



Basic Commands & Concepts

Artisan Commands

<code>php artisan --version</code>	Display the Laravel version.
<code>php artisan make:controller ControllerName</code>	Create a new controller.
<code>php artisan make:model ModelName</code>	Create a new Eloquent model.
<code>php artisan make:migration create_table_name_table</code>	Create a new migration file.
<code>php artisan migrate</code>	Run pending migrations.
<code>php artisan migrate:rollback</code>	Rollback the last migration.
<code>php artisan serve</code>	Start the built-in PHP development server.
<code>php artisan tinker</code>	Enter the interactive Tinker shell.
<code>php artisan route:list</code>	Display all registered routes.

Directory Structure

<code>app/</code>	- Contains the core code of your application.
<code>bootstrap/cache/</code>	- Framework bootstrap files.
<code>config/</code>	- Application configuration files.
<code>database/</code>	- Database migrations and seeds.
<code>public/</code>	- Publicly accessible files (CSS, JavaScript, images).
<code>resources/views/</code>	- Application views (Blade templates).
<code>routes/</code>	- Route definition files (web.php, api.php).
<code>storage/</code>	- Storage directory for files and sessions.

Routing and Controllers

Basic Routing

```
Route::get('/route', function () {
    return 'Hello, World!';
});

Route::post('/route', function () {
    // Handle POST request
});
```

Route Parameters

<code>Route::get('/user/{id}', function (\$id) {</code>	Required parameter.
<code> return 'User ID: ' .</code>	
<code> \$id;</code>	

<code>Route::get('/user/{name?}', function (\$name = null) {</code>	Optional parameter.
<code> return 'Name: ' .</code>	
<code> \$name;</code>	

Controllers

```
Route::get('/users',
    'App\Http\Controllers\UserController@index');

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class UserController extends Controller
{
    public function index()
    {
        return view('users.index');
    }
}
```

Eloquent ORM

Basic Model Operations

<pre>\$users = App\Models\User::all();</pre>	Get all records.
<pre>\$user = App\Models\User::find(1);</pre>	Find a record by primary key.
<pre>\$user = new App\Models\User; \$user->name = 'John Doe'; \$user->email = 'john@example.com'; \$user->password = bcrypt('secret'); \$user->save();</pre>	Create a new record.
<pre>\$user = App\Models\User::find(1); \$user->name = 'Jane Doe'; \$user->save();</pre>	Update an existing record.
<pre>\$user = App\Models\User::find(1); \$user->delete();</pre>	Delete a record.

Relationships

<pre>One To One</pre> : A user has one profile.
<pre>public function profile() { return \$this->hasOne('App\Models\Profile');</pre>
<pre>One To Many</pre> : A user has many posts.
<pre>public function posts() { return \$this->hasMany('App\Models\Post');</pre>
<pre>Many To Many</pre> : A post has many tags.
<pre>public function tags() { return \$this->belongsToMany('App\Models\Tag');</pre>

Blade Templating

Basic Syntax

<pre>{{ \$variable }}</pre>	Display a variable (automatically escaped).
<pre>{!! \$variable !!}</pre>	Display a variable without escaping.
<pre>@if (condition) @endif</pre>	Conditional statement.
<pre>@foreach (\$items as \$item) @endforeach</pre>	Looping.

Components and Layouts

<pre>@extends('layouts.app')</pre> - Extend a layout.
<pre>@section('content') ... @endsection</pre> - Define a section.
<pre>@include('partials.header')</pre> - Include a partial view.
<pre>@component('components.alert') ... @endcomponent</pre> - Render a component.

Directives

<pre>@auth ... @endauth</pre>	Check if the user is authenticated.
<pre>@guest ... @endguest</pre>	Check if the user is a guest.
<pre>@csrf</pre>	Generate a CSRF token field.