Первая форма

Для начала определимся, для чего они нужны. 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, они 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://forms.example.ru/" 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=НАТК

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

Задание

1. Для начала создайте новый HTML-документ с именем form.html
2. Заполните его стандартным содержанием(практическая работа №1)
3. Очистите тело документа, оставив только заголовок первого уровня - “Анкета”
4. В body создайте новую форму, атрибуту action присвойте значение #, метод 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://forms.example.ru/" 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

Задание

1. Создадим первое поле ввода, которое будет запрашивать имя анкетируемого
2. Атрибуту name задайте соответствующее имя (например userName)

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.

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

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

Задание

1. Добавим связанную подпись для поля. для этого перед наши input введем строку: “Укажите Ваше имя:”
2. Обернем все - и строку и input в тег label
3. Нажмите на подпись, чтобы проверить, что поле активируется

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.

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

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">
```

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

Задание

1. Создайте поле для фамилии согласно алгоритма приведенного выше

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

Задание

1. Создайте поле с атрибутом password
2. label укажите как - Введите пароль

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>

Задание

1. Создайте поле с подписью “Пару слов о себе” с помощью нового тега
2. Подберите ширину и высоту по своему желанию

Чекбокс

Чекбокс — это тег <input> с типом checkbox.

Чекбокс работает по принципу «либо да, либо нет». Если чекбокс включён, то браузер посылает переменную с именем поля на сервер, а если выключен, то не посылается ничего. Таким образом, атрибут value не является обязательным.

Чтобы сделать чекбокс включённым 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

Задание

1. Создайте input, type установите "checkbox", id, например "question"
2. Создайте подпись, в которой будет находиться сам вопрос: <label for="question">Вы студент?</label>
3. Чтобы вопрос по умолчанию был отмечен - установите для input атрибут checked

Радиокнопка

Радиокнопка — это тег <input> с типом radio.

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

Таким образом, атрибут value является для радиокнопок обязательным. Браузер отправляет на сервер значение value выбранной радиокнопки.

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

Кстати, имя у радиокнопок одной группы должно быть одинаковым, но идентификаторы всегда должны быть уникальными

Создадим новый вопрос в нашей анкете

Задание

1. Создайте вопрос внутри тега p - “Где ты учишься?”
2. Варианты ответа будем хранить в конструкции label - input, для того чтобы клик по тексту варианта автоматически выделял вариант радиокнопки:
3. создаем парный тег label
4. внутри создаем 3 радиокнопки, атрибуту name для каждой присваиваем значение “eduorg” (сокращение от educational organization),
5. значение value для первой school
6. значение value для второй college
7. значение value для третьей university
8. пример конструкции:

```
<label>
        <input type="radio" name="eduorg" value="school">
        Школа
</label>
```

Раскрывающийся список, или «селект»

Раскрывающийся список так же, как и радиокнопки, 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="city">
  <option value="Novosibirsk">Новосибирск</option>
  <option value="Tomsk">Томск</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

Задание

1. Создайте новый вопрос с помощью конструкции label - select
2. Вопрос - “Ваш год рождения?” должен находиться внутри label, сразу перед select
3. атрибуту name тега select задайте значение bdyear (сокращение от birthday year)
4. создайте 5 опций - от 2002 до 2006, с соответствующими значениями value

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.

Задание

1. Задайте для формы атрибут enctype необходимый для загрузки файлов.
2. Снова создаем конструкцию label - input
3. в label пишем - “Добавьте Ваше фото”
4. Добавляем поле для загрузки файлов с именем profilepic

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

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

В предыдущем задании мы сделали кнопку загрузки фото пользователя. Внешний вид полей с типом file очень сильно отличается в зависимости от операционной системы и очень плохо изменяется с помощью стилей, поэтому мы можем сделать небольшой трюк - скрыть кнопку! При этом не потеряв функционал загрузки

Задание

1. Задайте input атрибут hidden
2. Проверьте работает ли загрузка файла, кликнув по надписи “Добавьте Ваше фото” в браузере

Подсказка при заполнении полей

У полей, в которые вводятся текстовые значения (textarea, разные типы input и так далее) есть возможность вывести подсказку.

Для этого используется специальный атрибут placeholder:

```
<input type="text" placeholder="Текст подсказки">
```

Текст подсказки выводится внутри текстового поля, а при вводе значения — автоматически убирается.

Воспользуемся этим атрибутом, чтобы подсказывать что нужно вводить в поля анкеты

Задание

1. добавьте полям атрибут placeholder с текстом подсказки

Обязательные поля

Зачастую в формах есть поля, информация из которых обязательна к заполнению. Чтобы указать, что поле обязательно для заполнения, нужно добавить ему пустой атрибут required:

```
<input type="text" required>
```

При попытке отправить форму с незаполненными обязательными полями браузер выводит всплывающее предупреждение. Эта проверка работает на клиентской части и упрощает валидацию форм.

Задание

1. Установите для поля ввода имени атрибут required
2. Попробуйте отправить форму, не заполняя имя

Поле выбора даты

В форму можно включить поле даты. Мы уже делали выбор даты рождения из нескольких вариантов. А для выбора даты окончания обучения сделаем по новому. Выберем его из календаря, для этого существует тип поля ввода — date. При клике на данное поле в форме всплывает календарик. Пример записи:

<input type="date">

Если браузер не поддерживает поле для ввода даты, то вместо него отображается обычное текстовое поле

Задание

1. Создайте новую конструкцию label - input
2. Вопрос внутри label - “Дата окончания обучения”
3. Далее задайте для input тип "date”

Поле ввода телефона

Добавим в форму ещё одно поле с типом tel.

Новый тип поля tel появился в HTML5 и отвечает за ввод телефонных номеров.

В мобильных браузерах при фокусе на такое поле появляется клавиатура, позволяющая вводить только цифры и символы телефонных номеров. Спросим у пользователя номер телефона, либо личный, либо тот, о котором он мечтает

Задание

1. Создайте новую конструкцию label - input
2. Вопрос внутри label - например “Твой номер телефона”
3. Далее задайте для input тип "tel”

Поля ввода адресов сайтов и email

В HTML5 добавлены два типа полей email и url, предназначенные для ввода электронной почты и адреса сайта. Особенностью этих полей является то, что они автоматически проверяют формат введённых данных. Пример записи:

```
<input type="email">
<input type="url">
```

Внешне эти поля не отличаются от обычных текстовых полей, но обладают важной особенностью, которая очень полезна на мобильных устройствах.

Когда вы начинаете заполнять такое поле на мобильнике, там автоматически переключается раскладка клавиатуры. Например, для email отобразятся латинские символы, цифры, знак @ и некоторые другие.

Задание

1. Создайте 2 новые конструкции label - input
2. Создайте соответствующие вопросы о личной почте и любимом сайте
3. Создайте input’ы с подходящими типами

Запрет редактирования полей

Иногда возникают ситуации, когда какие-то поля требуется сделать недоступными для редактирования.

Есть два способа: использование атрибута readonly и использование атрибута disabled. Пример записи:

```
<input type="text" readonly>
<input type="text" disabled>
```

Атрибут readonly не дает пользователю изменять поле (вводить новый текст, модифицировать существующий). Введенное значение можно выделить и скопировать. Данные из этого поля отправляются на сервер.

Атрибут disabled не дает пользователю изменять поле (вводить новый текст, модифицировать существующий). Нельзя поставить фокус в это поле, введенное значение нельзя выделять и копировать. Данные из этого поля НЕ отправляются на сервер

Сброс введенных значений

Иногда появляется необходимость очистить форму. В HTML-формах есть специальная кнопка, которая сбрасывает введённые значения и возвращает изначально установленные. Это поле ввода с типом reset. Пример использования:

<input type="reset" value="Сбросить">

Обратите внимание, что кнопка не обнуляет значения, а возвращает те, которые были установлены в полях формы по умолчанию

Задание

1. Создайте кнопку сброса

Конспект «Формы»

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

- action задаёт URL, адрес для отправки формы;
- method задаёт метод отправки формы.

Пример:

```
<form action="https://forms.example.ru/" 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.

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

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.

От типа поля зависит то, как оно будет отображаться и вести себя. Самый распространённый тип — это text, который обозначает текстовое поле. Данный тип используется по умолчанию.

Для того, чтобы правильно обработать данные на сервере нужно имя поля. Обычно, имя поля должно быть уникальным в пределах формы. Для задания имени поля используют латинские буквы и цифры.

Идентификатор и значение по умолчанию

Атрибут id поля ввода обозначает идентификатор. Он должен быть уникальным 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. Также идентификаторы используют в JavaScript для работы с полями.

Идентификатор не передаётся на сервер. Лучше использовать идентификаторы, отличающиеся от имени поля.

Атрибут value задаёт значение поля ввода по умолчанию.

Подпись для поля ввода

Парный тег <label> создаёт подпись к полю формы.

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

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

```
<label>
  Подпись <input type="text" name="username">
</label>
```

При оборачивании текста в тег <label> он визуально никак не меняется.

Связь подписи и поля по id

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">
```

Поле ввода пароля

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

Кнопка отправки формы

Кнопка для отправки формы создаётся с помощью тега <input> с типом submit.

Надпись на кнопке можно задать с помощью атрибута value. Для кнопки отправки формы задавать имя необязательно, но если оно 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.

Многострочное поле ввода

Многострочное текстовое поле создаётся с помощью парного тега <textarea>. У него есть атрибуты name и id, аналогичные 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. Текст, расположенный между открывающим и закрывающим тегом <textarea> является значением по умолчанию.

Чекбокс

Чекбокс — это тег <input> с типом checkbox.

Он 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. Атрибут value не является обязательным.

Чтобы поле было отмечено по умолчанию, нужно добавить к тегу атрибут checked.

Имена чекбоксов в рамках одной формы должны быть разными, так как чекбокс не подразумевает выбор одного элемента из нескольких

Радиокнопка

Радиокнопка — это тег <input> с типом radio.

Значение радиокнопки задается с помощью value. Атрибут value является для радиокнопок обязательным. Браузер отправляет на сервер значение value выбранной радиокнопки. Если добавить ещё несколько <input> с таким же именем, но другими значениями value, то мы получим группу радиокнопок. Имя поля у радиокнопок одной группы должно быть одинаковым, но идентификаторы всегда должны быть уникальными.

Чтобы сделать какой-либо вариант выбранным по умолчанию, нужно добавить к соответствующему тегу <input> атрибут checked.

Раскрывающийся список, или «селект»

Раскрывающийся список позволяет выбрать один вариант ответа из нескольких. Такой список создаётся с помощью парного тега <select>, у которого есть 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 несколько вариантов. Для этого нужно добавить к тегу <select> атрибут multiple.

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, оно нужно только для корректной обработки данных в PHP.

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 всей форме атрибут enctype со значением multipart/form-data. 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.

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

Скрытое поле

Скрытое поле используют, когда в форме нужно отправить какие-то дополнительные служебные данные, которые не вводятся пользователем.

Скрытое поле — это тег <input> с типом hidden

Подсказка при заполнении полей

Для добавления подсказки к полю формы используется специальный атрибут placeholder:

<input type="text" placeholder="Текст подсказки">

Текст подсказки выводится внутри текстового поля, а при вводе значения — автоматически убирается.

Обязательные поля

Чтобы указать, что поле обязательно для заполнения, нужно добавить ему пустой атрибут required:

<input type="text" required>

При попытке отправить форму с незаполненными обязательными полями браузер выведет всплывающее предупреждение. Эта проверка работает на клиентской части и упрощает валидацию форм. Но всегда нужно проверять отправленные данные и на стороне сервера.

Поле выбора даты и времени

Для выбора даты из календаря существует новый тип поля ввода — date. При клике на данное поле в форме всплывает календарик.

Пример записи:

<input type="date">

Для указания времени существуют дополнительные «временные» типы полей, например, time для выбора времени.

<input type="time">

Также существуют следующие типы полей:

- datetime-local задаёт дату с указанием времени (без учета временной зоны);
- week задаёт порядковый номер недели в году и года;
- month задаёт месяц и год.

Если браузер не поддерживает указанный тип поля, то вместо него отображается обычное текстовое поле.

Поле ввода телефона

За ввод телефонных номеров отвечает тип поля tel. В такое поле полезно добавлять атрибут pattern, чтобы избежать ошибки при вводе данных.

В мобильных браузерах при фокусе на такое поле появляется клавиатура, позволяющая вводить только цифры и символы телефонных номеров.

Поля ввода адресов сайтов и email

В HTML5 добавлены два типа полей email и url, предназначенные для ввода электронной почты и адреса сайта. Эти поля автоматически проверяют формат введённых данных.

Пример записи:

<input type="email">

<input type="url">

Запрет редактирования полей

Для того, чтобы сделать поле недоступным для редактирования есть два способа: использование атрибута readonly и использование атрибута disabled

Атрибут readonly не дает пользователю изменять поле (вводить новый текст, модифицировать существующий). Введенное значение можно выделить и скопировать. Данные из этого поля отправляются на сервер.

Атрибут disabled не дает пользователю изменять поле (вводить новый текст, модифицировать существующий). Нельзя поставить фокус в это поле, введенное значение нельзя выделять и копировать. Данные из этого поля НЕ отправляются на сервер.

Сброс введенных значений

В HTML-формах есть специальная кнопка, которая сбрасывает введённые значения и возвращает изначально установленные. Это поле ввода с типом reset.

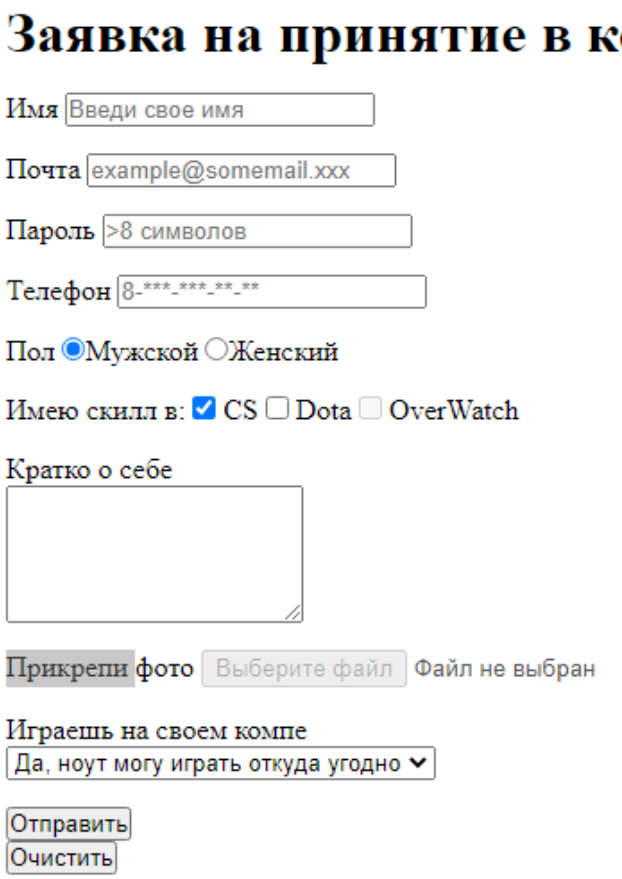
Пример использования:

<input type="reset" value="Сбросить">

Кнопка возвращает значения, которые были установлены в полях формы по умолчанию

Испытание: заявка на принятие в команду

Давайте представим, что мы админим небольшой районный портал о киберспорте. Игру можно выбрать на свой вкус. Иногда к нам обращаются знакомые и незнакомые игроки, с желанием попасть в команду. Чтобы упростить себе задачу - каждому из них можно скинуть первичный опросник об их игровых навыках. Создайте опросник по образцу из скрина ниже



1. Значение method form должно быть равно 'get', атрибут action не используем
2. Каждому input задаем атрибут name, подходящий в данном случае (не знаешь слова по английски - пиши транслитом)
3. Первые 4 пункта - обязательны к заполнению
4. В чекбоксе:
    a. атрибут value должен быть равен 'yes'
    b. третий пункт запретить к выбору
5. Размер поля “Кратко о себе” по высоте и ширине кратен 5, необходимо подобрать значения
6. Кнопка прикрепления фото запретить к выбору
7. Значения для выпадающего списка:
    a. Да, только из дома
    b. Нет, могу на вашем
    c. Да, ноут могу играть откуда угодно (должен быть выбран по умолчанию)

8. Создай файл стиле style.css и подключи его в head с помощью:
   <link rel="stylesheet" href="style.css">
9. В него скопирую данный код:

```
* {
  display: block;
  margin: 0;
  padding: 0;
}
body {
  width: 650px;
  margin: 0 auto;
  padding: 10px 20px;

}
input{
  display: inline;
 }

label{
  margin: 15px 0;
}
```