**КОНСПЕКТ “ФОРМЫ”**

**Первая форма**

Формы нужны для того, чтобы отправлять данные с веб-страницы на веб-сервер, который сможет эти данные обработать: зарегистрировать пользователя, создать сообщение на форуме, отправить письмо и так далее. В общем, формы в вебе просто необходимы.

Чтобы создать форму, нужно использовать парный тег <form>, внутри которого размещаются поля формы. У тега <form> есть два важных атрибута:

* action задаёт адрес, URL, отправки формы;
* method задаёт метод отправки формы.

Пример:

<form action="https://echo.htmlacademy.ru/courses" method="get">

*поля формы*

</form>

Для отправки формы обычно используют методы get или post. Если не указать атрибут method, то будет использован get.

# Текстовое поле ввода

Большинство полей форм создаётся с помощью одиночного тега <input>. У этого тега два обязательных атрибута:

* type задаёт тип поля;
* name задаёт имя поля.

Тип поля влияет на то, как оно будет отображаться и вести себя. Самый распространённый тип — это text, который обозначает текстовое поле. Он же используется по умолчанию. Пример:

<form action="https://echo.htmlacademy.ru/courses" method="get">

<input type="text" name="search">

</form>

Имя поля нужно, чтобы правильно обработать данные на сервере. Обычно, имя поля должно быть уникальным в пределах формы, хотя есть исключения.

# Идентификатор и значение по умолчанию

Атрибут id поля ввода обозначает идентификатор. Он должен быть уникальным не только в пределах формы, но и на всей странице.

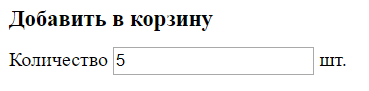
Обычно идентификаторы используют для повышения удобства работы с формой, например, создают подписи, связанные с мелкими полями. При нажатии на такие подписи активируется связанное поле. И это удобно, так как по большой подписи попасть легче, чем по маленькому полю. Также идентификаторы используют в JavaScript для работы с полями.

Идентификатор в отличие от имени поля не передаётся на сервер. Лучше использовать идентификаторы, отличающиеся от имени поля, особенно актуально это для полей множественного выбора, которые мы разберём далее в курсе.

Атрибут value задаёт значение поля ввода по умолчанию. Это тоже повышает удобство.

Согласитесь, приятно зайти в огромную анкету на каких-нибудь госуслугах, а там ваши паспортные данные уже подставлены в нужные поля и заполнять их не надо. И всё благодаря тому, что программист добавил к полям атрибут value с нужными данными.

<form action="https://echo.htmlacademy.ru/courses" method="post">

 Количество

<input type="text" name="total" **id="total-field" value="5"**> шт.

</form>

# Подпись для поля ввода

Можно подумать, что сделать подпись к полю очень просто. Пишем текст рядом с полем и всё готово:

Подпись <input type="text" name="username">

На самом деле этого недостаточно — мы получили просто кусок текста и поле, которые расположены рядом друг с другом, но логически никак не связаны.

Есть специальный тег, который позволяет смело сказать: «Этот кусок текста действительно подпись к этому полю!». Это **парный тег <label>.**

Он связывает текст и поле ввода логически. А ещё если нажать на текст в такой подписи, то курсор переместится в соответствующее поле.

Создавать подписи к полям с помощью <label> — хороший приём. Используйте его.

Первый способ создать подпись — просто обернуть текст подписи и тег поля в тег <label>, вот так:

**<label>**

Подпись <input type="text" name="username">

**</label>**

Надо отметить, что при оборачивании текста в тег <label> он визуально никак не меняется, ведь главная задача подписи — создать логическую связь.

# Связываем подпись и поле по id

Иногда обернуть поле и текст подписи в тег <label> нельзя. Например, когда они размещены в разных ячейках таблицы.

В этом случае можно связать подпись с полем с помощью атрибута id. Алгоритм такой:

1. Добавляем к полю ввода идентификатор с помощью атрибута id.
2. Оборачиваем текст подписи в тег <label>.
3. Добавляем тегу <label> атрибут for.
4. В атрибут for записываем такое же значение, что и в атрибуте id у поля.

Например:

<label for="user-field-id">Имя пользователя</label>

...

*много-много других тегов*

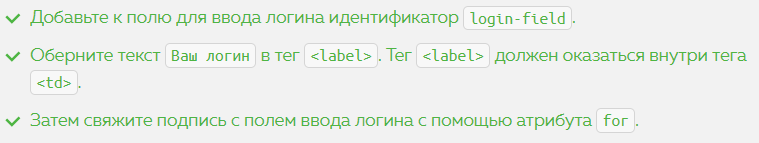
...

<input id="user-field-id" type="text" name="username">

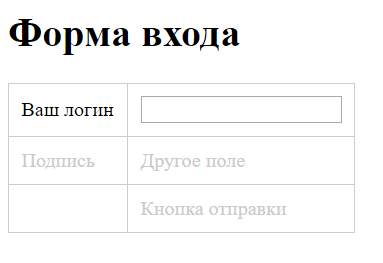
Конец формы

Было

Стало







# Поле для ввода пароля

Мы создаём простую форму входа. Уже есть два поля и связанные с ними подписи. Одно из полей предназначено для ввода пароля, но сейчас оно является простым текстовым полем.

Чтобы сделать его настоящим полем для ввода пароля, в котором текст будет отображаться «звёздочками», нужно просто изменить значение атрибута type на password.

<input type=**"password"** name="password" id="password-field" value="Где же звёздочки?">

# Кнопка отправки формы

Форма практически готова. Осталось добавить кнопку для отправки формы. Такая кнопка создаётся с помощью тега <input> c типом submit.

Надпись на кнопке можно задать с помощью атрибута value. Для кнопки отправки формы задавать имя необязательно. Но если имя задано, то на сервер будут отправляться имя и значение кнопки.

Обычно имя для кнопки отправки задают, когда в форме несколько кнопок, отвечающих за разные действия. Браузер отправляет на сервер имя и значение только той из них, на которую нажал пользователь. Таким образом, сервер может понять, какую кнопку нажали и что нужно сделать.

<input type="**submit**" **value**="Войти">

# Многострочное поле ввода

Мы научились создавать простейшие формы с текстовыми полями и кнопками. А теперь познакомимся с более сложными элементами формы.

Многострочное текстовое поле создаётся с помощью парного тега **<textarea>.** У него есть атрибуты name и id, которые аналогичны атрибутам текстового поля.

Атрибут rows принимает целочисленное значение и задаёт высоту многострочного поля в строках. Атрибут cols задаёт ширину поля в символах. В качестве ширины символа берётся некоторая «усреднённая ширина».

Атрибут value у многострочного поля отсутствует, а значение по умолчанию задаётся по-другому. Текст, расположенный между открывающим и закрывающим тегом <textarea> и является значением по умолчанию. Вот так:

<textarea>Значение по умолчанию</textarea>

<textarea name="comment" id="comment-field" rows="10">Текст комментария</textarea>

# Чекбокс или «галочка»

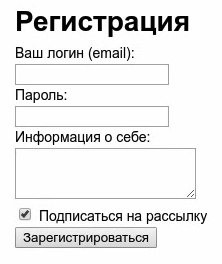
Поле-галочка — это тег <input> с типом checkbox.

Галочка работает по принципу «либо да, либо нет». Если галочка стоит, то браузер посылает переменную с именем поля на сервер, если галочки нет, то не посылается ничего. Таким образом, атрибут value не является обязательным.

Чтобы галочка стояла по умолчанию, нужно добавить к тегу атрибут checked. Вот так:

<input type="checkbox" checked>

Чекбокс не подразумевает выбор одного элемента из нескольких. Поэтому если в одной форме есть несколько чекбоксов, то имена у них должны быть разными.





# Переключатель или «радиобаттон»

Поле-переключатель — это тег <input> с типом radio.

Обычно переключатели размещают группами по несколько штук. Причём у переключателей из одной группы должно быть одинаковое имя и разные значения, которые задаются c помощью value.

Таким образом, атрибут value является для переключателей обязательным. Браузер отправляет на сервер значение value выбранного переключателя.

<input type="**radio**" name="question-one" **value**="p">

# Группа переключателей

Теперь добавим ещё один вариант ответа в наш переключатель. Для этого нужно добавить ещё один <input> с таким же именем, но другим значением value.

Подобным образом можно создавать группы переключателей с любым количеством вариантов.

Чтобы сделать какой-либо вариант в переключателе выбранным по умолчанию, нужно добавить к соответствующему тегу <input> атрибут checked, как у поля-галочки.

Кстати, имя поля у переключателей одной группы должно быть одинаковым, но идентификаторы всегда должны быть уникальными.

<label>

<input type="radio" name="question-one" value="p"> Тег &lt;p&gt;

</label>

<br>

<label>

<input type="radio" name="question-one" value="b" checked> Тег &lt;b&gt;

</label>

# Раскрывающийся список или «селект»

Раскрывающийся список так же, как и переключатель, позволяет выбрать один вариант ответа из нескольких.

Раскрывающийся список создаётся с помощью парного тега **<select>,** у которого есть знакомые атрибуты name и id.

Варианты ответов задаются с помощью парных тегов <option>, которые должны располагаться внутри тега <select>. Например:

<select name="theme">

<option value="light">Светлая тема</option>

<option value="dark">Тёмная тема</option>

...

</select>

В атрибуте value тега <option> задаётся значение варианта ответа, а внутри этого тега располагается подпись варианта ответа.

Если при отправке формы у выбранного варианта задан value, то на сервер отправится значение этого атрибута. В противном случае будет отправлен текст подписи.

# «Мультиселект»

Раскрывающийся список можно превратить в так называемый «мультиселект», то есть список, в котором можно выбрать не один, а несколько вариантов.

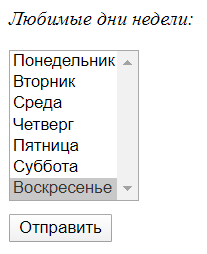
Чтобы сделать это, нужно добавить к тегу <select> атрибут multiple.

Выбрать несколько вариантов можно, щёлкая по ним с зажатой клавишей Ctrl на Windows или Command на MacOS.

Высоту мультиселекта можно изменять с помощью атрибута size тега <select>.

Чтобы отметить как выбранные по умолчанию одно или несколько значений, нужно к соответствующим тегам <option> добавить атрибут selected.

При отправке данных мультиселекта на сервер с PHP после имени в значении атрибута name ставятся символы квадратных скобок []. Например, <select name="days[]">. Это необязательное требование для имени мультиселекта, а нужно только для корректной обработки данных в PHP.



<select name="days" **multiple**  **size**="7">

<option>Понедельник</option>

<option>Вторник</option>

<option>Среда</option>

<option>Четверг</option>

<option>Пятница</option>

<option>Суббота</option>

<option **selected**>Воскресенье</option>

</select>

# Поле для загрузки файлов

Поле для загрузки файлов — это тег <input> с типом **file**. Для этого поля обязательным атрибутом является имя.

Чтобы поле заработало и браузер смог передать выбранный файл на сервер, необходимо добавить форме атрибут enctype со значением multipart/form-data. Не полю, а форме.

Этот атрибут указывает браузеру, в каком виде пересылать данные. Если вы хотите отправить на сервер файл, данные из формы эффективнее всего передавать по частям. За это и отвечает значение multipart/form-data.

Кстати, внешний вид полей с типом file очень сильно отличается в зависимости от операционной системы и очень плохо изменяется с помощью стилей.

<form action="https://htmlacademy.ru/courses" method="post" **enctype="multipart/form-data"**>

<p>Выберите изображение для своего аватара:</p>

<input type="**file**" name="avatar">

<input type="submit" value="Загрузить">

</form>

# Скрытое поле

И ещё одно невидимое и очень полезное поле. Это скрытое поле. Его используют, когда в форме нужно отправить какие-то дополнительные служебные данные, которые не вводятся пользователем.

Например, это могут быть реквизиты заказа или номер пользователя в форме оплаты.

Скрытое поле — это тег <input> с типом hidden.

<input type="**hidden**" name="account\_number" value="15">

