

Getting 'Drupalyzed' - Toolkit for the newbie

[Profile picture for user admin](#)

by admin Saturday, August 11, 2018 - 11:17 [comments](#)
[web development](#)

The web provides access to a wealth of resources stored in far flung servers but accessed in multiple ways. It is a language worth learning. It opens doors to the curious web surfer.

The beauty of this process is that you don't have to know it all to get started. Only the brave and daring though. You must be ready for some stumbling blocks but not too shy to ask for help in the multiple forum out there! Unless you have a background in web development or programming, Drupal requires you to be attentive to its standards. Unlike other free opensource CMS that have a quick turn around but probably less grit, Drupal aims to take you to the finest details that allows you control the output of your webpage. Your main limitation will be your host server capabilities. However with increasing focus on streamlined and fast pace content delivery, the astute web developer would consider upgrading their server PHP to at least version 7.0, deploy FastPHP (PHP-FPM) and ensure server side 'caching' using OPcache in conjunction with modules as AdvAgg, Websocket among others. Website speed can be optimised on Drupal regardless of what you may have heard. Its all the backend that makes the front look not just good but fit for purpose.

My singular attraction to Drupal include the following though not exhaustive list:

- User access control (UAC)
- Multiple applications (Modules extend and themes present and libraries integrate - just my way of understanding)
- Multilingual (all in the box) framework
- Multi-site hosting
- Methodologies "the Drupal way"
- Tools for social change and community development
- Messaging tools for communicating within and outside the site
- Drupal Commerce improved capabilities in Drupal 8 - thanks to the team at Commerce Guys who have worked tirelessly to make this possible.
- [Webforms](#) with intuitive design, now also improved to make 'addressing' easier courtesy of [Jacob Rockowitz](#).
- Security settings 'from the box' improved by use of various modules to minimize data breaches etc.

As they say 'horses for courses' and so it is with the choice of CMS for any web developer or business. Having had positive experiences in a few other CMSs, I understand that the choice of anyone CMS over another is a matter of know-how, use-case and then preference, possibly in that order. It can be demanding managing client sites post deployment if the clients are ill equipped or lack know-how of Drupal. My first client website was built using Wordpress. The client at the time was a small enterprise which soon grew to become a subregional interest, with rapidly growing staff numbers. Still new on Drupal at the time but filled with enthusiasm regarding its possibilities for this client, I proposed they could do better on Drupal (then version 7) with some guidance. I upgraded their website from Wordpress to Drupal 7. I probably should have known I was consigning myself to a 24-hour support service by this move. Despite efforts to train the communications team on its management, it remained a hands on support for most of the time which was not the plan. It was probably too big a leap and there was no real interest from 'client-side' to acquire the know-how. They hadn't really managed their Wordpress site sufficiently to gain the relevant skills on a more complex system as Drupal. This was a learning point for me. If your client can't go the mile they certainly can't make that journey. Arguably, that client's team learnt a few things and have now being able to deploy their website back on Wordpress.

For the moment I continue to explore the most I can out of Drupal and with the fast pace of changes in Drupal 8 in terms of modules and functionalities, I am optimistic it heralds better things for the future for Drupal lovers. There are of course [other PHP-based CMSs](#) out there essentially 'serving' the web but just doing it differently. Some such as Backdrop have roots linked to Drupal. Others; Wordpress, Joomla, Cake PHP, Modx, PHPWiki are configured differently though also utilizing PHP-based servers in the main. Most CMS are driven by frameworks which are applications (as in windows) that improves functionality of the website. The more advanced these applications the slimmer the package and of course the better the user experience. Drupal 8 is an attempt to optimise Drupal by switching to 'Twig' and Symphony which allows content to be coded or 'manipulated' more intuitively as entities, further expanding the capabilities of PHP exponentially - probably an over simplification here but you get the drift. [Other applications](#) with this capability include Laravel, Cake PHP, CodeIgniter, Zend Framework etc., which I am somewhat also curious to learn about.

The commonest reason some like me, who are self taught, avoid anything to do with cml (command line) interfaces was the a non-programming background. Since delving into the world of web developing and data systems management, I have braved the storm to learn, sometimes the hard way, from trial and errors. However there are loads of resources out there to help one find their way. You may get burnt, loose your way or just simply be stuck (the commonest experience). However if problem-solving is what keeps you going then you have just made the team!

Tools that would aid your Drupal experience

Having root access to your server would be ideal. This is easy if you have a windows compatible server e.g IIS for windows that allows you to run php and for some, ASP. Drupal can run on windows IIS (my experience) or MAMP, WAMP, XAMP - few of which I started off with at the beginning.

- Running as root user on a SSH (windows Bash, Linus server Putty - my experience) especially in drupal 8
- Familiarity with Composer:
 - essential for setting up a drupal installation package,
 - for installing themes and modules
 - for updating core/modules/themes/dependencies
 - for house keeping including clearing caches, running diagnostics, installing other libraries,
- Familiarity with Drush (Drupal Secure Shell). Drush does a host of things that makes the life of a Drupal user much easier. I was shy of Drush far too long.
- Drupal console (newer and less experience in my case),
- NPM (node packaging manager) - sourcing of some third party node-modules, not essential though.
- Github,
- Baseline knowledge of html,css, php, javascript, jquery, SQL, OOP (in various formats) and usually in that order.

Installing Drupal

The following link is a must read to effectively install your [Drupal website](#).

In summary you are usually installing an entirely new Drupal 'site' or you are re-installing a previous site: upgrading or just deploying on a new server (migrating).

The steps are slightly different but essentially have common footprints.

It is probably safest to learn by installing a brand new site and be comfortable engaging the server on the three main fronts below before attempting to reinstall a running site. And in all cases, do this on a development server.

Three main steps -

- Installing:

Drupal 8 core on the webroot, requires access via backend (cpanel files, cml via SSH using composer or Support (not ideal)), modules and themes and other libraries

- Creating:

A database with username and password for the site - via PhpMyadmin on most linux servers or also via cml if with root access to SSH.

- Running the new installation

Typically done via a web browser which takes you through visual steps of the installation process. But can also be done via cml using composer for more saavy drupalbytes.

At some stage it would become necessary to work with cml using composer, drush, Git etc. Anticipate that stage will come but like most beginners I would recommend starting with the basic browser interface. Drupal was designed for the end user but now recognizes it can be a steep curve with many slipping and not making it to the top.

•

Tags

[IT](#)

[Web CMS](#)

Section

[Information Technology](#)



[View PDF](#)