

Раздел 3:

Шаблонизация и организация кода



Содержание вебинара



Содержание вебинара

- Как работать с датой и временем



Содержание вебинара

- Как работать с датой и временем
- Подключение файлов в PHP-сценариях



Содержание вебинара

- Как работать с датой и временем
- Подключение файлов в PHP-сценариях
- Как организовывать код в проекте



Содержание вебинара

- Как работать с датой и временем
- Подключение файлов в PHP-сценариях
- Как организовывать код в проекте
- Шаблонизация: отделение логики от интерфейса



Содержание вебинара

- Как работать с датой и временем
- Подключение файлов в PHP-сценариях
- Как организовывать код в проекте
- Шаблонизация: отделение логики от интерфейса
- Буферизация вывода



Содержание вебинара

- Как работать с датой и временем
- Подключение файлов в PHP-сценариях
- Как организовывать код в проекте
- Шаблонизация: отделение логики от интерфейса
- Буферизация вывода
- Боремся с XSS-атаками (межсайтовый скрипting)



Работа с датой и временем в PHP

как посчитать разницу между двумя датами, узнать текущее время, показать дату с учетом локали и многое другое



Возможные операции с датой/временем



Возможные операции с датой/временем

- показать текущую дату/время



Возможные операции с датой/временем

- показать текущую дату/время
- узнать сколько дней/часов осталось до даты в будущем



Возможные операции с датой/временем

- показать текущую дату/время
- узнать сколько дней/часов осталось до даты в будущем
- посчитать разницу между двумя датами



Возможные операции с датой/временем

- показать текущую дату/время
- узнать сколько дней/часов осталось до даты в будущем
- посчитать разницу между двумя датами
- отформатировать дату



Возможные операции с датой/временем

- показать текущую дату/время
- узнать сколько дней/часов осталось до даты в будущем
- посчитать разницу между двумя датами
- отформатировать дату
- перевести время в другой часовой пояс



Возможные операции с датой/временем

- показать текущую дату/время
- узнать сколько дней/часов осталось до даты в будущем
- посчитать разницу между двумя датами
- отформатировать дату
- перевести время в другой часовой пояс
- узнать день недели



Метка времени Unixtime

Unixtime – это количество секунд, прошедших с 1 января 1970 года



Для чего нужен `unixtime`

Unixtime незаменим, когда когда требуется:

- посчитать разницу между датами
- максимально простой способ сохранить дату для последующей обработки



Для чего нужен **unixtime**

Unixtime незаменим, когда когда требуется:

- посчитать разницу между датами
- максимально простой способ сохранить дату для последующей обработки

В PHP можно:

- перевести дату в метку времени timestamp
- перевести метку времени в дату



Для чего нужен unixtime

Unixtime незаменим, когда когда требуется:

- посчитать разницу между датами
- максимально простой способ сохранить дату для последующей обработки

В PHP можно:

- перевести дату в метку времени timestamp
- перевести метку времени в дату

1546300800 → 1 января 2019 00:00

1 сентября 2007 14:20 → **1188656400**



Считаем разницу между датами

Самый простой способ узнать сколько дней между двумя датами - перевести их в unixtime



Считаем разницу между датами

Самый простой способ узнать сколько дней между двумя датами - перевести их в unixtime

Последовательность действий:

1. Конвертировать даты в timestamp
2. Вычесть одно число из другого
3. Разницу поделить на число секунд в сутках



Считаем разницу между датами

Самый простой способ узнать сколько дней между двумя датами - перевести их в unixtime

Последовательность действий:

1. Конвертировать даты в timestamp
2. Вычесть одно число из другого
3. Разницу поделить на число секунд в сутках

Продолжительность	Количество секунд
1 минута	60
1 час	3600
1 сутки	86400
1 неделя	604800



Считаем разницу между датами



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.

Решение:



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.

Решение:

1. Метка для 6 февраля 2018: **1517875200**



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.

Решение:

1. Метка для 6 февраля 2018: **1517875200**
2. Метка для 1 января 2019: **1546300800**



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.

Решение:

1. Метка для 6 февраля 2018: **1517875200**
2. Метка для 1 января 2019: **1546300800**
3. Считаем разницу:

$$1546300800 - 1517875200 = \mathbf{28425600 \text{ секунд}}$$



Считаем разницу между датами

Задача:

узнать сколько дней осталось до нового года

Исходные данные:

текущая дата - 6 февраля 2018 г.

Решение:

1. Метка для 6 февраля 2018: **1517875200**

2. Метка для 1 января 2019: **1546300800**

3. Считаем разницу:

$$1546300800 - 1517875200 = \mathbf{28425600 \text{ секунд}}$$

4. Считаем количество дней:

$$28425600 \% 86400 = \mathbf{329 \text{ дней}}$$

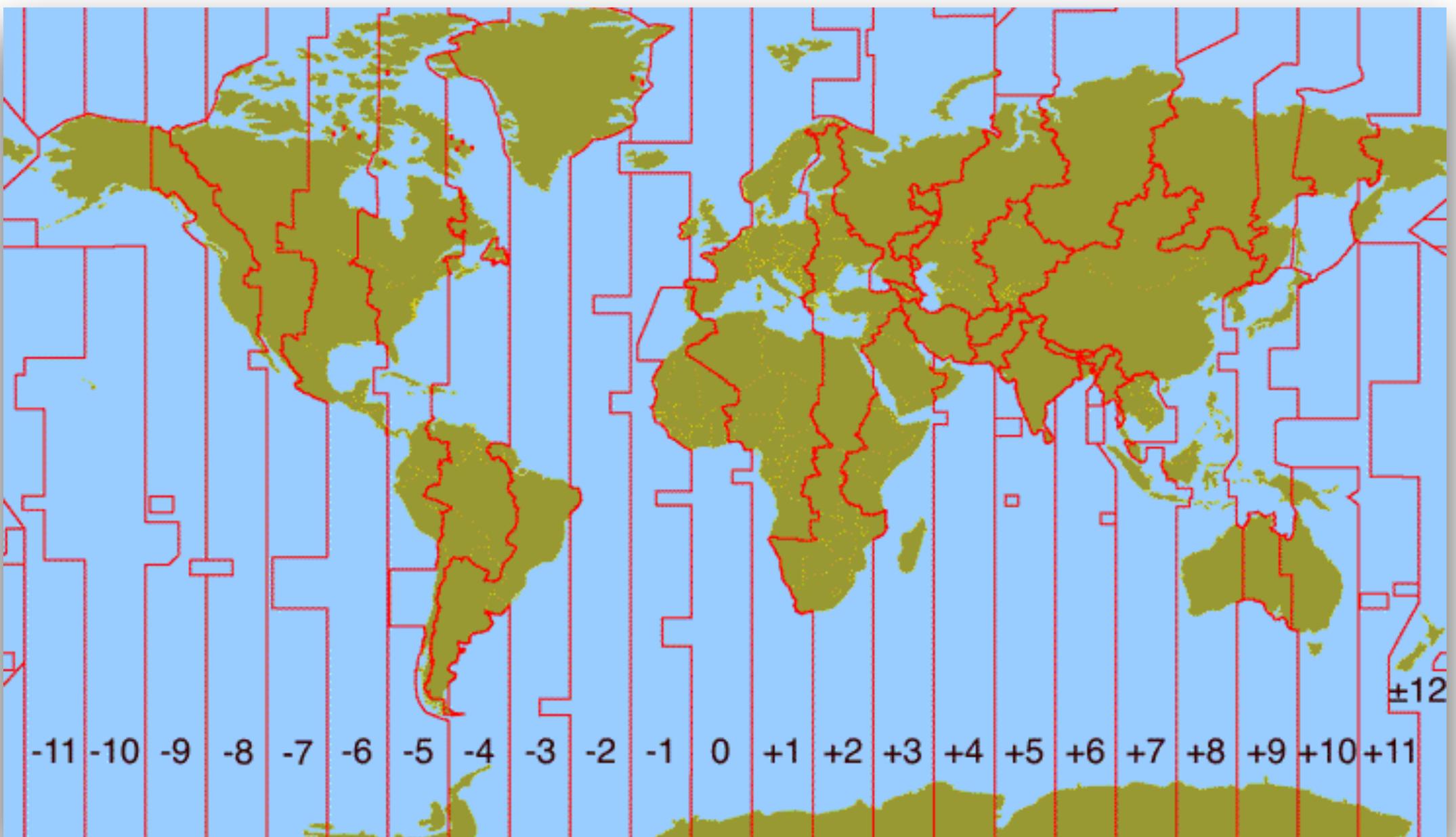


Часовая зона

По умолчанию, PHP показывает дату для временной зоны, установленной в его настройках

Как изменить в скрипте часовую зону:

```
date_default_timezone_set('Europe/Moscow')
```



Как узнать текущую дату/время?

Самая полезная функция для получения
даты/времени в любом формате:

```
date($format, $timestamp = false)
```



Как узнать текущую дату/время?

Самая полезная функция для получения
даты/времени в любом формате:

```
date($format, $timestamp = false)
```

Формат – это строка из специальных
символов-заместителей



Как узнать текущую дату/время?

Самая полезная функция для получения даты/времени в любом формате:

`date($format, $timestamp = false)`

Формат – это строка из специальных символов-заместителей

Символ в строке <code>format</code>	Описание	Пример возвращаемого значения
<code>d</code>	День месяца, 2 цифры с ведущим нулём	от 01 до 31
<code>l</code> (строчная 'L')	Полное наименование дня недели	от <i>Sunday</i> до <i>Saturday</i>
<code>F</code>	Полное наименование месяца, например <i>January</i> или <i>March</i>	от <i>January</i> до <i>December</i>
<code>m</code>	Порядковый номер месяца с ведущим нулём	от 01 до 12
<code>Y</code>	Порядковый номер года, 4 цифры	Примеры: <i>1999</i> , <i>2003</i>
<code>H</code>	Часы в 24-часовом формате с ведущим нулём	от 00 до 23
<code>i</code>	Минуты с ведущим нулём	от 00 до 59
<code>s</code>	Секунды с ведущим нулём	от 00 до 59



Демонстрация

Примеры работы с датой/временем



Подключение файлов

PHP позволяет включать одни сценарии в другие.

Так можно объединять множество разных сценариев

в один



Зачем разделять PHP-сценарии

В любом веб-проекте есть много повторяющейся логики и/или данных:

- общие пользовательские функции
- конфигурация сайта
- общие данные
- **шаблоны страниц**



Зачем разделять PHP-сценарии

В любом веб-проекте есть много повторяющейся логики и/или данных:

- общие пользовательские функции
- конфигурация сайта
- общие данные
- **шаблоны страниц**

Смысл разделения PHP-сценариев на отдельные файлы в возможности повторно использовать один код



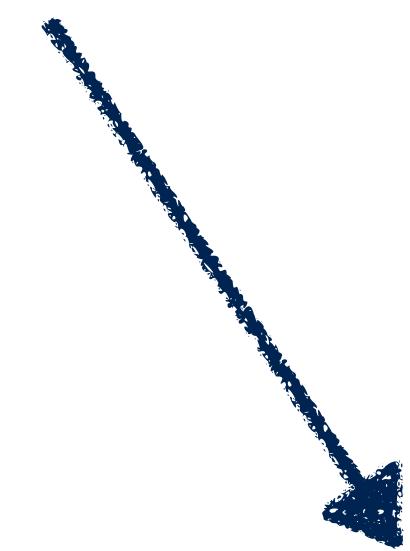
Разделение ответственности

view.php



Разделение ответственности

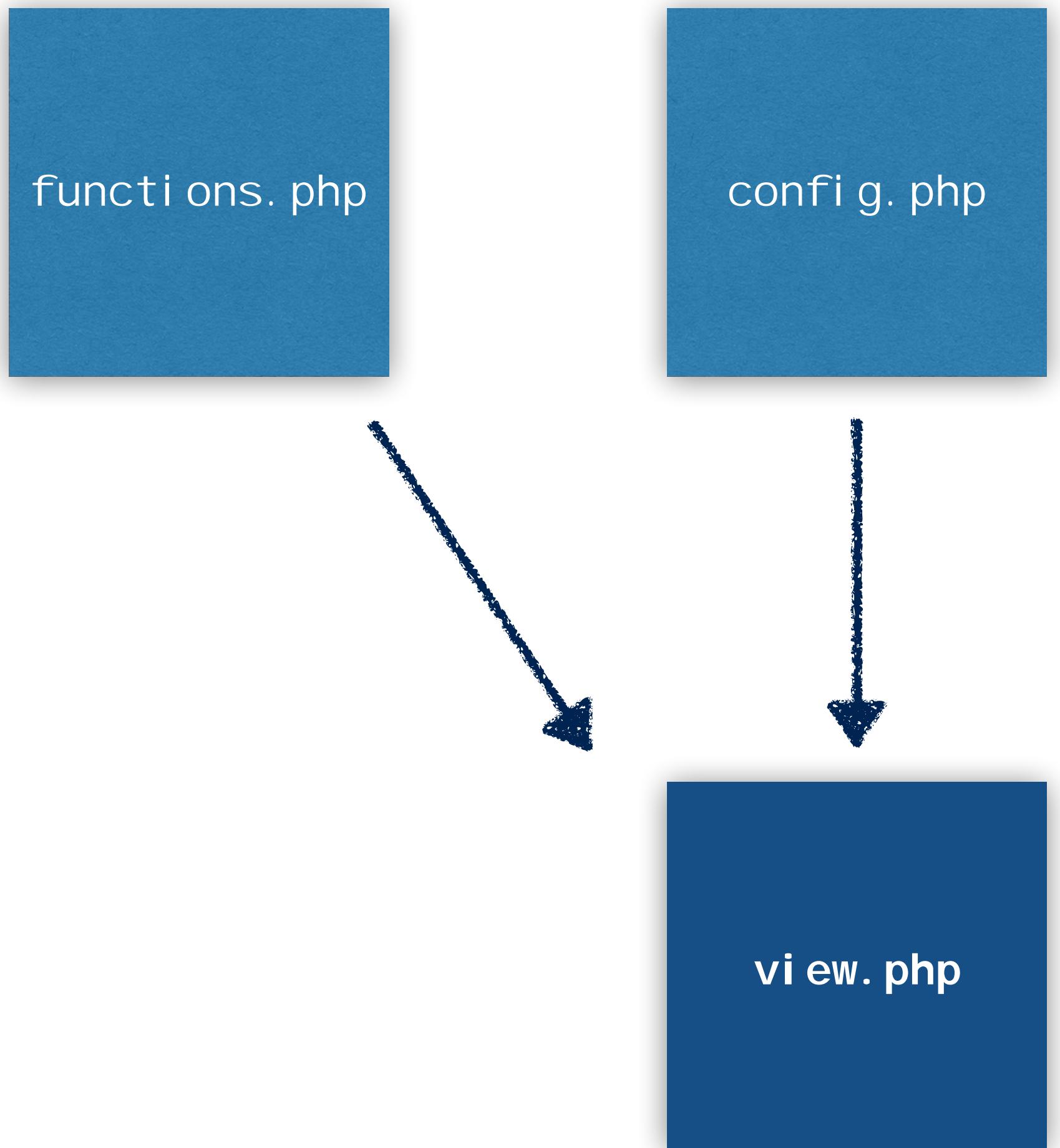
functions.php



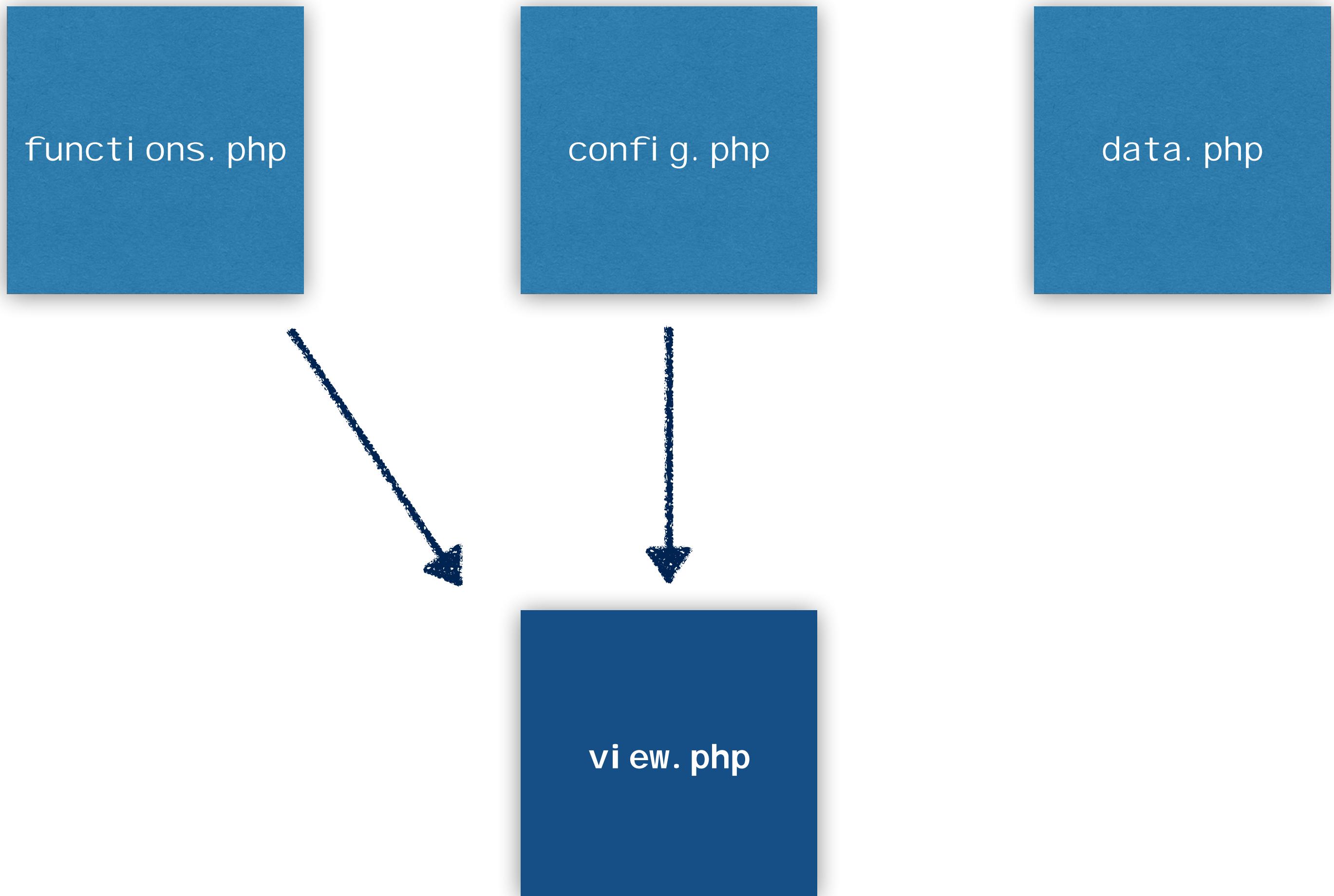
view.php



Разделение ответственности



Разделение ответственности



Как подключить сценарий



Как подключить сценарий

```
require('filename.php');
```

Подключает сценарий. Если файл сценария не существует, то вызывает фатальную ошибку и останавливает выполнение.



Как подключить сценарий

```
require('filename.php');
```

Подключает сценарий. Если файл сценария не существует, то вызывает фатальную ошибку и останавливает выполнение.

```
require_once('filename.php');
```

Подключает сценарий только один раз.

При повторном вызове не будет снова его подключать.



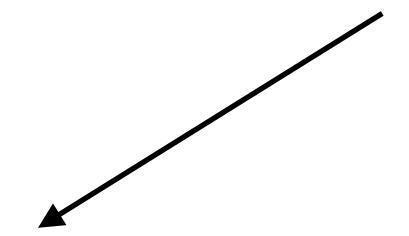
Подключение файлов в верстке



Подключение файлов в верстке

Подключение файла стиля в CSS

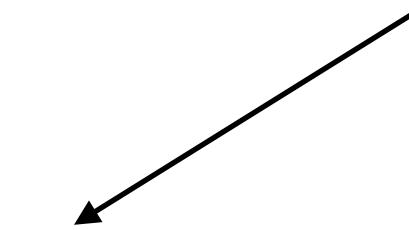
```
@import url ("common.css");
```



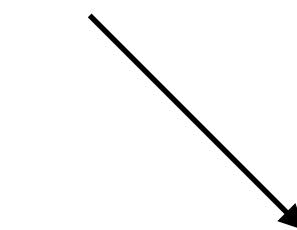
Подключение файлов в верстке

Подключение файла стиля в CSS

```
@import url ("common.css");
```



Подключение скрипта в HTML



```
<script src="main.js"></script>
```



Подключение файлов в PHP

index.php

The screenshot shows a code editor interface with a dark theme. On the left is a project tree labeled "Project" containing a folder named "project" which has an "inc" folder. Inside "inc" are three files: "config.php", "data.php", and "functions.php". Below the project tree is a code editor window with the following PHP code:

```
.php';
};

require_once 'inc/config.php';
require_once 'inc/data.php';
require_once 'inc/functions.php';

print("Hello world!");
```



Видимость переменных

Все переменные во включаемых сценариях
становятся доступными внутри общего сценария



Общая область видимости

Все подключаемые сценарии как будто склеиваются
в один большой и получают общую область видимости переменных



Общая область видимости

Все подключаемые сценарии как будто склеиваются
в один большой и получают общую область видимости переменных

index.php

data.php

```
$cats = ['Игры'];
```

functions.php

```
function cut_text() {};
```

```
cut_text();
print($cats[0]);
```



Подключение внутри функций

Если подключить сценарий внутри функции, то все
определенные в нём переменные останутся
в локальной области видимости функции



Подключение внутри функций

Если подключить сценарий внутри функции, то все определенные в нём переменные останутся в локальной области видимости функции

inc/data.php

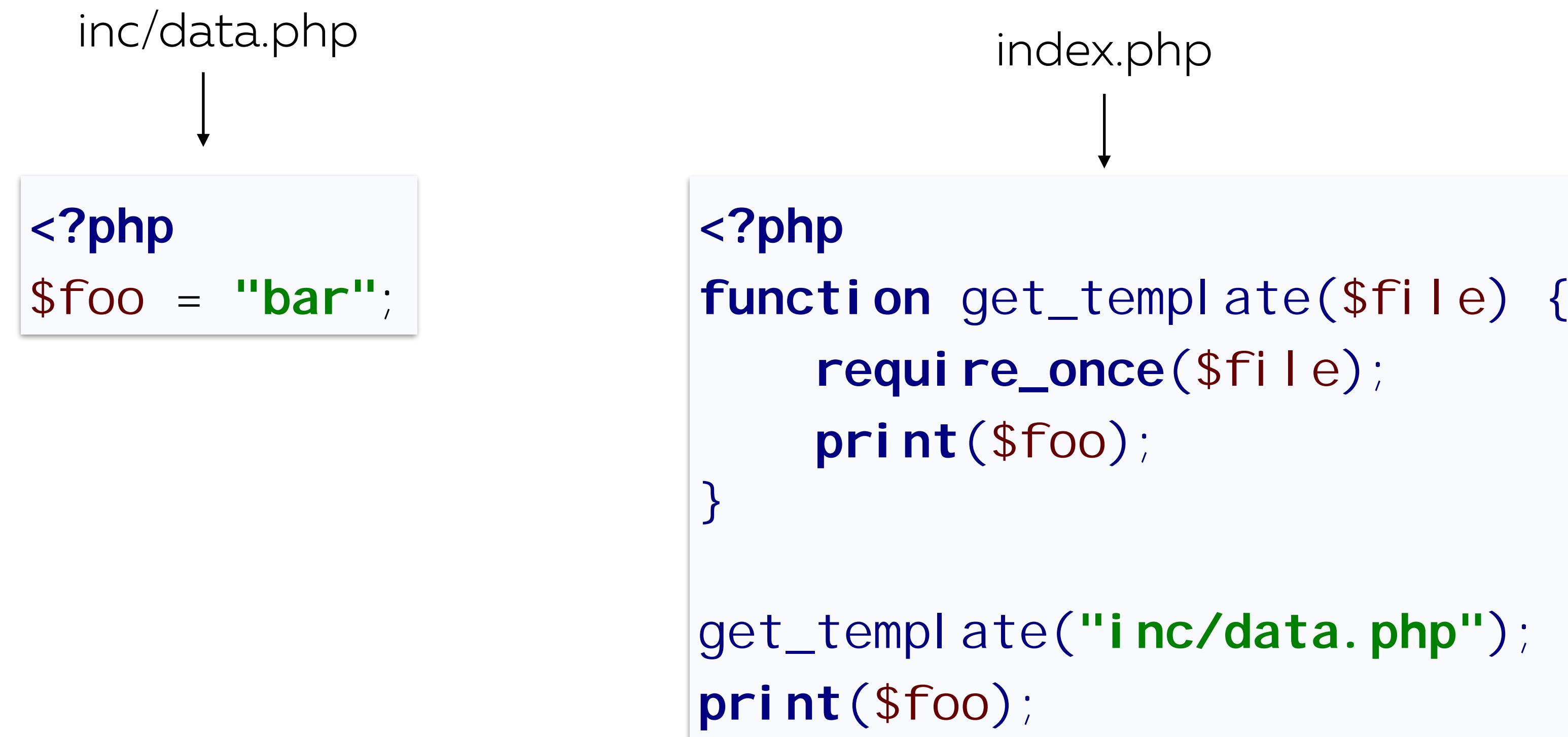


```
<?php  
$foo = "bar";
```



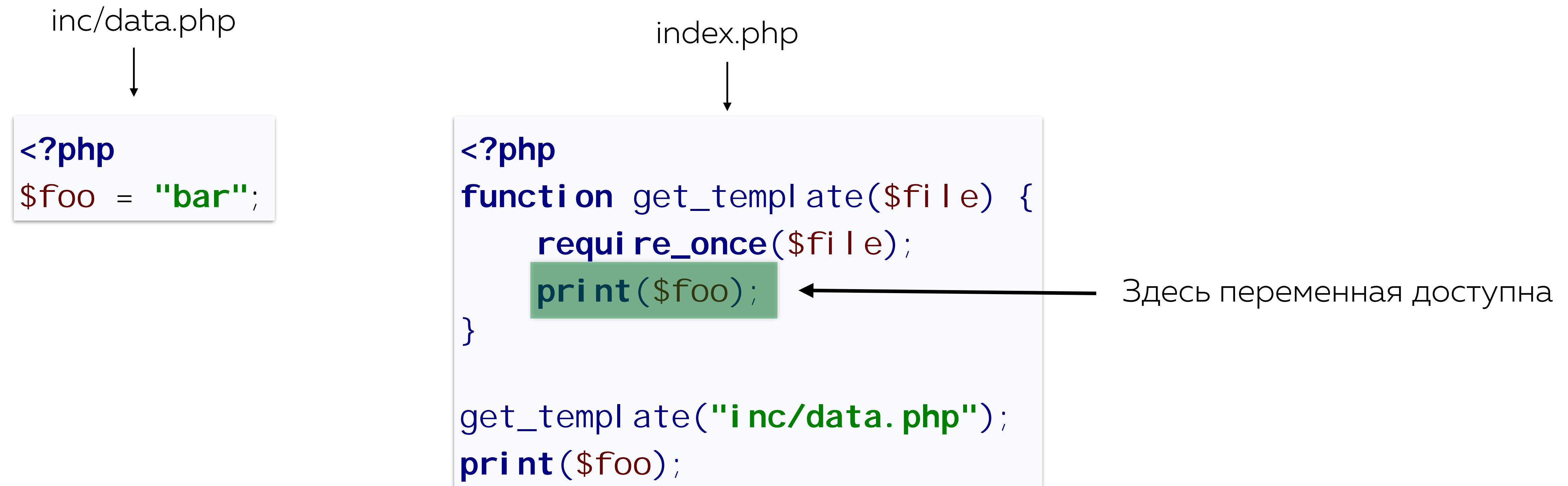
Подключение внутри функций

Если подключить сценарий внутри функции, то все определенные в нём переменные останутся в локальной области видимости функции



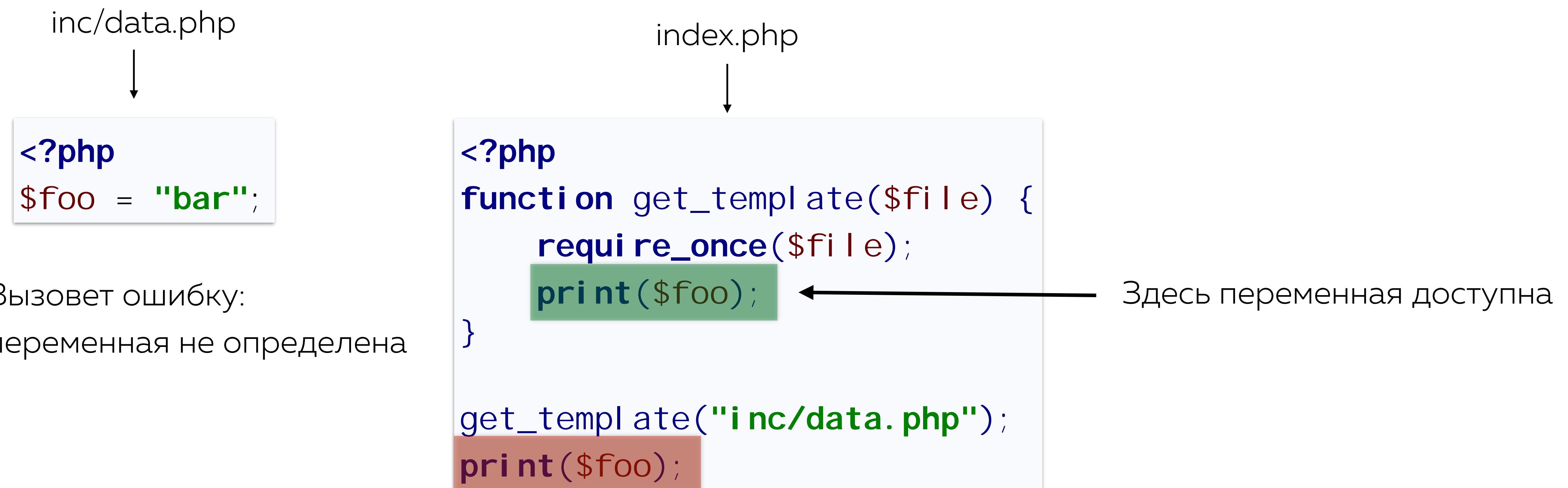
Подключение внутри функций

Если подключить сценарий внутри функции, то все определенные в нём переменные останутся в локальной области видимости функции



Подключение внутри функций

Если подключить сценарий внутри функции, то все определенные в нём переменные останутся в локальной области видимости функции



Шаблонизация

деление всей верстки сайта на отдельные блоки – шаблоны.

В сценариях такие шаблоны будут подключены по необходимости и заполнены данными



Что такое шаблон?

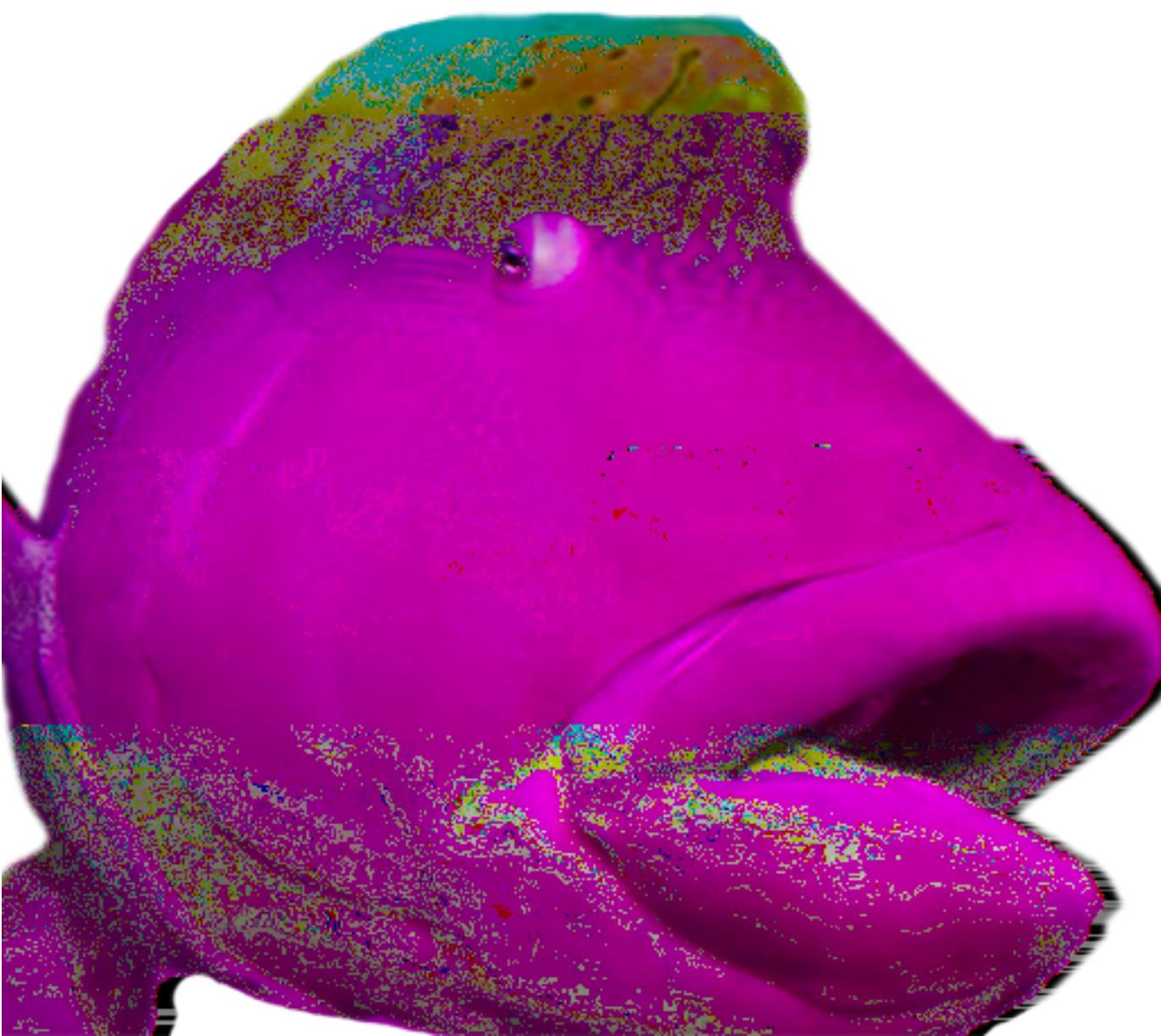
готовая верстка страницы или блока, которая
состоит только из оформления и не содержит
никакого контента (полезной информации)



«Рыба» – заполнитель текста

Рыба – это заполнитель.

Бессмысленный текст, который используется в верстке, чтобы показать как будет выглядеть страница, наполненная контентом.



**True Multiple Designs**

Dolore magna aliquautenim ad minim veniam nostru exercitation ullamco laboris, minim veniam exercitation ullamco.

**Power Packed Themes**

Dolore magna aliquautenim ad minim veniam nostru exercitation ullamco laboris, minim veniam exercitation ullamco.

**More Themes in Feature**

Dolore magna aliquautenim ad minim veniam nostru exercitation ullamco laboris, minim veniam exercitation ullamco.

Welcome to Medical and Healthcare Center

Web page editors now use Lorem Ipsum as their default model text, and a search for lorem ipsum will many web sites still in their recently with desktop publishing software like fancy various versions have evolved over the years.



Типовой процесс верстки



Типовой процесс верстки

index.html

HTML

<code>

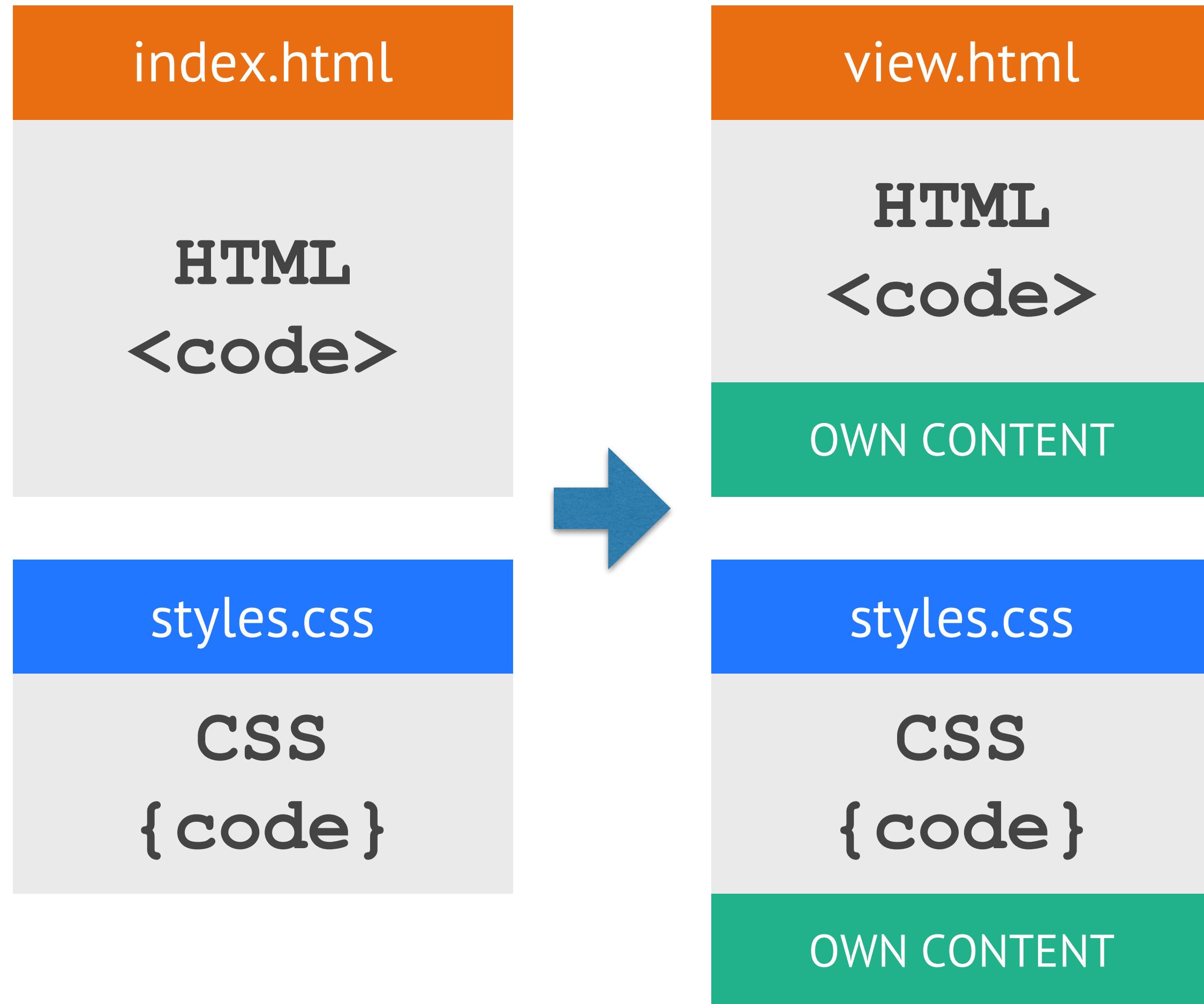
styles.css

CSS

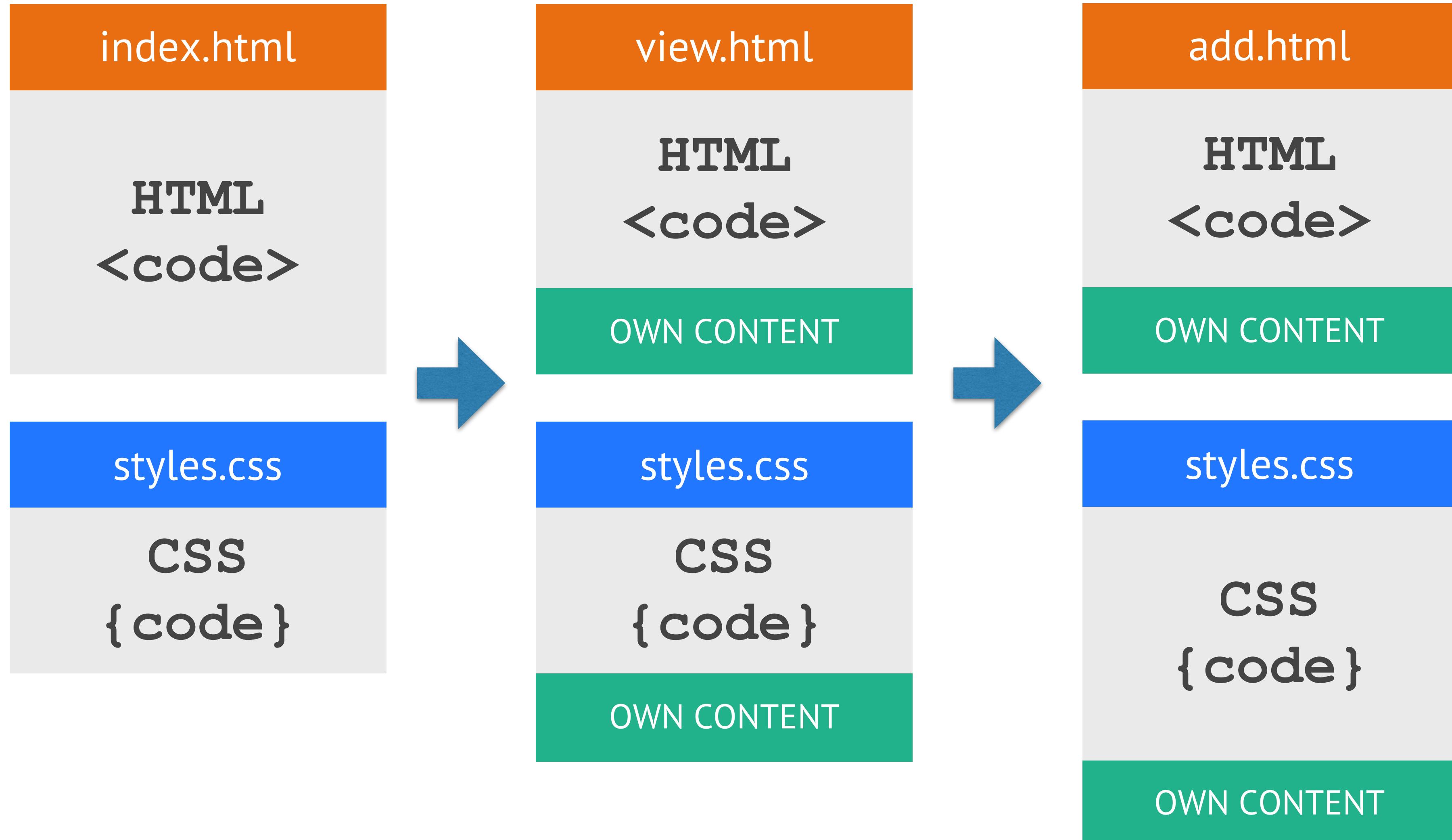
{ code }



Типовой процесс верстки



Типовой процесс верстки



Внесение правок в вёрстке



Внесение правок в вёрстке



Добавь сверху
меню и лого



Внесение правок в вёрстке



Добавь сверху
меню и лого

10 раз правит шапку
сайта



Внесение правок в вёрстке



Добавь сверху
меню и лого

10 раз правит шапку
сайта



Нужно
добавить лого
платежек в футер



Внесение правок в вёрстке



Добавь сверху
меню и лого

10 раз правит шапку
сайта



Нужно
добавить лого
платежек в футер

10 раз правит футер
сайта



Внесение правок в вёрстке



Внесение правок в вёрстке



А почему у нас в
половине страниц меню
съехало?



Внесение правок в вёрстке



А почему у нас в
половине страниц меню
съехало?



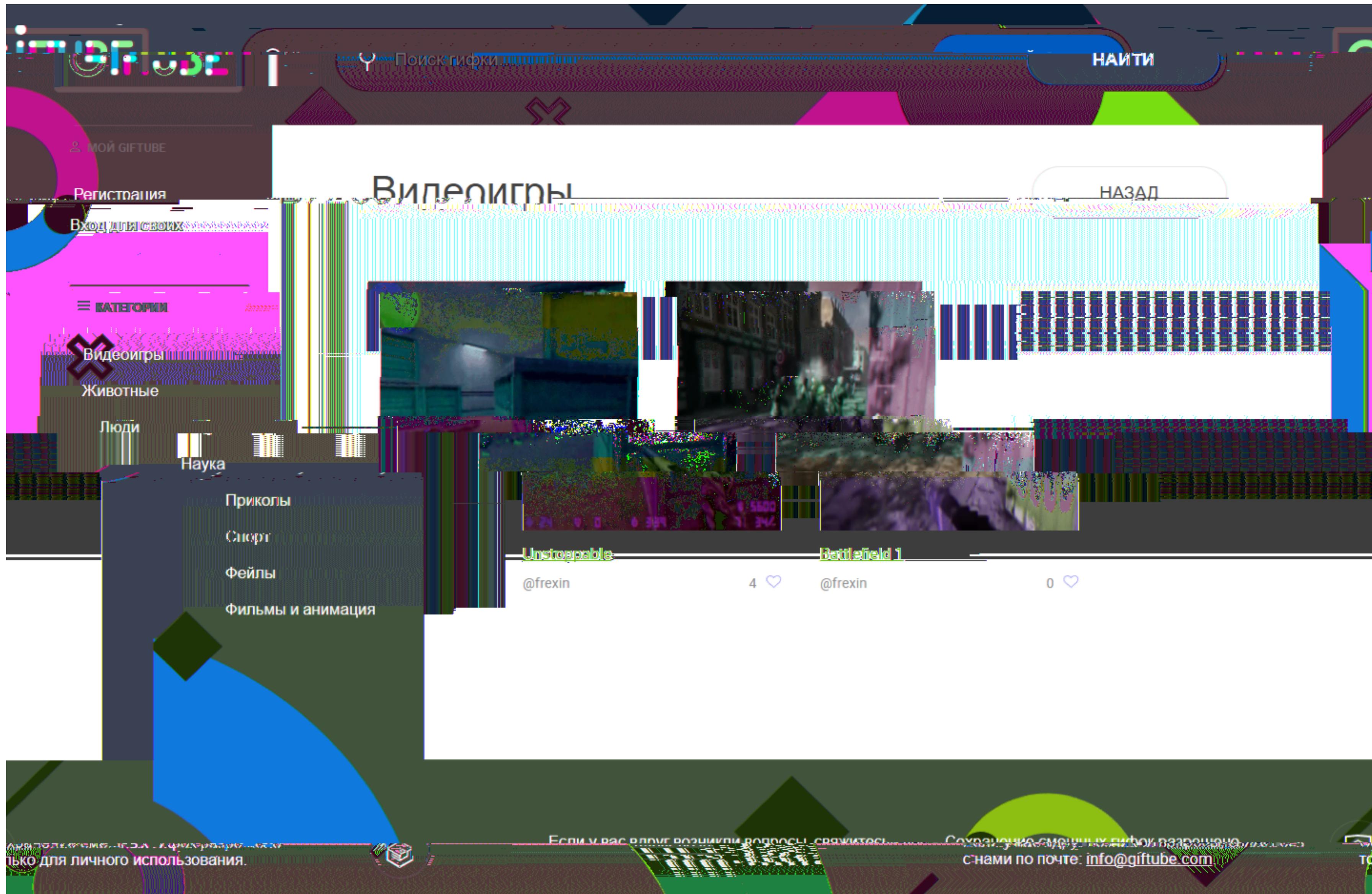
#%\$@!!111



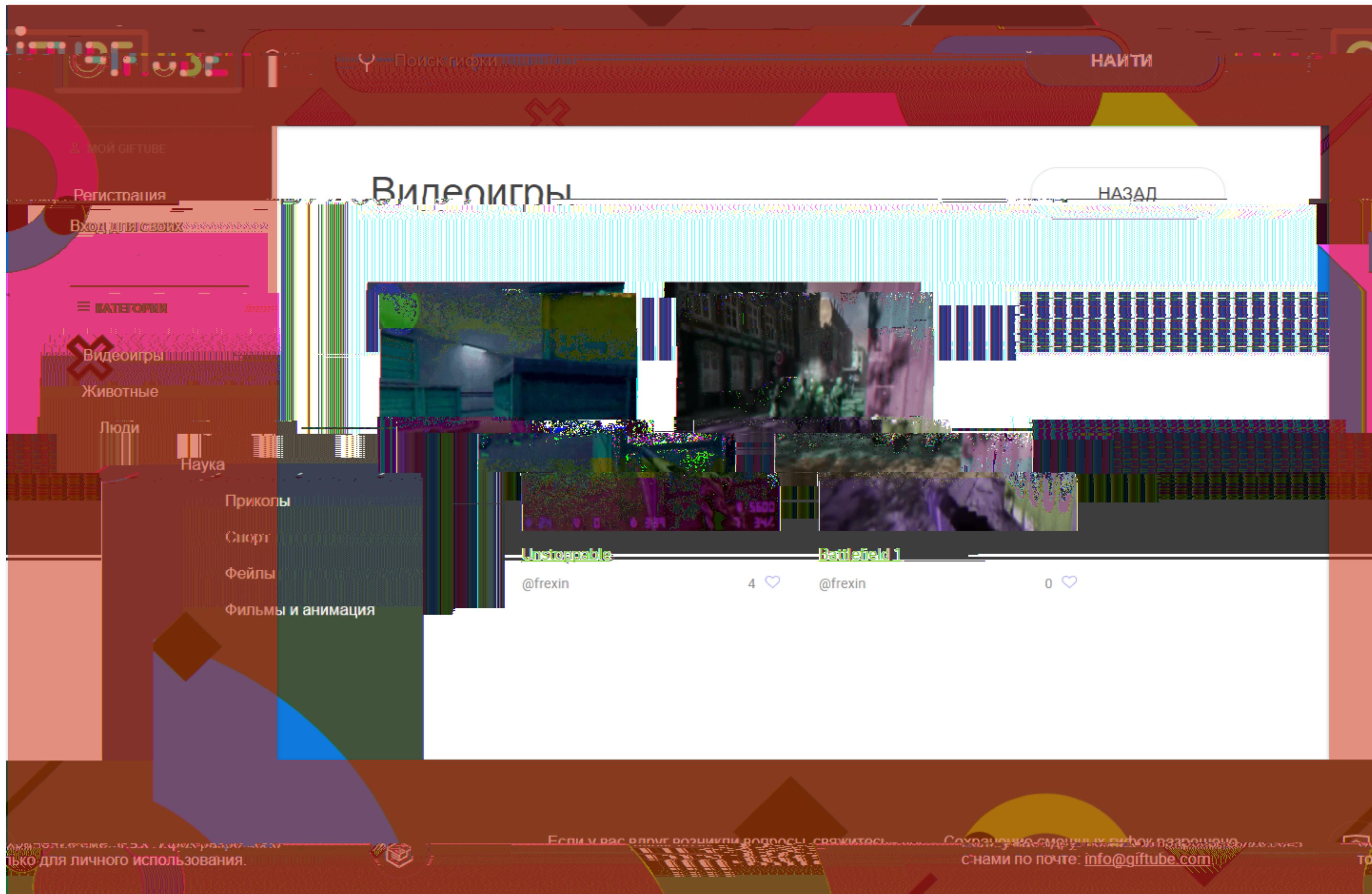
Общие части страниц



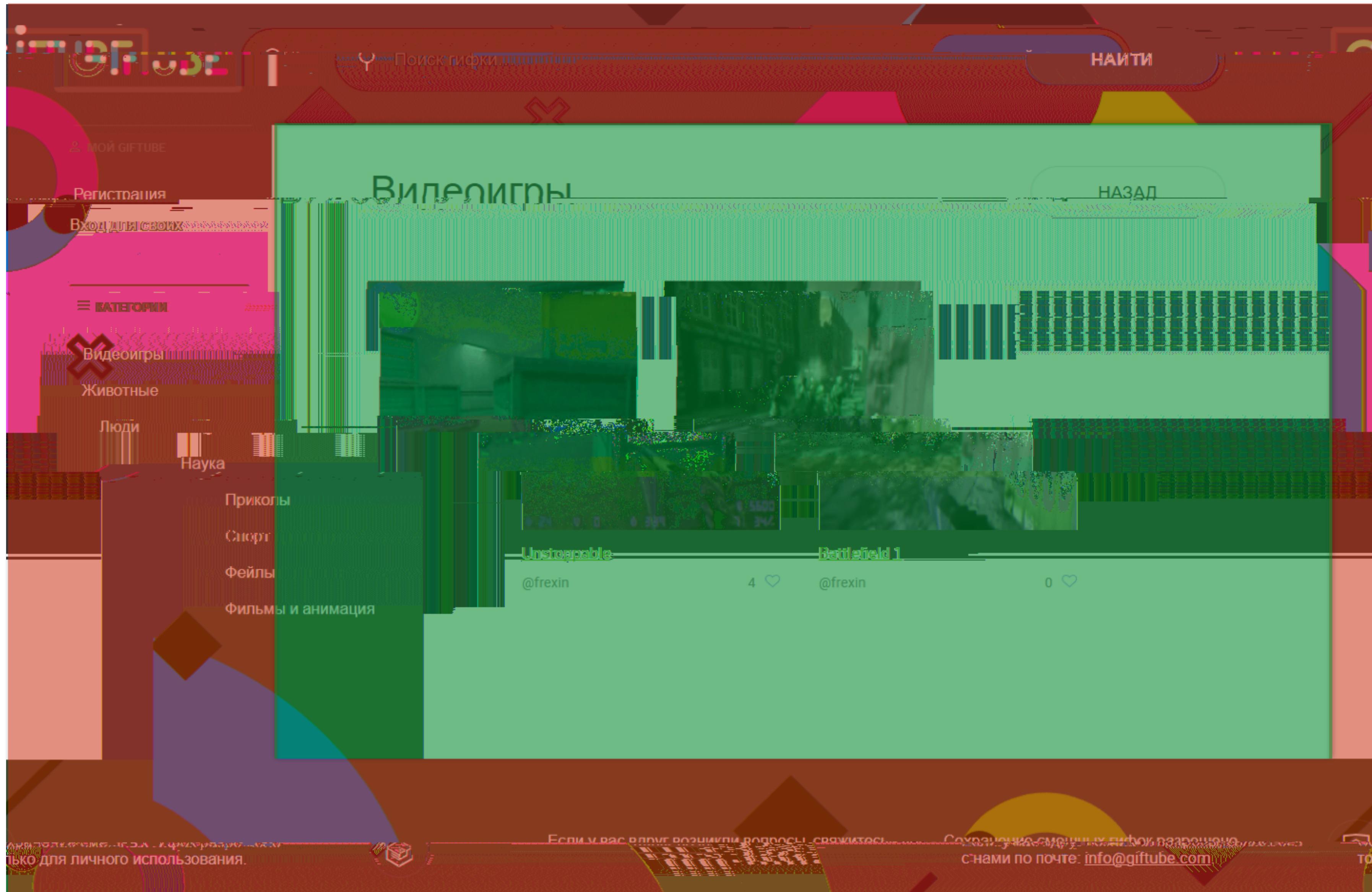
Общие части страниц



Общие части страниц



Общие части страниц



giftube.academy

Поиск гифки... Шапка сайта

НАЙТИ

МОЙ GIFTUBE

Регистрация

Вход для своих

Категории

Видеогames

Животные

Люди

Наука

Приколы

Спорт

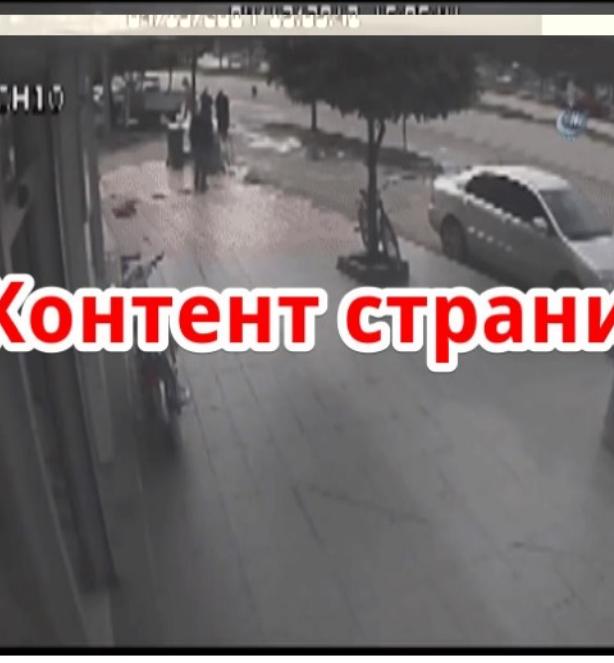
Фейлы

Фильмы и анимация

Приколы

НАЗАД

 Конец рабочего дня
@frexin 2 ❤️

 Контент страницы
@Чукча 1 ❤️

Если у вас вдруг возникли вопросы, свяжитесь с нами по почте: info@giftube.com.

Подвал

Сохранение смешных гифок разрешено только для личного использования.



Преимущества шаблонизации



Преимущества шаблонизации

- Сложные сайты состоят из множества общих блоков: шапка, подвал, боковое меню, и т.д.



Преимущества шаблонизации

- Сложные сайты состоят из множества общих блоков: шапка, подвал, боковое меню, и т.д.
- Разделение логики сценария и отображения



Преимущества шаблонизации

- Сложные сайты состоят из множества общих блоков: шапка, подвал, боковое меню, и т.д.
- Разделение логики сценария и отображения
- Легче читать и поддерживать



Преимущества шаблонизации

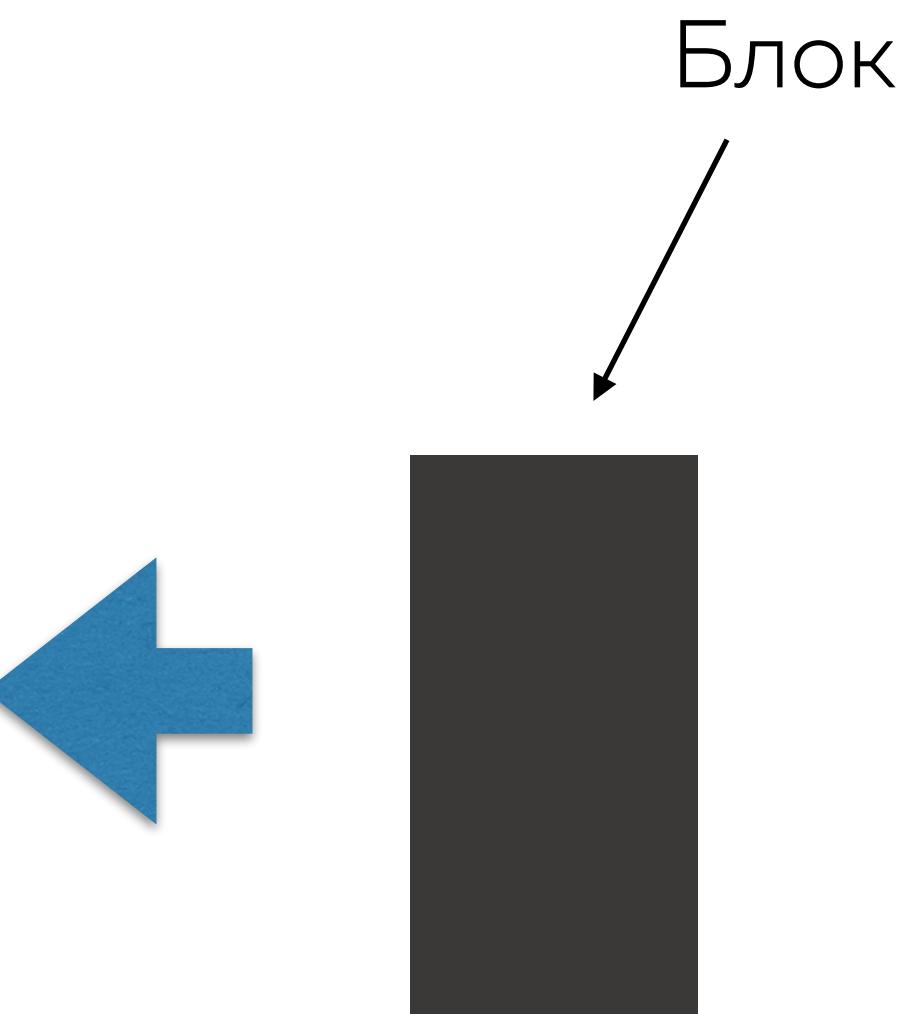
- Сложные сайты состоят из множества общих блоков: шапка, подвал, боковое меню, и т.д.
- Разделение логики сценария и отображения
- Легче читать и поддерживать
- Разделение труда:
 - верстальщик – верстает, программист – программирует



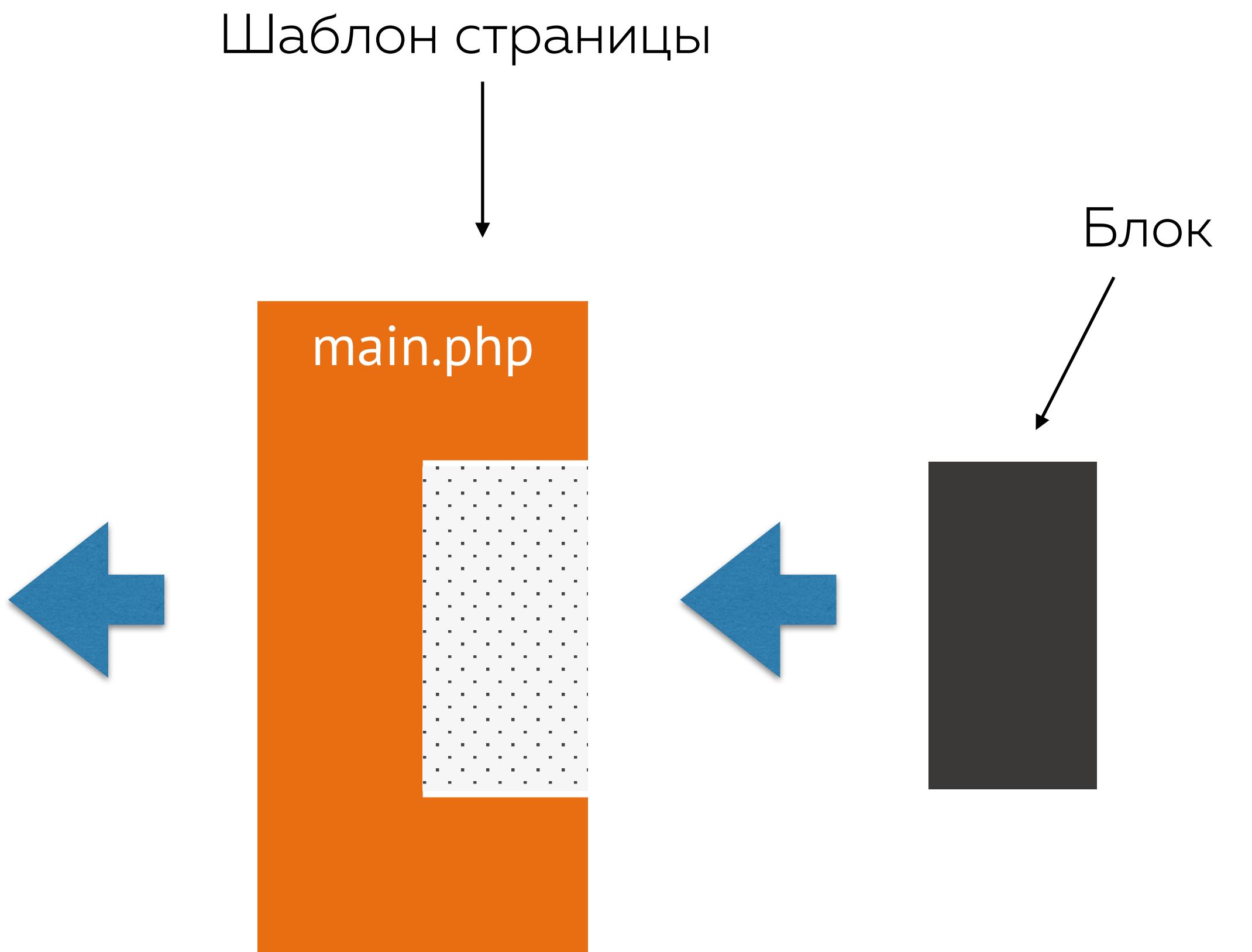
Сбор страницы из шаблонов



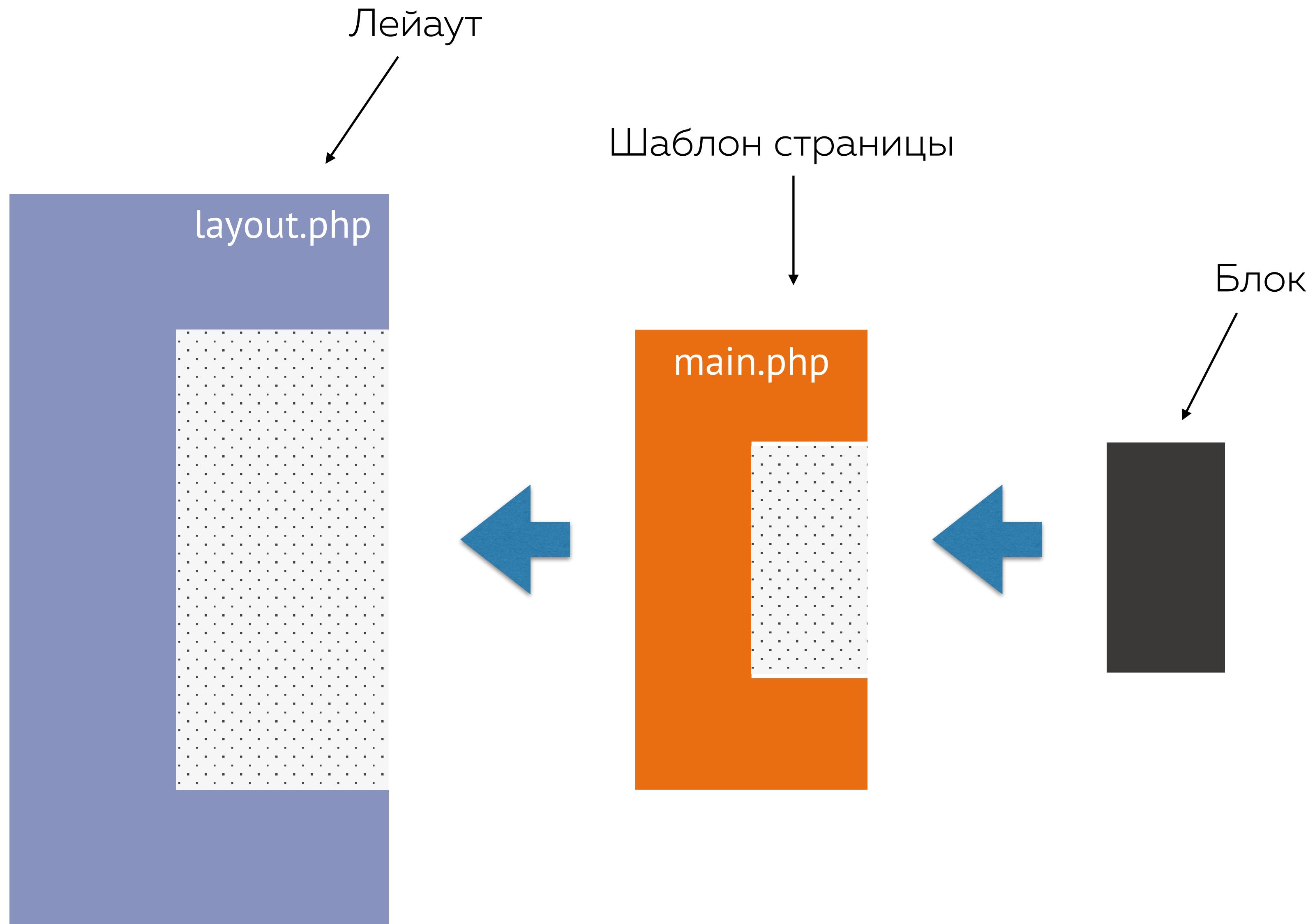
Сбор страницы из шаблонов



Сбор страницы из шаблонов



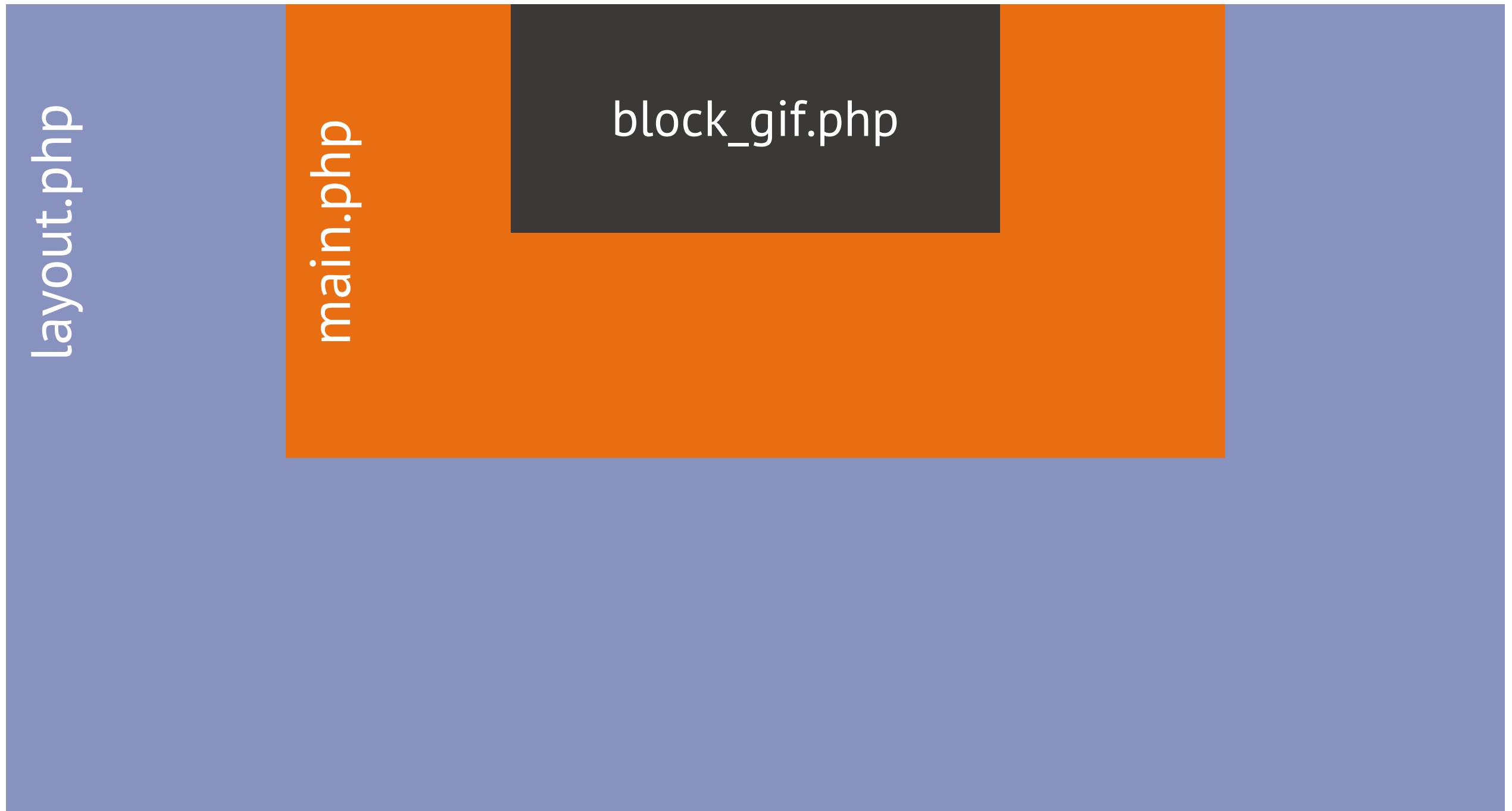
Сбор страницы из шаблонов



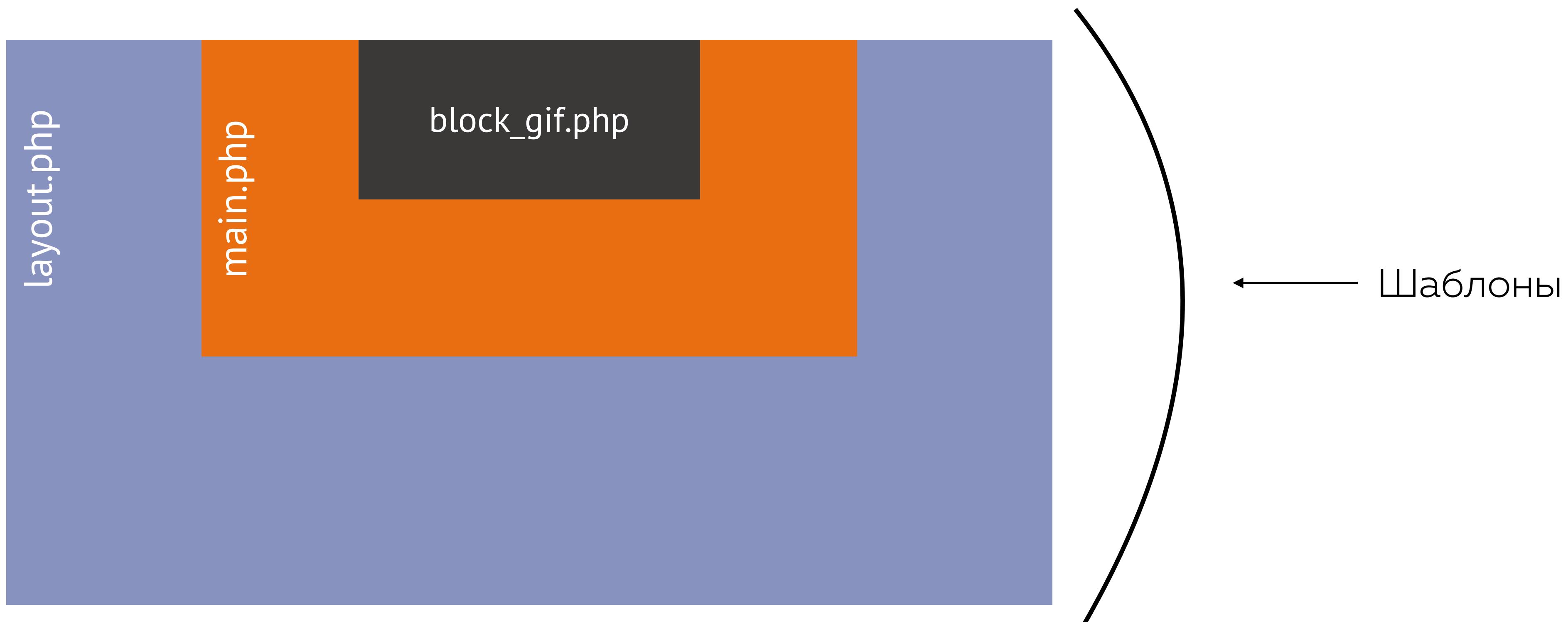
Сбор страницы из шаблонов



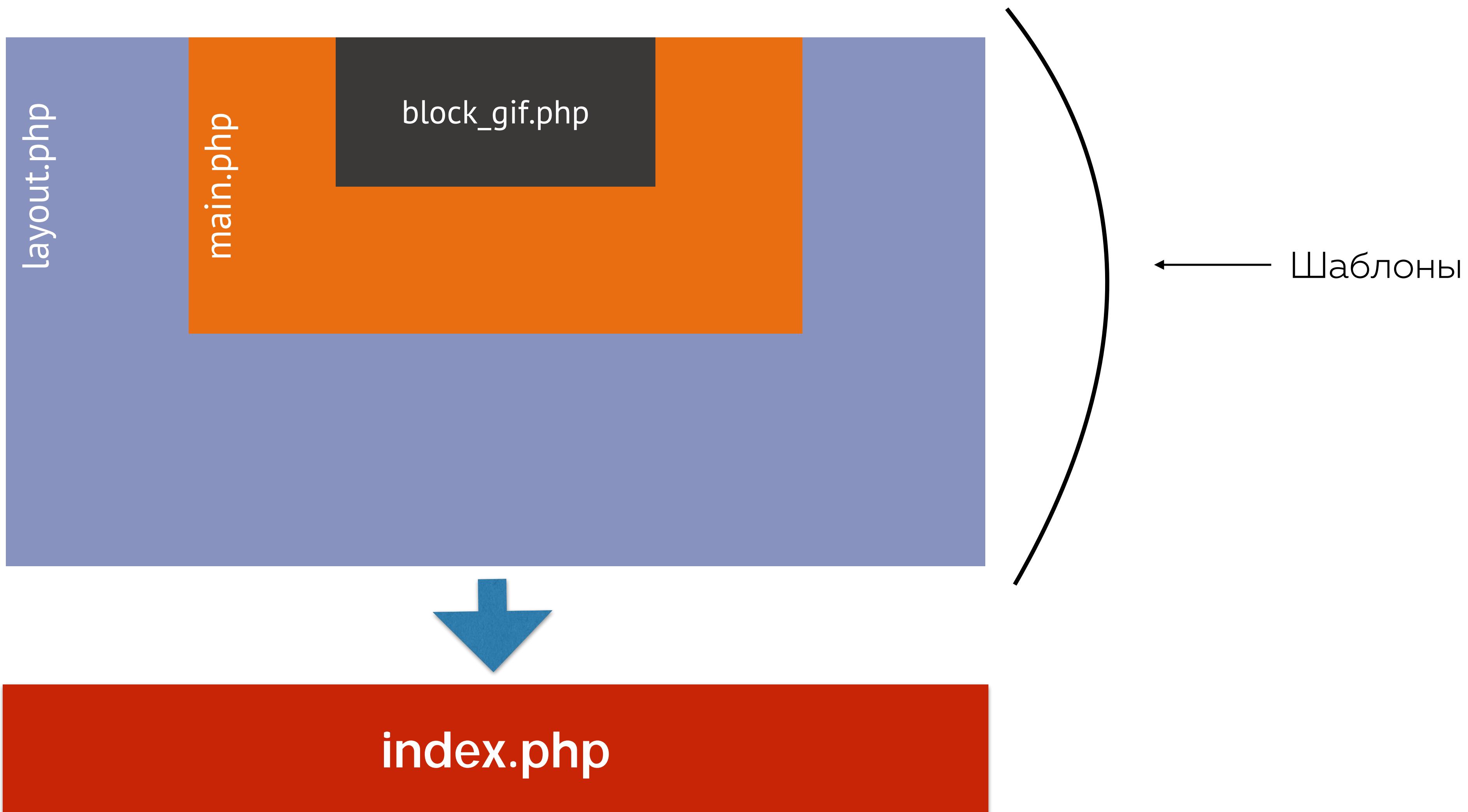
Сбор страницы из шаблонов



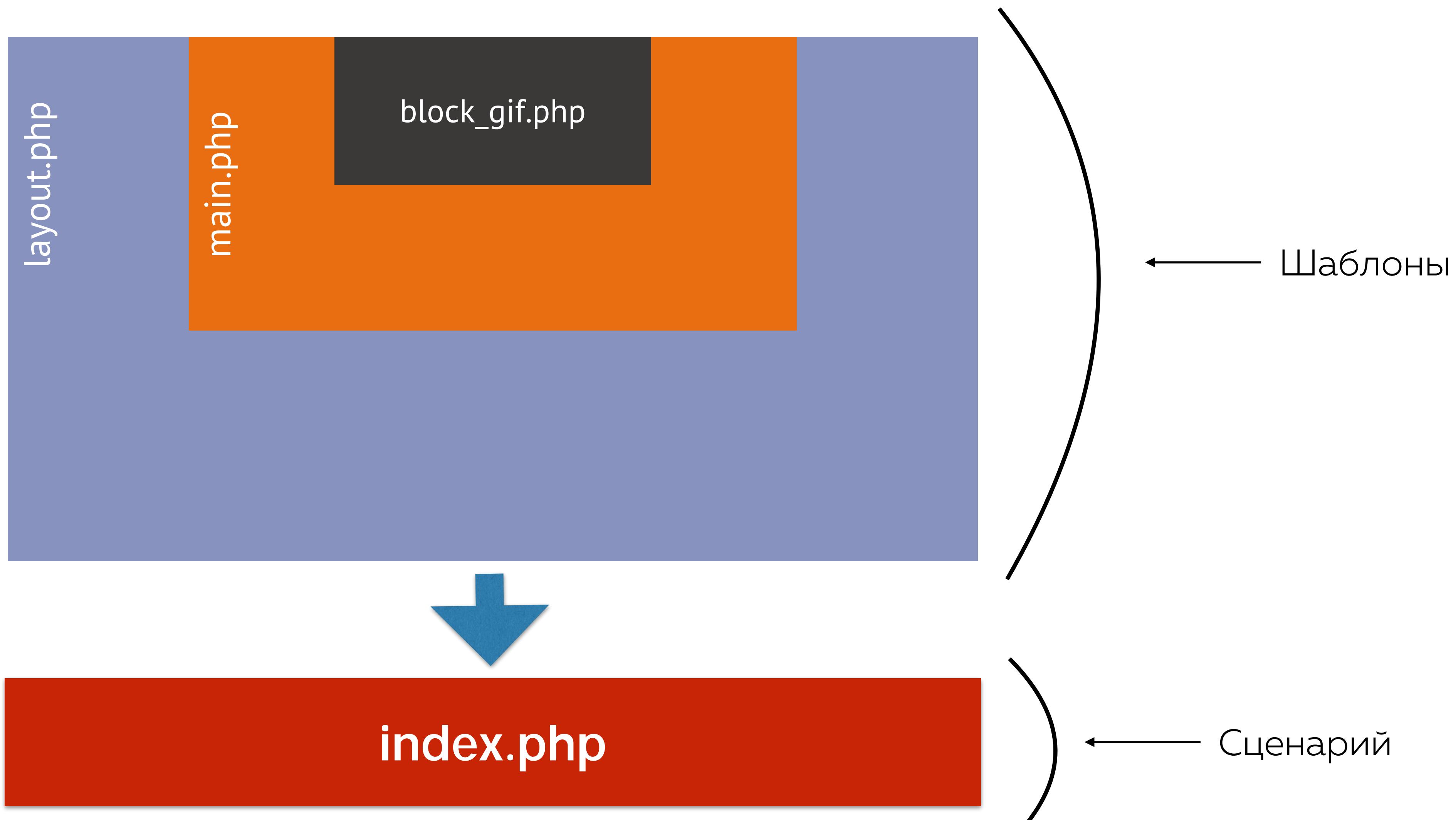
Сбор страницы из шаблонов



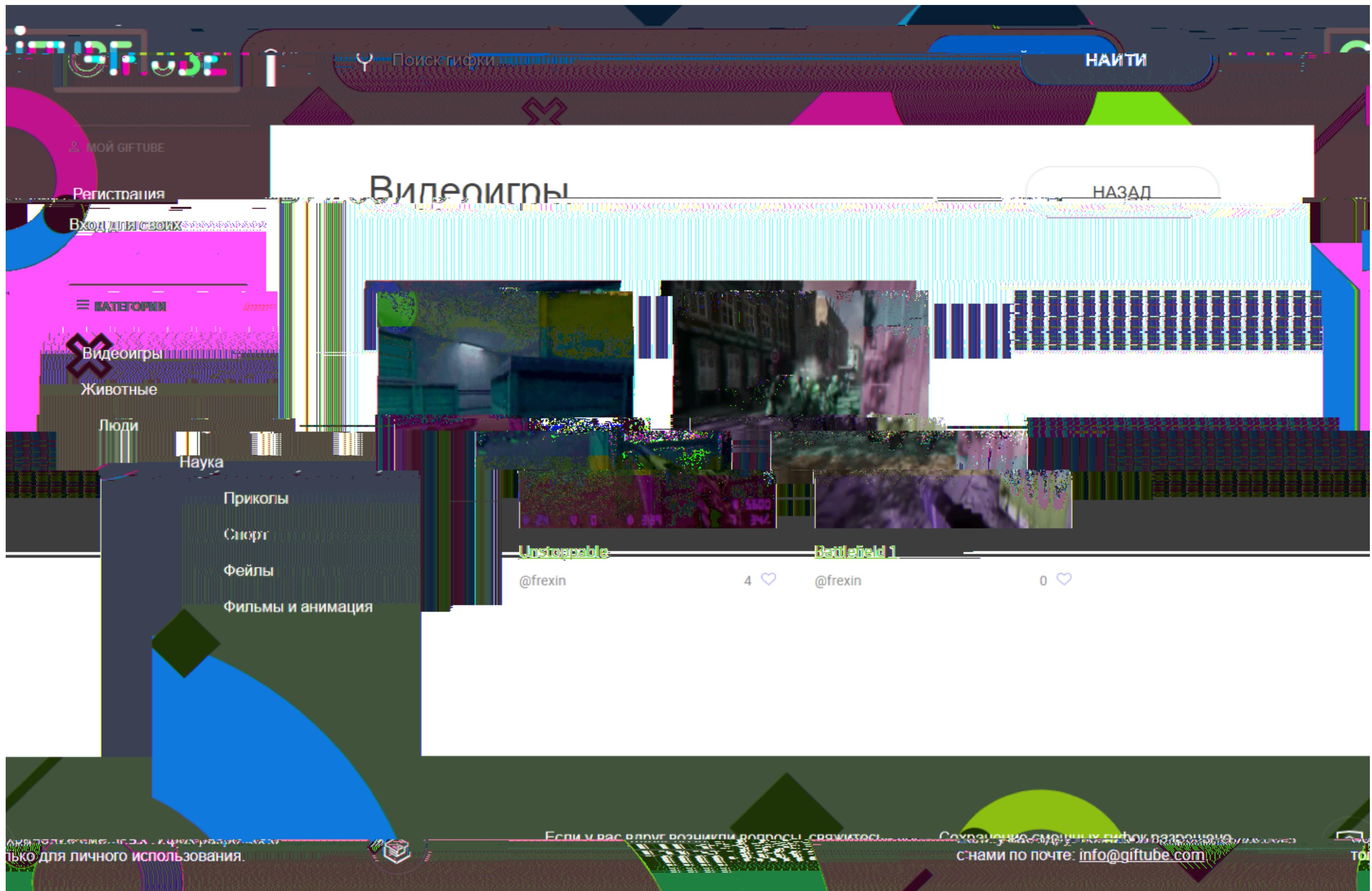
Сбор страницы из шаблонов

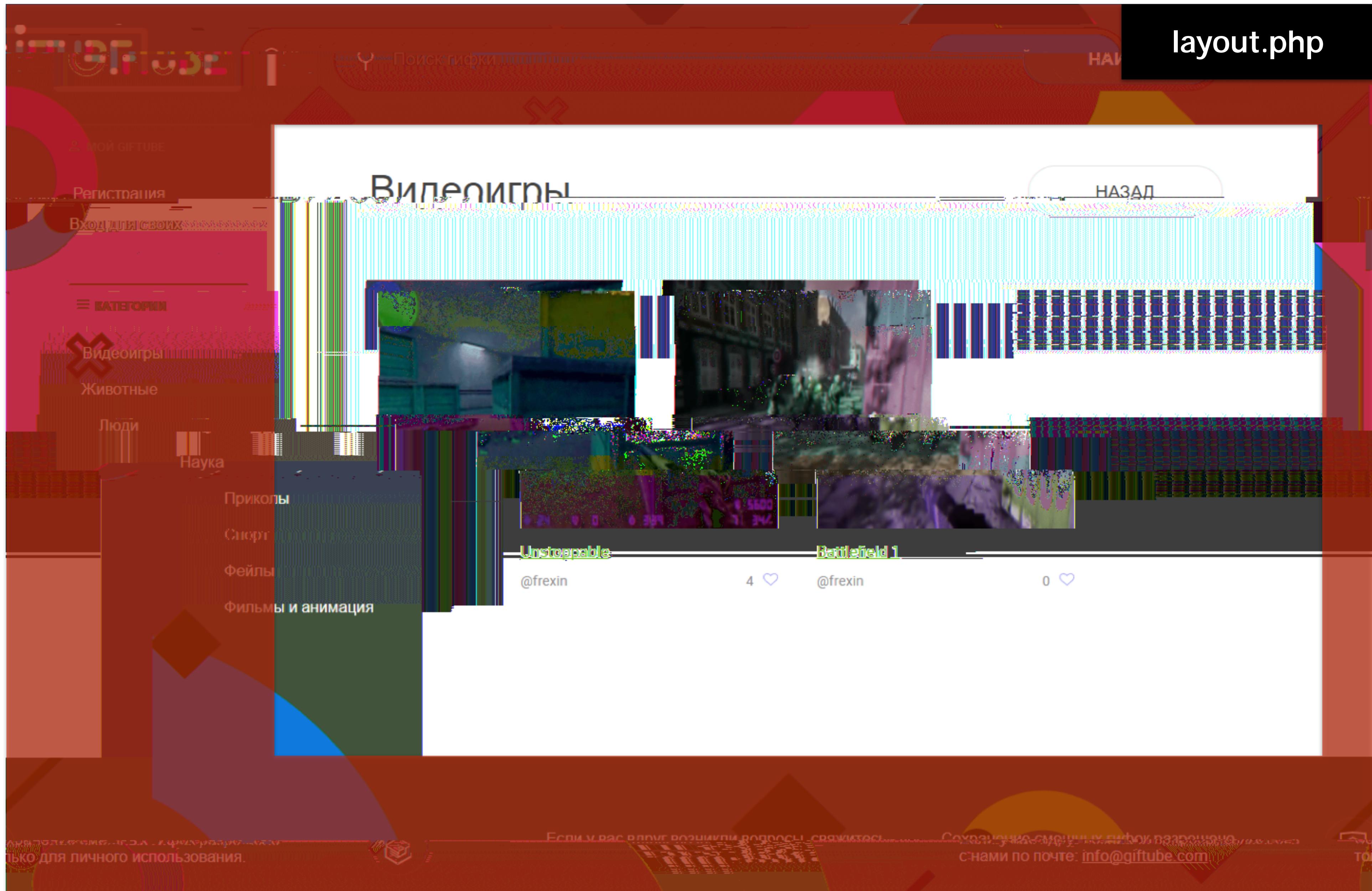


Сбор страницы из шаблонов



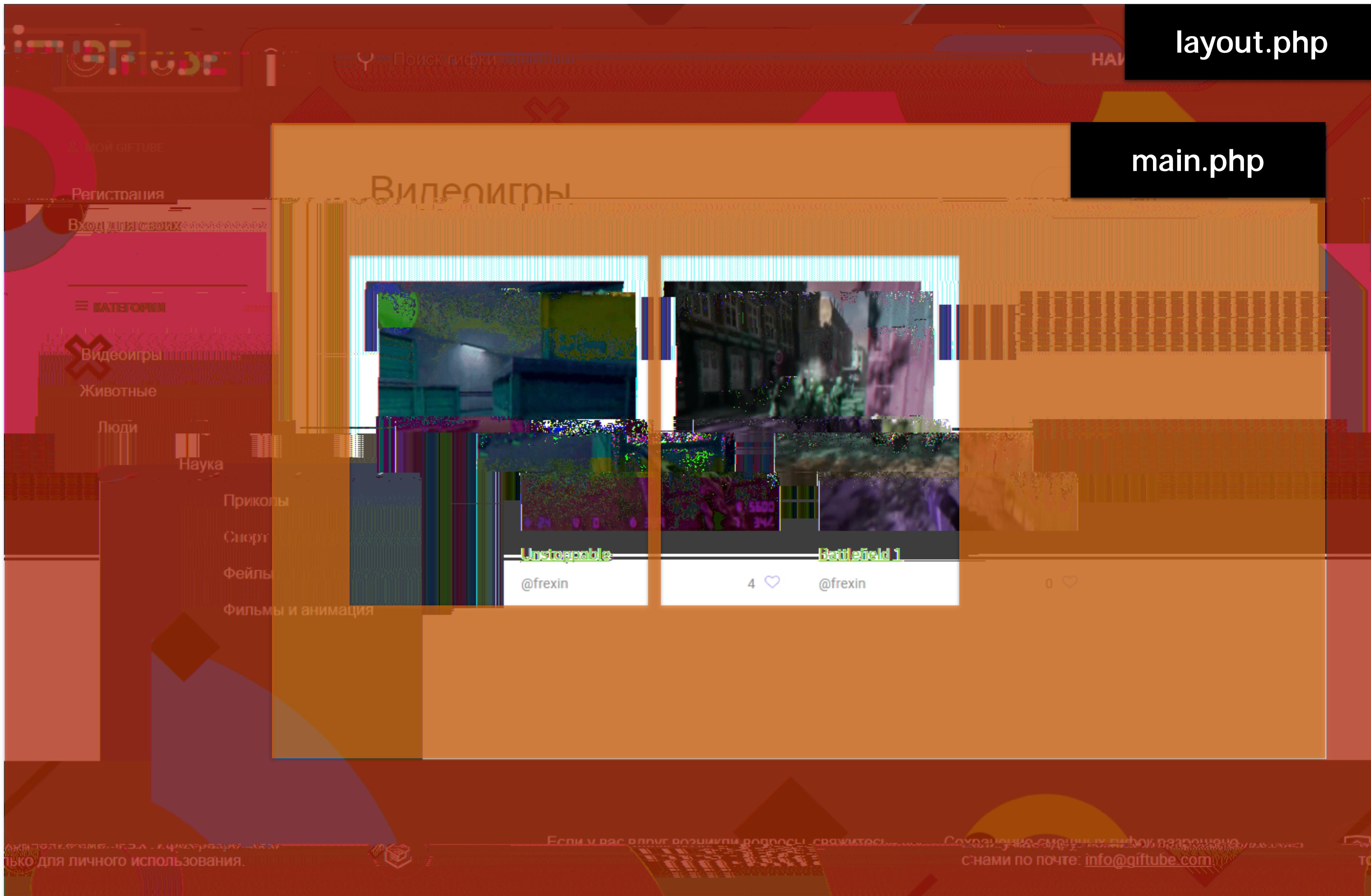


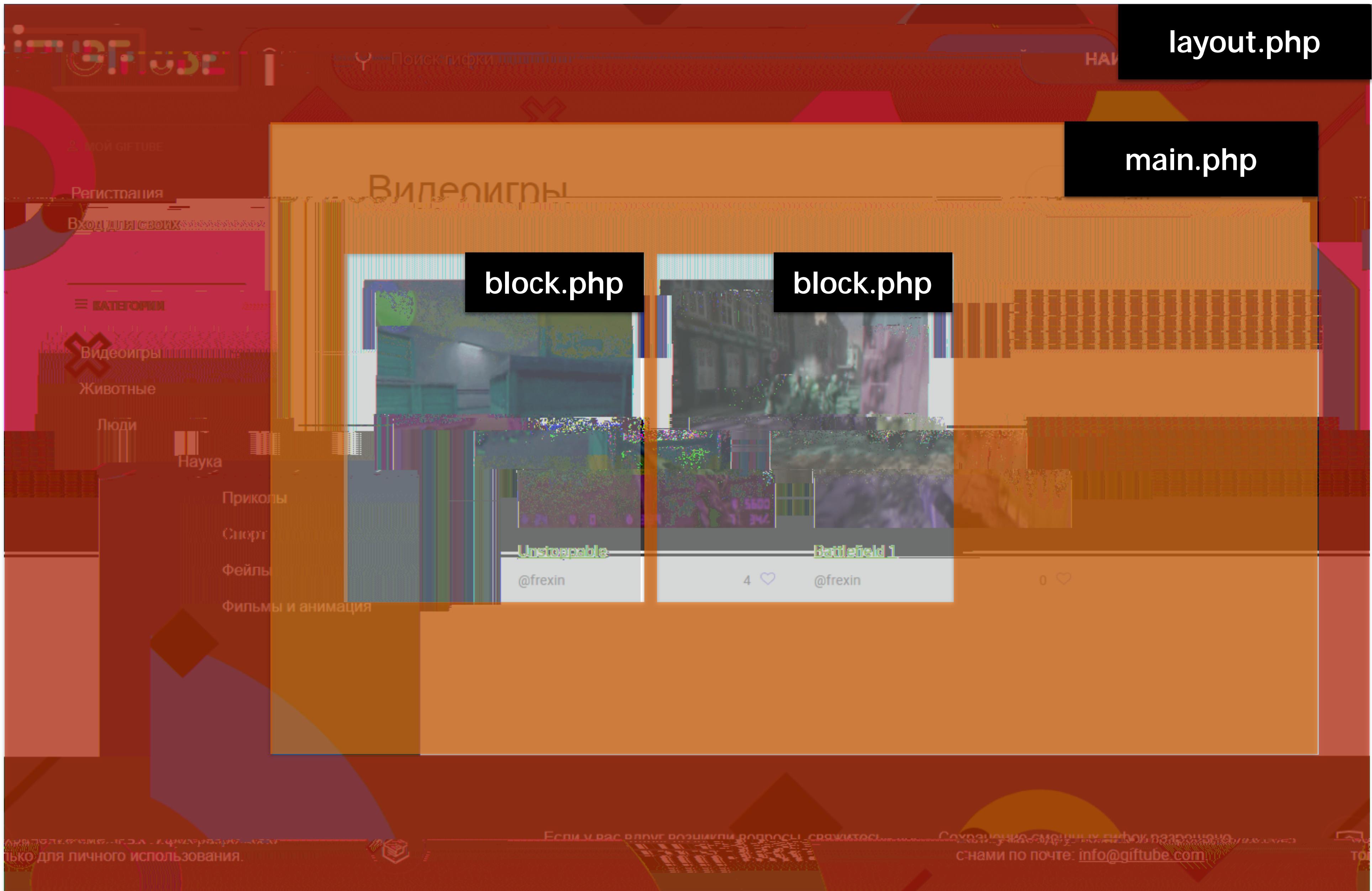




layout.php

main.php







Перерыв на 10 минут



dws4.me



Термины шаблонизации



Термины шаблонизации

Лейаут (layout)

Файл с HTML кодом общих частей страницы:

стили, метатеги, шапка, подвал.

Содержит область для вставки контента страницы.



Термины шаблонизации

Лейаут (layout)

Файл с HTML кодом общих частей страницы:

стили, метатеги, шапка, подвал.

Содержит область для вставки контента страницы.

Шаблон страницы

Файл с уникальным для одной страницы HTML кодом.

Например: страница регистрации, главная, список пользователей.

Может включать в себя блоки



Термины шаблонизации

Лейаут (layout)

Файл с HTML кодом общих частей страницы:

стили, метатеги, шапка, подвал.

Содержит область для вставки контента страницы.

Шаблон страницы

Файл с уникальным для одной страницы HTML кодом.

Например: страница регистрации, главная, список пользователей.

Может включать в себя блоки

Блок

Файл с HTML кодом небольшого блока страницы.

Например: превью гифки, меню.

Один блок может использоваться в разных шаблонах страниц



Изоляция данных

Любой компонент (лейаут, шаблон страницы, блок)
должен иметь доступ только к данным, которые ему явно передали

Чтобы обеспечить такую изоляцию, файл шаблона нужно
подключать внутри функции.

Сама функция должна принимать данные для шаблона.



Шаблонизатор

функция, которая подключает файл шаблона,
передаёт ему данные и возвращает
сгенерированный HTML



Как работает шаблонизатор



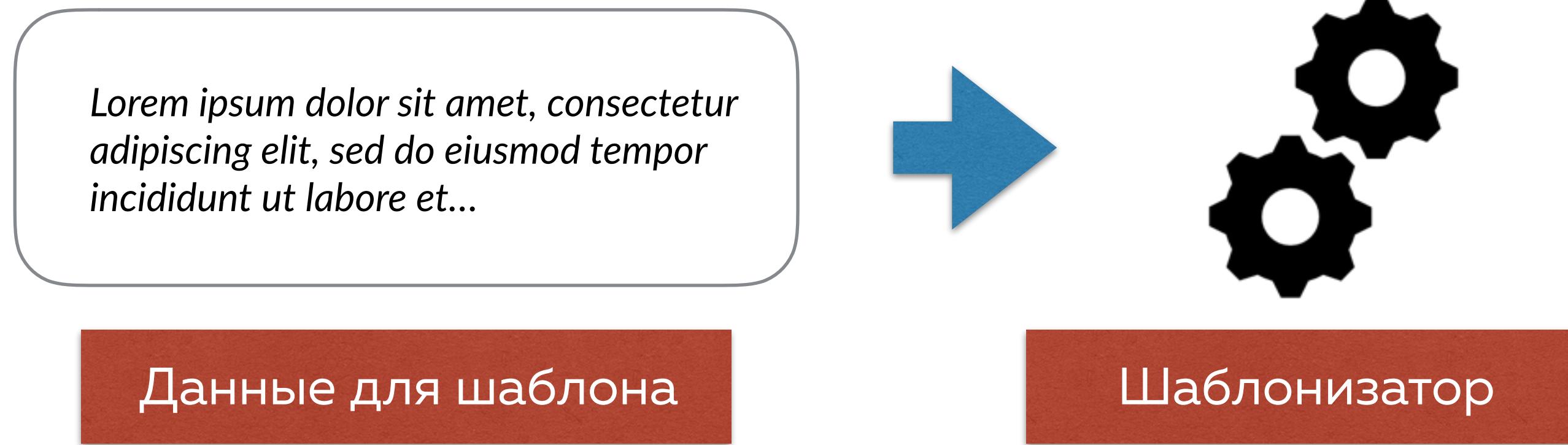
Как работает шаблонизатор

*Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor
incididunt ut labore et...*

Данные для шаблона



Как работает шаблонизатор



Как работает шаблонизатор



Из чего состоит шаблон

Любой шаблон (шаблон страницы, лейаут, блок)

может состоять только из:

- HTML кода
- Переменных и вызова функций
- Простых языковых конструкций (условия и циклы)
- Включения других шаблонов



План работы

1. Создать базовый лейаут
2. Создать блок для превью гифки
3. Создать шаблон для главной страницы
4. Специфицировать функцию шаблонизации
5. Написать сценарий, который подключит шаблоны
и покажет итоговый результат



```
<!DOCTYPE html>
<html lang="ru">
<head>
    <title><?= $title; ?></title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <header class="main-header">
        <h1 class="visual-h1">Gi ftube</h1>
    </header>
    <div class="main-content">
        <section class="navigation">
            <h3 class="navigation_title">Категории</h3>
            <nav class="navigation_links">
                <?php foreach ($categories as $cat) : ?>
                    <a href="/category?id=<?= $cat['id']; ?>"><?= $cat['name']; ?></a>
                <?php endforeach; ?>
            </nav>
        </section>
        <main class="content"><?= $content; ?></main>
    </div>
    <footer class="main-footer">
        <div class="main-footer_col">
            Gi fTube: <a href="mailto:info@gi ftube.com">info@gi ftube.com</a>.
        </div>
    </footer>
</div>
</body>
</html>
```



```
<!DOCTYPE html>
<html lang="ru">
<head>
    <title><?= $title; ?></title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <header class="main-header">
        <h1 class="visual-hidden">Giftube</h1>
    </header>
    <div class="main-content">
        <section class="navigation">
            <h3 class="navigation_title">Категории</h3>
            <nav class="navigation_links">
                <?php foreach ($categories as $cat) : ?>
                    <a href="/category?id=<?= $cat['id']; ?>"><?= $cat['name']; ?></a>
                <?php endforeach; ?>
            </nav>
        </section>
        <main class="content"><?= $content; ?></main>
    </div>
    <footer class="main-footer">
        <div class="main-footer_col">
            GifTube: <a href="mailto:info@giftube.com">info@giftube.com</a>.
        </div>
    </footer>
</div>
</body>
</html>
```

Динамическое содержимое
лейаута



```
<!DOCTYPE html>
<html lang="ru">
<head>
    <title><?= $title; ?></title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="container">
    <header class="main-header">
        <h1 class="visual-hidden">Giftube</h1>
    </header>
    <div class="main-content">
        <section class="navigation">
            <h3 class="navigation_title">Категории</h3>
            <nav class="navigation_links">
                <?php foreach ($categories as $cat) : ?>
                    <a href="/category?id=<?= $cat['id']; ?>"><?= $cat['name']; ?></a>
                <?php endforeach; ?>
            </nav>
        </section>
        <main class="content"><?= $content; ?></main>
    </div>
    <footer class="main-footer">
        <div class="main-footer_col">
            GifTube: <a href="mailto:info@giftube.com">info@giftube.com</a>.
        </div>
    </footer>
</div>
</body>
</html>
```

Динамическое содержимое
лейаута

Сюда будет помещён
шаблон страницы



Блок для превью гифки

```
<li class="gif gif-list-item">
  <div class="gif-picture">
    <a href="/gif/view?id=<?=$gif['id'];?>" class="gif-preview">
      
    </a>
  </div>
  <div class="gif-description">
    <h3 class="gif-description-title">
      <a href="/gif/view?id=<?=$gif['id'];?>><?=$gif['title'];?></a>
    </h3>

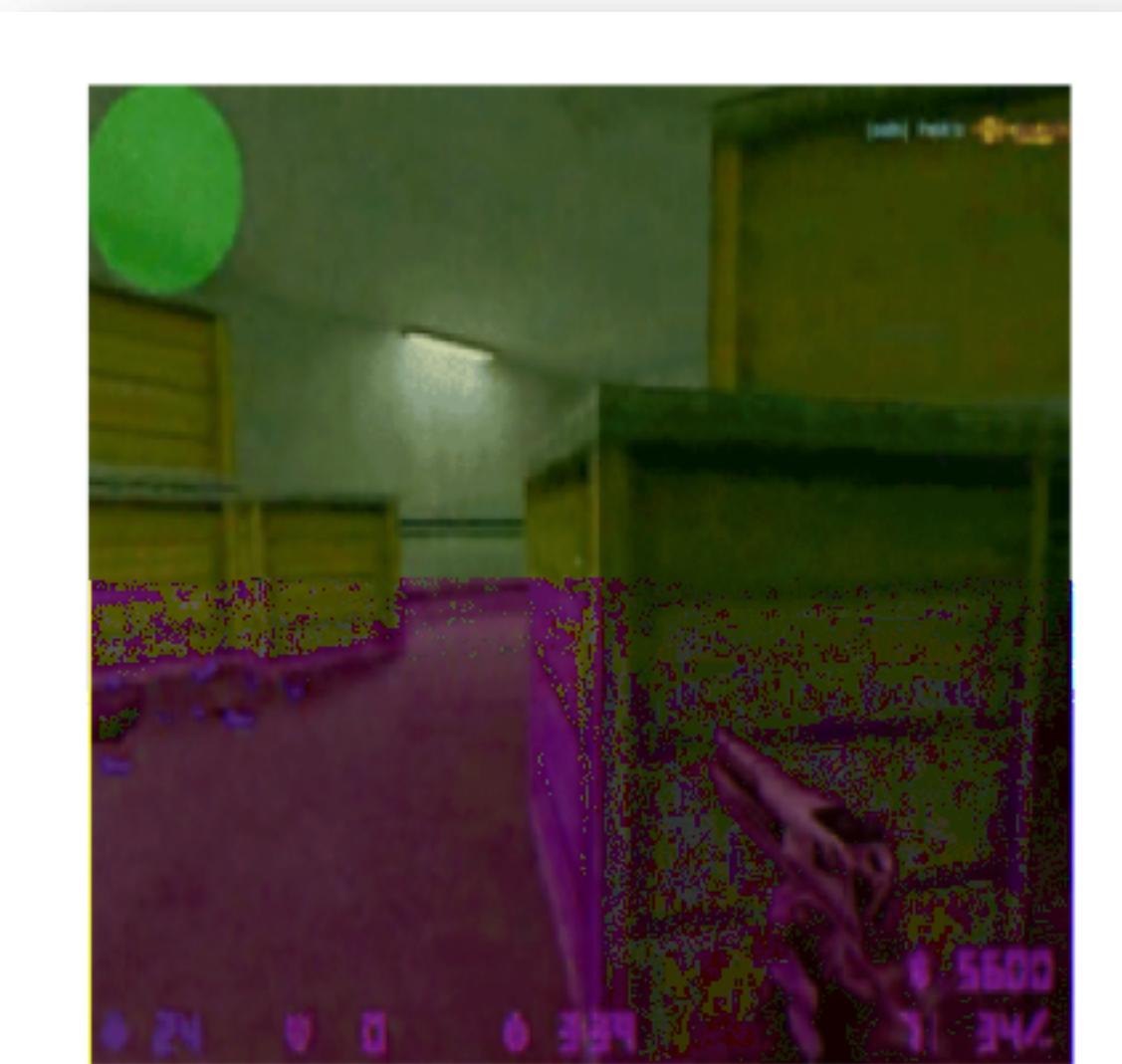
    <div class="gif-description-data">
      <span class="gif-likes"><?=$gif['like_count'];?></span>
    </div>
  </div>
</li>
```



Блок для превью гифки

```
<li class="gif gif-list-item">
  <div class="gif-picture">
    <a href="/gif/view?id=<?=$gf['id'];?>" class=""
      
    </a>
  </div>
  <div class="gif-description">
    <h3 class="gif-description-title">
      <a href="/gif/view?id=<?=$gf['id'];?>"><?=
    </h3>

    <div class="gif-description-data">
      <span class="gif-likes"><?=$gf['like_count']>
    </div>
  </div>
</li>
```

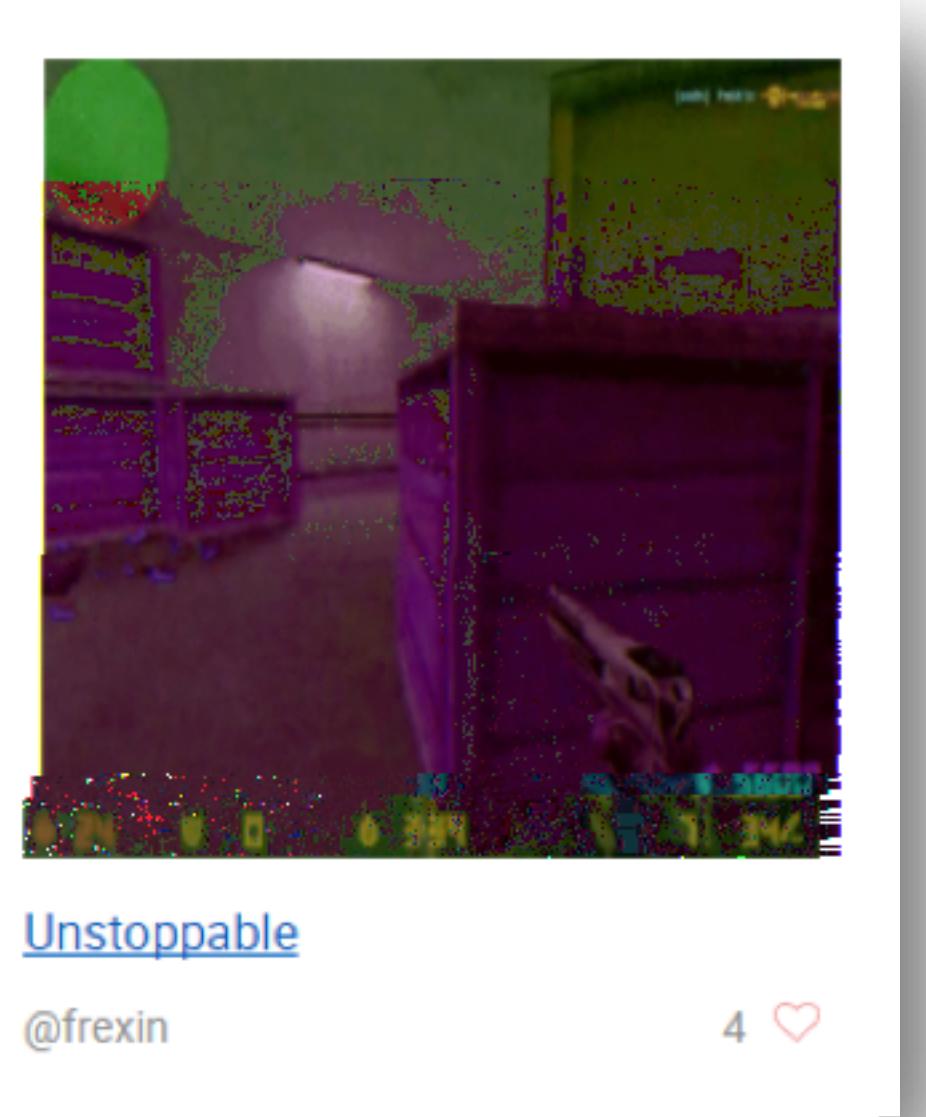


Unstoppable

@frexin

4





[Unstoppable](#)

@frexin

4 ❤

Похожие гифки:



[Unstoppable](#)

@frexin

6 ❤



[Battlefield 1](#)

@frexin

0 ❤

Топовые гифки Свежачок

Загрузить свою

Рыжая

Unstoppable

Типичный юзер

2 ❤ @frexin 10 ❤ @frexin 6 ❤ @frexin



Топовые гифки Свежачок

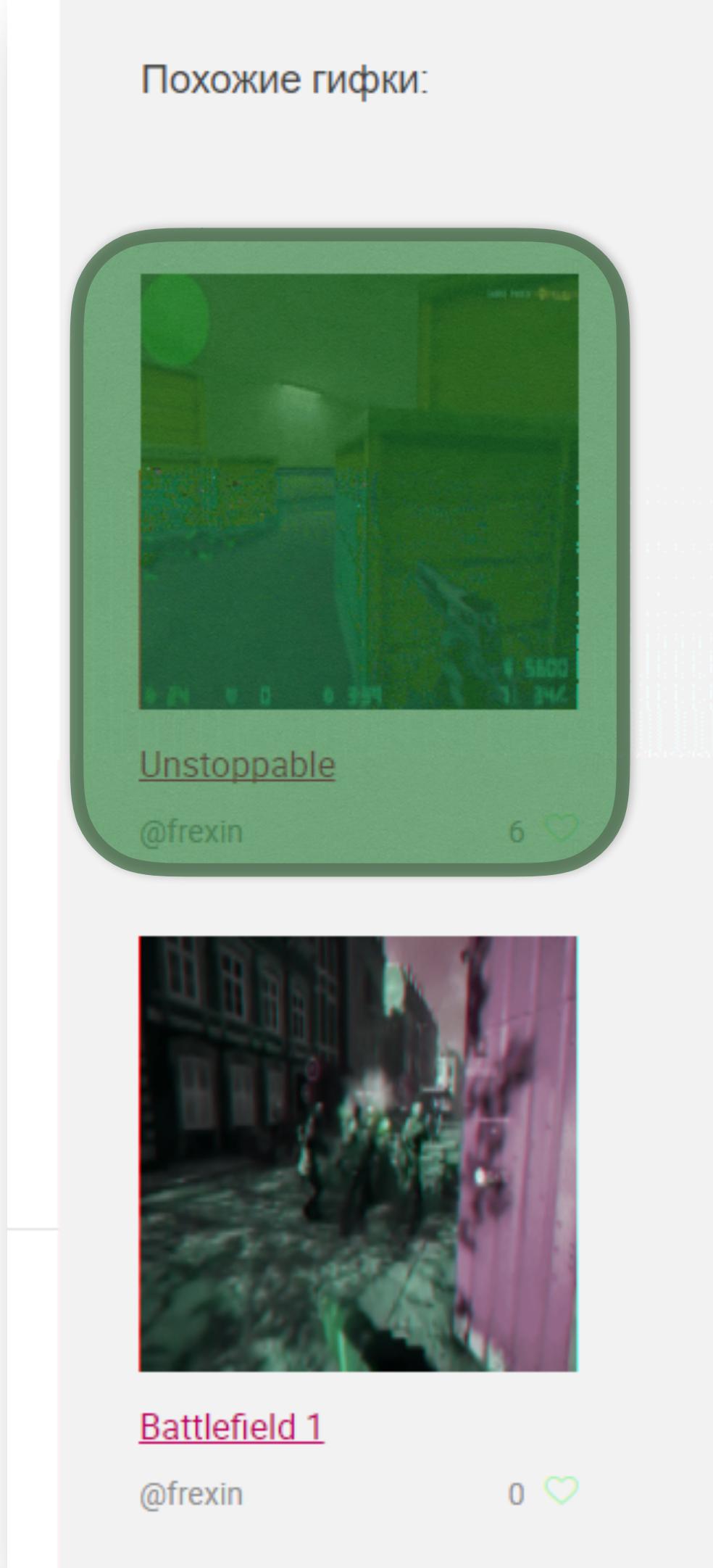
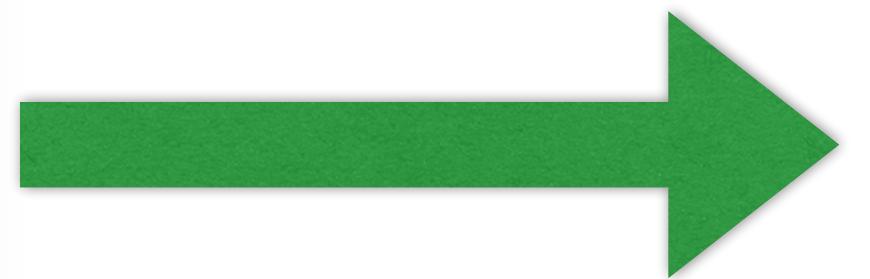
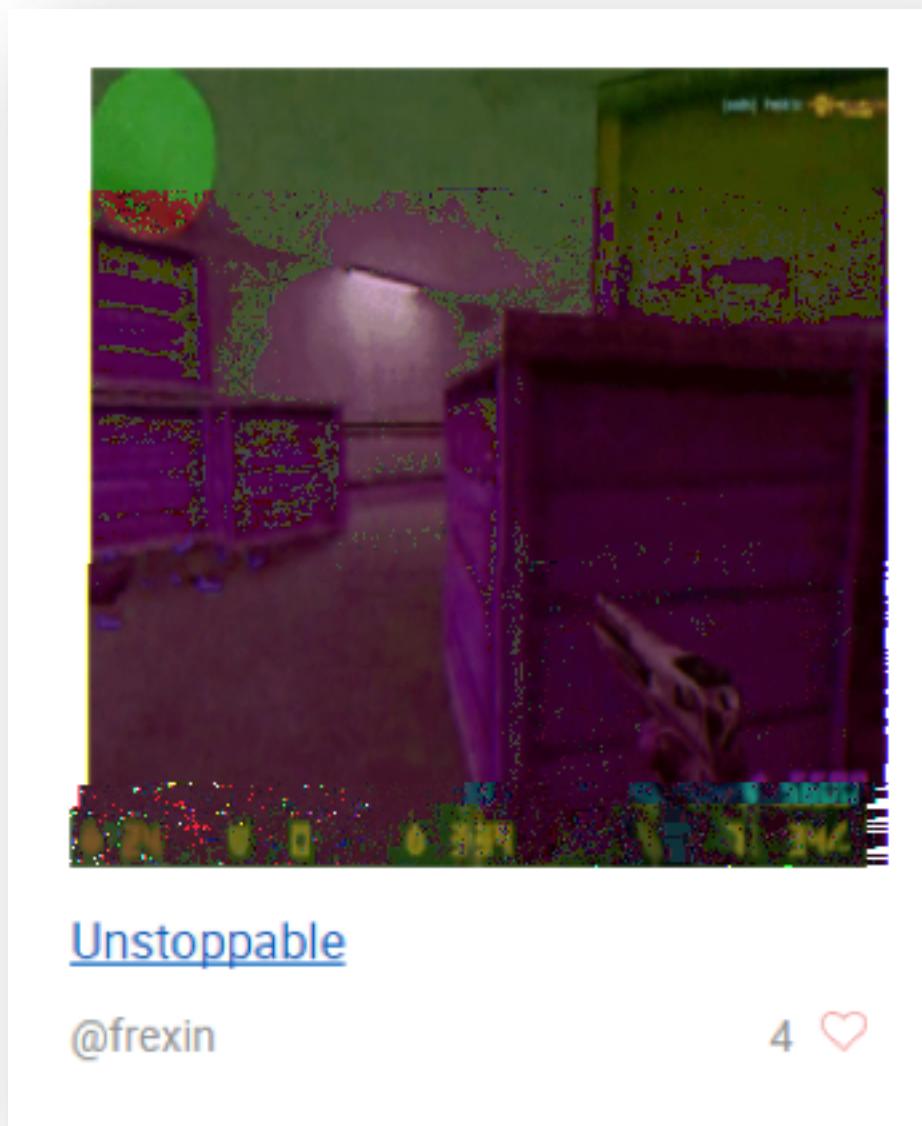
Загрузить свою

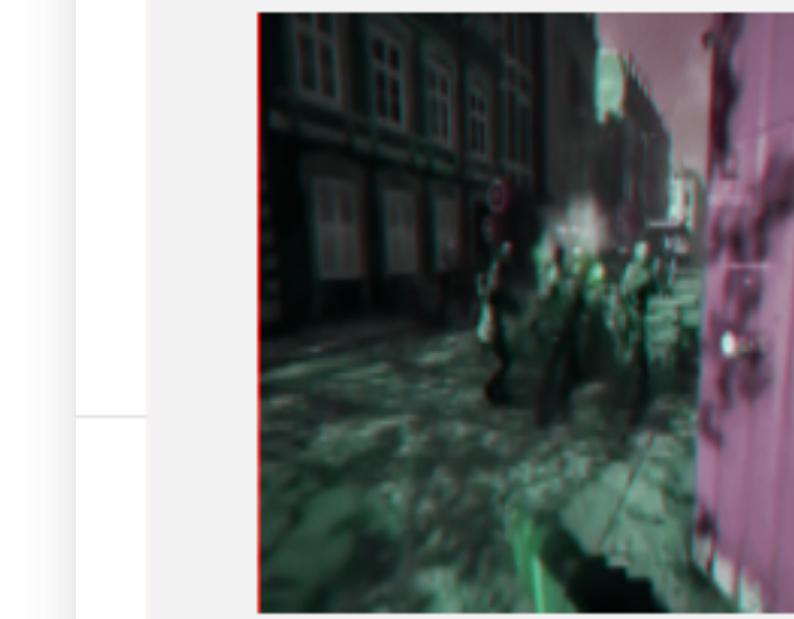
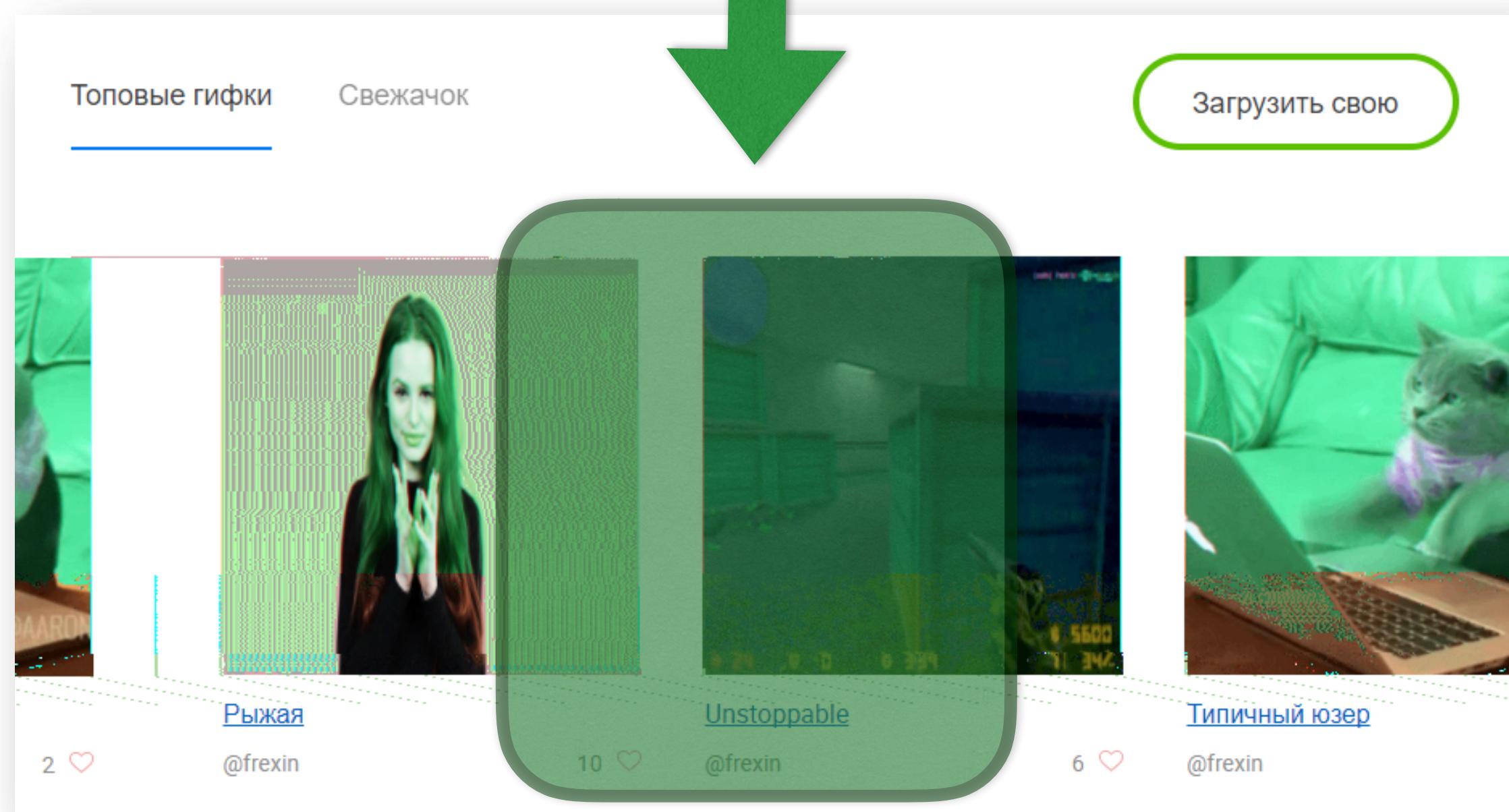
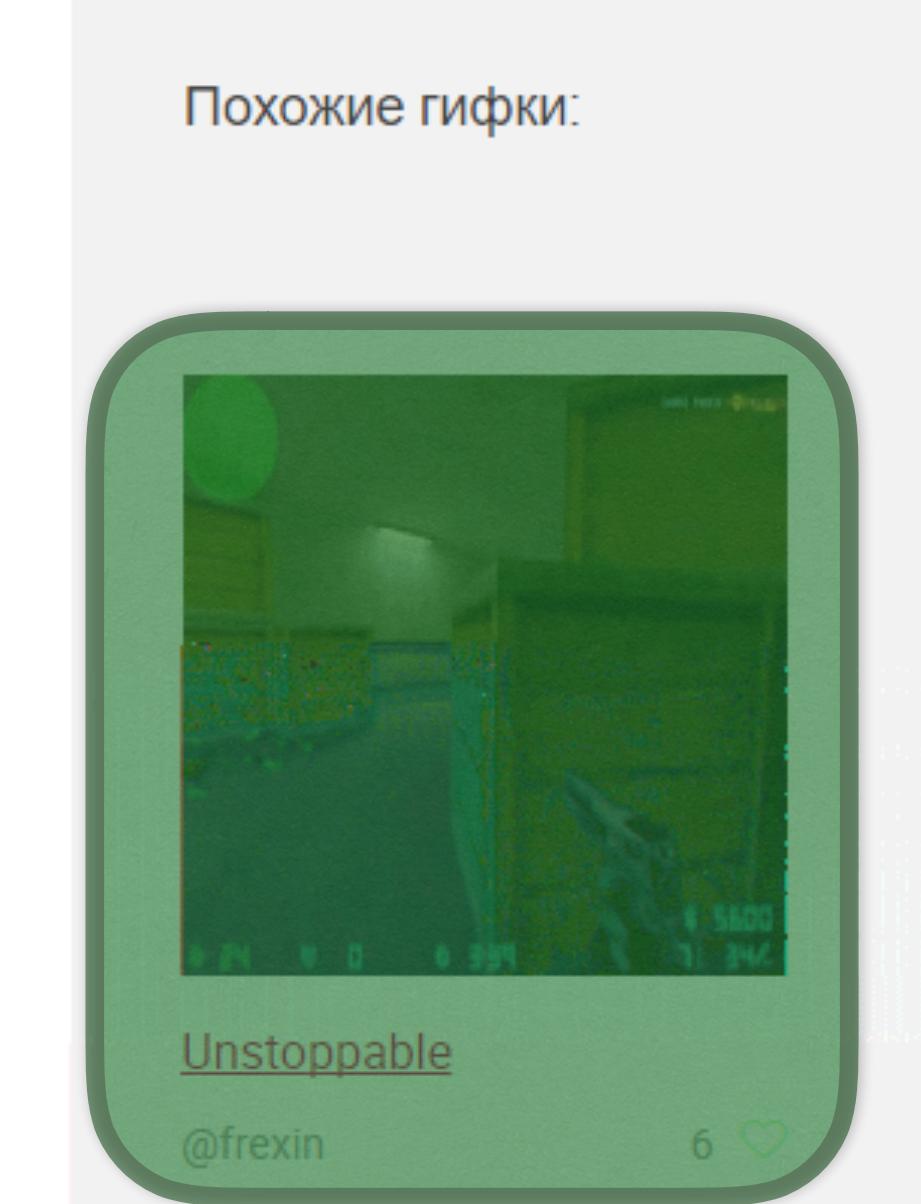
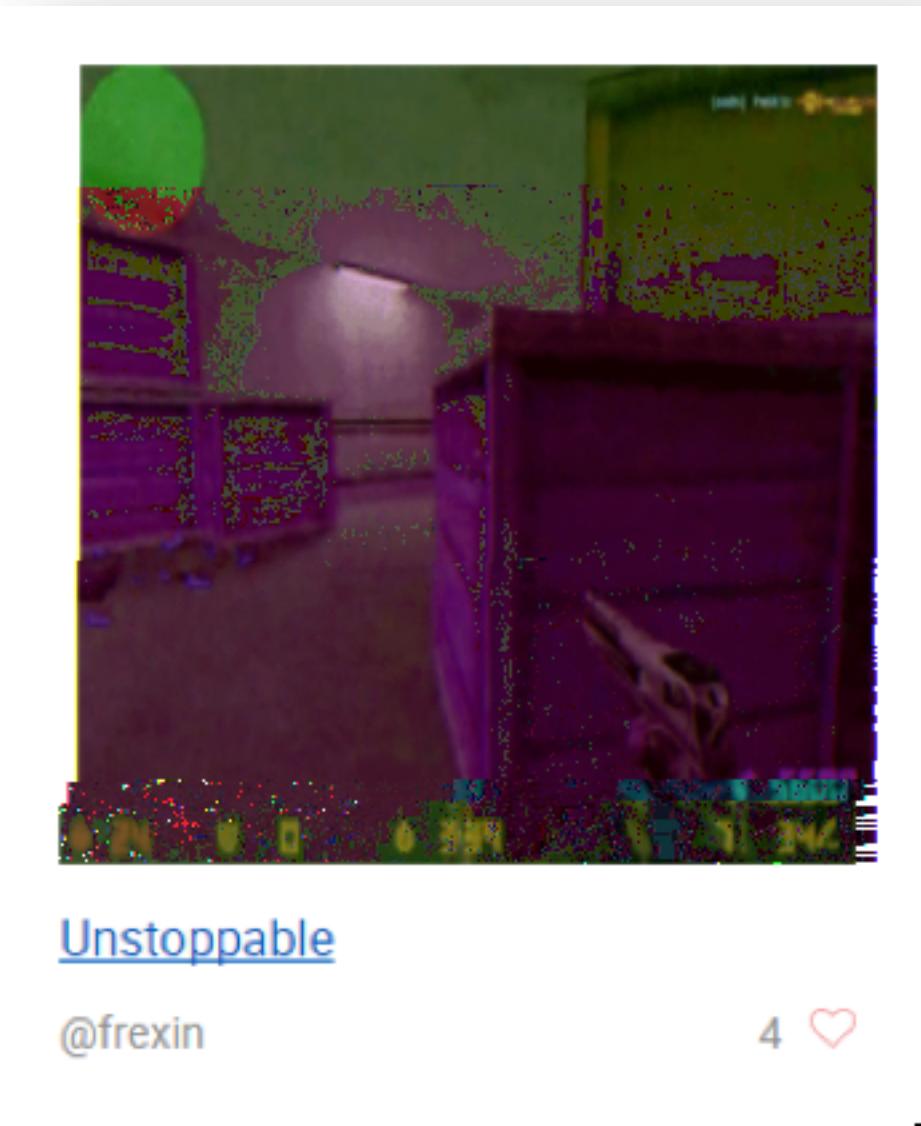
Рыжая
@frexin 2 ❤️

Unstoppable
@frexin 10 ❤️

Типичный юзер
@frexin 6 ❤️

Unstoppable
@frexin 4 ❤️





Шаблон страницы

Подключаем и выводим блок с превью гифки

```
<div class="content_main-col">
    <h2 class="visual-h1">Смешные гифки</h2>
    <a class="button" href="/gifs/add">Загрузить свою</a>

    <ul class="gifs-list">
        <?php foreach($gifs_col as $gifs): ?>
            <?=renderTemplate('inc/_gifs_grid', ['gifs' => $gifs]); ?>
        <?php endforeach; ?>
    </ul>
</div>
```



Шаблон страницы

Топовые гифки Свежачок

Загрузить свою

Рыжая @frexin 4 ❤️

Unstoppable @frexin 4 ❤️

Типичный юзер @frexin 1 ❤️

Функция-шаблонизатор



Функция-шаблонизатор

Аргументы функции:



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона
2. массив с данными шаблона



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона
2. массив с данными шаблона

Назначение:



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона
2. массив с данными шаблона

Назначение:

подключает переданный файл, захватывает его содержимое



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона
2. массив с данными шаблона

Назначение:

подключает переданный файл, захватывает его содержимое

Что возвращает:

возвращает сгенерированный HTML код шаблона



Функция-шаблонизатор

Аргументы функции:

1. путь к файлу-шаблона
2. массив с данными шаблона

Назначение:

подключает переданный файл, захватывает его содержимое

Что возвращает:

возвращает сгенерированный HTML код шаблона



Главный сценарий

```
<?php  
// двумерный массив с гифками  
$gifs_col = [];  
  
// HTML код главной страницы  
$page_content = renderTemplate('inc/main.php', ['gifs' => $gifs_col]);  
  
// окончательный HTML код  
$layout_content = renderTemplate('inc/Layout.php',  
['content' => $page_content, 'title' => 'GifTube - Главная']);  
  
print($layout_content);
```



Главный сценарий

```
Данные для передачи в шаблон  
<?php  
// двумерный массив с гифками  
$gi fs_col = [];  
  
// HTML код главной страницы  
$page_content = renderTemplate('inc/main.php', ['gi fs' => $gi fs_col]);  
  
// окончательный HTML код  
$layout_content = renderTemplate('inc/Layout.php',  
    ['content' => $page_content, 'title' => 'Gi fTube - Главная']);  
  
print($layout_content);
```



Главный сценарий

```
<?php  
// двумерный массив с гифками  
$gi fs_col = [];  
  
// HTML код главной страницы  
$page_content = renderTemplate('inc/main.php', ['gi fs' => $gi fs_col]);  
  
// окончательный HTML код  
$layout_content = renderTemplate('inc/Layout.php',  
['content' => $page_content, 'title' => 'Gi fTube - Главная']);  
  
print($layout_content);
```

Данные для передачи в шаблон

Подключаем шаблон страницы и передаем туда данные



Главный сценарий

```
<?php  
// двумерный массив с гифками  
$gi fs_col = [];  
  
// HTML код главной страницы  
$page_content = renderTemplate('inc/main.php', ['gi fs' => $gi fs_col]);  
  
// окончательный HTML код  
$layout_content = renderTemplate('inc/layout.php',  
['content' => $page_content, 'title' => 'Gi fTube - Главная']);  
  
print($layout_content);
```

Данные для передачи в шаблон

Подключаем шаблон страницы и передаем туда данные

Подключаем лейаут и передаем туда результат шаблонизации страницы



Главный сценарий

```
<?php  
// двумерный массив с гифками  
$gi fs_col = [];  
  
// HTML код главной страницы  
$page_content = renderTemplate('inc/main.php', ['gi fs' => $gi fs_col]);  
  
// окончательный HTML код  
$layout_content = renderTemplate('inc/layout.php',  
['content' => $page_content, 'title' => 'Gi fTube - Главная']);  
  
print($layout_content);
```

Данные для передачи в шаблон

Подключаем шаблон страницы и передаем туда данные

Подключаем лейаут и передаем туда результат шаблонизации страницы

Выводим результат на экран



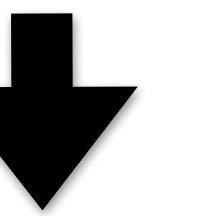
Буферизация вывода

накапливание пользовательского вывода во
внутреннем «буфере» перед его действительным
показом

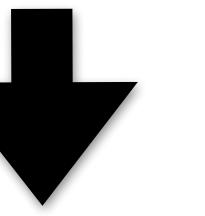


Как работает буферизация

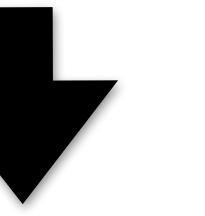
```
print("Привет!");
```



Пользовательский буфер



Системный буфер



Вывод



Системный буфер вывода

По умолчанию PHP накапливает весь вывод в своем буфере, а затем выводит всё вместе



Системный буфер вывода

По умолчанию PHP накапливает весь вывод в своем буфере, а затем выводит всё вместе

Но можно заставить PHP «сбрасывать» весь буфер на экран после каждого вывода



Буферизация вывода

Функции, вроде print или echo выводят переданную информацию на экран.

Но PHP позволяет «захватывать» их вывод, чтобы получить это содержимое и, например, записать его в файл или использовать позже.



Буферизация вывода

Функции, вроде print или echo выводят переданную информацию на экран.

Но PHP позволяет «захватывать» их вывод, чтобы получить это содержимое и, например, записать его в файл или использовать позже.

Техника буферизации состоит из двух частей:

1. включить буферизацию вывода
2. получить содержимое буфера и очистить его



Буферизация вывода

Пример использования буферизации

```
<?php
    ob_start();
    print("Любой текст, вместо показа будет сохранен в буфере");
?
<h1>some title</h1>
<?php $html = ob_get_clean();
```

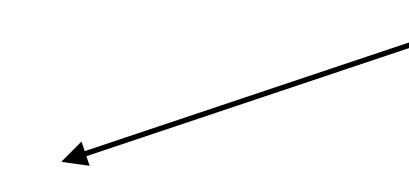


Буферизация вывода

Пример использования буферизации

```
<?php  
    ob_start();  
    print("Любой текст, вместо показа будет сохранен в буфере");  
?  
<h1>some title</h1>  
<?php $html = ob_get_clean();
```

Включаем буферизацию вывода



Буферизация вывода

Пример использования буферизации

```
<?php  
    ob_start();  
    print("Любой текст, вместо показа будет сохранен в буфере");  
?  
<h1>some title</h1>  
<?php $html = ob_get_clean();
```

Включаем буферизацию вывода

Всё содержимое сохранено
в переменной вместо вывода
на экран



Безопасность в шаблонах



Фильтрация данных

Любую информацию, полученную от пользователя, обязательно надо **фильтровать** перед выводом в шаблоне!



XSS – межсайтовый скрипting

внедрение вредоносного JS-кода на страницу,
которое становится возможным из за недостаточной
фильтрации данных от пользователя

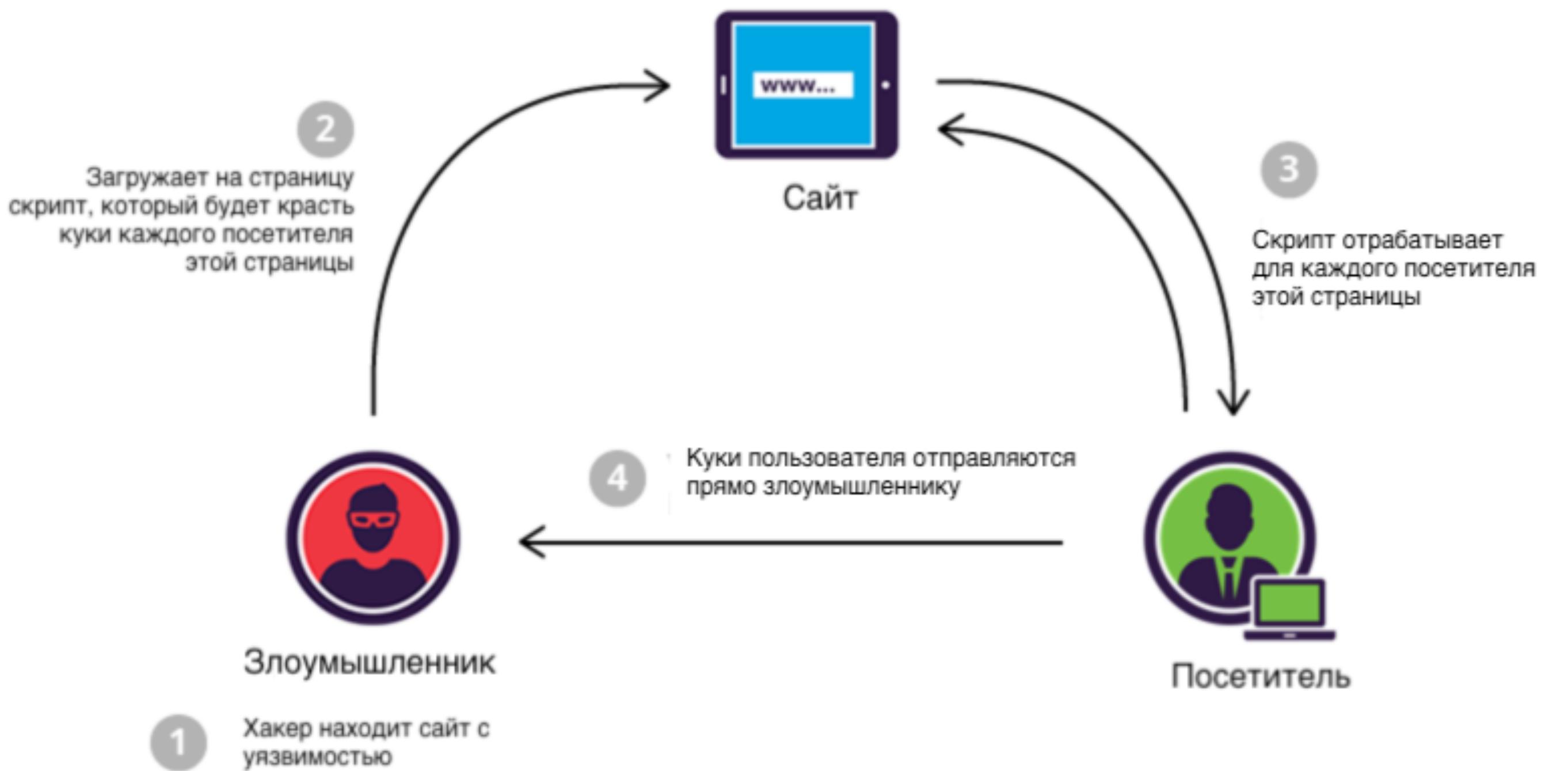


Принцип атаки

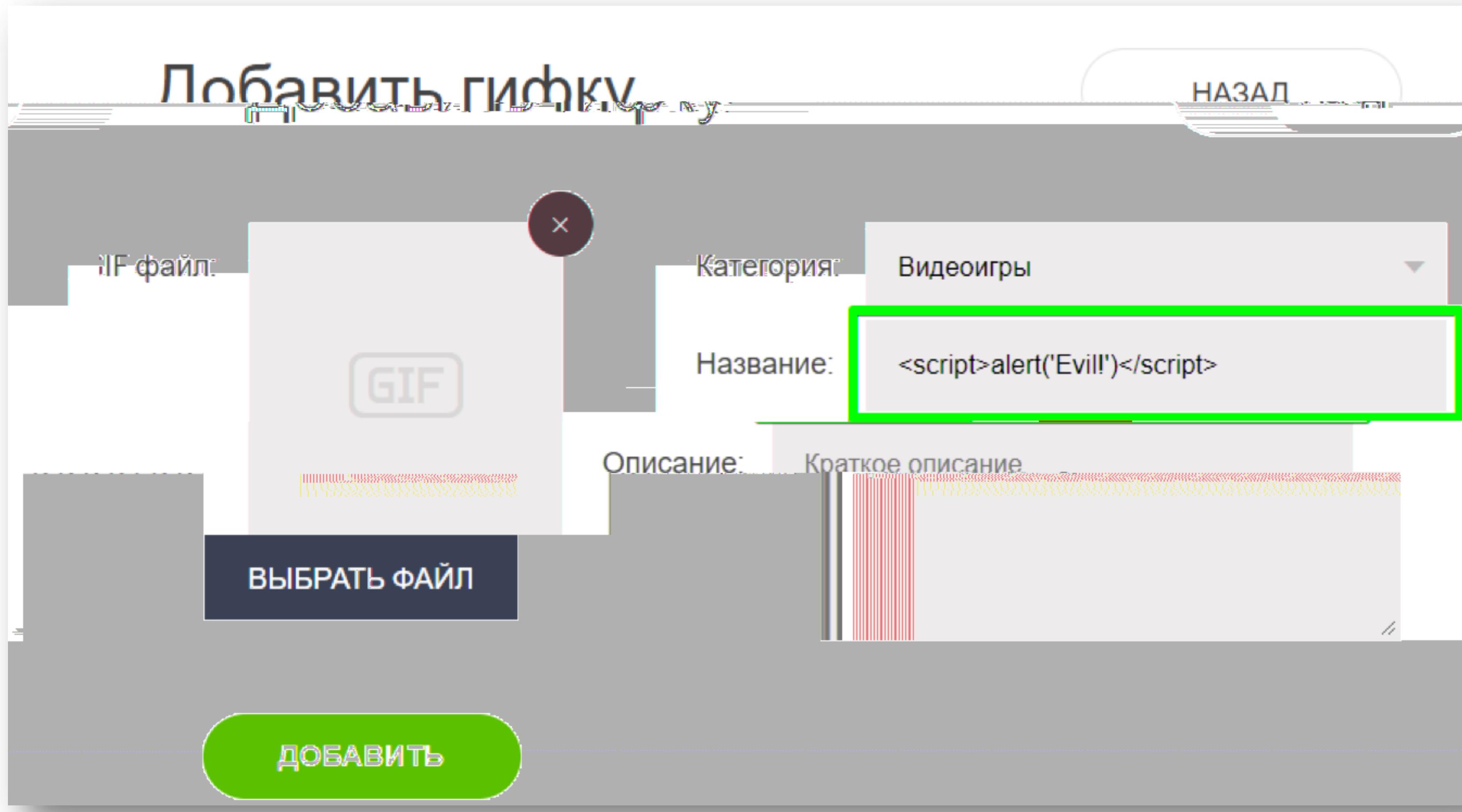
1. На сайте есть форма для публикации сообщения
2. Пользователь вместо простого текста отправляет HTML код и JS
3. Сообщение публикуется на сайте
4. Страница с просмотром сообщения «заражена»:
там выполняется JS код злоумышленника



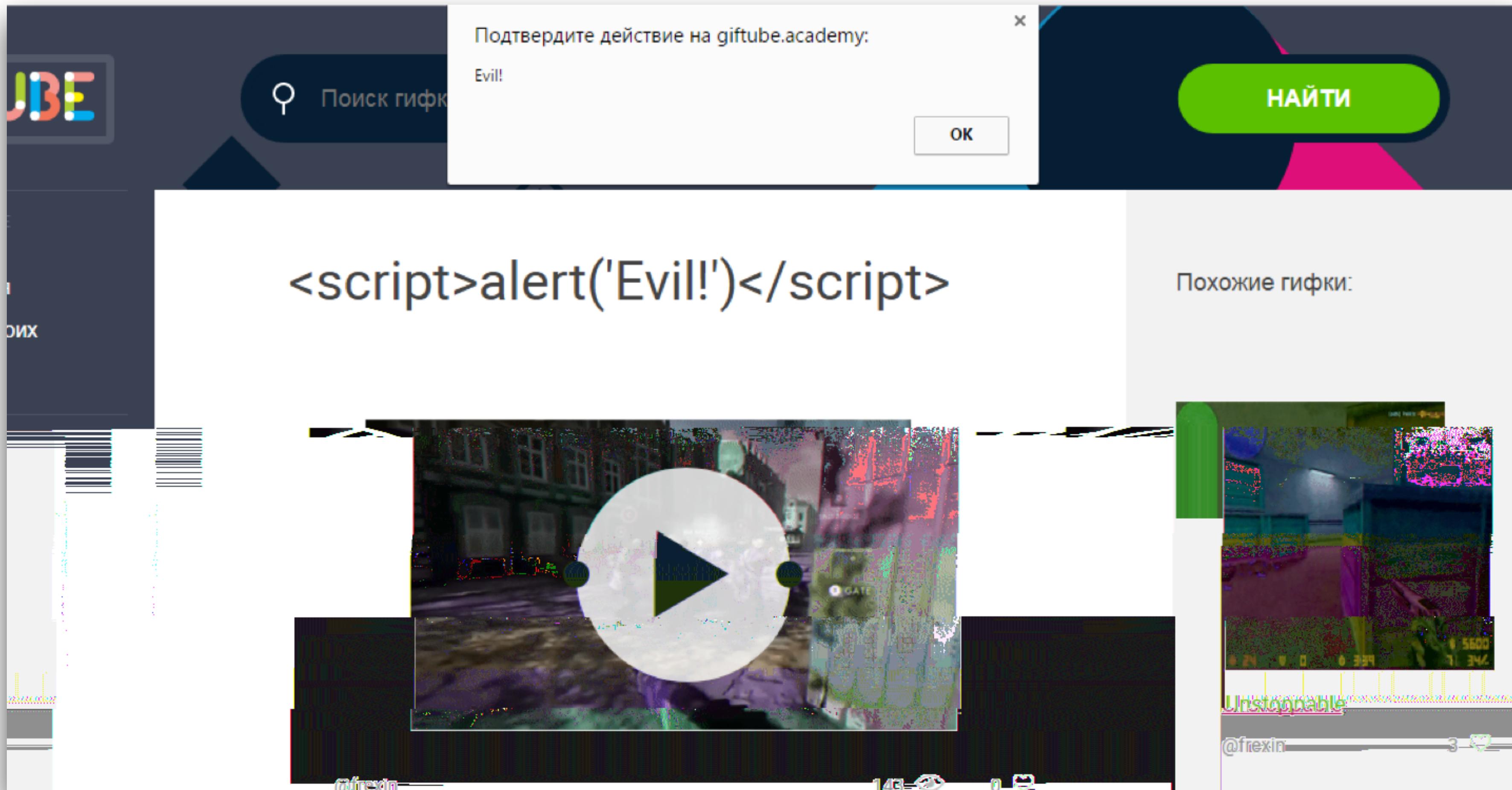
Как работает XSS



Пример атаки



Пример атаки



Что можно натворить



Что можно натворить

- испортить всю верстку страницы



Что можно натворить

- испортить всю верстку страницы
- перенаправить пользователя на свою страницу



Что можно натворить

- испортить всю верстку страницы
- перенаправить пользователя на свою страницу
- украсть куки



Что можно натворить

- испортить всю верстку страницы
- перенаправить пользователя на свою страницу
- украсть куки
- устроить DDoS-атаку



Что можно натворить

- испортить всю верстку страницы
- перенаправить пользователя на свою страницу
- украсть куки
- устроить DDoS-атаку
- множество других неприятных вещей



HTML-мнемоники

Мнемоника – это кодовое представление символа в HTML, который начинается со знака амперсанда "&" и завершается точкой с запятой ":".

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
\$	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®



Как обезопасить шаблон

Удалить из данных любые HTML теги, либо заменить их на HTML-мнемоники



Как обезопасить шаблон

Удалить из данных любые HTML теги, либо заменить их на HTML-мнемоники

Было

<script>



Как обезопасить шаблон

Удалить из данных любые HTML теги, либо заменить их на HTML-мнемоники

Было

```
<script>
```

Стало

```
&l t; scri pt&gt;
```



Как обезопасить данные

```
<script>alert('Evil!')</script>
```

`htmlspecialchars()`

`strip_tags()`

```
&lt;script&ampgtalert('Ev  
il!')&lt;/script&ampgt
```

```
alert('Evil!')
```

Па аа
HTML

В а а а



Пример в шаблоне

```
<li class="gf_gf-list-item">
  <div class="gf-picture">
    <a href="/gf/view?id=<?=$gf['id']; ?>" class="gf-preview">
      
    </a>
  </div>
  <div class="gf-description">
    <h3 class="gf-description-title">
      <a href="/gf/view?id=<?=$gf['id']; ?>">
        <?=htmlspecialchars($gf['title']); ?>
      </a>
    </h3>
    <div class="gf-description-data">
      <span class="gf-username">
        @<?=htmlspecialchars($gf['authorName']); ?>
      </span>
      <span class="gf-likes"><?=$gf['like_count']; ?></span>
    </div>
  </div>
</li>
```

