Vue.js Стартовий

Основи роботи з компонентами. Шаблони. Директиви

Vue.js Стартовий

Introduction

Кінаш Станіслав

Front-end dev

stanislav.kinash

stanislav.kinash

MCID: 9210561

Тема уроку

Основи роботи з компонентами: Шаблони. Директиви.

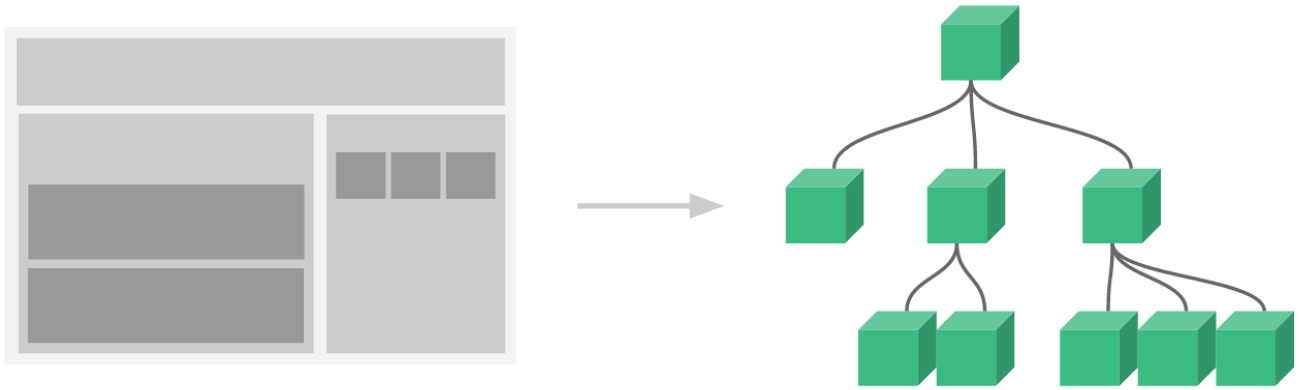
План уроку

1. Що таке компонент Vue.js?
2. Структура компонента Vue.
3. Шаблон. Робота з шаблонами Vue.
4. Директиви. Застосування директив.

Що таке компонент Vue.js?

Vue.js використовує мову шаблонів, щоб визначити, як має виглядати інтерфейс користувача. Ця мова використовує комбінацію HTML, CSS і JavaScript для визначення макета та стилю програми.

Компоненти — це будівельні блоки, які можна повторно використовувати в усій програмі. Ці компоненти створюються за допомогою мови шаблонів Vue.js і можуть використовуватися для створення складних, динамічних інтерфейсів користувача.



Vue.js Стартовий

Структура компонента Vue

1. Блок <template>

App.vue

```
<template>
  <img alt="Vue logo" src="./assets/logo.png">
  <HelloWorld msg="Welcome to Your Vue.js App"/>
</template>
```

2. Блок <script>

```
<script>
import HelloWorld from './components/HelloWorld.vue'

2 usages  neprostostas
export default {
  name: 'App',
  components: {
    HelloWorld
  }
}
</script>
```

3. Блок <style>

```
<style>
#app {
  text-align: center;
  margin-top: 60px;
  color: #2c3e50;
}
</style>
```

Шаблон. Робота з шаблонами Vue

Інтерполяції

Найпростіший спосіб зв'язування даних — це текстова інтерполяція з використанням синтаксису Mustache (подвійних фігурних дужок):

```
<span>Повідомлення: {{ msg }}</span>
```

Вираз у фігурних дужках буде замінено значенням властивості msg відповідного об'єкта даних.
Крім того, воно буде оновлено за будь-якої зміни цієї властивості (динамічно).

Шаблон. Робота з шаблонами Vue

Сирий HTML (Директива v-html)

Значення виразу у подвійних фігурних дужках підставляється як простий текст, а не як HTML. Для HTML необхідно використовувати директиву v-html:

```
<p>Інтерполяція: {{ rawHtml }}</p>
<p>Директива v-html: <span v-html="rawHtml"></span></p>
```

Вміст тега span буде замінено значенням властивості rawHtml, інтерпретованого як звичайний HTML — усі прив'язки даних ігноруються.

Шаблон. Робота з шаблонами Vue

Запам'ятайте!

Ви не можете використовувати v-html для вкладення шаблонів один в одного, тому що двигун шаблонів Vue не ґрунтується на рядках.

Натомість потрібно використовувати компоненти, що дозволяють поєднувати та перевикористовувати елементи UI.

Шаблон. Робота з шаблонами Vue

Атрибути

Синтаксис подвійних фігурних дужок не працює з HTML-атрибутами. Використовуйте замість нього директиву v-bind:

```
<div v-bind:id="dynamicId"></div>
```

Шаблон. Робота з шаблонами Vue

При використанні булевих атрибутів (коли їх наявність вже означає true) v-bind працює трохи інакше. У цьому прикладі:

```
<button v-bind:disabled="isButtonDisabled">Кнопка</button>
```

Якщо значенням isButtonDisabled буде null, undefined чи false, то атрибут disabled не додасться до елемента <button>.

Шаблон. Робота з шаблонами Vue

Використання виразів JavaScript

Поки ми пов'язували дані з властивостями у шаблонах лише за простими ключами. Однак насправді при зв'язуванні даних Vue підтримує всю потужність виразів JavaScript:

```
{{ number + 1 }}

{{ ok ? 'YES' : 'NO' }}

{{ message.split('').reverse().join('') }}

<div v-bind:id="'list-' + id"></div>
```

Шаблон. Робота з шаблонами Vue

Вирази будуть обчислені як JavaScript-код у області видимості відповідного екземпляра Vue.

Єдине обмеження в тому, що допускається лише один вираз, тому код нижче не спрацює:

```
{{ var a = 1 }}

{{ if (ok) { return message } }}
```

Директиви. Застосування директив.

Директиви

Це спеціальні атрибути із префіксом v-.

Директива реактивно застосовує DOM зміни при оновленні значення цього виразу.

```
<p v-if="seen">Hello, World!</p>
```

Директиви. Застосування директив.

Деякі директиви можуть приймати «аргумент», відокремлений від назви директиви двокрапкою.

Наприклад, директива v-bind використовується для реактивного оновлення атрибутів HTML:

```
<a v-bind:href="url"> ... </a>
```

Іншим прикладом буде директива v-on, яка відстежує події DOM:

```
<a v-on:click="doSomething"> ... </a>
```

Директиви. Застосування директив.

Скорочення v-bind

```
<!-- Повний синтаксис -->
<a v-bind:href="url"> ... </a>

<!-- Скорочений запис -->
<a :href="url"> ... </a>

<!-- Скорочений запис з динамічним
     іменем аргументу -->
<a :[key]="url"> ... </a>
```

Скорочення v-on

```
<!-- Повний синтаксис -->
<a v-on:click="doSomething"> ... </a>

<!-- Скорочений запис -->
<a @click="doSomething"> ... </a>

<!-- Скорочений запис з
динамічною назвою події -->
<a @[event]="doSomething"> ... </a>
```

Директиви. Застосування директив.

Директива v-if

Директива v-if використовується для умовного відтворення блоку. Блок буде відтворено, лише якщо вираз директиви повертає правдиве значення.

```
<h1 v-if="flag">Hello</div>
```

Директива v-else

Ви можете використовувати директиву v-else, щоб вказати «блок else» для v-if:

```
<button @click="flag = !flag">Click</div>
<h1 v-if="flag">Hello</div>
<h1 v-else>Bye</div>
```

Директиви. Застосування директив.

Директива v-else-if

Служить «блоком else if» для v-if.
Можна прив'язувати кілька разів:

```
<div v-if="type === 'A'">A</div>
<div v-else-if="type === 'B'">B</div>
<div v-else-if="type === 'C'">C</div>
<div v-else>Not A/B/C</div>
```

Директиви. Застосування директив.

Директива v-show

Іншим варіантом умовного відображення елемента є директива v-show. Використання в основному однакове:

```
<h1 v-show="show">Hi everyone!</h1>
```

Різниця полягає в тому, що елемент з v-show завжди відображатиметься та залишатиметься в DOM; v-show лише перемикає властивість CSS відображення елемента.

Директиви. Застосування директив.

Директива v-for

Використовуйте директиву v-for для відображення списку елементів на основі масиву даних.

Директиви. Застосування директив.

Відображення масиву елементів за допомогою v-for:

```
<ul id="example-1">
  <li v-for="item in items" :key="item.message">
    {{ item.message }}
  </li>
</ul>
```

```
const items = [
  { message: 'Foo' },
  { message: 'Bar' }
]
```

Директива v-for має особливий синтаксис запису:
item in items, де items — вихідний масив, а item — посилання на поточний елемент масиву:

Результат:

- Foo
- Bar

Директиви. Застосування директив.

Директива v-for для об’єкта

```
<ul id="v-for-object" class="test">
  <li v-for="(value, name, index) in object">
            {{ index }}. {{ name }}: {{ value }}
     </li>
</ul>
```

```
const object = {
    title: 'How to do lists in Vue’,
    author: 'Jane Doe’,
    publishedAt: '2016-04-10'
}
```

Результат:

0. title: How to do lists in Vue
1. author: Jane Doe
2. publishedAt: 2016-04-10

Директиви. Застосування директив.

Замість in роздільником можна використовувати of, як в ітераторах JavaScript:

```
<div v-for="item of items"></div>
```

Директиви. Застосування директив.

Директиви v-for і v-if

Коли присутні разом на одному елементі, v-for має більший пріоритет, ніж v-if. Тому v-if виконуватиметься кожної ітерації циклу. Це корисно, коли потрібно відобразити лише деякі елементи списку, наприклад:

```
<li v-for="todo in todos" v-if="!todo.isComplete">
  {{ todo }}
</li>
```

Директиви. Застосування директив.

Якщо необхідно за умовою пропускати виконання всього циклу, перемістіть v-if на зовнішній елемент.

Наприклад:

```
<ul v-if="flag">
    <li v-for="todo in todos">
        {{ todo }}
    </li>
</ul>
```


ITVDN
IT VIDEO DEVELOPERS NETWORK


TestProvider
Pearson
CyberBionic
systematics
Microsoft Partner
Windows Azure
EBA

Перевірка знань

TestProvider.com

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. За його допомогою ви можете оцінити свій рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Перевірте, як ви засвоїли даний матеріал на TestProvider.com

Успішне проходження фінального тестування дозволить отримати відповідний Сертифікат.

Vue.js Стартовий

Дякую за увагу! До нових зустрічей!

Кінаш Станіслав
Front-end dev

MCID: 9210561