Прочитайте всю лекцию, разберите теги и атрибуты форм. Запишите в тетрадь конспект по вопросам (обратите внимание, что примеры кода и многие атрибуты в тетрадь писать не нужно):

1. Какой тег используется для создания формы на сайте?
2. Опишите обязательный атрибут тега form
3. Опишите текстовые поля формы
4. Для чего служат атрибуты name и value у полей форм?
5. Как задать метки на форме?
6. Какой тег служит для задания флажка?
7. Какой тег служит для задания переключателя? Что нужно обязательно сделать, чтобы переключатели нельзя было нажать одновременно?
8. Какими двумя способами можно добавить кнопки на форму? Опишите возможные значения атрибута type для кнопок.
9. Запишите теги для загрузки файла, выпадающего списка и текстовой области.

**Формы и их элементы**

Форма – один из самых необходимых элементов современных сайтов. Формы позволяют получать информацию от пользователя, отправлять данные на сервер или администратору на почту.

Примерами форм на сайте являются формы авторизации и регистрации (рис. 1), формы обратной связи (рис. 2), формы для заказа товара, формы подписки, формы голосования и т.д.

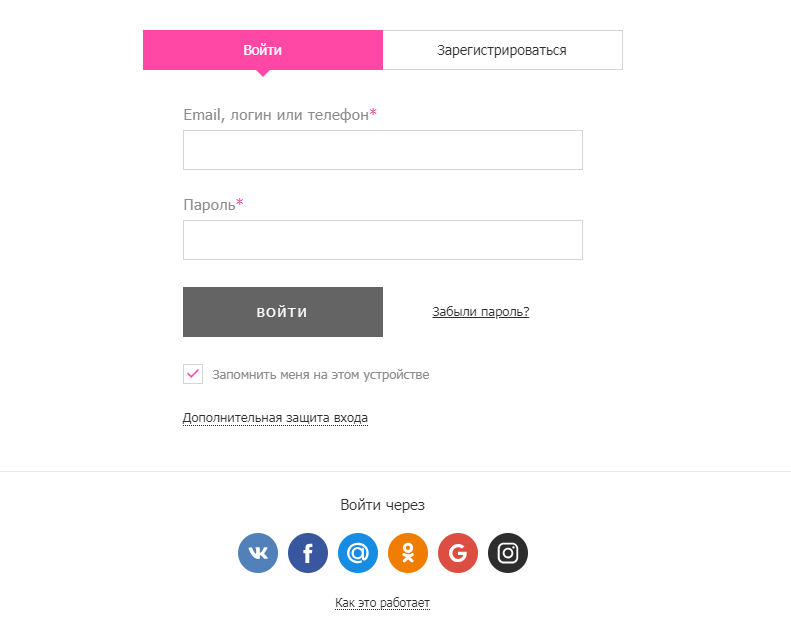
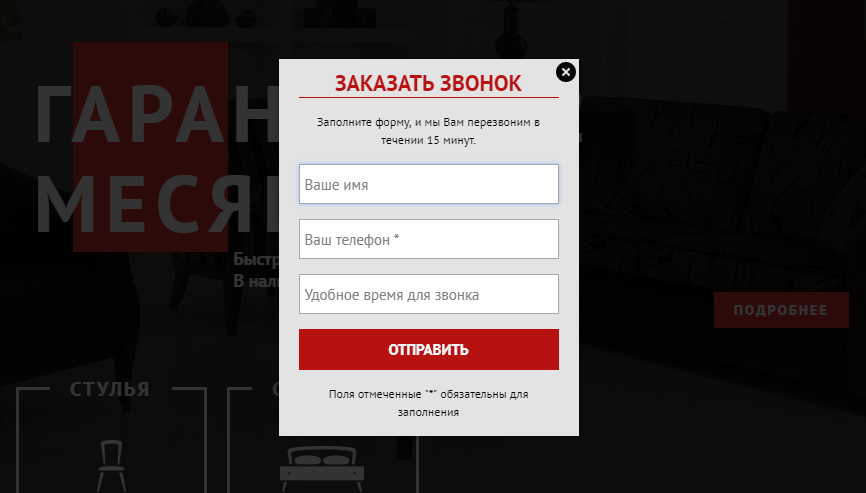
 

Рис. 1 Форма авторизации Рис. 2 Форма обратной связи

**1. Тег <form>**

**Для размещения формы на веб-странице используется контейнер <form> </form>.** Тэг **<form>** может содержать атрибуты, описанные в таблице 1.

Таблица 1. Атрибуты тега <form>

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | action | **Обязательный атрибут**. Определяет файл на сервере, который будет обрабатывать данную форму.  Синтаксис:  <form action = "action.php"> </form>  Если значение атрибута не указано, то данные формы не будут отправлены и после перезагрузки страницы элементы формы примут значения по умолчанию. |
|  | method | Определяет, каким образом данные из формы будут переданы обработчику.  Допустимые значения: method="post" и method="get".  Если значение атрибута не установлено, по умолчанию предполагается method="get". |
|  | enctype | Определяет, каким образом данные из формы будут закодированы для передачи обработчику.  enctype ="multipart/form-data" – данные не кодируются, это значение применяется при отправке файлов. |
|  | name | Задает имя формы, которое будет использоваться для доступа к ее элементам. |

Пример записи атрибутов тега <form> может быть следующим:

<form action = "index.php" method = "post" enctype ="multipart/form-data">

</form>

В данном случае обработка данных осуществляется в файле index.php, в котором используется серверный язык программирования php.

**2 Элементы (поля) форм**

Наиболее часто в формах используется тег **<input>**, позволяющий вводить и отправлять на сервер различные данные. Вид выводимого поля и тип передаваемых данных зависит от атрибута type.

1. ***Текстовые поля***

а) **текст:** **<input type="text">** - однострочное поле ввода текста;

любой текст

б) **пароль: <input type="password">** - поле ввода пароля, символы заменяются точками;

\*\*\*\*\*\*\*

в) **email: <input type="email">** - поле ввода электронной почты, может показывать предупреждение, если введён некорректный email;

adress@mail.ru

г) **число <input type="number">** - поле ввода числа.



Для поля ввода числа также можно добавить атрибуты, описанные в таблице 2.

Таблица 2. Атрибуты тега <input type="number">

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | min | Минимальное значение числа |
|  | max | Максимальное значение числа |
|  | step | Шаг приращения числа. Может быть целым (2) или дробным (0.5) |
|  | value | Начальное число, которые выводится в поле |
|  | size | Ширина поля |

Для многих элементов формы, в том числе и для тега <input>, используются универсальные атрибуты, описанные в таблице 3.

Таблица 3. Универсальные атрибуты элементов форм

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | **name** | **Имя поля, по которому обработчик формы идентифицирует элемент.** |
|  | size | Ширина поля в символах. |
|  | **value** | * **Начальный текст, отображаемый в поле.** * **Значение, отправляемое на сервер.** |
|  | required | Обязательное поле. Выводит сообщение о том, что пользователь должен ввести данные или выбрать значение. Атрибут используется без указания значения. |
|  | disabled | Поле, недоступное для ввода. Атрибут используется без указания значения. |
|  | autofocus | Поле автоматически получает фокус после загрузки страницы. Атрибут используется без указания значения. |

Текстовые поля могут отображать **подсказку**, которая исчезнет, как только будет введён некоторый текст. Для этого используется атрибут placeholder.

<form action = " ">

<input type="text" placeholder="Введите свое имя">

</form>

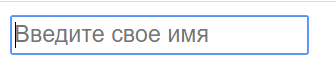


Рис. 3 Текстовое поле с атрибутом placeholder

Если начать набирать данные в текстовом поле, то текст «Введите своё имя» исчезнет.

В следующем примере показан html-код формы с различными текстовыми полями и их атрибутами.

<form action = " ">

<input type="text" size = "30" placeholder="Введите имя" >

<input type="password" size = "20" placeholder="Введите пароль" >

<input type="email" size = "20" placeholder="Введите email" >

<input type="number" size = "5" min="1" max="20" step="1" value="1">

</form>

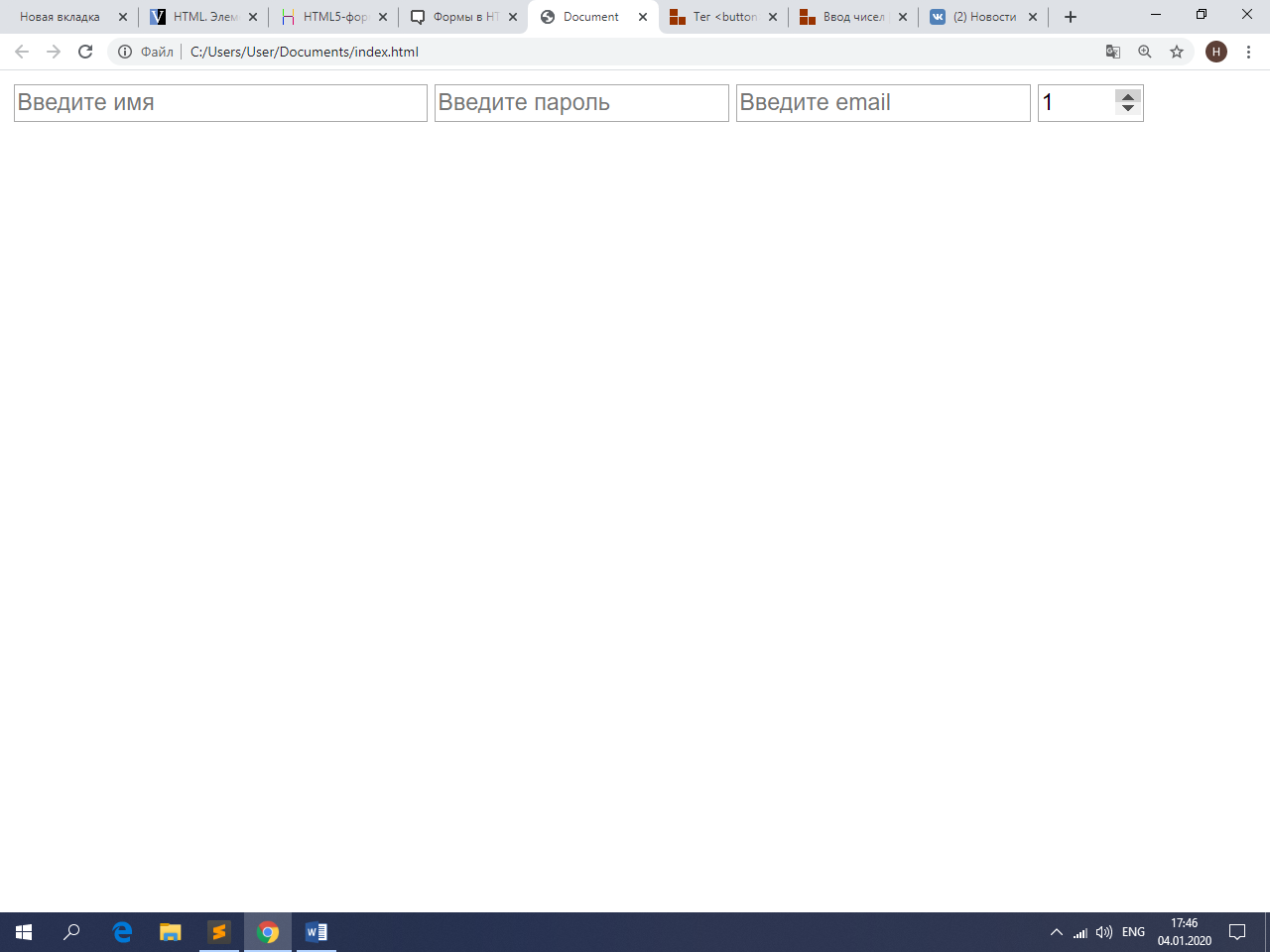


Рис. 4 Результат в браузере

Текстовые поля, как и многие другие элементы формы, являются строчными элементами, поэтому для переноса поля на следующую строку следует использовать тег <br> или свойства CSS. Также можно каждое текстовое поле обернуть в тег абзаца <p>.

Чтобы создать к элементам формы подсказки, видимые в любой момент, используются **текстовые метки <label> </label>**.

Метка <label> может быть связана с полем двумя способами:

1. Тег с полем помещается внутрь контейнера **<label> </label>.**

<label> Имя

<input type="text">

</label>

1. С помощью атрибута for, значение которого соответствует значению id (идентификатор) у поля.

<label for="first\_name">Имя </label>

<input type="text" id="first\_name">



Рис. 5 Текстовое поле с меткой

При щелчке по метке фокус переходит к текстовому полю и помещает курсор внутрь него. Эта связь особенно полезна при работе с флажками и переключателями.

1. ***Переключатели и флажки***

**Флажок** **(«чекбокс»)** используют, когда необходимо выбрать любое количество вариантов из предложенного списка.

**Для создания флажков используется тег: <input type="checkbox">.**



По небольшому флажку часто бывает сложно щелкнуть мышкой, поэтому рекомендуется помещать флажок и его описание внутрь метки <label>.

<label>

<input type="checkbox"> Я согласен с условиями

</label>

В данном примере можно щёлкнуть по тексту «Я согласен с условиями», чтобы переключить флажок.

По умолчанию флажок выключен. Чтобы пометить его по умолчанию включенным, используется атрибут checked.

<label>

<input type="checkbox" checked> Я согласен с условиями

</label>

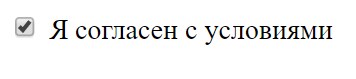


Рис. 6 Флажок

**Переключатели** используют, когда необходимо выбрать один единственный вариант из нескольких предложенных. Переключатели являются **взаимоисключающими**, т.е. выбор одного из вариантов исключает выбор другого.

Переключатели создаются с помощью тега: **<input type="radio">**.



**Чтобы переключатели относились к одной группе, необходимо чтобы у них был одинаковый атрибут name, иначе их можно будет нажать одновременно.**

По умолчанию пункты переключателя выключены. Чтобы пометить какой-либо пункт включенным, применяется атрибут checked.

Для переключателей и флажков используется атрибут value, который показывает, какое значение будет отправлено на сервер (какой пункт был выбран пользователем).

<label>Выберите Ваш пол</label>

<label>

<input type="radio" name="gender" value="men" checked> Мужской

</label>

<label>

<input type="radio" name="gender" value="women"> Женский

</label>

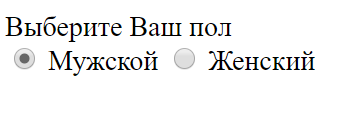


Рис. 7 Переключатели

В данном примере оба переключателя используют одинаковое значение атрибута name (значение gender), выбор одного из вариантов исключит выбор другого варианта. При выборе пункта *Мужской* на сервер будет отправлено значение men, при выборе пункта *Женский* – значение women.

1. ***Кнопки***

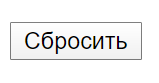
Кнопку на веб-странице можно создать двумя способами – с помощью тега **<input>** и парного тега **<button></button>**.

При добавлении кнопки через <input> возможно создание следующих типов кнопок:

1. кнопка отправки данных на сервер: **<input type="submit">**;



1. кнопка сброса введенных данных: **<input type="reset">**;



в) обычная кнопка, которая не имеет поведения по умолчанию: **<input type="button">**.



Такую кнопку можно использовать для запуска сценария JavaScript.

Второй способ создания кнопки основан на использовании контейнера **<button></button>**. Для корректного отображения элемента <button> разными браузерами у него нужно указывать атрибут type, например:

<button type="submit"> </button>

В следующем примере показан html-код создания различных кнопок.

<input type="submit" name = "button1" autofocus> <br>

<input type="reset" name = "button2"> <br>

<input type="image" name = "button3" src="knp.png"> <br>

<input type="button" name = "button4" value="Кнопка отключена" disabled> <br><br>

<button type = "button" name = "button5">

<img src = "photo.png">

<b>Нажми меня</b>

</button>

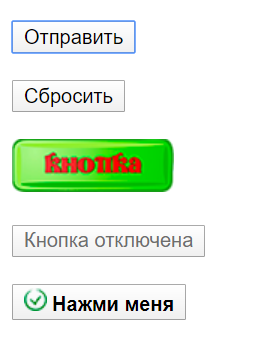


Рис. 8 Кнопки

1. ***Поле загрузки файла***

Чтобы загружать на сервер один или несколько файлов используется специальное поле: **<input type="file">.**

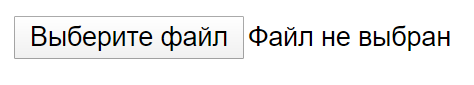


Рис. 9 Вид поля для загрузки файла в Chrome

При нажатии на кнопку открывается окно для выбора файла, где можно указать, какой файл пользователь желает использовать.

Чтобы поле заработало и браузер смог предать выбранный файл на сервер, необходимо для тега формы:

1. задать метод отправки данных POST (method="post");
2. установить у атрибута enctype значение multipart/form-data.

Далее рассмотрим другие теги для создания полей формы.

1. ***Выпадающий список***

Выпадающий список используется, если количество вариантов для выбора достаточно много. Выпадающий список работает подобно переключателям, отличаясь от них только внешним видом.

Для создания выпадающего списка используется контейнер **<selest></select>**. Элементы выпадающего списка указываются в контейнерах <option></option>.

<select>

<option> Декабрь </option>

<option> Январь </option>

<option> Февраль </option>

</select>

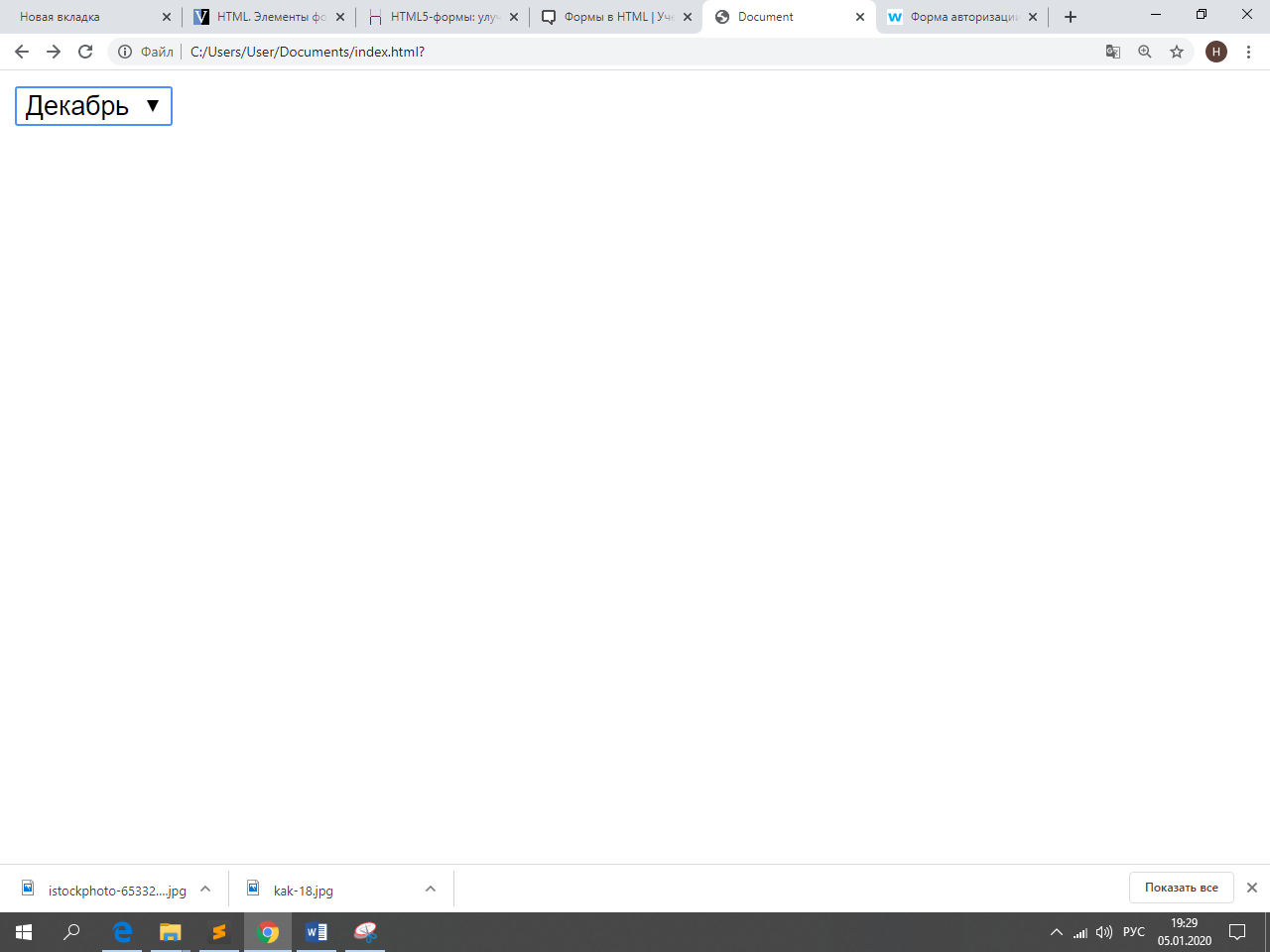
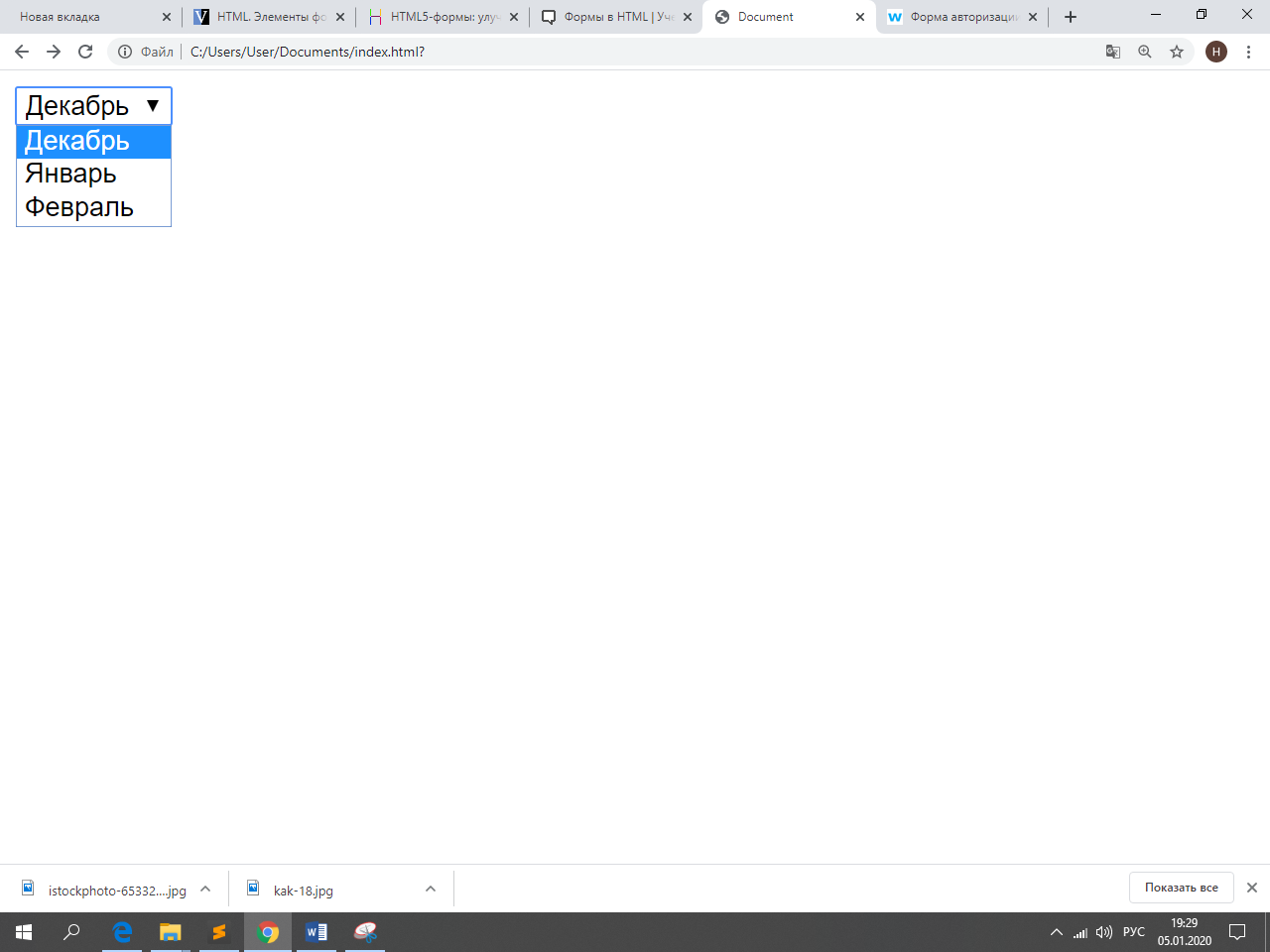
 

Рис. 10 Выпадающий список

По умолчанию в поле списка отображается его первый элемент. Атрибуты тегов <select> и <option> приведены в таблицах 6 и 7.

Таблица 6. Атрибуты тега <select>

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | multiple | Дает возможность выбора нескольких пунктов. При выборе нужно нажать и удерживать нажатой клавишу Ctrl. Атрибут используется без указания значения. |
|  | size | Задает количество отображаемых строк списка. Значение атрибута задается целым положительным числом. |

Таблица 7. Атрибуты тега <option>

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | selected | Отображает выбранный элемент списка по умолчанию при загрузке веб-страницы браузером. Атрибут используется без указания значения. |
|  | value | Указывает значение, отправляемое на сервер |

Пример использования тега <select>:

<label>Какими браузерами вы пользуетесь? </label>

<select multiple size = "3">

<option value ="explorer"> Internet Explorer </option>

<option value ="chrome" selected> Google Chrome </option>

<option value ="firefox"> Mozilla Firefox </option>

<option value ="opera"> Opera </option>

<option value ="safari"> Safari </option>

</select>

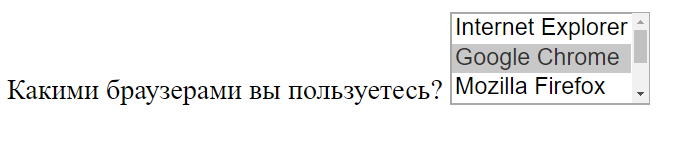


Рис. 11 Выпадающий список множественного выбора

1. ***Текстовая область***

Если необходимо создать многострочное поле для ввода большого текста, то вместо тега <input> используется элемент **<textarea></textarea>.**

Таблица 8. Атрибуты тега <textarea>

|  |  |  |
| --- | --- | --- |
| *№* | *Атрибут* | *Описание, принимаемое значение* |
|  | cols | Ширина текстового поля в символах. |
|  | rows | Высота текстового поля в строках текста. |
|  | wrap | Определяет параметры переноса строк в тексте, при отправке данных на сервер:   * wrap = "hard" – сохраняет перенос, обязательным условием использования значения hard является установленный атрибут cols; * wrap = "soft" – не сохраняет перенос (значение по умолчанию). |
|  | maxlength | Максимальное число символов, которое можно ввести в текстовое поле. |
|  | readonly | Указывает, что текстовое поле будет доступно только для чтения, т.е. текст невозможно будет изменить, но будет возможность его скопировать. Атрибут используется без указания значения. |
|  | disabled | Отключает возможность редактирования и копирования содержимого поля. Атрибут используется без указания значения. |