上节聊了vue模板语法，计算属性侦听器，以及class和style绑定

这节谈谈vue的条件渲染

v-if

在vue中我们可以使用v-if指令实现条件块的功能

形如

<p v-if='ok'>ok</p>

<p v-else>no</p>

上面的是在Mustache中实现的功能 v-if也可以在template元素上使用

形如

<template v-if='code>1'>

<p>张男人===zsb</p>

<p>马老肥===死胖子</p>

</template>

v-if v-else-if v-else 这三种构成一组流式控制语句，其中v-else 必须在v-if和v-else-if之后 否则不生效

vue会尽可能高效的渲染元素，通常会复用元素而不是重新渲染 我们可以通过v-if v-else实现

<template v-if="loginType === 'username'">

<label>Username</label>

<input placeholder="Enter your username">

</template>

<template v-else>

<label>Email</label>

<input placeholder="Enter your email address">

</template>

在上述的代码中 除placeholder外其他的都将被替换

有时我们也需要某几个元素是完全独立的不需要复用，此时我们就需要一个唯一性的的标识==》key

<template v-if="loginType === 'username'">

<label>Username</label>

<input placeholder="Enter your username" key="username-input">

</template>

<template v-else>

<label>Email</label>

<input placeholder="Enter your email address" key="email-input">

</template>

这样的话 每一个渲染后 input框的值都将被重新渲染

v-show

有时我们需要操作DOM元素的隐藏和显示 其实就是css样式的display的none和block 而v-show就是用来实现这个功能的

带有v-show的元素始终会被渲染并保留在DOM中，只是简单的切换元素的css属性display

需要特别注意的是 v-show不支持template元素，也不支持v-else

聊聊v-if和v-show的区别

v-if是“真正”的条件渲染，因为它会确保在切换过程中条件块内的事件监听器和子组件适当的被销毁和重建

v-if也是惰性的：如果在初始渲染条件为假时，那就什么都不做--直到条件第一次变为真时才会开始渲染条件块

相比之下，v-show就简单的多了--不管初始条件是什么，元素总会被渲染，并且只是简单的基于css进行切换

一般来说，v-if 有更高的切换开销，而 v-show 有更高的初始渲染开销。因此，如果需要非常频繁地切换，

则使用 v-show 较好；如果在运行时条件很少改变，则使用 v-if 较好。

当v-if和v-for一起使用时，v-for的优先级较高一点