

Умовний рендеринг

<https://ua.vuejs.org/guide/essentials/conditional.html>

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

```
<тег v-if = " умова_для_рендеринга ">
```

```
.....
```

```
</тег >
```

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

```
<тег v-if = " умова_для_рендеринга ">
    .....
</тег >
```

Приклад. Вводиться вік працівника. Виводити кнопку для реєстрації тільки тоді, коли вік >18.

```
<div id="app">
  <label> Age <input type="number" v-model="personAge" /> </label>
  <button
    v-if="isWorkerAgeValid"
  >Зареєструватись</button>
</div>
```

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

```
<div v-if = " умова_для_рендеринга ">
  .....
</div>
```

Приклад. Вводиться вік працівника. Виводити кнопку для реєстрації тільки тоді, коли вік >18.

```
<div id="app">
  <label> Age <input type="number" v-model="personAge" /> </label>
  <button
    v-if="isWorkerAgeValid"
  >Зареєструватись</button>
</div>
```

```
<script>
  const { createApp } = Vue

  const app = createApp({
    data() {
      return {
        personAge: null,
      },
    },
    computed: {
      isWorkerAgeValid() {
        return this.personAge && this.personAge >= 18
      },
    },
  }).mount('#app')
</script>
```

Age

Приклад. Користувача заходить на сайт і вводить свій вік. Якщо вік менше 16, то вивести повідомлення про заборону користування сайтом.

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

v-else -- елемент генерується тоді, коли умова не виконується

```
<тег v-if = " умова_для_рендеринга ">  
  .....  
</тег >
```

```
<тег v-else>  
  .....  
</тег >
```

**Рендеринг альтернативних елементів
в межах одного контейнера !!!**

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

v-else -- елемент генерується тоді, коли умова не виконується

Рендеринг альтернативних елементів в межах одного контейнера !!!

```
<тег v-if = " умова_для_рендеринга ">
  .....
</тег>
```

```
<тег v-else>
  .....
</тег>
```

```
<div id="app">
  <label> Age <input type="number" v-model="personAge" /> </label>
  <button v-if="isWorkerAgeValid">Зареєструватись</button>
  <div v-else>Введіть корктний вік користувача (>18)</div>
</div>
```

Приклад. Вводиться вік працівника. Виводити кнопку для реєстрації тільки тоді, коли вік >18. В іншому випадку вивести пояснююче повідомлення

Age

Введіть корктний вік користувача (>18)

Age

```
const app = createApp({
  data() {
    return {
      personAge: null,
    },
  },
  computed: {
    isWorkerAgeValid() {
      return this.personAge && this.personAge >= 18
    },
  },
}).mount('#app')
```

Приклад. Користувача заходить на сайт і вводить свій вік. Якщо вік менше 16, то вивести повідомлення про заборону користування сайтом, інакше вивести список топ товарів.

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

v-else-if -- елемент генерується тоді, коли умова виконується

v-else -- елемент генерується тоді, коли умова не виконується

```
<тег v-if = " умова_для_рендеринга_1 ">
```

```
.....
```

```
</тег >
```

```
<тег v-else-if = " умова_для_рендеринга_2 ">
```

```
.....
```

```
</тег >
```

```
<тег v-else-if = " умова_для_рендеринга_3 ">
```

```
.....
```

```
</тег >
```

```
.....
```

```
<тег v-else>
```

```
.....
```

```
</тег >
```

Рендеринг альтернативних елементів в межах одного контейнера !!!

```
<div id="app">
```

```
· · <label> Age <input type="number" v-model="personAge" /> </label>
```

```
· <div v-if="!personAge">Для реєстрації потрібно ввести вік</div>
```

```
· <div v-else-if="personAge<18">Введіть коректний вік</div>
```

```
· <button v-else>Зареєструватись</button>
```

```
</div>
```

Умовний рендеринг:

v-if -- елемент генерується тільки тоді, коли умова виконується

v-else-if -- елемент генерується тоді, коли умова виконується

v-else -- елемент генерується тоді, коли умова не виконується

```
<тег v-if = " умова_для_рендеринга_1 ">
```

```
.....
```

```
</тег >
```

```
<тег v-else-if = " умова_для_рендеринга_2 ">
```

```
.....
```

```
</тег >
```

```
<тег v-else-if = " умова_для_рендеринга_3 ">
```

```
.....
```

```
</тег >
```

```
.....
```

```
<тег v-else>
```

```
.....
```

```
</тег >
```

Age

Age

Рендеринг альтернативних елементів в межах одного контейнера !!!

```
<div id="app">
  <label> Age <input type="number" v-model="personAge" /> </label>
```

```
  <div v-if="!personAge">Для реєстрації потрібно ввести вік</div>
```

```
  <div v-else-if="personAge<18">Введіть коректний вік</div>
```

```
  <button v-else>Зареєструватись</button>
```

```
</div>
```

```
const app = createApp({
  data() {
    return {
      personAge: null,
    },
  },
  computed: {
    isWorkerAgeValid() {
      return this.personAge && this.personAge >= 18
    },
  },
}).mount('#app')
```

Приклад. З клавіатури вводиться кількість грошей користувача та ціна товару

- якщо грошей не вистачає, то повідомити скільки ще треба доплатити
- якщо грошей достатньо на купівлю ще й акційного подарунку (100 грн), то запропонувати купити (відобразити додаткову кнопку «Купити подарунок»)
- якщо кількість грошей співпадає з вартістю, то вивести смайл

Умовний рендеринг декількох елементів з однією умовою :

v-if + **template** - усі елементи всередині **template** генеруються за виконання однієї і тієїж умови

```
<template v-if = " умова_для_рендеринга_1 ">  
  . . . . дочірні елементи . . . . .  
</ template >
```

```
<template v-if="ok">  
  <h1>Заголовок</h1>  
  <p>Параграф 1</p>  
  <p>Параграф 2</p>  
</template>
```

Приклад. Користувач вводить логін. Якщо логін «admin» - відобразити поле для вводу пароля і кнопку «Увійти», інакше – вивести привітання «Раді бачити шановний Гість!».

Умовне відображення:

v-show -- елемент генерується, але не відображається (застосовується стиль **display:none**), якщо умова не виконується

```
<тег v-show = " умова_для_відображення ">
    .....
</тег >
```

```
<div id="app">
  <div>{{randomNumber}}</div>
  <button
    v-show="isVisible"
    @click="startGenerating"
  >
    Генерувати
  </button>
</div>
```

Умовне відображення:

v-show -- елемент генерується, але не відображається якщо умова не виконується

```
<тег v-show = " умова_для_відображення ">
    .....
</тег >
```

Приклад. При натисненні на кнопку кожному секунду генерується випадкове число. Якщо число парне, то кнопка зникає, непарне – з'являється.

```
<div id="app">
  <div>{{randomNumber}}</div>
  <button
    v-show="isVisible"
    @click="startGenerating"
  >
    Генерувати
  </button>
</div>
```

```
const app = createApp({
  data() {
    return {
      randomNumber: 1,
    },
  },
  computed: {
    isVisible() {
      return this.randomNumber && this.randomNumber % 2 !== 0
    },
  },
  methods: {
    startGenerating() {
      setInterval(() => {
        this.randomNumber = Math.floor(Math.random() * 10)
      }, 1000)
    },
  },
}).mount('#app')
```

Приклад. Вводиться інтервал. Після запуску відобразити таймер відліку (через секунду відображати і забирати червоний круг)