

# PHP. Взаимодействие PHP и MySQL.

**Файл конфигурации, подключение скриптов,  
сохранение времени, порядок вывода.**

# Повторение

- `$db = new PDO(  
 'mysql:host=localhost;dbname=notesproject',  
 'root', ''  
);`
- `$query = $db->prepare('SELECT title FROM notes WHERE id = ?;');`
- `$query->execute(array($id));`
- `$result = $query->FETCHALL(PDO::FETCH_ASSOC);`
- `foreach ($result as $e) { // ...`

# Сохранение даты

- **date\_default\_timezone\_set('UTC');** // использовать UTC время
- **date('Y-m-d H:i:s');** // получить текущее время в формате год-месяц-день час:минута:секунда - формате MySQL

# Преобразование даты

```
// преобразует дату из mysql формата в российский формат  
// также переводит время с Гринвича на московское  
function format_mysql_date($mysql_date) {  
    $date = DateTime::createFromFormat('Y-m-d H:i:s', $mysql_date);  
    $date->setTimezone(new DateTimeZone('Europe/Moscow'));  
    return $date->format('d.m.Y H:i');  
}
```

# Порядок вывода

- **ORDER BY поле DESC; // по убыванию поля**
- **ORDER BY поле ASC; // по возрастанию поля**
- **SELECT id, title FROM notes ORDER BY id DESC; // вывести все записки по убыванию id**

# LIMIT

- `SELECT id, title FROM notes LIMIT 5, 10;` // вывести 10 записок после 5-ой
- `SELECT id, title FROM notes LIMIT 10 OFFSET 5;` // аналогичный запрос

# LIMIT и PDO Prepared Statements

- **Проблема:** `q->execute(array(...))`; подставляет параметры в LIMIT и OFFSET в виде строк, из-за чего запрос становится некорректным:
- `SELECT * FROM messages LIMIT '10' OFFSET '5';`
- **Решение:** явно указать PDO, что параметры - числа (int)
- `SELECT * FROM messages LIMIT 10 OFFSET 5;`



# Код

- `$q = $db->prepare('SELECT * FROM `messages` ORDER BY id  
DESC LIMIT :limit OFFSET :offset');`
- `$q->bindValue(':limit', 10, PDO::PARAM_INT);`
- `$q->bindValue(':offset', 5, PDO::PARAM_INT);`
- `$q->execute();`



# Подсчёт количества строк в таблице

- `$query = $db->query('SELECT COUNT(id) AS total FROM messages');`
- `$query = $query->fetch();`
- `$amount = (int)$query['total'];`
- `// (int)` - явное приведение к целочисленному типу

# Замена всех подстрок

- **\$где** = str\_replace('что', 'на что', **\$где**);
- \$content = str\_replace(**PHP\_EOL**, '**<br>**', \$content); // заменить переносы строки textarea (**\n**) на переносы строки HTML (**<br>**)
- **str\_replace не изменяет строку, а возвращает изменённую**

# Шаблонизация

- Идея - разделить логику (php) и отображение (html)

```
// получает на вход строку и ассоциативный массив  
// заменяет все подстроки вида {{ ключ }} на значение по ключу  
function render($template, $context) {  
    foreach ($context as $key => $el)  
        $template = str_replace('{{ '.$key.' }}', $el, $template);  
    return $template;  
}
```

**index.php**

```
$template = file_get_contents('notes_list.html');  
$context = array(  
    'title' => 'Список записок',  
    'notes_list' => $notes_list,  
);  
echo render($template, $context);
```

**template.html**

```
<div class="list-group">  
    {{ notes_list }}  
</div>
```

# Подключение скрипта

- **include('путь');** // подключает скрипт и продолжает выполнение в случае, если файл не был найден
- **require('путь');** // подключает скрипт и прерывает выполнение, если файл не был найден
- Скрипт вставится так, как-будто его строки были написаны в том месте, из которого его подключили. Т. е. все переменные будут доступны.

# Файл конфигурации

- `$config = parse_ini_file('config.ini');`
- `echo $config['username'];`
- ИЛИ
- `$config = parse_ini_file('config.ini', true);`
- `echo $config['DATABASE']['username'];`

```
1  [DATABASE]
2  engine = mysql
3  host = localhost
4  name = notesproject
5  username = root
6  password =
```

# Сессии

- **session\_start(<empty or session\_id);** // Запускает новую или возобновляет сущ-ую сессию. Возвращает TRUE - в случае успеха, иначе FALSE.
- **session\_id();** // возвращает идентификатор сессии
- **\$\_SESSION['username'] = 'User';** // получить или изменить данные в сессионном хранилище
- **session\_destroy();** // удалить все данные связанные с сессией (но session\_id не будет удалён!)



# json\_encode

- **json\_encode**(array(
  - 'messages' => \$messages\_amount,
  - 'pages' => \$pages\_amount,
  - 'data' => \$messages));
- // Кодирует PHP значение любого типа (почти) в обычную строку



# \* заметка про массивы

- PHP  $\geq$  5.4:
  - `$a = array(1, 2, 3); // OK`
  - `$a = [1, 2, 3]; // OK`
- PHP  $<$  5.4:
  - `$a = array(1, 2, 3); // OK`
  - ~~`$a = [1, 2, 3]; // NOT OK!`~~