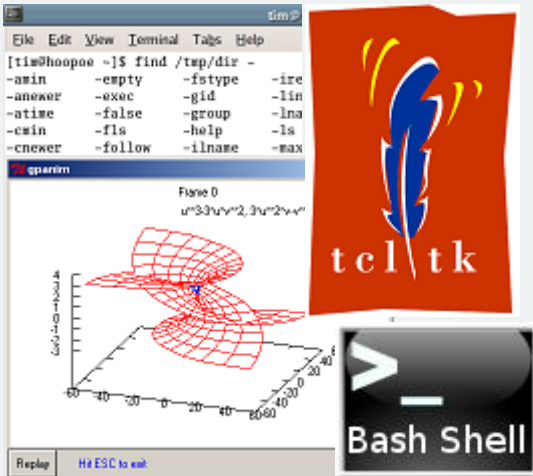




# Интерпретируемые языки программирования

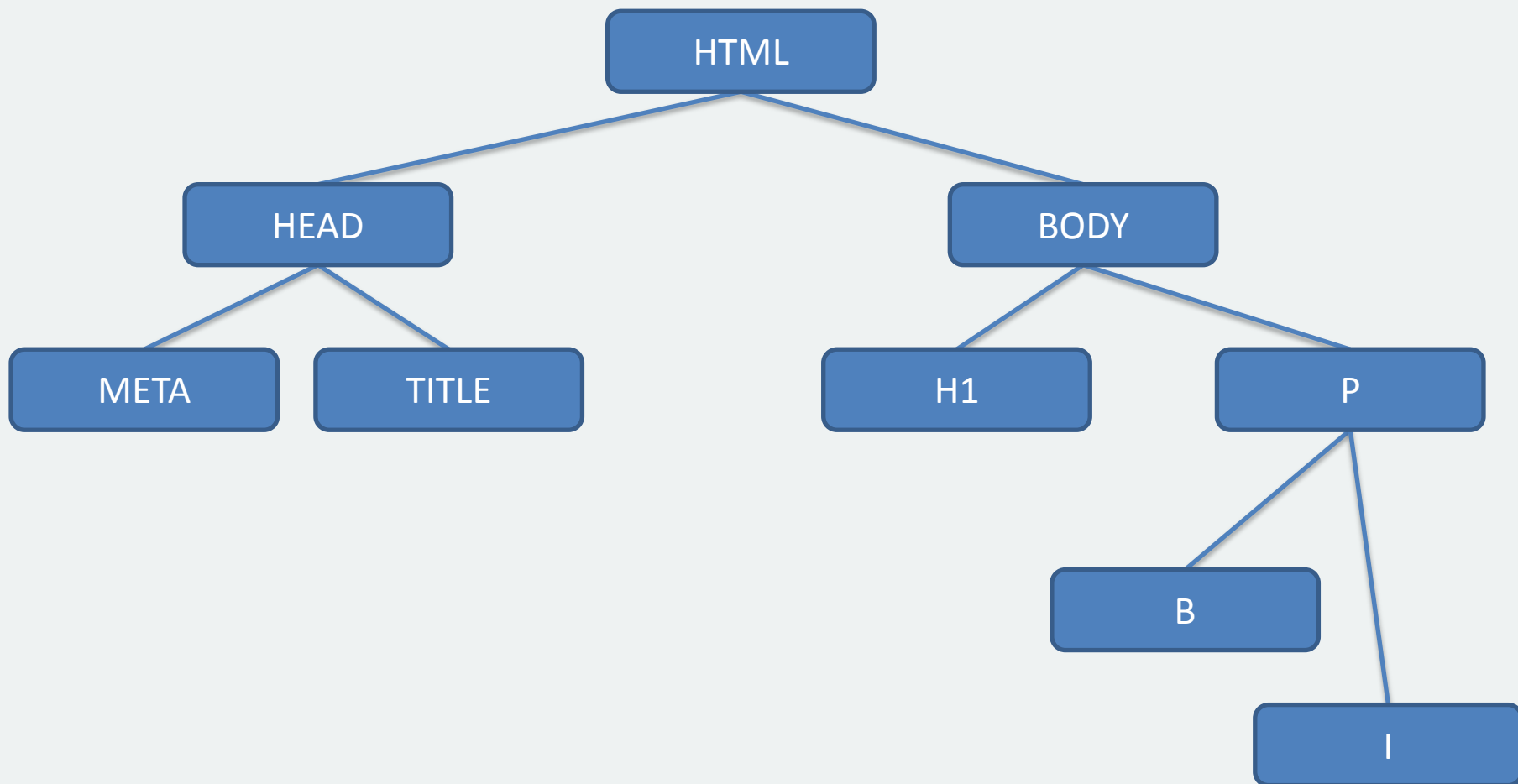


Лабораторная работа №7

Скриптовые языки для Web.



# Document Object Model



## Заголовок документа

```
<head>
```

```
  <!-- Служебная информация о документе -->  
  <meta charset="UTF-8">
```

```
  <meta name="keywords" lang="ru" content="МИЭТ,ПКИМС,магистратура,САПР">  
  <meta name="description" lang="ru" content="Лаба №7 по курсу ИЯП">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <meta name="robots" content="noindex, nofollow">
```

```
  <meta http-equiv="refresh" content="30">
```

```
  <title>Главная страница</title>
```

```
  <style>...</style>
```

```
  <link>...</link>
```

```
  <script>...</script>
```

```
</head>
```

## Структура документа: некоторые парные элементы

- Заголовки

```
<h1>Текст</h1>, <h2>Текст</h2>, ... , <h6>Текст</h6>
```

- Абзацы

```
<p>Текст</p>
```

- Стили текста

```
<b>Жирный</b>, <i>Курсив</i>, <u>Подчёркнут</u>, <s>Перечёркнут</s>
```

- Списки

```
<ul>  
  <li>Элемент 1</li>  
  <li>Элемент 2</li>  
</ul>
```

- Элемент разделения содержимого

```
<div></div>
```

## Структура документа: некоторые непарные элементы

- Принудительный перевод строки  
`<br />`
- Разрыв с отображением горизонтальной линии  
`<hr />`
- Изображение  
``  
``



## Ссылки: тег `<a></a>`

Это `<a href="#">` пустая ссылка `</a>`

Клики `<a href="http://site.ru/">` по ссылке `</a>` чтобы перейти

Клики `<a href="http://site.ru/" target="_blank">` по ссылке `</a>` чтобы открыть её  
в НОВОМ ОКНЕ

## Элементы управления

```
<p>
  <b>Логин:</b><br>
  <input type="text" size="40">
</p>

<p>
  <b>Пароль:</b><br>
  <input type="password" size="40">
</p>

<p>
  <b>Оцените наш сервис:</b><br>
  <input type="radio" name="mark" value="1"> 1
  <input type="radio" name="mark" value="2"> 2
  <input type="radio" name="mark" value="3"> 3
  <input type="radio" name="mark" value="4"> 4
  <input type="radio" name="mark" value="5"> 5
</p>

<p>
  <button>Кнопка</button>
</p>
```

**Логин:**

**Пароль:**

**Оцените наш сервис:**  
 1  2  3  4  5

## Стили элементов

<p>

Можно вот таким образом отделить текст в отдельный параграф.

А ещё к <b>тексту</b> <i>можно</i> применять стили.

А можно и не применять.<br>

<span style="color: red; font-size: 2em">П</span>ервая буква будет выделена красным цветом и будет не очень маленькая...

</p>

Можно вот таким образом отделить текст в с

**П**ервая буква будет выделена красным цв



## Каскадные листы стилей (CSS). Селекторы по элементу

```
<head>
...
<style type="text/css">
  span {
    color: red;
    font-size: 2em;
  }
</style>
</head>
```



**П**ервая прописная буква красная :)

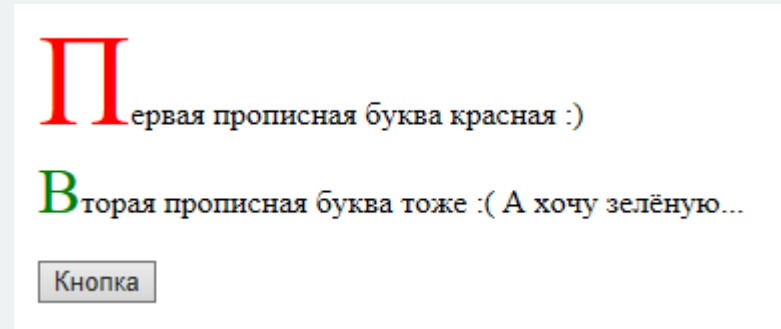
**В**торая прописная буква тоже :( А хочу зелёную...

```
<body>
...
<span>П</span>ервая прописная буква красная :)<br>
<span>В</span>торая прописная буква тоже :( А хочу зелёную...
...
</body>
```

## Каскадные листы стилей (CSS). Селекторы по классу и id

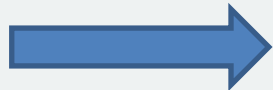
```
<style type="text/css">
  span.red {
    color: red;
    font-size: 2em;
  }
  #megabig {
    font-size: 4em;
  }
  span.green {
    color: green;
    font-size: 2em;
  }
</style>
```

```
...
<span class="red" id="megabig">П</span>ервая прописная буква красная :)<br>
<span class="green">В</span>торая прописная буква тоже :( А хочу зелёную...
...
<button>Кнопка</button>
```



## Подключение стилей из внешних файлов

```
<head>
...
<style type="text/css">
  .red {
    color: red;
    font-size: 2em;
  }
  #megabig {
    font-size: 4em;
  }
  span.green {
    color: green;
    font-size: 2em;
  }
</style>
...
</head>
```



### Файл .html

```
<head>
...
<link rel="stylesheet" href="./style.css">
...
</head>
```

### Файл style.css

```
span.red {
  color: red;
  font-size: 2em;
}
#big {
  font-size: 4em;
}
span.green {
  color: green;
  font-size: 2em;
}
```

## Вывод сообщения при нажатии на кнопку (1)

```
<button OnClick="alert('yo!');">Кнопка</button>
```

```
<body>
```

```
  <h1>Нажми на кнопку!</h1>
```

```
  <p>
```

```
    <button OnClick="OnButtonClick();">Кнопка</button>
```

```
  </p>
```

```
  <script type="text/javascript">
```

```
    function OnButtonClick() {
```

```
      alert('yo!');
```

```
    }
```

```
  </script>
```

```
</body>
```



## Вывод сообщения при нажатии на кнопку (2)

```
<body>
  <h1>Нажми на кнопку!</h1>
  <p>
    <button id="test">Кнопка</button>
  </p>

  <script type="text/javascript">

    document.getElementById("test").onclick = function() {
      alert('yo!');
    };

  </script>

</body>
```



## Изменение состояния элементов DOM (2)

```
<body>
  <p id="test">
    Я просто текст. Кликни по мне :)
  </p>

  <script type="text/javascript">

    document.getElementById("test").onclick = function() {

      document.getElementById("test").innerHTML = "Магия!!! :D";
      document.getElementById("test").style.color = "white";
      document.getElementById("test").style.backgroundColor = "blue";

    };

  </script>
</body>
```

## Библиотека jQuery

Подключение:

```
<script src="./jquery-3.4.1.min.js" ></script>
```

Использование:

```
<script type="text/javascript">  
    $("#test").click(function(){  
        alert("Yo!");  
    });  
</script>
```

Обращение к элементам DOM:

`$("#test")` – к элементу с `id=test`

`$(".test")` – к элементу с классом `class=test`

`$("#p#test")` – все элементы `<p></p>`, для которых задан `id=test`





## jQuery: обращение к элементу DOM

```
<body>
  <h1>Нажми на кнопку!</h1>
  <p>
    <button id="test">Нажми меня</button>
  </p>

  <script src="./jquery-3.4.1.min.js" ></script>

  <script type="text/javascript">
    $("#test").click(function(){
      $("#test").html("Спасибо!");
    });
  </script>
</body>
```



## jQuery: обращение к css элемента DOM (1)

```
<body>
  <p id="test">
    Я просто текст. Кликни по мне.
  </p>
  <p id="onoff">
    Я - абзац, который можно скрыть!
  </p>

  <script src="./jquery-3.4.1.min.js" ></script>
  <script type="text/javascript">
    $("#test").click(function(){
      $("#onoff").css("visibility", "hidden");
    });
  </script>
</body>
```

# PHP Hypertext Processor

```
<?php echo "Hello"; ?>
```

```
<?php echo "<h1>Hello!</h1>"; ?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Пример php кода</title>
```

```
</head>
```

```
<body>
```

```
  <h1><?php echo "Hello!"; ?></h1>
```

```
</body>
```

```
</html>
```





## Обработка GET-запросов внутри PHP

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Пример php кода</title>
</head>

<body>
  <h1>
    <?php
      if(isset($_GET) && isset($_GET['message']))
        echo $_GET['message'];
    ?>
  </h1>
</body>

</html>
```

## Реализация логина на сайт (1)

### Файл `index.php`

```
<form method="post" action="index.php">
  Username: <input type="text" name="username">
  Password: <input type="password" name="password" >
  <input type="submit" value="Залогиниться">
</form>

<hr/>

<?php print_r($_POST); ?>
```

Please log in



## Реализация логина на сайт (2)

```
<h1> <?php echo $status; ?> </h1>
```

```
<?php
    $status = 'Logged out';
    if (isset($_POST) &&
        isset($_POST['username']) &&
        isset($_POST['password'])) {

        if ($_POST['username'] == 'dima@pkims.ru' &&
            $_POST['password'] == '1234') {

            $status = "Logged in as dima@pkims.ru";
        } else {
            $status = "Login failed";
        }
    }
?>
<a href="?logout">Разлогиниться</a>
```



## Работа с сессиями (1)

```
<?php
    session_start();
?>

<?php
    $status = 'Logged out';
    if (isset($_POST['username']) && isset($_POST['password'])) {

        if ($_POST['username'] == 'dima@pkims.ru' &&
            $_POST['password'] == '1234') {

            $status = "Logged in as dima@pkims.ru";
            $_SESSION['status'] = 1;
        } else {
            $status = "Login failed";
        }
    }
?>
```



## Работа с сессиями (2)

```
<?php
    session_start();
?>
```

```
<?php
    if (isset($_SESSION['status']) && $_SESSION['status'] == 1) {
        $message = "User is logged in :)";
    } else {
        $message = "User is not logged in :(";
    }
?>
```

```
<h1> <?php echo $message; ?> </h1>
```



## Работа с сессиями : перенаправление пользователя

```
<?php
    session_start();

    if (!isset($_SESSION['status']) || $_SESSION['status'] != 1) {
        header('Location: index.php');
    }
?>
```





# Работа с БД:MySQL и phpMyAdmin

The screenshot shows the phpMyAdmin interface in a browser window. The address bar shows the URL: localhost/phpMyAdmin/server\_databases.php?server=1. The main content area displays the 'Databases' page. On the left, there is a sidebar with a tree view of databases: information\_schema, mysql, performance\_schema, and ritm2019. Below the tree view, there are buttons for 'Recent' and 'Favorites'. The main content area has a navigation bar with tabs for 'Databases', 'SQL', 'Status', 'User accounts', 'Export', and 'Import'. Below the navigation bar, there is a 'Create database' form with a text input field containing 'test\_db' and a dropdown menu set to 'utf8\_general\_ci'. A 'Create' button is next to the form. Below the form is a table listing the existing databases:

|                          | Database           | Collation         | Master replication | Action           |
|--------------------------|--------------------|-------------------|--------------------|------------------|
| <input type="checkbox"/> | information_schema | utf8_general_ci   | ✓ Replicated       | Check privileges |
| <input type="checkbox"/> | mysql              | latin1_swedish_ci | ✓ Replicated       | Check privileges |
| <input type="checkbox"/> | performance_schema | utf8_general_ci   | ✓ Replicated       | Check privileges |
| <input type="checkbox"/> | ritm2019           | utf8_general_ci   | ✓ Replicated       | Check privileges |
|                          | <b>Total: 4</b>    | latin1_swedish_ci |                    |                  |

At the bottom of the table, there is a 'Check all' checkbox and a 'With selected: Drop' button.



# Работа с БД:MySQL и phpMyAdmin

The image shows a screenshot of the phpMyAdmin web interface. The browser address bar displays `localhost/phpMyAdmin/server_databases.php?server=1`. The interface is divided into several sections:

- Left sidebar:** A tree view of databases. Under the `test_db` database, a list of tables is shown: `actions`, `messages`, `projects`, `specialities`, and `users`. A `New` button is visible next to the `test_db` entry.
- Top navigation:** A menu bar with options: `Structure`, `SQL`, `Search`, `Query`, `Export`, and `Import`.
- Main content area:** A message box with a warning icon and the text "No tables found in database." Below this is a `Create table` button. Underneath, there are two input fields: `Name:` with the value `users` and `Number of columns:` with the value `4`. A `Go` button is located at the bottom right of this section.



# Работа с БД:MySQL и phpMyAdmin

The screenshot shows the phpMyAdmin interface for a MySQL database. The browser address bar indicates the URL: localhost/phpMyAdmin/server\_databases.php?server=1. The interface shows the 'Structure' tab for a table named 'users' in the 'test\_db' database. The table has 1 column(s). The structure table is as follows:

| Name     | Type     | Length/Values | Default | Collation | Attributes | Null                     | Index   | A_I                                 | Comments          |
|----------|----------|---------------|---------|-----------|------------|--------------------------|---------|-------------------------------------|-------------------|
| id       | INT      |               | None    |           |            | <input type="checkbox"/> | PRIMARY | <input checked="" type="checkbox"/> | идентификатор пол |
| login    | TINYTEXT |               | None    |           |            | <input type="checkbox"/> | ---     | <input type="checkbox"/>            | логин             |
| password | TINYTEXT |               | None    |           |            | <input type="checkbox"/> | ---     | <input type="checkbox"/>            | пароль            |
| name     | TINYTEXT |               | None    |           |            | <input type="checkbox"/> | ---     | <input type="checkbox"/>            | имя               |

Table comments: Таблица пользователей  
Collation:   
Storage Engine: InnoDB

PARTITION definition:   
Partition by: ( Expression or column list )  
Partitions:

Buttons: Preview SQL, Save



# Работа с БД:MySQL и phpMyAdmin

localhost / localhost / test\_db

localhost/phpMyAdmin/tbl\_structure.php?db=test\_db&table=users

Server: localhost:3306 » Database: test\_db » Table: users "Таблица пользователей"

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

| #                        | Name              | Type     | Collation       | Attributes | Null | Default | Comments                   | Extra          | Action  |
|--------------------------|-------------------|----------|-----------------|------------|------|---------|----------------------------|----------------|---|
| <input type="checkbox"/> | 1 <b>id</b>       | int(11)  |                 |            | No   | None    | идентификатор пользователя | AUTO_INCREMENT | Change  Drop  Primary  Unique  Index  Spatial  More |
| <input type="checkbox"/> | 2 <b>login</b>    | tinytext | utf8_general_ci |            | No   | None    | логин                      |                | Change  Drop  Primary  Unique  Index  Spatial  More |
| <input type="checkbox"/> | 3 <b>password</b> | tinytext | utf8_general_ci |            | No   | None    | пароль                     |                | Change  Drop  Primary  Unique  Index  Spatial  More |
| <input type="checkbox"/> | 4 <b>name</b>     | tinytext | utf8_general_ci |            | No   | None    | имя                        |                | Change  Drop  Primary  Unique  Index  Spatial  More |

Check all With selected: Browse Change Drop Primary Unique Index

Print Propose table structure Move columns Improve table structure

Add  column(s)

Indexes

| Action     | Keyname | Type  | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|------------|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| Edit  Drop | PRIMARY | BTREE | Yes    | No     | id     | 0           | A         | No   |         |

Create an index on  columns



# Работа с БД:MySQL и phpMyAdmin

The screenshot shows the phpMyAdmin interface for a MySQL database. The browser address bar indicates the URL: localhost/phpMyAdmin/tbl\_change.php?db=test\_db&table=users. The interface shows the 'Table: users' structure with the following columns:

| Column          | Type     | Function | Null | Value  |
|-----------------|----------|----------|------|--|
| <u>id</u>       | int(11)  |          |      | 1  |
| <u>login</u>    | tinytext |          |      | <a href="mailto:dima@pkims.ru">dima@pkims.ru</a> |
| <u>password</u> | tinytext |          |      | 12345  |
| <u>name</u>     | tinytext |          |      | Дмитрий Булах                                    |



# Работа с БД:MySQL и phpMyAdmin

The screenshot shows the phpMyAdmin interface in a browser window. The address bar shows the URL: localhost/phpMyAdmin/sql.php?db=test\_db&table=users&pos=0. The browser's status bar indicates the server is localhost:3306, the database is test\_db, and the table is users. The interface includes a navigation menu with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. A green message bar at the top states: "Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)". Below this, the SQL query is displayed: `SELECT * FROM `users``. There are options for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Below the query, there are controls for "Show all", "Number of rows" (set to 25), and "Filter rows" (Search this table). The table data is shown with columns: id (идентификатор пользователя), login (логин), password (пароль), and name (имя). The first row contains the values: 1, dima@pkims.ru, 12345, and Дмитрий Булах. Below the table, there are options for "Check all", "With selected", Edit, Copy, Delete, and Export. At the bottom, there is a "Query results operations" section with buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `users`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

| id                         | login         | password | name          |
|----------------------------|---------------|----------|---------------|
| идентификатор пользователя | логин         | пароль   | имя           |
| 1                          | dima@pkims.ru | 12345    | Дмитрий Булах |

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view



## Работа с БД: код на PHP

```
<body>
```

```
<?php
```

```
    try{
```

```
        $db = new PDO("mysql:host=localhost;dbname=test_db;charset=utf8",  
                    "root", "root");
```

```
        $email="dima@pkims.ru";
```

```
        $request = $db->prepare("SELECT * FROM users WHERE login=:email");
```

```
        $request->execute(array(":email"=>$email));
```

```
        $row = $request->fetch(PDO::FETCH_ASSOC);
```

```
        $count = $request->rowCount();
```

```
        if($count) {
```

```
            echo $row['login'] . $row['password'];
```

```
        }
```

```
    } catch(PDOException $e){
```

```
        echo $e->getMessage();
```

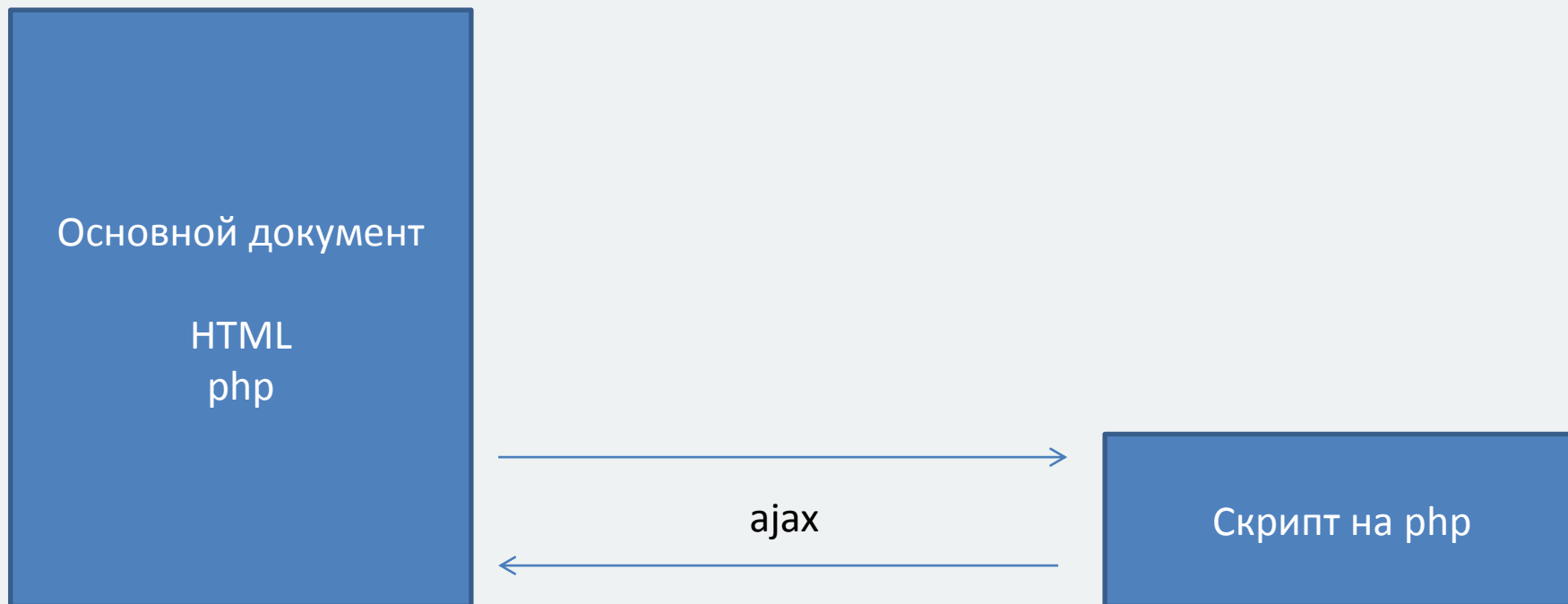
```
    }
```

```
?>
```

```
</body>
```

## Технология Asynchronous Javascript And Xml (AJAX)

AJAX позволяет получить данные из другого php скрипта не перезагружая основную страницу







## AJAX: вызываемый код

### Файл ajax.php

```
<?php
    try{
        $db = new PDO("mysql:host=localhost;dbname=test_db;charset=utf8",
            "root", "root");

        $email=trim($_POST['email']);

        $request = $db->prepare("SELECT * FROM users WHERE login=:email");
        $request->execute(array(":email"=>$email));
        $row = $request->fetch(PDO::FETCH_ASSOC);
        $count = $request->rowCount();

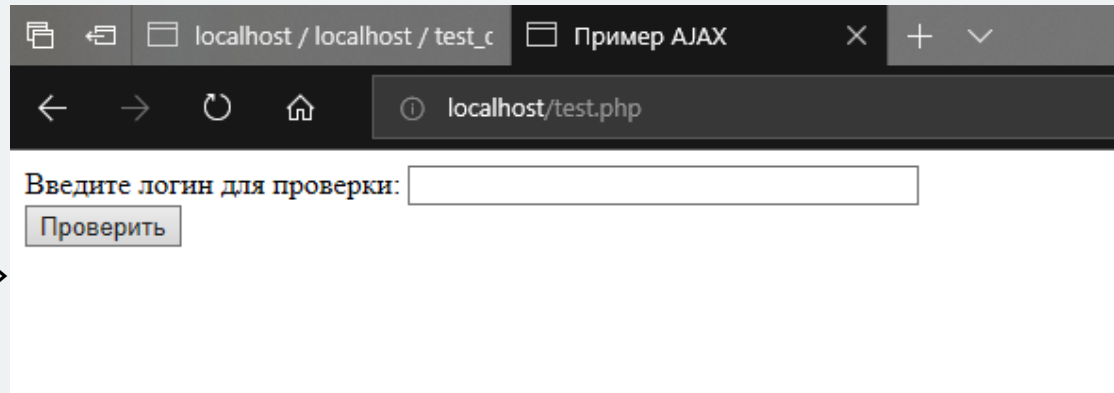
        if($count) {
            echo "ok";
        } else {
            echo "no";
        }
    } catch(PDOException $e){
        echo $e->getMessage();
    }
?>
```

## АJAX: вызывающий код (1)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Пример AJAX</title>
</head>

<body>
  Введите логин для проверки:
  <input type="text" size="40"><br>
  <button>Проверить</button>
<p></p>

  <script src="./jquery-3.4.1.min.js" ></script>
  <script type="text/javascript">
  </script>
</body>
</html>
```





## АJAX: вызывающий код (2)

Задаём идентификаторы нашим компонентам

Введите логин для проверки:

```
<input type="text" size="40" id="email"><br>  
<button id="btnSearch">Проверить</button>
```

```
<p id="result"></p>
```



## AJAX: вызывающий код (3)

Пишем код javascript:

```
<script type="text/javascript">
    $('#btnSearch').click(function() {

        $.ajax({
            type : 'POST',
            url  : 'ajax.php',
            data : {
                email : $("#email").val(),
            },
        });

    }
</script>
```



## АJAX: вызывающий код (4)

Пишем код javascript:

```
$.ajax({
  type : 'POST',
  url  : 'ajax.php',
  data : {
    email : $("#email").val(),
  },
  success : function(data) {
    if(data == "ok") {
      $("#result").html("Пользователь найден!");
    } else {
      $("#result").html("Такого пользователя нет.");
    }
  },
});
```



## АJAX: вызывающий код (6)

Пишем код javascript:

```
$.ajax({
  type : 'POST',
  url  : 'ajax.php',
  data : {
    email : $("#email").val(),
  },
  success : function(data) {
    if(data == "ok") {
      $("#result").html("Пользователь найден!");
    } else {
      $("#result").html("Такого пользователя нет.");
    }
  },
  error : function(data) {
    alert("Ошибка: '" + data + "'");
  }
});
```



## Задание

Реализовать простой движок сайта, который:

1. позволяет проводить регистрацию (самый простой вариант, никакую валидацию не делаем – ни проверку существования, ни проверку подтверждением по ссылке, просто ввели данные - зарегистрировали)
2. позволяет авторизовываться пользователю
3. позволяет разлогиниваться пользователю

На сайте есть две странички:

1. `index.php` – главная страничка, на ней вся работа
2. `about.php` – на неё может зайти только авторизованный пользователь; если пользователь не авторизован и пытается зайти – перенаправить его на `index.php` (см. слайд 24)

Реализовать с поддержкой БД (MySQL, phpMyAdmin в составе МАМАР)

Применять ли AJAX – выбор за вами, не принципиально.