AJAX

Web.Dev.
Courses

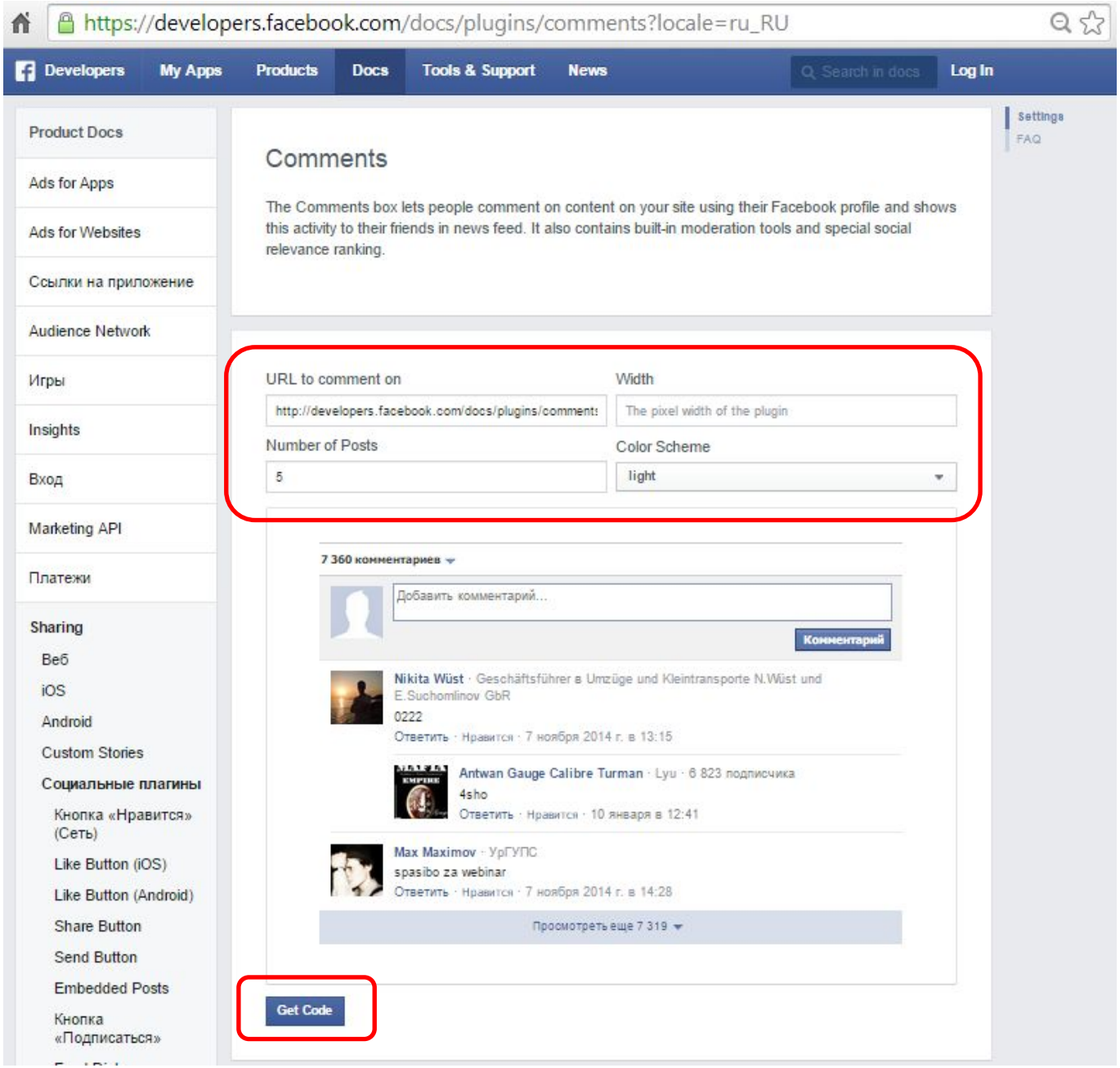
vk.com/web_dev_courses

web.dev.courses.dp.ua/ort/

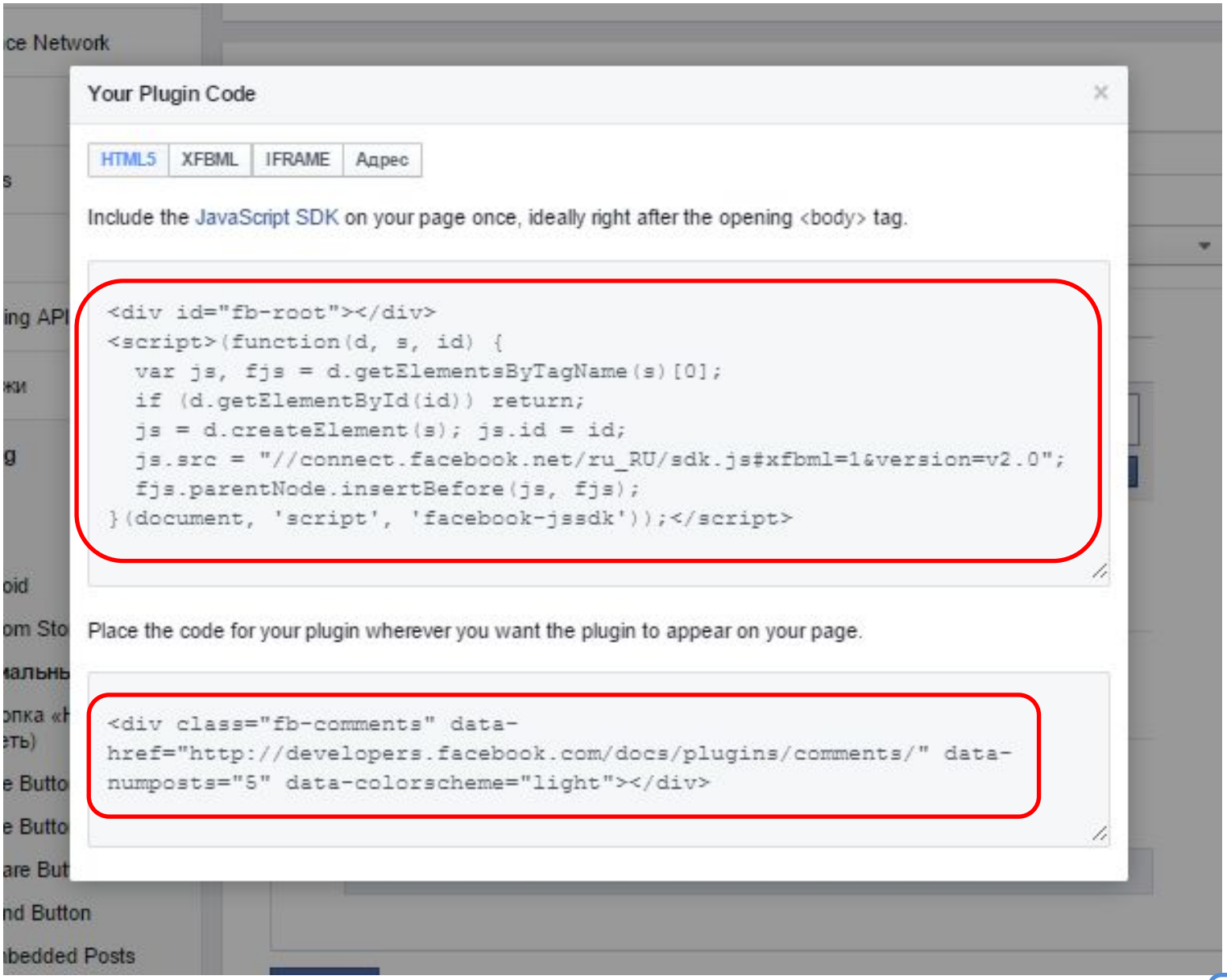
Интеграция

Как вставить на сайт
модуль
комментариев от Facebook.

https://developers.facebook.com/docs/plugins/comments?locale=ru_RU



Генератор кода для вставки на



Генератор кода для вставки на

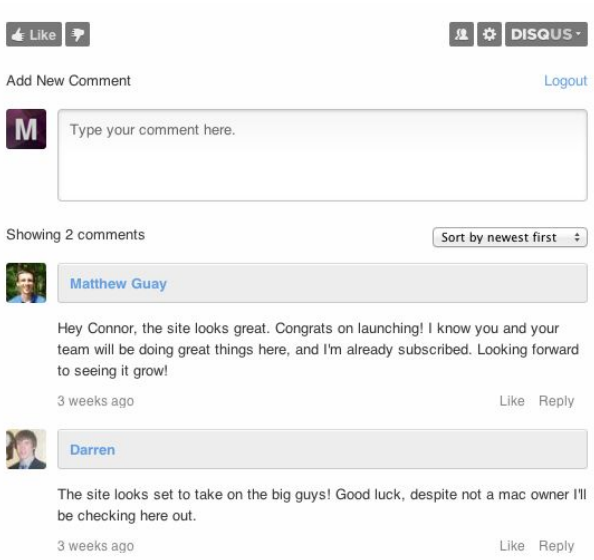
Сгенерированный код необходимо вставить в страницу сайта.

```
<?php get_header(); the_post(); ?>
<div id="fb-root"></div>
<script>(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/ru_RU/sdk.js#xfbml=1&version=v2.0";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));</script>
    <h2><a href="<?php bloginfo('url'); ?>">
    Новости</a> / <?php the_title(); ?></h2>
    <div>
        <span>#<?php the_ID(); ?></span>
        <span class="news-info"><?php the_time('d.m.Y G:i'); ?></span>
        <div><?php the_content(); ?></div>
    </div>
    <div class="fb-comments" data-href="<?php the_permalink(); ?>"
    data-numposts="5" data-colorscheme="light"></div>
<?php get_footer(); ?>
```

Модернизированная версия:
single.php

Домашнее задание

Познакомится с Disqus – один из лучших механизмов комментариев для сайтов.



https://disqus.com

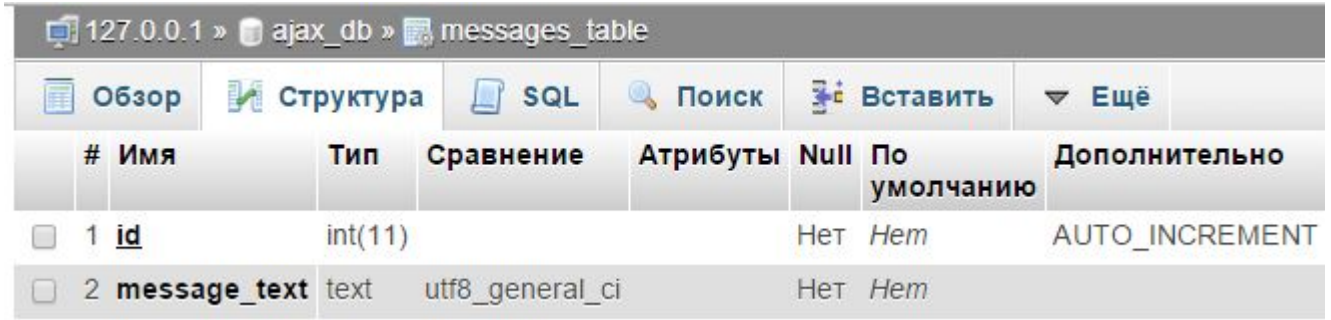
Вариант #1

Подготовьте базу данных и проект в Денвере

Название: **«ajax_db»**

Таблица: **«messages_table»**

Структура:



SQL:

```
CREATE TABLE IF NOT EXISTS `messages_table` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`message_text` text NOT NULL,  PRIMARY KEY (`id`))
ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=77 ;
```

Каталог в Денвер:

ajax.dp.ua/www

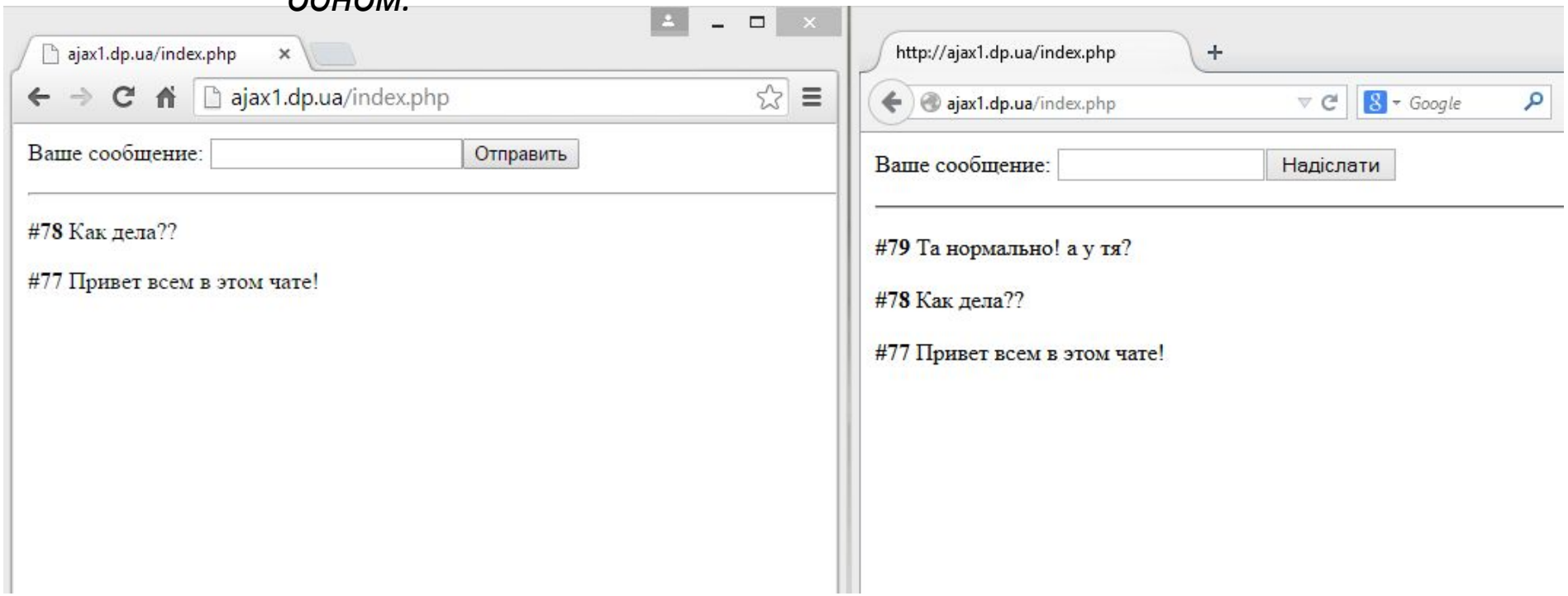
Простейший чат

```php
<?php //Сохраните файл как UTF-8 (без BOM).
    mysql_connect('localhost', 'root', '');
    mysql_select_db('ajax_db');

    if(strlen($_REQUEST['sms']) > 1){
        $sms = mysql_real_escape_string($_REQUEST["sms"]);
        mysql_query("INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')");
        header('Location: ./index.php');
    }

    $result = mysql_query("SELECT * FROM `messages_table` ORDER BY `id` DESC LIMIT 20");
    header('Content-Type: text/html; charset=utf-8');
?>
<html><head>

<style> body{ width:800px; margin:0 auto; padding: 10px; } </style></head>
<body>
    <form>
        Ваше сообщение: <input type="text" name="sms"><input type="submit">
    </form>
    <hr />
    <div id="messages">
        <?php while($one_message = mysql_fetch_array($result)){ ?>
        <p>
            <b>#<?php echo $one_message['id']; ?></b>
            <?php echo $one_message['message_text']; ?>
        </p>
        <?php } ?>
    </div>
</body></html>
```

index.php

Простейший чат

Откройте в двух браузерах, напишите сообщение в одном.



***В чём проблема**: при отправке сообщения, другие участники чата получат сообщение только когда обновят страницу (самостоятельно или после отправки сообщения).*

Простейший чат+

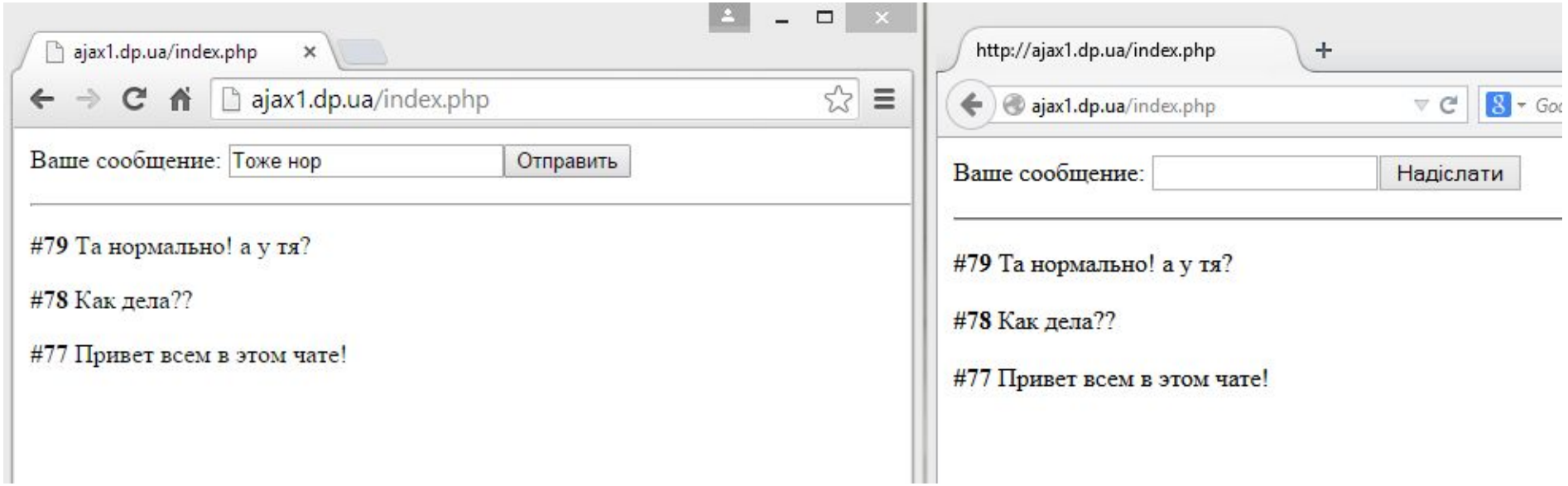
16.1a

```php
<?php //Сохраните файл как UTF-8 (без BOM).
    mysql_connect('localhost', 'root', '');
    mysql_select_db('ajax_db');

    if(strlen($_REQUEST['sms']) > 1){
        $sms = mysql_real_escape_string($_REQUEST["sms"]);
        mysql_query("INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')");
        header('Location: ./index.php');
    }

    $result = mysql_query("SELECT * FROM `messages_table` ORDER BY `id` DESC LIMIT 20");
    header('Content-Type: text/html; charset=utf-8');
?>
<html><head>
<META HTTP-EQUIV="refresh" CONTENT="5">
<style> body{ width:800px; margin:0 auto; padding: 10px; } </style></head>
<body>
    <form>
        Ваше сообщение: <input type="text" name="sms"><input type="submit">
    </form>
    <hr />
    <div id="messages">
        <?php while($one_message = mysql_fetch_array($result)){ ?>
        <p>
            <b>#<?php echo $one_message['id']; ?></b>
            <?php echo $one_message['message_text']; ?>
        </p>
        <?php } ?>
    </div>
</body></html>
```

index.php

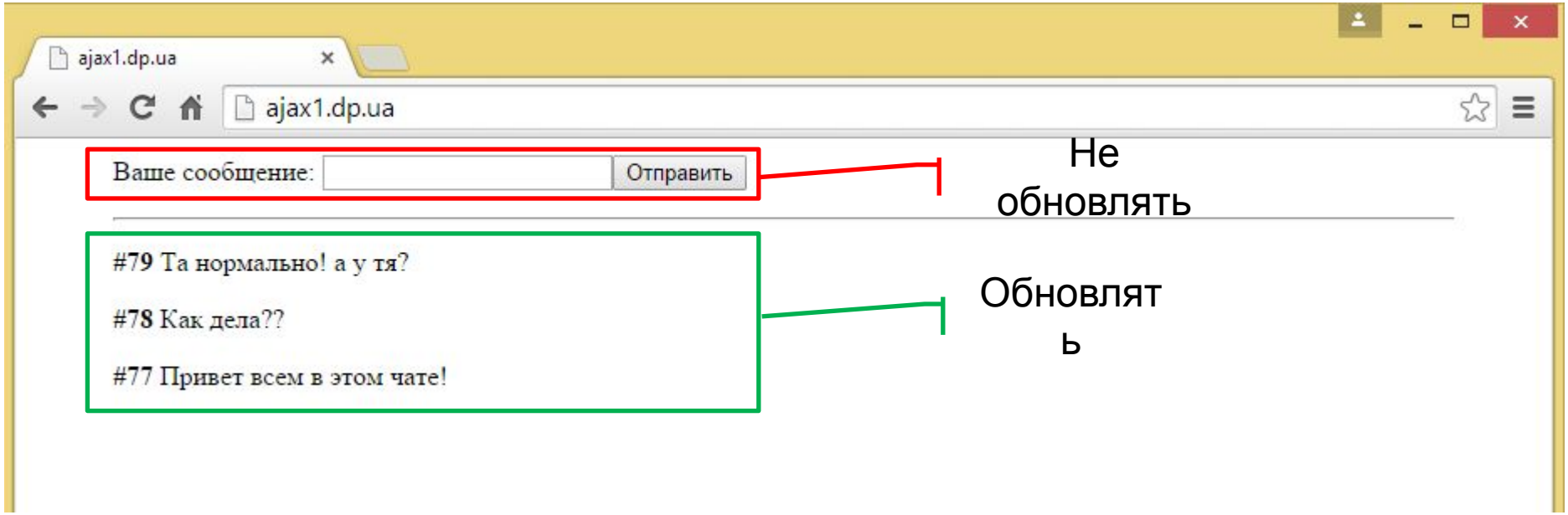
Простейший чат+

Откройте в двух браузерах, напишите сообщение в одном.



***В чём теперь проблема**: при отправке сообщения, другие участники чата получают его автоматически, но теперь проблематично набрать сообщение.*

Простейший чат+



Хорошо бы обновлять только часть страницы не затрагивая другую.

Вариант #2

Простейший чат 2.0

16.2

```php
<?php
    header('Content-Type: text/html; charset=utf-8');

    mysql_connect('localhost', 'root', '');
    mysql_select_db('ajax_db');

    $result = mysql_query("SELECT * FROM `messages_table` ORDER BY `id` DESC LIMIT 20");

    while($one_message = mysql_fetch_array($result)){
?>
    <p>

        <b>#<?php echo $one_message['id']; ?></b>

        <?php echo $one_message['message_text']; ?>

    </p>
<?php } ?>
```

get_data.php

Простейший чат 2.0

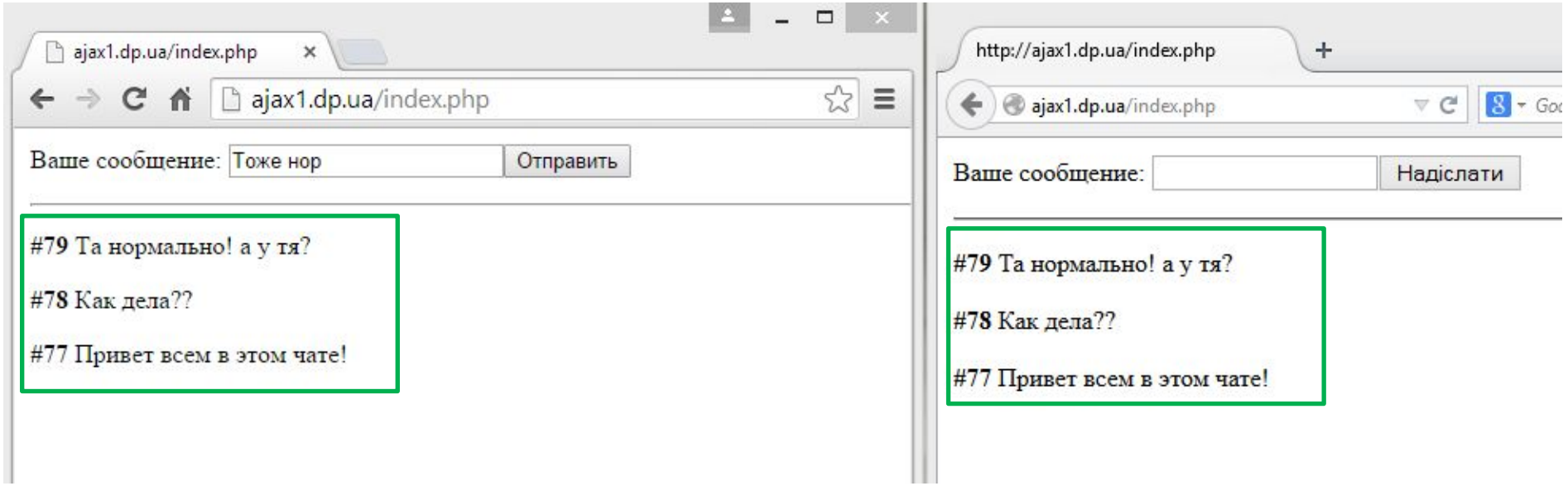
16.3

```php
<?php //Сохраните файл как UTF-8 (без BOM).
    mysql_connect('localhost', 'root', '');
    mysql_select_db('ajax_db');

    if(strlen($_REQUEST['sms']) > 1){
        $sms = mysql_real_escape_string($_REQUEST["sms"]);
        mysql_query("INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')");
        header('Location: ./index.php');
    }
    header('Content-Type: text/html; charset=utf-8');
?>
<html><head>
<script>
    window.onload = function(){
        async_load_data();
        setInterval(async_load_data, 3000);
    };

    function async_load_data(){
        var XHR = new XMLHttpRequest();
        XHR.onload = function(){
            document.getElementById("messages").innerHTML = XHR.responseText;
        };
        XHR.open("get", "get_data.php", true);
        XHR.send();
    };
</script>
<style> body{ width:800px; margin:0 auto; padding: 10px; } </style></head>
<body>
    <form>
        Ваше сообщение: <input type="text" name="sms"><input type="submit">
    </form>
    <hr />
    <div id="messages">
    </div>
</body></html>
```

index.php

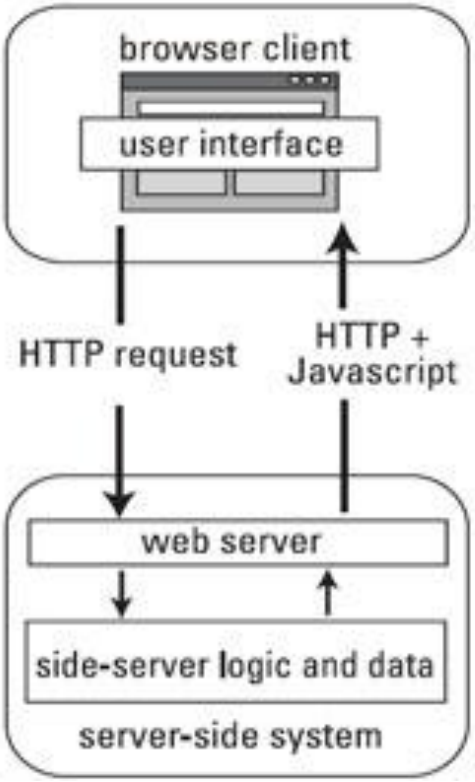
Простейший чат 2.0

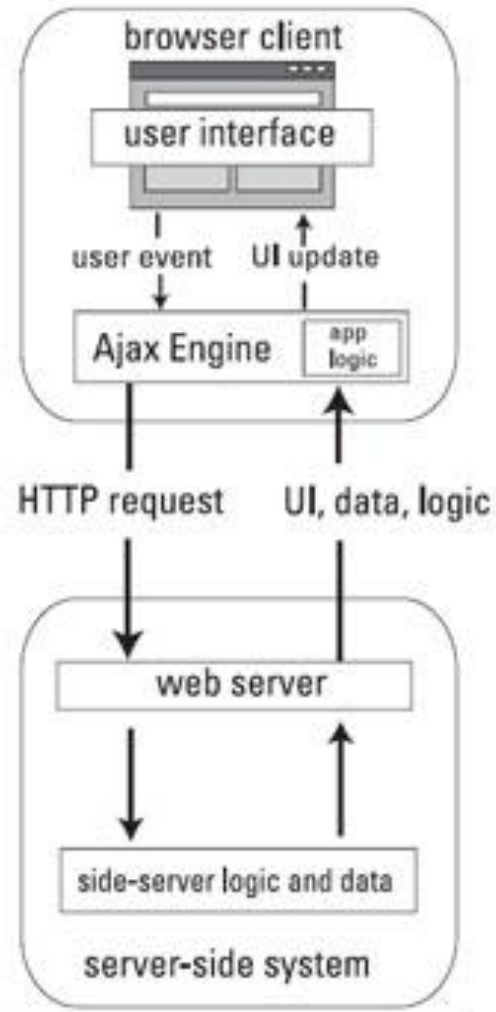
Откройте в двух браузерах, напишите сообщение в одном.

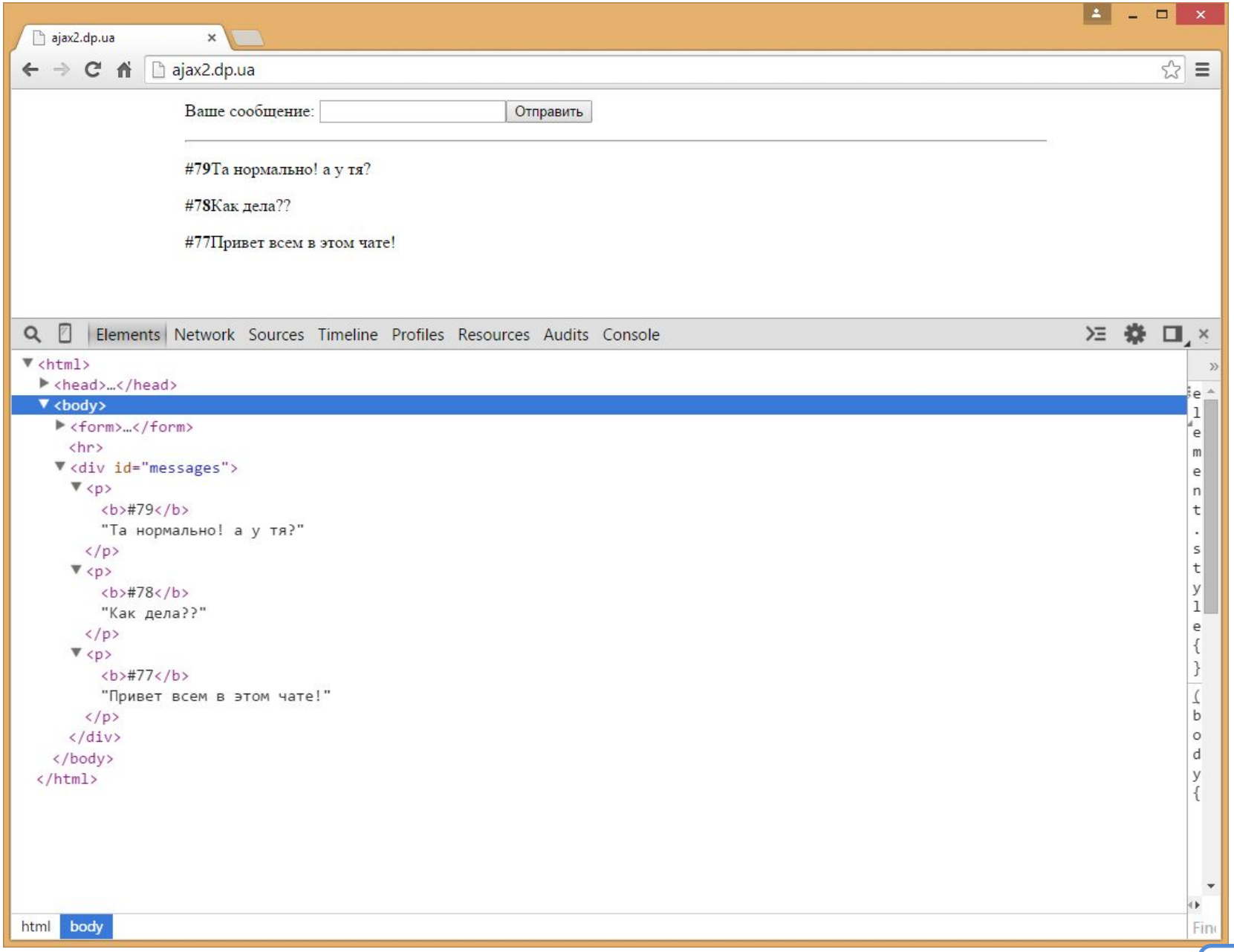


Теперь часть страницы обновляется каждый 3 секунды, тем самым обеспечивая подгрузку свежих сообщений, при этом ничего не мешает набирать сообщение.

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. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.






ajax2.dp.ua
ajax2.dp.ua
Ваше сообщение:
Отправить
#79Та нормально! а у тя?
#78Как дела??
#77Привет всем в этом чате!
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<html>
<head>…</head>
<body>
<form>…</form>
<hr>

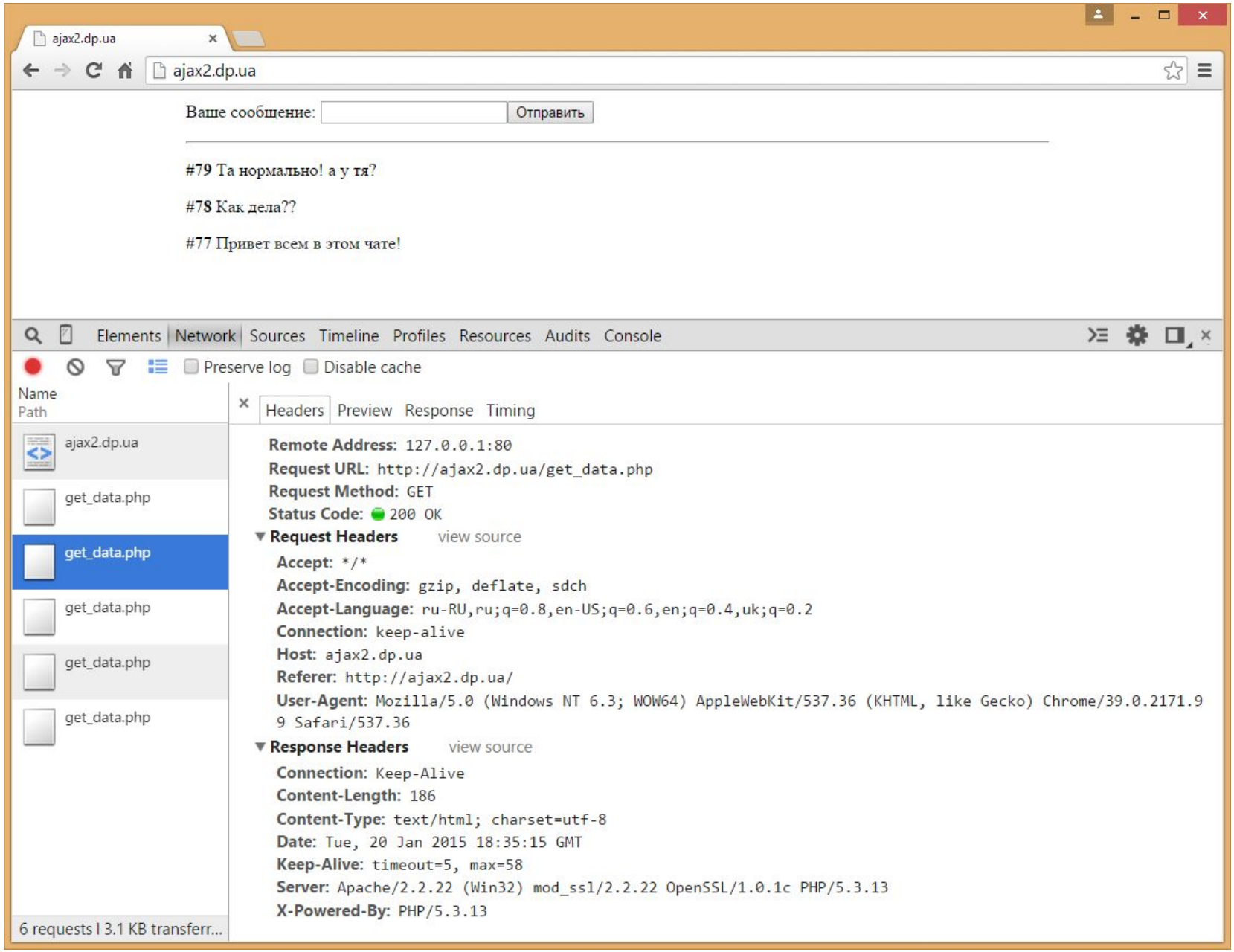
#79
"Та нормально! а у тя?"

#78
"Как дела??"

#77
"Привет всем в этом чате!"

</body>
</html>
element.style {
body {
html
body

AJAX в



AJAX в действии

Вариант #3

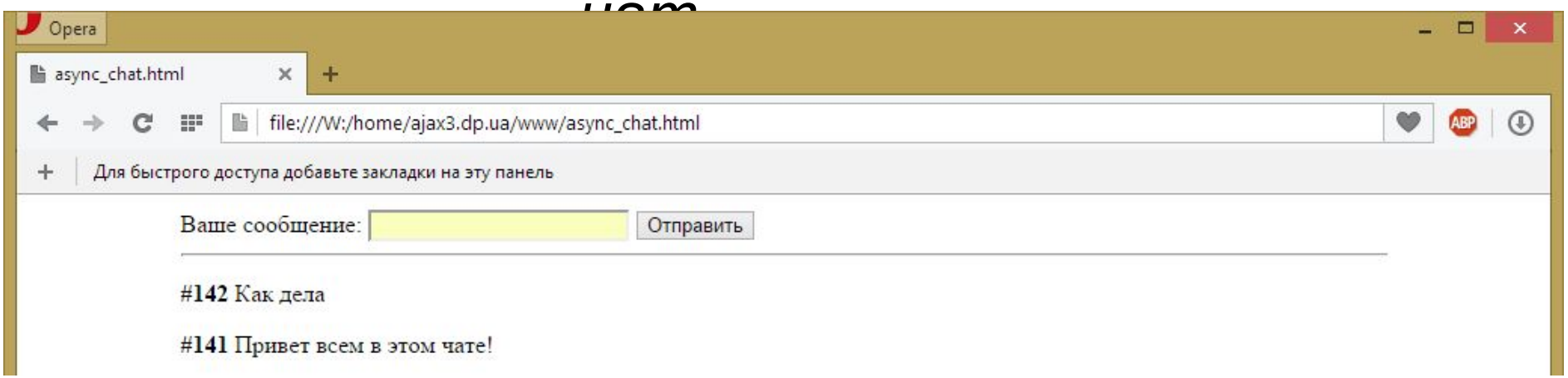
Реальный чат в чистом виде, без jQuery

Создайте обычный HTML файл в UTF-8 с

```
<html><head>
<script>
    window.onload = function(){
        async_load_data(); //Первичная загрузка данных
        setInterval(async_load_data, 3000);
        document.getElementById("send_btn").onclick = async_send_data;
    };

    function async_send_data(){
        btn = document.getElementById('send_btn');
        sms = document.getElementById("sms");

        btn.disabled = true;

        var virtual_from = new FormData();
        virtual_from.append("sms", sms.value);

        var XHR = new XMLHttpRequest();
        XHR.onload = function(){
            sms.value = "";
            btn.disabled = false;
            async_load_data();
        };
        XHR.open("post", "http://web.dev.courses.dp.ua/ort/ajax/add_data.php", true);
        XHR.send(virtual_from);
    };

    function async_load_data(){
        var XHR = new XMLHttpRequest();
        XHR.onload = function(){
            document.getElementById("messages").innerHTML = XHR.responseText;
        };
        XHR.open("get", "http://web.dev.courses.dp.ua/ort/ajax/get_data.php", true);
        XHR.send();
    };
</script>
<style> body{ width:800px; margin:0 auto; padding: 10px; } </style></head>
<body>
    <div>Ваше сообщение: <input type="text" id="sms"> <input id="send_btn" type="button" value="Отправить"></div>
    <hr />
    <div id="messages"></div>
</body></html>
```

async_chat.html

Реальный чат



Час работает с заготовленным API.

Интерфейс программирования приложений (англ. **application programming interface, API**) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

API реального чата

web.dev.courses.dp.ua/ort/ajax/get_data.php

Возвращает текстовую HTML разметку содержащую 20 последних сообщений в чате (отсоритрованные по новизне) в приведенном формате:

```
<p> <b> #Номер_сообщения </b> Текст сообщения </p>
<p> <b> #56 </b> Привет всем!!! </p>
```

Исходный

```php
<?php
    header('Content-Type: text/html; charset=utf-8');
    header('Access-Control-Allow-Origin: *');

    mysql_connect('localhost', 'root', '');
    mysql_select_db('courses_chat_db');

    $result = mysql_query("SELECT * FROM `messages_table` ORDER BY `id` DESC LIMIT 20");

    while($one_message = mysql_fetch_array($result)){
?>
    <p>
        <b>#<?php echo $one_message['id']; ?></b>
        <?php echo $one_message['message_text']; ?>
    </p>
<?php } ?>
```

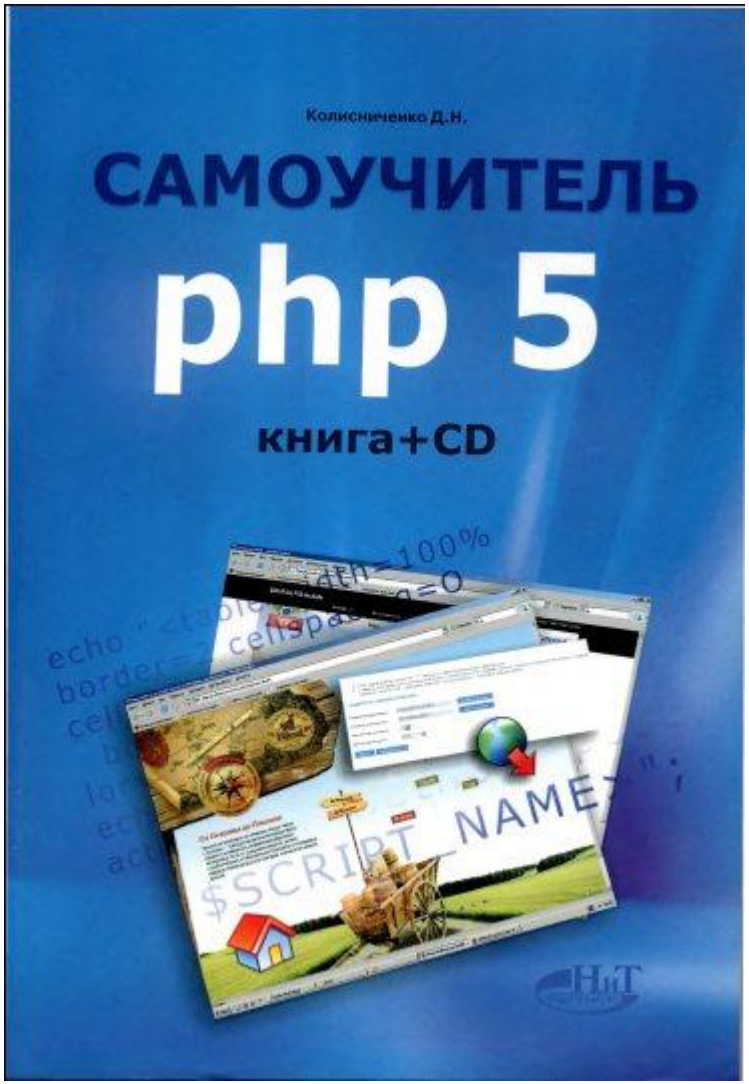
API реального чата

web.dev.courses.dp.ua/ort/ajax/add_data.php

Принимает новое сообщение в чат и заносит его в базу данных, имя принимаемого параметра ‘*sms*’ метод передачи параметра GET и POST. В случае успешного добавления возвращается текст ‘OK’.

Исходный код:

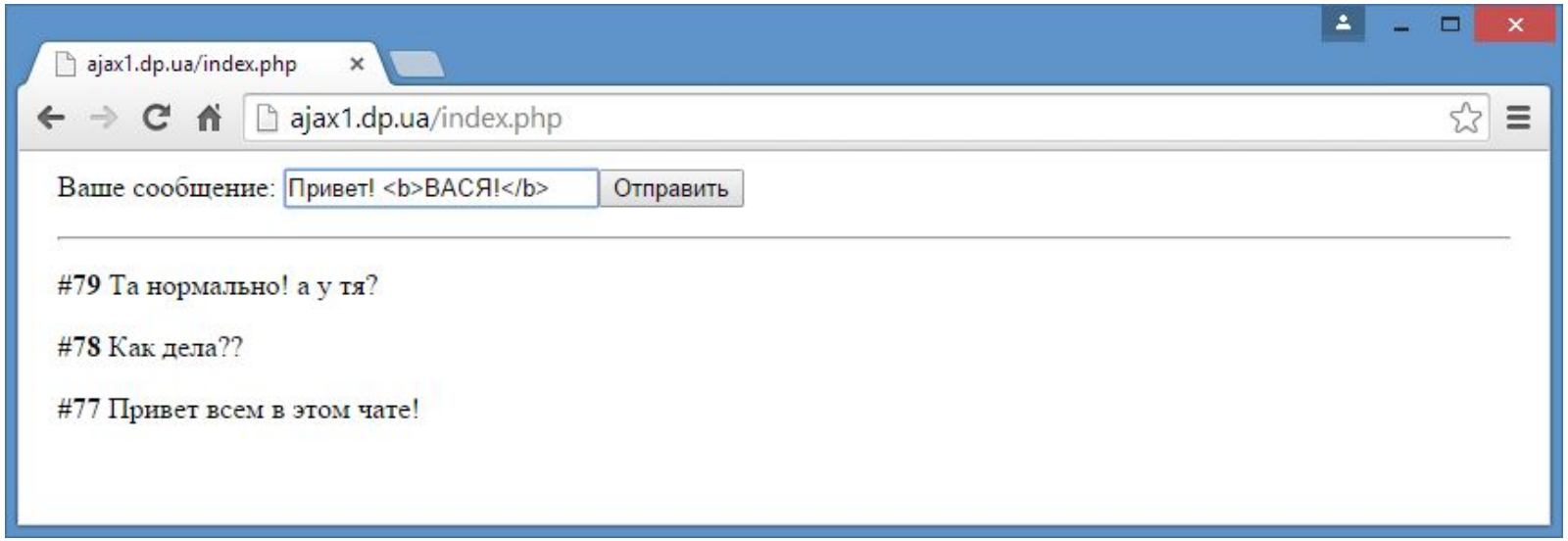
```php
<?php
    mysql_connect('localhost', '', '');
    mysql_select_db('courses_chat_db');

    if(strlen($_REQUEST['sms']) > 1){
        $sms = mysql_real_escape_string($_REQUEST["sms"]);
        mysql_query("INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')");
    }

    header('Content-Type: text/html; charset=utf-8');
    header('Access-Control-Allow-Origin: *');

    echo "ok";
?>
```

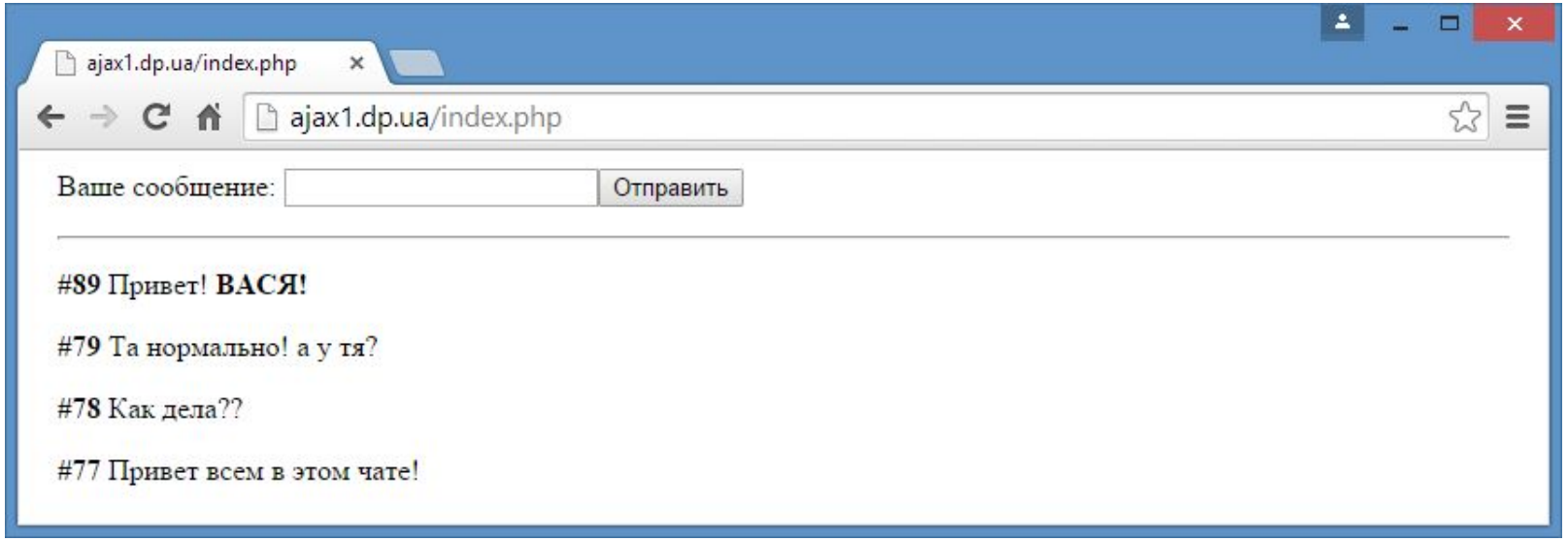


PHP и MySQL. Разработка Web-приложений.
Денис Колисниченко

Безопасность

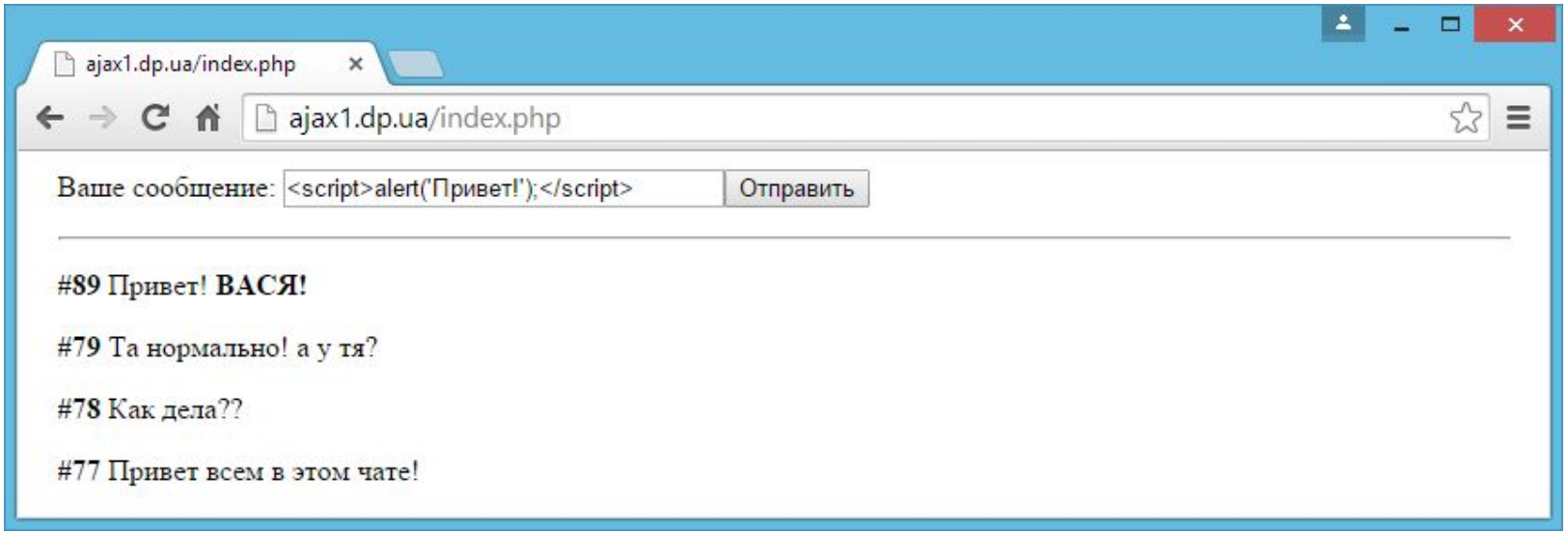
Инъекции кода – Code

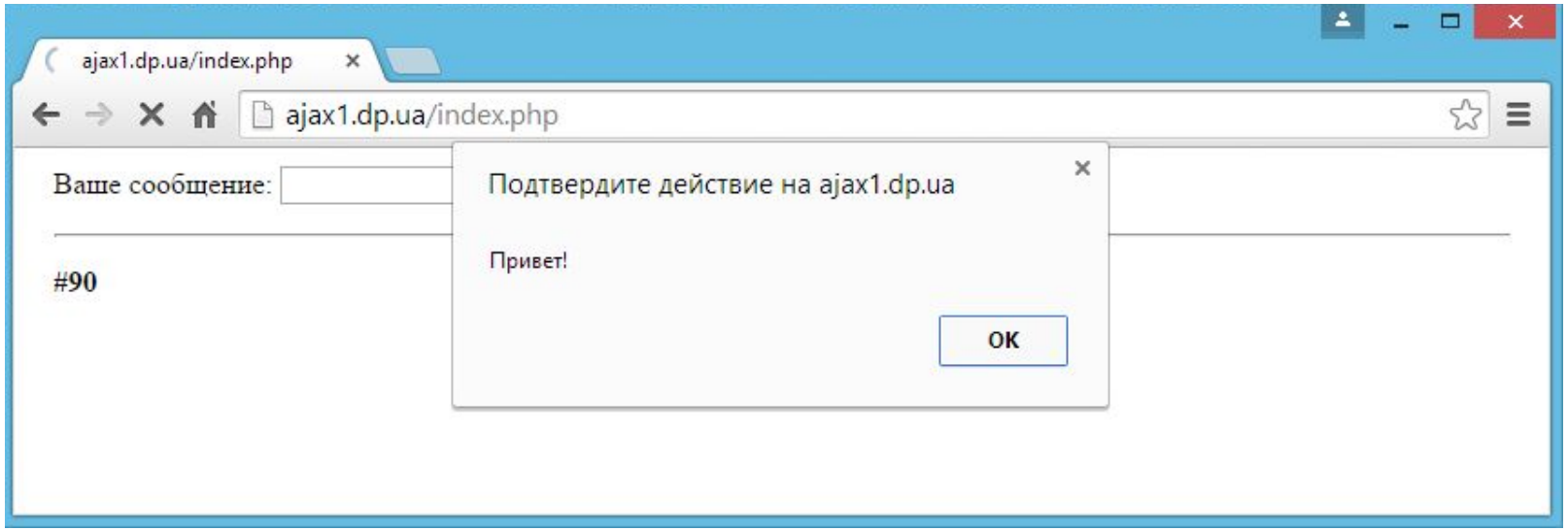




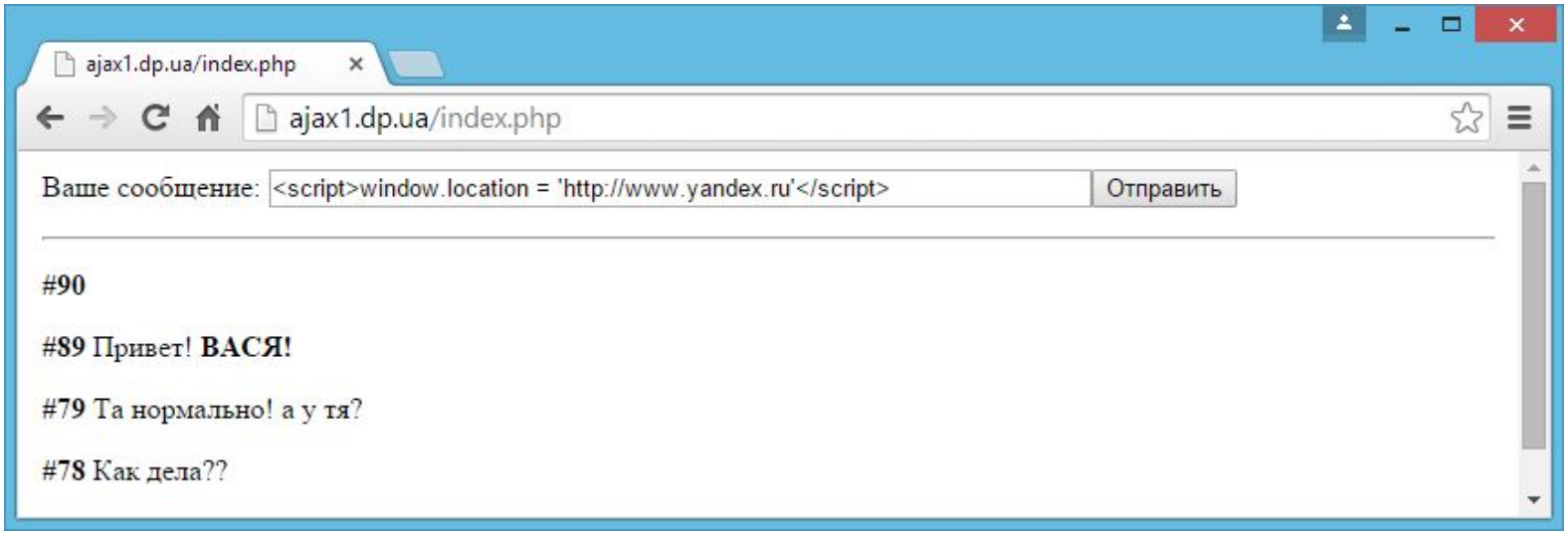
Инъекции кода – Code injection

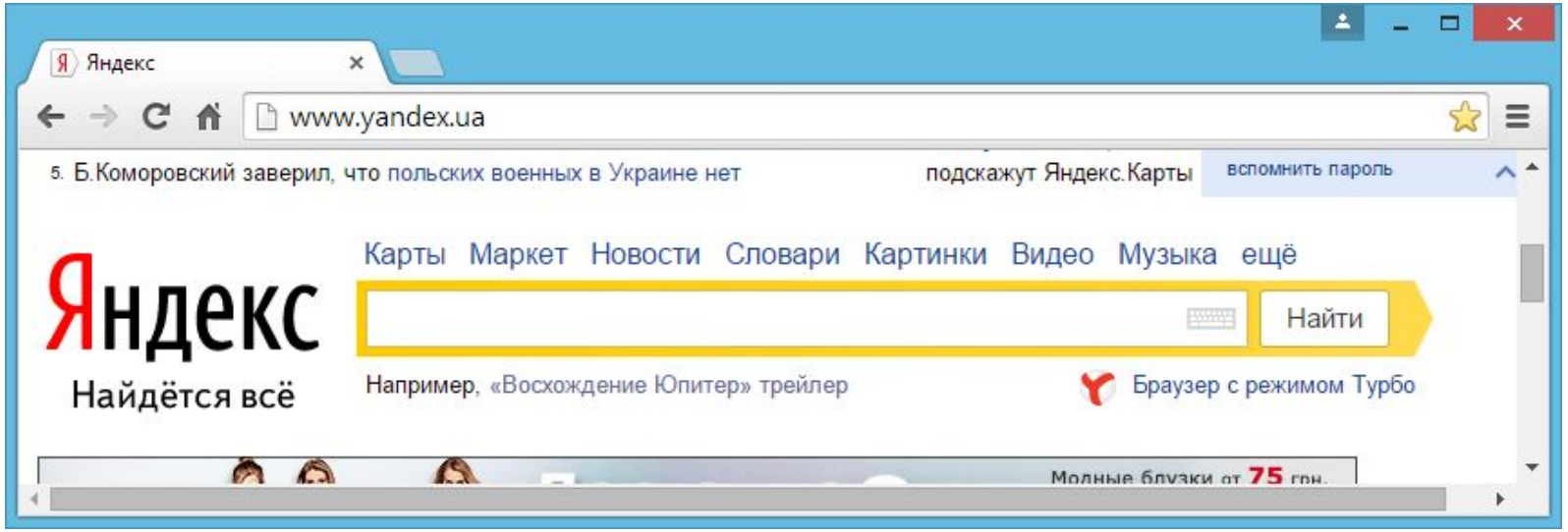
Инъекции кода – Code





Инъекции кода – Code





Инъекции кода – Code injection ***Какая защита?***

<script>alert('Hi!');</script>

⇩

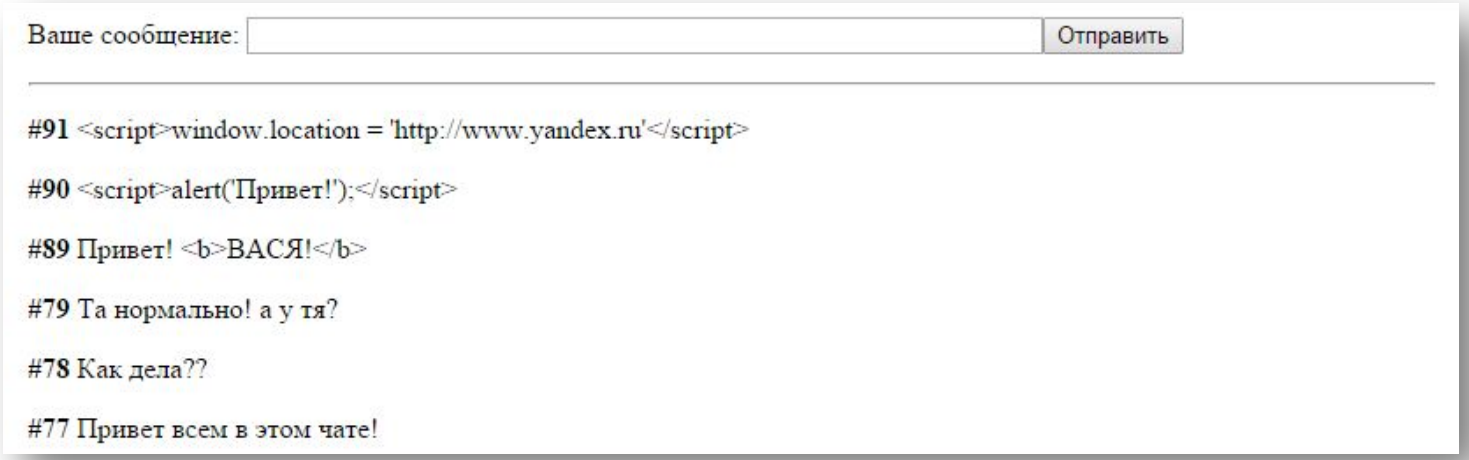
php **htmlspecialchars()**

⇩

<script>alert('Привет!');</script>

```
24      <p>
25          <b>#<?php echo $one_message['id']; ?></b>
26          <?php echo htmlspecialchars($one_message['message_text']); ?>
27      </p>
```

⇩



Инъекция SQL – SQL injection

```
$sms = $_REQUEST["sms"];
$query = "INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')"
```

Ваше сообщение: xx'); DROP TABLE messages_table; -- Отправить

```
"INSERT INTO `messages_table` (`message_text`) VALUES ('xx'); DROP TABLE
messages_table; -- ')"
```

Какая

```
5   if(strlen($_REQUEST['sms']) > 1){
6       $sms = mysql_real_escape_string($_REQUEST["sms"]);
7       mysql_query("INSERT INTO `messages_table` (`message_text`) VALUES ('$sms')");
8       header('Location: ./index.php');
9   }
```

xx'); DROP TABLE messages_table; --

xx\'); DROP TABLE messages_table; --

Cookie/Сессии/Параметры в URL

На стороне клиента можно хранить только его