

**5317**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# **ОСНОВЫ ПОСТРОЕНИЯ HTML-ДОКУМЕНТОВ**

**Часть 4**

**Методические указания к лабораторной работе**



**Рязань 2018**

УДК 004.4

Основы построение HTML-документов. Часть 4: методические указания к лабораторной работе / Рязан. гос. радиотехн. ун-т; сост.: А.П. Шибанов, А.Н. Сапрыкин, А.М. Гостин. Рязань, 2018. 16 с.

Содержат описание лабораторной работы, используемой в курсах "Системные основы промышленных сетей", "Распределенные информационные системы", "Веб-программирование", "Интернет-технологии".

Предназначены для обучающихся очной, очно-заочной и заочной форм обучения направлений "Информатика и вычислительная техника" и "Конструирование и технология электронных средств".

Ил. 13. Библиогр.: 5 назв.

*Язык HTML, HTML-теги, веб-страница, гипертекст*

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра САПР вычислительных средств Рязанского государственного радиотехнического университета (зав. кафедрой В.П. Корячко)

Основы построения HTML-документов. Часть 4

Составители: Ш и б а н о в Александр Петрович  
С а п р ы к и н Алексей Николаевич  
Г о с т и н Алексей Михайлович

Редактор М.Е. Цветкова

Корректор С.В. Макушина

Подписано в печать 20.10.18. Формат бумаги 60х84 1/16.

Бумага писчая. Печать трафаретная. Усл. печ. л. 1,0.

Тираж 50 экз. Заказ

Рязанский государственный радиотехнический университет.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.

## **Лабораторная работа № 6**

### **Формы в HTML**

Цель работы: получение практических навыков создания форм в HTML-документах. Изучение тегов для создания форм в HTML-документах и их основных атрибутов.

#### **1. Введение**

Данная работа посвящена формированию практических навыков по созданию и использованию форм в HTML-документах.

Форма (англ. form) в HTML – это раздел HTML-документа, позволяющий пользователю вводить информацию для последующей обработки веб-приложением. Получение данных от пользователя осуществляется через типовые элементы управления: поля ввода, скрытые поля, поля для отправки файлов, области редактирования, выпадающие списки, переключатели, флажки и кнопки и т.п.

#### **2. Теоретическая часть**

##### **2.1. Создание простейшей формы**

Для взаимодействия клиента с веб-сервером в HTML используются клиентские формы. Форма, имеющая поля ввода и другие элементы управления, заполняется клиентом с помощью браузера, проверяется обработчиком JavaScript на стороне клиента и затем отправляется на веб-сервер, где управление передается отдельному серверному приложению, написанному на одном из серверных скриптовых языков, например PHP. Взаимодействие клиента с сервером осуществляется с помощью сетевого протокола HTTP (*HyperText Transfer Protocol*).

Для создания формы используется контейнерный тег `<form>`. Атрибуты тега `<form>`:

- `action` – указывает URL адрес серверного приложения. Если данные формы обрабатываются и отправляются на сервер сценарием Javascript, то в качестве значения атрибута `action` можно указать пустой URL адрес: `action="#"`;
- `name` – имя формы. Используется для идентификации формы в Javascript;
- `method` – метод HTTP отправки формы на сервер. Атрибут может принимать значения GET или POST. Если атрибут не задан, используется метод GET. В случае использования метода GET данные

формы отправляются на сервер в строке с URL адресом серверного приложения после знака ? (вопрос) в виде пар: имя=значение, разделенных символом & (амперсанд), и передаются управляющей программе. Имена определяются атрибутами name элементов управления, а значения вводятся пользователем или устанавливаются по умолчанию с помощью атрибутов value. Национальный алфавит и специальные символы кодируются в формате %nn, представляющем собой шестнадцатеричный код UTF-8.

Пример:

```
http://htmlbook.ru/handler.php?nick=%ED%FF&page=5
```

В случае использования метода *POST* данные формы также передаются управляющей программе, но не в строке URL, а в содержании HTTP запроса:

```
http://htmlbook.ru/handler.php
nick=%C2%E0%ED%FF&page=5
```

■ *enctype* – способ кодирования данных формы. Атрибут может принимать значения:

- 1) *Application/x-www-form-urlencoded* – для передачи данных, закодированных в UTF-8;
- 2) *Multipart/form-data* – для передачи некодированных бинарных данных (файлов);
- 3) *Text/plain* – для передачи некодированных текстовых данных.

Если атрибут не задан, используется первый способ кодирования *application/x-www-form-urlencoded*.

Пример простейшей HTML-формы:

```
<html>
<head>
  <title>Формы в html</title>
</head>
<body>
  <form action="#" method="post" name="exampleform">
    <table>
      <tr>
        <td>Фамилия:</td>
        <td><input type="text" id="username"></td>
      </tr>
      <tr>
        <td>Имя:</td>
        <td><input type="text" id="nick"></td>
      </tr>
      <tr>
        <td>Адрес эл. почты:</td>
```

```

        <td><input type="text" id="email"></td>
    </tr>
    <tr>
        <td>Пол:</td>
        <td><input type="radio" name="sex" value="male">М<br>
        <input type="radio" name="sex" value="female">Ж</td>
    </tr>
    <tr>
        <td><input type="submit" value="Отправить"></td>
        <td><input type="reset" value="Отменить"></td>
    </tr>
</table>
</form>
</body>
</html>

```

Результат выполнения примера представлен на рис. 1.

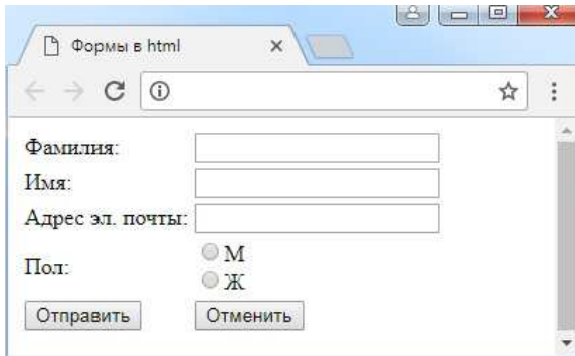


Рис. 1. Пример создания простейшей HTML-формы

## 2.2. Метки

Меткой является вспомогательный контейнерный тег `<label>`. Метки служат для связи дополнительной текстовой информации с элементами формы. Существует два способа связывания элемента формы и метки:

- использование атрибута `id` элемента формы и указание его имени в качестве атрибута `for` элемента `<label>`;
- элемент формы помещается внутрь контейнера `<label>`.

## 2.3. Элементы управления

К элементам управления формы относятся поля ввода, скрытые поля, поля для отправки файлов, области редактирования,

выпадающие списки, переключатели, флажки и кнопки. Каждый элемент управления имеет свой вид отображения в браузере.

Поля ввода, скрытые поля, поля для отправки файлов, переключатели, флажки и кнопки создаются с помощью одиночного тега `<input>`. Тип элемента управления определяется значением атрибута `type`.

Для группировки элементов формы используется контейнерный тег `<fieldset>`. Такая группировка облегчает работу с формами, содержащими большое число данных. Элемент `<fieldset>` отображается браузером в виде рамки. Рамка может включать заголовочную надпись, определяемую содержимым вложенного контейнерного тега `<legend>`. Пример:

```
<fieldset>
  <legend>Владение языками</legend>
  <input type="checkbox">русский
  <input type="checkbox">английский
</fieldset>
```

### 2.3.1. Текстовое поле ввода

Представляет собой контейнер для ввода различной текстовой информации (однострочное текстовое поле). Пример:

```
<input type="text" name="fio" size="25"
maxlength="50" value="Иванов И.И.">
```

Результат выполнения примера представлен на рис. 2.

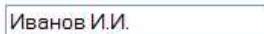


Рис. 2. Текстовое поле ввода

Некоторые используемые атрибуты:

- `name` – имя элемента;
- `value` – значение поля по умолчанию;
- `size` – ширина поля в моноширных символах;
- `maxlength` – максимальное количество вводимых символов;
- `autofocus` – устанавливает автофокус (атрибут без содержания);
- `form` – имя формы. Используется для привязки отдельного элемента к заданной форме;
- `disabled` – блокирует доступ и изменение элемента (атрибут без содержания);
- `readonly` – блокирует изменение элемента (атрибут без содержания);

- `placeholder` – задает текст подсказки, выводимой в поле;
- `required` – проверяет заполненность вводимого поля (атрибут без содержания). В случае если поле не заполнено, в браузер выводится предупреждающее сообщение;
- `pattern` – устанавливает регулярное выражение в соответствии с ECMA-262 (*ECMAScript 2015 Language Specification*) для фильтрации вводимого текста. В браузере вводится только текст, разрешенный шаблоном.

Примеры регулярных выражений: `\d` – одна цифра; `\w` – одна латинская буква; `\s` – один пробел; `[A-Z]` – один символ из диапазона; `\D` – не цифра; `\S` – не пробел; `^` – начало строки; `$` – конец строки; `.` – один любой символ; `.*` – любое количество любых символов; `^\S+` – первое слово; `\S+$` – последнее слово; `\d{3}` – три цифры; `\d{3,6}` – от трех до шести цифр; `\.` – десятичная точка; `[^0]` – не ноль; `(one|two)` – фраза `one` или `two` и т.д.

### 2.3.2. Поле ввода пароля

Предназначение: маскировка вводимых символов браузером.

Пример:

```
<input name="passw" type="password">
```

Результат выполнения примера представлен на рис. 3.



Рис. 3. Поле ввода пароля

Обратите внимание, что, несмотря на маскировку символов браузером, при отправке данных на сервер значения полей ввода паролей передаются по протоколу HTTP в незашифрованном виде. Для передачи конфиденциальной информации используйте шифрование трафика SSL (*Secure Socket Layer*) и протокол HTTPS.

Виды и значения остальных атрибутов совпадают с текстовым полем ввода.

### 2.3.3. Скрытое поле ввода

Предназначение: передача не отображаемых браузером данных на сервер. Может включать атрибут `value`, содержащий предустановленное значение.

Пример:

```
<input name="userid" type="hidden" value="135">
```

### 2.3.4. Текстовая кнопка

Предназначение: связывает событие `onclick` элемента с обработчиком JavaScript. Используется для предварительной обработки данных формы перед отправкой их на сервер.

Пример:

```
<input name="btn" type="button" value="Ввод">
```

Результат выполнения примера представлен на рис. 4.



Рис. 4. Текстовая кнопка

Некоторые используемые атрибуты:

- `name` – имя элемента;
- `value` – текст кнопки, отображаемый браузером;
- `onclick` – функция обработчика события Javascript.

### 2.3.5. Кнопка отправки данных

Предназначение: отправка данных формы на сервер.

Пример:

```
<input name="btn" type="submit">
```

Результат выполнения примера представлен на рис. 5.

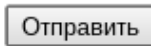


Рис. 5. Кнопка отправки данных

Атрибуты те же, что и у обычной кнопки.

### 2.3.6. Кнопка сброса

Предназначение: возвращение данных формы в первоначальное состояние.

Пример:

```
<input name="btn" type="reset">
```

Результат выполнения примера представлен на рис. 6.

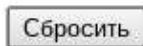


Рис. 6. Кнопка сброса

Атрибуты те же, что и у обычной кнопки.



### 2.3.7. Флажки

Флажки предлагают пользователю ряд вариантов и разрешают выбор одного или нескольких из них.

Если при отправке формы флажок не установлен, данные элемента не передаются. Пример использования:

```
<label>Выберите теплые цвета:
  <input name="opt1" type="checkbox">красный
  <input name="opt2" type="checkbox">желтый
</label>
```

Результат выполнения примера представлен на рис. 7.

Выберите теплые цвета:  
☒ красный ☒ желтый

Рис. 7. Флажок выбора опций

Некоторые используемые атрибуты:

- name – имя элемента;
- value – значение элемента. Если этот атрибут отсутствует, то при выборе опции элемент принимает значение on;
- checked – делает флажок установленным по умолчанию;
- disabled – блокирует доступ к элементу (атрибут без содержания). Независимо от состояния элемента данные не передаются;
- required – проверяет наличие установленного флажка (атрибут без содержания).

### 2.3.8. Переключатели

Предназначение: выбор одного из нескольких предложенных элементов из группы. Все элементы переключателя должны иметь одинаковое имя name. Если при отправке формы переключатель не установлен, его данные не передаются.

Пример использования:

```
<label>Выберите верный ответ:
  <input name="opt" type="radio" value="1">да
  <input name="opt" type="radio" value="0">нет
</label>
```

Результат выполнения примера представлен на рис. 8.

Выберите верный ответ:

☒ да ☐ нет

Рис. 8. Переключатель

Атрибуты те же, что и у флажка выбора опций. Атрибут `checked` может быть только у одного элемента из группы.

В случае, если форма содержит несколько независимых переключателей, элементы в них группируются по одноименному атрибуту `name`.

### 2.3.9. Поле для отправки файлов

Предназначение: прикрепление содержания файла к отправляемой форме. Форма должна отправляться на сервер с помощью метода `POST`, способ кодирования данных должен быть установлен как `multipart/form-data`. При нажатии на кнопку "Обзор" в браузере открывается диалог выбора файла. После выбора файла его имя появляется в поле ввода, а содержимое прикрепляется к форме и подготавливается для отправки на сервер.

Пример использования:

```
<input name="attach" type="file">
```

Результат выполнения примера представлен на рис. 9.

Выберите файл payment\_1...4669.pdf

Рис. 9. Поле отправки файла

Дополнительные атрибуты:

- `accept` – устанавливает фильтр на MIME-типы прикрепленных файлов. При нескольких значениях MIME-типы перечисляются через запятую. Возможно обобщение типов, например `image/*` – для всех графических файлов;
- `multiple` – позволяет указывать одновременно несколько файлов в поле (атрибут без содержания);
- `maxlength`, `size` и `value` – аналогичные текстовому полю ввода.

### 2.3.10. Поле электронной почты

Предназначение: указание адреса электронной почты. Адрес проверяется на корректность ввода.

Пример:

```
<input name="email" type="email">
```

### 2.3.11. Поле ввода чисел

Предназначение: ввод целых чисел. Содержит кнопки-стрелки “вверх-вниз”, изменяющие введенное значение на один шаг.

Пример:

```
<input name="num" type="number">
```

Дополнительные атрибуты:

- value – значение по умолчанию;
- min – минимальное значение;
- max – максимальное значение;
- step – шаг приращения (по умолчанию 1).

### 2.3.12. Поле выбора чисел из диапазона

Предназначение: выбор целых чисел из диапазона. Отображает ползунок, изменяющий значение на один шаг.

Пример:

```
<input name="num" type="range">
```

Дополнительные атрибуты:

- value – значение по умолчанию;
- min – минимальное значение (по умолчанию 0);
- max – максимальное значение (по умолчанию 100);
- step – шаг приращения (по умолчанию 1).

### 2.3.13. Область редактирования

Предназначение: ввод многострочного текстового содержимого, предоставление возможности онлайн редактирования. Область образуется с помощью контейнерного тега <textarea>.

Пример использования:

```
<textarea name="content">
  Область редактирования
</textarea>
```

Результат выполнения примера представлен на рис. 10.

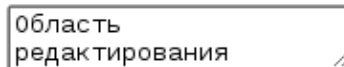


Рис. 10. Область редактирования

Дополнительные атрибуты:

- `cols` – ширина текстовой области в символах моноширного текста;
- `rows` – высота текстовой области в строках, отображаемых без прокрутки содержимого;
- `wrap` – режим переноса на следующую строку. Может принимать одно из трех значений: `physical`, `virtual` и `off`;
- `disabled`, `readonly`, `placeholder` и `required` – аналогичные текстовому полю ввода.

#### 2.3.14. Выпадающий список

Предназначение: выбор одного или нескольких элементов из заданного списка. Список задается контейнерным тегом `<select>`, включающим контейнерные элементы `<option>` с текстовым содержимым. При выборе элемента из списка его значение `value` устанавливается для контейнера `<select>` и передается на сервер.

Пример использования:

```
<select name="lang">
  <option value="RU">русский</option>
  <option value="EN">английский</option>
</select>
```

Результат выполнения примера представлен на рис. 11.

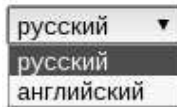


Рис. 11. Выпадающий список

Дополнительные атрибуты тега `<select>`:

- `multiple` – возможность множественного выбора элементов списка (атрибут без содержания). Множественный выбор элементов осуществляется с участием нажатой клавиши `<Ctrl>`;
- `size` – определяет высоту списка в элементах (по умолчанию, 1);
- `disabled` – блокирует выбор элементов (атрибут без содержания);
- `required` – проверяет, выбран ли какой-либо элемент из списка (атрибут без содержания). В случае, если элемент не выбран, в браузер выводится предупреждающее сообщение;
- `value` – значение элемента списка. Атрибут относится к тегу `<option>`. Если атрибут не установлен, то значение выбранного элемента определяется его содержимым;

▪ `selected` – выделение элемента списка по умолчанию (атрибут без содержания). Атрибут относится к одному или нескольким (в случае множественного выбора) тегам `<option>`.

Элементы списка могут быть сгруппированы в иерархию с помощью специального контейнерного тега `<optgroup>`. Атрибут `label` используется для обозначения заголовка группы списка. Заголовок группы отображается выделенным текстом в браузере и является невыбираемым элементом.

Пример использования:

```
<select name="lang">
  <optgroup label="Языки ввода">
    <option value="RU">русский</option>
    <option value="EN">английский</option>
  </optgroup>
</select>
```

Результат выполнения примера представлен на рис. 12.

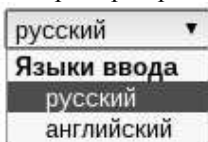


Рис. 12. Выпадающий список с группировкой элементов

### 2.3.15. Комбинированный список

Предназначение: выбор при наборе в текстовом поле. Список задается контейнерным тегом `<datalist>`, включающим контейнерные элементы `<option>` с текстовым содержимым. С помощью атрибута `id` тега `<datalist>` список связывается с текстовым полем `<input>`, имеющим соответствующий атрибут `list`. Список становится доступным при получении полем фокуса или при наборе текста.

Пример:

```
<input name="lang" list="lang">
<datalist id="lang">
  <option value="RU">русский</option>
  <option value="EN">английский</option>
</datalist>
```

### 3. Практическая часть

#### 3.1. Самостоятельное задание

Перед тем, как приступить к выполнению самостоятельного задания, внимательно изучите теоретическую часть методических указаний.

1. Создайте новую веб-страницу, содержащую веб-форму, представленную на рис. 13. Заголовок страницы должен включать ФИО студентов, выполнивших работу, и номер их бригады. Элементы формы должны располагаться в таблице, при этом для каждого из них должно быть задано значение по умолчанию. URL адрес сценария для обработки формы: <http://192.168.32.150/study/lab7/index.php>.

<b>Форма регистрации</b>	
ФИО	<input type="text"/>
E-mail	<input type="text"/>
Пароль	<input type="text"/>
Пол	<input checked="" type="radio"/> Мужской <input type="radio"/> Женский
№ бригады	<input type="text" value="Бригада № 1"/>
Наиболее интересные работы	<input checked="" type="checkbox"/> ЛР № 1. Тег <input type="checkbox"/> ЛР № 2. Списки <input type="checkbox"/> ЛР № 3. Ссылки <input type="checkbox"/> ЛР № 4. Графика <input type="checkbox"/> ЛР № 5. Таблицы <input type="checkbox"/> ЛР № 6. Формы
Предложения	<input type="text"/>
	<input type="button" value="Отправить"/> <input type="button" value="Очистить"/>

Рис. 13. Образец веб-формы

Имена элементов управления (name) формы:

- ФИО – fio;
- e-mail – mail;

- пароль — pass;
- пол — sex;
- № бригады — team;
- наиболее интересные работы — lab1..lab6;
- предложения — offer.

Параметры элементов управления веб-формы, представленные ниже, должны быть заданы в соответствии с номером варианта.

### ***Задание по вариантам***

1. Ширина текстового поля ввода в символах: 25. Максимальное количество вводимых в текстовое поле ввода символов: 20. Режим переноса слов в текстовой области: физический. Высота текстовой области в строках: 2.

2. Ширина текстового поля ввода в символах: 20. Максимальное количество вводимых в текстовое поле ввода символов: 25. Режим переноса слов в текстовой области: выключенный. Высота текстовой области в строках: 3.

3. Ширина текстового поля ввода в символах: 35. Максимальное количество вводимых в текстовое поле ввода символов: 30. Режим переноса слов в текстовой области: физический. Высота текстовой области в строках: 4.

4. Ширина текстового поля ввода в символах: 40. Максимальное количество вводимых в текстовое поле ввода символов: 350. Режим переноса слов в текстовой области: выключенный. Высота текстовой области в строках: 2.

5. Ширина текстового поля ввода в символах: 25. Максимальное количество вводимых в текстовое поле ввода символов: 25. Режим переноса слов в текстовой области: физический. Высота текстовой области в строках: 3.

6. Ширина текстового поля ввода в символах: 35. Максимальное количество вводимых в текстовое поле ввода символов: 30. Режим переноса слов в текстовой области: выключенный. Высота текстовой области в строках: 4.

7. Ширина текстового поля ввода в символах: 40. Максимальное количество вводимых в текстовое поле ввода символов: 35. Режим переноса слов в текстовой области: физический. Высота текстовой области в строках: 2.

8. Ширина текстового поля ввода в символах: 30. Максимальное количество вводимых в текстовое поле ввода символов: 25. Режим переноса слов в текстовой области: выключенный. Высота текстовой области в строках: 3.

9. Ширина текстового поля ввода в символах: 25. Максимальное количество вводимых в текстовое поле ввода символов: 30. Режим переноса слов в текстовой области: физический. Высота текстовой области в строках: 4.

10. Ширина текстового поля ввода в символах: 25. Максимальное количество вводимых в текстовое поле ввода символов: 20. Режим переноса слов в текстовой области: выключенный. Высота текстовой области в строках: 2.

2. Создайте новую веб-страницу, содержащую форму в соответствии с заданным вариантом. Заголовок страницы должен включать ФИО студентов, выполнивших работу, и номер их бригады. Форма должна содержать все рассмотренные элементы управления, которые должны располагаться в таблице, при этом для каждого из них должно быть задано значение по умолчанию.

Форма регистрации на сайте

Форма заказа товара

Форма анкеты трудоустройства

Форма обратной связи

Форма анкеты для опроса

### ***Задание по вариантам***

1. Форма регистрации на сайте электронного магазина.
2. Форма анкеты трудоустройства.
3. Форма заказа товара в интернет-магазине.
4. Форма обратной связи на сайте.
5. Форма анкеты для опроса.
6. Форма регистрации на сайте электронного магазина.
7. Форма анкеты трудоустройства.
8. Форма заказа товара в интернет-магазине.
9. Форма обратной связи на сайте.
10. Форма анкеты для опроса.

### **Контрольные вопросы**

1. Что представляет собой форма в HTML?
2. Перечислите основные элементы управления формы.
3. Назовите основные атрибуты тега `<input>`?
4. Какой тег отвечает за создание многострочного текстового поля?
5. Какой атрибут тега `<select>` определяет отправляемое на сервер значение при выборе?



**Библиографический список**

1. Дронов В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011. – 416 с.
2. Дунаев В.В. HTML, скрипты и стили. – 3-е изд. – СПб.: БХВ-Петербург, 2011. – 816 с.
3. Прохоренок Н.А., Дронов В.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. – СПб.: БХВ-Петербург, 2010. – 912 с.
4. Хоган Б. HTML 5 и CSS 3. Веб-разработка по стандартам нового поколения. – СПб.: Питер, 2012. – 270 с.
5. Справочник по HTML и CSS. [Электронный ресурс]. URL: [http:// http://htmlbook.ru/](http://htmlbook.ru/) (дата обращения: 01.03.2018).

## Оглавление

Лабораторная работа № 6. Формы в HTML .....	1
1. Введение .....	1
2. Теоретическая часть .....	1
2.1. Создание простейшей формы .....	1
2.2. Метки.....	3
2.3. Элементы управления .....	3
2.3.1. Текстовое поле ввода.....	4
2.3.2. Поле ввода пароля .....	5
2.3.3. Скрытое поле ввода .....	5
2.3.4. Текстовая кнопка .....	6
2.3.5. Кнопка отправки данных.....	6
2.3.6. Кнопка сброса .....	6
2.3.7. Флажки.....	7
2.3.8. Переключатели .....	7
2.3.9. Поле для отправки файлов.....	8
2.3.10. Поле электронной почты .....	8
2.3.11. Поле ввода чисел .....	9
2.3.12. Поле выбора чисел из диапазона .....	9
2.3.13. Область редактирования .....	9
2.3.14. Выпадающий список.....	10
2.3.15. Комбинированный список .....	11
3. Практическая часть.....	12
3.1. Самостоятельное задание.....	12
Контрольные вопросы .....	14
Библиографический список.....	15