

## 第一章 入门以及如何使用 vueJS

### 1.1 关于 vueJS 系统自带常规指令的用法

- 1 : v-show
- 2 : v-if (v-else)
- 3 : v-for
- 4 : v-bind
- 5 : v-bind: class
- 6 : v-bind:style
- 7 : v-bind : href
- 8 : v-click

1.1.2 关于 vueJS 、 avalon 、 Knockout 等 MVVM 标签渲染 原理解读

第一章到此结束 文档已 归置 github

## 第一章 入门以及如何使用 vueJS

### 1.1 关于 vueJS 系统自带常规指令的用法

#### 1 : v-show

```
<div v-show="true">
  this is true
</div>
```

#### 2 : v-if (v-else)

```
<div v-if="true">
  this is A ( true )
</div>

<div v-else>
  this is B ( fasle)
</div>
```

#### 3 : v-for

```

<div>
  <ul>
    <li v-for='item in items'>
      {{item.name}}
      <span>{{item.age}}</span>
    </li>
  </ul>
</div>

```

```

new Vue({
  el: 'body',
  data: {
    items: [
      {name: 'zhangsan', age: 21},
      {name: 'lisi', age: 22},
      {name: 'wangwu', age: 23},
      {name: 'zhao Liu', age: 24}
    ]
  }
})

```

注一：\* v-for 的 render 效果如下\*

```

<div>
  <ul>
    <li> zhangsan <span>21</span> </li>
    <li> lisi <span>22</span> </li>
    <li> wangwu <span>23</span> </li>
    <li> zhao Liu <span>24</span> </li>
  </ul>
</div>

```

#### 4 : v-bind

```

<div>
  <span v-bind='isA'></span>
</div>

```

```
new Vue({
  el: 'body',
  data: {
    isA: 'this is dataA'
  }
})
```

注一： *v-bind* 的 *render* 效果如下

```
<div>
  <span>this is dataA</span>
</div>
```

注二： *v-bind* 同样也可以直接用 胡子标签替换

```
<div>
  <span> {{isA}} </span>
</div>
```

## 5 : v-bind: class

```
<div v-bind:class="{ 'active': isActive}">
  <span> {{isA}} </span>
</div>
```

```
// 这里是css文件
.active{
  color: red;
}
```

```
new Vue({
  el: 'body',
  data: {
    isA: 'this is dataA',
    isActive: true
  }
})
```

注一： **v-bind:class** 的 *render* 效果如下

```
<div class="active">
  <span> this is dataA </span>
</div>
```

注二： **v-bind:class** 同样也可以直接用 **:class** 简写

```
<div :class="{ 'active': isActive}">
  <span> {{isA}} </span>
</div>
```

## 6 : v-bind:style

注一： **v-bind:style** 的用法同 **v-class** 类似

```
<div v-bind:style="{ color: isColorA, fontSize: fontSize + 'px'}">
  <span> this is dataA </span>
</div>
```

```

new Vue({
  el: 'body',
  data: {
    isA: 'this is dataA',
    isActive: true,
    isColorA: blue,
    fontSize: 20
  }
})

```

注二： *v-bind:style* 的 *render* 效果如下

```

<div style="color: blue; font-size: 20px;">
  <span> this is dataA </span>
</div>

```

注三： *v-bind:style* 可以简写如下

```

<div :style="{ color: isColorA, fontSize: fontSize + 'px' }">
  <span> this is dataA </span>
</div>

```

## 7 : v-bind : href

注一： *v-bind : href* 的用法同 *v-class* 类似

```

<div v-bind:style="{ color: isColorA, fontSize: fontSize + 'px' }">
  <span> this is dataA </span>
  <a v-bind:href="hrefUrl"> this is a link </a>
</div>

```

```

new Vue({
  el: 'body',
  data: {
    isA: 'this is dataA',
    isActive: true,
    isColorA: blue,
    fontSize: 20,
    hrefUrl: 'http://h5.runjiaoyu.com.cn'
  }
})

```

注二： *v-bind : href* 的 *render* 效果如下

```

<div style="color: blue; font-size: 20px;">
  <span> this is dataA </span>
  <a href='http://h5.runjiaoyu.com.cn'></a>
</div>

```

注三： *v-bind : href* 可以简写如下

```

<div :style="{ color: isColorA, fontSize: fontSize + 'px' }">
  <span> this is dataA </span>
</div>

```

注四： *v-bind : href* 同样也可以简写如下

```

<div :style="{ color: isColorA, fontSize: fontSize + 'px' }">
  <span> this is dataA </span>
  <a href={{hrefUrl}}> this is a link </a>
</div>

```

## 8 : v-click

```

    <div v-bind:style="{ color: isColorA, fontSize: fontSize + 'px' }">
      <span> this is dataA </span>
      <a href={{hrefUrl}}> this is a link </a>
      <button v-click='todo($index)'> click Me </button>
      <button v-click='nextToDo($event)'> click Me </button>
    </div>

```

```

new Vue({
  el: 'body',
  data: {
    isA: 'this is dataA',
    isActive: true,
    isColorA: blue,
    fontSize: 20,
    hrefUrl: 'http://h5.runjiaoyu.com.cn'
  },
  methods: {
    todo: function() {
      // to do something
    },
    nextToDo: function(e) {
      console.log(e.target)
      // to do something else
    }
  }
})

```

注一：v-click 可以简写如下

```

<div :style="{ color: isColorA, fontSize: fontSize + 'px' }">
  <span> this is dataA </span>
  <button @click='todo'> click Me </button>
  <button @click='nextToDo'> click Me </button>
</div>

```

## 1.1.2 关于 vueJS 、 avalon 、 Knockout 等 MVVM 标签渲染 原理 解读

```
var greeting = 'my name is ${name},age ${age}';  
var result = greeting.render({name:'XiaoMing',age:11});  
console.log(result); // my name is XiaoMing,age 11
```

我们给出一个 render 的方法 实现以上需求  
代码如下：

```
String.prototype.render = function (obj) {  
    var str = this, reg;  
  
    Object.keys(obj).forEach(function (v) {  
        reg = new RegExp('\\$\\(' + v + '\\)', 'g');  
        str = str.replace(reg, obj[v]);  
    });  
  
    return str;  
}  
  
var greeting = 'my name is ${name},age ${age}';  
var result = greeting.render({name:'XiaoMing',age:11});  
console.log(result); // my name is XiaoMing,age 11
```

第一章到此结束 文档已 归置 github