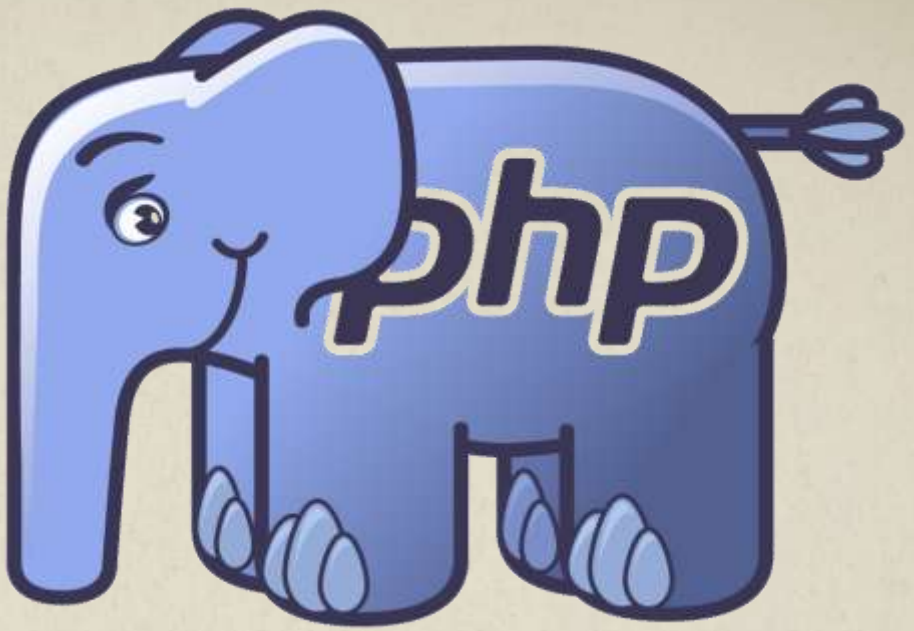
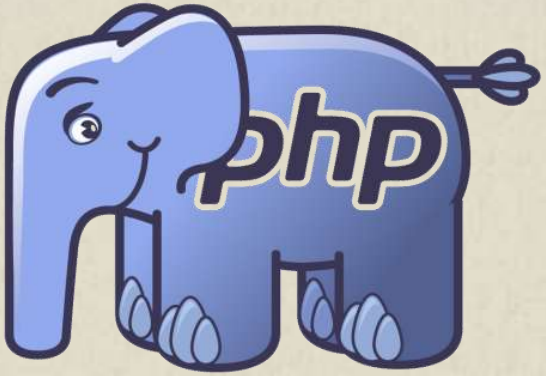


PHP: ВВЕДЕНИЕ В ПРОФЕССИЮ

УРОК 4. РАБОТА С ФАЙЛАМИ НА СЕРВЕРЕ. ЗАГРУЗКА ОТ КЛИЕНТА.



ФАЙЛЫ НА СЕРВЕРЕ

Еще один цикл – «ПОКА»

- Для начала нам нужно понять, что такое цикл WHILE («пока»)

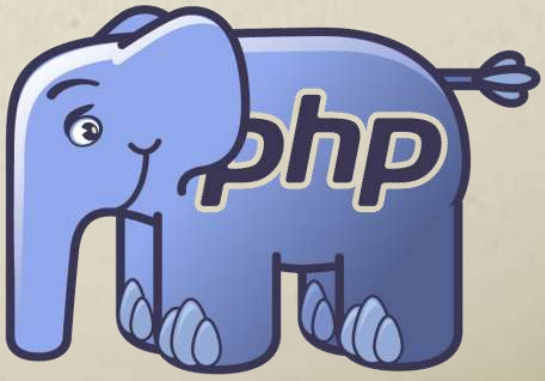
```
// while (условие) { тело цикла }
// цикл будет выполняться,
// ПОКА условие - ИСТИНА

$i = 1;
while ($i <= 10) {
  echo $i;
  $i++;
}
```

- Возможна (но редко применяется) и обратная форма записи:

```
$i = 0;
do {
  echo $i;
  $i++;
} while ($i <= 10)
```

- Разница в том, что цикл с пост-условием всегда выполнится хотя бы один раз

НЕМНОГО ТЕОРИИ



ЧТЕНИЕ файлов – способы прочитать данные из файла в свою программу

- Для начала нам нужно открыть файл. При этом мы получим «ресурс» – ссылку на открытый файл, с которой потом сможем работать:

```
$res = fopen(ПУТЬКФАЙЛУ, 'r');
```

- А затем в цикле читать строки из файла:

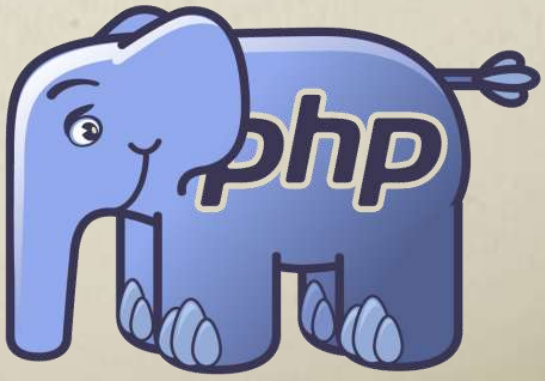
```
while ( !feof($res) ) {
  $line = fgets($res, 1024)
}
```

- И закрыть файл:

```
fclose($res);
```

- Функции, которые вам нужно знать:
  - `fopen()`
  - `fclose()`
  - `fread()`
  - `fgets()`

ЧТЕНИЕ ИЗ ФАЙЛОВ



ЧТЕНИЕ файлов – способы прочитать данные из файла в свою программу

Всё это прекрасно, но нельзя ли как-то проще?

Конечно можно, это же PHP! ☺

- `$lines = file(ПУТЬКФАЙЛУ);`

  Чтение целиком файла в массив. Каждый элемент массива – строка.

- `$str = file_get_contents(ПУТЬКФАЙЛУ);`

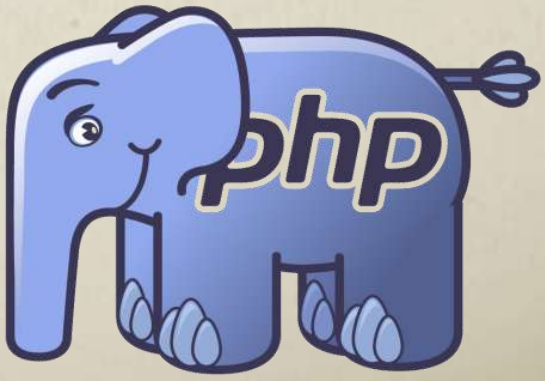
  Чтение целиком файла в строку. Самый оптимальный по производительности вариант

И еще функции, которые вам нужно знать:

- `readfile()`
- `file_exists()`
- `is_readable()`

И, конечно, константа `__DIR__` !

ЧТЕНИЕ ИЗ ФАЙЛОВ



ЗАПИСЬ в файл – способы из программы записать данные в файл

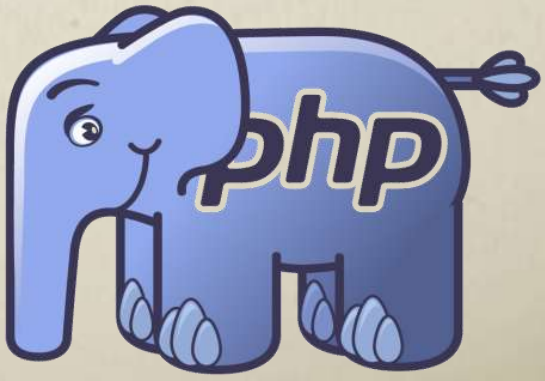
Способ первый, трудоемкий, но зато всё под контролем:

```
$res = fopen(ПУТЬКФАЙЛУ, 'w');
fwrite($res, $data); // string!
fclose($res);
```

Настала пора подробно поговорить о режимах открытия файлов:

- `r` – чтение
- `r+` - чтение и запись
- `w` – запись. файл будет создан, если не существовал или «обнулен»
- `w+` – запись и чтение. файл будет создан, если не существовал или «обнулен»
- `a` – запись. файл будет создан, если не существовал. запись в конец файла

ЗАПИСЬ В ФАЙЛЫ

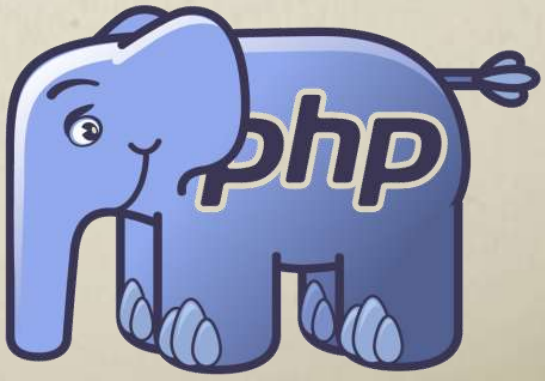


ЗАПИСЬ в файл – способы из программы записать данные в файл

Способ второй, простой, **потому что это PHP**:

```
file_put_contents(ПУТЬКФАЙЛУ, $data);
```

ЗАПИСЬ В ФАЙЛЫ



INCLUDE не так прост, как вы думаете

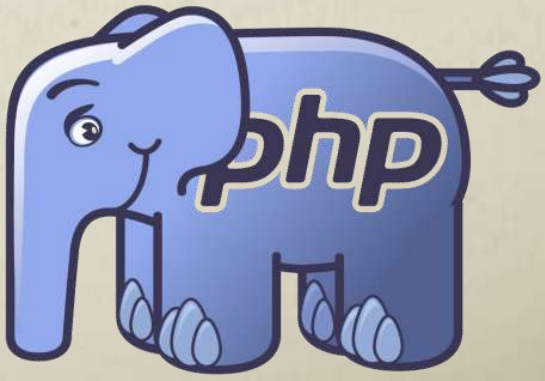
Это же близко к «чтению из файлов», не так ли?

```
// foo.php

<?php
$res = 2 + 2;
return $res;

// index.php

<?php
$foo = include(__DIR__ . '/foo.php');
echo $foo;
```

Таким образом файлы в PHP, как и функции, могут возвращать значения с помощью оператора `return`.

Чтобы это значение получить – достаточно получить то, что вернет конструкция `include`

КСТАТИ...

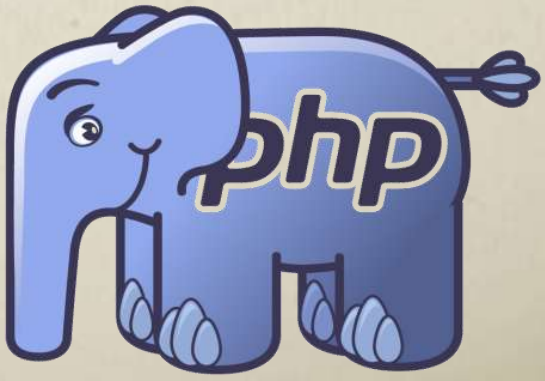


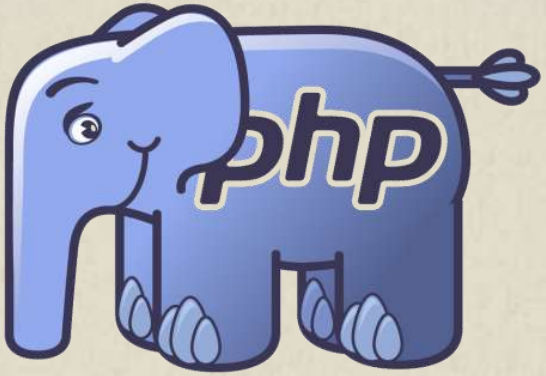
А еще есть средства для работы не только с отдельными файлами, но с файловой системой в целом

Например, имеются следующие функции:

- `scandir($path)`
  Возвращает массив имён всех файлов, содержащихся в папке `$path`
- `dirname($path)`
  Имя родительской папки для указанного файла (или папки)
- `pathinfo($path)`
  Возвращает массив с частями пути (путь, имя файла, расширение)
- `filesize($path)`
  Размер файла `$path` в байтах
- `realpath($path)`
  Возвращает канонический абсолютный путь для указанного. Раскрывает все «.», «..», ссылки и так далее

КСТАТИ...



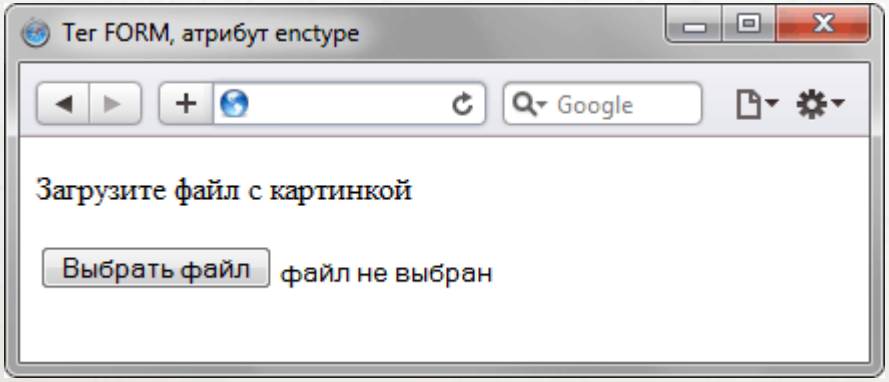


ЗАГРУЗКА ФАЙЛОВ

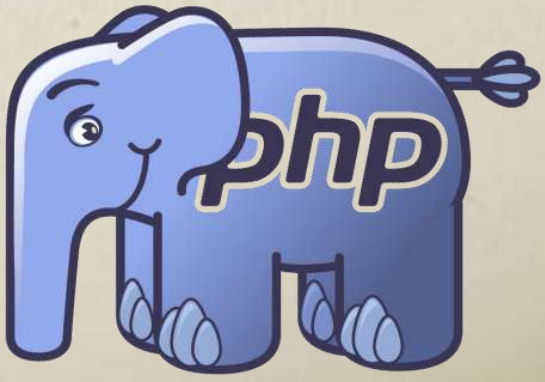
Для того, чтобы иметь возможность загрузить файлы от клиента (из браузера) на сервер, нужно:

- Специальным образом построить форму загрузки:

```
<form
  action="/upload.php"
  method="post"
  enctype="multipart/form-data"
>

  <input type="file" name="myimg">

</form>
```



ФОРМА ДЛЯ ЗАГРУЗКИ



Для того, чтобы иметь возможность загрузить файлы от клиента (из браузера) на сервер, нужно:

- Прочитать данные из суперглобального массива **$_FILES**:

```
if ( isset($_FILES['myimg']) ) {

 if (0 == $_FILES['myimg']['error']) {

    $res = move_uploaded_file(
      $_FILES['myimg']['tmp_name'],
      ПОЛНЫЙПУТЬНОВОЕИМЯНАСЕРВЕРЕ
    );

  }
}
```

Важно:

- Обращать внимание на размер загружаемого файла, на него могут быть установлены ограничения на сервере
- Использовать корректные пути!

ПРИЕМ ФАЙЛА НА СЕРВЕРЕ