VueJS - обчислювальні властивості

Ми вже бачили методи для екземпляру Vue та компонентів. Обчислені властивості подібні до методів, але з деякою різницею в порівнянні з методами, про які ми будемо обговорювати в цьому розділі.

Наприкінці цього розділу ми зможемо прийняти рішення про те, коли використовувати методи та коли використовувати розраховані властивості.

Давайте розуміємо обчислювальні властивості, використовуючи приклад.

Приклад

```
<html>
   <head>
     <title>VueJs Instance</title>
     <script type = "text/javascript" src = "js/vue.js"></script>
   </head>
   <body>
     <div id = "computed_props">
         FirstName : <input type = "text" v-model = "firstname" /> <br/>
        LastName : <input type = "text" v-model = "lastname"/> <br/>
        <h1>My name is {{firstname}} {{lastname}}</h1>
         <h1>Using computed method : {{getfullname}}</h1>
      </div>
      <script type = "text/javascript" src = "js/vue_computedprops.js"></script>
   </body>
</html>
```

vue_computeprops.js

```
var vm = new Vue({
   el: '#computed_props',
   data: {
      firstname :"",
      lastname :"",
      birthyear : ""
   computed :{
      getfullname : function(){
         return this.firstname +" "+ this.lastname;
})
```

Тут ми створили .html файл з ім'ям і прізвищем. Ім'я і Прізвище - це текстова скринька, яка зв'язана за допомогою властивостей, що мають ім'я та прізвище.

Ми викликаємо обчислений метод getfullname, що повертає ім'я та ім'я, що вводиться.

```
computed :{
  getfullname : function(){
      return this.firstname +" "+ this.lastname;
```

Коли ми вводимо текстове поле, те саме повертається функцією, коли змінюється ім'я або прізвище властивостей. Таким чином, за допомогою обчисленого ми не повинні робити нічого конкретного, наприклад, пам'ятаючи, що викликати функцію. При обчисленні вона отримує назву сама по собі, як властивості, що використовуються всередині змін, тобто ім'я та прізвище.

Те саме буде показано у наступному браузері. Введіть текстову скриньку і отримайте оновлення, використовуючи обчислену функцію. VueJs Instance

```
(i) localhost/vueexamples/vue_computedproperties.html
FirstName: Ben
LastName : Tennyson
My name is Ben Tennyson
Using computed method: Ben Tennyson
```

значення. У випадку методу ми називаємо це як функцію, а для обчислення як властивість. Використовуючи наступний приклад, давайте зрозуміти різницю

Тепер спробуємо зрозуміти різницю між методом та обчисленим властивістю. Обидва є об'єктами. Внутрішні функції визначені, що повертає

між методом та обчисленим властивістю.

```
<html>
  <head>
     <title>VueJs Instance</title>
      <script type = "text/javascript" src = "js/vue.js"></script>
  </head>
   <body>
     <div id = "computed_props">
         <h1 style = "background-color:gray;">Random No from computed property: {{getrandomno}}</h1>
         <h1>Random No from method: {{getrandomno1()}}</h1>
         <h1>Random No from method : {{getrandomno1()}}</h1>
         <h1 style = "background-color:gray;">Random No from computed property: {{getrandomno}}</h1>
         <h1 style = "background-color:gray;">Random No from computed property: {{getrandomno}}</h1>
         <h1 style = "background-color:gray;">Random No from computed
            property: {{getrandomno}}</h1>
         <h1>Random No from method: {{getrandomno1()}}</h1>
         <h1>Random No from method: {{getrandomno1()}}</h1>
      </div>
     <script type = "text/javascript">
         var vm = new Vue({
            el: '#computed_props',
            data: {
               name : "helloworld"
            methods: {
               getrandomno1 : function() {
                  return Math.random();
            computed :{
               getrandomno : function(){
                  return Math.random();
         });
      </script>
  </body>
</html>
```

Вона відображається в браузері, як показано нижче. Метод та обчислене властивість називають багато разів, щоб показати різницю. ⋈ VueJs Instance

У наведеному вище коді ми створили метод getrandomno1 та обчислене властивість з функцією getrandomno. Обидва дають випадкові числа,

← → C ① localhost/vueexamples/vue_computemethod.html

```
Random No from computed property: 0.48429179774559517
                              Random No from method: 0.1502662449501908
                              Random No from method: 0.9325732312551429
                              Random No from computed property: 0.48429179774559517
                              Random No from computed property: 0.48429179774559517
                              Random No from computed property: 0.48429179774559517
                              Random No from method: 0.8198847522808101
                              Random No from method: 0.5996234621480727
Якщо ми подивимось на вищевказані значення, ми побачимо, що випадкові числа, повернені з обчисленого властивості, залишаються однаковими
незалежно від кількості разів, коли він викликається. Це означає, що кожного разу він називається, останнє значення оновлюється для всіх. Тоді
як для методу це функція, отже, кожен раз, коли він називається, він повертає інше значення.
```

У цьому розділі ми дізнаємося про отримання / встановлення функцій у обчислюваних властивостях, Отримати / Встановити в обчислювальні властивості використовуючи приклад. Приклад

<html> <head>

<title>VueJs Instance</title> <script type = "text/javascript" src = "js/vue.js"></script> </head>

<html>

<head>

</head>

<body>

<body>

використовуючи Math.random ().

```
<div id = "computed_props">
         <input type = "text" v-model = "fullname" />
         <h1>{{firstName}}</h1>
         <h1>{{lastName}}</h1>
      </div>
      <script type = "text/javascript">
         var vm = new Vue({
            el: '#computed_props',
            data: {
               firstName : "Terry",
               lastName : "Ben"
            methods: {
            computed :{
               fullname : {
                  get : function() {
                     return this.firstName+" "+this.lastName;
         });
      </script>
   </body>
</html>
Ми визначили одне поле введення, яке має повне ім'я, тобто обчислене властивість. Він повертає функцію get, яка дає повну назву, тобто ім'я
та прізвище. Також ми показали ім'я та прізвище як -
<h1>{{firstName}}</h1>
<h1>{{lastName}}</h1>
Давайте перевіримо те саме в браузері.
```

⋈ VueJs Instance × (i) localhost/vueexamples/vue_computegetset.html

```
Terry Ben
Terry
```

Тепер, якщо ми змінимо ім'я в текстове поле, ми побачимо те саме, не відображається в імені, яке відображається на наступному скріншоті.

(i) localhost/vueexamples/vue_computegetset.html Ter Ben

Ben

VueJs Instance

Terry Ben Додамо функцію setter в обчисленому властивості повного імені. <script type = "text/javascript" src = "js/vue.js"></script>

<input type = "text" v-model = "fullname" /> <h1>{{firstName}}</h1> <h1>{{lastName}}</h1>

<title>VueJs Instance</title>

<div id = "computed_props">

var fname = name.split(" ");

this.firstName = fname[0];

this.lastName = fname[1]

```
</div>
      <script type = "text/javascript">
         var vm = new Vue({
            el: '#computed_props',
            data: {
               firstName : "Terry",
               lastName : "Ben"
            methods: {
            computed :{
               fullname : {
                  get : function() {
                     return this.firstName+" "+this.lastName;
                  set : function(name) {
                     var fname = name.split(" ");
                     this.firstName = fname[0];
                     this.lastName = fname[1]
         });
      </script>
   </body>
</html>
Ми додали задану функцію в обчисленому властивості повного імені.
computed :{
   fullname : {
      get : function() {
         return this.firstName+" "+this.lastName;
      set : function(name) {
```

Тепер, коли ми запускаємо код та редагуємо текстове поле, те саме буде відображатися у веб-переглядачі. Ім'я і прізвище буде оновлено через задану функцію. Функція get повертає ім'я та прізвище, а задана функція оновлює її, якщо щось відредаговане. VueJs Instance

Вона має ім'я як параметр, який є не що інше, як повне ім'я в текстовому полі. Пізніше він розбитий на простір, а ім'я та прізвище оновлюються.

① localhost/vueexamples/vue_computegetset.html JohnTerry Ben10 **John Terry** Ben10

Тепер те, що набрано в текстове поле відповідає тому, що відображається, як показано на знімку екрана вище.