先列举一些例子

:class="['content',{'radioModel':checkType}]"

:class="['siteAppListDirNode',{open:appitem.open==true}]"

:class="['portalCenterMenu',{showNav:!showHideNav,hideNav:showHideNav}]"

:class="{shortcutMenuShow:!showHideNav,shortcutMenuHide:showHideNav}"

:style="{height:checkType?'423px':'385px'}"

:src="userInfo.userFace?userInfo.userFace:defaultHead"

再来看详细解释

Vue.js 的核心是一个响应的数据绑定系统，它允许我们在普通 HTML 模板中使用特殊的语法将 DOM “绑定”到底层数据。被绑定的DOM 将与数据保持同步，每当数据有改动，相应的DOM视图也会更新。基于这种特性，通过vue.js动态绑定class就变得非常简单。

1. 数据绑定

vue 指令以 v- 前缀标示，数据绑定的指令 v-bind:属性名, 简写为 :属性名, 简单的数据绑定例子如下：

<a v-bind:href="http://www.cnblogs.com/">博客园首页</a>

简写：

<a :href="http://www.cnblogs.com/">博客园首页</a>

2. 动态绑定class

vue 的分隔符默认是 {{ }}, 在分隔符里面的字符串会被认为是数据变量，可以通过 class="{{ className }}" 方式设置class，但是vue不推荐这种方式与 v-bind:class 的方式混用，二者只能选其一。v-bind:class 虽然与class属性里绑定变量的方式不能共存，却可以与原生的class特性共存， 一个DOM标签中允许同时出现原生class和v-bind:class。

2.1 v-bind:class 支持string类型，不建议这样使用，因为string值是固定不变的，无法实现动态改变class的需求

HTML代码：

<div :class=" 'classA classB' ">Demo1</div>

渲染后的HTML:

<div class="classA classB">Demo1</div>

2.2 v-bind:class 支持数据变量，当变量值改变时，将同时更新class。v-bind:class指令的值限定为绑定表达式，如javascript表达式

HTML代码：

<div :class="classA">Demo2</div>

Javascript代码：

data: {

classA: 'class-a' //当classA改变时将更新class

}

渲染后的HTML:

<div class="class-a">Demo2</div>

写在指令中的值会被视作表达式，如javascript表达式，因此v-bind:class接受三目运算：

HTML代码：

<div :class="classA ? 'class-a' : 'class-b' ">Demo3</div>

渲染后的HTML:

<div class="class-a">Demo3</div>

2.3 v-bind:class 支持对象，对象改变时会动态更新class

HTML代码：

<div :class="{ 'class-a': isA, 'class-b': isB}">Demo4</div>

Javascript代码：

data: {

isA: false, //当isA改变时，将更新class

isB: true //当isB改变时，将更新class

}

渲染后的HTML:

<div class="class-b">Demo4</div>

HTML代码：

<div :class="objectClass">Demo5</div>

Javascript代码：

data: {

objectClass: {

class-a: true,

class-b: false

}

}

渲染后的HTML:

<div class="class-a">Demo5</div>

2.4: v-bind:class支持数组, 数组里的变量改变时，会动态更新class列表

HTML代码：

<div :class="[classA, classB]">Demo6</div>

Javascript代码：

data: {

classA: 'class-a',

classB: 'class-b'

}

渲染后的HTML:

<div class="class-a class-b">Demo6</div>

数组中可以包含object类型，数组中的object对象改变，也会更新class列表

HTML代码：

<div :class="[classA, classB]">Demo7</div>

Javascript代码：

data: {

classA: 'class-a',

objectClass: {

classB: 'class-b', // classB 的值为class-b, 则将classB的值添加到class列表

classC: false, // classC值为false,将不添加classC

classD: true // classD 值为true，classC将被直接添加到class列表

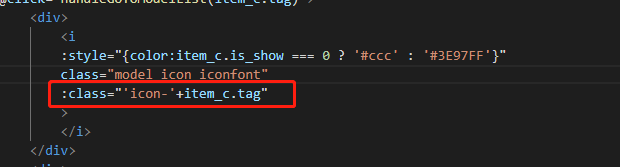
}

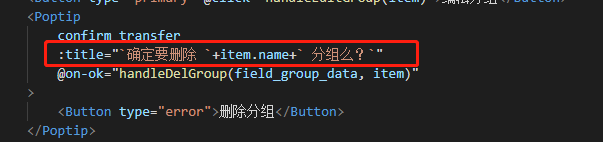
}

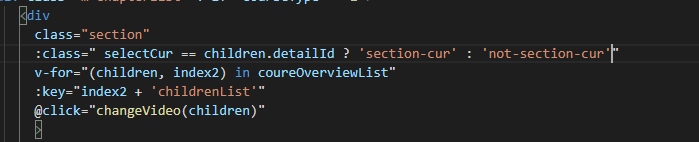
渲染后的HTML:

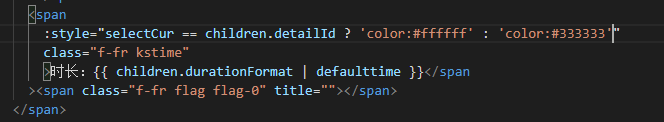
<div class="class-a class-b classD">Demo7</div>

总结：触类旁通，基本上都一样。









  <div :style="`width:${100\*item.target\_targetScore/5.0}%`"></div>

 <div :style="`width:${100\*item.target\_score/5.0}%`"></div>