==================================== Computed ======================================================

<https://ru.vuejs.org/v2/guide/computed.html>

При створенні компонента значення, які необхідні ми передаємо через props, а значення, які можна обчислити (як правило за допомогою інших даних з props, data або інших computed властивостей) описуємо у розділі computed . computed-властивість має вигляд функції, яка обов’язково повинна повертати значення !!! Спосіб використання такий же, як і інших властивостей з props чи data.

|  |  |
| --- | --- |
| Загальний вигляд | Приклад |
| <template>  <div>  . . .  <тег : атрибут = " **назва\_властивості** " />  . . .  <тег>  {{ **назва\_властивості** }}  </тег>  . . .  </div>  </template>  <script>  export default {  . . .  **computed**: {  **назва\_властивості** ( ) {  . . .  return **обчислене\_значення**  },  . . .  },  . . .  </script> | <template>    <table>      <tr>  <td>Назва продукту </td>  <td>{{ title }}</td> </tr>      <tr>  <td>Вартість</td> <td>{{ price }}</td> </tr>      <tr>  <td>ПДВ</td> <td>{{ tax }}</td>    </tr>      <tr>   <td>До оплати</td><td>{{ total }}</td>      </tr>    </table>  </template>  <script>  export default {    name: 'ProductInfo',    props: { // 🡨 Необхідні значення      title: {        type: String,        required: true,      },      price: {        type: Number,        default: 0,      },      tax: {        type: Number,        default: 5,      }    },    data() {      return {        delivery: 50,      }    },    computed: { // 🡨 Обчислювані значення      total() {        return this.price + (this.price \* this.tax) / 100 + this.delivery      },    },  }  </script>  <style lang="scss" scoped></style> |
|  |  |

Обчислювати значення можна і за допомогою функції. Основна відмінність від computed полягає у тому, що значення computed-властивостей кешується (якщо значення залежних реактивних даних не змінилося, то використовується попереднє обчислене значення), а при викоирстанні функції значення обчислюється кожного разу. Також при використанні функцій ми можемо у функцію передати додаткові параметри, а у computed ні.

|  |  |
| --- | --- |
| Загальний вигляд | Приклад |
| <template>  <div>  . . .  <тег : атрибут = " **назва\_функції ( парметри)** " />  . . .  <тег>  {{ **назва\_функції ( парметри)**}}  </тег>  . . .  </div>  </template>  <script>  export default {  . . .  **methods**: {  **назва\_функції** ( **форм\_парметри** ) {  . . .  return **обчислене\_значення**  },  . . .  },  . . .  </script> | <template>    <table>      <tr>  <td>Назва продукту </td>  <td>{{ title }}</td> </tr>      <tr>  <td>Вартість</td> <td>{{ price }}</td> </tr>      <tr>  <td>ПДВ</td> <td>{{ tax }}</td>    </tr>      <tr>   <td>До оплати</td><td>{{ getTotal()  }}</td>      </tr>    </table>  </template>  <script>  export default {    name: 'ProductInfo',    props: { // 🡨 Необхідні значення      title: {        type: String,        required: true,      },      price: {        type: Number,        default: 0,      },      tax: {        type: Number,        default: 5,      }    },    data() {      return {        delivery: 50,      }    },    methods: { //🡨 Аналог обчислення значення з функцією      getTotal() {        return this.price + (this.price \* this.tax) / 100 + this.delivery      },    },  }  </script>  <style lang="scss" scoped></style> |
|  |  |

Зміна обчислювального значення (get, set)

================================================== Виведення списків ============================================

|  |  |
| --- | --- |
|  |  |
|  | <template>    <div >      <ul >        <!-- <li v-for="(елемент, індекс) in колекція"> -->        <li v-for="(item, ind) in itemsList" :key="ind">          {{ item }}        </li>      </ul>    </div>  </template>  <script>  export default {    name: 'TextList',    props: {      itemsList: {        type: Array,        default: () => [],      },    },    data() {      return {        a: true,        b:1      }    },  }  </script>  <style lang="scss" scoped></style> |

|  |  |
| --- | --- |
|  |  |
|  | <template>    <div>      <div v-for="(item, index) in items" :key="item.id">        <span>{{ index + 1 }}) </span>        <span>{{ item.name }} </span>        <span>{{ item.salary }}</span>      </div>    </div>  </template>  <script>  export default {    name: 'ObjList',    props: {      items: {        type: Array,        default: () => [],      },    },  }  </script>  <style lang="scss" scoped></style> |
| Використання списку | <template>    <div id="app">      <product-info title="Cheese" :price="100" :tax="5" />      <hr />      <text-list :items-list="workersList" />      <hr />      <obj-list :items="workersList2" />    </div>  </template>  <script>  // import HelloWorld from './components/HelloWorld.vue'  import ProductInfo from './components/ProductInfo'  import TextList from './components/TextList'  import ObjList from './components/ObjList'  export default {    name: 'App',    components: {      ProductInfo,      TextList,      ObjList,    },    data() {      return {        workersList: ['Іванов', 'Петров', 'Сидоров'],        workersList2: [          {            id: '111',            name: 'Іванов',            salary: 10000,          },          {            id: '111',            name: 'Петров',            salary: 20000,          },          {            id: '111',            name: 'Сидоров',            salary: 50000,          },        ],      }    },  }  </script>  <style>  #app {    font-family: Avenir, Helvetica, Arial, sans-serif;    -webkit-font-smoothing: antialiased;    -moz-osx-font-smoothing: grayscale;    text-align: center;    color: #2c3e50;    margin-top: 60px;  }  </style> |

З об»єктами

|  |  |
| --- | --- |
|  |  |
|  | <template>    <div>      <div v-for="(item, index) in items" :key="item.id">        <span>{{ index + 1 }}) </span>        <span>{{ item.name }} </span>        <span>{{ item.salary }}</span>      </div>    </div>  </template>  <script>  export default {    name: 'ObjList',    props: {      items: {        type: Array,        default: () => [],      },    },  }  </script>  <style lang="scss" scoped></style> |
| Приколад використання | <template>    <div id="app">      <product-info title="Cheese" :price="100" :tax="5" />      <hr />      <text-list :items-list="workersList" />      <hr />      <obj-list :items="workersList2" />    </div>  </template>  <script>  // import HelloWorld from './components/HelloWorld.vue'  import ProductInfo from './components/ProductInfo'  import TextList from './components/TextList'  import ObjList from './components/ObjList'  export default {    name: 'App',    components: {      ProductInfo,      TextList,      ObjList,    },    data() {      return {        workersList: ['Іванов', 'Петров', 'Сидоров'],        workersList2: [          {            id: '111',            name: 'Іванов',            salary: 10000,          },          {            id: '111',            name: 'Петров',            salary: 20000,          },          {            id: '111',            name: 'Сидоров',            salary: 50000,          },        ],      }    },  }  </script>  <style>  #app {    font-family: Avenir, Helvetica, Arial, sans-serif;    -webkit-font-smoothing: antialiased;    -moz-osx-font-smoothing: grayscale;    text-align: center;    color: #2c3e50;    margin-top: 60px;  }  </style> |