

# Laravel Blade Cheat Sheet

---

Blade is a powerful templating engine provided with Laravel for working with PHP and HTML. It provides various directives to make your views more expressive and concise.

## Table of Contents

- [Laravel Blade Cheat Sheet](#)
  - [Table of Contents](#)
  - [Extending Layouts](#)
    - [@extends](#)
    - [@section](#)
    - [@yield](#)
    - [@parent](#)
  - [Including Sub-Views](#)
    - [@include](#)
  - [Control Structures](#)
    - [@if](#)
    - [@empty](#)
  - [Looping Structures](#)
    - [@foreach](#)
    - [@for](#)
    - [@forelse](#)
    - [@while](#)
  - [Outputting Data](#)
    - [The `{{ }}` Directive](#)
    - [The `{{ !! !! }}` Directive](#)
  - [Comments](#)
  - [Authentication Directives](#)
    - [@auth](#)
    - [@guest](#)
  - [Authorization Directives](#)
    - [@can](#)
    - [@cannot](#)
    - [@canany](#)
  - [Environment Directives](#)
    - [@env](#)
    - [@production](#)

## Extending Layouts

### @extends

The `@extends` directive is used to inherit a layout.

```
@extends('layout.name')
```

### @section

The `@section` directive defines a section that can be overwritten in child views.

```
@section('sectionName')  
    // content of the section  
@endsection
```

### @yield

The `@yield` directive is used to output the contents of a section.

```
@yield('sectionName')
```

### @parent

The `@parent` directive is used to display the content of the parent section.

```
@section('sectionName')  
    @parent  
    // additional content  
@endsection
```

## Including Sub-Views

### @include

The `@include` directive is used to include a sub-view.

```
@include('view.name')
```

## Control Structures

### @if

The `@if` directive is used for conditional statements.

```
@if($condition)
    // code to execute if condition is true
@elseif($anotherCondition)
    // code to execute if another condition is true
@else
    // code to execute if none of the conditions are true
@endif
```

### @empty

The `@empty` directive checks if a variable is empty.

```
@empty($variable)
    // code to execute if the variable is empty
@endempty
```

## Looping Structures

### @foreach

The `@foreach` directive is used to iterate over arrays or objects.

```
@foreach($items as $item)
    // code to execute for each item
@endforeach
```

### @for

The `@for` directive is used for simple loop iterations.

```
@for($i = 0; $i < 5; $i++)
    // code to execute for each iteration
@endfor
```

### @forelse

The `@forelse` directive is used to display a list of items with an optional message if the list is empty.

```
@forelse($items as $item)
    // code to execute for each item
@empty
    // code to execute if the list is empty
@endforelse
```

## @while

The `@while` directive is used for indefinite loop iterations.

```
@while($condition)
    // code to execute while the condition is true
@endwhile
```

## Outputting Data

### The `{{ }}` Directive

The `{{ }}` directive is used to echo the contents of a variable.

```
{{ $variable }}
```

### The `{!! !!}` Directive

The `{!! !!}` directive outputs the contents without escaping HTML entities.

```
{!! $htmlContent !!}
```

## Comments

The `{{-- --}}` directive is used to add comments in Blade templates.

```
{{-- This is a blade comment --}}
```

## Authentication Directives

### @auth

The `@auth` directive checks if the user is authenticated.

```
@auth
  // code to execute if the user is authenticated
@endauth
```

### @guest

The `@guest` directive checks if the user is a guest.

```
@guest
  // code to execute if the user is a guest
@endguest
```

## Authorization Directives

### @can

The `@can` directive checks if the user has a certain permission.

```
@can('permissionName')
  // code to execute if the user has the permission
@endcan
```

### @cannot

The `@cannot` directive checks if the user does not have a certain permission.

```
@cannot('permissionName')
  // code to execute if the user does not have the permission
@endcannot
```

## @canany

The `@canany` directive checks if the user has any of the given permissions.

```
@canany(['permission1', 'permission2', ... , 'permissionN'])
    // code to execute if the user has any of the given permissions
@endcanany
```

## Environment Directives

### @env

The `@env` directive checks the application environment.

```
@env('local')
    // code to execute if the application is in the local environment
@endenv
```

### @production

The `@production` directive checks if the application is in production environment.

```
@production
    // code to execute if the application is in production environment
@endproduction
```

These are the commonly used Blade directives in Laravel for templating and controlling the presentation logic of your views.