

ЛАБОРАТОРНАЯ РАБОТА №3 HTML-ФОРМЫ. ОБРАБОТКА ДАННЫХ, ВВЕДЕННЫХ В ФОРМУ

Цель работы: изучить основные элементы, которые могут быть размещены в HTML-форме и научиться обрабатывать данные, введенные в форму с помощью языка программирования PHP.

Ход выполнения лабораторной работы должен быть отражен в отчете. Отчет должен содержать титульный лист, номера заданий, коды программ, скриншот с результатом выполнения программы.

3.1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Примечание – Для правильного выполнения PHP-скрипта страница, содержащая такой скрипт, должна быть сохранена с расширением `.php` в каталоге, который указан в настройках сервера в директиве `DocumentRoot`. Эта директива определяет местонахождение корневого каталога документов сервера. При установке web-сервера Apache, PHP и MySQL, используя денвер, по умолчанию в директиве `DocumentRoot` устанавливается значение `Z:\home\localhost\www`. В этой папке необходимо создать папку с номером вашей группы, а в ней – папку с вашей фамилией (папки должны быть названы латинскими буквами или цифрами без пробелов и специальных символов). Для запуска созданного скрипта необходимо открыть браузер и в адресной строке ввести следующий адрес: `http://localhost/Группа/Фамилия/имя_скрипта.php`.

3.1.1 Элемент FORM

Получение и обработка данных, введенных пользователем, стали неотъемлемой частью большинства успешных web-сайтов. Ввод информации в основном реализуется с применением HTML-форм.

При вводе данных в форму используются различные управляющие элементы: поля ввода текста и пароля, скрытые поля, поля со списком, переключатели, флажки, кнопки и т. д.

Одна страница может содержать несколько форм. Обработка элементов формы производится с помощью скриптов.

Одним из распространенных языков программирования для написания скриптов, обрабатывающих данные, введенные в формы, является PHP. В официальной документации язык PHP подается как встраиваемый в HTML скриптовый язык с обработкой на сервере.

Форма в HTML-документе реализуется тегом-контейнером FORM:

```
<FORM ACTION = "действие" METHOD = "метод">
```

– элементы формы –

```
</FORM>
```

Тег **FORM** имеет следующие атрибуты.

ACTION – адрес сценария (скрипта), которому будет послана форма. Это единственный обязательный атрибут.

METHOD – определяет метод HTTP, используемый для пересылки данных формы от браузера к серверу. Атрибут **METHOD** может принимать два значения: **GET** и **POST**. Если используется метод **GET** (он же установлен по умолчанию), то все поля, описанные в форме, передаются в строке URL в следующем виде: URL?name=value&name=value. При использовании метода **POST** поля формы кодируются таким же образом, но передаются через скрытые переменные, не используя строку URL. Метод **POST** обычно используется при передаче большого объема данных или если необходимо скрыть от пользователя передаваемый набор полей.

Начальный и конечный теги **FORM** задают границы формы, поэтому их указание является обязательным.

Пример 3.1

Обработка данных, вводимых пользователем на странице form1.html, при помощи скрипта reader.php, расположенного в том же каталоге.

```
<HTML>
  <HEAD>
    <TITLE> Пример HTML-формы </TITLE>
  </HEAD>
  <BODY>
    <H1> Пример HTML-формы </H1>
    <FORM METHOD="POST" ACTION="reader.php" >
      Здесь должны быть элементы формы.
    </FORM>
  </BODY>
</HTML>
```

Задание 3.1:

3.1.1 Запустить web-сервер Apache (D:\work\Start servers.lnk).

3.1.2 Проверить работу web-сервера Apache: запустить web-браузер Internet Explorer и в адресной строке ввести адрес локального сервера <http://localhost/> (рисунок 3.1).

3.1.3 В корневом каталоге сервера Z:\home\localhost\www создать папку со своей фамилией и номером группы (желательно латинскими буквами без пробелов): Z:\home\localhost\www\171501\Ivanov\.

3.1.4 Создать HTML-документ, содержащий форму и сохранить его под именем form1.php в своем каталоге на диске Z. Заголовок HTML-документа – «Форма ввода персональных данных пользователя».

3.1.2 Текстовое поле

В текстовых полях обычно вводится короткая текстовая информация –

скажем, адрес электронной почты, почтовый адрес или имя. Особой разновидностью текстовых полей является поле ввода паролей и скрытое поле. В поле ввода пароля вводимые символы заменяются звездочками. Скрытые поля не отображаются в браузере и обычно используются для передачи служебной информации между сценариями.

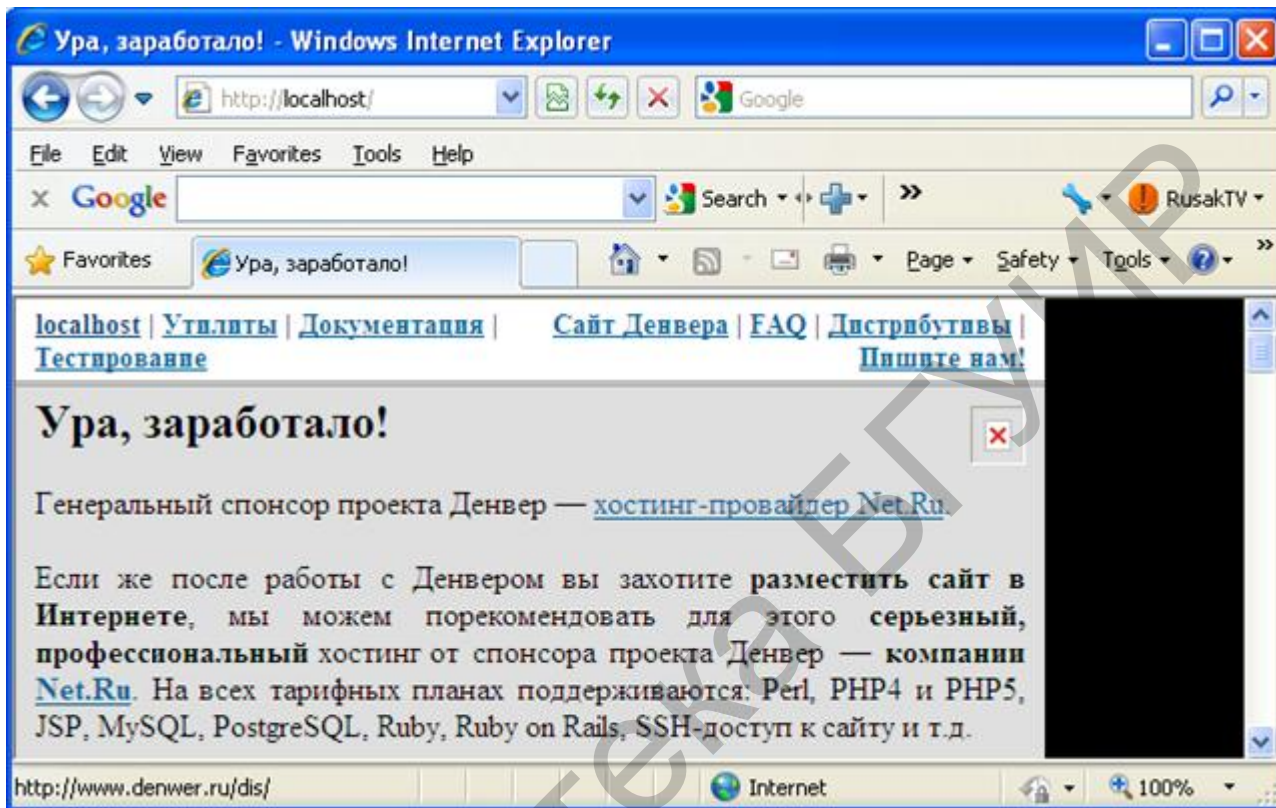


Рисунок 3.1 – Проверка работы локального web-сервера

Синтаксис определения текстового поля:

```
<INPUT TYPE="тип" NAME="имя_переменной" SIZE="N"
MAXLENGTH="N" VALUE="">
```

Определение текстового поля включает пять атрибутов.

TYPE – тип элемента (для текстовых полей – TEXT, для полей ввода паролей – PASSWORD, для скрытых полей – HIDDEN).

NAME – имя переменной, в которой сохраняются введенные данные.

SIZE – общий размер текстового поля в браузере.

MAXLENGTH – максимальное количество символов, вводимых в текстовом поле.

VALUE – значение, отображаемое в текстовом поле по умолчанию.

Пример 3.2

Создание в форме поля для ввода имени пользователя и пароля. Для удобства форматирования элементы формы можно разместить в таблице.

...

```
<FORM METHOD="POST" ACTION="reader.php" >
```

```

<TABLE>
<TR><TD>Введите ваше имя:</TD>
    <TD><INPUT NAME = "Name" TYPE="TEXT" > </TD>
</TR>
<TR><TD>Введите ваш пароль:</TD>
    <TD><INPUT NAME = "Password" TYPE="PASSWORD" > </TD>
</TR>
</TABLE>
</FORM>
...

```

Задание 3.2. Добавить в созданную форму текстовые поля для ввода ФИО пользователя, логина, пароля и электронного адреса. Все элементы формы разместить в таблице. Таблица располагается по центру, и границы таблицы не отображаются.

3.1.3 Кнопка отправки данных и кнопка сброса

Кнопка отправки данных инициирует действие, заданное атрибутом ACTION тега <FORM>. Кнопка сброса отменяет все изменения, внесенные в элементы формы. Синтаксис определения кнопок:

```
<INPUT TYPE="тип" VALUE="текст_на_кнопке">
```

Определение кнопки включает два атрибута.

TYPE – тип элемента (для кнопки отправки данных – SUBMIT, для кнопки сброса данных – RESET).

VALUE – текст, который будет отображаться на кнопке.

Пример 3.3

Добавление к предыдущему примеру кнопки отправки данных.

```

...
<FORM METHOD= "POST" ACTION= "reader.php" >
<TABLE>
  <TR><TD>...</TD> <TD>... </TD></TR>
  <TR><TD>...</TD> <TD>... </TD></TR>
  <TR><TD COLSPAN=2 >
    <INPUT TYPE="SUBMIT" VALUE="Отправить">
  </TD></TR>
</TABLE>
</FORM>
...

```

Задание 3.3. Добавить в созданную форму кнопку отправки данных и кнопку сброса.

3.1.4 Обработка введенных в форму данных

Введенные в форму данные посылаются на обработку после щелчка на кнопке формы отправки данных (SUBMIT). Информация из полей формы передается затем на страницу, определенную атрибутом ACTION элемента FORM. Когда запрос этого URL и соответствующие данные формы приходят на сервер, вызывается указанная страница и данные передаются ей для обработки.

При использовании для обработки данных языка программирования PHP все данные из полей формы помещают в глобальный массив: \$_GET, \$_POST и \$_REQUEST. \$_GET[] и \$_POST[] – ассоциативные массивы, которые содержат все значения, передаваемые в сценарий с помощью метода формы GET или POST соответственно. \$_REQUEST[] – ассоциативный массив, который содержит все значения, передаваемые в сценарий методов POST и GET. Ключами этих массивов являются имена переменных, которые указаны в атрибуте NAME элементов формы, а значениями – введенные или выбранные в элементах формы данные.

Скрипты на PHP легко встраиваются в HTML-код. Для обеспечения необходимой гибкости при построении динамических web-приложений можно внедрить в страницу несколько сценариев PHP. При внедрении нескольких сценариев переменные, значения которых были присвоены в одном сценарии, могут использоваться в другом сценарии той же страницы. Кроме того, код HTML может интегрироваться прямо в команды PHP.

Существуют четыре варианта оформления перехода в PHP:

- стандартные теги: (`<?php print "Welcome to the world of PHP!"; ?>`);
- короткие теги (`<? print "Welcome to the world of PHP!"; ?>`);
- теги script;
- теги в стиле ASP (`<%php print "Welcome!"; %>`).

Стандартные теги используются программистами PHP чаще остальных способов, что объясняется наглядностью и удобством этой формы записи.

Пример 3.4

Файл reader.php, в котором данные, введенные в форму, содержащуюся в файле form1.html, выводятся на экран (в окно браузера), используя суперглобальный массив \$_POST.

```
<HTML>
<HEAD>
  <TITLE> Обработка данных формы </TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H1>Вы ввели в форму следующие значения: </H1>
    Ваше имя <?php echo $_POST["Name"]; ?>
  </CENTER>
</BODY>
</HTML>
```

Задание 3.4. Создать файл `reader.php` для вывода на экран в табличной форме пользовательских данных, введенных в форму, которая хранится в файле `form1.html`.

3.1.5 Флажок

Флажки используются в ситуациях, когда пользователь выбирает один или несколько вариантов из готового набора. Синтаксис определения флажка:

```
<INPUT TYPE="CHECKBOX" NAME="имя_переменной"  
      VALUE="начальное_значение" [CHECKED]>
```

Определение флажка включает четыре атрибута.

TYPE – тип элемента (для флажков – CHECKBOX).

NAME – имя переменной, в которой сохраняются введенные данные (в данном случае – состояние элемента).

VALUE – значение, присваиваемое переменной по умолчанию. Если флажок установлен, именно это значение будет присвоено переменной с указанным именем. Если флажок не установлен, значение атрибута VALUE не используется.

CHECKED – если указан этот атрибут, то флажок по умолчанию включен.

Пример 3.5

Создание в форме трех флажков (файл `form.html`).

```
...  
<B>Какие товары вы покупаете в интернет-магазине:</B><BR>  
<INPUT TYPE="CHECKBOX" NAME="book"  
VALUE="Книги">Книги<BR>  
<INPUT TYPE="CHECKBOX" NAME="CD" VALUE="CD-диски">CD-  
диски<BR>  
<INPUT TYPE="CHECKBOX" NAME="DVD" VALUE="DVD-  
диски">DVD-диски<BR>  
...
```

Если флажок неактивен, то параметры вообще не будут переданы на сервер и соответствующие глобальные переменные не будут созданы. Следовательно, при попытке обращения к такой переменной будет выведено сообщение о том, что переменная не существует. Поэтому необходимо предварительно проверить, существует переменная или нет, для этого используется функция `isset()`.

Пример 3.6

Вывод на экран значений выбранных флажков (файл `reader.php`).

```
...  
<P>В интернет-магазине вы покупаете: </P>  
<?  
if (isset($_POST["book"])) echo $_POST["book"]."<BR>";
```

```

if (isset($_POST["CD"])) echo $_POST["CD"]."<BR>";
if (isset($_POST["DVD"])) echo $_POST["DVD"]."<BR>";
?>
...

```

Задание 3.5. Добавить в форму флажки для выбора пользователем иностранных языков, которыми он свободно владеет. Вывести на экран выбранные значения вместе с остальными данными.

3.1.6 Переключатель

Переключатель представляет собой разновидность флажка; он работает практически так же, за одним исключением – в любой момент времени в группе может быть установлен лишь один переключатель. Синтаксис определения переключателя:

```

<INPUT TYPE="RADIO" NAME="имя_переменной"
VALUE="начальное_значение" [CHECKED]>

```

Как видите, синтаксис почти не отличается от определения флажка.

Определение переключателя поля включает четыре атрибута.

TYPE – тип элемента (для переключателей – RADIO).

NAME – имя переменной, в которой сохраняются введенные данные (в данном случае – состояние элемента).

VALUE – значение, присваиваемое переменной по умолчанию. Если флажок установлен, именно это значение будет присвоено переменной с указанным именем. Если переключатель не включен, значение атрибута VALUE не используется.

CHECKED – если указан этот атрибут, то переключатель по умолчанию активен.

Пример 3.7

Создание в форме переключателя (файл form.html).

```

...
<B>Ваш пол:</B><BR>
<INPUT TYPE="RADIO" NAME="sex" VALUE="Мужчина"
CHECKED >Мужчина<BR>
<INPUT TYPE="RADIO" NAME="sex"
VALUE="Женщина">Женщина<BR>
...

```

Пример 3.8

Вывод на экран значения переключателя (файл reader.php).

```

...
echo "Ваш пол: ".$_POST["sex"]."<BR>";
...

```

Задание 3.6. Добавить в форму два переключателя – для выбора пользователем его пола и уровня образования (среднее, среднее специальное, высшее и т. д.). Вывести на экран выбранные значения вместе с остальными данными.

3.1.7 Раскрывающийся список

Раскрывающиеся списки позволяют выбрать один вариант из множества. Как правило, раскрывающиеся списки применяются при работе с относительно большими наборами данных. Синтаксис определения раскрывающегося списка:

```
<SELECT NAME="имя_переменной">  
  <OPTION VALUE="значение_переменной1">Вариант 1</OPTION>  
  <OPTION VALUE="значение_переменной2">Вариант 2</OPTION>  
  <OPTION VALUE="значение_переменной3">Вариант 3</OPTION>  
  <OPTION VALUE="значение_переменной4">Вариант 4</OPTION>  
</SELECT>
```

Определение раскрывающегося списка поля включает два атрибута.

NAME – имя переменной, в которой сохраняются введенные данные (в данном случае – строка, выбранная в списке).

VALUE – значение переменной, которое соответствует выбранной строке.

Пример 3.9

Создание в форме раскрывающегося списка (файл form.html).

```
...  
<B>Какая пора года вам нравится больше всего?</B><BR>  
<SELECT NAME="pora_goda">  
  <OPTION VALUE="Зима"> Зима </OPTION>  
  <OPTION VALUE="Лето"> Лето </OPTION>  
  <OPTION VALUE="Осень"> Осень </OPTION>  
  <OPTION VALUE="Весна"> Весна </OPTION>  
</SELECT>
```

Пример 3.10

Вывод на экран значения, выбранного в раскрывающемся списке (файл reader.php).

```
...  
echo "Ваша любимая пора года: ".$_POST["pora_goda"]."<BR>";  
...
```

Задание 3.7. Добавить в форму два раскрывающихся списка – для выбора города проживания и профессии. Вывести на экран выбранные значения вместе с остальными данными.

3.1.8 Текстовая область

Текстовая область используется для ввода нескольких больших объемов текста, не ограничивающихся простым именем или адресом электронной почты. Синтаксис определения текстовой области:

```
<TEXTAREA NAME="имя_переменной" ROWS="N" COLS="N"
VALUE=""></TEXTAREA>
```

Определение текстового поля включает четыре атрибута.

NAME – имя переменной, в которой сохраняются введенные данные.

ROWS – количество строк в текстовой области.

COLS – количество столбцов в текстовой области.

VALUE – значение, присваиваемое переменной по умолчанию.

Пример 3.11

Создание в форме текстовой области (файл form.html).

...

```
<B>Дополнительная информация:</B><BR>
```

```
<TEXTAREA NAME="comments" ROWS="5" COLS="5" VALUE="Здесь
вы можете написать свои комментарии"></TEXTAREA>
```

...

Пример 3.12

Вывод на экран значения текстового поля (файл reader.php).

...

```
echo " Дополнительная информация: " .$_POST["comments"]."<BR>";
```

...

Задание 3.8. Добавить в форму текстовую область для ввода вопроса пользователя. Вывести на экран введенный вопрос вместе с остальными данными.

3.1.9 Ввод и обработка данных формы в одном сценарии

При вводе и обработке данных в одном сценарии атрибут ACTION формы должен ссылаться на ту же страницу, в которой определяется сама форма. С помощью условной конструкции if необходимо проверить состояние переменной окружения REQUEST_METHOD, которая хранится в суперглобальном массиве \$_SERVER. Переменная окружения REQUEST_METHOD получает свое значение, равное методу передачи информации GET или POST, только после нажатия кнопки отправки данных.

Пример 3.13

Ввод и обработка данных в одном скрипте, который сохранен в файле form.php.

```
<HTML>
```

```
<HEAD>
```

```

<TITLE> Пример HTML-формы </TITLE>
</HEAD>
<BODY>
<?
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    echo "<h1>Привет, <b>" . $_POST["Name"] . "!</b></h1>";
}
else {
?>
<H1> Пример HTML-формы </H1>
<FORM METHOD= "POST" ACTION= "form.php" >
<TABLE>
<TR><TD>Введите ваше имя:</TD>
    <TD><INPUT NAME = "Name" TYPE="TEXT" ></TD>
</TR>
<TR>
    <TD COLSPAN=2 >
        <INPUT TYPE="SUBMIT" VALUE="Отправить">
    </TD>
</TR>
</TABLE>
</FORM>
<? }?>
</BODY>
</HTML>

```

3.2 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

1 Создать один скрипт, который бы при первом запуске выводил на экран форму для ввода пользовательских данных (форма должна содержать все возможные элементы управления), а после нажатия кнопки обработки данных выводил бы на экран введенные пользователем данные.

2 Добавить в форму поле для ввода подтверждения пароля. В обработчике проверить правильность ввода пароля (в поле с паролем и в поле с подтверждением пароля должен быть введен один и тот же пароль).

Пример проверки:

```

<? if ($_POST["pass1"]!= $_POST["pass2"])
    echo "Введенные пароли отличаются";
?>

```

3 Добавить в форму поля ФИО, e-mail, пароль пользователя и введенный пользователем вопрос – обязательные поля. Проверить заполнение этих полей, в случае если пользователь не заполнил эти поля, выдать сообщение об ошибке и повторно вывести на экран форму.

Пример проверки:

```
<? if ($_POST["Name"]=="")  
    echo "Не введена фамилия";  
?>
```

4 Обязательные поля в форме отметить красной звездочкой и написать внизу формы примечание, что эти поля должны быть заполнены обязательно.

3.3 КОНТРОЛЬНЫЕ ВОПРОСЫ

1 Какие элементы используются для задания элементов в форме?

- <INPUT ...>;
- <SELECT ...>;
- ;
- <TEXTAREA ...>;
- <SUBMIT ... >.

2 Какие теги используются для создания текстовых полей ввода в форме?

- <INPUT TYPE="text">;
- <TEXTAREA>;
- <SELECT>;
- <INPUT TYPE="password">;
- <OPTION>.

3 Какие методы можно применять для отправки данных формы?

- метод GET;
- метод HEAD;
- метод POST;
- метод MAILTO;
- нет верного ответа.

4 Если адрес в браузере имеет вид `http://ex.com/script.cgi?time=+5`, каким методом передавались данные, введенные в форму?

- метод GET;
- метод HEAD;
- метод POST;
- метод MAILTO;
- нет верного ответа.

5 При каком методе передачи данных формы данные никак не интерпретируются сервером, а пересылаются непосредственно сценарию?

- метод GET;
- метод HEAD;
- метод POST;
- метод MAILTO;
- нет верного ответа.

6 Способ отправки параметров сценарию, когда данные помещаются в командную строку URL, называется:

- метод GET;
- метод HEAD;
- метод POST;
- метод MAILTO;
- нет верного ответа.

7 С помощью какого атрибута элемента FORM указывается адрес, по которому нужно отправлять данные формы?

- ACTION;
- HREF;
- LOCATION;
- METHOD;
- TARGET.

8 С помощью какого атрибута элемента FORM указывается метод, которым будут передаваться данные формы?

- TARGET;
- METHOD;
- LOCATION;
- HREF;
- ACTION.

9 Сколько форм может содержать одна HTML-страница?

- только одну;
- две;
- HTML-страница не может содержать форму;
- три;
- любое количество.

10 Выберите правильные варианты кода для создания скрытого текстового поля:

- `<input type="text" name="text" size="5" maxlength="10" value="">`;
- `<input type="password" name="name">`;
- `<input type="hidden" name="hidden" value="y">`;
- `<input type="hidden" name="name" value="y">`;
- `<textarea name="text" rows="5" cols="5" value=""></textarea>`.

11 Выберите правильные варианты кода для создания текстового поля для ввода пароля:

- `<input type="text" name="password" size="5" maxlength="10" value="">`;
- `<input type="password" name="name">`;
- `<input type="password" name="hidden" value="password">`;
- `<input type="password" name="password" value="y">`;
- `<textarea name="text" rows="5" cols="5" value=""></textarea>`.

12 Выберите правильные варианты кода для создания текстового поля:

- `<input type="text" name="name" size="5" maxlength="10">`;
- `<input type="text" name="password">`;
- `<input type="text" name="hidden" value="y">`;
- `<input type="submit" name="text">`;
- `<input type="checkbox" name="text">`.

13 Выберите правильные варианты кода для создания текстовой области:

- `<textarea name="text" rows="5" cols="5" value="">`;
- `<textarea name="name" rows="5" cols="5" value=""></textarea>`;
- `<textarea name="text" rows="5" cols="5" value="text"></textarea>`;
- `<input type="textarea" name="name" rows="5" cols="5" value="">`;
- `<textarea name="name" rows="5" cols="5" value="">`.

14 Выберите правильные варианты кода для создания переключателей:

- `<input type="checkbox" name="name" value="Ваш выбор">`;
- `<input type="checkbox" name="radio" value="Ваш выбор">`;
- `<input type="radio" name="checkbox" value="Ваш выбор">`;
- `<input type="radio" name="name" value="Ваш выбор">`;
- `<radio name="name" value="Ваш выбор"></radio>`.

15 Выберите правильные варианты кода для создания флажка:

- `<input type="checkbox" name="name" value="Ваш выбор">`;
- `<input type="checkbox" name="radio" value="Ваш выбор">`;
- `<input type="radio" name="checkbox" value="Ваш выбор">`;
- `<input type="radio" name="name" value="Ваш выбор">`;
- `<checkbox name="name" value="Ваш выбор"></checkbox>`.

16 Выберите правильные варианты кода для создания раскрывающегося списка:

- `<select name="spisok"><option value="Ваш выбор1"><option value="Ваш выбор2"></select>`;
- `<select name="spisok" option value="Ваш выбор"></select>`;
- `<input type="select" name="name" value="Ваш выбор">`;
- `<select name="spisok" option value="Ваш выбор">`;
- `<select name="spisok">Ваш выбор</select>`.

17 Выберите правильные варианты кода для создания кнопки:

- `<input type="submit" value="Кнопка">`;
- `<input type="reset" value="Кнопка">`;
- `<input type="reset">`;
- `<input type="Submit">`;
- `<submit name="Кнопка"></submit>`.