

Спостерігачі. Watch

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати *set*
- для змінних-моделей – відслідковувати значення з використанням *watch*

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати **set**
- для змінних-моделей – відслідковувати значення з використанням **watch**

Загальна форма	Приклад
<pre>computed: { властивість: { get() { }, set(val) { встановлення значення виконання додаткових дій }, }, },</pre>	<pre>computed: { userName: { get() { return this.userNameValue }, set(val) { this.userNameValue = val localStorage.setItem('userName', val) }, }, },</pre>

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати **set**
- для змінних-моделей – відслідковувати значення з використанням **watch**

Приклад. Зберігати ім'я користувача у localStorage. При наступному заході завантажувати це ім'я.

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати **set**

- для змінних-моделей – відслідковувати значення з використанням **watch**

Приклад. Зберігати ім'я користувача у localStorage. При наступному заході завантажувати це ім'я.

```
<div id="app">
  <div>
    <label>
      Введіть ім'я :
      <input type="text" v-model="userName" />
    </label>
  </div>
</div>
```

При зміні значення викликається **set**, у якому можемо виконати потрібні дії (у даному випадку – це збереження імені у **localStorage**)

```
const app = createApp({
  data() {
    return {
      userNameValue: null,
    }
  },
  computed: {
    userName: {
      get() {
        return this.userNameValue
      },
      set(val) {
        this.userNameValue = val
        localStorage.setItem('userName', val)
      },
    },
  },
  mounted() {
    this.userNameValue = localStorage.getItem('userName')
  },
}).mount('#app')
```

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати **set**
- для змінних-моделей – відслідковувати значення з використанням **watch**

При зміні значення викликається метод обробник.

Загальна форма	Приклад
<pre>watch: { модель(newValue, oldValue) { потрібні дії } }</pre>	<pre>watch: { userName(newValue, oldValue) { localStorage.setItem('userName', newValue) }, },</pre>

Як виконати якісь додаткові дії при зміні значення даних?

- для обчислювальних властивостей - використати **set**
- для змінних-моделей – відслідковувати значення з використанням **watch**

При зміні значення викликається метод обробник.

Приклад. Зберігати ім'я користувача у localStorage. При наступному заході завантажувати це ім'я.

```
<div id="app">
  <div>
    <label>
      Введіть ім'я :
      <input type="text" v-model="userN&gt;
```

Введіть ім'я : Petro

```
const app = createApp({
  data() {
    return {
      userName: null,
    }
  },
  watch: {
    userName(newValue, oldValue) {
      localStorage.setItem('userName', newValue)
    },
  },
  mounted() {
    this.userName = localStorage.getItem('userName')
  },
}).mount('#app')
```

Як відслідковувати зміни полів об'єктів?

Вказуємо шлях до потрібної властивості

```
user: {  
    name: 'Ivan',  
    age: 21  
},
```

Загальна форма

```
watch: {  
    'some.nested.key' (newValue, oldValue) {  
        .....  
        потрібні дії  
        .....  
    }  
}
```

Приклад

```
watch: {  
    'user.name' (newValue, oldValue) {  
        localStorage.setItem('user', JSON.stringify(this.user))  
    },  
    'user.age' (newValue, oldValue) {  
        localStorage.setItem('user', JSON.stringify(this.user))  
    },  
},
```

Як відслідковувати зміни полів об'єктів?

Приклад. Зберігати ім'я та вік користувача у localStorage.

```
<div id="app">
  <div>
    <label>
      Введіть ім'я :
      <input type="text" v-model="user.name" />
    </label>
  </div>
  <div>
    <label>
      Введіть вік :
      <input type="number" v-model="user.age" />
    </label>
  </div>
</div>
```

Введіть ім'я : Ivan

Введіть вік : 21

```
user: {
  name: 'Ivan',
  age: 21
},
```

```
const app = createApp({
  data() {
    return {
      user: {
        name: 'Ivan',
        age: 21,
      },
    },
  },
  watch: {
    'user.name'(newValue, oldValue) {
      localStorage.setItem('user', JSON.stringify(this.user))
    },
    'user.age'(newValue, oldValue) {
      localStorage.setItem('user', JSON.stringify(this.user))
    },
  },
  mounted() {
    this.userName = JSON.parse(localStorage.getItem('user'))
  },
}).mount('#app')
```

Як відслідковувати зміни всіх полів об'єктів?

Загальна форма	Приклад
<pre>watch: { someObject: { handler(newValue, oldValue) { потрібні дії }, deep: true } }</pre>	<pre>watch: { user: { handler(newValue, oldValue) { localStorage.setItem('user', JSON.stringify(this.user)) }, deep: true, }, }</pre>

Як відслідковувати зміни всіх полів об'єкти?

Приклад. Зберігати ім'я та вік користувача у localStorage.

```
<div id="app">
  <div>
    <label>
      Введіть ім'я :
      <input type="text" v-model="user.name" />
    </label>
  </div>
  <div>
    <label>
      Введіть вік :
      <input type="number" v-model="user.age" />
    </label>
  </div>
</div>
```

Введіть ім'я : Petro

```
const app = createApp({
  data() {
    return {
      user: {}
    }
  },
  watch: {
    user: {
      handler(newValue, oldValue) {
        localStorage.setItem('user', JSON.stringify(this.user))
      },
      deep: true,
    },
    mounted() {
      this.userName = JSON.parse(localStorage.getItem('user'))
    },
  }).mount('#app')
```

Приклад.