# Практическая работа №7

**Отправка форм на сервер**

**Работа с полями ввода форм**

**Цели:**

Отправка форм на сервер

Работа с полями ввода форм

**Теория:**

Одним из основных способов передачи данных веб-сайту является обработка форм. Формы представляют специальные элементы разметки HTML, которые содержат в себе различные элементы ввода - текстовые поля, кнопки и т.д. И с помощью данных форм мы можем ввести некоторые данные и отправить их на сервер. А сервер уже обрабатывает эти данные.

Создание форм состоит из следующих аспектов:

Создание элемента <form><form> в разметке HTML

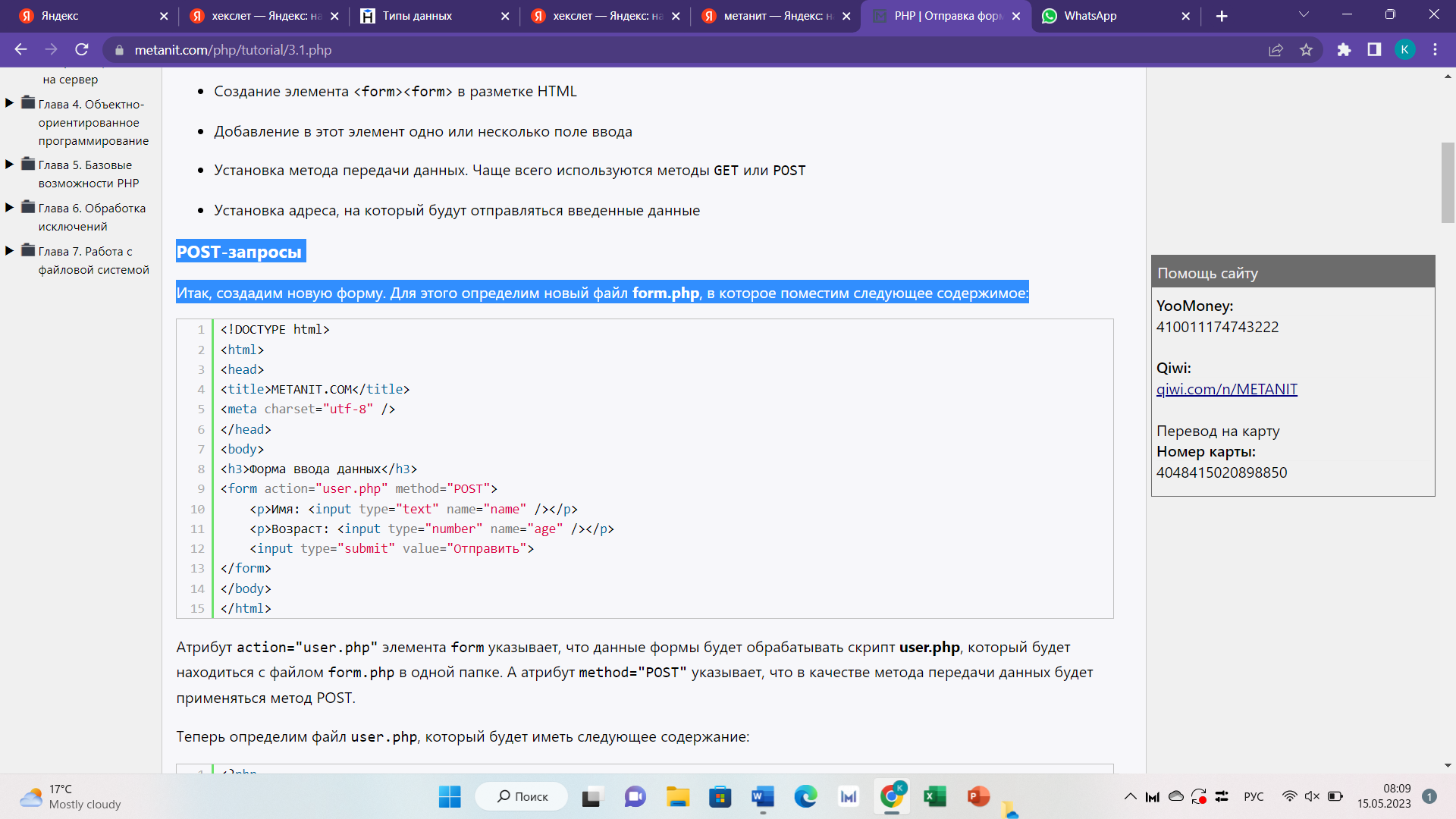
Добавление в этот элемент одно или несколько поле ввода

Установка метода передачи данных. Чаще всего используются методы GET или POST

Установка адреса, на который будут отправляться введенные данные

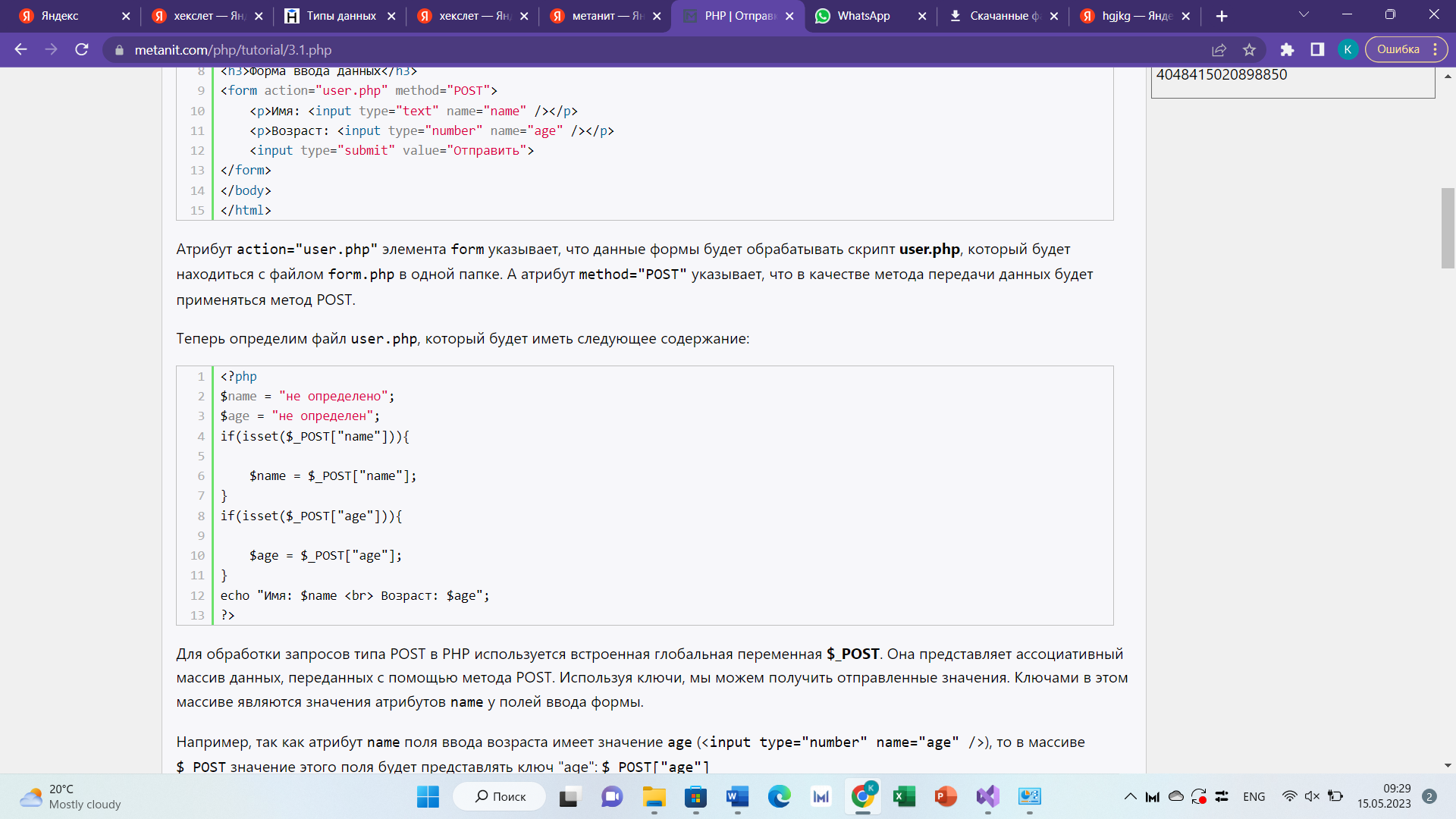
POST-запросы

Итак, создадим новую форму. Для этого определим новый файл form.php, в которое поместим следующее содержимое:



Атрибут action="user.php" элемента form указывает, что данные формы будет обрабатывать скрипт user.php, который будет находиться с файлом form.php в одной папке. А атрибут method="POST" указывает, что в качестве метода передачи данных будет применяться метод POST.

Теперь определим файл user.php, который будет иметь следующее содержание:

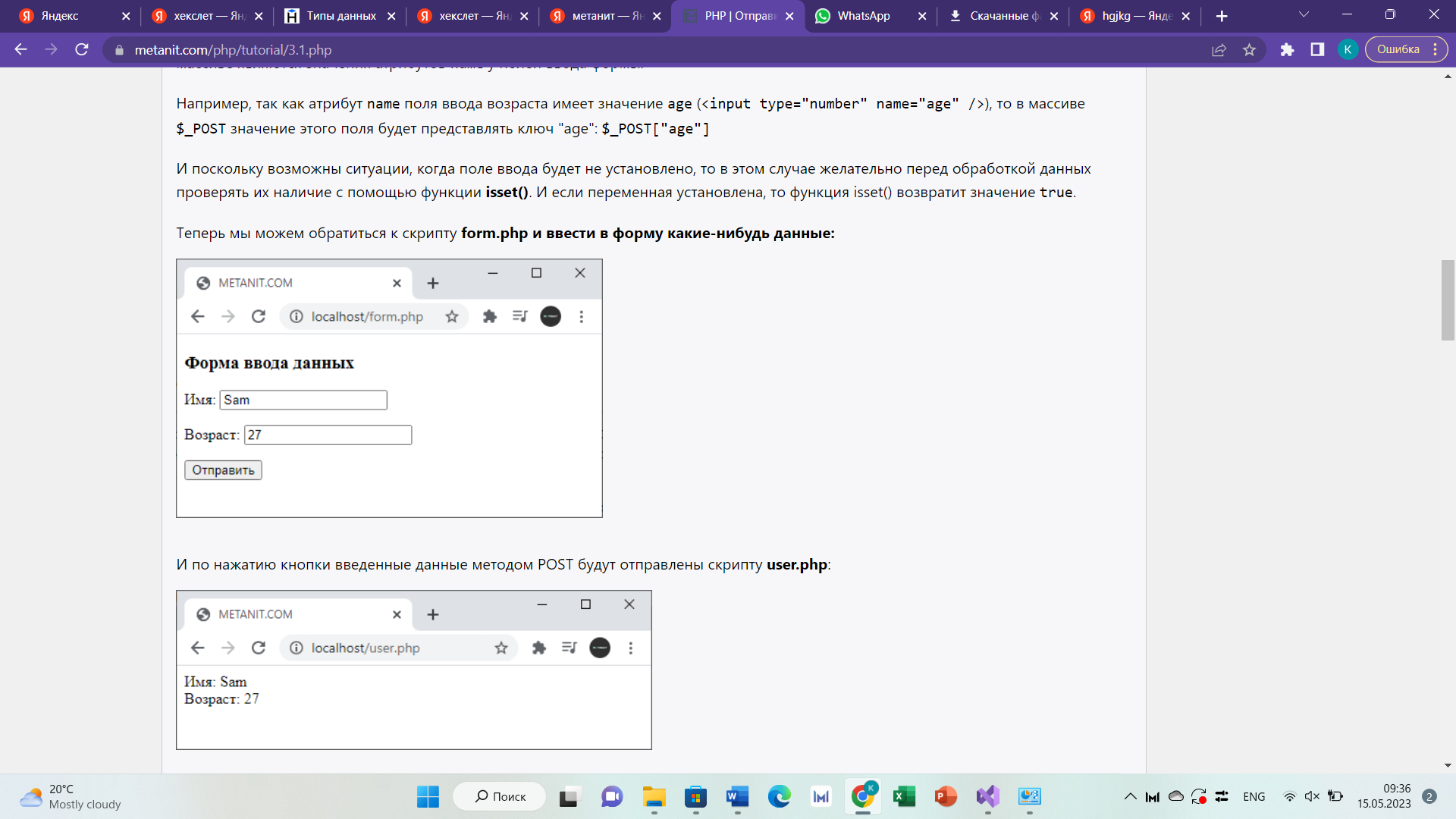


Для обработки запросов типа POST в PHP используется встроенная глобальная переменная $\_POST. Она представляет ассоциативный массив данных, переданных с помощью метода POST. Используя ключи, мы можем получить отправленные значения. Ключами в этом массиве являются значения атрибутов name у полей ввода формы.

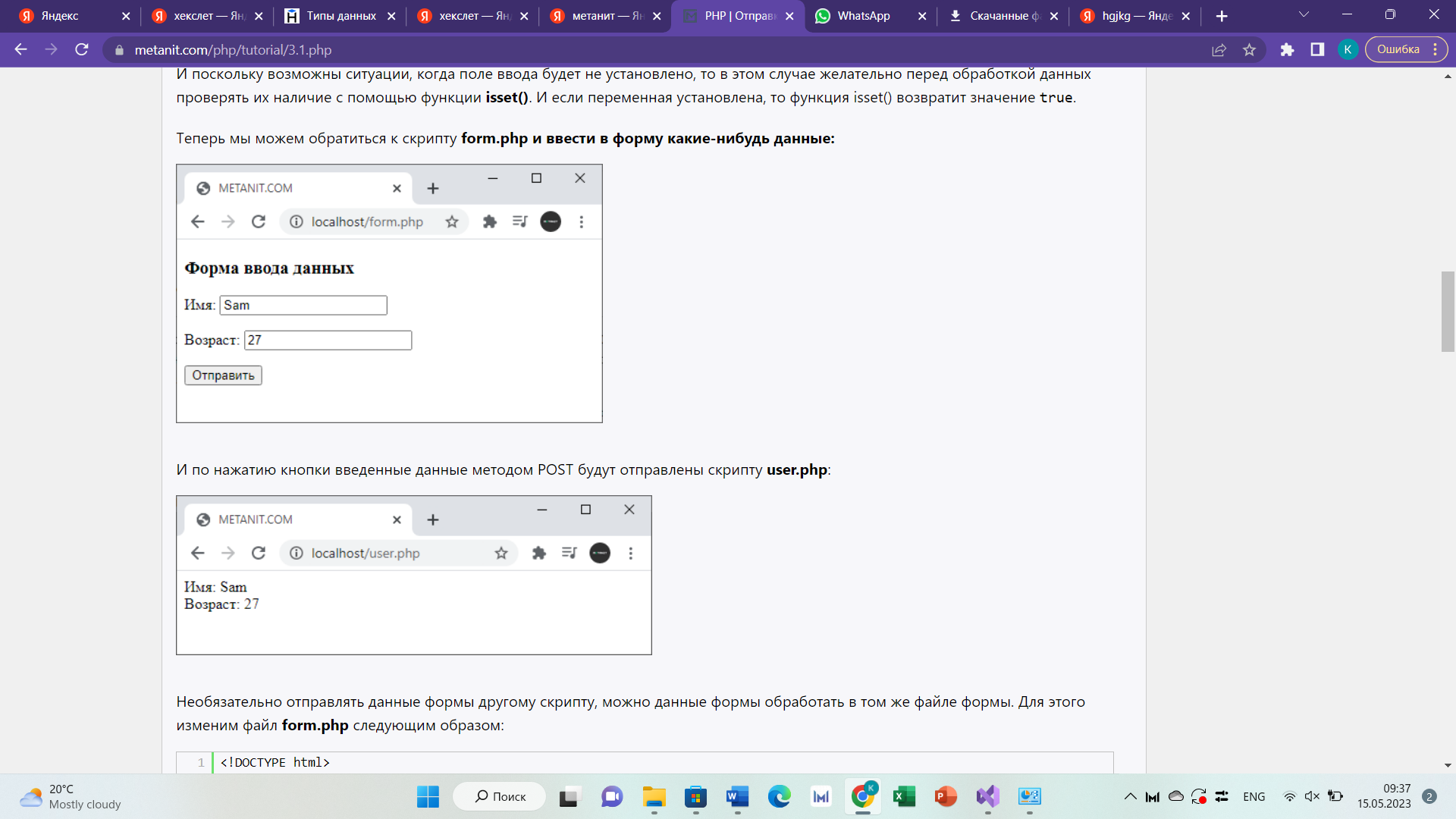
Например, так как атрибут name поля ввода возраста имеет значение age (<input type="number" name="age" />), то в массиве $\_POST значение этого поля будет представлять ключ "age": $\_POST["age"]

И поскольку возможны ситуации, когда поле ввода будет не установлено, то в этом случае желательно перед обработкой данных проверять их наличие с помощью функции isset(). И если переменная установлена, то функция isset() возвратит значение true.

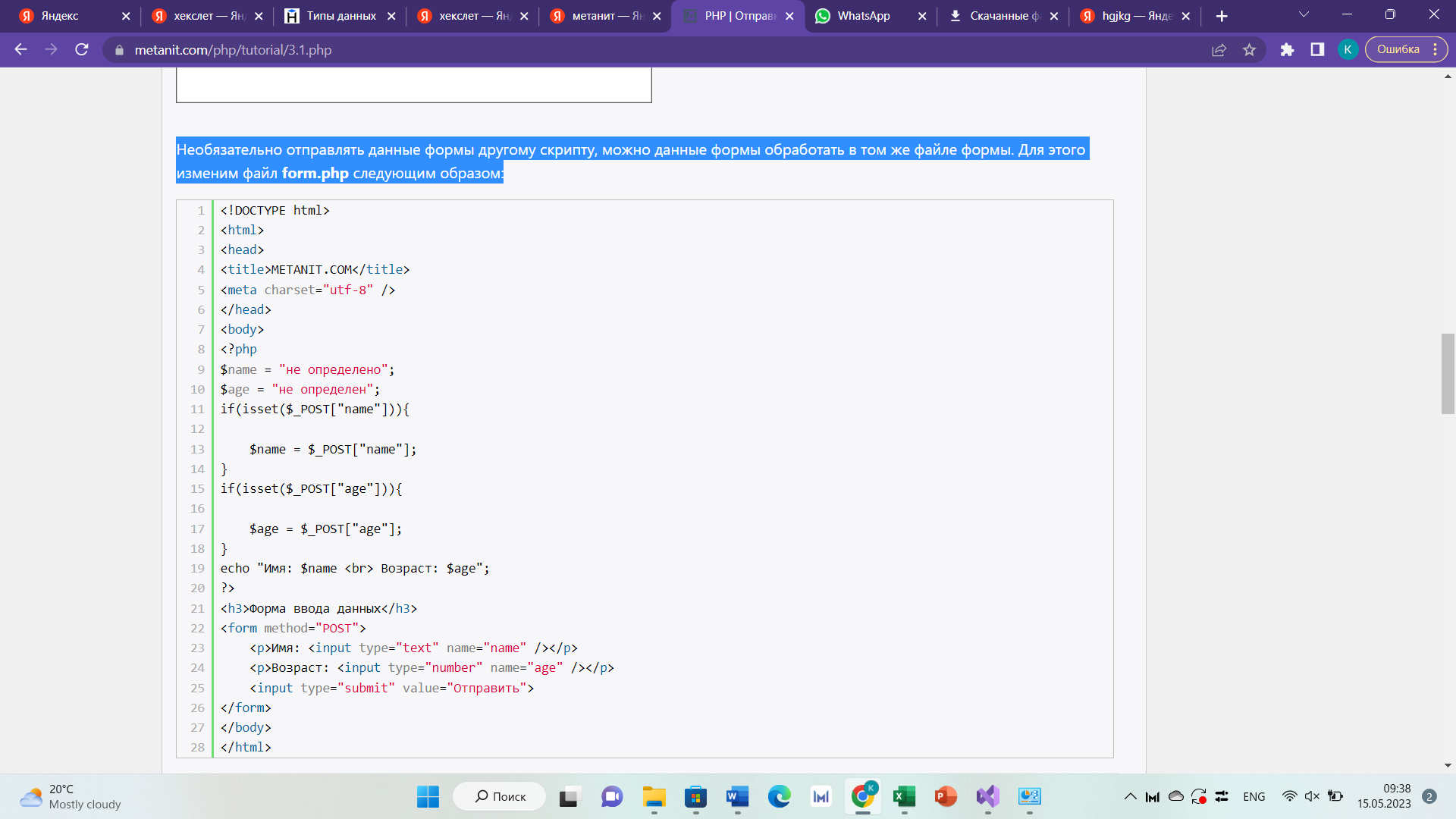
Теперь мы можем обратиться к скрипту form.php и ввести в форму какие-нибудь данные:



И по нажатию кнопки введенные данные методом POST будут отправлены скрипту user.php:



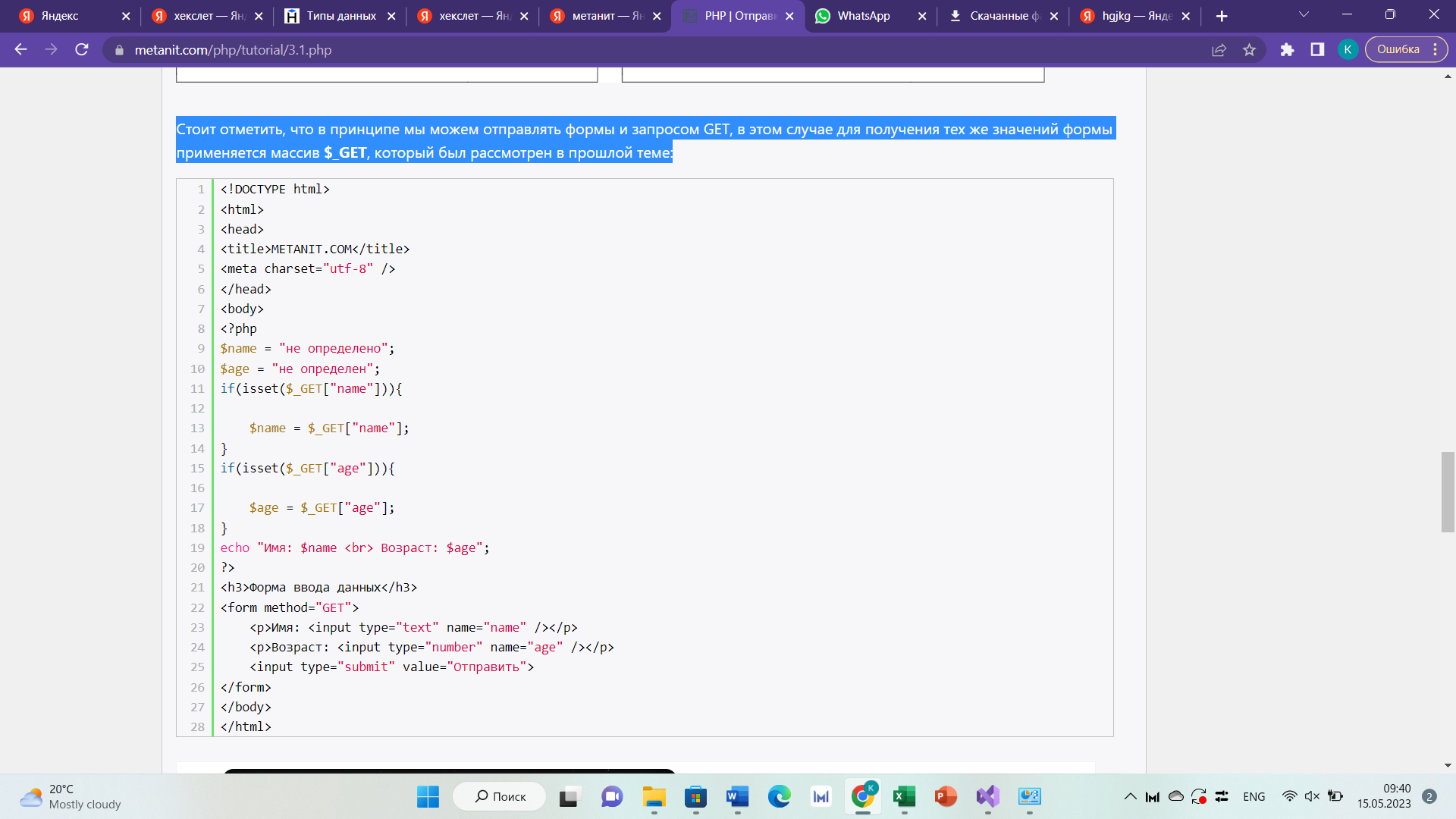
Необязательно отправлять данные формы другому скрипту, можно данные формы обработать в том же файле формы. Для этого изменим файл form.php следующим образом:



Поскольку в данном случае мы отправляем данные этому же скрипту - то есть по тому же адресу, то у элемента форма можно не устанавливать атрибут action.



Стоит отметить, что в принципе мы можем отправлять формы и запросом GET, в этом случае для получения тех же значений формы применяется массив $\_GET, который был рассмотрен в прошлой теме:

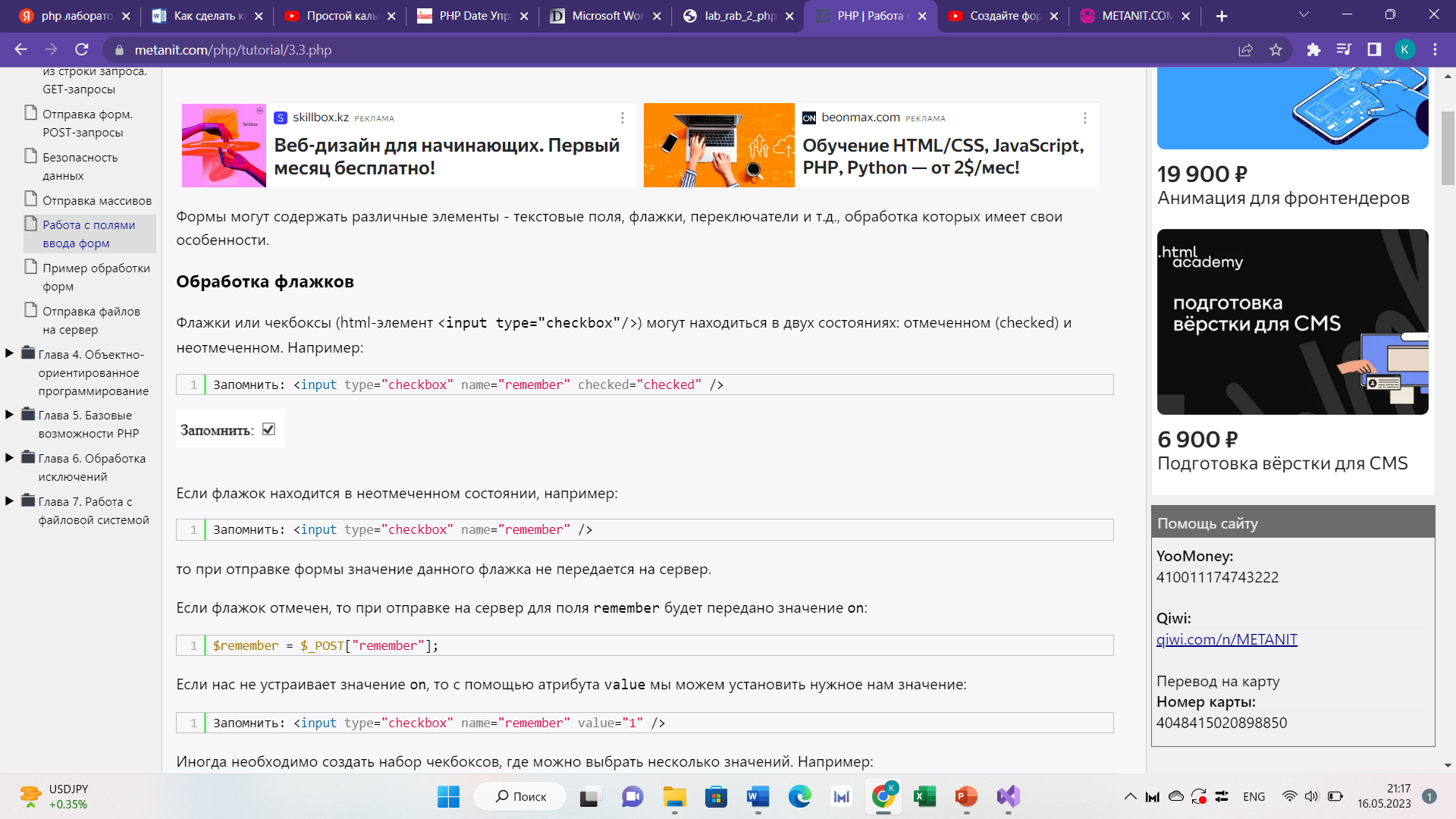


**Работа с полями ввода форм**

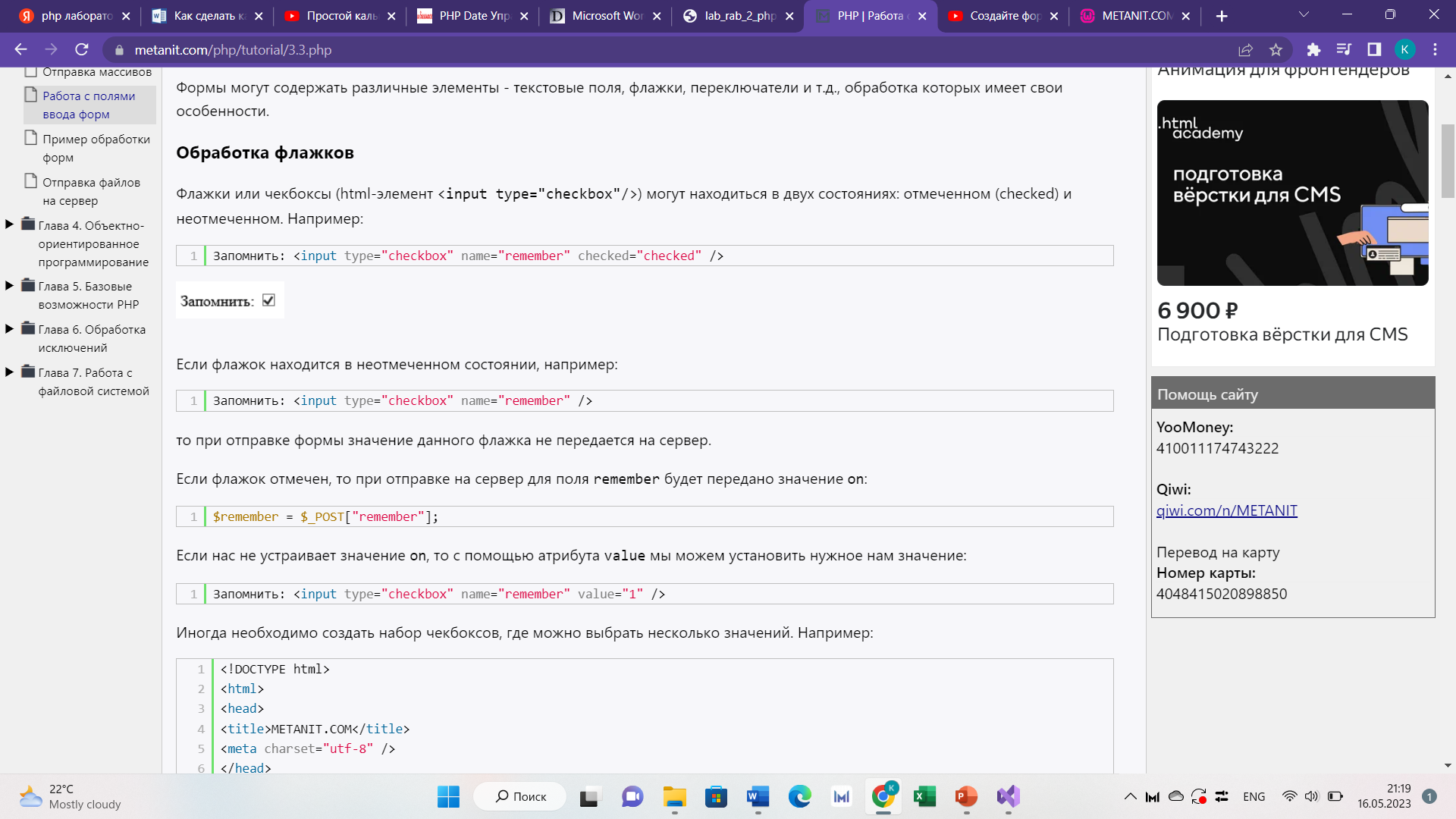
Формы могут содержать различные элементы - текстовые поля, флажки, переключатели и т.д., обработка которых имеет свои особенности.

Обработка флажков

Флажки или чекбоксы (html-элемент <input type="checkbox"/>) могут находиться в двух состояниях: отмеченном (checked) и неотмеченном. Например:

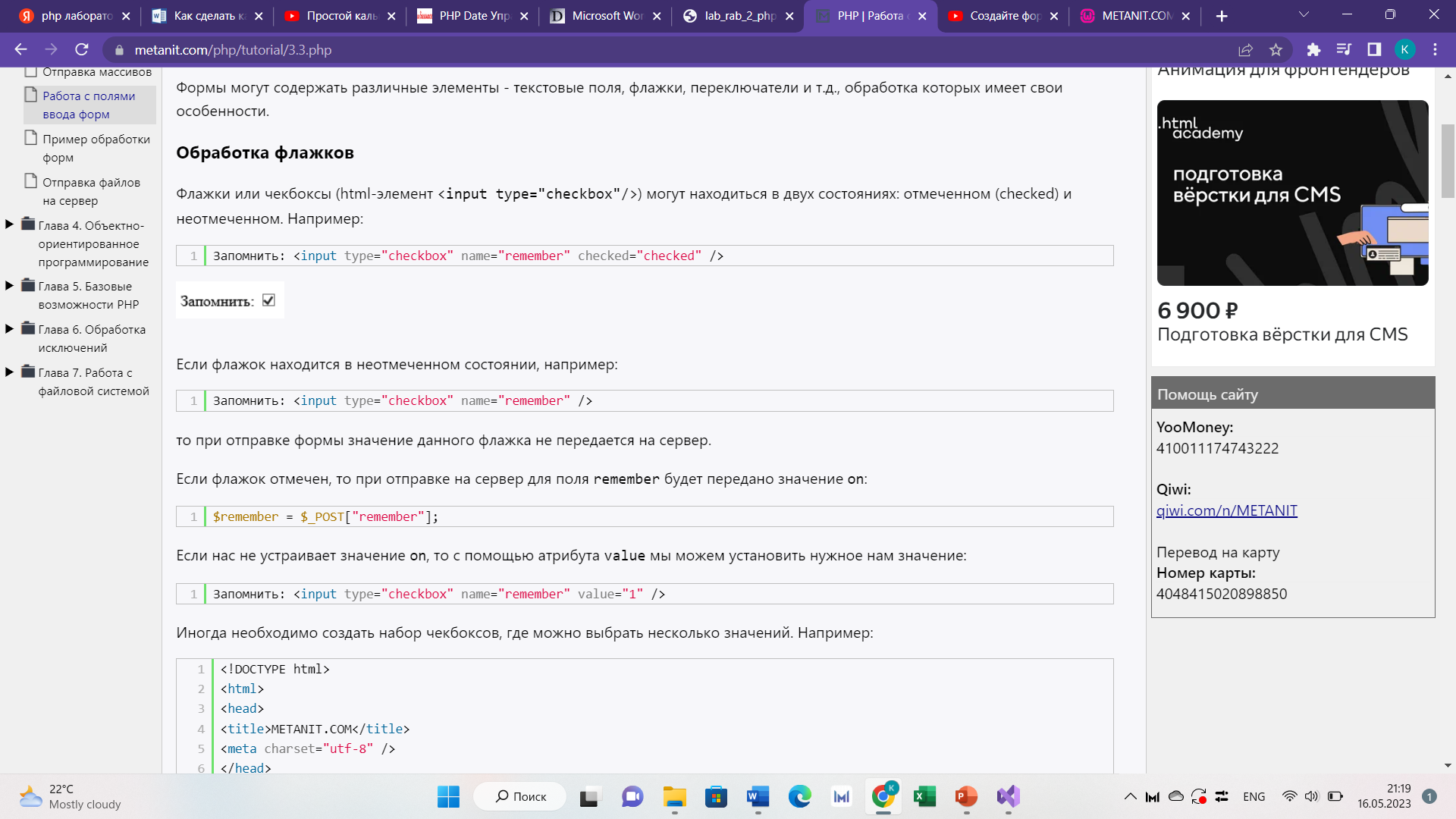


Если флажок находится в неотмеченном состоянии, например:

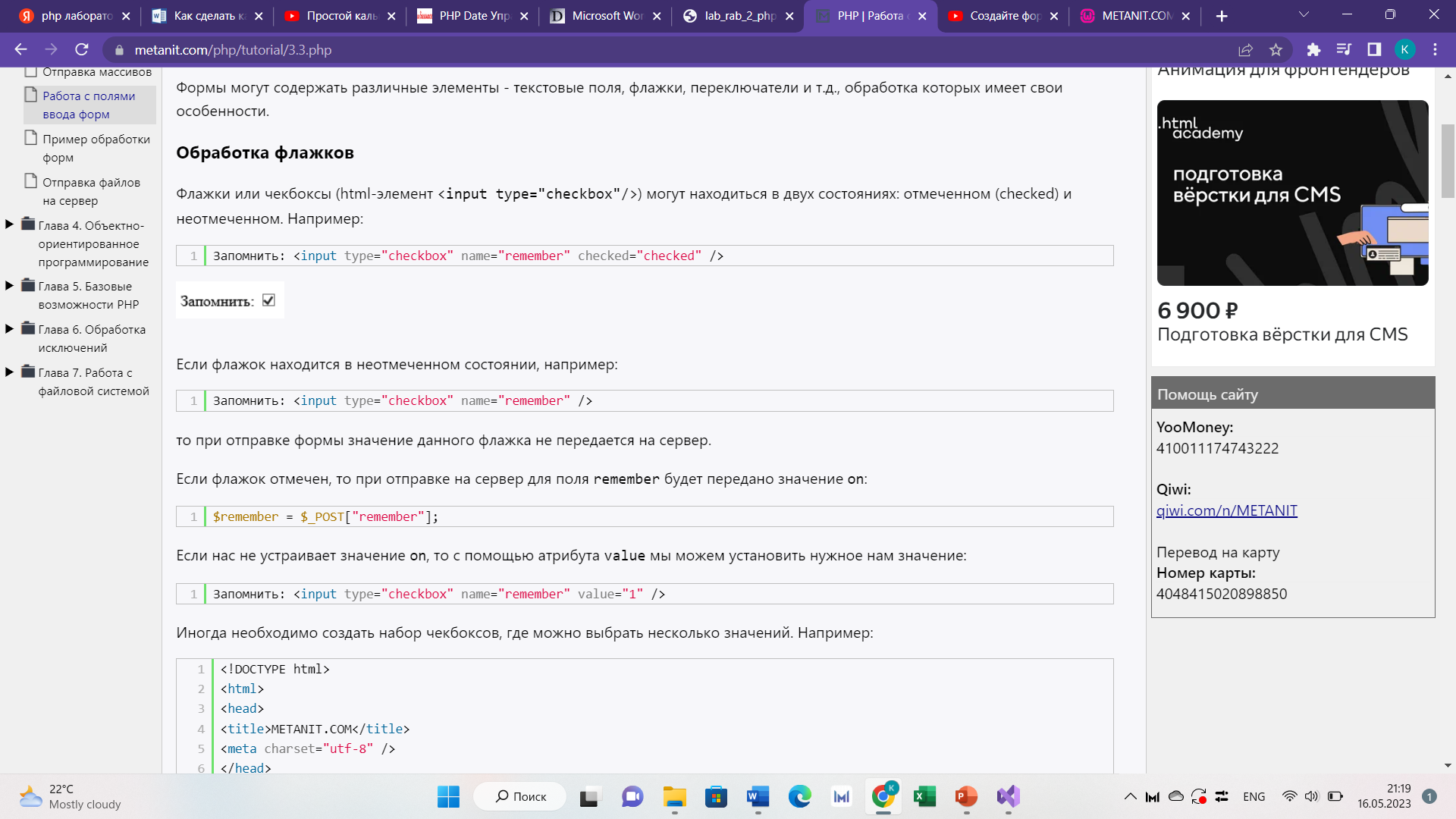


то при отправке формы значение данного флажка не передается на сервер.

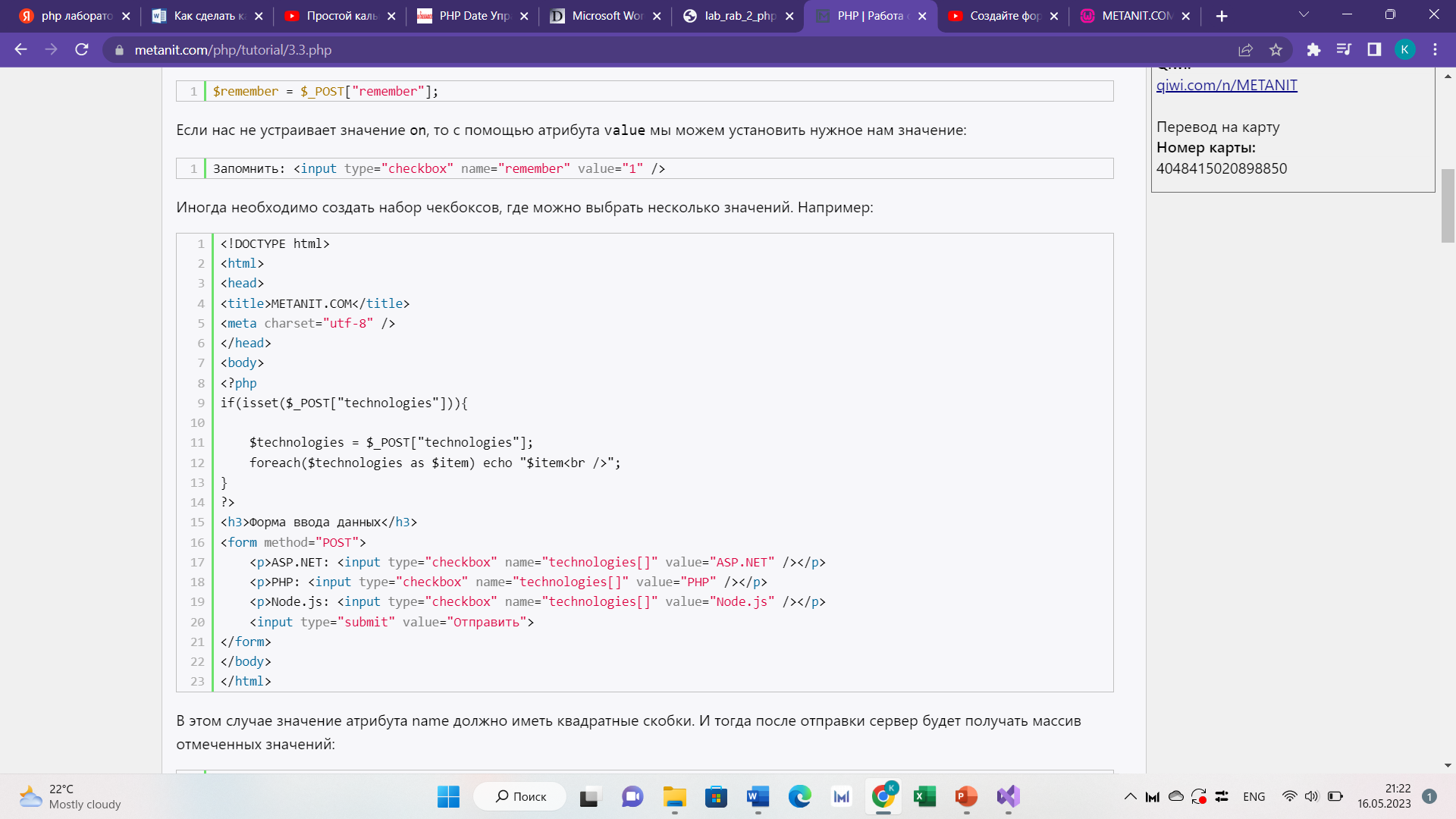
Если флажок отмечен, то при отправке на сервер для поля remember будет передано значение on:



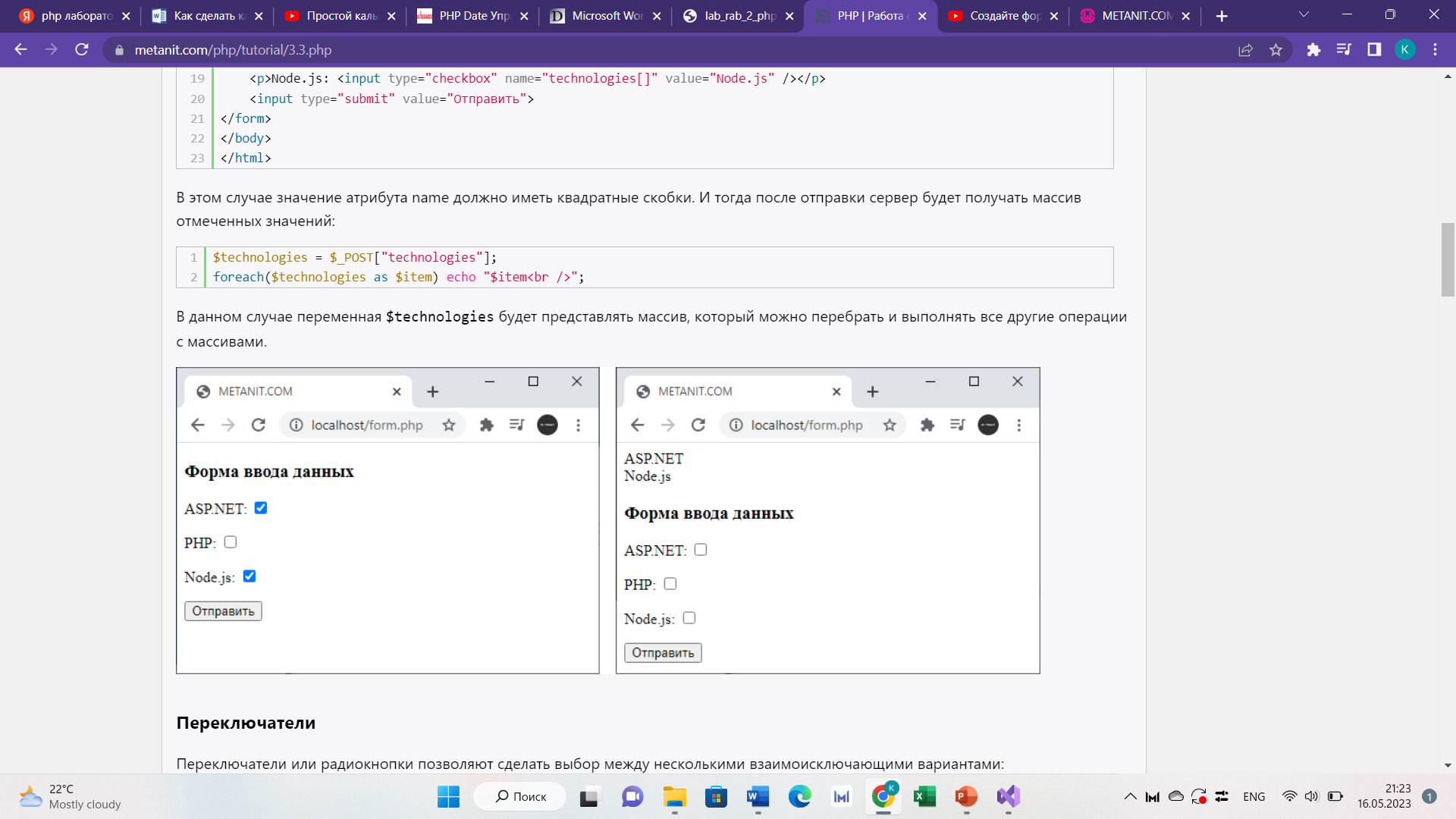
Если нас не устраивает значение on, то с помощью атрибута value мы можем установить нужное нам значение:



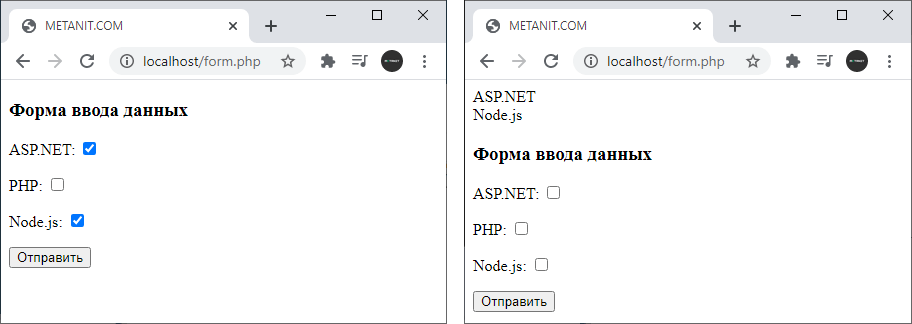
Иногда необходимо создать набор чекбоксов, где можно выбрать несколько значений. Например:



В этом случае значение атрибута name должно иметь квадратные скобки. И тогда после отправки сервер будет получать массив отмеченных значений:

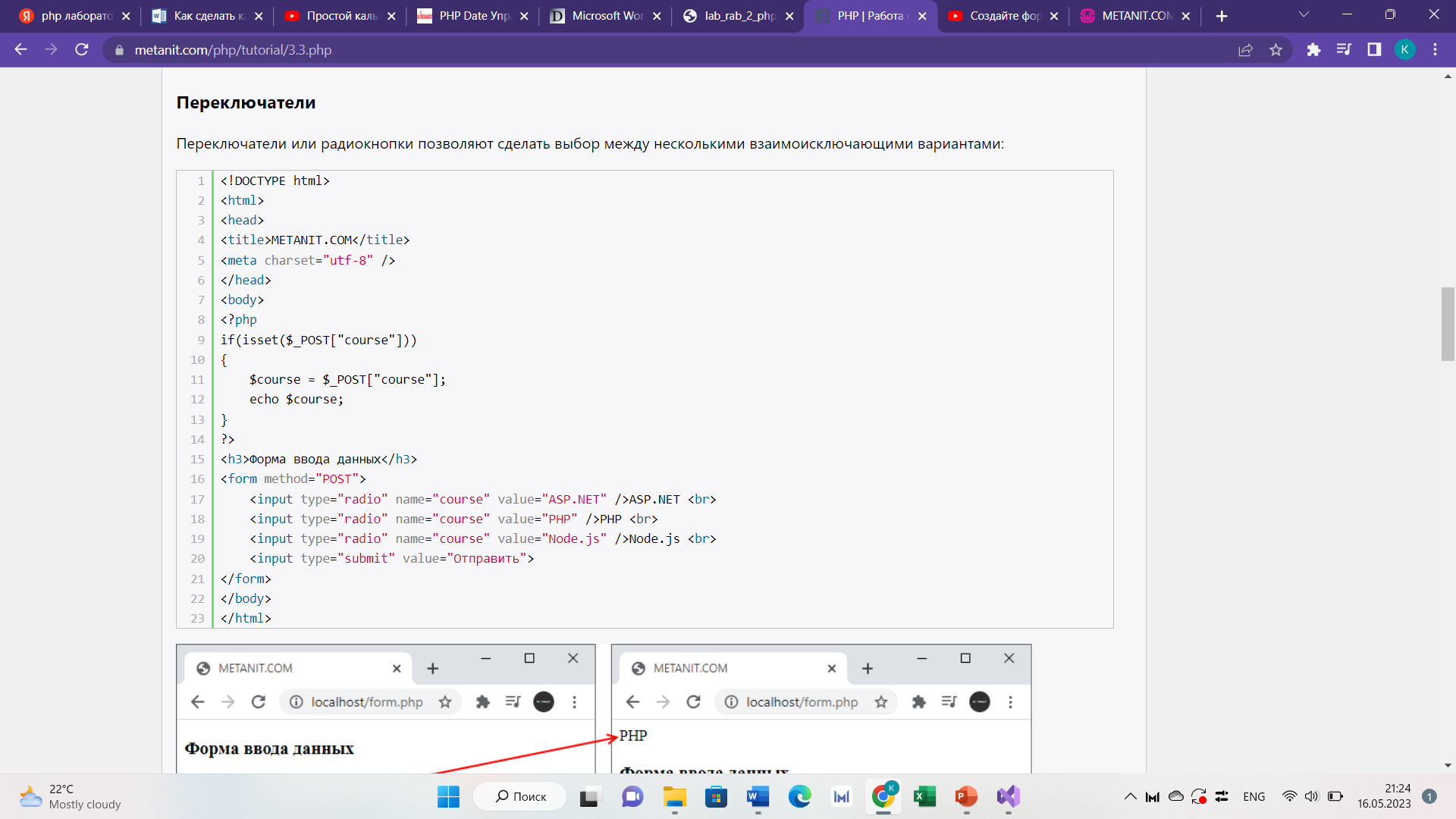


В данном случае переменная $technologies будет представлять массив, который можно перебрать и выполнять все другие операции с массивами.



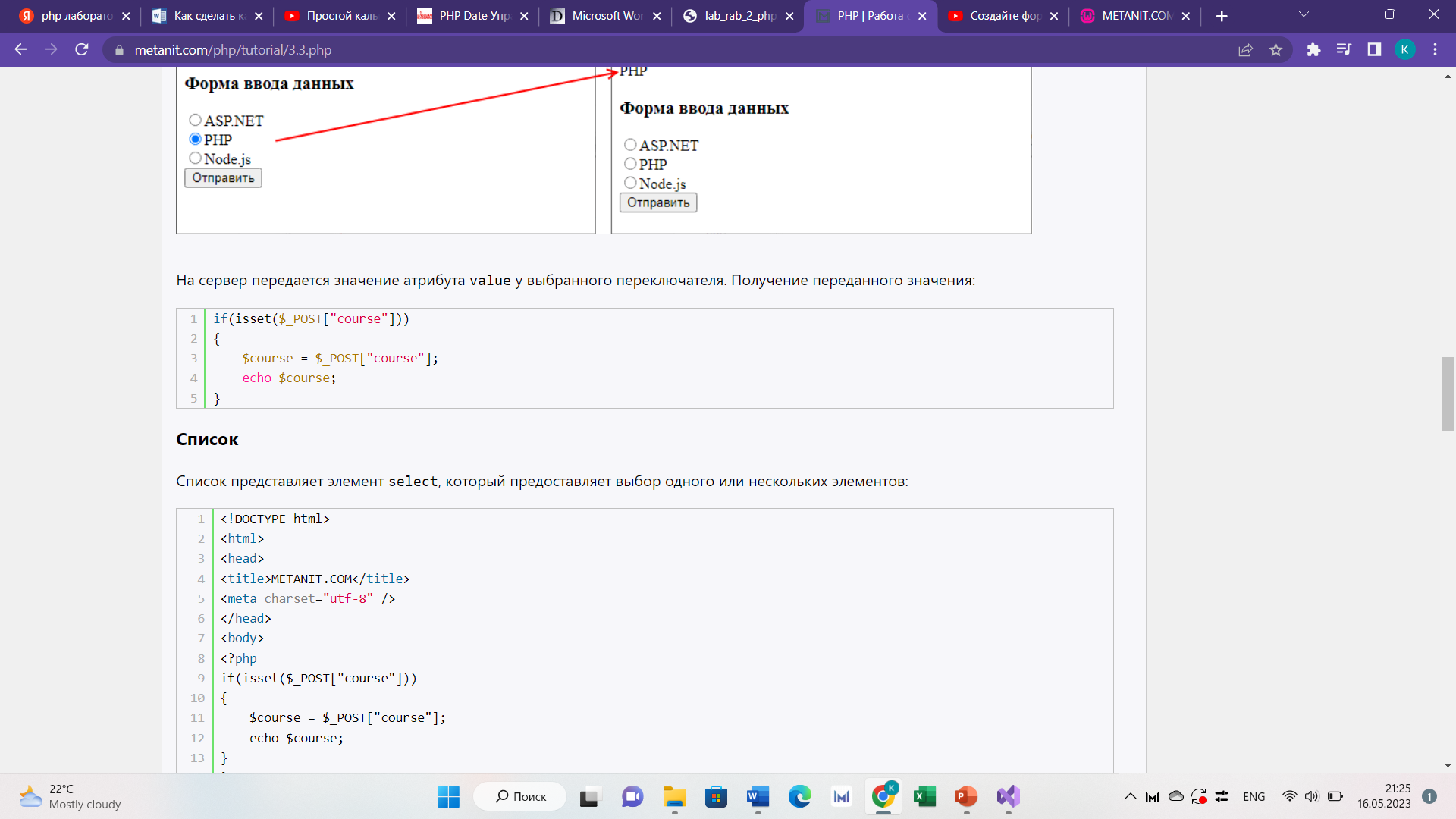
Переключатели

Переключатели или радиокнопки позволяют сделать выбор между несколькими взаимоисключающими вариантами:



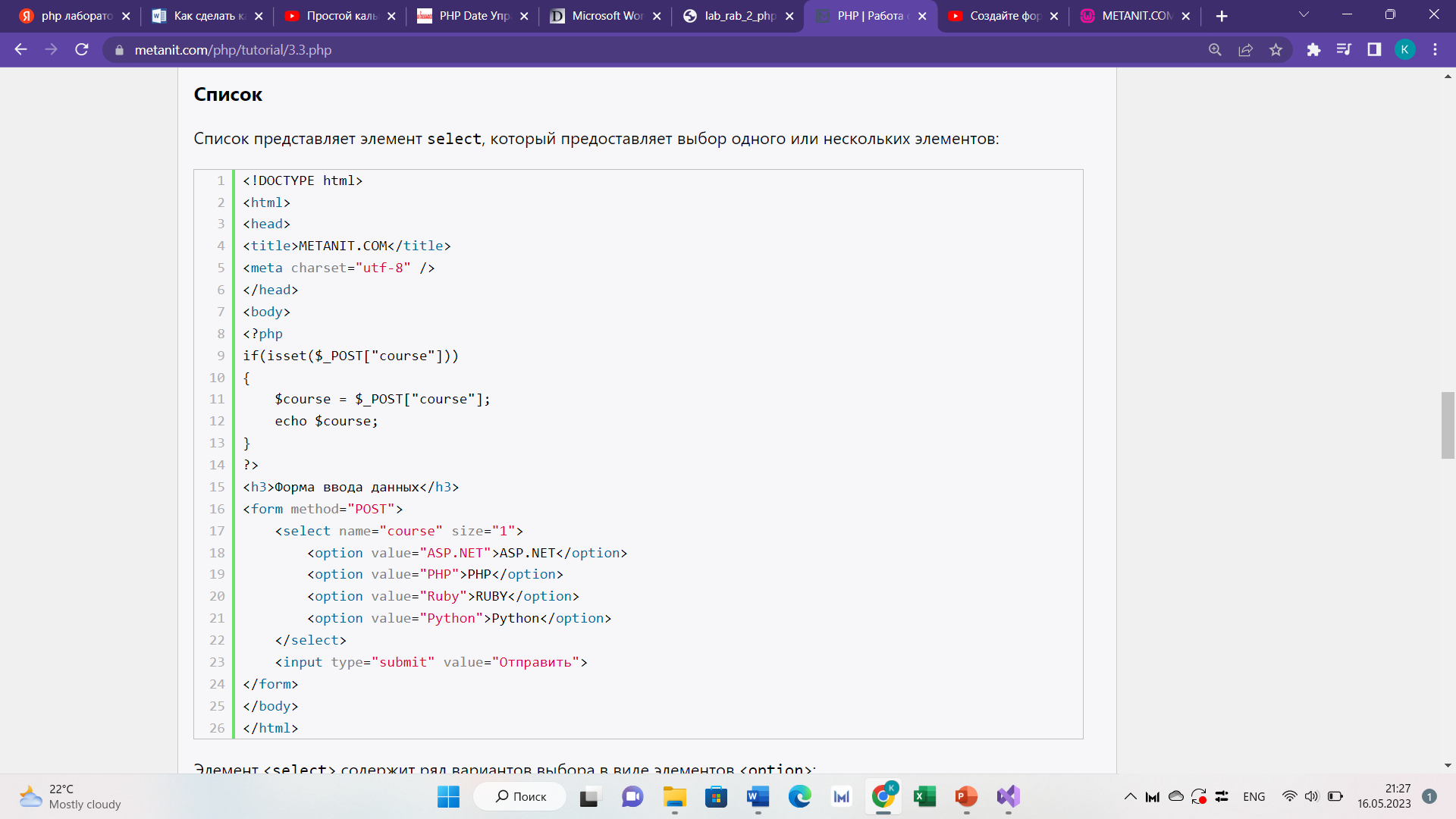


На сервер передается значение атрибута value у выбранного переключателя. Получение переданного значения:

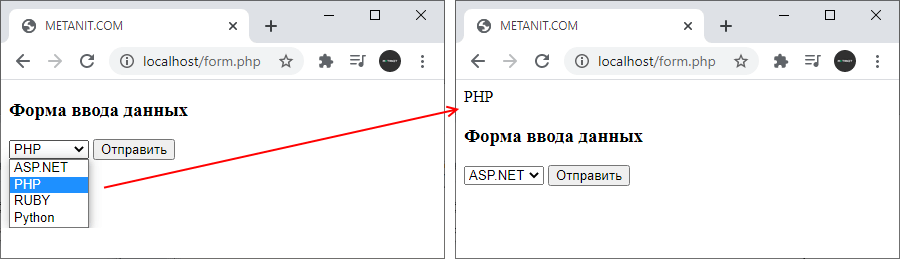


Список

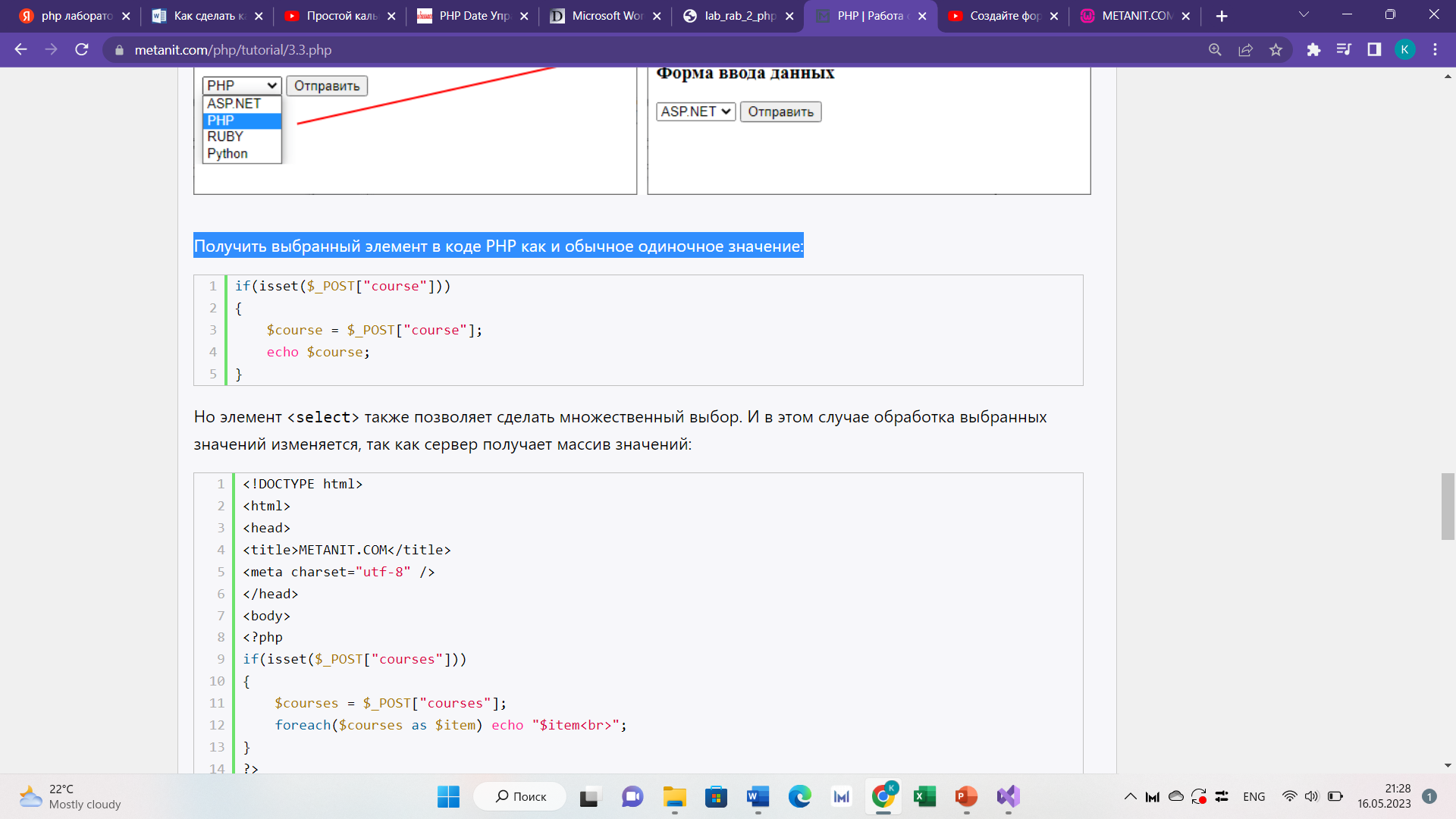
Список представляет элемент select, который предоставляет выбор одного или нескольких элементов:



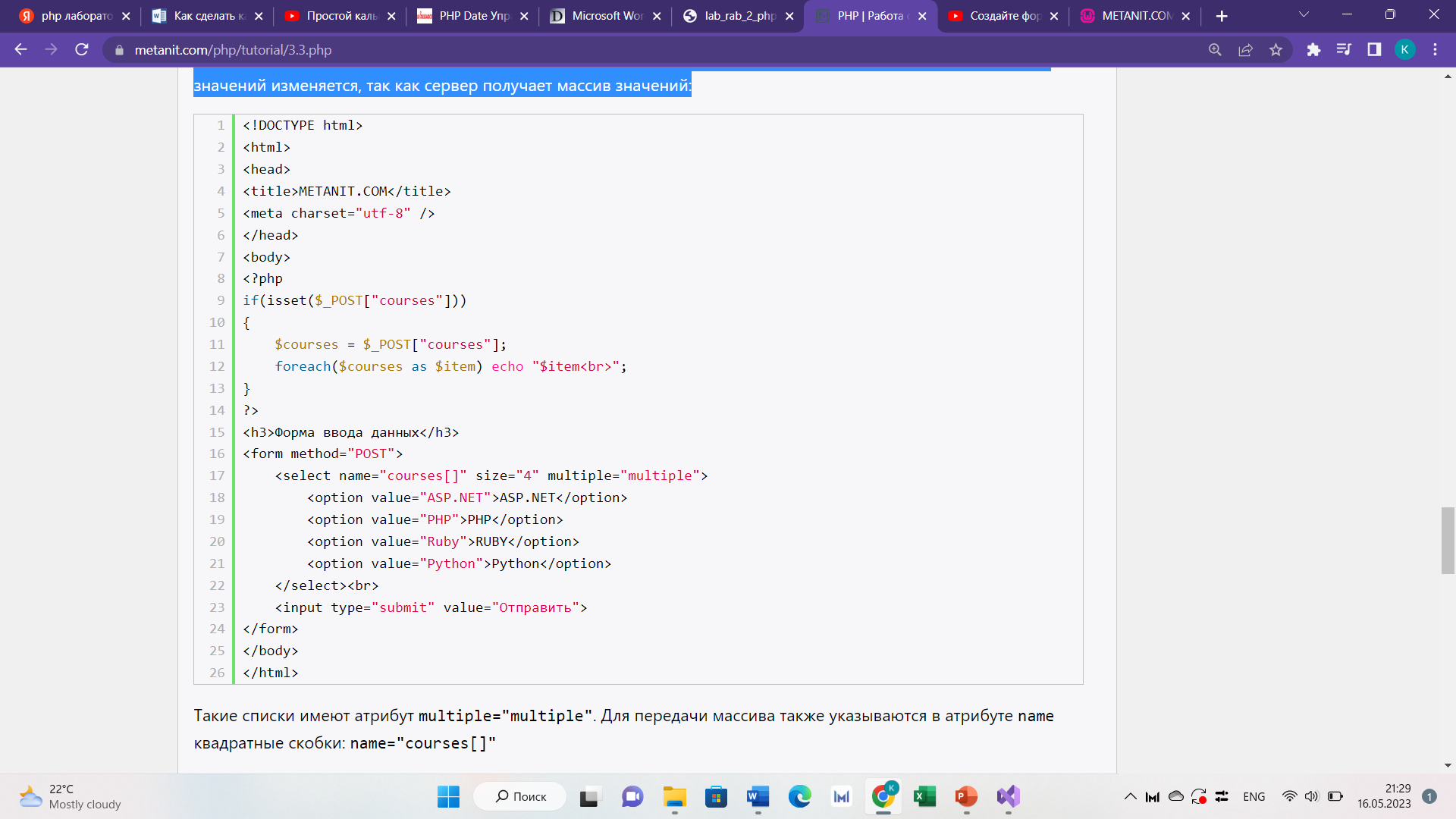
Элемент <select> содержит ряд вариантов выбора в виде элементов <option>:



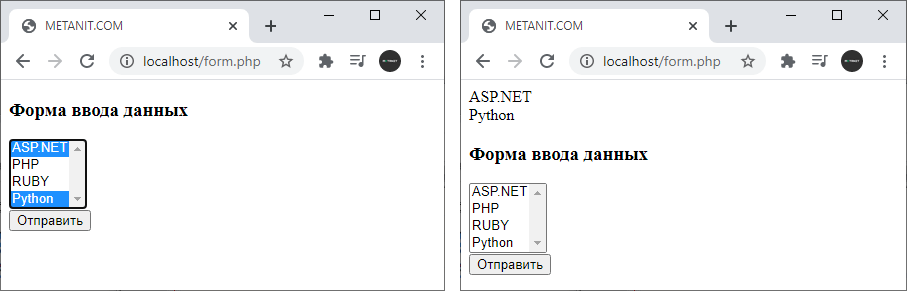
Получить выбранный элемент в коде PHP как и обычное одиночное значение:



Но элемент <select> также позволяет сделать множественный выбор. И в этом случае обработка выбранных значений изменяется, так как сервер получает массив значений:



Такие списки имеют атрибут multiple="multiple". Для передачи массива также указываются в атрибуте name квадратные скобки: name="courses[]"



**Самостоятельная работа**

## 1. Простая форм

Создайте форму с одним полем ввода с запросом любимого города пользователя. Используйте для формы метод "post".

После того, как пользователь щелкнет кнопку отправки, выведите название города, введенное пользователем, сразу же под кнопкой "Отправить".

Подсказка: переменная, содержащая пользовательский ввод, является массивом. Переменная массива - $\_POST['name'], где 'name' - имя вашего поля ввода.

## 2. Интерактивная форма с утверждением If-Else

Возьмите за основу форму с одним полем ввода из предыдущего примера. С помощью условного оператора if-else отобразите, в случае заполнения поля ввода и нажатия кнопки "Отправить", название введенного города. Если поле не заполнено - отобразите форму.

Подсказка: вам понадобится способ узнать, отправлена ​​ли форма. Используйте функцию isset() для определения того, установлена ​​ли переменная и не является ли она нулевым значением.

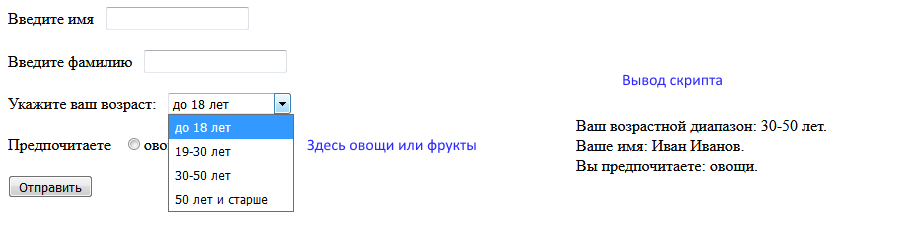
## 3. Интерактивная форма с конструкцией If-Elseif-Else

В этом упражнении PHP вы будете использовать тот же формат, что и в предыдущем упражнении, запрашивая ввод в первой части скрипта и отвечая во второй с помощью оператора if-else PHP. В первом разделе создайте выпадающий список с днями недели и попросите пользователя выбрать день недели.

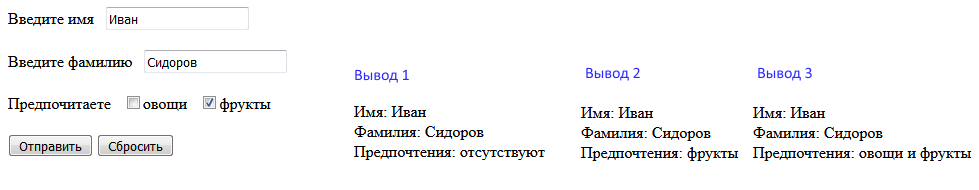
Для второй части скрипта, в зависимости от выбранного дня выведите соответствующую строку:  
"Понедельник - 1-й день недели"  
"Вторник это день второй"  
А за ним идет – среда"  
"За средой идет – четверг"  
"Пятница ударница"  
"День шестой – суббота"  
"Воскресенье - полный выходной"

Используя конструкцию else-elseif-else, настройте каждую строку для вывода в ответ на день выбора пользователем.

4. Создайте форму, состав и внешний вид полей которой показан на скриншоте. Скрипт, который будет обрабатывать данные формы на сервере, должен выводить либо сообщения с данными формы либо просить заполнить все поля формы (см. скриншот задачи). Для проверки заполнения полей имени и фамилии используйте условный оператор if и функцию mb\_strlen(). Не забывайте, что после скачивания файла содержимое папки нужно разместить в корневой папке test нашего учебного сайта на локальном сервере (иначе скрипт работать не будет).



5. Создайте в корневой папке test страницу с такой формой, добавив в нее кнопку «Сбросить». Что касается php-скрипта, то его нужно усложнить. Скрипт должен делать проверку заполнения полей имени и фамилии и в случае ошибки прерывать дальнейшее выполнение скрипта и выводить сообщение. Также скрипт должен проверять, был ли сделан выбор предпочтения и в зависимости от этого делать соответствующий вывод (см. скриншот).



6. Создайте форму, состав и внешний вид полей которой показан на скриншоте. Скрипт, который будет обрабатывать данные формы на сервере, должен проверять заполнение полей имени и фамилии и в случае ошибки выводить предупреждение и прерывать скрипт. Что касается аватарки, то скрипт должен либо вывести аватарку после имени и фамилии, либо сообщить о незагруженной аватарке (см. скриншот задачи). Для получения имени, размера и временной папки для хранения загруженных файлов используйте суперглобальный массив $\_FILES. Для перемещения загруженной аватарки в целевую папку используйте функцию move\_uploaded\_file() (в качестве целевой создайте папку avatars). Чтобы правильно указать путь к вставляемой на страницу аватарке, не забудьте начать его с http://localhost/test/. При желании можете добавить проверку загружаемой аватарки на соответствие типу изображения и т. д.

