1) Базовая разметка:

<!doctype html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Урок 1</title>  
 <link rel="stylesheet" href="css/bootstrap.min.css">  
 <link href="css/styles.css" rel="stylesheet">  
 <script src="js/vue.js"></script>  
</head>  
<body>  
 <div class="wrapper">  
 <div class="sample">  
 <input type="text" v-model="***name***">  
 <hr>  
 <h2>Hello, {{ name }}</h2>  
 </div>  
 </div>  
  
  
 <script>  
  
 let ***app*** = ***Vue***.createApp({ // (создаем корневой компонент - могут быть другие объекты) создаем application - кусок разметки, которое вью контролирует  
 // сюда передается такой же объект, как и в new Vue()

data: ()=>({ // data - функция. которая возвращает объект () - скобки нужны для того, чтобы вернуть объект  
 name: ''  
 }),

});  
  
 let ***root*** = ***app***.mount('.sample'); // с помощью метода mount вписываем разметку в див sample  
 ***console***.log(***root***)  
 </script>  
</body>  
</html>

2) Как можно по другом переписать data ?

<script>  
 let ***app*** = ***Vue***.createApp({  
  
 data: function () {  
 return {  
 name: ''  
 }  
 }  
 });  
  
 let ***root*** = ***app***.mount('.sample');   
 ***console***.log(***root***)  
</script>

Но это древний способ.

Здесь реактивность данных работает на основе автообновления шаблона.

3) Как можно без v-model ?

<div class="wrapper">  
 <div class="sample">  
 <input type="text" v-bind:value="***name***" v-on:input="***name*** = ***$event***.target.value">  
 <hr>  
 <h2>Hello, {{ name }}</h2>  
 </div>  
</div>

4) С использованием метода

<div class="wrapper">  
 <div class="sample">  
 <input type="text" :value="***name***" @input="onInput">  
 <hr>  
 <h2>Hello, {{ name }}</h2>  
 </div>  
</div>

<script>  
 let ***app*** = ***Vue***.createApp({  
 data: function () {  
 return {  
 name: ''  
 }  
 },  
 methods:{  
 onInput(e){  
 this.name = e.target.value  
 }  
 }  
 });  
 let ***root*** = ***app***.mount('.sample');  
</script>

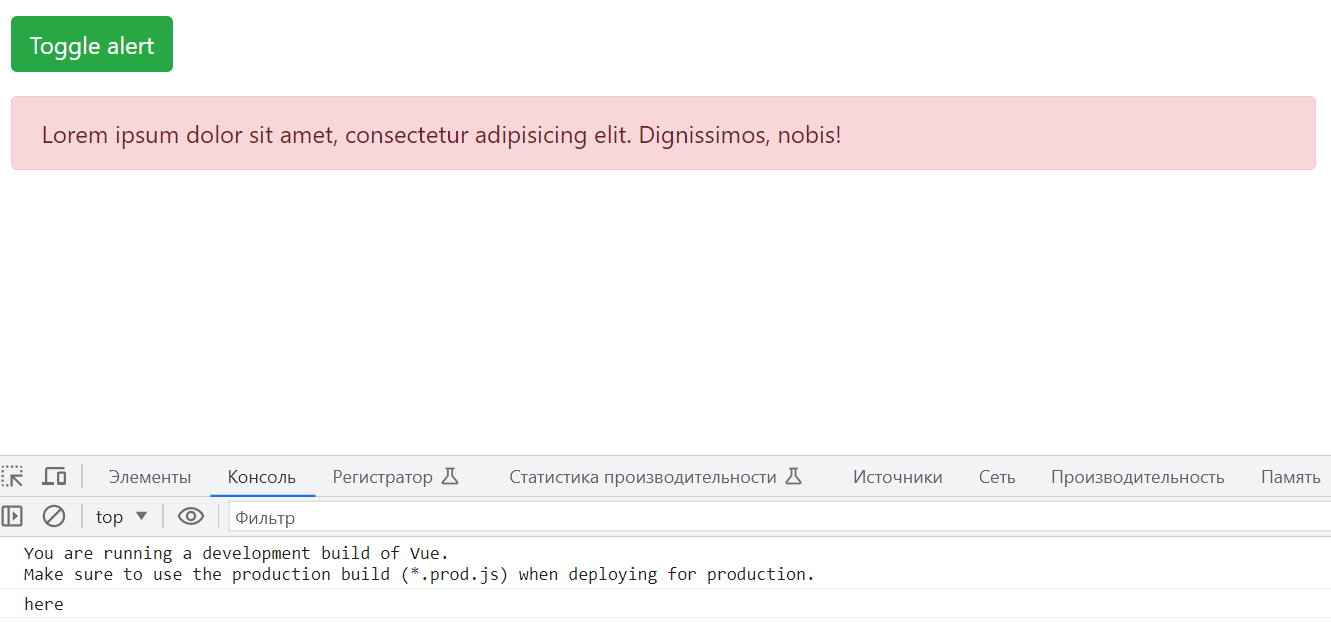
5) computed свойство

Производное поле. Т.е. данные вычисляются на основании других данных.

<script>  
 let ***app*** = ***Vue***.createApp({  
 data: function () {  
 return {  
 showAlert: false,  
 name: '',  
 firstName:'',  
 lastName:'',  
 }  
 },  
 methods:{  
 onInput(e){  
 this.name = e.target.value  
 },  
 },  
 computed:{  
 fullName(){  
 ***console***.log('here')  
 return (this.firstName + ' ' + this.lastName).trim()  
 }  
 }  
 });  
 let ***root*** = ***app***.mount('.sample');  
</script>

<h3>Hi {{fullName}}</h3> <!--но при таком потходе будет каждый раз рендерится страница-->

И теперь при toggle fullname не выполняется:



**Computed** это некий сеттер, который где-то там сохраняется.