

Раздел

[Веб-компоненты](#)

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться

[Редактировать на GitHub](#) → [Веб-компоненты](#) 18-го февраля 2020

Слоты теневого DOM, композиция

Многим типам компонентов, таким как вкладки, меню, галереи изображений и другие, нужно какое-то содержимое для отображения.

Так же, как встроенный в браузер `<select>` ожидает получить контент пунктов `<option>`, компонент `<custom-tabs>` может ожидать, что будет передано фактическое содержимое вкладок, а `<custom-menu>` – пунктов меню.

Код, использующий меню `<custom-menu>`, может выглядеть так:

```
1 <custom-menu>
2   <title>Сладости</title>
3   <item>Леденцы</item>
4   <item>Фруктовые тосты</item>
5   <item>Кексы</item>
6 </custom-menu>
```

...Затем компонент должен правильно его отобразить – как обычное меню с заданным названием и пунктами, обрабатывать события меню и т.д.

Как это реализовать?

Можно попробовать проанализировать содержимое элемента и динамически скопировать и переставить DOM-узлы. Это возможно, но если мы будем перемещать элементы в теновой DOM, CSS-стили документа не будут применяться, и мы потеряем визуальное оформление. Кроме того, нужно будет писать дополнительный код.

К счастью, нам этого делать не нужно. Теновой DOM поддерживает элементы `<slot>`, которые автоматически наполняются контентом из обычного, «светлого» DOM-дерева.

Именованные слоты

Давайте рассмотрим работу слотов на простом примере.

Теновой DOM `<user-card>` имеет два слота, заполняемых из обычного DOM:

```
1 <script>
2   customElements.define('user-card', class extends HTMLElement {
3     connectedCallback() {
4       this.attachShadow({mode: 'open'});
5       this.shadowRoot.innerHTML = `
6         <div>Имя:
7           <slot name="username"></slot>
8         </div>
9         <div>Дата рождения:
10          <slot name="birthday"></slot>
11        </div>
12      `;
13    }
14  });
15 </script>
16
17 <user-card>
18   <span slot="username">Иван Иванов</span>
19   <span slot="birthday">01.01.2001</span>
20 </user-card>
```

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



Имя: Иван Иванов
Дата рождения: 01.01.2001

В теновом DOM `<slot name="X">` определяет «точку вставки» – место, где отображаются элементы с `slot="X"`.

Затем браузер выполняет «композицию»: берёт элементы из обычного DOM-дерева и отображает их в соответствующих слотах теневого DOM-дерева. В результате мы получаем именно то, что хотели – компонент, который можно заполнить данными.

После выполнения скрипта структура DOM выглядит следующим образом (без учёта композиции):

```
1 <user-card>
2   #shadow-root
3     <div>Имя:
4       <slot name="username"></slot>
5     </div>
6     <div>Дата рождения:
7       <slot name="birthday"></slot>
8     </div>
9     <span slot="username">Иван Иванов</span>
10    <span slot="birthday">01.01.2001</span>
11  </user-card>
```

Мы создали теневой DOM, он изображён под `#shadow-root`. Теперь у элемента есть два DOM-дерева: обычное («светлое») и теневое.

Чтобы отобразить содержимое, для каждого `<slot name="...">` в теновом DOM браузер ищет `slot="..."` с таким же именем в обычном DOM. Эти элементы отображаются внутри слотов:

```
<user-card>
  #shadow-root
    <div>Name:
      <slot name="username"></slot>
    </div>
    <div>Birthday:
      <slot name="birthday"></slot>
    </div>
    <span slot="username">John Smith</span>
    <span slot="birthday">01.01.2001</span>
  </user-card>
```

В результате выстраивается так называемое «развёрнутое» (flattened) DOM-дерево:

```
1 <user-card>
2   #shadow-root
3     <div>Имя:
4       <slot name="username">
5         <!-- элемент слота вставляется в слот -->
6         <span slot="username">Иван Иванов</span>
7       </slot>
8     </div>
9     <div>Дата рождения:
10      <slot name="birthday">
11        <span slot="birthday">01.01.2001</span>
12      </slot>
13    </div>
14  </user-card>
```

...Но развёрнутое DOM-дерево существует только для целей отображения и обработки событий. Это то, что мы видим на экране. Оно, в некотором

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



плане, «виртуальное». Фактически в документе расположение узлов не меняется.

Это можно легко проверить, запустив `querySelector`: все узлы находятся на своих местах.

```
1 // узлы светлого DOM находятся в том же месте, в `<user`  
2 alert( document.querySelector('user-card span').length
```

Так что развёрнутый DOM составляется из теневого вставкой в слоты. Браузер использует его для рендеринга и при всплытии событий (об этом позже). Но JavaScript видит документ «как есть» – до построения развёрнутого DOM-дерева.

⚠ Атрибут `slot="..."` могут иметь только дети первого уровня

Атрибут `slot="..."` работает только на непосредственных детях элемента-хозяина теневого дерева (в нашем примере это элемент `<user-card>`). Для вложенных элементов он игнорируется.

Например, здесь второй `` игнорируется (так как он не является потомком верхнего уровня элемента `<user-card>`):

```
1 <user-card>  
2   <span slot="username">Иван Иванов</span>  
3   <div>  
4     <!-- некорректный слот, должен быть на верхнем  
5     <span slot="birthday">01.01.2001</span>  
6   </div>  
7 </user-card>
```



Если в светлом DOM есть несколько элементов с одинаковым именем слота, они добавляются в слот один за другим.



Например, этот код:

```
1 <user-card>  
2   <span slot="username">Иван</span>  
3   <span slot="username">Иванов</span>  
4 </user-card>
```

Даст такой развёрнутый DOM с двумя элементами в `<slot name="username">`:

```
1 <user-card>  
2   #shadow-root  
3     <div>Имя:  
4       <slot name="username">  
5         <span slot="username">Иван</span>  
6         <span slot="username">Иванов</span>  
7       </slot>  
8     </div>  
9     <div>Дата рождения:  
10      <slot name="birthday"></slot>  
11    </div>  
12 </user-card>
```

Содержимое слота «по умолчанию»

Если мы добавляем данные в `<slot>`, это становится содержимым «по умолчанию». Браузер отображает его, если в светлом DOM-дереве отсутствуют данные для заполнения слота.

Например, в этой части теневого дерева текст `Аноним` отображается, если в светлом дереве нет значения `slot="username"`.

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
1 <div>Имя:
2   <slot name="username">Аноним</slot>
3 </div>
```

Слот по умолчанию (первый без имени)

Первый `<slot>` в теновом дереве без атрибута `name` является слотом по умолчанию. Он будет отображать данные со всех узлов светлого дерева, не добавленные в другие слоты

Например, давайте добавим слот по умолчанию в наш элемент `<user-card>`; он будет собирать всю информацию о пользователе, не занесённую в другие слоты:

```
1 <script>
2   customElements.define('user-card', class extends HTML1
3     connectedCallback() {
4       this.attachShadow({mode: 'open'});
5       this.shadowRoot.innerHTML = `
6         <div>Имя:
7           <slot name="username"></slot>
8         </div>
9         <div>Дата рождения:
10          <slot name="birthday"></slot>
11        </div>
12        <fieldset>
13          <legend>Другая информация</legend>
14          <slot></slot>
15        </fieldset>
16      `;
17    }
18  });
19 </script>
20
21 <user-card>
22   <div>Я люблю плавать.</div>
23   <span slot="username">Иван Иванов</span>
24   <span slot="birthday">01.01.2001</span>
25   <div>...И играть в волейбол!</div>
26 </user-card>
```

Имя: Иван Иванов
Дата рождения: 01.01.2001
Другая информация
Я люблю плавать.
...И играть в волейбол!

Всё содержимое обычного дерева, не добавленное в слоты, попало в `<fieldset>` «Другая информация».

Элементы добавляются в слот по очереди, один за другим, поэтому оба элемента данных, которые не были добавлены в слоты, попадают в слот по умолчанию.

Развёрнутое DOM-дерево выглядит так:

```
1 <user-card>
2   #shadow-root
3     <div>Имя:
4       <slot name="username">
5         <span slot="username">Иван Иванов</span>
6       </slot>
7     </div>
8     <div>Дата рождения:
9       <slot name="birthday">
10        <span slot="birthday">01.01.2001</span>
11      </slot>
12    </div>
```

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
13     <fieldset>
14       <legend>Обо мне</legend>
15       <slot>
16         <div>Привет!</div>
17         <div>Я Иван!</div>
18       </slot>
19     </fieldset>
20 </user-card>
```

Пример меню

Давайте вернёмся к меню `<custom-menu>`, упомянутому в начале главы.

Мы можем использовать слоты для распределения элементов.

Вот разметка для меню `<custom-menu>`:

```
1 <custom-menu>
2   <span slot="title">Сладости</span>
3   <li slot="item">Леденцы</li>
4   <li slot="item">Фруктовые тосты</li>
5   <li slot="item">Кексы</li>
6 </custom-menu>
```

Шаблон теневого DOM-дерева с правильными слотами:

```
1 <template id="tpl">
2   <style> /* стили меню */ </style>
3   <div class="menu">
4     <slot name="title"></slot>
5     <ul><slot name="item"></slot></ul>
6   </div>
7 </template>
```

1. `` попадает в `<slot name="title">`.
2. В шаблоне много элементов `<li slot="item">`, но только один слот `<slot name="item">`. Поэтому все такие `<li slot="item">` добавляются в `<slot name="item">` один за другим, формируя список.

Развёрнутое DOM-дерево становится таким:

```
1 <custom-menu>
2   #shadow-root
3     <style> /* стили меню */ </style>
4     <div class="menu">
5       <slot name="title">
6         <span slot="title">Сладости</span>
7       </slot>
8       <ul>
9         <slot name="item">
10          <li slot="item">Леденцы</li>
11          <li slot="item">Фруктовые тосты</li>
12          <li slot="item">Кексы</li>
13        </slot>
14      </ul>
15    </div>
16 </custom-menu>
```

Можно заметить, что в валидном DOM-дерева тег `` должен быть прямым потомком тега ``. Но это развёрнутое DOM, который описывает то, как компонент отображается, в нём такая ситуация нормальна.

Осталось только добавить обработчик `click` для открытия и закрытия списка, и меню `<custom-menu>` готово:

```
1 customElements.define('custom-menu', class extends HTML
```

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
2  connectedCallback() {
3    this.attachShadow({mode: 'open'});
4
5    // tmpl -- шаблон для теневого DOM-дерева (выше)
6    this.shadowRoot.append( tmpl.content.cloneNode(true) );
7
8    // мы не можем выбирать узлы светлого DOM, поэтому
9    this.shadowRoot.querySelector('slot[name="title"]')
10     // открыть/закрыть меню
11     this.shadowRoot.querySelector('.menu').classList.toggle('open');
12   };
13 }
14 };
```

Вот полное демо:



Конечно, мы можем расширить функциональность меню, добавив события, методы и т.д.

Обновление слотов

Что если внешний код хочет динамически добавить или удалить пункты меню?

Браузер наблюдает за слотами и обновляет отображение при добавлении и удалении элементов в слотах.

Также, поскольку узлы светлого DOM-дерева не копируются, а только отображаются в слотах, изменения внутри них сразу же становятся видны.

Таким образом, нам ничего не нужно делать для обновления отображения. Но если код компонента хочет узнать об изменениях в слотах, можно использовать событие `slotchange`.

Например, здесь пункт меню вставляется динамически через 1 секунду, и заголовок меняется через 2 секунды:

```
1  <custom-menu id="menu">
2    <span slot="title">Сладости</span>
3  </custom-menu>
4
5  <script>
6    customElements.define('custom-menu', class extends HTML
7      connectedCallback() {
8        this.attachShadow({mode: 'open'});
9        this.shadowRoot.innerHTML = `<div class="menu">
10          <slot name="title"></slot>
11          <ul><slot name="item"></slot></ul>
12        </div>`;
13
14        // shadowRoot не может иметь обработчиков событий,
15        this.shadowRoot.firstElementChild.addEventListener(
16          e => alert("slotchange: " + e.target.name)
17        );
18      }
19    });
20
21    setTimeout(() => {
22      menu.insertAdjacentHTML('beforeEnd', '<li slot="item">
23    }, 1000);
24
25    setTimeout(() => {
26      menu.querySelector('[slot="title"]').innerHTML = "Нов
27    }, 2000);
28  </script>
```

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



Отображение меню обновляется каждый раз без нашего вмешательства.

Здесь есть два события `slotchange` :

1. При инициализации:

`slotchange: title` запускается сразу же, как только `slot="title"` из обычного дерева попадает в соответствующий слот.

2. Через 1 секунду:

`slotchange: item` запускается, когда добавляется новый элемент `<li slot="item">`.

Обратите внимание, что событие `slotchange` не запускается через 2 секунды, когда меняется контент `slot="title"`. Это происходит потому, что сам слот не меняется. Мы изменяем содержимое элемента, который находится в слоте, а это совсем другое.

Если мы хотим отслеживать внутренние изменения обычного DOM-дерева из JavaScript, можно также использовать более обобщённый механизм: [MutationObserver](#).

API слотов

И, наконец, давайте поговорим о методах JavaScript, связанных со слотами.

Как мы видели раньше, JavaScript смотрит на «реальный», а не на развёрнутый DOM. Но если у теневого дерева стоит `{mode: 'open'}`, то мы можем выяснить, какие элементы находятся в слоте, и, наоборот, определить слот по элементу, который в нём находится:

- `node.assignedSlot` – возвращает элемент `<slot>`, в котором находится `node`.
- `slot.assignedNodes({flatten: true/false})` – DOM-узлы, которые находятся в слоте. Опция `flatten` имеет значение по умолчанию `false`. Если явно изменить значение на `true`, она просматривает развёрнутый DOM глубже и возвращает вложенные слоты, если есть вложенные компоненты, и резервный контент, если в слоте нет узлов.
- `slot.assignedElements({flatten: true/false})` – DOM-элементы, которые находятся в слоте (то же самое, что выше, но только узлы-элементы).

Эти методы можно использовать не только для отображения содержимого, которое находится в слотах, но и для его отслеживания в JavaScript.

Например, если компонент `<custom-menu>` хочет знать, что он показывает, он может отследить событие `slotchange` и получить пункты меню из `slot.assignedElements`:

```
1 <custom-menu id="menu">
2   <span slot="title">Сладости</span>
3   <li slot="item">Леденцы</li>
4   <li slot="item">Фруктовые тосты</li>
5 </custom-menu>
6
7 <script>
8   customElements.define('custom-menu', class extends HTML
9     items = []
10
11     connectedCallback() {
12       this.attachShadow({mode: 'open'});
13       this.shadowRoot.innerHTML = `<div class="menu">
14         <slot name="title"></slot>
15         <ul><slot name="item"></slot></ul>
16       </div>`;
17
18       // слотовый элемент добавляется/удаляется/заменяется
19
20
21
```

Раздел

Веб-компоненты

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



Редактировать на GitHub



```
22     this.shadowRoot.firstElementChild.addEventListener(  
23         let slot = e.target;  
24         if (slot.name == 'item') {  
25             this.items = slot.assignedElements().map(elem => {  
26                 alert("Items: " + this.items);  
27             }  
28         });  
29     }  
30 });  
31  
32 // пункты меню обновятся через 1 секунду  
33 setTimeout(() => {  
    menu.insertAdjacentHTML('beforeEnd', '<li slot="item">');  
    }, 1000);  
</script>
```

Итого

Обычно, если у элемента есть теневое дерево, то содержимое обычного, светлого DOM не показывается. Слоты позволяют показать элементы светлого DOM на заданных местах в теневом DOM.

Существует два вида слотов:

- Именованные слоты: `<slot name="X">...</slot>` – получают элементы светлого DOM с `slot="X"`.
- Слот по умолчанию: первый `<slot>` без имени (последующие неименованные слоты игнорируются) – показывает элементы элементов светлого дерева, которые не находятся в других слотах.
- Если одному слоту назначено несколько элементов, они добавляются один за другим.
- Содержимое элемента `<slot>` используется как резервное. Оно отображается, если в слоте нет элементов из светлого дерева.



Процесс отображения элементов внутри слота называется «композицией». В результате композиции строится «развёрнутый DOM».



При композиции не происходит перемещения узлов – с точки зрения JavaScript, DOM остаётся прежним.

JavaScript может получить доступ к слотам с помощью следующих методов:

- `slot.assignedNodes/Elements()` – возвращает узлы/элементы, которые находятся внутри `slot`.
- `node.assignedSlot` – обратный метод, возвращает слот по узлу.

Если мы хотим знать, что показываем, мы можем отследить контент слота следующими способами:

- событие `slotchange` – запускается, когда слот наполняется контентом в первый раз, и при каждой операции добавления/удаления/замещения элемента в слоте, за исключением его потомков. Сам слот будет `event.target`.
- [MutationObserver](#) для более глубокого просмотра содержимого элемента в слоте и отслеживания изменений в нём.

Теперь, когда мы научились показывать элементы светлого DOM в теневом DOM, давайте посмотрим, как их правильно стилизовать. Основное правило звучит так: теневые элементы стилизуются внутри, а обычные элементы – снаружи; однако есть заметные исключения.

Мы рассмотрим их подробно в следующей главе.

Проводим [курсы по JavaScript и фреймворкам](#).



Комментарии

перед тем как писать...

Раздел

[Веб-компоненты](#)

Навигация по уроку

Именованные слоты

Содержимое слота «по умолчанию»

Слот по умолчанию (первый без имени)

Пример меню

Обновление слотов

API слотов

Итого

Комментарии

Поделиться



[Редактировать на GitHub](#)

