

# ФОРМЫ

# Назначение форм

---

Формы предназначены для пересылки данных от удаленного пользователя к Web-серверу.

HTML-документ с формой отличается от обычного HTML-документа наличием элементов управления: полей ввода текста, флажков, кнопок и т.д.

Данные, введенные в форму, обрабатываются на стороне сервера CGI-приложениями и на стороне клиента сценариями.

# Создание формы

Тэг `<form> ... </form>`

Атрибуты:

- Action – задает значение URL-адреса приложения, которое будет обрабатывать данные
- Method – задает способ отправки данных, введенных в форму. Может принимать значения `get` (по умолчанию) или `post`.

# Методы передачи данных

Взаимодействие между клиентом-браузером и Web-сервером осуществляется по правилам, заданным протоколом HTTP, и состоит из запросов клиента и ответов сервера. Запрос разбивается на три части. Запрос клиента состоит из:

- метода, URL-адреса запрашиваемого файла и номера версии протокола HTTP
- заголовка запроса
- тела запроса (данные, посылаемые серверу)

# Метод GET

При использовании метода GET данные формы пересылаются в составе URL-запроса, к которому присоединяются после символа "?" в виде совокупности пар *переменная = значение*, разделенных символом "&". В этом случае первая строка запроса может иметь следующий вид:

```
GET / cgi-bin / cgi-program.pl ? name = Саша &  
surname = Иванов HTTP / 1.1
```

Метод GET используется для отправки небольших объемов данных.

# Метод POST

При использовании метода POST данные формы пересылаются Web-серверу в теле запроса, после чего передаются сервером в CGI-программу через стандартный ввод.

Если используется метод POST, необходимо задать значение атрибута ENCTYPE. Значением параметра ENCTYPE является *media-type*, определяющий формат кодирования данных при передаче их от браузера к серверу. Возможны два значения этого параметра: *application/x-www-form-urlencoded* (по умолчанию) и *multipart/form-data*.



Общий вид задания формы:

`<form action = URL-адрес`

`method = метод передачи`

`данных`

`enctype = медиа-тип >`

`содержание формы`

`</form>`

# Элементы управления

Тег `<INPUT>` предназначен для создания элементов управления и всегда используется совместно с атрибутом `TYPE`.

## Атрибуты:

- `Type` - задает тип интерфейсного элемента
- `Name` – задает имя элемента управления
- `Value` – задает начальное значение для полей ввода текста и полей для ввода имен файлов, надпись на кнопке
- `Checked` – задает начальное состояние для флажков и переключателей



- Size – задает размер элемента управления
- Maxlength – задает максимальную длину текста, который может быть введен в текстовое поле
- Title – задает описание элемента управления, может использоваться браузером в качестве всплывающей подсказки
- Align – задает горизонтальное выравнивание элемента управления

Пример: <form> Текстовое поле

```
<input type=text size=20>  
</form>
```

# Типы элементов управления

Значение атрибута	Описание
text	Однострочное текстовое поле
password	Однострочное текстовое поле. Вводимый текст скрывается символом ●
checkbox	Выключатель (флажок)
radio	Переключатель. Используется только в группе с другими переключателями. У всех переключателей в группе атрибут name должен иметь одинаковое значение
button	Пользовательская кнопка

# Типы элементов управления

Значение атрибута	Описание
submit	Кнопка передачи, нажатие которой вызывает пересылку на сервер всего содержимого формы
file	Поле для ввода имени локального файла, сопровождаемое кнопкой <b>Обзор</b>
hidden	Скрытый элемент управления. Пользователь не видит и не может изменить содержимое этого элемента. Но данные скрытого элемента отправляются на сервер вместе с данными формы

# Типы элементов управления

Значение атрибута	Описание
image	Изображение, при щелчке на которое отправляются данные формы
reset	Кнопка, нажатие которой заменяет значения всех элементов управления значениями по умолчанию

# Стандартные элементы управления

Текстовое поле

Поле для ввода пароля

Флажок

Переключатели

Поле с именем файла

Щелчок по изображению

приведет к отправке данных



# Многострочное текстовое поле

Тэг `<textarea> ... </textarea>` - создаёт область для ввода большого количества строк текста.

## Атрибуты:

- **Name** - указывает имя области, может быть любым, является обязательным. По данному имени приложение обращается к содержимому данного элемента
- **Cols** - количество символов в строке



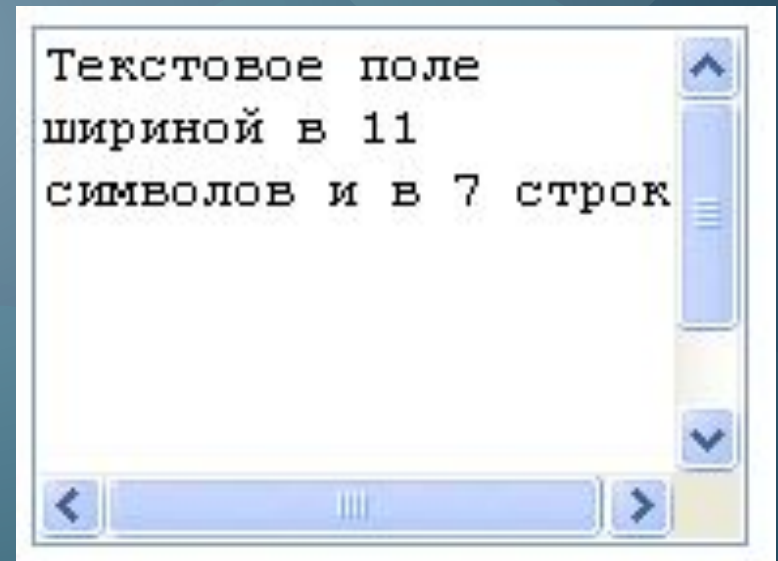
- 
- Rows – задаёт количество строк, которые отображаются без вертикальной полосы прокрутки
  - Disabled - делает элемент недоступным
  - Readonly - текст только для чтения. Нельзя редактировать текст.

# Пример:

```
<textarea cols="11"  
rows="7">
```

Текстовое поле шириной  
в 11 СИМВОЛОВ и в 7 СТРОК

```
</textarea>
```



# СПИСОК

Тег `<select>` создает список или выпадающий список.

## Атрибуты:

- Size - число отображаемых элементов. Если равно 1 или не указано, то выводится одна строка. Если значение больше 1, то выводится список с прокруткой
- Name - задаёт имя списка
- Multiple - даёт возможность выбора нескольких значений из списка

# Создание элементов списка

Тег `<option>` и необязательный парный ему `</option>`

Каждый элемент `select` должен содержать хотя бы один или более элементов `option`, с помощью которого создаются строки списков

## Атрибуты:

- `Selected` - индикатор, указывающий, что данная строка является выбранной по умолчанию. Если данный индикатор отсутствует, то такой строкой является первый элемент
- `Value` - задаёт значение, которое присваивается выбранному элементу списка и отсылается серверу

# Пример:

```
<select size=3>
```

```
<option> Первая строка
```

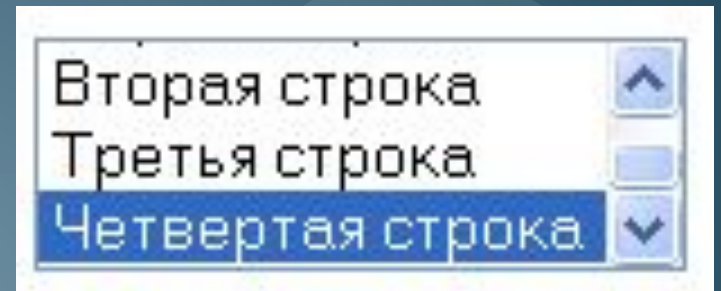
```
<option> Вторая строка
```

```
<option> Третья строка
```

```
<option selected>
```

```
    Четвертая строка
```

```
</select>
```



# Пример:

Выпадающий список

```
<select size=1>
```

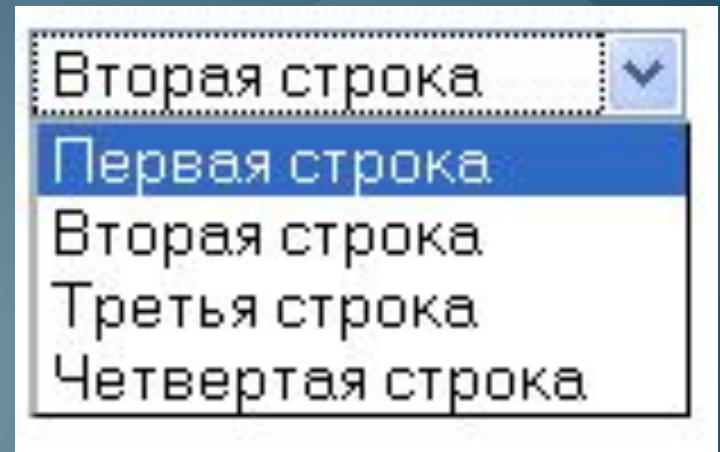
```
<option> Первая строка
```

```
<option> Вторая строка
```

```
<option> Третья строка
```

```
<option> Четвертая строка
```

```
</select>
```



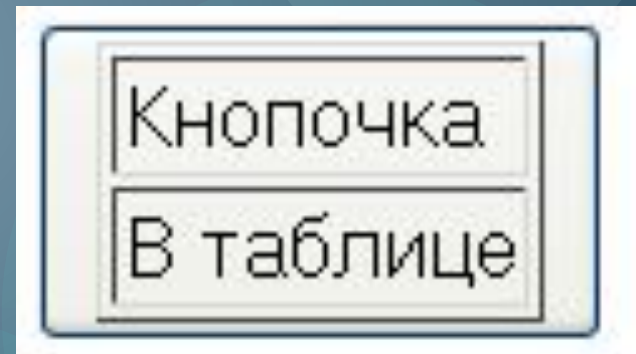


# Пользовательские кнопки

Тэг `<button>...</button>` позволяет создать пользовательскую кнопку.

Пример:

```
<button>  
<table border=1>  
<tr><td> Кнопочка  
</td></tr>  
<tr><td> В таблице  
</td></ tr>  
</table>  
</button>
```



# Группировка элементов

---

Тэг `<fieldset>...</fieldset>` задает рамку вокруг текста и других объектов, содержащихся в контейнере - логически группирует элементы формы.

Тэг `<legend>...</legend>` - задаёт название логических групп элементов.

# Пример:

```
<fieldset>
<legend> Группа кнопок </legend>
<input type="button"
      value="Кнопка 1 ">
<input type="button"
      value="Кнопка 2"> <br>
<input type="button"
      value="Кнопка 3 ">
<input type="button"
      value="Кнопка 4"> <br>
</fieldset>
```

