

# 7. Управляющие конструкции

## ***Условные директивы***

---

Шаблонизатор Blade предоставляет удобные сокращения для распространенных **управляющих конструкций PHP**, таких как **условные операторы** и **циклы**.

Эти сокращения обеспечивают очень чистый и краткий способ работы с управляющими конструкциями PHP и при этом остаются очень похожими на свои PHP-прообразы.

Конструировать оператор if можно при помощи директив **@if**, **@elseif**, **@else** и **@endif**. Эти директивы работают идентично своим PHP-прообразам:

```
@if (count($teams) == 1)

    // массив имеет одну запись

}elseif (count($teams) > 1)

    // массив имеет много записей

else

    // массив не имеет записей

@endif
```

В условиях, так же как и в чистом PHP, можно использовать более сложные конструкции.

- **count ()** - подсчитывает количество элементов массива.
- **isset ()** - позволяет определить, инициализирована ли переменная или нет. Если переменная определена, то `isset()` возвращает значение `true`.
- **empty ()** - проверяет переменную на "пустоту". "Пустая" переменная - это переменная, значение которой равно **null**, **0**,

**false** или **пустой строке** `""` - в этом случае функция `empty()` возвращает `true`.

## ***Выполнение кода в представлении***

---

В некоторых ситуациях полезно встроить PHP-код в сами представления. Для этих целей можно использовать директиву Blade **@php**.

Директива позволяет выполнить блок простого PHP кода внутри шаблона:

```
@php
    // ...
@endphp
```

что точно так же, как:

```
<?php
    // ...
?>
```

Файл **ex\_1/MainController.php**. Формирование данных контроллером

```
// обработчик маршрута http://host/music/team/{параметр}
public function acTeam ($id) {
    // формируем массив
    $team["name"] = "Deep Purple";
    $team["album"] = "Deep Purple in Rock";
    $team["composition"] = ["Иэн Гиллан", "Ричи Блэкмор",
        "Роджер Гловер", "Джон Лорд Иэн Пэйс"];
    // передаем массив в представление
    return view("page")->with(["team" => $team]);
}
```

Файл **ex\_1/page.blade.php**. Визуализация страницы

```
<body>

    <h1>Группа: {{ $team["name"] }}</h1>

    <h2>Альбом: {{ $team["album"] }}</h2>

    <h2>Состав:</h2>

    @php
        $composition = $team['composition']

    @endphp

    @if(count($composition))

        <ul>

            <li>{{ $composition[0] }}</li>

            <li>{{ $composition[1] }}</li>

            <li>{{ $composition[2] }}</li>

            <li>{{ $composition[3] }}</li>

        </ul>

    @else

        <p>Нет информации о составе</p>

    @endif

</body>
```

## ***Работа с циклами***

---

Разберем директиву **@foreach** шаблонизатора Blade, представляющую собой **цикл для перебора массива**.

Работа данной директивы идентична работе директивы `foreach` чистого PHP:

```
@foreach ($albums as $value)

    <p>Альбом: {{ $value }}</p>

@endforeach
```

Файл **ex\_2/MainController.php**. Формирование данных контроллером

```
// обработчик маршрута http://host/music/team/{параметр}

public function acTeam ($id) {

    // формируем массив

    $team["name"] = "Deep Purple";

    $team["album"] = "Deep Purple in Rock";

    $team["composition"] = ["Иэн Гиллан", "Ричи Блэкмор",
    "Роджер Гловер", "Джон Лорд Иэн Пэйс"];

    // передаем массив в представление

    return view("page")->with(["team" => $team]);

}
```

Файл **ex\_2/page.blade.php**. Визуализация страницы

```
<body>

    <h1>Группа: {{ $team["name"] }}</h1>

    <h2>Альбом: {{ $team["album"] }}</h2>

    <h2>Состав:</h2>

    <ul>

        @foreach($team["composition"] as $val)

            <li>{{ $val }}</li>

        @endforeach

    </ul>

</body>
```

## **Файлы раздаточного материала**

---

**ex\_1/** - условные директивы.

**ex\_2/** - вывод массива с использованием цикла **foreach**.

**ex\_3/** - проверка массива на пустоту.

**ex\_4/** - проверка объекта на существование.