

Раздел

[Формы, элементы управления](#)

Навигация по уроку

Навигация: формы и элементы

Обратная ссылка: `element.form`

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠 → Браузер: документ, события, интерфейсы](#)
[→ Формы, элементы управления](#)

9-го июня 2020

Свойства и методы формы

Формы и элементы управления, такие как `<input>`, имеют множество специальных свойств и событий.

Работать с формами станет намного удобнее, когда мы их изучим.

Навигация: формы и элементы

Формы в документе входят в специальную коллекцию `document.forms`.

Это так называемая «именованная» коллекция: мы можем использовать для получения формы как её имя, так и порядковый номер в документе.

- 1 `document.forms.my` - форма с именем `"my"` (`name="my"`)
- 2 `document.forms[0]` - первая форма в документе

Когда мы уже получили форму, любой элемент доступен в именованной коллекции `form.elements`.

Например:

```
1 <form name="my">
2   <input name="one" value="1">
3   <input name="two" value="2">
4 </form>
5
6 <script>
7   // получаем форму
8   let form = document.forms.my; // <form name="my"> ele
9
10  // получаем элемент
11  let elem = form.elements.one; // <input name="one"> e
12
13  alert(elem.value); // 1
14 </script>
```

Может быть несколько элементов с одним и тем же именем, это часто бывает с кнопками-переключателями `radio`.

В этом случае `form.elements[name]` является коллекцией, например:

```
1 <form>
2   <input type="radio" name="age" value="10">
3   <input type="radio" name="age" value="20">
4 </form>
5
6 <script>
7   let form = document.forms[0];
8
9   let ageElems = form.elements.age;
10
11  alert(ageElems[0]); // [object HTMLInputElement]
12 </script>
```

Эти навигационные свойства не зависят от структуры тегов внутри формы. Все элементы управления формы, как бы глубоко они не находились в форме, доступны в коллекции `form.elements`.

Раздел

Формы, элементы управления

Навигация по уроку

Навигация: формы и элементы

Обратная ссылка: `element.form`

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



<fieldset> как «подформа»

Форма может содержать один или несколько элементов `<fieldset>` внутри себя. Они также поддерживают свойство `elements`, в котором находятся элементы управления внутри них.

Например:

```
1 <body>
2   <form id="form">
3     <fieldset name="userFields">
4       <legend>info</legend>
5       <input name="login" type="text">
6     </fieldset>
7   </form>
8
9   <script>
10    alert(form.elements.login); // <input name="lc
11
12    let fieldset = form.elements.userFields;
13    alert(fieldset); // HTMLFieldSetElement
14
15    // мы можем достать элемент по имени как из ф
16    alert(fieldset.elements.login == form.elements
17  </script>
18 </body>
```

Сокращённая форма записи: `form.name`

Есть более короткая запись: мы можем получить доступ к элементу через `form[index/name]`.

Другими словами, вместо `form.elements.login` мы можем написать `form.login`.

Это также работает, но есть небольшая проблема: если мы получаем элемент, а затем меняем его свойство `name`, то он всё ещё будет доступен под старым именем (также, как и под новым).

В этом легче разобраться на примере:

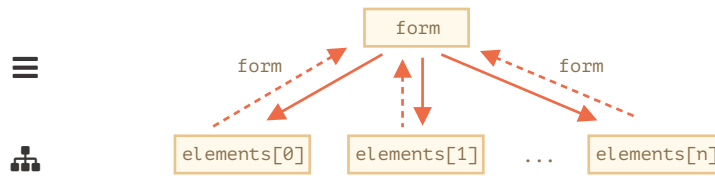
```
1 <form id="form">
2   <input name="login">
3 </form>
4
5 <script>
6   alert(form.elements.login == form.login); // true
7
8   form.login.name = "username"; // изменяем свойств
9
10  // form.elements обновили свои имена:
11  alert(form.elements.login); // undefined
12  alert(form.elements.username); // input
13
14  // а в form мы можем использовать оба имени: но
15  alert(form.username == form.login); // true
16 </script>
```

Обычно это не вызывает проблем, так как мы редко меняем имена у элементов формы.

Обратная ссылка: `element.form`

Для любого элемента форма доступна через `element.form`. Так что форма ссылается на все элементы, а эти элементы ссылаются на форму.

Вот иллюстрация:



Пример:

```
1 <form id="form">
2   <input type="text" name="login">
3 </form>
4
5 <script>
6   // form -> element
7   let login = form.login;
8
9   // element -> form
10  alert(login.form); // HTMLFormElement
11 </script>
```



Элементы формы

Рассмотрим элементы управления, используемые в формах.

input и textarea

К их значению можно получить доступ через свойство `input.value` (строка) или `input.checked` (булево значение) для чекбоксов.

Вот так:

```
1 input.value = "Новое значение";
2 textarea.value = "Новый текст";
3
4 input.checked = true; // для чекбоксов и переключателей
```

⚠ Используйте `textarea.value` вместо `textarea.innerHTML`

Обратим внимание: хоть элемент `<textarea>...</textarea>` и хранит своё значение как вложенный HTML, нам не следует использовать `textarea.innerHTML` для доступа к нему.

Там хранится только тот HTML, который был изначально на странице, а не текущее значение.

select и option

Элемент `<select>` имеет 3 важных свойства:

1. `select.options` – коллекция из подэлементов `<option>`,
2. `select.value` – значение выбранного в данный момент `<option>`,
3. `select.selectedIndex` – номер выбранного `<option>`.

Они дают три разных способа установить значение в `<select>`:

1. Найти соответствующий элемент `<option>` и установить в `option.selected` значение `true`.
2. Установить в `select.value` значение нужного `<option>`.
3. Установить в `select.selectedIndex` номер нужного `<option>`.

Первый способ наиболее понятный, но (2) и (3) являются более удобными при работе.

Раздел

Формы, элементы управления

Навигация по уроку

Навигация: формы и элементы

Обратная ссылка: `element.form`

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub

Вот эти способы на примере:

Раздел

Формы, элементы управления

Навигация по уроку

Навигация: формы и элементы

Обратная ссылка: `element.form`

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



```
1 <select id="select">
2   <option value="apple">Яблоко</option>
3   <option value="pear">Груша</option>
4   <option value="banana">Банан</option>
5 </select>
6
7 <script>
8   // все три строки делают одно и то же
9   select.options[2].selected = true;
10  select.selectedIndex = 2;
11  select.value = 'banana';
12 </script>
```

В отличие от большинства других элементов управления, `<select>` позволяет нам выбрать несколько вариантов одновременно, если у него стоит атрибут `multiple`. Эту возможность используют редко, но в этом случае для работы со значениями необходимо использовать первый способ, то есть ставить или удалять свойство `selected` у подэлементов `<option>`.

Их коллекцию можно получить как `select.options`, например:

```
1 <select id="select" multiple>
2   <option value="blues" selected>Блюз</option>
3   <option value="rock" selected>Рок</option>
4   <option value="classic">Классика</option>
5 </select>
6
7 <script>
8   // получаем все выбранные значения из select с multip
9   let selected = Array.from(select.options)
10    .filter(option => option.selected)
11    .map(option => option.value);
12
13   alert(selected); // blues,rock
14 </script>
```

Полное описание элемента `<select>` доступно в спецификации <https://html.spec.whatwg.org/multipage/forms.html#the-select-element>.

new Option

Элемент `<option>` редко используется сам по себе, но и здесь есть кое-что интересное.

В спецификации есть красивый короткий синтаксис для создания элемента `<option>`:

```
1 option = new Option(text, value, defaultSelected, selec
```

Параметры:

- `text` – текст внутри `<option>`,
- `value` – значение,
- `defaultSelected` – если `true`, то ставится HTML-атрибут `selected`,
- `selected` – если `true`, то элемент `<option>` будет выбранным.

Тут может быть небольшая путаница с `defaultSelected` и `selected`. Всё просто: `defaultSelected` задаёт HTML-атрибут, его можно получить как `option.getAttribute('selected')`, а `selected` – выбрано значение или нет, именно его важно поставить правильно. Впрочем, обычно ставят оба этих значения в `true` или не ставят вовсе (т.е. `false`).

Пример:

Раздел

Формы, элементы
управления

Навигация по уроку

Навигация: формы и
элементы

Обратная ссылка:
element.form

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



Тот же элемент, но выбранный:



```
1 let option = new Option("Текст", "value", true, true);
```

Элементы `<option>` имеют свойства:

option.selected

Выбрана ли опция.

option.index

Номер опции среди других в списке `<select>`.

option.text

Содержимое опции (то, что видит посетитель).

Ссылки

- Спецификация: <https://html.spec.whatwg.org/multipage/forms.html>.

Итого

Свойства для навигации по формам:

document.forms

Форма доступна через `document.forms[name/index]`.

form.elements

Элементы формы доступны через `form.elements[name/index]`, или можно просто использовать `form[name/index]`. Свойство `elements` также работает для `<fieldset>`.

element.form

Элементы хранят ссылку на свою форму в свойстве `form`.

Значения элементов формы доступны через `input.value`, `textarea.value`, `select.value` и т.д. либо `input.checked` для чекбоксов и переключателей.

Для элемента `<select>` мы также можем получить индекс выбранного пункта через `select.selectedIndex`, либо используя коллекцию пунктов `select.options`.

Это были основы для начала работы с формами. Далее в учебнике мы встретим ещё много примеров.

В следующей главе мы рассмотрим такие события, как `focus` и `blur`, которые могут происходить на любом элементе, но чаще всего обрабатываются в формах.

✓ Задачи

Добавьте пункт к выпадающему списку

важность: 5

Имеется `<select>`:

```
1 <select id="genres">
2   <option value="rock">Рок</option>
3   <option value="blues" selected>Блюз</option>
4 </select>
```

Используя JavaScript:

Раздел

Формы, элементы
управления

Навигация по уроку

Навигация: формы и
элементы

Обратная ссылка:
element.form

Элементы формы

Ссылки

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



решение

1. Выведите значение и текст выбранного пункта.
2. Добавьте пункт: `<option value="classic">Классика</option>`.
3. Сделайте его выбранным.

Проводим курсы по JavaScript и фреймворкам.



Комментарии

перед тем как писать...

© 2007—2020 Илья Кантор | [о проекте](#) | [связаться с нами](#) | [пользовательское соглашение](#) | [политика конфи](#)

