

Лабораторная работа № 7. Работа с формами

Цель: научиться создавать на веб-странице формы, изучить возможные элементы формы и их свойства, а также способы доступа к элементам формы.

Теория

HTML формы могут содержать такие элементы ввода как:

- текстовые поля;
- флажки;
- радио-кнопки;
- кнопки отправления и др.

Формы также могут содержать списки выбора, многострочные текстовые поля, метки и др.

Для создания формы в HTML используется тег **<form>**.

Элементы ввода используются для приема пользовательских данных. Они отличаются друг от друга в зависимости от значения атрибута **type**.

Текстовое поле **<input type="text"/>** определяет однострочное текстовое поле, в которое пользователь может вводить различную информацию.

Поле пароля **<input type="password"/>** определяет поле для ввода пароля.

Флажок **<input type="checkbox"/>** позволяет пользователям выбирать несколько пунктов с предварительно заполненной информацией из группы.

Радио-кнопка **<input type="radio"/>** позволяет выбрать только один пункт с предварительно заполненной информацией из группы.

Кнопка отправления **<input type="submit"/>** предназначена для отправления на сервер введенных данных. Адрес, на который будут пересылаться данные формы, указывается в атрибуте тэга **form – action**. Если данный атрибут отсутствует, данные будут отправлены на текущую страницу.

Текстовая область **<textarea>** позволяет ввести многострочный текст.

Список **<select>** создает выпадающий список, элементы которого определяются с помощью тэга **<option>**. С помощью атрибута **multiple** можно указать, что в выпадающем списке могут быть выбраны одновременно несколько элементов.

Заголовки в формах **<fieldset>** позволяют сгруппировать желаемую часть формы и затем с помощью тэга **<legend>** установить желаемое заглавие.

Пример формы:

Данные о пользователе

Имя:

Фамилия:

Отчество:

Укажите ваш пол:

☐ Мужской

☐ Женский

Анкета:

Какую специальность Вы считаете перспективной?

☐ Информационные системы и технологии

☐ Программное обеспечение информационных технологий

☐ ПО информационной безопасности мобильных систем

☐ Дизайн электронных и веб-изданий

Представленная выше форма создана следующим кодом:

```
<form>
<fieldset>
<legend>Данные о пользователе</legend>
Имя: <br /><input type='text' name='firstname' /><br />
Фамилия:<br /> <input type='text' name='lastname' /><br />
Отчество: <br /><input type='text' name='lastname' /><br />
<p>Укажите ваш пол:</p>
<input type='radio' name='s' value='m' /> Мужской<br />
<input type='radio' name='s' value='f' /> Женский
</fieldset>
<fieldset>
<legend> Анкета: </legend>
<p>Какую специальность Вы считаете перспективной?</p>
<input type='checkbox' name='space' value='1' />
Информационные системы и технологии<br />
<input type='checkbox' name='space' value='2' />
Программное обеспечение информационных технологий <br />
<input type='checkbox' name='space' value='3' />
ПО информационной безопасности мобильных систем <br />
<input type='checkbox' name='space' value='3' />
Дизайн электронных и веб-изданий <br />
</fieldset>
</form>
```

Доступ к элементам формы.

С помощью JavaScript можно отправить данные формы на сервер, очистить ее, а также получить доступ к любому элементу формы для изменения его значений.

Существует несколько способов обращения к формам с использованием объектной модели.

При создании формы автоматически создается массив **forms**. Для доступа к форме требуется указать номер элемента или имя формы, заданное параметром **name**.

```
<html> <body>
<form name=data>
...
</form>
<script>
alert(document.forms.length) // количество форм на странице
alert(document.forms[0].name) // имя первой формы
alert(document.forms.data.length) // количество элементов в
форме с именем data
alert(document.forms["data"].length) // То же самое
</script>
</body> </html>
```

Нумерация элементов массива всегда начинается с нуля, поэтому обращение к первой форме будет **document.forms[0]**, ко второй – **document.forms[1]**.

Второй способ – через семейство **all**. Обращение к форме происходит как к элементу массива с именем, совпадающим с именем формы – **document.all["data"]** или напрямую – **document.all.data**.

```
<html> <body>
<form name=data>
...
</form>
<script>
alert(document.all["data"].length)
alert(document.all.data.name)
</script>
</body> </html>
```

Обращение к элементам формы осуществляется посредством семейства **elements** или напрямую по имени элемента.

```
<html><body>
<form name=data> <input type=text name=userName value="Введите
ваше имя"> </form>
<script>
alert(document.forms.data.elements.length) // Общее число
элементов в форме
```

```
alert(document.forms[0].elements[0].value) // Значение первого
элемента
alert(document.forms["data"].userName.value) // Значение
элемента с именем userName
</script>
</body></html>
```

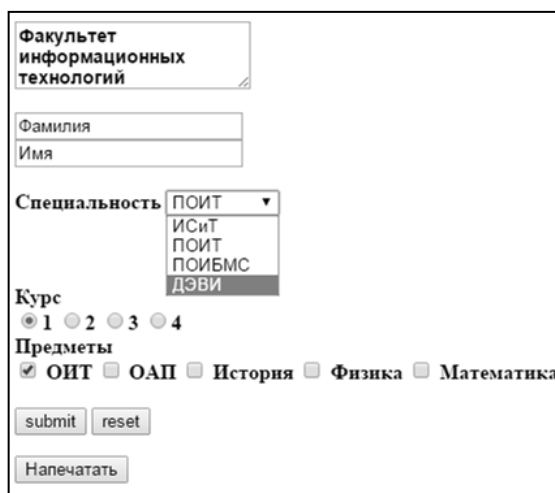
Нумерация массива **elements**, как и в случае с семейством **forms** ведется с нуля, поэтому обращение к первому элементу формы будет **elements[0]**. Для большого количество данных в форме значения элементов лучше получать по их имени.

Доступ к элементам формы можно также осуществить по их ID с помощью метода **document.getElementById()**.

Задания к лабораторной работе № 7

Задание 1. Проанализировать код для создания формы, представленный выше. Изучить правила создания элементов формы.

Задание 2. Создать форму, содержащую элементы (текстовая область, 2 текстовых поля, список, 4 радиокнопки, 5 флажков, командные кнопки). Вид формы представлен ниже.



Задание 3. Для кнопки «Напечатать» написать функцию, выводящую на страницу примерно такой текст (подчеркнутые фразы взять из элементов формы, используя обращение к ним по имени, а также по номеру элемента):

Факультет информационных технологий

Студент Фамилия специальность ИСиТ курс 1 должен сдавать следующие предметы:

- ОИТ
- История
- Математика

Задание 4. Создать еще одну форму с одним элементом – списком. Заполнить список предметами, выбранными на первой форме.

Можно придумать свою форму с разными элементами. Это приветствуется.