指令是带有 v- 前缀的特殊属性。

指令用于在表达式的值改变时，将某些行为应用到 DOM 上。如下例子：

<template>

  <div id="app">

    <p v-if="seen">现在你看到我了</p>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      seen: true

    }

  }

};

</script>

参数

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：

<template>

  <div id="app">

    <pre><a v-bind:href="url">菜鸟教程</a></pre>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      url: "http://www.runoob.com"

    };

  }

};

</script>

修饰符

修饰符是以半角句号 . 指明的特殊后缀，用于指出一个指令应该以特殊方式绑定。例如，.prevent 修饰符告诉 v-on 指令对于触发的事件调用 event.preventDefault()：

<form v-on:submit.prevent="onSubmit"></form>

**用户输入**

在 input 输入框中我们可以使用 v-model 指令来实现双向数据绑定：

<template>

  <div id="app">

    <p>{{ message }}</p>

    <input v-model="message">

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      message: 'Runoob!'

    };

  }

};

</script>

v-model 指令用来在 input、select、textarea、checkbox、radio 等表单控件元素上创建双向数据绑定，根据表单上的值，自动更新绑定的元素的值。

按钮的事件我们可以使用 v-on 监听事件，并对用户的输入进行响应。

<template>

  <div id="app">

    <p>{{ message }}</p>

    <button v-on:click="reverseMessage">反转字符串</button>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      message: "Runoob!"

    };

  },

  methods: {

    reverseMessage: function() {

      this.message = this.message

        .split("")

        .reverse()

        .join("");

    }

  }

};

</script>

**过滤器**

Vue.js 允许你自定义过滤器，被用作一些常见的文本格式化。

<template>

  <div id="app">{{ message | capitalize }}</div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      message: "runoob"

    };

  },

  filters: {

    capitalize: function(value) {

      if (!value) return "";

      value = value.toString();

      return value.charAt(0).toUpperCase() + value.slice(1);

    }

  }

};

</script>

过滤器可以串联：

{{ message | filterA | filterB }}

过滤器是 JavaScript 函数，因此可以接受参数，message 是第一个参数：

{{ message | filterA('arg1', arg2) }}

**缩写**

v-bind 缩写

Vue.js 为两个最为常用的指令提供了特别的缩写：

<!-- 完整语法 -->

<a v-bind:href="url"></a>

<!-- 缩写 -->

<a :href="url"></a>

v-on 缩写

<!-- 完整语法 -->

<a v-on:click="doSomething"></a>

<!-- 缩写 -->

<a @click="doSomething"></a>