



Лекция 3

Взаимодействие с пользователем

Цели на сегодня:


- Разобраться с тегами `<input>`, `<textarea>`, `<label>` и `<select>`;
- Узнать о кнопках и их отличии от ссылок;
- Научиться объединять элементы в формы и работать с сервером.

Элементы ввода <input>

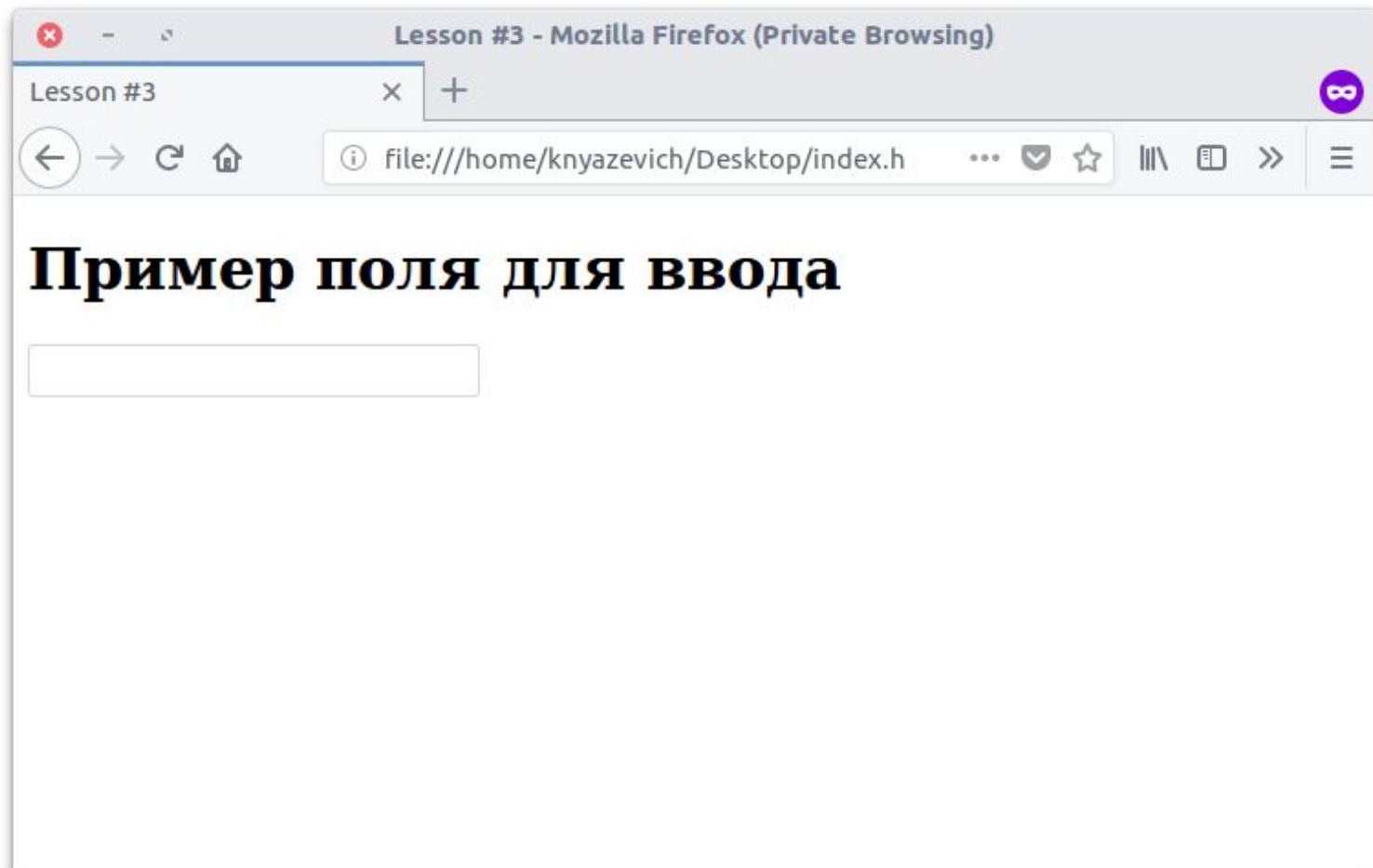
Элементы взаимодействия <input>

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

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



```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Lesson #3</title>
</head>
<body>
  <h1>Пример поля для ввода</h1>
  <input type="text">
</body>
</html>
```



Элементы взаимодействия <input>

Атрибут type					
text	checkbox	submit	image	month	url
password	hidden	reset	color	number	week
radio	button	file	date	range	
email	tel	time	datetime-local	search	

Элементы взаимодействия <input>

Другие важные атрибуты	
required	Делает поле обязательным для ввода, пользователь не отправит пустую форму
name	Задаёт имя полю (для якорей и сервера)
value	Задаёт полю значение, которое и отправляется на сервер
placeholder	Подсказка для пользователя
accept	Если используется <i>type=file</i> , то тут можно указать тип файла, который можно передать

Атрибут disabled

Атрибут не принимает значений и позволяет отключить поле. В таком случае поле станет серым и пользователь не сможет с ним взаимодействовать.

Атрибут checked

Атрибут не принимает значений и используется в чекбоксах и радиокнопках для указания активного состояния.

Атрибут pattern

Указывает регулярное выражение, согласно которому требуется вводить и проверять данные в поле формы. Если присутствует атрибут pattern, то форма не будет отправляться, пока поле не будет заполнено правильно.

Подписи <label>

Тег <label>

Создает связь между элементом `<label>` и элементом формы (`<input>`, `<select>`, `<textarea>`).

Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст или другой элемент.

Кроме того, с помощью `<label>` можно устанавливать горячие клавиши на клавиатуре и переходить на активный элемент подобно ссылкам.



```
<h1>Разные типы полей ввода</h1>
```

```
<div>
```

```
  <label for="email-field">Ваш Email</label>
```

```
  <input type="email" id="email-field">
```

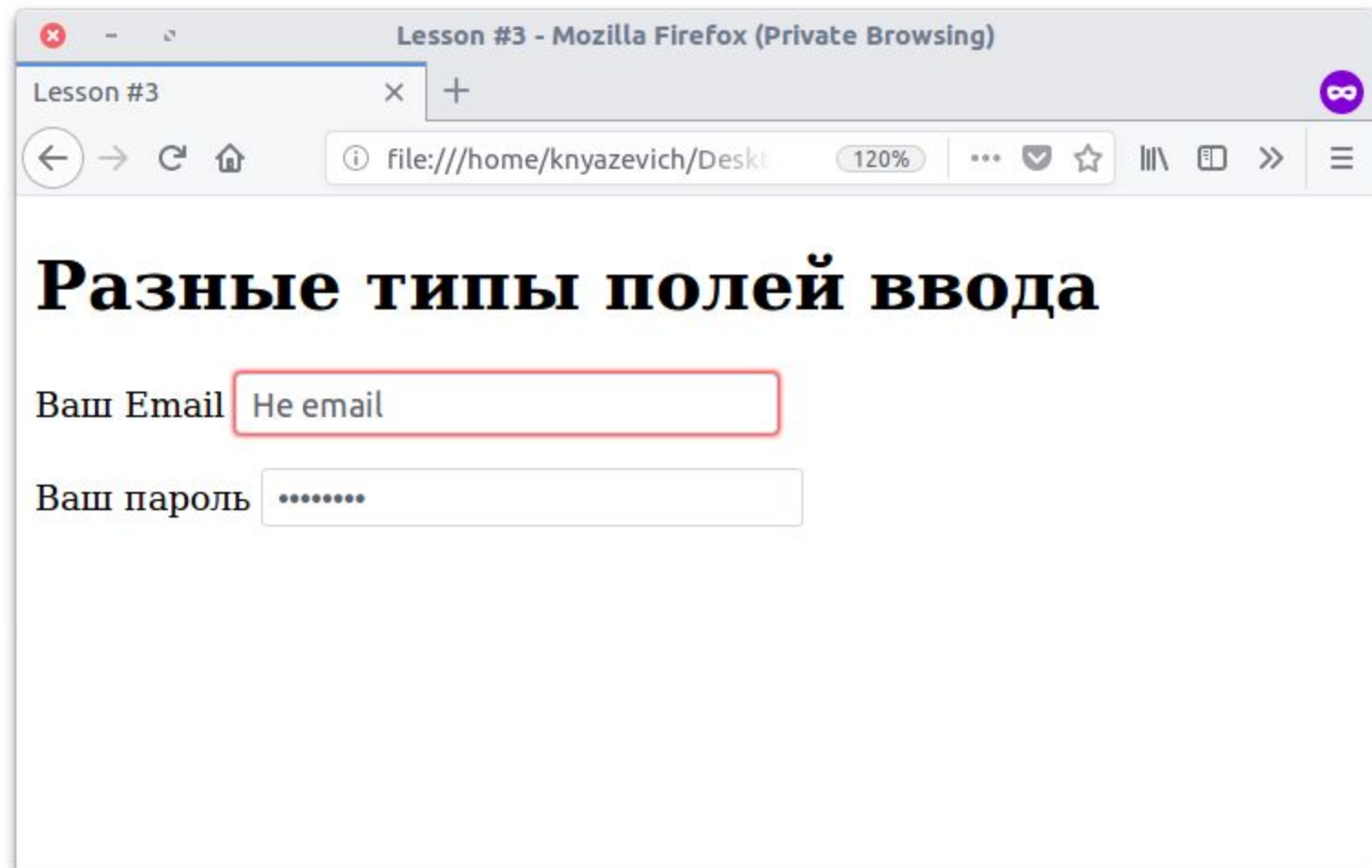
```
</div>
```

```
<div>
```

```
  <label for="password-field">Ваш пароль</label>
```

```
  <input type="password" id="password-field">
```

```
</div>
```



Как объединить радиокнопки?

Как объединить радиокнопки?

Объединение радиокнопок в группу является рутинной задачей. Это нужно для того, чтобы пользователь мог выбрать только один из возможных вариантов.

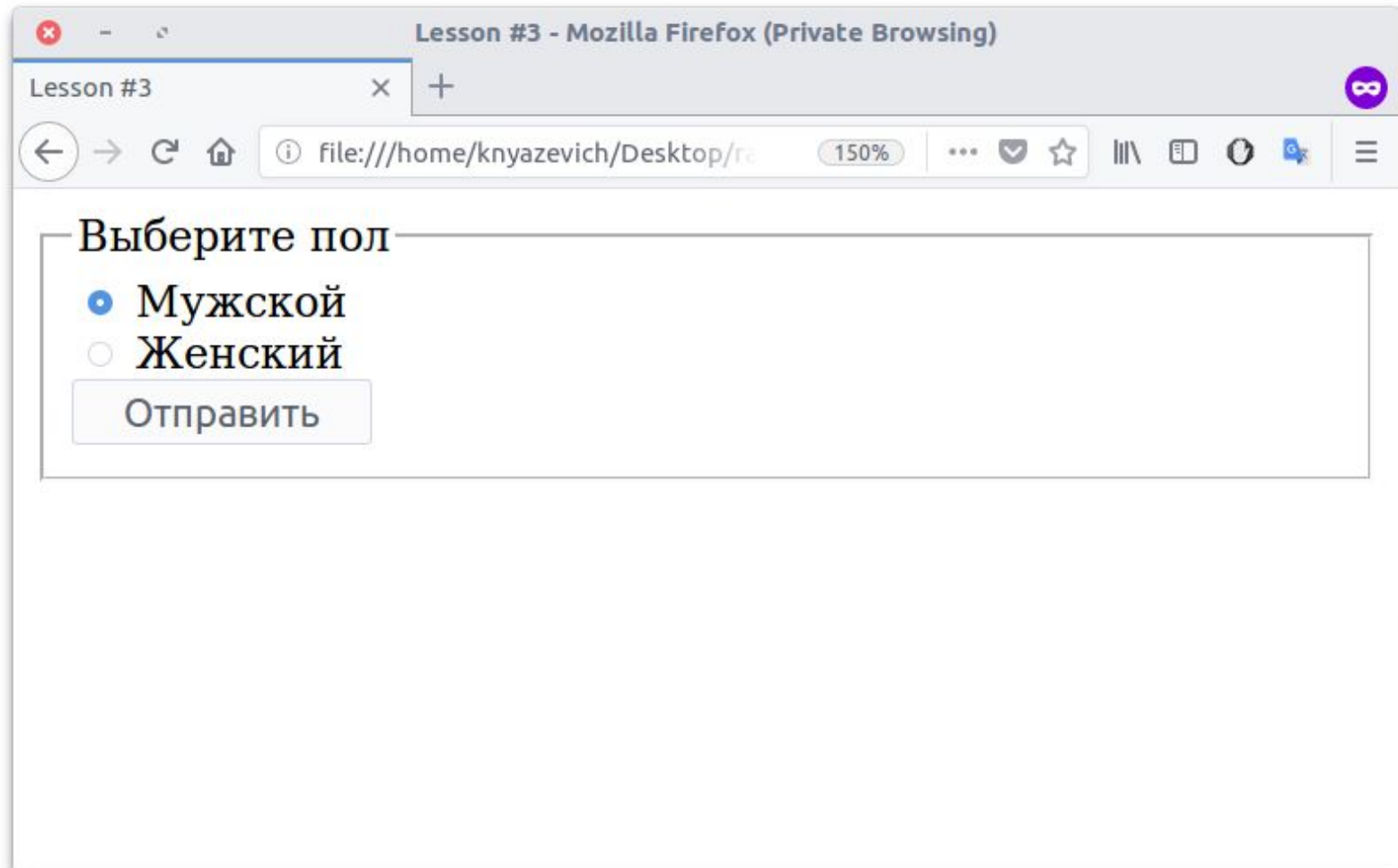
У всех радиокнопок в группе должен быть одинаковый атрибут **name**. Для радиокнопок так же важно указание атрибута **value**, поскольку именно это значение и отправляется на сервер.


```
<form action="https://maximumstart-echo.herokuapp.com/login" method="GET">
  <fieldset>
    <legend>Выберите пол</legend>

    <input type="radio" id="male-radio" name="gender" value="male">
    <label for="male-radio">Мужской</label><br>

    <input type="radio" id="female-radio" name="gender" value="female">
    <label for="female-radio">Женский</label><br>

    <input type="submit" value="Отправить">
  </fieldset>
</form>
```



Выпадающий список <select>

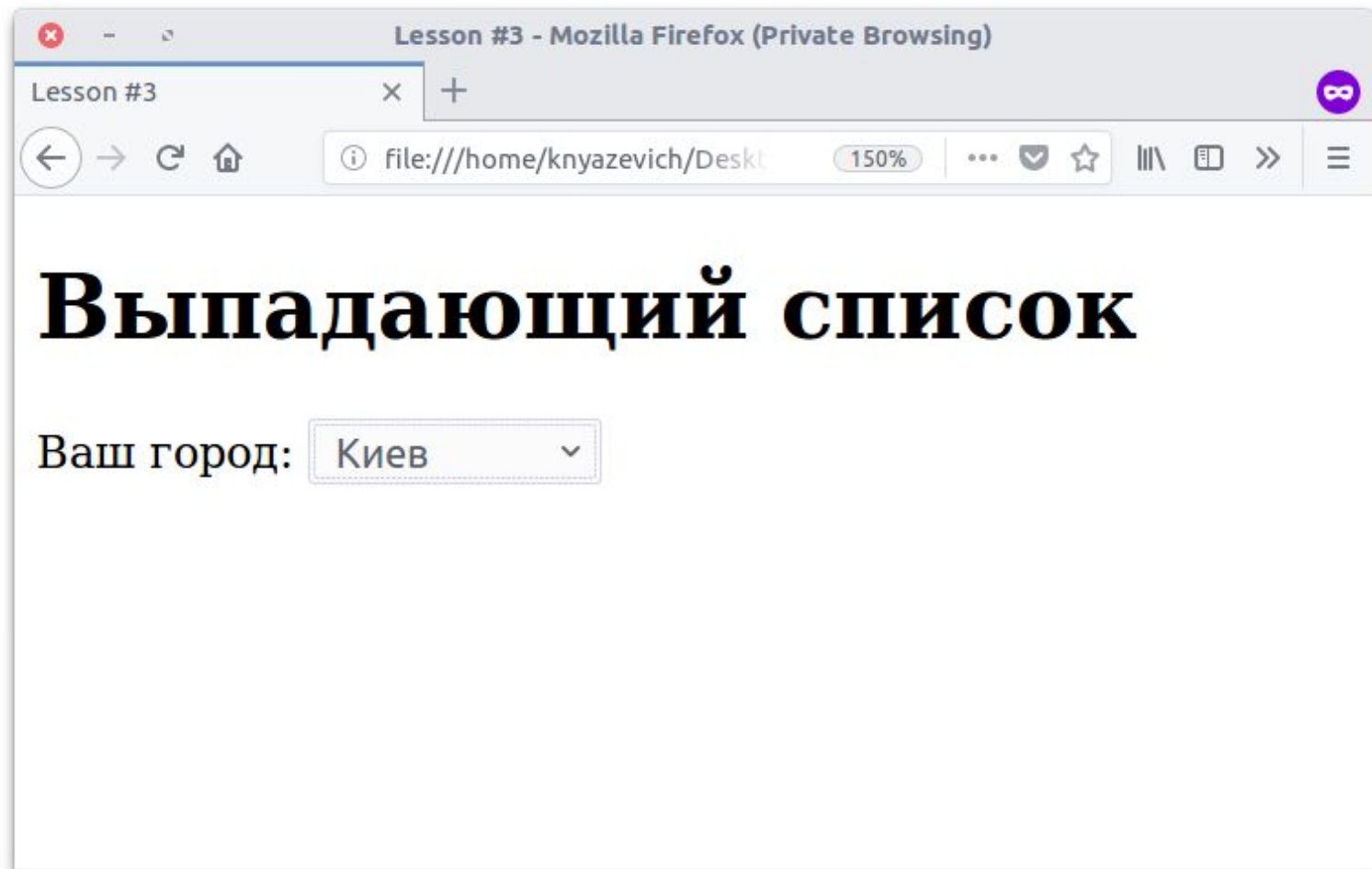
Тег <select>

Тег `<select>` является контейнером для элементов `<option>`.
Каждый `<option>` это один элемент выпадающего списка.

Если для `<option>` указать атрибут `value`, то на сервер отправится это значение, если нет, то отправится значение между тегами `<option>...</option>`.



```
<h1>Выпадающий список</h1>
<div>
  <label for="city-field">Ваши любимые города:</label>
  <select id="city-field">
    <option value="1">Киев</option>
    <option value="2">Днепр</option>
    <option value="3">Запорожье</option>
    <option value="4">Львов</option>
    <option value="5">Харьков</option>
  </select>
</div>
```




Атрибут **value**

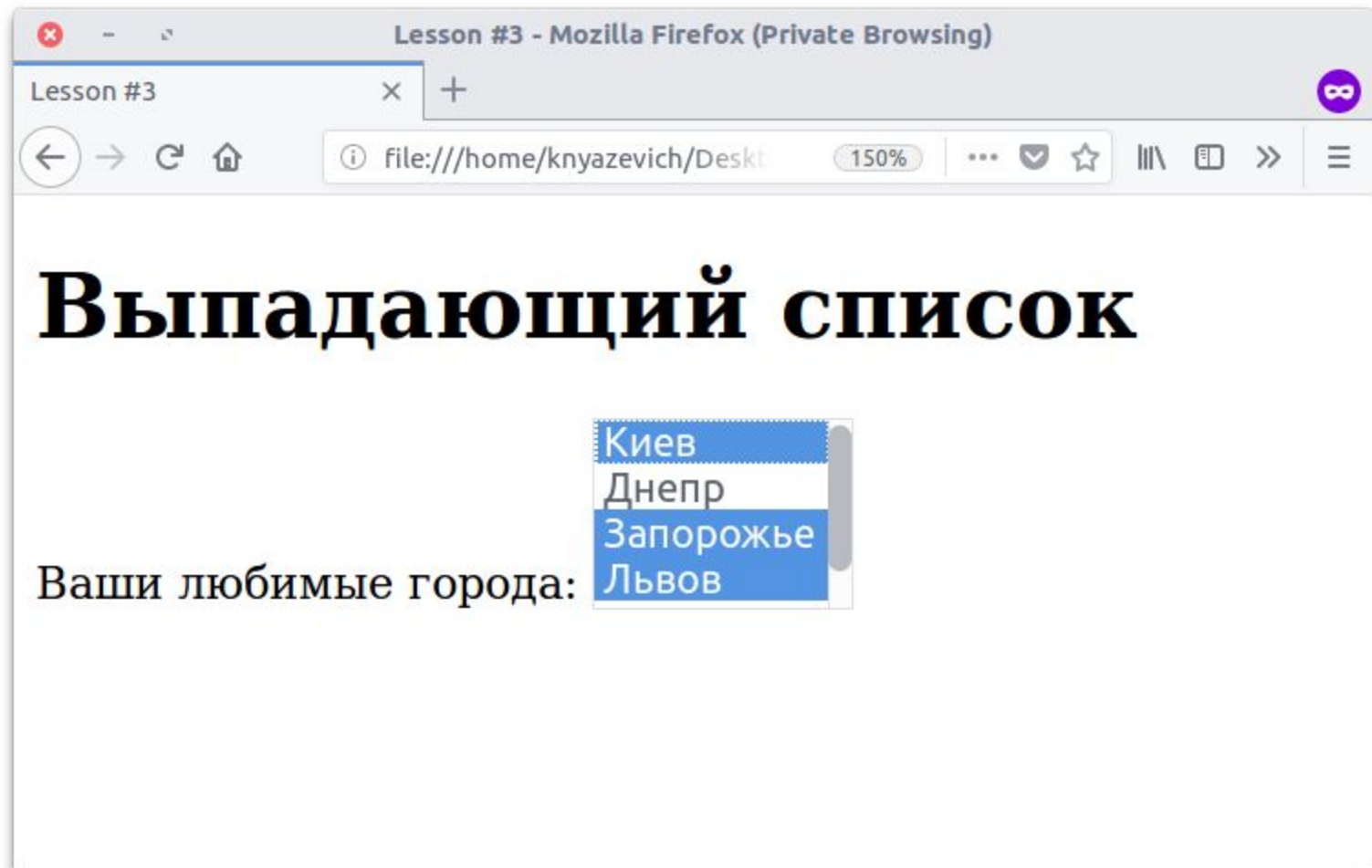
Этот атрибут необязателен. Если у элемента `<option>` указать **value**, то на сервер будет отправляться это значение. Если атрибут не указан, то на сервер отправится содержимое тега.

Атрибут **multiple**

Этот логический атрибут указывает что возможен выбор нескольких опций в списке. Если данный атрибут не указан, то можно выбрать только одну опцию.



```
<h1>Выпадающий список</h1>
<div>
  <label for="city-field">Ваши любимые города:</label>
  <select id="city-field" multiple>
    <option>Киев</option>
    <option>Днепр</option>
    <option>Запорожье</option>
    <option>Львов</option>
    <option>Харьков</option>
  </select>
</div>
```





Кнопки <button>

Нужно грамотно подходить к выбору: “Кнопка или ссылка?”.

Кнопки <button>

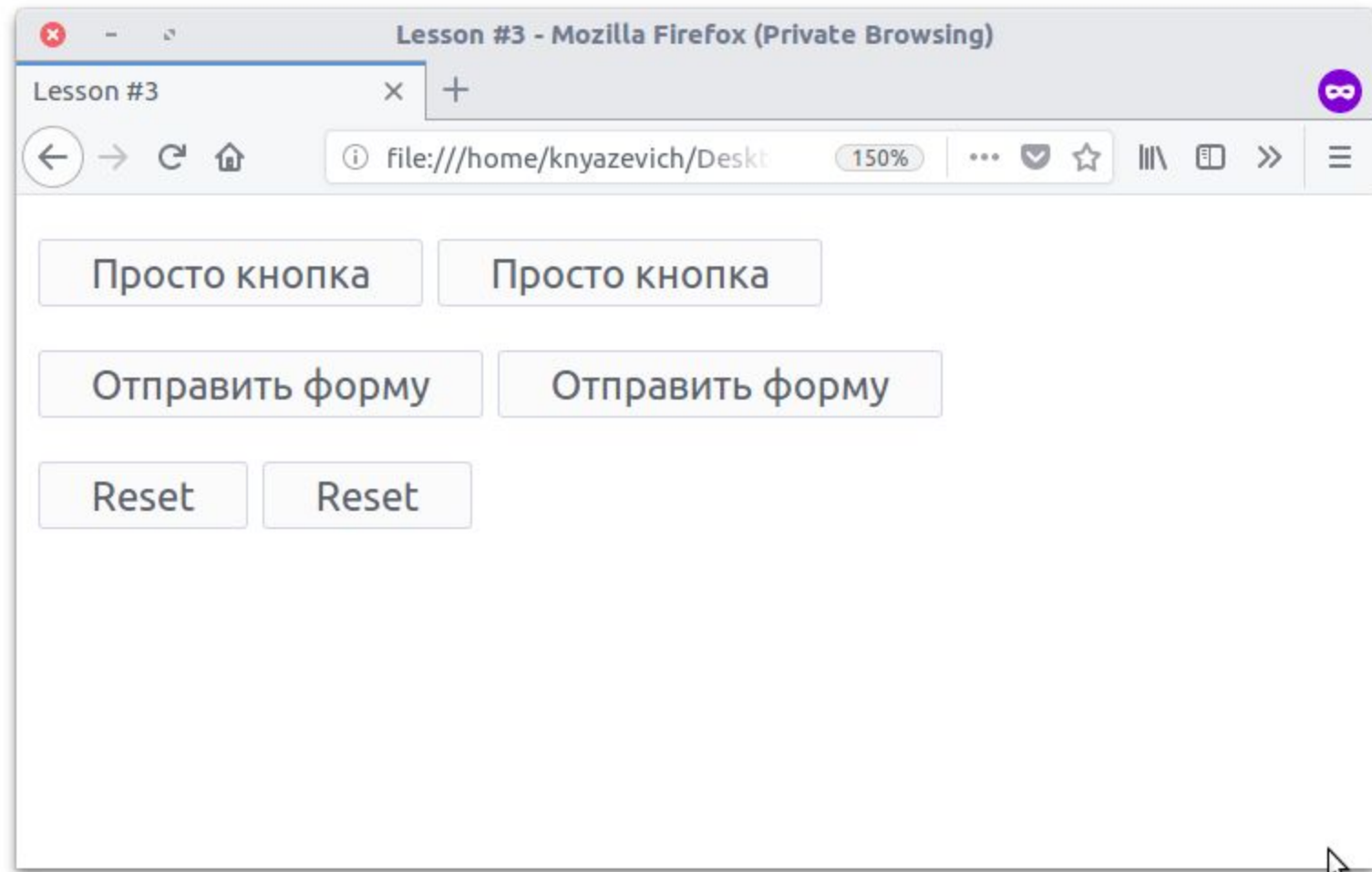
Используются для работы со скриптами и взаимодействия с сервером (отправки форм, сброса форм).



```
<div>
  <!-- Кнопки, которые ничего не делают -->
  <button>Просто кнопка</button>
  <button type="button">Просто кнопка</button>
</div>
```

```
<div>
  <!-- Кнопки, которые отправляют форму -->
  <input type="submit" value="Отправить форму">
  <button type="submit">Отправить форму</button>
</div>
```

```
<div>
  <!-- Кнопки, которые сбрасывают форму -->
  <input type="reset">
  <button type="reset">Reset</button>
</div>
```



Текстовое поле <textarea>

Текстовое поле <textarea>

Поле <textarea> представляет собой элемент формы для создания области, в которую можно вводить **несколько строк текста**. В отличие от тега <input> в текстовом поле допустимо делать переносы строк и они сохраняются при отправке данных на сервер.

Между тегами <textarea> и </textarea> можно поместить любой текст, который будет отображаться внутри поля.



```
<label>Введите сообщение  
    <textarea cols="50" rows="10"></textarea>  
</label>
```




Протокол HTTP

Сетевой протокол – набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включенными в сеть устройствами.

По факту, это набор инструкций который знает каждая машина в сети и с помощью которого происходит обмен данными между ними.

HTTP – HyperText Transfer
Protocol (протокол передачи
гипертекста)

HTTP – широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов.

Различают протоколы высокого и низкого уровня. HTTP/HTTPS относится к высокоуровневым протоколам.

Получить доступ к HTTP запросам
можно в инструментах
разработчика

Структура HTTP запроса

Рассмотрим структуру на примере одного HTTP запроса.

```
1 GET /other-19 HTTP/1.1
2 Host: www.scriptsite.ru
3 User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; ru; rv:1.9.1.5) Gecko/20091102
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: ru,en-us;q=0.7,en;q=0.3
6 Accept-Encoding: gzip,deflate
7 Accept-Charset: windows-1251,utf-8;q=0.7,*;q=0.7
8 Keep-Alive: 300
9 Connection: keep-alive
```

- Метод
- Ресурс
- Протокол
- HTTP заголовки в формате «Ключ: значение»

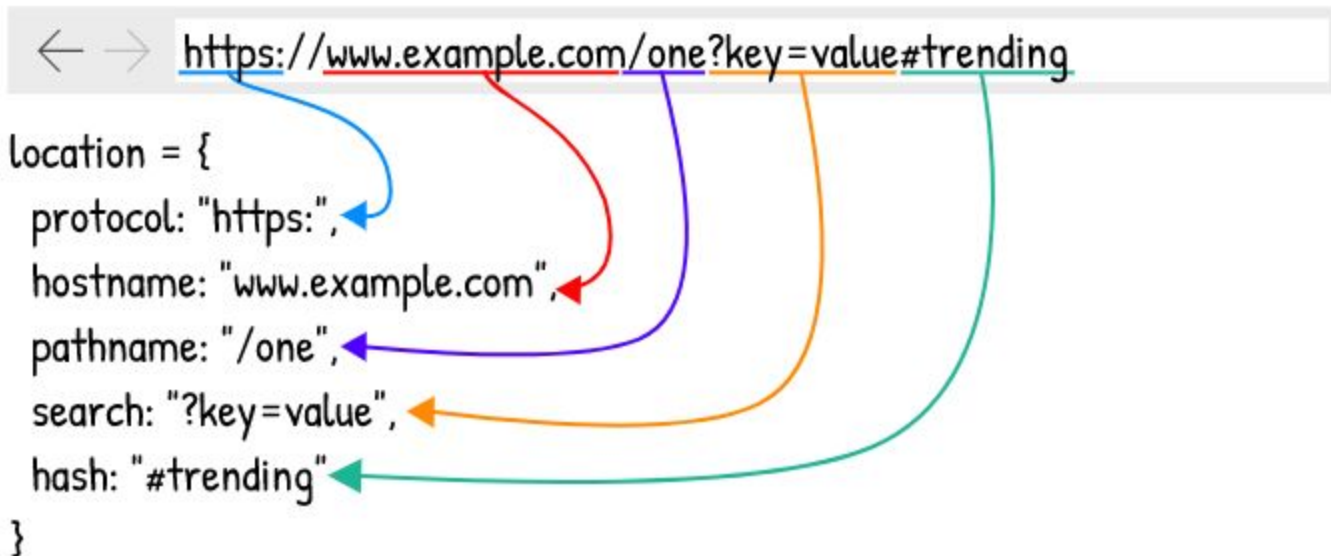
Метод (*method*) характеризует тип запроса на ресурс.

Существует множество методов, самые основные из них это:
POST, GET, PUT, DELETE.

Ресурс	POST создать	GET прочитать	PUT обновить	DELETE удалить
/post?id=1	создает новый ресурс	возвращает ресурс	обновляет указанный ресурс	удаляет указанный ресурс

← → https://www.example.com/one?key=value#trending

```
location = {  
  protocol: "https:",  
  hostname: "www.example.com",  
  pathname: "/one",  
  search: "?key=value",  
  hash: "#trending"  
}
```



Код состояния HTTP (англ. *HTTP status code*) — часть первой строки ответа сервера при запросах по протоколу HTTP. Это целое число из трех десятичных цифр. Первая цифра указывает на **класс состояния**.

- 1xx: Informational (информационные);
- 2xx: Success (успешно);
- 3xx: Redirection (перенаправление);
- 4xx: Client Error (ошибка клиента);
- 5xx: Server Error (ошибка сервера);

Формы <form>

Формы <form>

Тег <form> устанавливает форму на веб-странице.

Форма предназначена для обмена данными между пользователем и сервером.

Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Формы <form>

Важные атрибуты	
<code>action</code>	Адрес программы или документа, который обрабатывает данные формы.
<code>method</code>	HTTP метод, который браузер использует, для отправки формы (GET, POST...).

```
<form action="https://maximumstart-echo.herokuapp.com" method="GET">
  <fieldset>
    <legend>Войти на сайт</legend>
    <div>
      <label for="email-field">Email</label>
      <input type="email" id="email-field" name="email" required>
    </div><br>
    <div>
      <label for="password-field">Пароль</label>
      <input type="password" id="password-field" name="password" required>
    </div><br>
    <input type="submit" value="Войти">
  </fieldset>
</form>
```

Как отправить файл на сервер?



```
<form action="http://localhost:3000" method="POST" enctype="multipart/form-data">  
  <input type="file" name="document">  
  <input type="submit" value="Отправить">  
</form>
```

Самостоятельная работа

Разметить формы по образцам ниже.

Регистрация нового пользователя	
Email:	<input type="text" value="test@example.com"/>
Фамилия:	<input type="text"/>
Имя:	<input type="text"/>
Отчество:	<input type="text"/>
Пол:	<input type="radio"/> Мужской <input type="radio"/> Женский
Дата рождения:	<input type="text" value="mm / dd / yyyy"/>
Логин:	<input type="text"/>
Пароль:	<input type="password"/>
Повторите пароль:	<input type="password"/>

Pizza Shop 2.0	
Name	<input type="text"/>
Pizza Topping	<input type="radio"/> Supreme <input type="radio"/> Vegetarian <input type="radio"/> Hawaiian
Pizza Sauce	<input type="text" value="Tomato"/>
Optional Extras	<input type="checkbox"/> Extra Cheese <input type="checkbox"/> Gluten Free Base
Delivery Instructions: <div><div></div></div>	
<input type="button" value="Send my Order"/>	

Регистрация

Разобранные требования

- Запрещено стилизовать элементы при помощи атрибута `style`;
- Для построения сетки запрещено использовать таблицы;
- Должен присутствовать `favicon`.