

HTML – формы

HTML-формы необходимы, когда вы хотите собрать некоторые данные от посетителя сайта. Например, при регистрации пользователя вы хотели бы собрать такую информацию, как имя, адрес электронной почты, кредитная карта и т. Д.

Форма будет получать данные от посетителя сайта, а затем отправлять ее в фоновое приложение, такое как CGI, ASP Script или PHP-скрипт и т. Д. Внутреннее приложение будет выполнять необходимую обработку переданных данных на основе определенной бизнес-логики внутри приложения.

Доступны различные элементы формы, такие как текстовые поля, поля текстовой области, раскрывающиеся меню, переключатели, флажки и т. Д.

HTML- тег **<form>** используется для создания HTML-формы и имеет следующий синтаксис:

```
<form action = "Script URL" method = "GET|POST">  
  form elements like input, textarea etc.  
</form>
```

Атрибуты формы

Помимо общих атрибутов, ниже приведен список наиболее часто используемых атрибутов формы —

Sr.No	Атрибут и описание
1	действие Скрипт готов к обработке переданных данных.
2	метод Метод, который будет использоваться для загрузки данных. Чаще всего используются методы GET и POST.
3	цель Укажите целевое окно или фрейм, где будет отображаться результат скрипта. Он принимает значения, такие как <code>_blank</code> , <code>_self</code> , <code>_parent</code> и т. Д.
4	ENCTYPE Вы можете использовать атрибут <code>enctype</code> , чтобы указать, как браузер кодирует данные перед их отправкой на сервер. Возможные значения — application / x-www-form-urlencoded — это стандартный метод, который большинство форм используют в простых сценариях. mutlipart / form-data — используется, если вы хотите загрузить двоичные данные в виде файлов, таких как изображения, файлы слов и т. Д.

действие

Скрипт готов к обработке переданных данных.

метод

Метод, который будет использоваться для загрузки данных. Чаще всего используются методы GET и POST.

цель

Укажите целевое окно или фрейм, где будет отображаться результат скрипта. Он принимает значения, такие как `_blank`, `_self`, `_parent` и т. Д.

ENCTYPE

Вы можете использовать атрибут `enctype`, чтобы указать, как браузер кодирует данные перед их отправкой на сервер. Возможные значения —

application / x-www-form-urlencoded — это стандартный метод, который большинство форм используют в простых сценариях.

multipart / form-data — используется, если вы хотите загрузить двоичные данные в виде файлов, таких как изображения, файлы слов и т. д.

Элементы управления HTML-формы

Существуют различные типы элементов управления формы, которые вы можете использовать для сбора данных с помощью формы HTML —

- Элементы управления вводом текста
- Флажки управления
- Управление радиоприемником
- Выберите Box Controls
- Коробки выбора файлов
- Скрытые элементы управления
- Кликабельные кнопки
- Кнопка отправки и сброса

Элементы управления вводом текста

В формах используются три типа ввода текста:

- **Элементы управления однострочным вводом текста** — этот элемент управления используется для элементов, для которых требуется только одна строка ввода пользователя, таких как поля поиска или имени. Они создаются с использованием HTML- тега `<input>` .

- **Элементы управления вводом пароля** — это также однострочный ввод текста, но он маскирует символ, как только пользователь вводит его. Они также создаются с использованием HTML тега `<input>`.

- **Многострочные элементы управления вводом текста** — это используется, когда пользователь должен предоставить детали, которые могут быть длиннее, чем одно предложение. Многострочные элементы управления вводом создаются с использованием тега HTML `<textarea>` .

Элементы управления однострочным вводом текста — этот элемент управления используется для элементов, для которых требуется только одна строка ввода пользователя, таких как поля поиска или имени. Они создаются с использованием HTML- тега `<input>` .

Элементы управления вводом пароля — это также однострочный ввод текста, но он маскирует символ, как только пользователь вводит его. Они также создаются с использованием HTML тега `<input>`.

Многострочные элементы управления вводом текста — это используется, когда пользователь должен предоставить детали, которые могут быть длиннее, чем одно предложение. Многострочные элементы управления вводом создаются с использованием тега HTML `<textarea>` .

Элементы управления однострочным вводом текста

Этот элемент управления используется для элементов, для которых требуется только одна строка ввода пользователя, например, поля поиска или имени. Они создаются с использованием HTML-тега `<input>`.

пример

Вот базовый пример однострочного ввода текста, используемого для ввода имени и фамилии —

```
<!DOCTYPE html>
<html>

<head>
```

```

<title>Text Input Control</title>
</head>

<body>
  <form >
    First name: <input type = "text" name = "first_name" />
    <br>
    Last name: <input type = "text" name = "last_name" />
  </form>
</body>

</html>

```

Атрибуты

Ниже приведен список атрибутов для тега `<input>` для создания текстового поля.

Sr.No	Атрибут и описание
1	тип Указывает тип элемента управления вводом, и для элемента управления вводом текста он будет установлен на текст .
2	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
3	значение Это может быть использовано для предоставления начального значения внутри элемента управления.
4	размер Позволяет указать ширину элемента управления вводом текста в терминах символов.
5	максимальная длина Позволяет указать максимальное количество символов, которое пользователь может ввести в текстовое поле.

Контроль ввода пароля. Это также однострочный ввод текста, но он маскирует символ, как только пользователь вводит его. Они также создаются с использованием тега HTML `<input>`, но для атрибута `type` установлено значение **password** .

Вот основной пример однострочного ввода пароля, используемого для получения пароля пользователя:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Password Input Control</title>
  </head>
  <body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>

```

```

        Password: <input type = "password" name = "password" />
    </form>
</body>
</html>

```

Атрибуты. Ниже приведен список атрибутов для тега <input> для создания поля пароля.

Sr.No	Атрибут и описание
1	тип Указывает тип управления вводом и для управления вводом пароля будет установлен пароль .
2	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
3	значение Это может быть использовано для предоставления начального значения внутри элемента управления.
4	размер Позволяет указать ширину элемента управления вводом текста в терминах символов.
5	максимальная длина Позволяет указать максимальное количество символов, которое пользователь может ввести в текстовое поле.

Вот базовый пример многострочного ввода текста, используемого для описания элемента:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Multiple-Line Input Control</title>
  </head>
  <body>
    <form>
      Description : <br />
      <textarea rows = "5" cols = "50" name = "description">
        Enter description here...
      </textarea>
    </form>
  </body>
</html>

```

Атрибуты

Ниже приведен список атрибутов для тега <textarea>.

Sr.No	Атрибут и описание
-------	--------------------

1	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
2	строки Указывает количество строк текстового поля.
3	смещ_по_столбцам Указывает количество столбцов текстового поля

Флажок управления. Флажки используются, когда требуется выбрать более одной опции. Они также создаются с использованием тега HTML `<input>`, но для атрибута `type` установлено значение **checkbox**.

Вот пример HTML-кода для формы с двумя флажками —

```
<!DOCTYPE html>
<html>
  <head>
    <title>Checkbox Control</title>
  </head>
  <body>
    <form>
      <input type = "checkbox" name = "maths" value = "on"> Maths
      <input type = "checkbox" name = "physics" value = "on"> Physics
    </form>
  </body>
</html>
```

Атрибуты

Ниже приведен список атрибутов для тега `<checkbox>`.

Sr.No	Атрибут и описание
1	тип Указывает тип управления вводом и для флажка управления вводом будет установлен checkbox .
2	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
3	значение Значение, которое будет использоваться, если флажок установлен.
4	проверено Установите checkbox , если вы хотите выбрать его по умолчанию.

Радио кнопка управления. Радиокнопки используются, когда из множества параметров требуется выбрать только один параметр. Они также создаются с использованием тега HTML `<input>`, но для атрибута `type` установлено значение **radio**.

пример

Вот пример HTML-кода для формы с двумя переключателями —

```

<!DOCTYPE html>
<html>
  <head>
    <title>Radio Box Control</title>
  </head>
  <body>
    <form>
      <input type = "radio" name = "subject" value = "maths"> Maths
      <input type = "radio" name = "subject" value = "physics"> Physics
    </form>
  </body>
</html>

```

Атрибуты

Ниже приведен список атрибутов для радио кнопки.

Sr.No	Атрибут и описание
1	тип Указывает тип управления входом и для флажка управления входом он будет установлен на радио.
2	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
3	значение Значение, которое будет использоваться, если выбран переключатель.
4	проверено Установите <i>флажок</i> , если вы хотите выбрать его по умолчанию.

Выберите Box Control. Поле выбора, также называемое раскрывающимся списком, которое предоставляет возможность перечисления различных параметров в форме раскрывающегося списка, из которого пользователь может выбрать один или несколько параметров.

пример

Вот пример HTML-кода для формы с одним выпадающим списком

```

<!DOCTYPE html>
<html>
  <head>
    <title>Select Box Control</title>
  </head>
  <body>
    <form>
      <select name = "dropdown">
        <option value = "Maths" selected>Maths</option>
        <option value = "Physics">Physics</option>
      </select>
    </form>

```

```
</body>
</html>
```

Атрибуты

Ниже приведен список важных атрибутов тега `<select>` —

Sr.No	Атрибут и описание
1	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
2	размер Это может быть использовано для отображения списка прокрутки.
3	множественный Если установлено значение «несколько», то позволяет пользователю выбирать несколько пунктов из меню.

Ниже приведен список важных атрибутов тега `<option>` —

Sr.No	Атрибут и описание
1	значение Значение, которое будет использоваться, если выбрана опция в поле выбора.
2	выбранный Указывает, что этот параметр должен быть изначально выбранным значением при загрузке страницы.
3	этикетка Альтернативный способ маркировки опций

Поле загрузки файла. Если вы хотите разрешить пользователю загружать файл на ваш веб-сайт, вам нужно будет использовать окно загрузки файла, также известное как поле выбора файла. Это также создается с использованием элемента `<input>`, но для атрибута `type` установлено значение **file**.

Пример. Вот пример HTML-кода для формы с одним полем загрузки файла —

```
<!DOCTYPE html>
<html>
  <head>
    <title>File Upload Box</title>
  </head>
  <body>
    <form>
      <input type = "file" name = "fileupload" accept = "image/*" />
    </form>
  </body>
</html>
```

Атрибуты

Ниже приведен список важных атрибутов окна загрузки файла —

Sr.No	Атрибут и описание
1	название Используется для присвоения имени элементу управления, который отправляется на сервер для распознавания и получения значения.
2	принимать Указывает типы файлов, которые принимает сервер.

Кнопки управления. В HTML есть разные способы создания кнопок, которые можно нажимать. Вы также можете создать интерактивную кнопку, используя тег `<input>`, установив для ее атрибута типа **кнопку**. Атрибут `type` может принимать следующие значения:

Sr.No	Тип и описание
1	Отправить Это создает кнопку, которая автоматически отправляет форму.
2	сброс Это создает кнопку, которая автоматически сбрасывает элементы управления формы до их начальных значений.
3	кнопка Это создает кнопку, которая используется для запуска сценария на стороне клиента, когда пользователь нажимает эту кнопку.
4	образ Это создает активируемую кнопку, но мы можем использовать изображение в качестве фона кнопки.

Вот пример HTML-кода для формы с тремя типами кнопок:

```
<!DOCTYPE html>
<html>
  <head>
    <title>File Upload Box</title>
  </head>
  <body>
    <form>
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset" value = "Reset" />
      <input type = "button" name = "ok" value = "OK" />
      <input type = "image" name = "imagebutton" src = "/html/images/logo.png" />
    </form>
  </body>
</html>
```

Элементы управления скрытой формой. Элементы управления скрытой формой используются для сокрытия данных внутри страницы, которые впоследствии могут быть переданы на сервер. Этот элемент управления скрывается внутри кода и не отображается на реальной странице. Например, следующая скрытая форма используется для сохранения номера текущей страницы. Когда пользователь нажимает следующую страницу, тогда

значение скрытого элемента управления будет отправлено на веб-сервер, и там он решит, какая страница будет отображаться следующей, на основе переданной текущей страницы.

Вот пример кода HTML, чтобы показать использование скрытого элемента управления:

```
<!DOCTYPE html>
<html>

  <head>
    <title>File Upload Box</title>
  </head>

  <body>
    <form>
      <p>This is page 10</p>
      <input type = "hidden" name = "pagename" value = "10" />
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset" value = "Reset" />
    </form>
  </body>

</html>
```