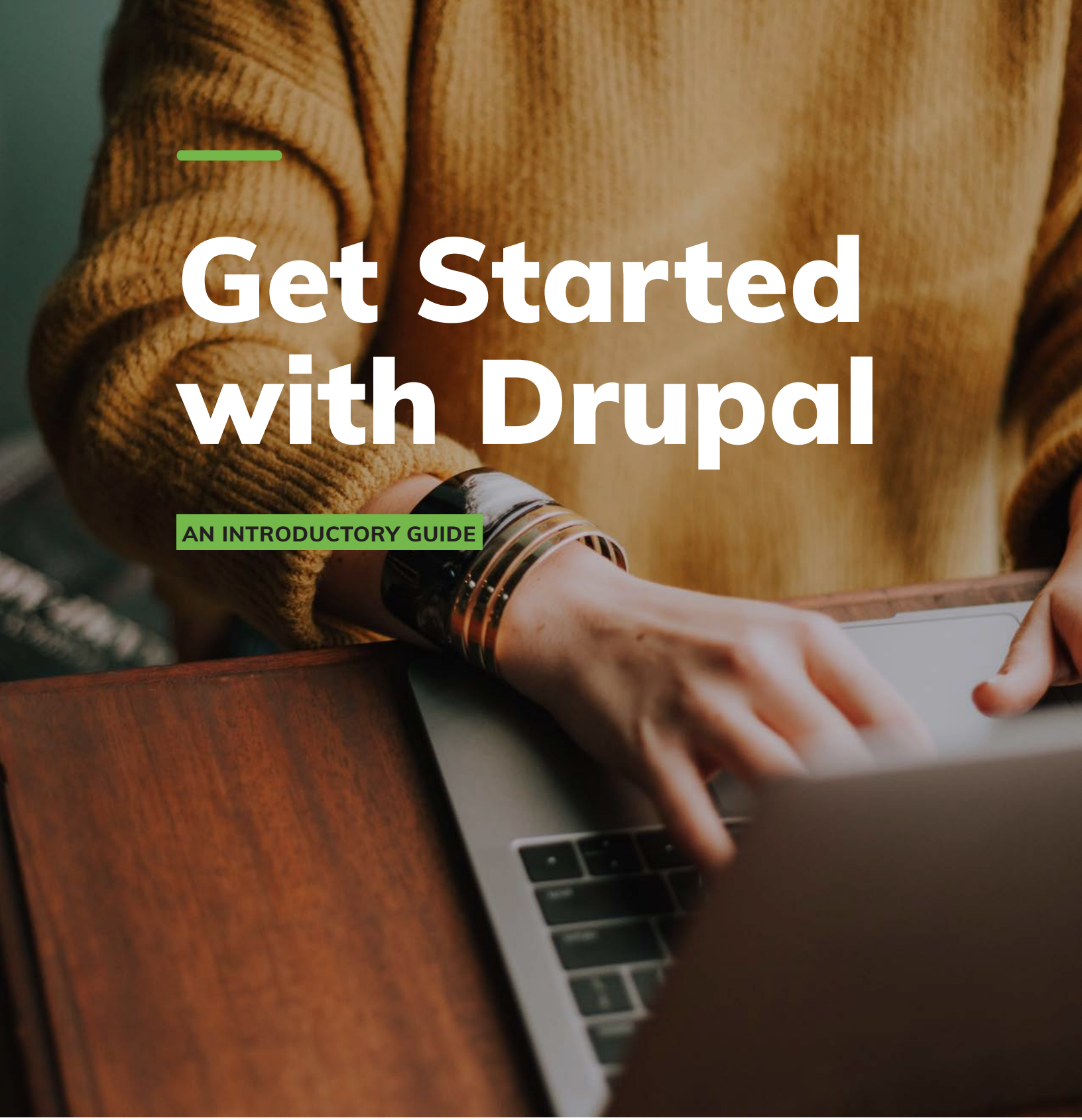


Get Started with Drupal

AN INTRODUCTORY GUIDE

evolvingweb

Follow us!

Contents

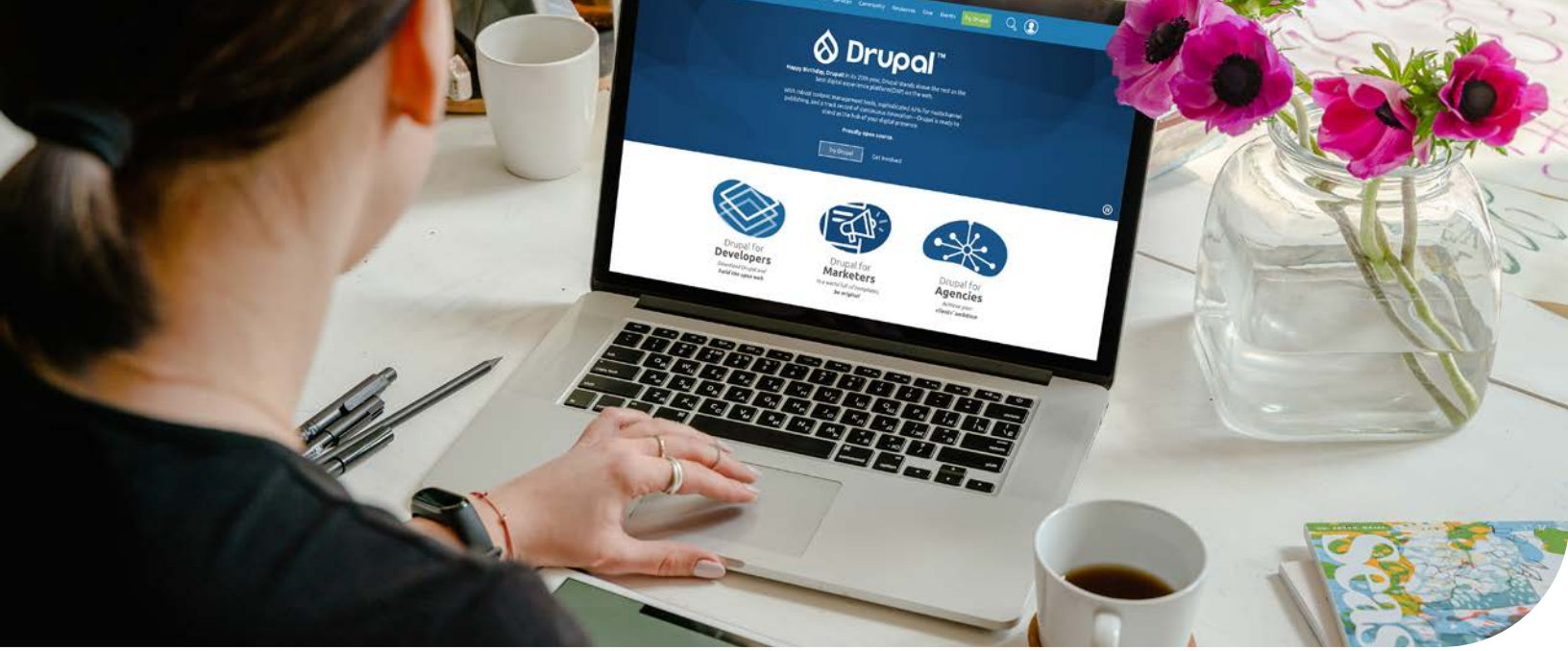
Why not build a website from scratch?

If you take any random selection of websites, you'll see the same types of interfaces and features appearing again and again. Most websites have menus to navigate content, unique URLs for individual pages, and features for content editors to log in and build new pages, and upload images and files. Building all of this from scratch every time you create a website is extremely inefficient.

On the flipside, going with a cookie-cutter, one-size-fits all solution isn't much better – without sufficient flexibility, you'll spend valuable development time and effort trying to reengineer things so they work and look as you want them to.

The solution lies somewhere in the middle: you want a framework that supports your development efforts without restricting them and gives you a baseline set of functionality that you can take and run with, however you see fit.

Drupal does just that, providing a flexible, extensible foundation for all sorts of digital experiences.



Drupal 101

What is Drupal?

Drupal is a digital experience platform (DxP), which is essentially a content management system on steroids.

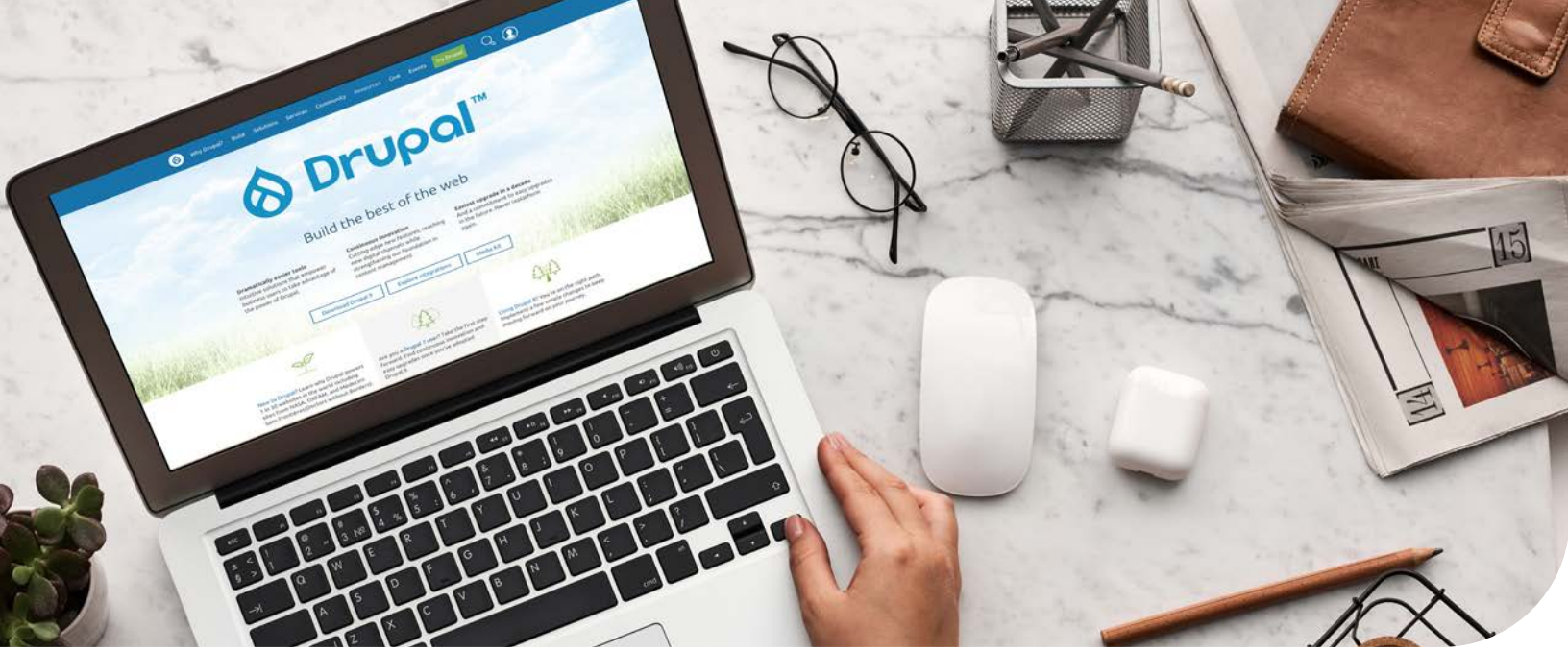
Like other platforms, Drupal does more than just let you build websites—it also integrates with your full marketing and analytics toolset, allows decoupled and distributed content publishing workflows, and supports the development of web apps and other rich interactive experiences.

Drupal has lots of benefits:

- Flexible, easy-to-use content authoring tools
- “Create once, publish anywhere” approach
- Highly customizable featureset
- Large number of freely available modules (add-on functionality)
- Innovation, powered by a massive, engaged community

One major advantage it has over other platforms is the fact that it’s so flexible—while most other platforms focus on serving a specific use case, Drupal was designed to accommodate whatever type of project you can dream of.

Read more about the benefits here.



Overview of Drupal functionality

Flexible features for building a rich user experience:

- **Views**, a UI tool to easily create dynamic content listings in a variety of formats (e.g. latest news, map of locations, events calendar, list of content recommendations)
- **Customizable blocks** (aka content components) to create calls to action, banners, promotional cards, featured content items, etc. that content editors can use to build pages
- **Role-based security** allows you to create granular user roles with configurable permissions
- **Built-in multilingual support** allows you can install Drupal in any language, or enable multiple languages and translate all your content and configuration
- **Flexible navigation system**, allowing site administrators to add, update, and reorganize navigation menus as needed
- **Drupal's search module** provides basic full-text search and filtering by taxonomy, and can be integrated with Views so that any list of content becomes a search interface

Content management features that content editors will love:

- **A WYSIWYG** authoring and editing tool with the ability to add rich media
- **Layout Builder:** a powerful no-code solution to easily build engaging pages with templated layouts and drag-and-drop UI for placing content on the page
- **Authoring tools** allow content editors and admins to easily search for, draft, edit, preview, archive, publish and update content
- **Customizable workflows** and approvals, as well as revisions, so that you can track every content update and revert back to a previous version
- **Media** supports local audio, video, images, files, as well as remote content from YouTube, Vimeo, Twitter, etc.
- **Media Library** allows users to add existing media assets to a site as well as upload new items directly into the library

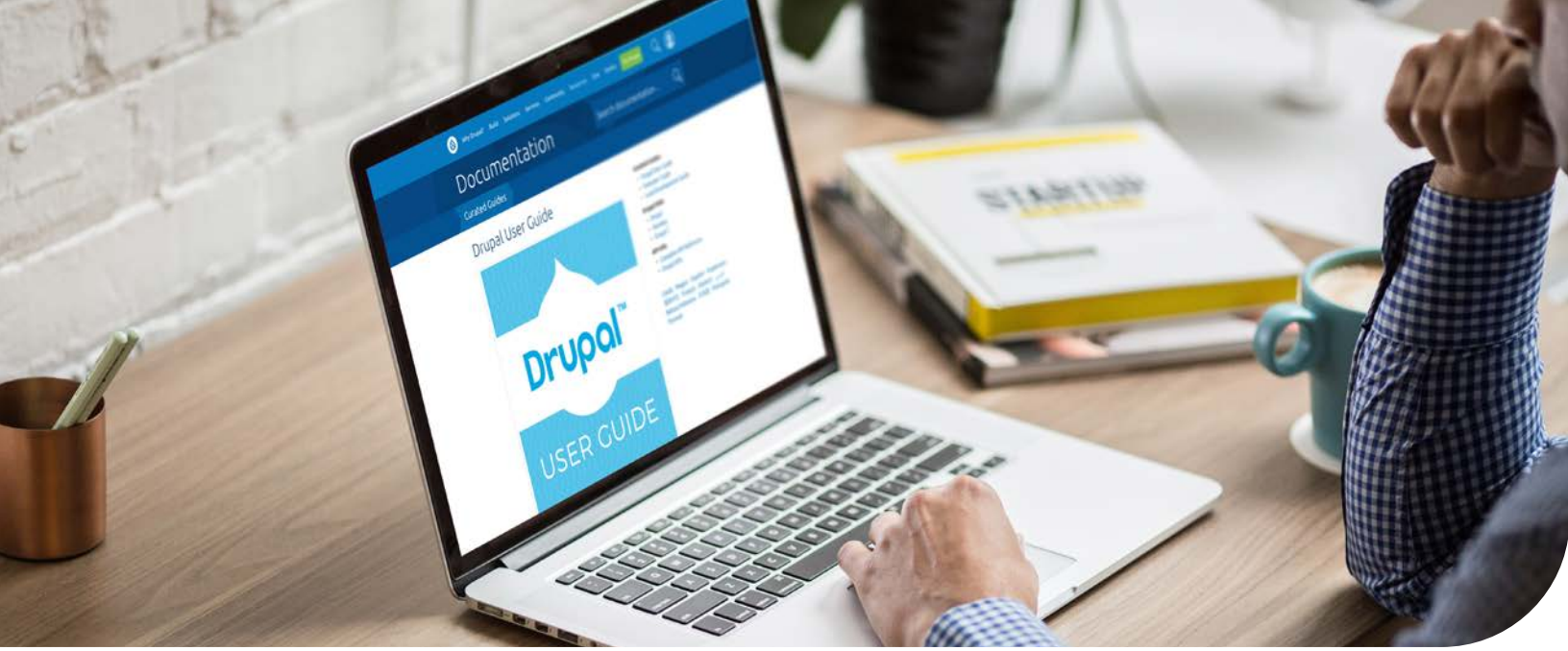
Features that make Drupal secure, accessible, and enterprise-ready

- **API-first architecture** means that you can feed data from Drupal to third-party systems using a JSON feed or REST API
- **Migrate** system allows you to feed data into Drupal
- **Accessibility** compliance out-of-the-box (WCAG AA)
- **Performance** optimized caching mechanisms
- **Theme system** allows you to create a completely custom, responsive front-end, built according to your brand guidelines and using your framework of choice
- **Recommended add-on (contrib) modules**
- **Webform:** Easy-to-use form builder to create anything from a simple contact form or survey to complex, multi-step application forms
- **Scheduler:** Schedule when your content is published in advance
- **Pathauto:** Customizable, user-friendly URLs
- **Metatag:** Configurable metatags for every piece of content
- **Search API:** A robust search experience and integration with enterprise-grade search engines (Elasticsearch or Solr)

Key Drupal Terminology

Here are some common terms you'll come across when reading about Drupal. You can find a **full glossary over at Drupal.org**.

- **Node:** A template for a specific type of node (blog post, event listing, landing page, etc.) Usually each content type has a set of fields that authors use to create it. A piece of content. Usually, every node has a unique URL.
- **Content type:** A template for a specific type of node (blog post, event listing, landing page, etc.) Usually each content type has a set of fields that authors use to create it.
- **Taxonomy:** Vocabularies and terms used to organize your content. For example, this is what allows you to tag and categorize blog posts or news items.
- **View:** A list of content (a simple news list or a more exciting list like a map or a calendar)
- **Module:** Code that you can add to your Drupal website to enable new functionality
- **Theme:** Defines the layout and design of the user interface
- **Block:** Container for displaying anything on a page (the search form, the logo, the copyright notice in the footer)
- **Permission:** A task that a user can do (e.g. viewing content, posting a comment, editing an event)
- **Role:** A type of user (e.g. author, editor, or member)
- **Drupal core:** The out-of-the-box features and functionality that Drupal provides
- **Contrib module:** Add-on functionality, made available by the Drupal community
- **Custom module:** Add-on functionality, built in-house to address the need for a specific project (e.g. a module that integrates with a custom CRM)



Planning your Drupal project

Make your goals clear, rather than prescribing a solution

Start with your end goal and work backwards from there. Drupal has a host of modules and built-in content editing features that you can take advantage of, so it's well worth taking the time to properly assess your needs. Rather than creating detailed specs or wireframes for every authoring feature you can think of, try installing Drupal and experimenting with the tools it gives you out-of-the-box.

One way to do this is to create a prototype of your project, starting with Drupal's core functionality, and then evaluate where your gaps are. Don't be afraid to workshop things and try out some of the modules you can find on **Drupal.org**. Prototyping various options does take time, but it's also the secret to so-called "thinking outside the box". Encourage teams to integrate prototyping into their workflows.

Leave time to research, test, and evaluate modules

Sometimes, the best option isn't the most obvious. Drupal lets you do basically anything you want, and there are usually a host of different ways to get it done. Give your developers time to not only research available solutions, but also test them hands-on.

Bring your team up to speed with Drupal

Everyone on your team should have a foundational understanding of key notions like accessibility, information architecture, and basic content management, in addition to being comfortable with Drupal-specific terminology and concepts.

Additionally, different roles will need to familiarize themselves with specific areas. Here's a short list of skills different members of your team should acquire when moving to Drupal.

Project managers

- Project estimation
- Digital best practices, technologies, and trends
- Agile & web deployment processes
- Drupal content management tools

Content editors & marketers

- Content accessibility
- SEO best practices
- Content governance & workflows
- Content style guides
- Drupal content management tools

Site builders

- Information architecture & content strategy
- Selecting modules
- Drupal configuration fundamentals (content types, taxonomy, views)
- Paragraphs & layout builder
- Media and image styles
- Configuration management

Front-end developers

- Drupal configuration fundamentals (content types, taxonomy, views)
- HTML, JavaScript, SASS
- Twig templating
- Creating Drupal themes
- Atomic design
- Web accessibility
- Drupal development workflows

Back-end developers

- Drupal configuration fundamentals (content types, taxonomy, views)
- Selecting modules
- Creating and extending Drupal modules
- Drupal development workflows
- Writing content migrations
- Testing and applying patches

Interested?
Evolving Web offers in-depth Drupal training for every member of your team.
Check out our upcoming course listing!

Adopting Drupal

Is your team ready to take their first steps with Drupal? Here are some best practices that'll help you set things up in a way that keeps things running smoothly.

Don't start with your biggest build!

Before you start rebuilding your entire portfolio of websites from scratch, it's probably a good idea to get a grasp on the basics in a more forgiving scenario. If you'd like a sandbox in which to play around with Drupal and learn the ropes, there are a couple of options:

- Work with Drupal locally on your computer, installing tools like Visual Studio Code (**instructions here**)
- Create a test site in the cloud (no local installation required) with a free trial from **Pantheon**

Spend time on design and user experience

When you think about your website's design and UX, what first comes to mind is usually the public-facing side. But your site has another kind of user that's just as important: internal team members who create, publish, maintain and manipulate its contents.

Make sure to establish good communication between website managers and developers. A mutual understanding of each other's needs and workflows can help streamline everything from content publication to taxonomy management.

Create a set of best practices for your designers, developers, and content editors

Adopting a new technology means revisiting existing processes and workflows and adapting them to the new context. Oftentimes, building something new is the easy part – what can be difficult, especially in large organizations, is changing processes.

Be agile and open to change, and don't be afraid to challenge your team's preconceived ideas. If you're getting started with Drupal, think of it as a clean slate.

Set up a development workflow

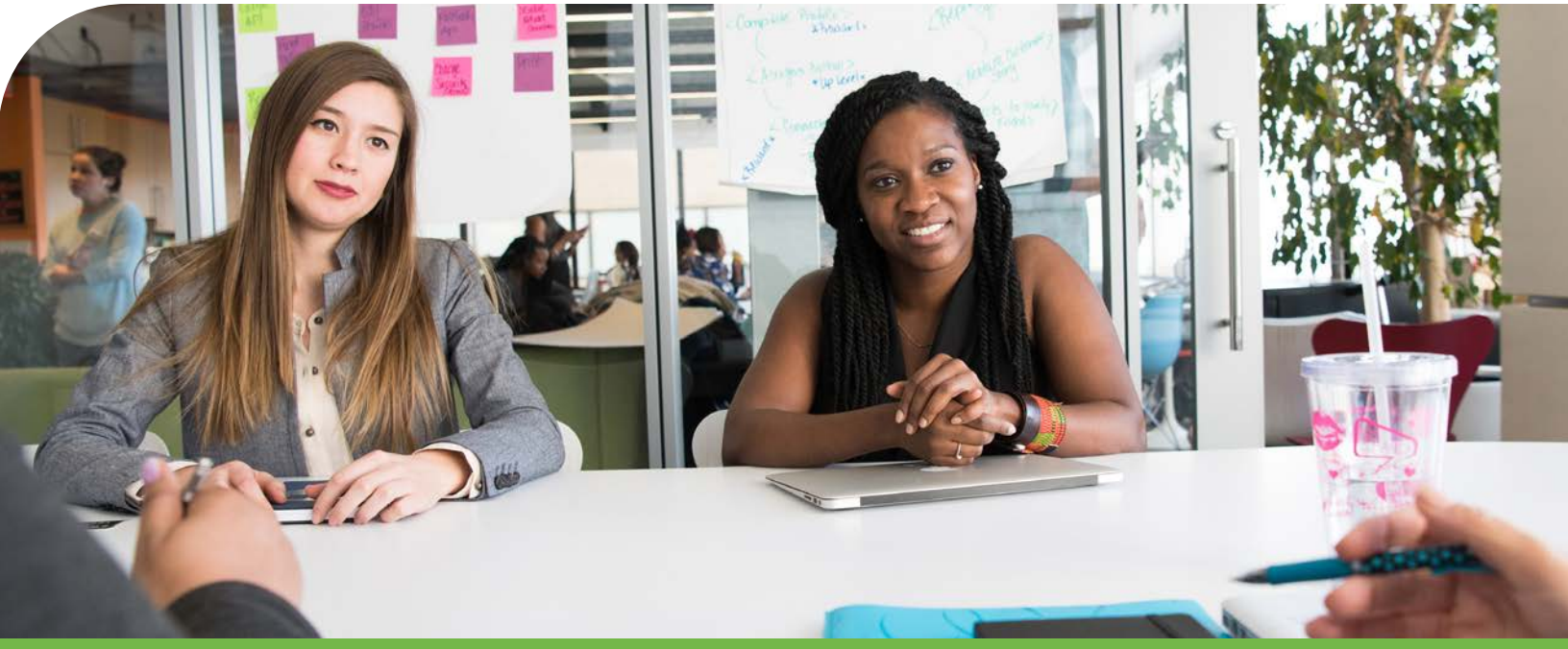
In the same vein, building an efficient and effective development workflow might seem like a lot of work, but it's always worth it.

Create environments for staging, development and production from the beginning.

Think about how different pieces fit together before everyone sets off and does their own thing (and likely picks up some bad habits along the way). It will save you a ton of time down the road, guaranteed.

Need help with this?

Sign up for our next in-depth training session about development workflows for Drupal.maintain, and facilitate onboarding for other developers who are working on your website.



Test and apply security updates

While it's vulnerable to security flaws like any other piece of software, Drupal has the benefit of being open-source, which means a huge development community is ready to spring into action and promptly release patches when necessary. All you need to do is, well, apply those patches in a timely manner—but only after testing them in a staging environment, just to be safe.

Invest in the right infrastructure to get the most out of Drupal

One way you can make your Drupal experience even better is by partnering with the right technological partners. Rather than going with on-premise hosting or a generic Cloud solution, choosing a Drupal-focused cloud hosting provider can save your team a lot of time, and ensure that you're using the best-in-class tools for security and performance.

Our partners at **Acquia** and **Pantheon** provide enterprise-grade Cloud hosting for Drupal. Benefits include:

- Development/staging/production environments ready-to-go
- A user interface to move your content, files, and code from one environment to another
- Support from Drupal experts
- Support for add-on features so you can integrate Drupal with an enterprise-search solution and other complementary tools
- Command-line tools that work directly with Drupal
- Performance optimization and security features
- Automated backups

Similarly, if you decide to work with an agency to develop or redesign your website, pick one that has demonstrated experience with Drupal specifically.

Did you know?
Cloud hosting can reduce the time it takes to deploy your website, make your site easier to maintain, and facilitate onboarding for other developers who are working on your website.

Useful resources

Some Drupal essentials for your bookmarks folder.

Official Drupal Documentation

Drupal Evaluator Guide

The Weekly Drop newsletter

Drupal Sun

/r/Drupal

Evolving Web's **blog**, **resource library** and **YouTube channel**

Twitter accounts to follow: **@drupal**, **@dries, @drupalcon**, **@DrupalAssoc**, and of course **@EvolvingWeb**