

## Практическая работа №3

### Формы в HTML-документах

**Цель работы:** изучение способов добавления форм на веб-страницы.

**Задачи:**

1. Изучить теоретический материал
2. Выполнить практическое задание
3. Ответить на контрольные вопросы.
4. Оформить отчет.

**Теоретический материал**

Формы передают информацию программам-обработчикам в виде пар [имя переменной]=[значение переменной]. Имена переменных следует задавать латинскими буквами.

Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Форма открывается тегом <FORM> и заканчивается меткой </FORM>. HTML-документ может содержать в себе несколько форм, однако формы не должны находиться одна внутри другой.

Тег <FORM> может содержать три атрибута, один из которых является обязательным:

**ACTION** Обязательный атрибут. Определяет, где находится обработчик формы.


**METHOD** Определяет, каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику. Допустимые значения: METHOD=POST и METHOD=GET. Если значение атрибута не установлено, по умолчанию предполагается METHOD=GET.

**GET:** методом "get" HTTP браузер берёт значение action, добавляет "?" к нему, затем присоединяет набор данных формы, закодированный с использованием типа содержимого "application/x-www-form-urlencoded". Затем перенаправляет всё по гиперссылке на этот URL. В этом сценарии данные формы ограничены кодами ASCII (нельзя использовать спецсимволы) и имеют весьма жесткие ограничения на объем вводимой информации.

**POST:** методом "post" HTTP браузер проводит транзакцию HTTP "post" (в теле HTTP-запроса), используя значение атрибута action и сообщение, созданное в соответствии с типом содержимого, определённым атрибутом enctype.

**ENCTYPE** Определяет, каким образом данные из формы будут закодированы для передачи обработчику. Если значение атрибута не установлено, по умолчанию предполагается ENCTYPE=application/x-www-form-urlencoded.

"Кнопка", чтобы запустить процесс передачи данных из формы на сервер, создается с помощью тега.

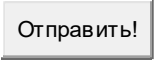
<INPUT TYPE=submit> исполнение==> 

Встретив такую строчку внутри формы, браузер нарисует на экране кнопку с надписью Submit, при нажатии на которую все имеющиеся в форме данные будут переданы обработчику, определённому в метке.

Надпись на кнопке можно задать любую путем введения атрибута VALUE="[Надпись]" например:

<INPUT TYPE=submit VALUE="Отправить!"> исполнение==> 

Надпись, нанесенную на кнопку, можно при необходимости передать обработчику путем введения в определение кнопки атрибута NAME=[имя] например:

<INPUT TYPE=submit NAME=button VALUE="Отправить!"> исполнение==> 

При нажатии на такую кнопку обработчик вместе со всеми остальными данными получит и переменную `button` со значением `Отправить!` (т.е. `button=Отправить!`, это можно видеть в адресной строке).

В форме может быть несколько кнопок типа `submit` с различными именами и/или значениями.

Обработчик, таким образом, может действовать по-разному в зависимости от того, какую именно кнопку `submit` нажал пользователь.

Существуют и другие типы элементов `<INPUT>`. Каждый элемент `<INPUT>` должен включать атрибут `NAME=[имя]`, определяющий имя переменной, которая будет передана обработчику. Имя должно задаваться только латинскими буквами. Большинство элементов `<INPUT>` должны включать атрибут `VALUE="[значение]"`, определяющий значение, которое будет передано обработчику под этим именем.

Основные типы элементов `<INPUT>`:

`TYPE=text`

Определяет окно для ввода строки текста. Может содержать дополнительные атрибуты `SIZE=[число]` (ширина поля для ввода, в символах) и `MAXLENGTH=[число]` (максимально допустимая длина вводимой строки в символах).

Пример:

```
<INPUT TYPE=text SIZE=30 NAME=student VALUE="Вася Пупкин"> ==>
```

Вася Пупкин

Определяет ширину поля в 30 символов, для ввода текста. По умолчанию в окне находится текст `Вася Пупкин`, который пользователь может редактировать. Отредактированный (или неотредактированный) текст передается обработчику в переменной `student` (`student=содержимое_поля`). Попробуйте отредактировать поле.

`TYPE=password`

Определяет окно для ввода пароля. Абсолютно аналогичен типу `text`, только вместо символов вводимого текста показывает на экране звездочки (\*), чтобы посторонний не мог прочесть.

Пример:

```
<INPUT TYPE=password NAME=pswd SIZE=20 MAXLENGTH=10> ==>
```

Определяет окно шириной 20 символов для ввода пароля. Максимально допустимая длина пароля — 10 символов. Введенный пароль передается обработчику в переменной `pswd` (`pswd=содержимое_поля`). Попробуйте ввести информацию в поле.

`TYPE=radio`

Определяет радиокнопку. Может содержать дополнительный атрибут `checked` (показывает, что кнопка помечена). В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка.

Пример:

```
<INPUT TYPE=radio NAME=modem VALUE="9600" checked> 9600 бит/с ==> ☒ 9600  
бит/с
```

```
<INPUT TYPE=radio NAME=modem VALUE="14400"> 14400 бит/с ==> ☐ 14400 бит/с
```

```
<INPUT TYPE=radio NAME=modem VALUE="28800"> 28800 бит/с ==> ☐ 28800 бит/с
```

Определяет группу из трех радиокнопок, подписанных 9600 бит/с, 14400 бит/с и 28800 бит/с. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку, обработчику будет передана переменная `modem` со значением 9600 (`modem=9600`). Если пользователь отметит вторую кнопку, обработчику будет передана переменная `modem` со значением 14400 (`modem=14400`).

`TYPE=checkbox`

Определяет квадрат, в котором можно сделать пометку. Может содержать дополнительный атрибут `checked` (показывает, что квадрат помечен). В отличие от радиокнопок, в группе квадратов с одинаковыми именами может быть несколько помеченных квадратов.

Пример:

<INPUT TYPE=checkbox NAME=comp VALUE="PC"> Персональные компьютеры ==>

☐ Персональные компьютеры

<INPUT TYPE=checkbox NAME=comp VALUE="WS" checked> Рабочие станции ==> ☒

Рабочие станции

<INPUT TYPE=checkbox NAME=comp VALUE="LAN"> Серверы локальных сетей ==>

☐ Серверы локальных сетей

<INPUT TYPE=checkbox NAME=comp VALUE="IS" checked> Серверы Интернет ==>

☒ Серверы Интернет

Определяет группу из четырех квадратов. Первоначально помечены второй и четвертый квадраты. Если пользователь не произведет изменений, обработчику будут передана одна переменная comp с двумя значениями (comp=WS и comp=IS).

TYPE=hidden

Определяет скрытый элемент данных, который не виден пользователю при заполнении формы и передается обработчику без изменений. Такой элемент иногда полезно иметь в форме, в него можно спрятать от пользователя служебные данные.

Пример:

<INPUT TYPE=hidden NAME=id VALUE="1">

Определяет скрытую переменную индексную id, которая передается обработчику со значением 1.

TYPE=reset

Определяет кнопку, при нажатии на которую форма возвращается в исходное состояние (обнуляется). Поскольку при использовании этой кнопки данные обработчику не передаются, кнопка типа reset может и не иметь атрибута name. Пример:

<INPUT TYPE=reset VALUE="Очистить поля формы"> ==>

Очистить поля формы

Определяет кнопку Очистить поля формы, при нажатии на которую форма возвращается в исходное состояние.

Элемент <SELECT>:

Меню <SELECT> из n элементов выглядит примерно так:

<SELECT NAME="[имя]">

<OPTION VALUE="[значение 1]">[текст 1]

<OPTION VALUE="[значение 2]">[текст 2]

...

<OPTION VALUE="[значение n]">[текст n]

</SELECT>

[текст 1] ▼

Метка <SELECT> содержит обязательный атрибут NAME, определяющий имя переменной.

Метка <SELECT> может также содержать атрибут MULTIPLE, присутствие которого показывает, что из меню можно выбрать несколько элементов. Большинство браузеров показывают меню <SELECT MULTIPLE> в виде окна, в котором находятся элементы меню (высоту окна в строках можно задать атрибутом SIZE=[число]). Для выбора нескольких значений одновременно удерживают кнопку "SHIFT" и выбирают значения мышкой.

<SELECT MULTIPLE SIZE=3 NAME="[имя]">

<OPTION VALUE="[значение 1]">[текст 1]

<OPTION VALUE="[значение 2]">[текст 2]

<OPTION VALUE="...">[...]

<OPTION VALUE="[значение n]">[текст n]

</SELECT>

Метка `<OPTION>` определяет элемент меню. Обязательный атрибут `VALUE` устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Метка `<OPTION>` может включать атрибут `selected`, показывающий, что данный элемент отмечен по умолчанию.

Пример:

```
<SELECT NAME="selection">
<OPTION VALUE="option1">Вариант 1
<OPTION VALUE="option2" selected>Вариант 2
<OPTION VALUE="option3">Вариант 3
</SELECT>
```

Такой фрагмент определяет меню из трех элементов: Вариант 1, Вариант 2 и Вариант 3. По умолчанию выбран элемент Вариант 2. Обработчику будет передана переменная `selection` (`selection=...`) значение которой может быть `option1` (по умолчанию), `option2` или `option3`.

Элемент `<TEXTAREA>`:

Пример:

```
<TEXTAREA NAME=address ROWS=5 COLS=50>
```

Поле для ввода большого текста, разбитого на абзацы.

```
</TEXTAREA>
```

Все атрибуты обязательны. Атрибут `NAME` определяет имя, под которым содержимое окна будет передано обработчику (в примере — `address`). Атрибут `ROWS` устанавливает высоту окна в строках (в примере — 5). Атрибут `COLS` устанавливает ширину окна в символах (в примере — 50).

Текст, размещенный между метками `<TEXTAREA>` и `</TEXTAREA>`, представляет собой содержимое окна по умолчанию. Пользователь может его отредактировать или просто стереть.

### **Практические задания**

1. Создать форму-анкету для студента. Предусмотреть ввод таких данных, как ФИО, возраст, выбор города, специальности, вуза, сферы интересов. Также предусмотреть поле ввода для дополнительной информации.

### **Контрольные вопросы**

1. Для чего используются формы в html документах?
2. Каковы атрибуты тега `<FORM>`?
3. Как добавить кнопку на форму?
4. Как добавить радиокнопку на страницу?
5. Как добавить поле ввода?
6. Как добавить выпадающее меню?
7. Как добавить поле ввода большого текста?