

 1-го августа 2019

Приём проектирования "поведение"

Материал на этой странице устарел, поэтому скрыт из оглавления сайта.

Более новая информация по этой теме находится на странице <https://learn.javascript.ru/event-delegation>.

Шаблон проектирования «поведение» (behavior) позволяет задавать хитрые обработчики на элементе декларативно, установкой специальных HTML-атрибутов и классов.

Описание

Приём проектирования «поведение» состоит из двух частей:

1. Элементу ставится атрибут, описывающий его поведение.
2. При помощи делегирования ставится обработчик на документ, который ловит все клики и, если элемент имеет нужный атрибут, производит нужное действие.

Пример

Например, добавим «поведение», которое всем элементам, у которых стоит атрибут `data-counter`, будет при клике увеличивать значение на 1 :

```
1 Счётчик:
2 <button data-counter>1</button>
3 Ещё счётчик:
4 <button data-counter>2</button>
5
6 <script>
7   document.onclick = function(event) {
8     if (!event.target.hasAttribute('data-counter')) return;
9
10    var counter = event.target;
11
12    counter.innerHTML++;
13  };
14 </script>
```

Счётчик: Ещё счётчик:

Если запустить HTML-код выше, то при клике на каждую кнопку – её значение будет увеличиваться.

Конечно, нам важны не счётчики, а общий подход, который они демонстрируют.

Элементов `data-counter` может быть сколько угодно. Новые могут добавляться в HTML в любой момент. При помощи делегирования мы, фактически, добавили новый «псевдо-стандартный» атрибут в HTML, который добавляет элементу новую возможность («поведение»).

Ещё пример

Добавим ещё поведение.

Сделаем так, что при клике на элемент с атрибутом `data-toggle-id` будет скрываться/показываться элемент с заданным `id` :

```
1 <button data-toggle-id="subscribe-mail">
2   Показать форму подписки
3 </button>
4
5 <form id="subscribe-mail" hidden>
6   Ваша почта: <input type="email">
```

```
7 </form>
8
9 <script>
10   document.onclick = function(event) {
11     var target = event.target;
12
13     var id = target.getAttribute('data-toggle-id');
14     if (!id) return;
15
16     var elem = document.getElementById(id);
17
18     elem.hidden = !elem.hidden;
19   };
20 </script>
```

Показать форму подписки

Ещё раз заметим, что мы сделали. Теперь для того, чтобы добавить скрытие-раскрытие любому элементу – даже не надо знать JavaScript, можно просто написать атрибут `data-toggle-id`.

Это бывает очень удобно – не нужно писать JavaScript-код для каждого элемента, который должен служить такой кнопкой. Просто используем поведение.

Обратите внимание: обработчик поставлен на `document`, клик на любом элементе страницы пройдёт через него, так что поведение определено глобально.

Не только атрибут

Для своих целей мы можем использовать в HTML любые атрибуты, но стандарт рекомендует для своих целей называть атрибуты `data-*`.

В обработчике `document.onclick` мы могли бы проверять не атрибут, а класс или что-то ещё, но с атрибутом – проще и понятнее всего.

Также для добавления обработчиков на `document` рекомендуется использовать `addEventListener`, чтобы можно было добавить более одного обработчика для типа события.

Итого

Шаблон «поведение» удобен тем, что сколь угодно сложное JavaScript-поведение можно «навесить» на элемент одним лишь атрибутом. А можно – несколькими атрибутами на связанных элементах.

Здесь мы рассмотрели базовый пример, который можно как угодно модифицировать и масштабировать. Важно не переусердствовать.

Приём разработки «поведение» рекомендуется использовать для расширения возможностей разметки, как альтернативу мини-фрагментам JavaScript.

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



Комментарии

перед тем как писать...