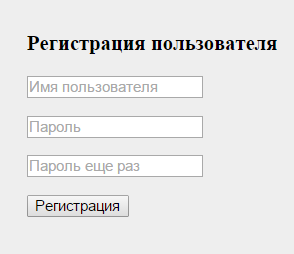
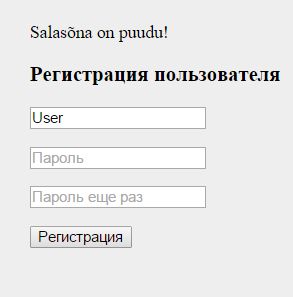
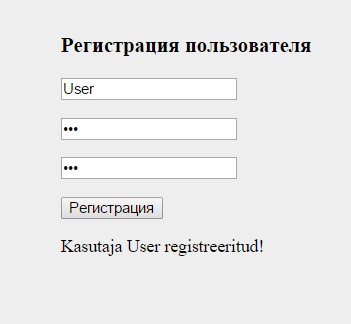
# Тема 5. Создание форм. Использование скриптов ajax

**AJAX, Ajax** (от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.

## *Полезные ссылки*

1. <http://metanit.com/web/jquery/1.1.php>
2. <http://www.stijit.com/javascript/podklyuchit-biblioteku-jquery>
3. <http://veynik.ru/jquery/connecting-jquery-google/>

## Форма, результат

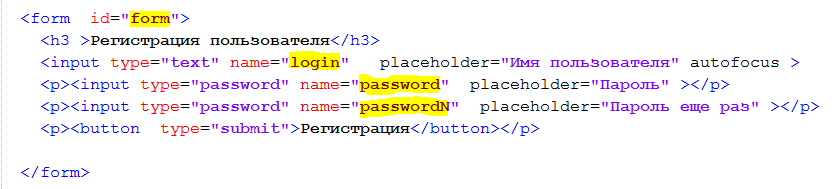
  

## Код. HTML, JS script, PHP

### Форма для ввода данных



div для вывода ошибок

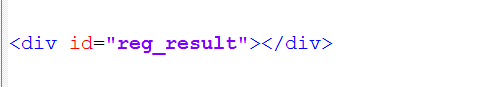


### Форма

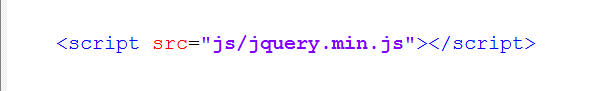
Для формы определяем **id** для обращения к ней в java скрипте, для объектов определяем **name** –для обращения в PHP коде.

В форме или в любое другое место кода добавляем **div** с **id =”login\_result**”. Для вывода сообщений об ошибках.

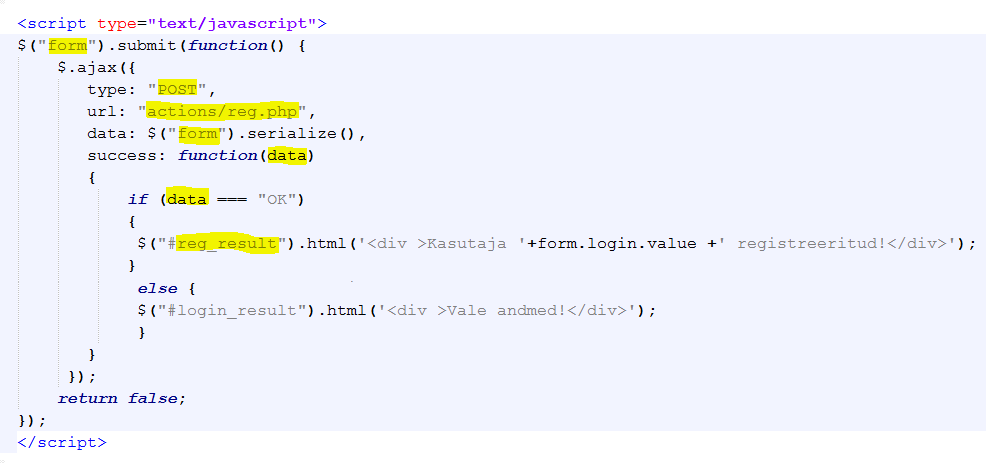
После формы вставляем **div** с **id =”reg\_result”**. Для вывода сообщения о результатах регистрации.



Для работы страницы необходим скрипт. Подключение скрипта:



### Пользовательский скрипт



Тег скрипта: **<script type="text/javascript"> </script>**

**$("form").submit(function()** функция отправки данных на сервер из формы

**success** (функция) – данная функция вызывается после успешного завершения запроса. Функция получает от 1 до 3 параметров (в зависимости от используемой версии библиотеки). Но первый параметр всегда содержит возвращаемые с сервера данные.

**data** (объект/строка) – пользовательские данные, которые передаются на запрашиваемую страницу.

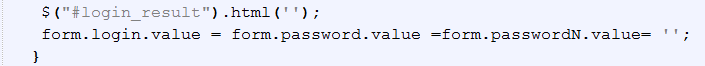
**dataType** (строка) – возможные значения: xml, json, script или html. Описание типа данных, которые ожидаются в ответе сервера.

**type** (строка) – тип запроса. Возможные значения: GET или POST. По умолчанию: GET.

**url** (строка) – адрес URL для запроса.

Полезная ссылка: <http://ruseller.com/lessons.php?rub=32&id=1452>

**Код для очистки окна вывода ошибок и удаления данных формы**



**Полный скрипт обработки данных формы**

<script type="text/javascript">

$("form").submit**(**function() {

**$.ajax({**

type: "POST",

url: "actions/**reg.php**",

data: $("form").serialize(),

success: function(data)

{

if (data === "OK")

{

$("#reg\_result").html('<div >Kasutaja '+form.login.value +' registreeritud!</div>');

$("#login\_result").html('');

form.login.value = form.password.value =form.passwordN.value= '';

}

else if (data==="no login")

{

$("#login\_result").html ('<div >Kasutajanimi on puudu! </div>');

$("#reg\_result").html ('');

}

else if (data==="no pass")

{

$("#login\_result").html ('<div >Salasõna on puudu! </div>');

$("#reg\_result").html('');

}

else if(data==="no passN")

{

$("#login\_result").html('<div >Salasõna on vale! Kontrolli salasõna</div>');

form.password.value =form.passwordN.value= '';

$("#reg\_result").html('');

}

else

{

$("#login\_result").html('<div >Vale andmed!</div>');

$("#reg\_result").html('');

}

}

**}**);

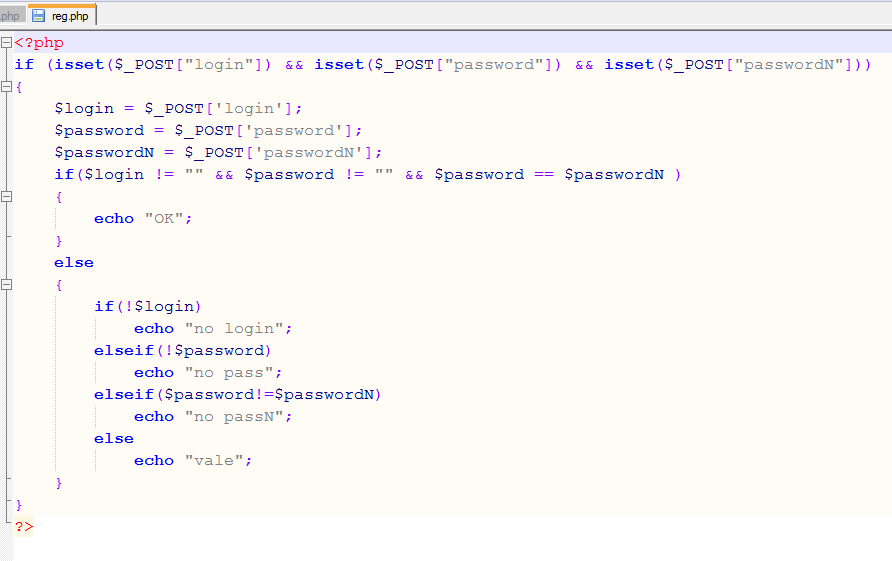
return false;

}**)**;

</script>

PHP файл **actions/reg.php**

Проверяем данные формы, результат –> это сообщение, которое потом проверяется в скрипте выдается сообщение на странице.

****

**Полный код файла**

<?php

if (isset($\_POST["login"]) && isset($\_POST["password"]) && isset($\_POST["passwordN"]))

{

$login = $\_POST['login'];

$password = $\_POST['password'];

$passwordN = $\_POST['passwordN'];

if($login != "" && $password != "" && $password == $passwordN )

{

echo "***OK***";

}

else

{

if(!$login)

echo "***no login***";

elseif(!$password)

echo "***no pass***";

elseif($password!=$passwordN)

echo "no passN";

else

echo "***vale***";

}

}

?>