**Знакомство с формами**

|  |
| --- |
| [Первая форма](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/1) |
| [Текстовое поле ввода](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/2) |
| [Идентификатор и значение по умолчанию](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/3) |
| [Подпись для поля ввода](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/4) |
| [Связываем подпись и поле по id](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/5) |
| [Добавим ещё одно поле](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/6) |
| [Поле для ввода пароля](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/7) |
| [Кнопка отправки формы](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/8) |
| [Многострочное поле ввода](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/9) |
| [Чекбокс или «галочка»](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/10) |
| [Испытание: форма регистрации](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/11) |
| [Переключатель или «радиобаттон»](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/12) |
| [Группа переключателей](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/13) |
| [Раскрывающийся список или «селект»](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/14) |
| [«Мультиселект»](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/15) |
| [Поле для загрузки файлов](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/16) |
| [Скрытое поле](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/17) |
| [Испытание: редактирование профиля](https://npoed.htmlacademy.ru/htmlcss110/course/6/run/18) |

# Первая форма [1/18]

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="/echo" method="get">

поля формы

</form>

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

Метод get посылает данные формы в строке запроса, то есть они видны в адресной строке браузера и следуют после знака вопроса. Например:

https://www.google.com/search?q=htmlacademy

Метод get лучше использовать в поисковых формах, потому что он позволяет получить ссылку на результаты поиска и передать её кому-то.

Метод post посылает данные в теле HTTP-запроса и используется, когда нужно отправить много данных и ссылка на результат обработки этих данных не нужна. Например, при редактировании личного профиля.

# Текстовое поле ввода [2/18]

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="/echo" 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. Для задания имени поля используют латинские буквы и цифры.

# Идентификатор и значение по умолчанию [3/18]

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 щелчке по таким подписям 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.

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.

# Подпись для поля ввода [4/18]

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 щёлкнуть по тексту 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 [5/18]

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">

# Добавим ещё одно поле [6/18]

Пришло время добавить в форму ещё одно поле.

Сделайте это по всем правилам хорошего тона: помимо имени задайте полю идентификатор, добавьте подпись к полю и свяжите их.

Мы могли бы связать поле и подпись, просто обернув их в тег <label>. Но в этом задании мы тренируем более сложный вариант — с использованием id и for.

# Поле для ввода пароля [7/18]

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.

# Кнопка отправки формы [8/18]

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.

# Многострочное поле ввода [9/18]

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>

# Чекбокс или «галочка» [10/18]

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.

# Испытание: форма регистрации [11/18]

Создайте простейшую форму регистрации с использованием уже изученных полей.

*Подсказки:*

* Вам нужно размечать только HTML-код.
* Для переноса полей на новую строку используйте обычный <br>.
* Ширину полей задавать не нужно, в частности cols у многострочного поля.
* Высоту многострочного поля подберите с помощью rows.

Так как поля форм могут очень сильно отличаться от системы к системе, то ориентируйтесь по вкладкам «Образец» и «Различия», где показано как наш сервер видит вашу форму.

# Переключатель или «радиобаттон» [12/18]

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.

В этом задании мы начнём создавать переключатель.

# Группа переключателей [13/18]

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.

# Раскрывающийся список или «селект» [14/18]

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» [15/18]

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 OS X.

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.

# Поле для загрузки файлов [16/18]

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.

Кстати, внешний вид таких полей очень сильно отличается от операционной системы и очень плохо изменяется с помощью стилей.

# Скрытое поле [17/18]

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.

# Испытание: редактирование профиля [18/18]

Создайте более сложную форму редактирования профиля.

*Подсказки:*

* Вам нужно размечать только HTML-код.
* Для переноса полей на новую строку используйте обычный <br>.
* Поля формы по умолчанию располагаются на одной строке, если помещаются по ширине.

Так как поля форм могут очень сильно отличаться от операционной системы, то ориентируйтесь по вкладкам «Образец» и «Различия», где показано как наш сервер видит вашу форму.