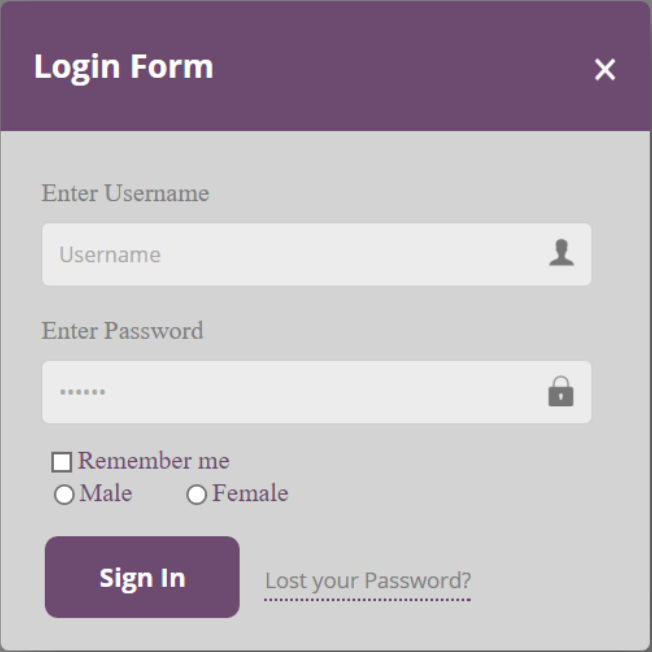


ЗАНЯТИЕ 6. РАБОТА С ФОРМАМИ

Формы встречаются почти на каждом сайте: для входа, регистрации, обратной связи, подписок и многих других вещей. Словом, умение работать с формами нам просто необходимо.



The image shows a login form titled "Login Form" with a close button (X) in the top right corner. The form has a light gray background and is set against a dark gray background. It contains the following elements:

- A label "Enter Username" above a text input field with the placeholder "Username" and a user icon on the right.
- A label "Enter Password" above a password input field with six dots as a placeholder and a lock icon on the right.
- A checkbox labeled "Remember me".
- Two radio buttons labeled "Male" and "Female".
- A purple "Sign In" button.
- A link "Lost your Password?" with a dotted underline.

Итак, начнем.

Для создания формы в html используется тег `<form>`, закрывающий тег для него обязателен.

Документ может содержать бесконечное количество форм, однако на сервер одновременно сможет отправиться только одна, поэтому не стоит делать формы зависимыми друг от друга.

Для отправки формы на сервер используется кнопка [Submit](#), если ее нет — форма отправляется по нажатию на Enter. Когда форма отправляется на

сервер, управление данными передается программе, заданной атрибутом **action** тега `<form>`.

Общий синтаксис:

```
<form action="URL">  
...  
</form>
```

Для данного тега можно задать следующие атрибуты:

Атрибут	Назначение
name	Задаёт имя формы
action	Адрес программы или документа, который будет обрабатывать данные формы
target	Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат
method	Метод протокола HTTP
accept-charset	Устанавливает кодировку на прием и обработку данных сервером
autocomplete	Включает автозаполнение формы
enctype	Способ кодирования данных формы
novalidate	Отменяется встроенная проверка на корректность ввода.

Как же создавать элементы формы?

Тег `<input>` является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем.

Общий синтаксис:

`<input атрибуты>`

Главным образом `<input>` предназначен для создания текстовых полей, различных кнопок, переключателей и флажков.

Хотя элемент `<input>` не требуется помещать внутрь контейнера `<form>`, определяющего форму, но если введенные пользователем данные должны быть отправлены на сервер, где их обрабатывает серверная программа, то указывать `<form>` обязательно.

Для данного тега можно задать следующие атрибуты:

<code>accept</code>	Устанавливает фильтр на типы файлов, которые вы можете отправить через поле загрузки файлов.
<code>align</code>	Определяет выравнивание изображения.
<code>alt</code>	Альтернативный текст для кнопки с изображением.
<code>autocomplete</code>	Включает или отключает автозаполнение.
<code>autofocus</code>	Устанавливает фокус в поле формы.

border	Толщина рамки вокруг изображения.
checked	Предварительно активированный переключатель или флажок.
disabled	Блокирует доступ и изменение элемента.
form	Связывает поле с формой по её идентификатору.
name	Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.
pattern	Устанавливает шаблон ввода.
placeholder	Выводит подсказывающий текст.
readonly	Устанавливает, что поле не может изменяться пользователем.
required	Обязательное для заполнения поле.
type	Сообщает браузеру, к какому типу относится элемент формы.
value	Значение элемента.