



# PHR. Архитектура

[sd@freematiq.com](mailto:sd@freematiq.com)



# PHP программа

<?php  
код

?> - не обязательно

Всё, что содержится как текст - пойдёт как  
ВЫВОД



# Пример

text.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>
<body>
  Hello from php!
  Current year is <?= date('Y') ?><br/>
  Current year is <?php echo date('Y') ?>
</body>
</html>
```



# Подключение файлов

```
require 'template.php';  
require_once('template.php');
```

```
include 'templates/index.php';  
include_once '../view.php';
```



# Синтаксис шаблона

<?php - просто код ?>

<?= ВЫВОД ?>

<?php if (: ?>...<?php endif; ?>

<?php foreach(: ?>...<?php endforeach; ?>



# Пример

condition\_loop.php

```
<body>
  <?php if (isset($_GET['n'])): ?>
    <?php $n = (int) $_GET['n']; ?>
    Последовательность от 1 до <?= $n ?>:
    <ul>
      <?php for($i = 1; $i <= $n; $i++): ?>
        <li><?= $i ?></li>
      <?php endfor; ?>
    </ul>
  <?php else: ?>
    Введите параметр n
  <?php endif; ?>
</body>
```



# Архитектура

MVC - model, view, controller;

Controller:

```
$n = $_GET['n'] ?? 0;  
require 'view.php';
```

Контроллер - звено, соединяющее модель, виды и другие компоненты в рабочее приложение. Содержит мало кода: “тонкие” контроллеры



# Архитектура

V - view

View:

```
<?php if ($n > 0): ?>
```

Последовательность от 1 до <?= \$n ?>:

```
<ul>
```

```
<?php for($i = 1; $i <= $n; $i++): ?>
```

```
<li><?= $i ?></li>
```

```
<?php endfor; ?>
```

```
</ul>
```

```
<?php else: ?>
```

Введите параметр n

```
<?php endif; ?>
```

Можно было передать элементы





# View, Вид

- Используется для отображения данных
- Не должны обращаться к базам данных (модель)
- Не должны работать с данными из запросов пользователя (контроллер)
- Обращается к моделям шаблона



# Модель

- Содержит бизнес-логику
- Не должна взаимодействовать со вводом (контроллер)
- Не должна генерировать код (вид)



# Page Controller

localhost:8000/main.php

localhost:8000/user.php?id=10

localhost:8000/number.php?n=5



# Front Controller

localhost:8000/index.php

localhost:8000/index.php?action=hello

localhost:8000/index.php?action=numbers&n=7



# Front Controller

Задача:

- / - если есть GET параметр n, то вывод последовательности
- /hello - Hello, PHP
- Иначе: 404



# Front Controller

Все пути обрабатывает один файл (front.php)

```
<?php
```

```
require_once 'controller.php';
```

```
$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
```

```
if ('/' === $uri) {
```

```
    $n = $_GET['n'] ?? 0;
```

```
    show_numbers($n);
```

```
} elseif('/hello' === $uri) {
```

```
    say_hello();
```

```
} else {
```

```
    header('HTTP/1.1 404 Not Found');
```

```
    echo '<html><body>404 Page Not Found</body></html>';
```

```
}
```



# Front Controller

## Controller

```
<?php
```

```
function say_hello()  
{  
    $name = 'PHP';  
    require 'templates/hello.php';  
}
```

```
function show_numbers(int $n)  
{  
    $r = [];  
    if ($n > 0) {  
        $r = range(1, $n);  
    }  
    require 'templates/numbers.php';  
}
```



# Front Controller

Hello view

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>
<body>
  Hello, <?= $name ?>
</body>
</html>
```





# Front Controller

## Numbers view

```
<body>
  <?php if (!empty($r)): ?>
    Последовательность от 1 до <?= count($r) ?>:
    <ul>
      <?php foreach ($r as $i): ?>
        <li><?= $i ?></li>
      <?php endforeach; ?>
    </ul>
  <?php else: ?>
    Введите GET параметр n
  <?php endif; ?>
</body>
```



# Итого

- PHP - встраиваемый язык
- Подключение файлов - require
- MVC архитектура
- Model - бизнес-логика
- View - отображение моделей
- Controller - обработка ввода



# Домашнее задание

Создать MVC приложение, содержащее следующие страницы:

- Главная, содержит ссылки на другие разделы
- Раздел 1 - генерирует случайный массив из 10 элементов и выводит максимальный и минимальный элемент
- Раздел 2 - Выводит таблицу умножения
- Раздел 3 (несколько ссылок) - Меню на главной, которое строится из массива и открывается с содержимым