**Теория по работе с HTML формами**

# Тег form

Тег **form** создает форму для отправки данных на сервер, где эти данные будут обработаны с помощью языка **PHP**.

Также данные формы могут быть обработаны с помощью языка **JavaScript**.

Тег **form** служит контейнером для тегов input, textarea, select, button, fieldset. Указанные теги **не обязательно** следует вложить в тег **form**, но, если этого не сделать, **не произойдет отправки** формы на сервер.

Отправка формы на сервер осуществляется по нажатию на кнопку, создаваемую тегом input с атрибутом **type** в значении **submit**, или на кнопку button.

**Атрибуты**

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| action | Куда отправляем форму, на какую страницу сайта (своего или чужого). Если оставить пустым - форма отправится на текущую страницу. **В HTML5 необязательный атрибут.** В атрибуте можно указывать ссылку на определенное место страницы (через #, как в теге a). В этом случае после отправки формы браузер откроет страницу, заданную в атрибуте action, на указанном месте. |
| method | Метод отправки формы, может быть **POST** или **GET**. **Необязательный атрибут.** По умолчанию атрибут имеет значение **GET**. |
| target | Указывает ссылке в какой вкладке ей открываться: в новой, в текущей или во фрейме. **Необязательный атрибут.** |
| enctype | Определяет способ кодирования данных формы. **Необязательный атрибут.** |

**Значения атрибута target**

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| \_blank | Страница откроется в **новой** вкладке браузера. |
| \_self | Страница откроется в **текущей** вкладке браузера. |
| \_parent | Страница откроется во **фрейме-родителе.** |
| \_top | Отменяет все фреймы и загружает страницу в полном окне браузера. |

Значение по умолчанию: **\_self**.

**Пример формы**

<form action="" method="GET">

<input type="text"><br><br>

<input type="checkbox">

<input type="checkbox" checked>

<input type="radio" name="radio">

<input type="radio" name="radio">

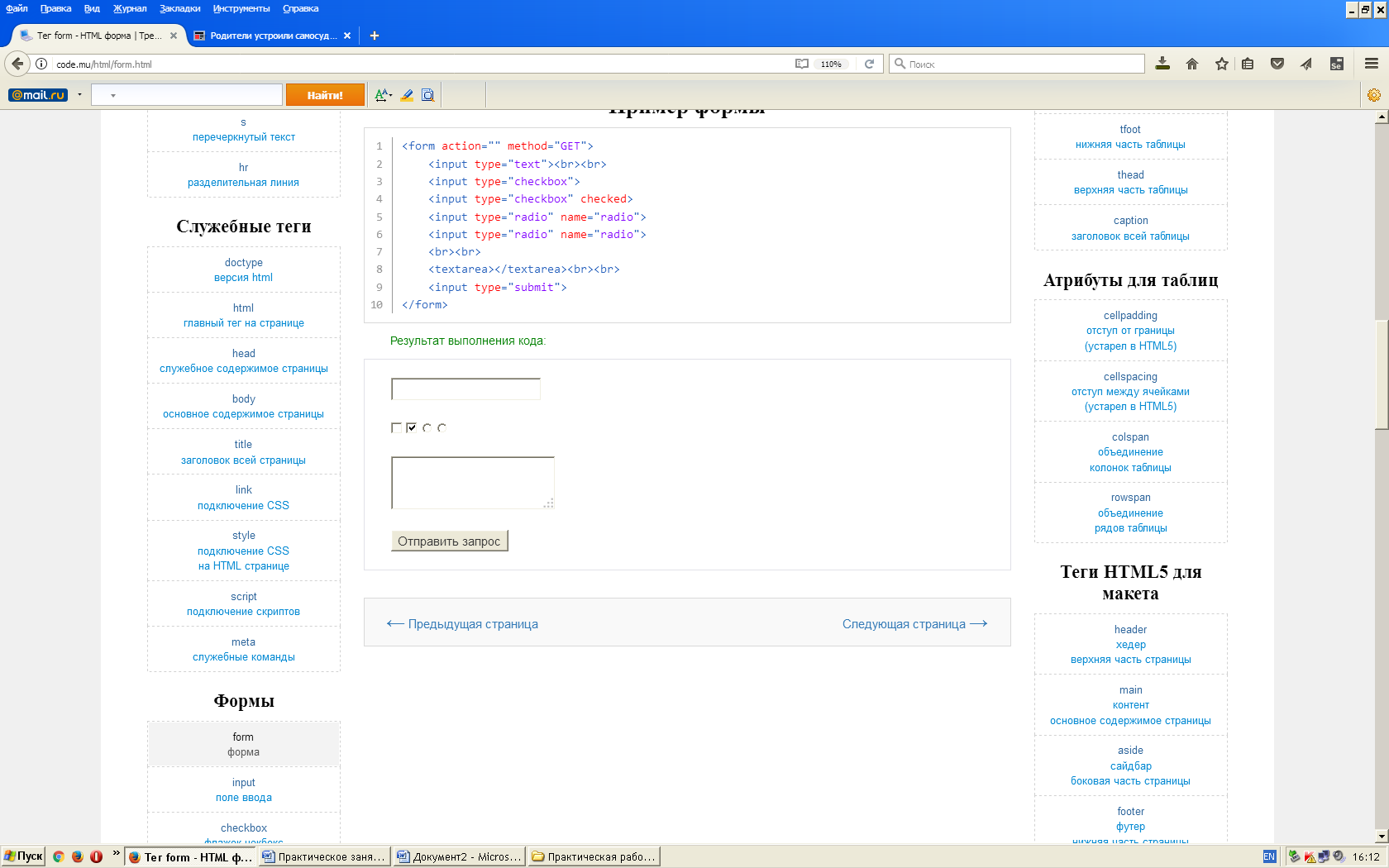
<br><br>

<textarea></textarea><br><br>

<input type="submit">

</form>

Результат выполнения кода:

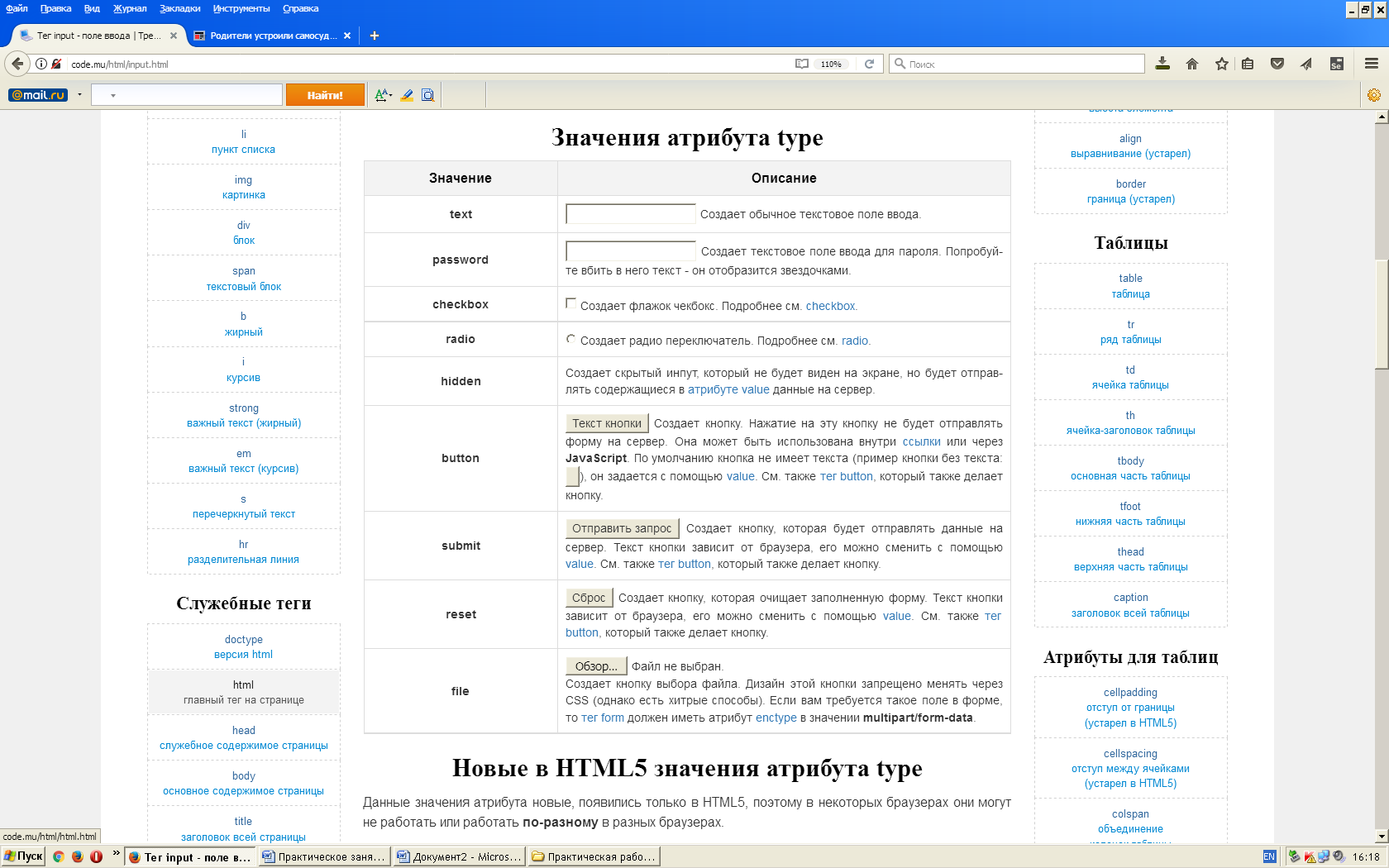


Тег **input** создает различные элементы HTML формы: обычное поле ввода, поле для ввода пароля, флажок checkbox (чекбокс), радио кнопочки (radio), кнопку.

Тег **input** должен лежать внутри тега form. Это **не обязательно**, необходимо для отправки данных на сервер и последующей их обработки через PHP. **Не требует закрывающего тега.**

**Атрибуты**

**Значения атрибута type**

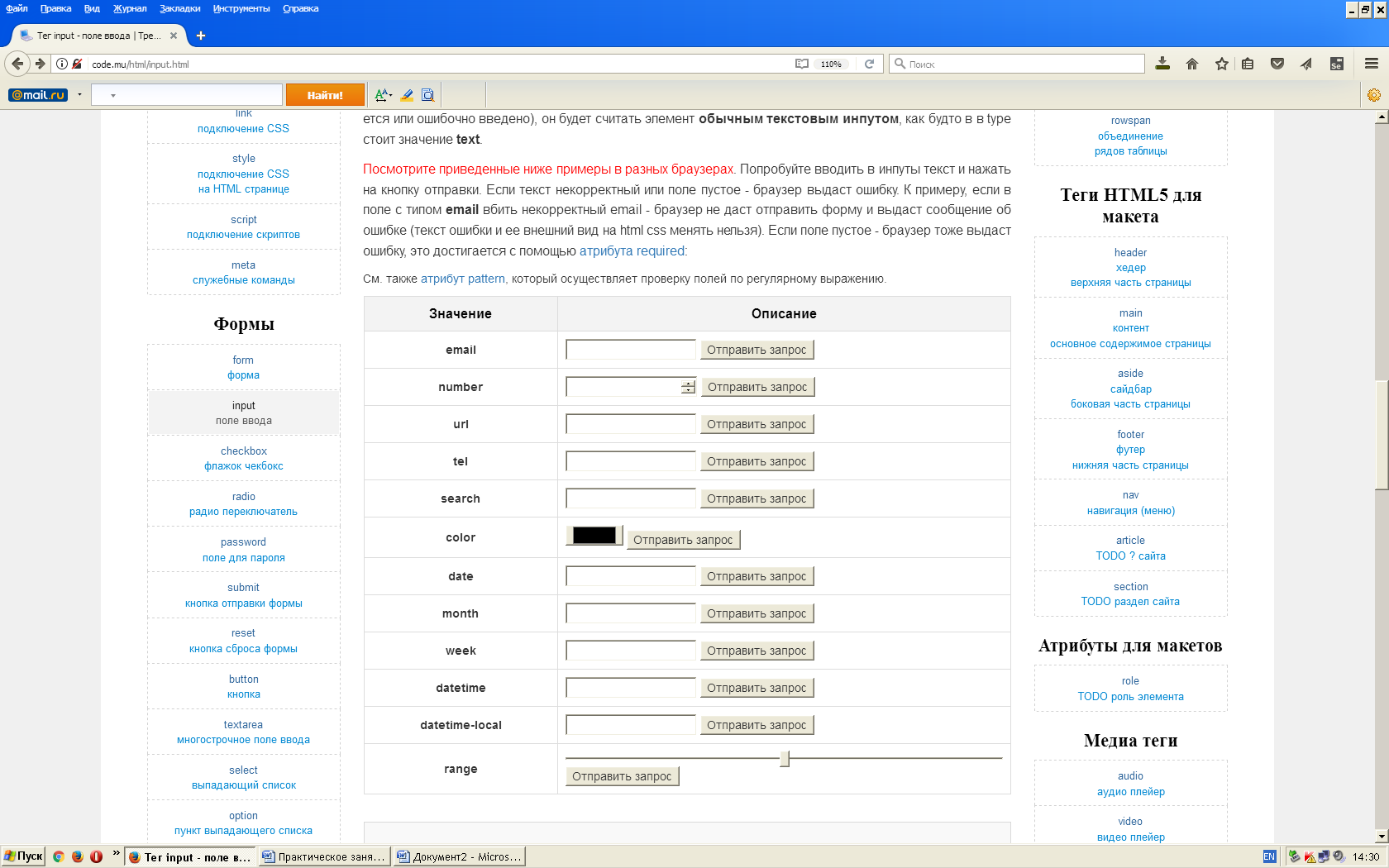
****

**Новые в HTML5 значения атрибута type**

Данные значения атрибута новые, появились только в HTML5, поэтому в некоторых браузерах они могут не работать или работать **по-разному** в разных браузерах.

В случае, если браузер **не может понять** содержимое атрибута **type** (например, если оно не поддерживается или ошибочно введено), он будет считать элемент **обычным текстовым инпутом**, как будто в в type стоит значение **text**.

Посмотрите приведенные ниже примеры в разных браузерах. Попробуйте вводить в инпуты текст и нажать на кнопку отправки. Если текст некорректный или поле пустое - браузер выдаст ошибку. К примеру, если в поле с типом **email** вбить некорректный email - браузер не даст отправить форму и выдаст сообщение об ошибке (текст ошибки и ее внешний вид на html css менять нельзя). Если поле пустое - браузер тоже выдаст ошибку, это достигается с помощью атрибута required:



# Флажок checkbox

Флажок **checkbox** представляет собой элемент (галочку или птичку) HTML формы, который может иметь два состояния: отмечен и не отмечен.

Флажок создается с помощью тега input с атрибутом **type** в значении **checkbox** (понажимайте на него мышкой - вы увидите как он меняет свое состояние с отмеченного на неотмеченное и наоборот):

<input type="checkbox">

Результат выполнения кода:



Отмеченный чекбокс будет посылать на сервер содержимое атрибута value. Если атрибута **value** у чекбокса нет - он будет посылать строку "on". Если чекбокс не отмечен - ничего не будет присылаться на сервер не зависимо от наличия атрибута **value**.

**Пример**

В данном примере сделаны два флажка checkbox: первый будет отмечен (так как ему дан атрибут checked), а второй - нет:

<input type="checkbox" checked> <input type="checkbox">

Результат выполнения кода:



**Пример . Заблокированный чекбокс**

В данном примере демонстрируется заблокированный с помощью атрибута disabled флажок. При нажатии на него его состояние не будет менятся:

<input type="checkbox" disabled>

Результат выполнения кода:



**Пример . Заблокированный и отмеченный чекбокс**

В данном примере флажок заблокирован атрибутом **disabled** и при этом находится в отмеченном состоянии, так как ему задан атрибут **checked**:

<input type="checkbox" disabled checked>

Результат выполнения кода:



# Радио переключатель

**Радио переключатель** представляет собой элемент HTML формы в виде круглой кнопочки, которая может быть отмечена и нет.

Радио кнопочка создается с помощью тега input с атрибутом **type** в значении **radio**:

<input type="radio">

Результат выполнения кода:



**Один** радио переключатель практически никогда **не используется**, их следует использовать группой. При этом в группе может быть отмечен **только один** из переключателей. Переключатели будут формировать группу только если у них **одинаковое** значение атрибута name (см. примеры ниже).

**Пример 1**

В данном примере сделаны два **радио переключателя**: первый отмечен (так как ему дан атрибут checked), а второй - нет. При этом переключатели представляют собой **группу**, в которой может быть отмечен только один из них (так как у них одинаковое значение атрибута **name**).

Обратите внимание на то, что данным радио кнопочкам даны разные значения атрибута value. Это нужно для того, чтобы после отправки данных на сервер, PHP мог определить, какой из переключателей был выбран:

<input type="radio" name="test" checked value="1"> <input type="radio" name="test" value="2">

Результат выполнения кода:



**Пример 2**

В данном примере сделаны две группы **радио переключателя**: первая группа имеет одно значение атрибута **name** (radio1), а вторая - другое (radio2). Попробуйте понажимать по ним, вы увидите, что группы переключаются независимо друг от друга:

<input type="radio" name="radio1" checked>

<input type="radio" name="radio1">

<input type="radio" name="radio1"><br>

<input type="radio" name="radio2" checked>

<input type="radio" name="radio2">

<input type="radio" name="radio2">

Результат выполнения кода:


# Поле для ввода пароля

Поле для ввода **пароля** представляет собой инпут с атрибутом **type** в значении **password**. Такое поле отображает текст звездочками, чтобы его не могли подсмотреть через плечо и украсть пароль пользователя.

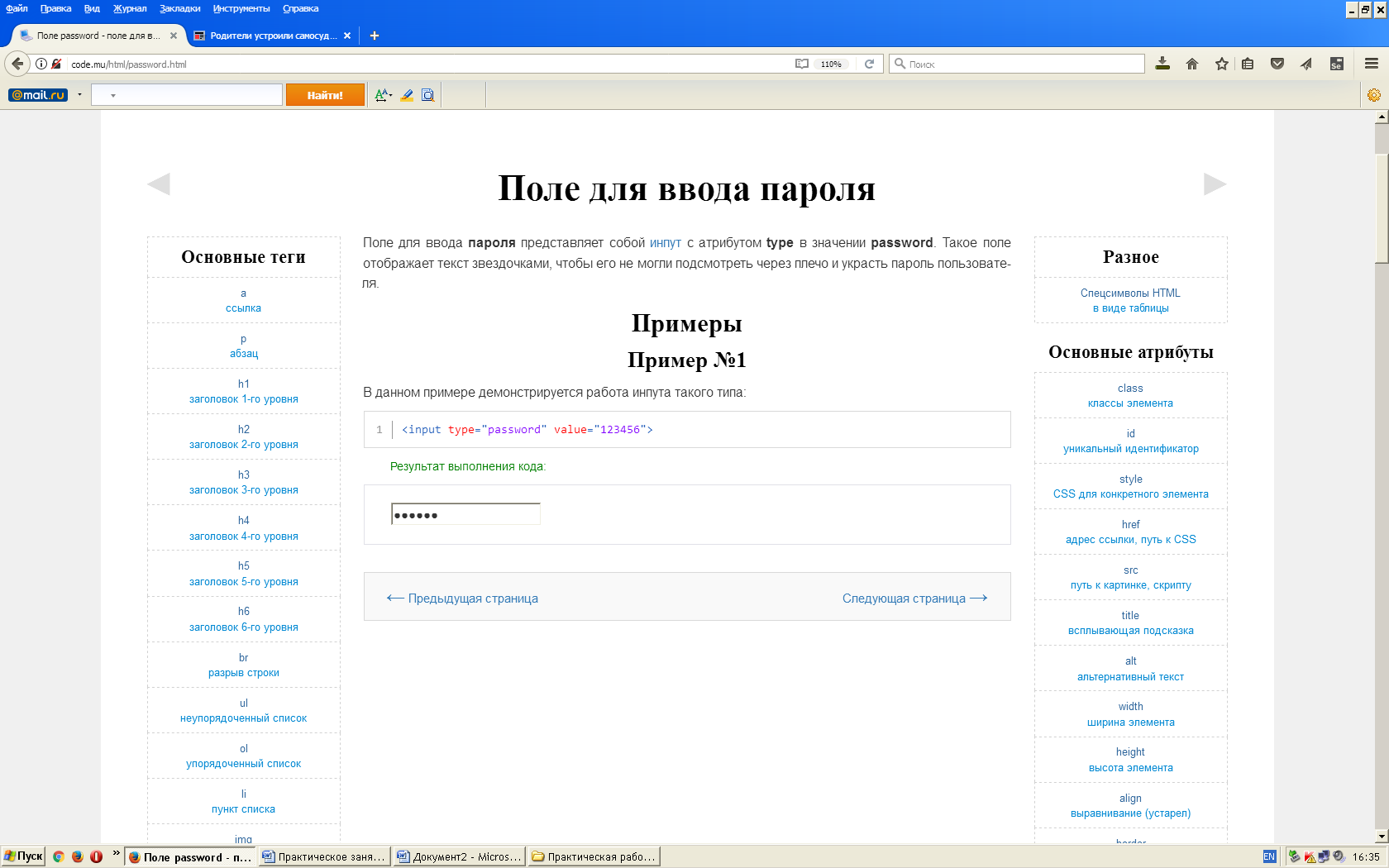
## Примеры

### Пример

В данном примере демонстрируется работа инпута такого типа:

<input type="password" value="123456">

Результат выполнения кода:



# Кнопка отправки формы

Тег **input** с атрибутом **type** в значении **submit** создает кнопку отправки HTML формы, а также просто как элемент дизайна (в этом случае ее часто вкладывают в HTML ссылку).

**Кнопку отправки формы** также можно создать с помощью тега button.

Если атрибуту **type** тега input дать значение **button** - то появится кнопка, которая не будет отправлять форму, но будет иметь **внешний вид кнопки**. Ее можно использовать в декоративных целях или для **JavaScript**. Подобную кнопку также можно создать с помощью тега button.

**Пример**

В данном примере создана кнопка отправки формы. Можете нажать на нее и убедится, что форма действительно отправится:

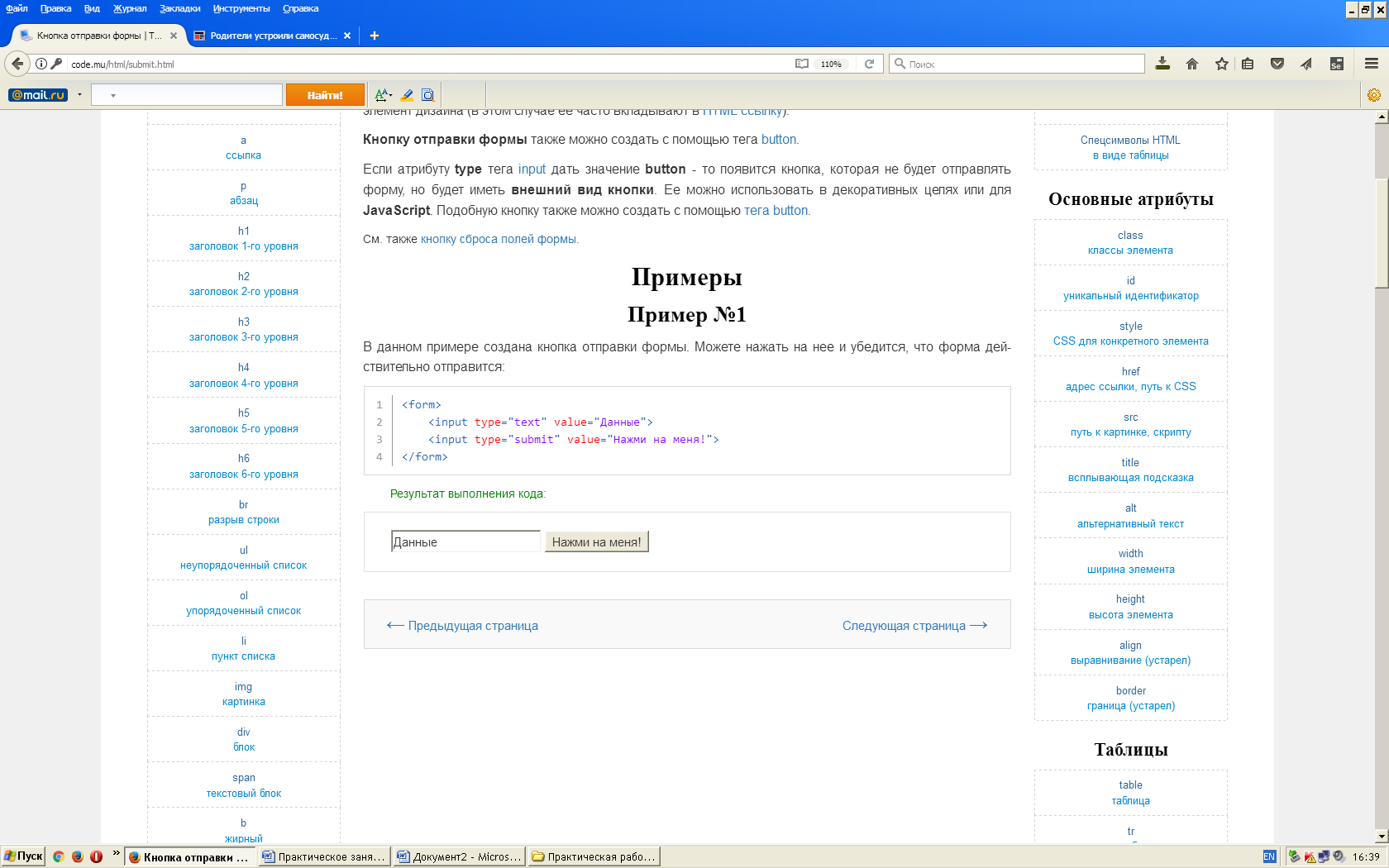
<form>

<input type="text" value="Данные">

<input type="submit" value="Нажми на меня!">

</form>

Результат выполнения кода:



# Кнопка сброса полей формы

Тег **input** с атрибутом **type** в значении **reset** создает кнопку сброса полей HTML формы. После нажатия на такую кнопку поля формы приобретут тот вид, который они имели при заходе на страницу (до внесенных пользователем изменений).

**Кнопку сброса формы** также можно создать с помощью тега button.

Если атрибуту **type** тега input дать значение **button** - то появится кнопка, которая не будет отправлять форму, но будет иметь **внешний вид кнопки**. Ее можно использовать в декоративных целях или для **JavaScript**. Подобную кнопку также можно создать с помощью тега button.

**Пример**

Измените поля формы, а затем нажмите на кнопку сброса, чтобы вернуть их в изначальное состояние:

<form>

<input type="checkbox" checked>

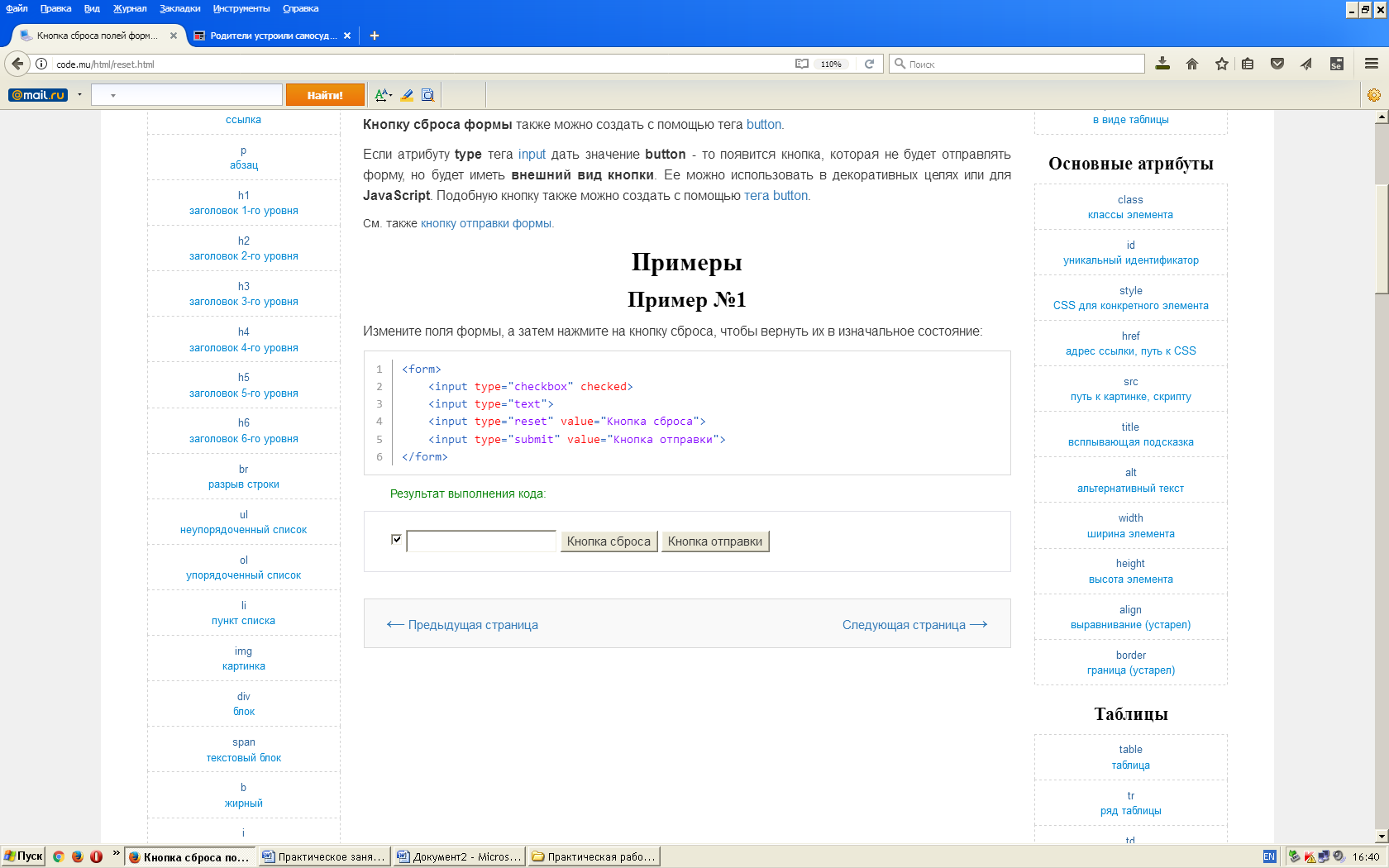
<input type="text">

<input type="reset" value="Кнопка сброса">

<input type="submit" value="Кнопка отправки">

</form>

Результат выполнения кода:



# Тег button

Тег **button** создает кнопку для использования в формах, а также просто как элемент дизайна (в этом случае ее часто вкладывают в HTML ссылку).

Кнопку также можно создать с помощью тега input с атрибутом **type** в значениях **button**, **submit** или **reset**.

## Атрибуты

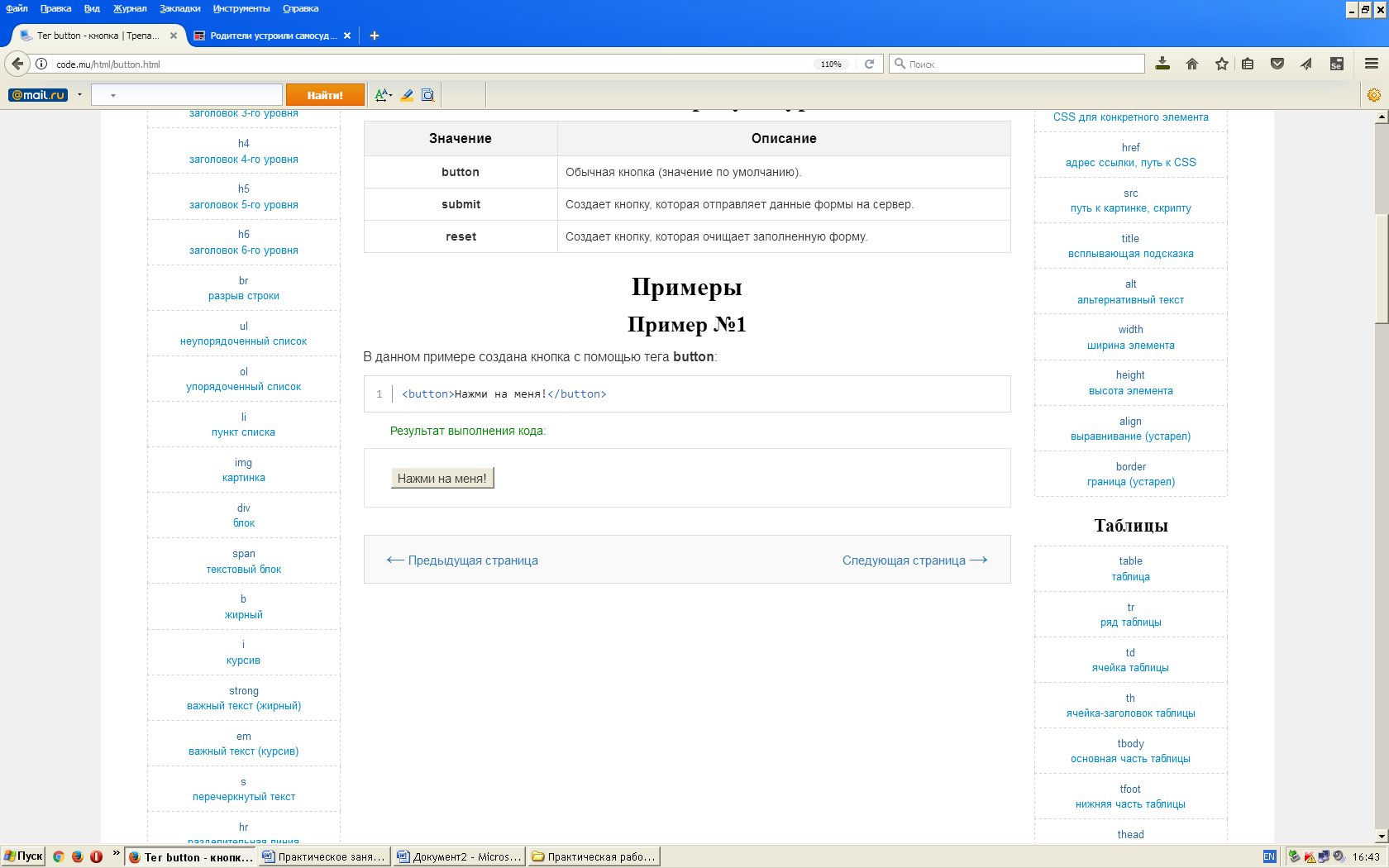
### 

### Пример

В данном примере создана кнопка с помощью тега **button**:

<button>Нажми на меня!</button>

Результат выполнения кода:



# Тег textarea

Тег **textarea** создает многострочное поле ввода для использования в HTML формах.

В отличие от тега input, в **textarea** можно вбивать много строк текста, разделяя их клавишей Enter.

Тег **textarea** должен лежать внутри тега form. Это не обязательно, необходимо для отправки данных на сервер и последующей их обработки через PHP.

По умолчанию пользователь может **изменять размеры** тега **textarea**, потянув за нижний правый угол. Это поведение можно изменить с помощью CSS свойства resize.

**Примеры**

**Пример . Просто текстареа**

В данном примере демонстрируется вид тега **textarea**:

<textarea></textarea>

Результат выполнения кода:



**Пример . Текст по умолчанию**

В данном примере тегу **textarea** добавлен текст по умолчанию. Обратите внимание на то, что в отличие от тега input, тегу **textarea** не задается атрибут value, а текст по умолчанию пишется между открывающим тегом **<textarea>** и закрывающим **</textarea>**:

<textarea>Какой-то текст.</textarea>

Результат выполнения кода:



**Пример . Паразитные пробелы**

Обратите внимание на то, что если разнести открывающий тег **<textarea>** и закрывающий **</textarea>** на разные строки, то в результате в этом поле ввода появятся **лишние пробелы** (как бороться: не разносите на разные строки):

<textarea>

Какой-то текст.

</textarea>

Результат выполнения кода:

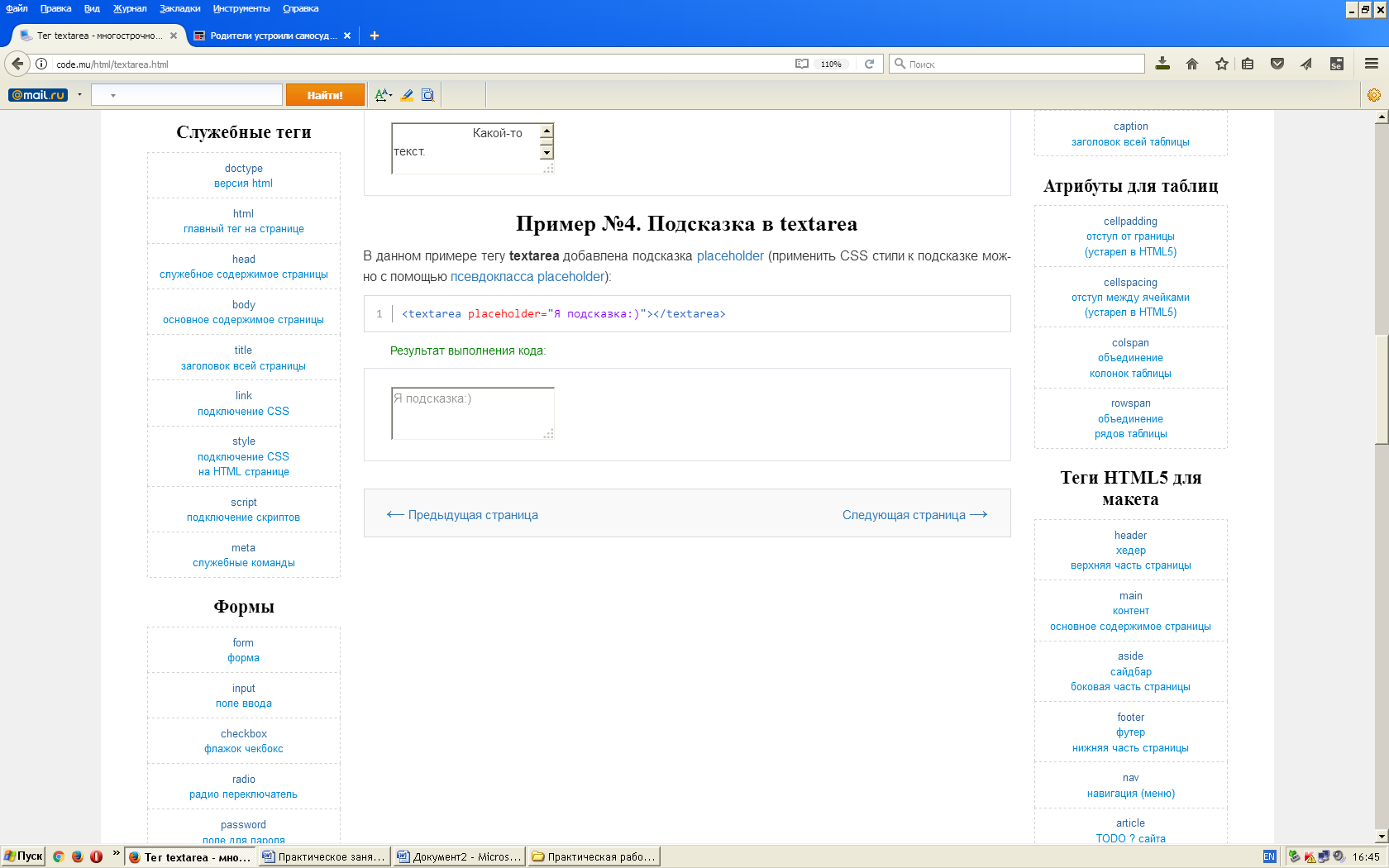


**Пример . Подсказка в textarea**

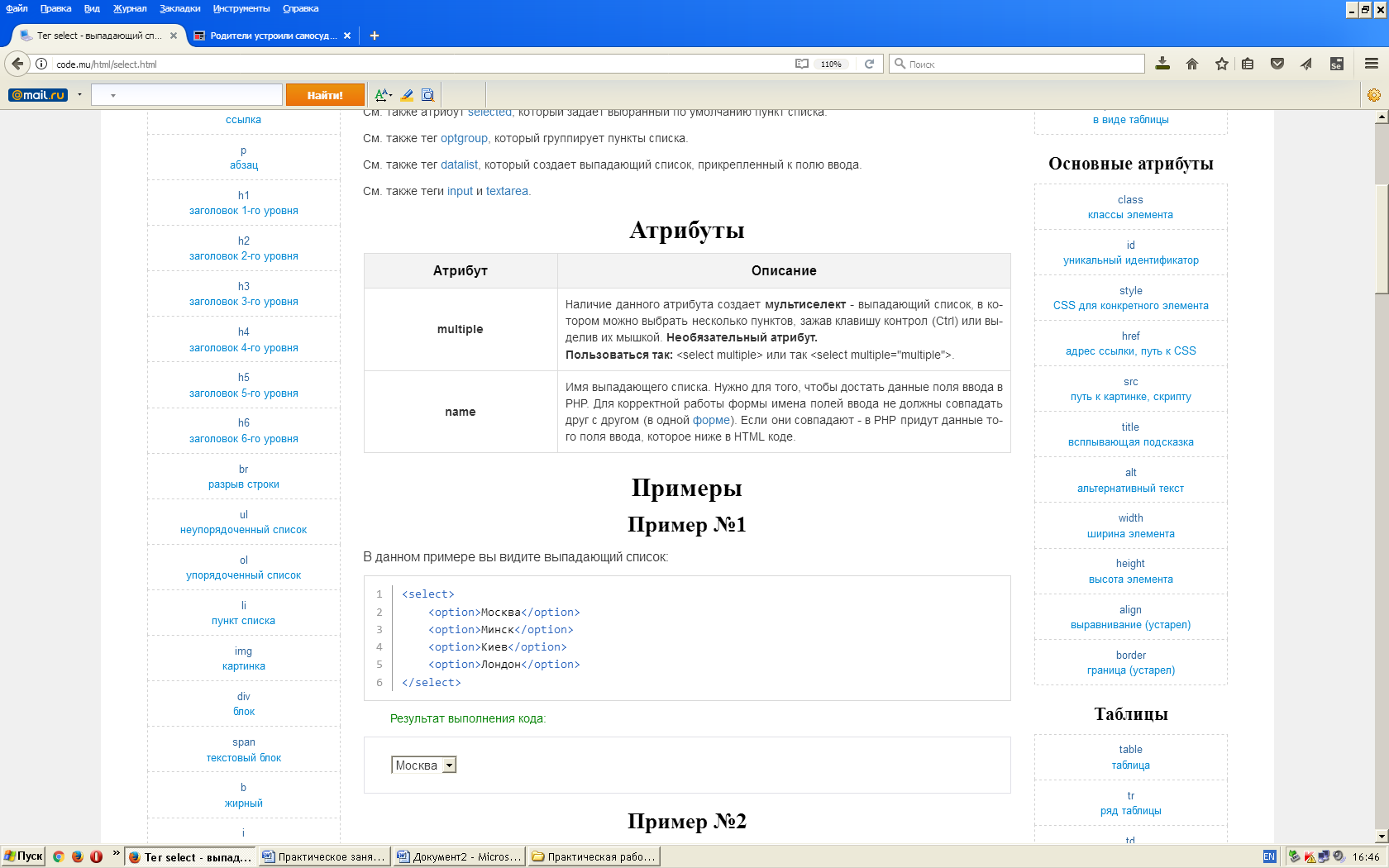
В данном примере тегу **textarea** добавлена подсказка placeholder (применить CSS стили к подсказке можно с помощью псевдокласса placeholder):

<textarea placeholder="Я подсказка:)"></textarea>

Результат выполнения кода:



# Тег select



**Пример**

В данном примере вы видите выпадающий список:

<select>

<option>Москва</option>

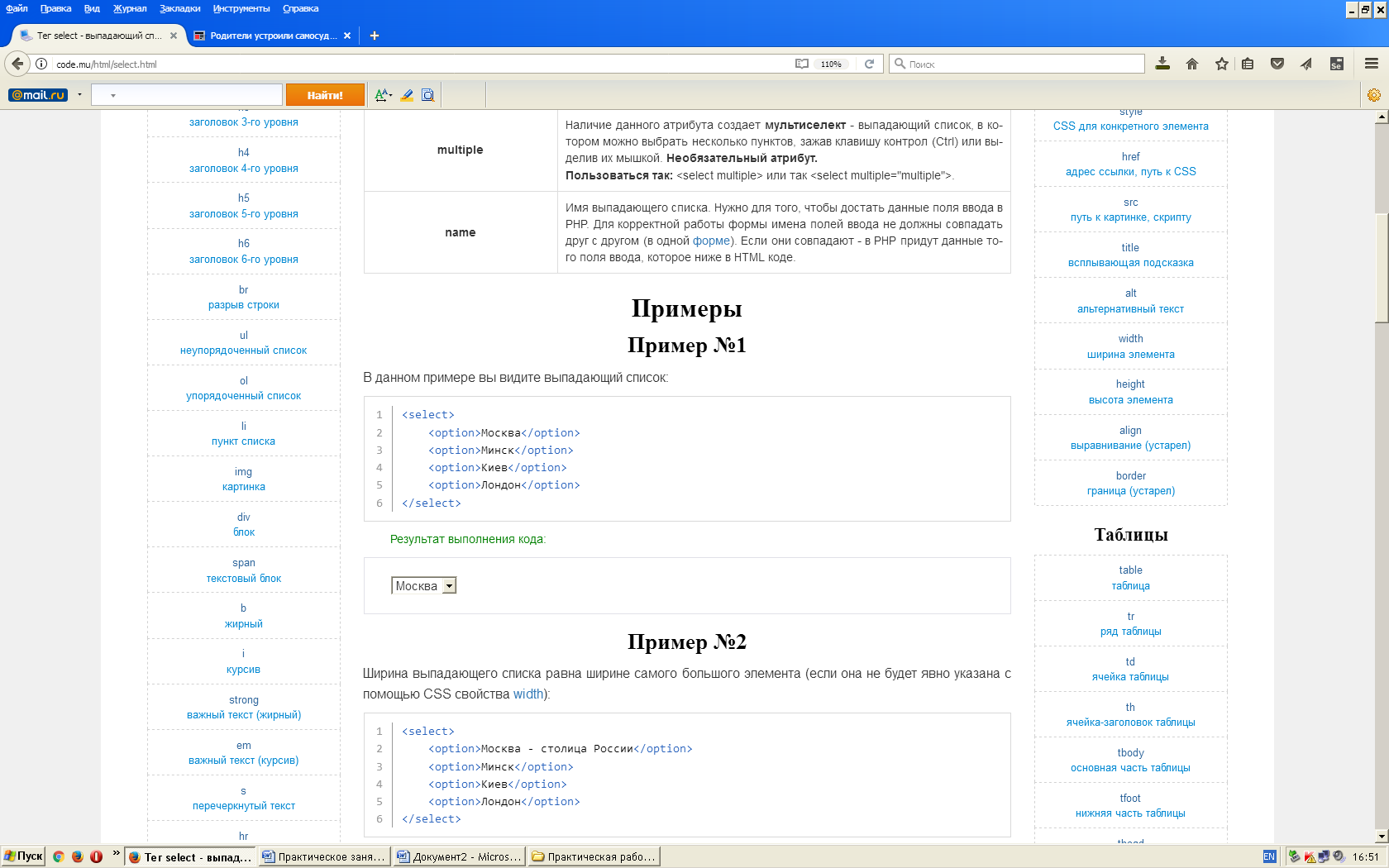
<option>Минск</option>

<option>Киев</option>

<option>Лондон</option>

</select>

Результат выполнения кода:



**Пример**

Ширина выпадающего списка равна ширине самого большого элемента (если она не будет явно указана с помощью CSS свойства width):

<select>

<option>Москва - столица России</option>

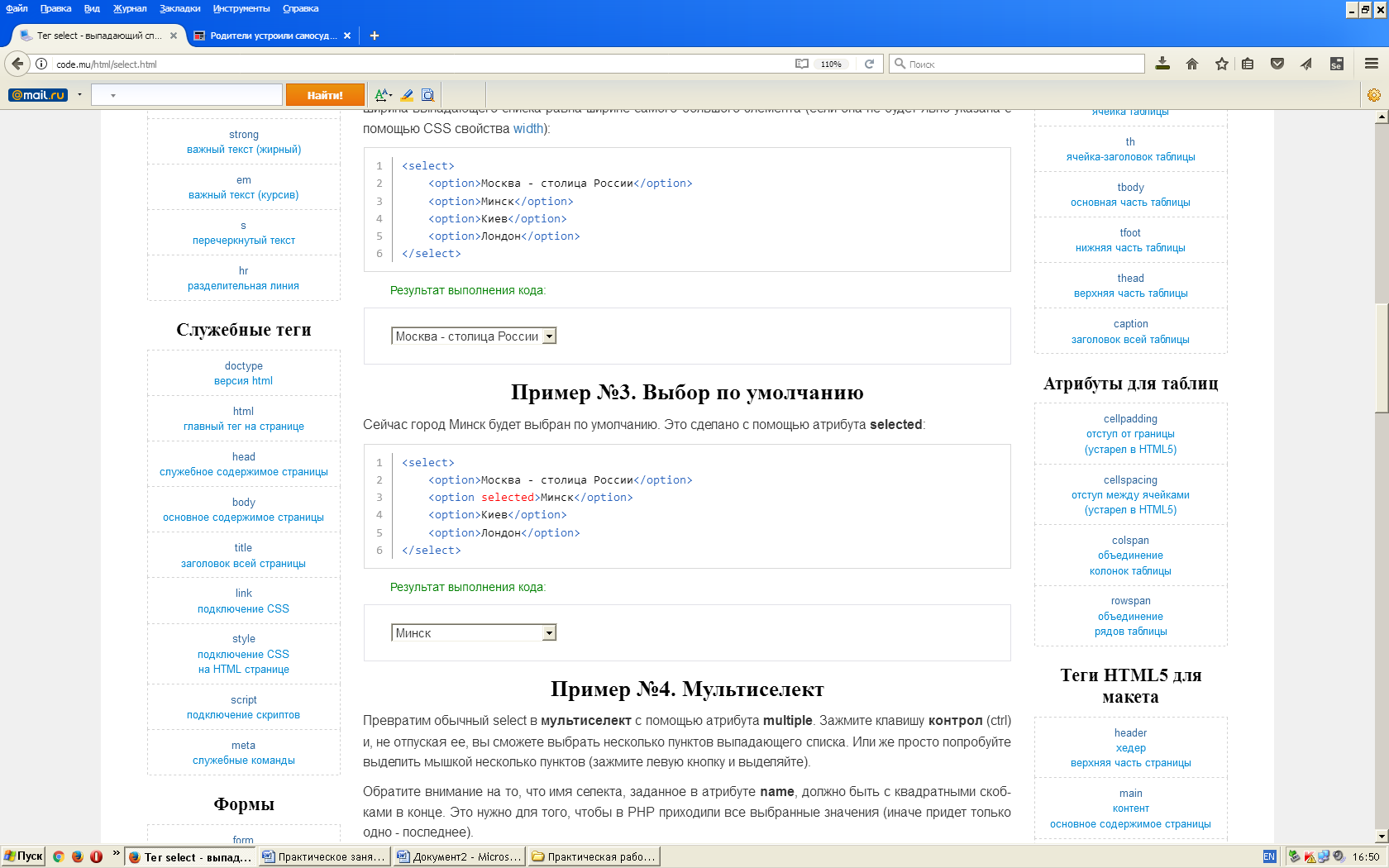
<option>Минск</option>

<option>Киев</option>

<option>Лондон</option>

</select>

Результат выполнения кода:



**Пример . Выбор по умолчанию**

Сейчас город Минск будет выбран по умолчанию. Это сделано с помощью атрибута **selected**:

<select>

<option>Москва - столица России</option>

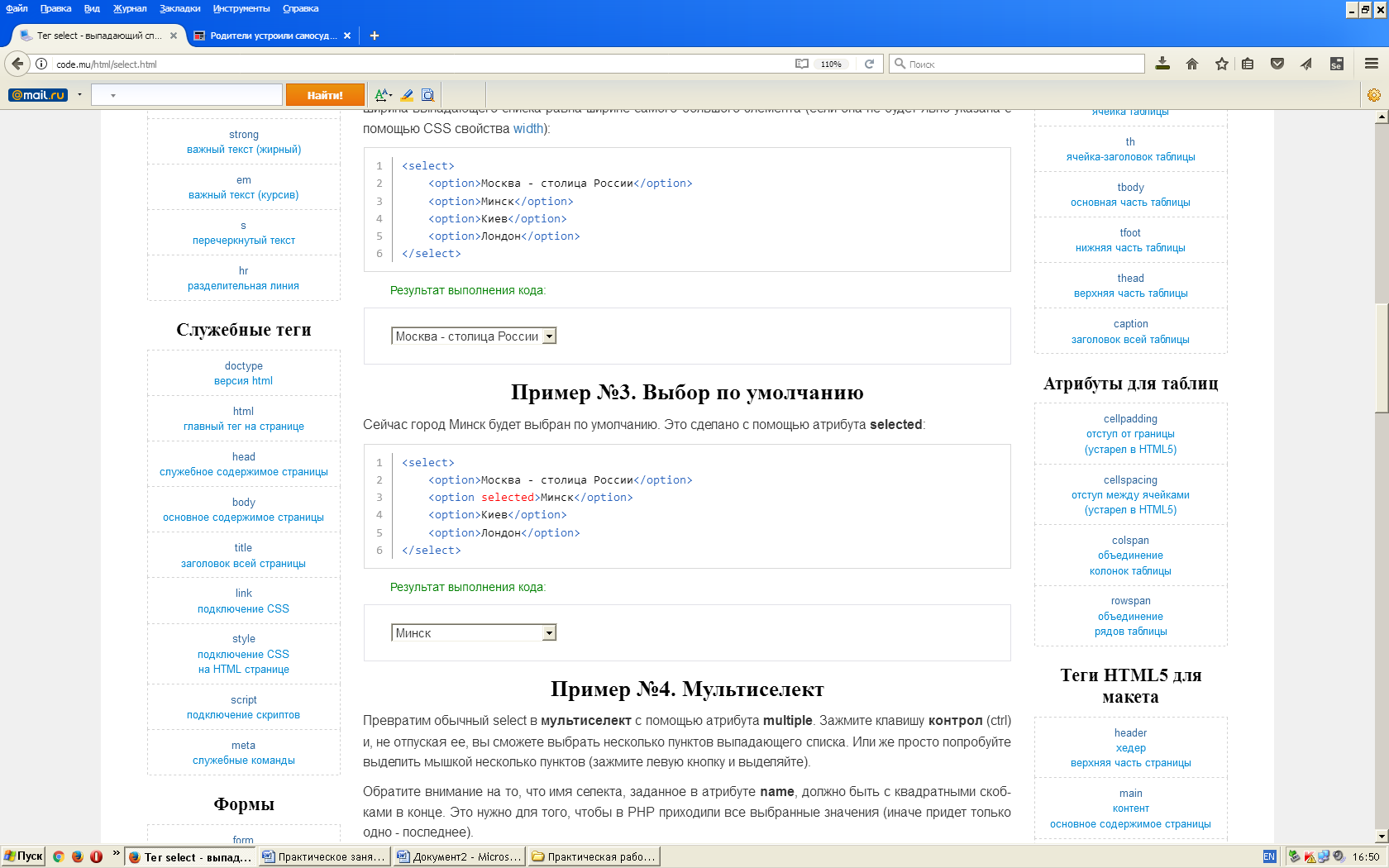
<option selected>Минск</option>

<option>Киев</option>

<option>Лондон</option>

</select>

Результат выполнения кода:



**Пример . Мультиселект**

Превратим обычный select в **мультиселект** с помощью атрибута **multiple**. Зажмите клавишу **контрол** (ctrl) и, не отпуская ее, вы сможете выбрать несколько пунктов выпадающего списка. Или же просто попробуйте выделить мышкой несколько пунктов (зажмите левую кнопку и выделяйте).

Обратите внимание на то, что имя селекта, заданное в атрибуте **name**, должно быть с квадратными скобками в конце. Это нужно для того, чтобы в PHP приходили все выбранные значения (иначе придет только одно - последнее).

<select multiple name="city[]">

<option>Москва - столица России</option>

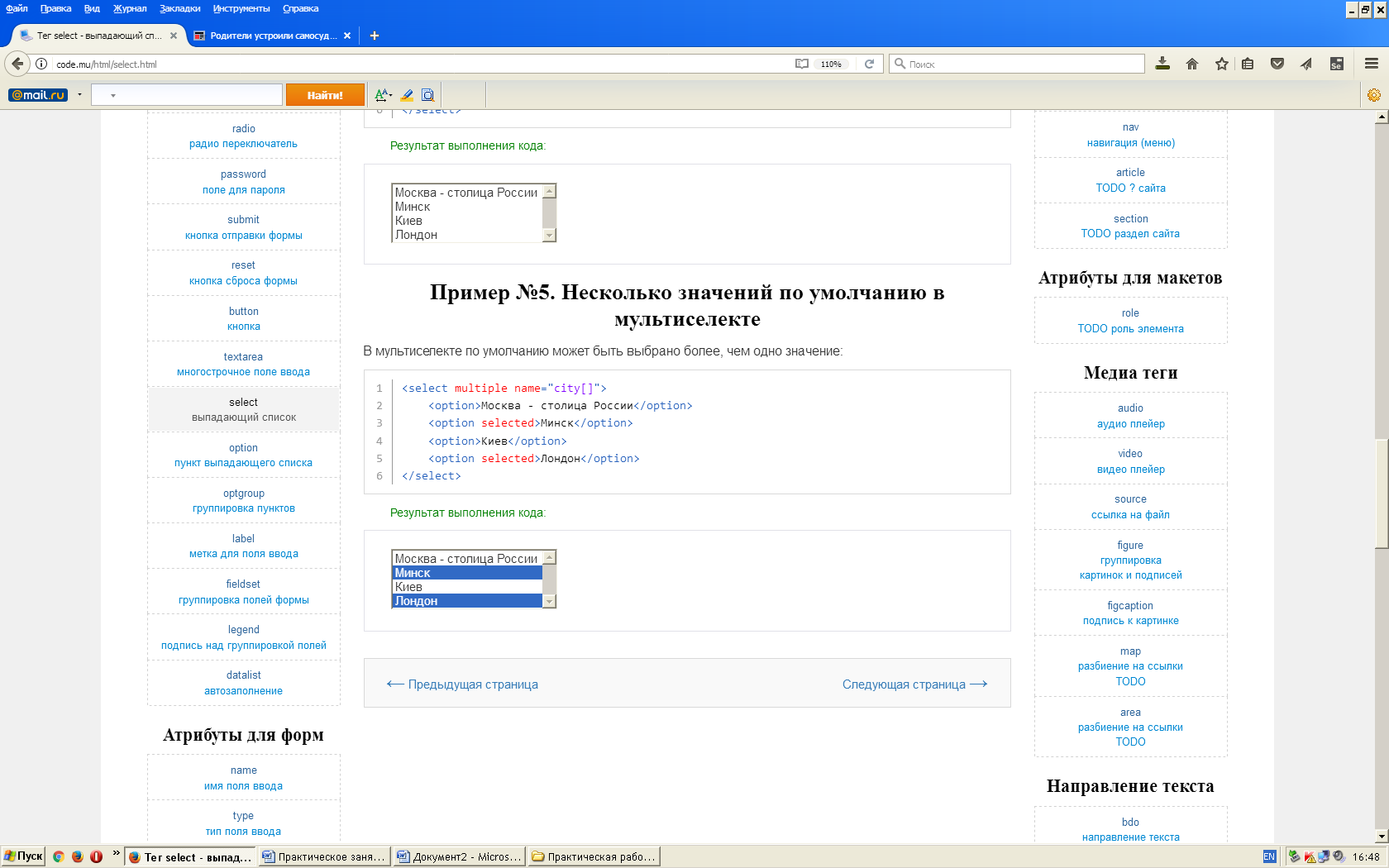
<option>Минск</option>

<option>Киев</option>

<option>Лондон</option>

</select>

Результат выполнения кода:



**Пример . Несколько значений по умолчанию в мультиселекте**

В мультиселекте по умолчанию может быть выбрано более, чем одно значение:

<select multiple name="city[]">

<option>Москва - столица России</option>

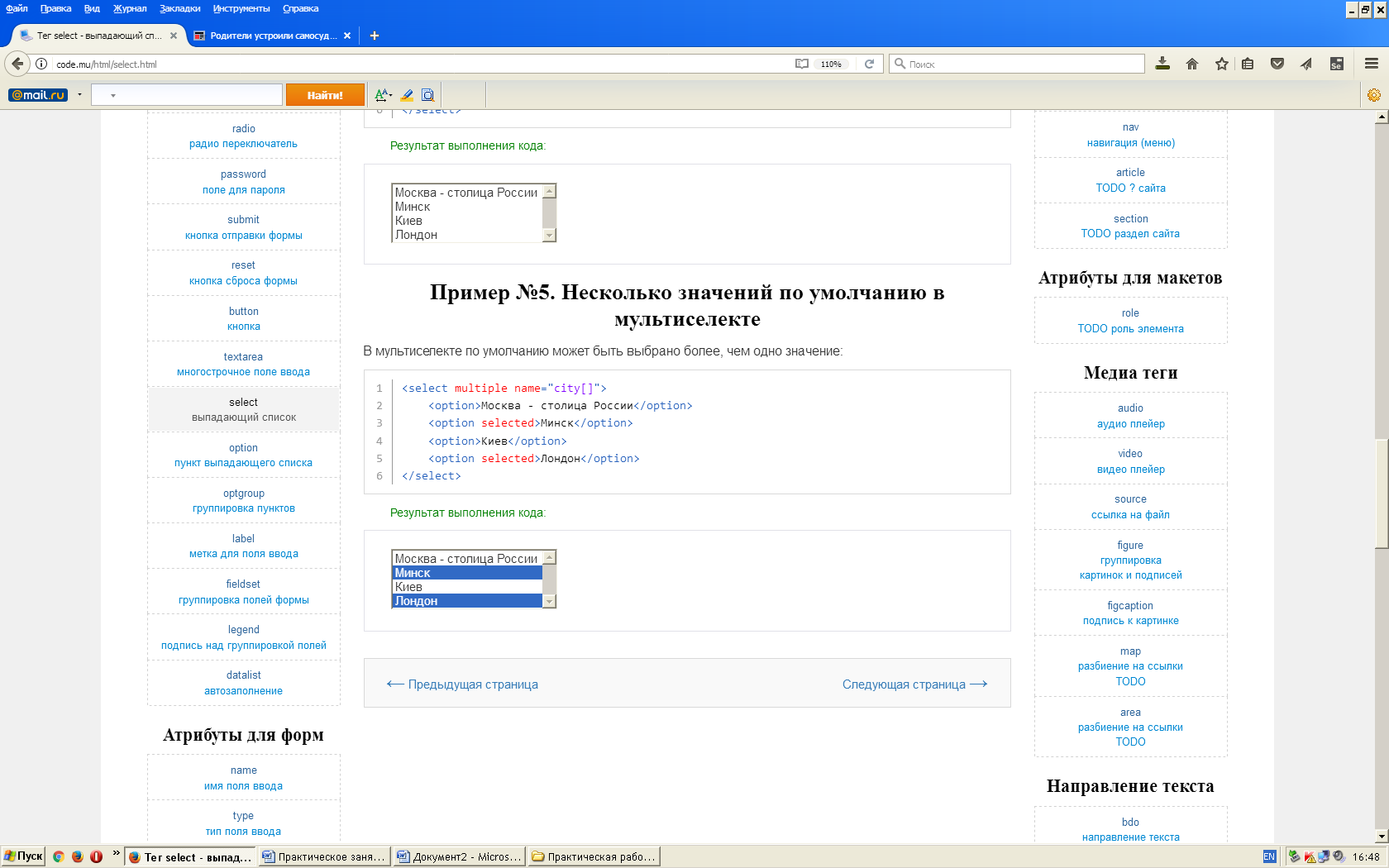
<option selected>Минск</option>

<option>Киев</option>

<option selected>Лондон</option>

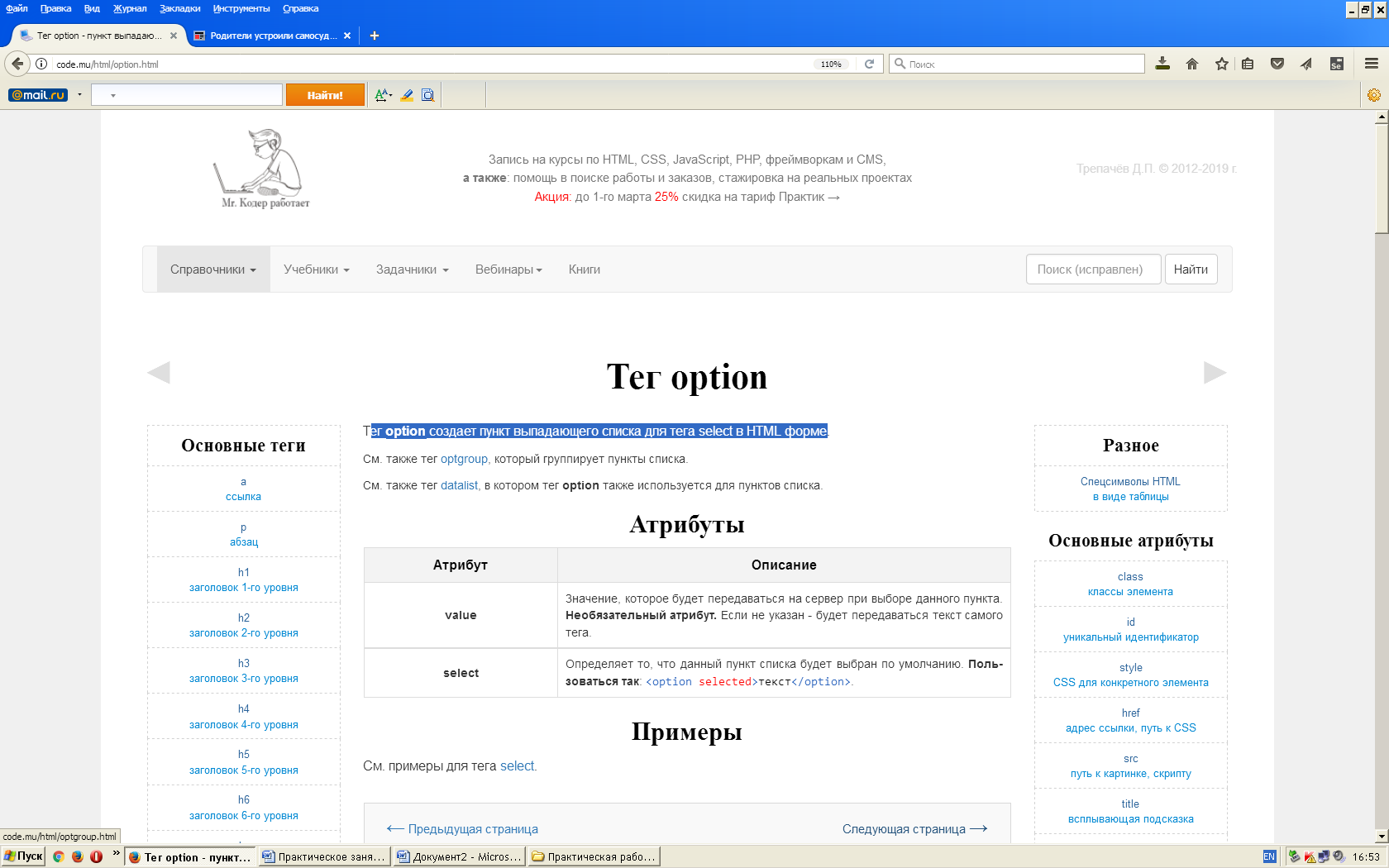
</select>

Результат выполнения кода:



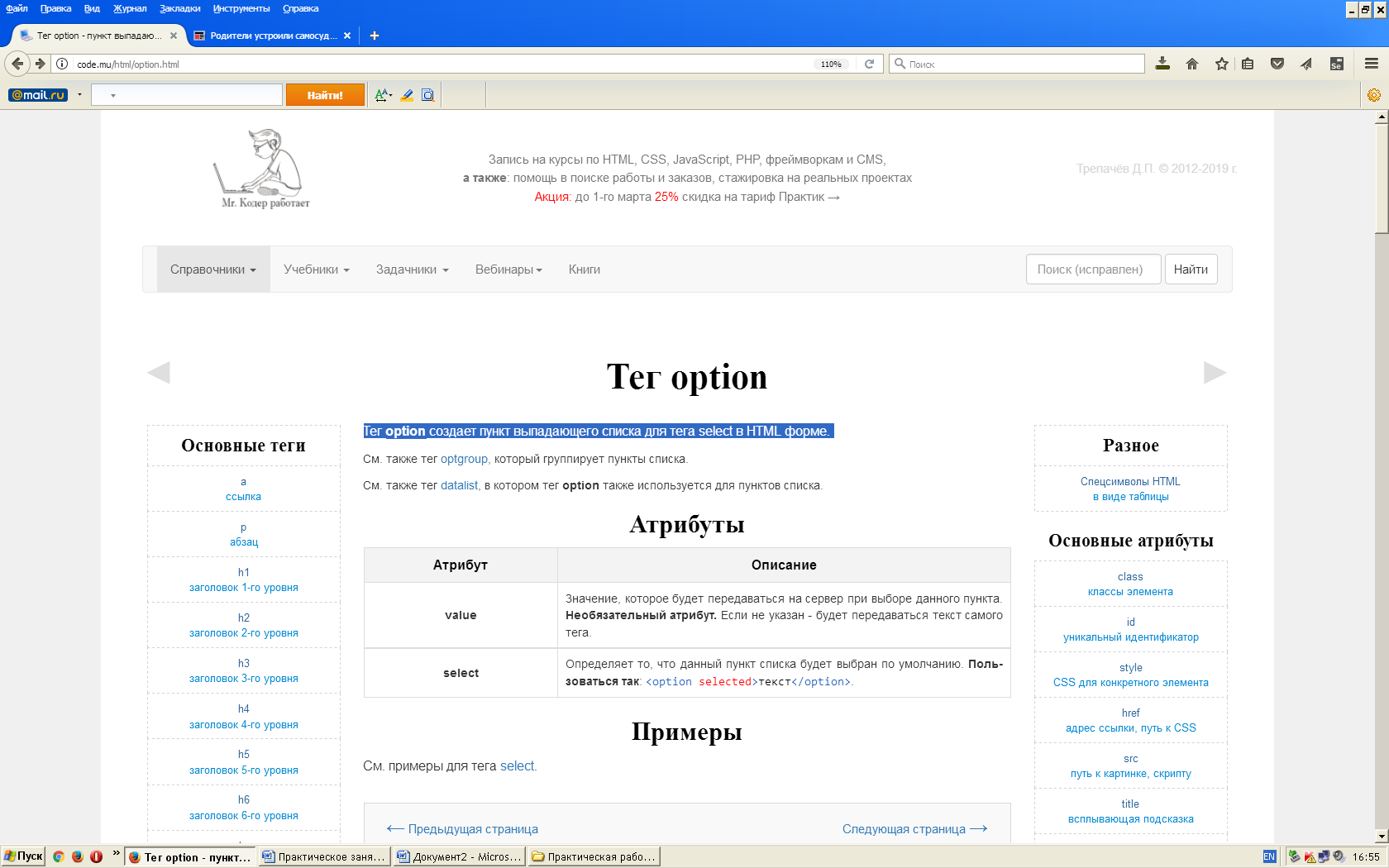
# Тег option

Тег **option** создает пункт выпадающего списка для тега select в HTML форме



# Тег option

Тег **option** создает пункт выпадающего списка для тега select в HTML форме.



# Тег optgroup

Тег **optgroup** группирует пункты option выпадающего списка select.

Группировка нужна для того, чтобы дать **общий заголовок** определенной группе пунктов списка. Этот заголовок задается в атрибуте **label** тега **optgroup**.

**Атрибуты**

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| label | Задает текст, который будет заголовком для группы пунктов списка. |

**Пример**

В данном примере с помощью тега **optgroup** пункты списка сгруппированы в две группы:

<select>

<optgroup label="Города России">

<option>Москва</option>

<option>Смоленск</option>

</optgroup>

<optgroup label="Города Беларуси">

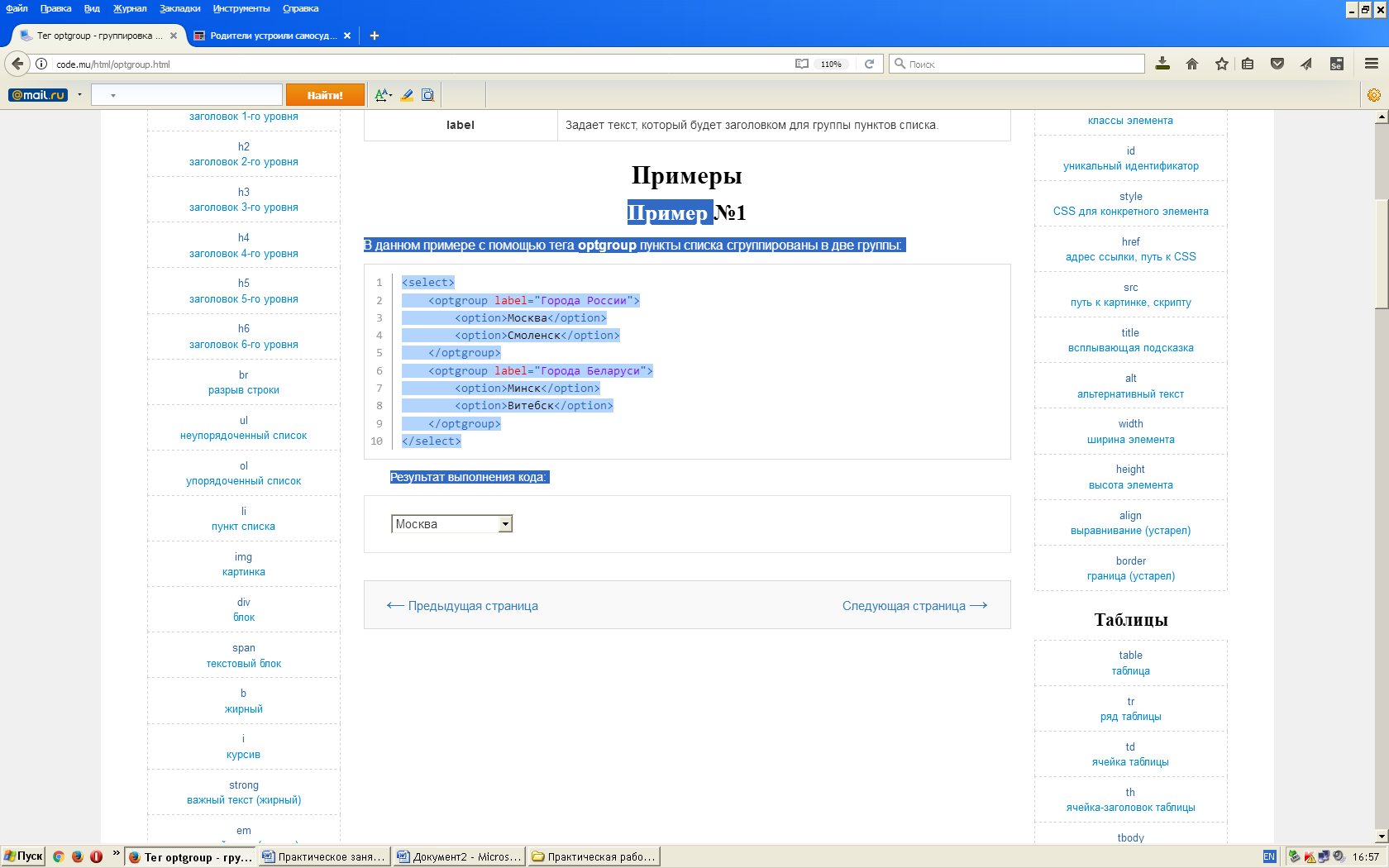
<option>Минск</option>

<option>Витебск</option>

</optgroup>

</select>

Результат выполнения кода:



# Тег label

Тег **label** задает метку для поля ввода типа input, textarea, select и так далее.

Метка привязывается к определенному полю ввода с помощью атрибута **for**. В нем следует указывать значение атрибута id поля ввода, к которому привязана метка. Также метка привязывается к элементу, если этот элемент положить вовнутрь тега label. В этом случае атрибут for указывать не надо. Смотрите примеры для лучшего понимания.

По нажатию на метку привязанные к ней чекбоксы или радио кнопочки будут менять свое состояние с отмеченного на неотмеченное и наоборот. Метка в данном случае нужна для удобства: в мелкие элементы формы тяжело попасть курсором, а в длинную метку с текстом - на порядок легче.

Привязанное к метке текстовое поле ввода типа input и textarea будет получать фокус ввода. **Фокус** - это когда курсор моргает в поле ввода, в этом случае если вы что-то будете писать на клавиатуре - текст попадет в это.

Метку **label** можно использовать для имитации чекбокса или радио кнопочки. Это нужно для того, чтобы сделать checkbox или radio со **своим дизайном** (что на CSS запрещено, но с помощью хитрых приемов возможно).

**Атрибуты**

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| for | В данном атрибуте следует указывать значение атрибута id того поля ввода, к которому привязана метка label. |
| accesskey | Задает горячую клавишу, с помощью которой можно перейти к привязаному к метке (через атрибут **for**) элементу формы. |

**Пример 1. Чекбокс**

В данном примере к флажку checkbox привязана метка с помощью атрибута **for**. Нажатие на метку будет приводить к **смене состояния** флажка с отмеченного на неотмеченное и наоборот:

<input type="checkbox" id="checkbox1">

<label for="checkbox">Я привязан к чекбоксу с id checkbox.</label>

Результат выполнения кода:

Я привязан к чекбоксу с id checkbox.

**Пример 2. Внутри label**

Можно также не привязывать метку через атрибут for, а ложить элементы прямо в тег label, в этом случае нажатие на текст label также будет приводить к активации элемента:

<label><input type="checkbox"> метка</label>

Результат выполнения кода:

метка

**Пример . Текстовое поле ввода**

В данном примере к тегу input привязана метка с помощью атрибута **for**. Нажатие на метку будет приводить к **получению фокуса** полем ввода. При этом содержащийся в поле текст (добавлен с помощью атрибута value) станет **выделенным**. Если начать писать что-либо на клавиатуре - этот текст затрется. **Избавится от выделения**, не удалив при этом текст, можно **кликнув мышкой** по полю ввода: фокус при этом не исчезнет, а выделение пропадет:

<input type="text" id="input1" value="Какой-то текст">

<label for="input1">Я привязан к инпуту с id input1.</label>

Результат выполнения кода:

Я привязан к инпуту с id input1.

# Тег fieldset

Тег **fieldset** группирует поля HTML формы.

Результат при этом обводится рамкой, а подпись, которая создается с помощью тега legend, разрывает эту рамку, создавая неповторимый другими средствами эффект (см. пример).

Если назначить тегу **legend** какие-либо CSS стили для позиционирования (float, position и т.п.), то его содержимое уйдет с рамки и станет в нужное место (**разрыва** в рамке при этом **больше не будет**).

### Пример

В данном примере форма разбита на два логических блока с помощью двух тегов **fieldset**:

<form>

<fieldset>

<legend>Данные покупателя</legend>

<input type="text">

</fieldset>

<fieldset>

<legend>Данные продавца</legend>

<input type="text">

</fieldset>

</form>

# Тег legend

Тег **legend** задает подпись над группировкой полей HTML формы, сделанной с помощью тега fieldset.

Если назначить тегу **legend** какие-либо CSS стили для позиционирования (float, position и т.п.), то его содержимое уйдет с рамки и станет в нужное место (разрыва в рамке при этом больше не будет).

### Пример

В данном примере форма разбита на два логических блока с помощью двух тегов fieldset, а подписи к блокам сделаны с помощью тега **legend**:

<form>

<fieldset>

<legend>Данные покупателя</legend>

<input type="text">

</fieldset>

<fieldset>

<legend>Данные продавца</legend>

<input type="text">

</fieldset>

</form>

# Тег datalist

Тег **datalist** создает автозаполнение для полей ввода input в HTML форме.

**Суть автозаполнения:** когда пользователь будет пытаться набрать что-то в поле ввода, снизу будет появляться всплывающая подсказка, которая будет представлять собой **список доступных вариантов выбора**. Пользователь сможет выбрать один из предложенных вариантов, не вводя его до конца.

Чтобы привязать к тегу input автозаполнение, ему необходимо дать атрибут **list**, в котором следует указать атрибут id привязываемого тега **datalist**.

Варианты выбора хранятся в тегах **option**, которые в свою очередь хранятся в теге **datalist**.

**Пример**

В данном примере в поле ввода нужно ввести страну. Введите вначале букву 'б' - вы увидите вплывающую подсказку из трех стран. Затем введите еще и букву 'е' - количество предложенных стран сократится для двух:

<input type="text" list="country">

<datalist id="country">

<option>Беларусь</option>

<option>Бельгия</option>

<option>Болгария</option>

</datalist>

Результат выполнения кода:



**Пример**

В данном примере к тегу **option** добавлен атрибут **value**. При выборе определенной страны в привязанный инпут попадет не название страны, а содержимое атрибута **value**:

<input type="text" list="country-value">

<datalist id="country-value">

<option value="Беларусь">Беларусь - страна озер и картошки</option>

<option value="Бельгия">Бельгия - страна, в которой живут бельгийцы</option>

<option value="Болгария">Болгария - страна для отдыха</option>

</datalist>

Результат выполнения кода:



**Пример . Влияние атрибута autocomplete**

На выпадающий список подсказок оказывает влияние атрибут autocomplete. Если он включен (а так и будет по умолчанию), то к значениям, которые вы задали в теге **datalist**, будут **примешиваться** те значения, которые пользователь **вводил в это поле ранее**.

Вбейте в поле 'Бразилия' и **нажмите кнопку отправки** (без отправки браузер **не запомнит** то, что вы вводили ранее, в нашем случае Бразилию). Затем вернитесь к этому примеру и введите букву 'б' - в появившемся списке вы увидите не только 3 страны из datalist, но и вбитую ранее Бразилию (кроме того, если вы проделаете то же самое со словом 'Беларусь' - в выпадающем списке будет **две** Беларуси).

Если вам это мешает - отключите атрибут autocomplete, добавив ему значение 'off'.

Проделайте описанные выше манипуляции с примером:

<form action="">

<input type="text" list="country" autocomplete="on">

<datalist id="country">

<option>Беларусь</option>

<option>Бельгия</option>

<option>Болгария</option>

<input type="submit">

</datalist>

</form>

Результат выполнения кода:

