



Слоты

ПЕРЕДАЧА ЗНАЧЕНИЙ ОТ РОДИТЕЛЬСКОГО КОМПОНЕНТА ДОЧЕРНЕМУ

Рассмотрим пример:

```
<div id="app">
  <form-component
    :author="authorLabel"
    :title="titleLabel">
  </form-component>
</div>
<script>
  const FormComponent = {
    template: `<div>
      <form>
        <label for="title">{{title}}</label>
        <input id="title" type="text"/><br/>
        <label for="author">{{author}}</label>
        <input id="title" type="text"/><br/>
        <button>Submit</button>
      </form>
    </div>`,
    props: ['author', 'title']
  };
};
```

ПЕРЕДАЧА ЗНАЧЕНИЙ ОТ РОДИТЕЛЬСКОГО КОМПОНЕНТА ДОЧЕРНЕМУ

Рассмотрим пример (продолжение):

```
const app = new Vue({
  el: '#app',
  components: {
    'form-component': FormComponent
  },
  data: function() {
    return {
      authorLabel: 'The author:',
      titleLabel: 'The title:'
    }
  }
});
</script>
```

По мере усложнения формы придется иметь дело с дополнительными атрибутами. Поля, предназначенные для ISBN, даты и года публикации, потребуют создания в компоненте новых входных параметров и атрибутов. Отслеживание большого количества свойств может оказаться утомительным и привести к ошибкам в вашем коде. Облегчить такую ситуацию помогут слоты.

ВВЕДЕНИЕ В СЛОТЫ

Слоты представляют способ создания фиксированной структуры компонента, при котором содержимое для различных частей компонента определяет родительский компонент.

Во Vue.js слоты реализуются через элемент `<slot>`, вместо которого родительский компонент вставляет содержимое в дочерний компонент.

ВВЕДЕНИЕ В СЛОТЫ

Рассмотрим пример:

```
<div id="app">
  <ads>
    <div>
      <p>Открыт набор на курсы Web Разработчик(NodeJS).</p>
      <p>Заявки принимаются до 22 октября</p>
    </div>
  </ads>
</div>
<script>
  Vue.component('ads', {
    template: `<div>
      <h3>Объявление</h3>
      <slot></slot>
    </div>`
  });
  new Vue({
    el: '#app'
  });
</script>
```

ВВЕДЕНИЕ В СЛОТЫ

Пример с содержимым по умолчанию:

```
<div id="app">
  <ads></ads>
  <ads>
    <div>
      <p>Открыт набор на курсы Web Разработчик(NodeJS).</p>
      <p>Заявки принимаются до 22 октября</p>
    </div>
  </ads>
</div>
<script>
  Vue.component('ads', {
    template: `<div>
      <h3>Объявление</h3>
      <slot>Здесь могла бы быть ваша реклама</slot>
    </div>`
  });
  new Vue({
    el: '#app'
  });
</script>
```

ИМЕНОВАННЫЕ СЛОТЫ

Каждому слоту можно назначить определенное имя и через это имя передавать в него некоторое содержимое из родительского компонента. Для этого в родительском компоненте применяется элемент `<template>` (или на теге компонента – если одиночный слот), в котором устанавливается специальный атрибут `v-slot`. Пример:

```
<div id="app">
  <container>
    <template v-slot:header>
      <!-- Сокращение <template #header> -->
      <h1>Заголовок статьи</h1>
      <p>Дата статьи: 13/10/2017</p>
    </template>
    <!-- <template v-slot:default> -->
    <p>Первый абзац.</p>
    <p>Второй абзац.</p>
    <!-- </template> -->
    <template v-slot:footer>
      <p>Футер</p>
    </template>
  </container>
</div>
```

ИМЕНОВАННЫЕ СЛОТЫ

Пример (продолжение):

```
<script>
  Vue.component('container', {
    template: `<div>
      <header>
        <slot name="header">Заголовок по умолчанию</slot>
      </header>
      <article>
        <slot></slot>
      </article>
      <footer>
        <slot name="footer"></slot>
      </footer>
    </div>`
  });
  new Vue({
    el: '#app'
  });
</script>
```


СЛОТЫ С ОГРАНИЧЕННОЙ ОБЛАСТЬЮ ВИДИМОСТИ

Слоты с ограниченной областью видимости определяют шаблон, в который можно передавать данные из дочернего компонента. И эти данные ограничены слотом.

Для определения шаблона в родительском компоненте применяется элемент `<template>` (или на теге компонента – если одиночный слот), в котором устанавливается специальный атрибут `v-slot`. Значение этого атрибута - название объекта с данными, которые передаются от дочернего компонента.

Чтобы определить какие данные передаются от дочернего компонента слоту (входные параметры слота), необходимо в дочернем шаблоне привязать в слоту данные.

СЛОТЫ С ОГРАНИЧЕННОЙ ОБЛАСТЬЮ ВИДИМОСТИ

Пример:

```
<div id="app">
  <user>
    <template v-slot:default="slotProps">
      <!-- Или <template v-slot="slotProps"> -->
      <!-- Сокращение <template #default="slotProps">-->
      <h3>Данные о пользователе</h3>
      <p>Имя: {{ slotProps.username }}</p>
      <p>Имя: {{ slotProps.usage }}</p>
    </template>
  </user>
</div>
<script>
  Vue.component('user', {
    template: `<div class="user">
      <slot username="Tom" usage="31"></slot>
    </div>`
  });
  new Vue({
    el: '#app'
  });
</script>
```

СЛОТЫ С ОГРАНИЧЕННОЙ ОБЛАСТЬЮ ВИДИМОСТИ

Пример деструктурирование входных параметров слота:

```
<div id="app">
  <user>
    <template v-slot="{username, usage}">
      <h3>Данные о пользователе</h3>
      <p>Имя: {{ username }}</p>
      <p>Имя: {{ usage }}</p>
    </template>
  </user>
</div>
<script>
  Vue.component('user', {
    template: `<div class="user">
      <slot username="Tom" usage="31"></slot>
    </div>`
  });
  new Vue({
    el: '#app'
  });
</script>
```

СЛОТЫ С ОГРАНИЧЕННОЙ ОБЛАСТЬЮ ВИДИМОСТИ

Ещё пример:

```
<div id="app">
  <userslist :users="users">
    <template #default="userProp">
      <!-- или <template v-slot="userProp"> -->
      <li>{{ userProp.username }}</li>
    </template>
  </userslist>
</div>
<script>
  Vue.component('userslist', {
    props: ["users"],
    template: `<ul>
      <slot v-for="user in users" :username="user"></slot>
    </ul>`
  });
  new Vue({
    el: '#app',
    data: {
      users: ['Tom', 'Sam', 'Bob']
    }
  });
</script>
```

ДОПОЛНИТЕЛЬНЫЕ ССЫЛКИ

Официальное руководство по Vue.js:

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

Руководство по Vue.js (онлайн учебник)

<https://metanit.com/web/vuejs/>