Drupal 8 Theming with Twig

Drupal 8 is an open source content management system and powerful framework that helps deliver great websites to individuals and organizations, including non-profit, commercial, and governments around the globe.

Starting from the bottom up, we will install, set up, and configure Drupal 8. We’ll navigate through the admin interface so you can learn how to work with core themes and create new custom block layouts. We’ll also walk through a real-world project to create a Twig theme from concept to completion while adopting best practices to implement CSS frameworks and JavaScript libraries. We will see just how quick and easy it is to create beautiful, responsive Drupal 8 websites while avoiding the common mistakes that many frontend developers make.

Who this book is written for

This book is intended for frontend developers, designers, and anyone who is generally interested in learning all the new features of Drupal 8 theming. Discover what has changed from Drupal 7 to Drupal 8 and immerse yourself in the new Twig PHP templating engine. Prior experience with setting up and configuring a standalone development environment is required as we will be working with PHP and MySQL.

What you will learn from this book

- Navigate the Drupal 8 admin interface
- Build custom block layouts with reusable and foldable blocks
- Create subthemes based on the Bartik and Classy base themes
- Construct a responsive theme with Twitter Bootstrap
- Work with the new Twig PHP templating engine
- Configure Drupal for Twig debugging
- Enable pre-processing of Twig variables
- Develop a theme from scratch following a step-by-step project outline

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Setting Up Our Development Environment'
- A synopsis of the book’s content
- More information on **Drupal 8 Theming with Twig**
About the Author

Ever since Chaz can remember, he has been picking up a crayon, a pencil, a pen, or a computer, whether to draw a picture, write a poem, share an article, or develop the next great website. Looking back at these qualities, it's the reason why he chose to use those passions to give back to the open source community. His journey has opened doors to great experiences as a consultant, instructor, trainer, author, web developer, technical architect, Drupalist, and most importantly, a father. However, he could be none of these without first being a student and learning from life, love, and a passion for education, community development, and of course, science and technology.

The sum of these things has allowed Chaz to share his experiences as an active member of the Drupal community, having participated in developing Lynda.com Drupal training videos, authoring multiple Drupal books for Packt Publishing, and giving back his knowledge by contributing numerous articles and presentations at NYCCamp, BadCamp, and DrupalCon, and bringing top-notch Drupal expertise to his clients' work.

However, his greatest journey is still in front of him as he continues to learn new creative ways to architect open source websites, write about the next best things, and laugh with his colleagues along the way.
Preface

Starting from the bottom, *Drupal 8 Theming with Twig* will walk you through setting up and configuring a new Drupal 8 website. Navigate across the admin interface, learn how to work with core themes, and create new custom block layouts. Take a real-world project and create a Twig theme that adopts best practices to implement CSS frameworks and JavaScript libraries. See just how quick and easy it is to create beautiful and responsive Drupal 8 websites along with avoiding some of the common mistakes many frontend developers run into.

If you consider yourself a frontend developer, you will be right at home. However, since no PHP knowledge will be necessary, anyone who can create HTML websites with basic HTML5 and CSS3 skills will learn how to create a great Drupal 8 theme.

**What this book covers**

*Chapter 1, Setting up Our Development Environment*, will begin with walking you through setting up a development workflow. You will learn how to use an AMP (Apache, MySQL, PHP) stack to configure a local web server. We will introduce the process of installing Drupal 8 and walk through the admin interface in preparation for working with the new Twig templating engine.

*Chapter 2, Theme Administration*, provides a glance at the Appearance page, where you will learn how to install, uninstall, and configure various themes and settings. We will take a look at the new Block layout system as we explore how to add fields to blocks, reuse blocks, and assign chunks of content to various regions. Learn how Drupal 8 has reconfigured files and folders, where themes are now placed, and how to find core themes.
**Preface**

*Chapter 3, Dissecting a Theme,* begins with discussing the importance of a proper development environment and the steps involved to ensure that you're ready for Drupal 8. This will help you learn the differences and similarities between core and custom themes and how configuration has changed. Break down the metadata that makes up a theme, libraries, and regions. You will learn the role of templates and how the theme layer interacts with the Twig templating engine.

*Chapter 4, Getting Started – Creating Themes,* starts with creating a starter theme that allows us to work with assets while learning common techniques to integrate various CSS frameworks. You can learn how to rethink layout strategies as we dive into the theme layer and work with Twig. Then, we will wrap up with creating a subtheme that extends the new Classy base theme.

*Chapter 5, Prepping Our Project,* covers reviewing a real-world project that we will be building and how to break down how design and functionality should come together in Drupal 8. We will create the new theme structure, define metadata, add regions, and implement several CSS and JavaScript libraries.

*Chapter 6, Theming Our Homepage,* begins with working with the site branding block to add a logo. We will create a basic HTML wrapper and homepage template using Twig as we convert our homepage mockup into a fully functioning Drupal 8 front page. You will learn how to convert static markup into Twig variables as we theme global components, such as the Search block, menus, and custom blocks. Use the new libraries.yml file to work with assets, such as Twitter Bootstrap and Flex Slider and then attach them to specific templates using the new `{{ attach_library() }}` function.

*Chapter 7, Theming Our Interior Page,* will show you how to review mockups and identify what regions, blocks, content types, and views will need to be developed to recreate functionality. You will learn how to reuse Drupal 8 regions, work with the new Page title block, and how Twig plays an important part in rewriting Views output.

*Chapter 8, Theming Our Blog Listing Page,* starts with the best practice approaches to managing content types. From adding new display modes to enabling and formatting fields, learn how to use content display modes with Views to display data in lieu of using fields directly. You can work with Node templates to add CSS classes directly to our markup, work with content variables, and learn how to suppress fields. You will use Twig filters to format dates and manage individual field templates while creating a listing page.
Chapter 9, *Theing Our Blog Detail Page*, teaches us how to reuse Twig templates and display different content simply by using file name suggestions to target-specific display modes. You will work with the new comment types to add commenting functionality to pages as fields, learn how to alter comment display using field templates to theme comment threads, and add social sharing functionality to pages using custom blocks and JavaScript libraries that we can attach directly to a block.

Chapter 10, *Theing Our Contact Page*, introduces contact forms in Drupal 8 that we can use to create fieldable forms that users can interact with. You will learn how to integrate Google Maps into custom blocks that take advantage of library assets and vendor JavaScript.

Chapter 11, *Theing Our Search Results*, covers core search functionality, how to index database content, and work with search results templates. Core search can sometimes be limited, so we will take a more advanced look at the Search API module to create a better search experience. You will learn how to add individual fields to search and use display modes to output content while creating a Search view that allows us to expose filters to the users to find exactly what they are looking for.

Chapter 12, *Tips, Tricks, and Where to Go from Here*, introduces how to theme common admin sections of Drupal 8. You will learn how to modify the markup for local tasks and status messages. Reuse Twig templates using extends to share layouts while working with pages and blocks. Finally, we will leave you with some great contributed modules to take a look at and introduce you to the Drupal community.
Setting Up Our Development Environment

Regardless of you being a seasoned web developer or someone who is just about to start learning Drupal, there are few things that everybody needs to have in place before we can get started:

• First, is to make sure that we have an Application stack that will meet Drupal 8’s system requirements. MAMP provides us with a standalone web server that is generally referred to as an AMP (Apache, MySQL, PHP) stack and is available for both OS X and Windows. We will look at installing and configuring this local web server in preparation to install Drupal 8.

• Second, is to set up a Drupal 8 instance and learn the process of installing Drupal instances into our AMP stack. There are a few changes on how the configuration and installation processes work in Drupal 8, so we will take a closer look to ensure that we all begin development from the same starting point.

• Third, we will be reviewing the Admin interface, including the new responsive Admin menu and any configuration changes that have been made as we navigate to familiar sections of our site. We will also look at how to extend our website using contributed modules, review changes to the files and folder structure that make up Drupal 8, and discuss best practices to manage your files.

• Finally, we will review the exercise files that we will be using throughout the series, including how to download and extract files, how to use phpMyAdmin—a database administration tool to back up and restore database snapshots—and how to inspect elements within our HTML structure using Google Chrome.
Let's get started by installing our web environment that we will be using as we take an exciting look at Drupal 8 theming with Twig.

### Installing an AMP (Apache, MySQL, PHP) stack

To install and run Drupal 8, our server environment must meet and pass certain requirements. These requirements include a web server (Apache, NGINX, or Microsoft IIS) that can process server-side languages such as PHP, which Drupal 8 is built on.

Our server should also contain a database that can manage the data and content that Drupal 8's content management system will store and process. The preferred database is MySQL. However, Drupal 8 can also support PostgreSQL along with Microsoft SQL Server and Oracle with an additional module support.

Finally, Drupal 8 requires PHP 5.5.9 or later, with the CURL extension.

However, because this book is not meant to be a "How-to" on installing and configuring Apache, MySQL, or PHP, we will take all the guesswork and trial by fire out of the equation and instead turn to MAMP.

### Introducing MAMP

MAMP can be found at https://www.mamp.info/en and is a tool that allows us to create Drupal sites locally without the need or knowledge of installing and configuring Apache, MySQL, or PHP on a specific platform.

The application stack will consist of the following:

- Apache: The world's most popular web server
- MySQL: The world's most popular database server
- PHP: Various versions of PHP
- phpMyAdmin: A tool to manage MySQL databases via your browser
Chapter 1

Downloading MAMP

Let's begin with the steps involved in quickly downloading, installing, and configuring our very own AMP stack along with an initial instance of Drupal that we will be using throughout the rest of this book. Begin by opening up our web browser and navigating to https://www.mamp.info/en/downloads and selecting either Mac OS X or Windows and then clicking on the Download button, as shown in the following image:

MAMP will allow us to install a local web server on either Mac or Windows and provides us with all the tools we will need to develop most open source websites and applications including Drupal 8.

Installing MAMP

Once the download has completed, we will need to locate the .dmg (Mac users) or .exe (Windows users) installation file and double-click on it to begin the installation process. Once the executable is opened, we will be presented with a splash screen that will guide us through the process of installing and configuring MAMP.

Clicking on the Continue button located on the Introduction pane, will take us to the Read Me information. MAMP will notify us that two folders will be created: one for MAMP and the other for MAMP PRO. It is important to not move or rename these two folders.
Click on the **Continue** button, which will take us to the license information. Simply accept the terms of the license agreement by clicking on **Continue** and then on **Agree** when prompted.

We can finally click on the **Install** button to complete the installation process. Depending on the operating system, we may need to enter our credentials for MAMP to be able to continue and configure our local web server. Once the install has completed, we can click on the **Close** button.

**A quick tour of MAMP PRO**

Let's begin by opening up **MAMP** and taking a quick tour of the various settings and how we can go about using our local web server to install and configure a new Drupal 8 instance.

When we first open up MAMP, we will be prompted to launch either MAMP or MAMP PRO, as shown in the following image:

![Warning](image)

While MAMP is the free version of the local web server, it is strongly recommended that we use MAMP PRO for configuration and easy setup of a website. We can continue by clicking on **Launch MAMP PRO**, which will prompt us one more time to accept the **Initialization** of the remaining components that MAMP PRO needs before we can begin using it. Now, click on **OK** to continue.
We can use MAMP PRO free for 14 days and at the end of that period, we can decide whether to purchase a license or continue using the free version. Click on OK to continue.

**General settings for MAMP PRO**

If this is the first time you're using MAMP PRO, then there is some quick housekeeping we will want to take care of, beginning with the general settings. MAMP PRO tries to make sure that it does not interfere with any other possible web servers we may be running by setting the default ports of Apache to 8888 and MySQL to 8889. Although this is nice, the recommendation is to click on the Set ports to 80, 443 & 3306 button that will make sure that MAMP PRO is running on more standardized ports for web development.

If we want to make sure that Apache or MySQL are active at all time, we will also check Start Apache and MySQL on system startup and uncheck Stop Apache and MySQL when quitting MAMP PRO. Once we have made these changes, we can click on the Save button. Our changes should now be applied as shown in the following image and MAMP PRO will now prompt us to Start servers now.
Setting Up Our Development Environment

Host settings

The next tab we will look at is the Hosts tab, which is where we will create and configure basic websites. By default, MAMP PRO creates a localhost entry for us, which is common when developing a web application.

We will be using the Hosts tab to create an additional website when we install Drupal 8, so let's take a moment to locate some of the common settings we will need to know. Take a look at the following image; we can see that localhost is the Server name of our default website, uses the default PHP version of 5.6.10, and has a Document root pointing to the htdocs folder of our MAMP installation.

Another nice ability of MAMP PRO is to be able to click on the arrow icon located to the right of the Server name and have our default web browser open up to the localhost page, as shown in the following image:
It is important to point out that the **Server name** always equates to the name of the URL in our browser that displays our website.

MAMP PRO is quite a robust and powerful local web server and while there are many more configuration options and settings that we could spend time looking through, most of our time will be spent on working from the **Hosts** tab creating new websites or configure existing sites.

So far, MAMP PRO has configured everything for us, but how to create a new website and, in more general, install a Drupal 8 instance? Let’s look at it in the following section.

## Installing Drupal 8

In order to install Drupal 8 within our local MAMP PRO server, we will need to perform a series of steps:

1. We will need to grab a copy of the latest Drupal 8 release and extract the files to a location on our computer that will be the document root of our website.
2. We will have to create a new host entry with the server name that we will want to use for our URL and point our host entry to the proper document root containing our Drupal 8 instance.
3. We will have to create a MySQL database that we can point Drupal to during the installation process.

We will walk through each of these steps in detail to ensure that we all have a copy of Drupal 8 installed properly that we will build upon as we work through each lesson.
Setting Up Our Development Environment

Downloading Drupal 8

Drupal.org is the authority on everything about Drupal. We will often find ourselves navigating to Drupal.org to learn more about the community, look for documentation, post questions within the support forum, or review contributed modules or themes that can help us extend Drupal's functionality. Drupal.org is also the place where we can locate and download the latest release of Drupal 8.

We can begin by navigating to https://www.drupal.org/node/2627402 and locate the latest release of Drupal 8. Click on the compressed version of Drupal 8 that we prefer, which will begin downloading the files to our computer. Once we have a copy of Drupal 8 on our computer, we will want to extract the contents to a location where we can easily work with Drupal and its folders and files.

Creating our document root

A document root is the main folder that our host entry will point to. In the case of Drupal, this will be the extracted root folder of Drupal itself. Generally, it is a best practice to maintain some sort of folder structure that is easy to manage and that can contain multiple websites.

For the sake of demonstration, we will create a Sites folder and then copy the compressed Drupal files to our new folder and extract the contents, as shown in the following image:

```
- Sites
  - drupal-8.0.1
    - core
    - modules
    - profiles
    - sites
    - themes
    - vendor
    - autoload.php
    - composer.json
    - composer.lock
    - example.gitignore
    - index.php
    - LICENSE.txt
    - README.txt
    - robots.txt
    - update.php
    - web.config
```

Creating our host entry

A host entry represents our website, which, in this case, is our Drupal 8 instance. Hosts always contain a server name that equates to the URL we will use to navigate to Drupal within our browser.
Begin by opening MAMP PRO and clicking on the **Hosts** tab. To add a new host entry, we can click on the plus icon at the bottom of the **Server Name** column, as shown in the following image:

![MAMP PRO Hosts Tab](image)

By default, this will add a new host entry that will require us to configure with three very important pieces of information:

- We will have to change the **Server name** from the default to `drupal8`.
- Then, we will want to verify that the required version of PHP is being used; in our case, the default of **5.6.10** will work just fine.
- Finally, we will need to click on the folder icon within the **Document root** section and choose our Drupal 8 folder that we placed within our sites folder earlier from the **Please select a Document Root folder** dialog.
Setting Up Our Development Environment

The **General** settings for our new host entry should look as shown in the following image:

![General settings for host entry](image)

We can now apply our changes by clicking on the **Save** button and then clicking on the **Yes** button when prompted to have MAMP PRO restart the servers.

**Creating a new database for Drupal**

Drupal 8 requires a database available to install any tables that make up the content management system. These tables will hold configuration data, users and permissions, content, and any extendable functionality that makes Drupal 8 so powerful.
Lucky for us, MAMP PRO installs a MySQL database server that we can take advantage of to create a new database that Drupal 8 can point to. This same database server we will also be working with to back up and restore our database content as we progress through each lesson.

**Using phpMyAdmin**

MAMP PRO also installs a free software tool written in PHP for the sole purpose of managing MySQL databases. phpMyAdmin allows us to perform a multitude of tasks from browsing tables, views, fields, and indexes to exporting and importing database backups and much more.

If we switch back over to MAMP PRO, we can locate the MySQL tab and click on the **phpMyAdmin** link located in the **Administer MySQL with** region, as shown in the following image:
Setting Up Our Development Environment

We should now be presented with **phpMyAdmin** within our browser. Currently, we are interested in creating a new database. We will revisit phpMyAdmin a little later to learn how to back up and restore our database. The following are the steps to create a database:

1. Begin by clicking on the **New** link in the left sidebar, as shown in the following image:

   ![Image of phpMyAdmin interface showing the 'New' link and creating a new database](image1)

2. Next, we will want to enter a name of **drupal8** within the **Create database** field, as shown in the following image, and then click on the **Create** button.

   ![Image of phpMyAdmin interface showing databases with 'drupal8' being created](image2)
We have now created our first MySQL database, which we will use when configuring Drupal 8 in the next step.

**Completing Drupal 8 installation**

Now that we have all of our basic requirements completed, we can open up our favorite web browser and navigate to http://drupal8/core/install.php to begin the installation process.

Since this may be the first time installing Drupal 8, one thing we will notice is that the install screen looks a little different. The install screen has been given a makeover, but the steps are similar to that of Drupal 7, starting with choosing a language.

**Choosing a language**

The installation process will prompt us to choose a language that we want Drupal 8 to be installed in. This language will control how the Admin area appears, and in many cases, the default of English is acceptable. We will need to click on the **Save and continue** button to proceed to the next step, as shown in the following image:

![Choose language](image_url)
Choosing a profile
Our next step is to choose an installation profile. We can think of this as Drupal's way of preconfiguring items for us that will make our job easier when developing. By default, we can leave the Standard profile selected, but if we choose to configure Drupal ourselves, we can always choose Minimal. Click on the Save and continue button to proceed to the next step, as shown in the following image:

Verifying requirements
The next screen allows us to review any requirements that Drupal needs or recommends for optimal performance. From here, we can see web server information, PHP version, memory limits, and more:
The requirements review can also alert us to any configuration settings that will allow Drupal to perform better. In our example, we forgot to enable OPcode caching, which allows PHP to compile down to bytecode. Without going into the details of caching, we can easily enable this feature in MAMP PRO.
Begin by opening up the MAMP PRO console and clicking on the PHP tab. Next, we will want to select OPcache from the Cache module to speed up PHP execution dropdown, as shown in the following image:

Click on the Save button and then allow MAMP PRO to restart servers if prompted. Now, we can refresh our Drupal install in the browser, and we will be taken to the next step in the installation process.

Database configuration
Database configuration can sometimes be a tricky part of installing Drupal for the first time. This is generally due to selecting the incorrect database type, wrong database name, or password, or by not specifying the correct host or port number.
The settings we will want to use are as follows:

- **Database type**: Leave the default of MySQL selected.
- **Database name**: This is the name of the database that was created upon import. In our case, it should be `drupal8`.
- **Database username**: `root`.
- **Database password**: `root`.

With these settings, we can click on the **Save and continue** button to proceed.

If this is successful, we can see the **Installing Drupal** screen and watch as Drupal installs the various modules and configurations. This process may take a few minutes. If this process fails in any way, please go back and review the previous steps to make sure that they match what we have used.
Setting Up Our Development Environment

Configuring the site
Before we can wrap up our Drupal 8 installation, we need to configure our site by inputting various settings for site information, site maintenance account, regional settings, and update notifications. Let's proceed now by entering our Site Information.

The site information consists of the following:

- **Site name**: To be consistent, let's call our site `drupal8`. We may give our new site any name that we like. As usual, we can change the site name later from the Drupal admin.

- **Site e-mail address**: Enter an e-mail address that we will want to use for automated e-mails such as registration information. It is the best practice to use an e-mail that ends in our site's domain to prevent e-mails from being flagged as spam.
Next, we will want to set up the site maintenance account. This is the primary account used to manage Drupal to perform such tasks as updating the core instance, module updates, and any development that needs user 1 permissions.

Site maintenance account
The site maintenance account information consists of the following:

- **Username:** Because we are developing a demo site, it makes sense to generally use admin for the username. Feel free to choose whatever is easy to remember, but don't forget it.

- **Password:** Security sticklers will ask to create something strong and unique, and Drupal displays a visual interface to let us know how strong our password is. For the sake of demonstration, we will use admin as our password so that your username and password match and are easy to remember.

- **E-mail address:** Generally, using the same e-mail that is used for the site e-mail address makes for consistency but is not required as we can choose any e-mail that we don't mind receiving security and module update notices.

Regional settings
Regional settings consist of default country and default time zone. These are often neglected and left with their defaults. The defaults are not recommended, as they are important in the development and design of Drupal 8 websites, specifically, when it comes to dates and how they are used to capture data and display dates back to the end user.

For our specific installation, choose the country and time zone for our region.

Update notifications
At last, we have come to our final set of configurations. Update notifications should always be left-checked unless we have no reason to receive security updates to Drupal core or module updates. By default, they should be checked. Click on **Save and continue** to finalize the configuration and installation of Drupal 8.

Drupal installation is now complete, and we should see the home page of our new website. Say "hello" to Drupal 8.
Reviewing the new admin interface

Our local instance of Drupal 8 is similar to Drupal 7 when first viewing the site. You will note the default Bartik theme with the friendly Drupal drop logo and a tabbed main menu. It is here though where the similarities stop. Drupal 8 has been reworked from the ground up, including a brand new responsive layout and admin menu.

Exploring the admin menu

One of the nice new features of Drupal 8 is the rebuilt admin menu. Everything has been moved under the Manage menu item. The admin menu itself is responsive and will change from text and icon to icon only, as soon as the browser is resized to tablet screen size.

The admin menu can also be pinned to the left side of the window by clicking on the arrow icon to the right of the Help menu item.
The flexibility of the new admin menu enables the admin user to manage Drupal 8 websites from the browser or a tablet or a smartphone very easily.

**Previewing the interface**

Taking a closer look at the menu items contained in our admin menu, we begin to see some differences in how things are named and may wonder where to find once familiar settings and configurations. Let’s quickly walk through these menu items now so that it is easier to find things as we progress later on in future chapters:

- **Content**: This section displays any user-generated content, comments, and files with the ability to filter by *Published status, Type, Title, and Language*. The display for content is also now a view and can be customized with additional fields and filters as needed.
Setting Up Our Development Environment

- **Structure**: This section is to manage **Block layout**, **Comment types**, **Contact forms**, **Content types**, **Display modes**, **Menus**, **Taxonomy**, and **Views**. We will explore some of the changes and new functionality contained within this section later on in the book.

- **Appearance**: This section is to enable, disable, and configure default themes as well as administrative themes.
• **Extend:** Formerly known as Modules, this section is for listing, updating, and uninstalling core, custom, and contributed modules. New is the ability to search for modules using a filter. Various contributed modules have been moved into core, including Views and CKEditor.
Setting Up Our Development Environment

- **Configuration**: This section is designed to configure both core and contributed modules. Each area is grouped into functional sections and allows us to manage site information to file system to performance tuning.

- **People**: This section allows us to manage users, permissions, and roles. The display for users is now a View as well and can be customized to add additional fields and filters as needed.
• **Reports**: This section is designed to view available updates, recent log messages, field lists, status reports, top "access denied" errors, top "page not found" errors, top search phrases, and View plugins.
Setting Up Our Development Environment

- **Help**: This section is designed to obtain helpful information on functionality necessary to know in administering a Drupal 8 website. This includes a Getting Started section and help topics on items, such as Block, Views, User, and more.

---

### Exploring Drupal 8 folder structure

There are several changes to Drupal 8 with regard to how files and folders are structured. Let's walk through the core, modules, sites, and themes folders and discuss some best practices for how each of these folders should be managed when creating a Drupal 8 website.

#### The core folder

One of the first things to point out is that the files and folder structure of Drupal 8 have changed from its predecessor Drupal 7. The first change is that everything that Drupal 8 needs to run is contained within the new core folder. No longer is there any confusion of having the modules and themes folders contained within a sites folder and having to ask "did I place my files in the correct location?".
The **core** folder consists of miscellaneous files needed by Drupal to bootstrap the content management system as well as the following folders:

- **assets**: Various external libraries used by core (jquery, modernizr, backbone, and others)
- **config**: It contains misc configuration for installation and database schema
- **includes**: It contains files and folders related to the functionality of Drupal
Setting Up Our Development Environment

- **lib**: Drupal core classes
- **misc**: Various JavaScript files and images used by core
- **modules**: Drupal core modules and Twig templates
- **profiles**: Installation profiles
- **scripts**: Various CLI scripts
- **tests**: Drupal core tests
- **themes**: Drupal core themes

The modules folder
While a lot of functionality, which was generally contained in a contributed module, has been moved into the core instance of Drupal 8, you will still find yourself needing to extend Drupal. Previously, you would locate a contributed module, download it, and then extract its contents into your `sites/all/modules` folder so that Drupal could then use it.

Contributed and custom modules are now placed into the `modules` folder, which is no longer contained inside your `sites` folder.
Best practices are to create a few subdirectories inside the `modules` folder for contributed modules—the modules built by third parties that we will use to extend your project, such as `contrib`, and `custom`, for the modules that we create on a per project basis. We will also occasionally find ourselves with a `features` folder if we plan to use the Features module to break out functionality that needs to be managed in code for purposes of migrating it easily to development, staging, and production instances of our website.

**The sites folder**

We are all familiar with the `sites` folder in Drupal 7. However, in Drupal 8, the `sites` folder only contains our Drupal instance configuration and files.
The themes folder

Finally, we have the themes folder. So, why don't we see the default themes that Drupal generally ships with inside this folder? That is because Drupal's default themes now are contained within the core folder. Finally, we will actually use the themes folder to place our custom or contributed themes inside it for use by Drupal.

We will be exploring the themes folder in more detail in later chapters as we begin creating custom themes.
Using the project files
As we work through each chapter of the book, we will be using exercise files that contain examples of how each page will be themed is laid out. This will include database snapshots, HTML, CSS, and images for our Home, About, Portfolio, Blog, and Contact pages. Before we begin using these files, we need to know where to download them from and the best location to extract them to for future use.

Downloading and extracting the exercise files
We can find the exercise files at https://www.packtpub.com/support. Click on the download link and save the compressed file to the desktop. Once the download is finished, we will need to extract the contents. Let's take a quick look at what we have:

- Several chapter folders containing files that we will use for whatever task we are working on in that chapter
- A Mockup folder that contains the finished HTML version of our theme
- A database snapshot contained within various chapter folders that we will use to restore on top of our current Drupal instance to ensure that we always have the same configuration at specific points along the way

Since we will be working with database snapshots at various points, we will want to look at how we can manage these files using the MySQL database tool named phpMyAdmin.

Database backup
It is important to know how to backup our database when working in Drupal 8 as most of our content and configurations are contained within a database. Make sure that phpMyAdmin is open in the browser.
Next, we will want to make sure that any database exports are saved as a file versus just plain SQL script. Because this is a global setting, we will need to make sure that we have not selected any specific database. We can make sure that we are affecting global settings by clicking on the house icon in the left-hand sidebar underneath the phpMyAdmin logo. Next, we can navigate to Settings and then Export, as shown in the following image:

Next, click on the Save as file option and the Apply button.

**Export settings**

One little gotcha when using phpMyAdmin is making sure that when we create our database export, we ensure that the SQL also drops any tables before trying to recreate them when we do the import later.

We can create our database export by following these steps:

1. Select the drupal8 database from the left sidebar.
2. Click on the Export tab.
3. Select **Custom - display all possible options** as our export method.
4. Select **SQL** as our format.
5. Choose **Add DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT/ TRIGGER statement** from **Object creation options**.
6. Click on the **Go** button.

At this point, we have a new file named `drupal8.sql`, which contains a backup of our database. Next, we will use this file we just created to restore our database.

**Database restore**

Restoring a database is simpler than backing our database up. Except this time, we will be using an existing database snapshot that either we have taken or that we were provided to overwrite our current database files with. Let's begin by following these steps:

1. Click on the **Import** tab.
2. Click on the **Choose File** button and locate the `drupal8.sql` file we created earlier.
3. Click on the **Go** button to begin the restoration process.

The process of restoring the file can sometimes take a minute to complete, so please be patient while the file is being restored.

While phpMyAdmin allows us to manage database operations, we can choose to use other database tools or even the command line, which is a lot faster to export and import databases.

Now that we have a good understanding of how to back up and restore our database, it's time to take a quick look at how we will be using Google Chrome to review our HTML and CSS structure within Drupal 8 and our theme Mockup.

**Using Google Chrome to inspect elements**

While there are many different browsers to view web content on, Google Chrome is definitely a favorite browser when theming in Drupal. It is not only standards compliant ensuring that most HTML and CSS work properly but Chrome also allows us to inspect the HTML and CSS and preview changes "live" within the browser using the **Developer Tools** option.
Setting Up Our Development Environment

Begin by browsing to our local Drupal 8 instance in Google Chrome and then selecting **Developer Tools** from the **View** | **Developer** | **Developer Tools** menu. The **Developer Tools** will open up in the bottom of your browser, as shown in the following image:

![Developer Tools in Chrome](image)

There are several tools available for our use, but the one we will use the most is inspecting **Elements** on the page, which allows us to view the HTML structure and any CSS being applied to that element from the **Elements** and **Styles** panels. We can navigate through the HTML structure, or if we prefer to isolate an element on the page, we can place our cursor on that element and right-click to open up a context menu where we can select **Inspect**. Doing so will target that element in the **Elements** pane for us.

As we dive deeper into theming, we will use this set of tools to help preview changes as well as isolate any issue we may be experiencing as our HTML structure changes based on what Drupal 8 outputs.
Summary
We collected a lot of information to start our series on Drupal 8 themes. Let's review exactly what we covered so far:

- We successfully configured an AMP (Apache MySQL, PHP) stack by downloading and installing MAMP PRO.
- We set up our first Drupal 8 instance by downloading the latest version from Drupal.org, importing the Drupal instance into our AMP stack, and completing the Drupal 8 install by choosing our language, profile, database settings, and site information.
- We also had our first look at Drupal 8 and some of the new responsive functionality that it provides. We familiarized ourselves with the admin menu and the new admin interface, which included the Content, Structure, Appearance, Extend, Configuration, People, Reports, and Help sections. Having a better knowledge of Drupal 8 and its folder structure has given us insight into how to apply best practices to manage our theme and its assets.
- By using the project files, we learned how to manage database snapshots through importing and exporting SQL files inside phpMyAdmin.
- Finally, we learned how to use Google Chrome to inspect our HTML and CSS to have a better understanding of our theme and its markup.

In the next chapter, we will take a closer look at "theme administration" and answer the question: what is a theme? We will explore the "appearance interface" and discuss how Drupal's default themes function. Finally, we will follow up with looking closer at how to use prebuilt themes and managing content with blocks and custom block layouts.
Where to buy this book

You can buy Drupal 8 Theming with Twig from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.