



Ready to *Become a WordPress Master*? This is the full version of our seven day free course that teaches you the 20% of knowledge you need to get 80% of WordPress work done.

If you're not already signed up, [check out MasterWP.co](https://masterwp.co).

Here's what's in this PDF:

1. How to choose quality WordPress themes
2. How to use WordPress theme support
3. WordPress customization basics (useful tools and plugins to edit your theme)
4. Make your site load really quickly
5. Using Photon and CDNs for extra speed
6. Harnessing WordPress, WooCommerce and email marketing to sell more products
7. Crash course in WordPress security.

If you have any queries feel free to reach out to us via email:
hello@masterwp.co.

Enjoy!

Happy learning :)

– Alex and Ben

How To Choose Quality WordPress Themes

There are tens of thousands of WordPress themes available, some free and some “premium” (which just means paid). These are a huge asset and one of the main reasons for using WordPress as they allow vast amounts of time to be saved when setting up a site.

With so many themes to choose from there are huge disparities of quality. A bad theme can do your site a lot of harm; a decent theme can be okay; a brilliant theme can make a huge positive impact.

Whether you're installing your first or thousandth WordPress theme, being able to quickly spot quality is an invaluable skill which can **dramatically cut the time** it takes to get a site together.

With so many themes to choose from, how do you choose a quality WordPress theme? We'll run you through what to look for and what to avoid.

What to avoid when choosing a WordPress theme

There are some red flags you'll always want to avoid:

1. Theme reviews and roundups need to be taken with a heavy pinch of salt.
2. “Kitchen sink” themes that do it all can be a lot less appealing than they first appear.

We'll look at these both in more detail now.

Theme reviews and roundups

Don't take theme reviews, roundups or recommendations at face value.

Theme companies offer affiliate programs and the companies with the most generous referral payments get recommended the most. Theme companies will pay significant sums to the blogs with the highest ranking posts for popular search terms to appear on their recommended lists – the recommendations aren't typically made on merit. Bear this in mind when reading roundups.

Individual theme reviews suffer from the same problem and rarely tell you anything insightful, offering at best superficial analysis. Best to just come to your own conclusions.

We'll show you how to spot genuine theme recommendations and what to look for when evaluating themes later on.

'Do it all' themes that do a lot less

Some of the most popular themes are uber-flexible "kitchen sink" or "do-it-all" themes. The appeal is obvious: buy one theme and have it do everything.

The caveat is themes which offer to do everything "without customising code" make it very hard to make changes *outside* of what the theme offers, especially if you want to customise code. It takes longer learn how to customize what looks like a less flexible theme, but once you've learned, you'll have a lot more control (we'll cover this in detail in a lesson later this week).

The need to cater for all circumstances also typically makes these themes much more bloated. With page speed a major consideration for visitors and Google a leaner theme is the much better option.

Kitchen sink themes are often complex to customise and can take a lot of time to look the way you want. The demos are set up by people with fantastic design skills who know the theme inside out. Most users don't have the knowledge to reproduce these things - at least not without investing a lot of time.

What to look for in a WordPress theme

Once you know what the red flags are they are easy to avoid. We can now run through your easy checklist for finding a really good WordPress theme.

The WordPress theme essentials

There are a couple of essentials you'll want to always tick off and then some more subjective items. The essentials include:

Check out the documentation. Most theme providers will let you see documentation before you buy. Is it thorough? Does it look well written and useful? Are there video tutorials? Solid documentation is generally a good sign the rest of the theme is sound too. If the documentation isn't publicly viewable, ask if you can see it.

Check ongoing support and update costs. You should expect to receive support and updates for at least twelve months and to then pay a reduced fee for continued updates. If it's anything other than this, make sure you're aware of the ongoing costs when making your purchasing decision. Not paying an annual fee

can be bad as it's much more likely the developer will cease to find it financially viable to update your theme. Those updates are key for security, as we'll discuss in a future email.

What support is included? This is a really important one. Good support can save you a ton of time (we'll show you how to get the most out of premium theme support in the next email) so it's important to know what you're getting. Is just theme setup covered, or will you get customisations too? What if you need help with something "above and beyond"? A good test is to ask a pre-sales question (even if you know the answer) to check the response rate and quality. You don't want to be waiting too long for a response.

Check the refund policy. Does it exist? What time period does it offer? There's always a chance a theme just doesn't work out in practice – being able to get a no-queries refund straightaway is useful here. You'll be able to tell if this is the case immediately so seven days is ample here.

Is the theme GPL? WordPress is released under the GPL license. This means that the code is free to use as you wish. Since themes make use of WordPress they should also be GPL. If theme shops do not mention this or try to hide the fact, then it's possible they are hiding other things as well.

Bonus: Google common problems. This extra check can be helpful: do a quick Google search for "theme name" broken and see what comes up. Issues showing up isn't necessarily a problem if they're fixed promptly; it's recurring issues with no fixes that you want to be on the lookout for.

Extra bonus: search or ask for discounts. Most theme shops have sales from time to time. Before you buy do a search for discounts. If you can't find any and there's a discount code box on the checkout, ask for one. Be reasonable – 20% off is fine, 50% off is unreasonable – but you may get what you ask for.

These are all technical points but going through the essentials first lets you eliminate the bad themes straight away, rather than being seduced by badly made themes that manage to look good.

The WordPress theme subjective features

Once you've covered off all of the essentials, it's time to think about **design**. You want a theme that looks good, but "good" means different things to different people. A couple of things to think about:

- Is the theme suited to your kind of content? Imagine your posts on the site, not the specially crafted and carefully edited content of the demo. Do you

have suitable imagery for a media-heavy theme?

- Is content readable? Look for a decent font size and good line spacing.
- Is it easy to publish content? Or is the theme adding lots of extra bits of content you'll need to add each time. Be wary of adding extra time to publishing content.

Theme demos are the place to try out all of these things. If possible remove any layover boxes and see the theme as visitors will see it on your site. Whilst you're doing this, resize your browser window to test the responsive design features. Test for tablet and mobile devices; in a well made theme everything should work flawlessly.

The final thing to consider is **SEO**. This is subjective as it's difficult to measure without seeing any code.

Running the demo through [Google PageSpeed](#) is a good starting point; you want to be looking for a decent score there. Otherwise, any well-built theme will do a good job when coupled with some speed improvements and SEO plugins (we'll cover both of these in future emails).

A general rule of thumb is if the theme has passed all of the checks up-to now, it's probably fine. Look out for audits by SEO agencies but take other alleged SEO benefits with an awareness that, as they are difficult to prove, they may just be marketing speak.

Where to find quality WordPress themes

Where do you go to find all of these feature rich, well engineered and good looking themes? Generally independent theme shops run by a small team of passionate developers are the places you want to be looking.

Marketplaces such as Creative Market or ThemeForest don't have a guarantee of quality across all their themes and should thus be avoided.

We have personally used and can recommend:

- [ProThemeDesign](#)
- [Array Themes](#)
- [WPZOOM](#)
- [Elmastudio](#)
- [Themetry](#)
- [Theme Foundry](#)
- [Anariel Design](#)

Otherwise, theme shops who sell on wordpress.com tend to have good quality themes since the wordpress.com team reviews all of the code in the themes they sell, so the standards will be high. In addition the WordPress.org Commercial

Themes page are usually pretty good (although make sure to consider the checks).

Good themes provide solid foundations

It is worth spending some time getting the right theme. As you work through your journey to become a WordPress Master you'll get faster at choosing quality themes – and you'll find that picking out the right theme saves time and money down the road as you don't have to swap your mediocre theme in six months as you run up against its limitations.

How To Use WordPress Theme Support

The most efficient way to get your site set up the way you want it is to utilise support that comes with your WordPress theme. If you're using a free theme you can probably get limited support from the developer and/or volunteers, but if you've paid for a theme you can expect personal support from the theme developers.

Different themes come with different levels of support, but it's typical to get six or twelve months of support and updates included with your purchase. After that period you'll likely need to purchase an extension for an extra fee. As we discussed in the last lesson, it's something to check when purchasing a premium theme. If you've already made the purchase, check now.

However long your support period is, the limited time frame means it's best to get all your queries out of the way as soon as you make your purchase. That means setting up and customising your site right away. By knowing what to ask and how to ask it you can get a lot more out of premium support. In this lesson we'll show you how to **minimise your time and maximise results**.

Troubleshooting your own problems

It's often easier to ask someone to fix your issues, but it's certainly faster to do it yourself. If you know the right questions to ask you can eliminate common problems with WordPress themes. These are your pre-support checks:

1. Check documentation and FAQs

Before you do anything, check your theme's documentation and general FAQs (if there are any). If you're using an especially popular theme, also Google your problem. A quick search can often yield the answer you're looking for and save going through the support process at all.

If it's a common issue then it's likely in your theme's documentation - so a developer is likely to just link you to the relevant docs. So it's worth checking first - it can save everyone time.

2. Make sure everything is up-to-date.

Check for updates to WordPress, themes and plugins. WordPress includes automatic updates but they don't always kick in (we'll cover how to automate this in a later lesson).

It's really easy to check if WordPress is up-to-date: from the Dashboard, head to Updates and you'll find all versions of WordPress, plugins and themes that have updates available. Follow the prompts to update each in turn. You should find the process straightforward and reliable. If your site gets stuck in maintenance mode [check this tutorial for how to fix it](#).

If you're using a **premium theme** then you may not have automatic updates available. It's exceptionally important you make sure your theme is up-to-date as if it's not that'll be the first thing you're told to do by support staff. You want to check the version number of your theme install, which you can do by heading to Appearance and then Themes, hovering over the active theme and clicking Theme Details and you'll see the version number in the top right. Make a note.

You should have some sort of "Member Area" from your theme provider. Log in there and check the downloads section. If the version number is higher than what you've got installed, you need to update. Your theme provider will likely have some sort of update tutorial, but the general process for manually updating a theme looks like this:

1. Download the latest theme version from your Member Area.
2. From Appearance and Themes on the Dashboard, deactivate your current theme (do this by activating another theme).
3. Hover over your premium theme and click Theme Details.
4. Press Delete, in the bottom right corner and then confirm.
5. Press Add New and then Upload Theme on the top left.
6. Upload the zipped file.
7. Done!

It's a bit of a clunky process but it does work. The way to avoid this, of course, is to get a theme with auto-updates built in :)

3. Check for plugin conflicts

This is an extremely common cause of WordPress theme problems and the second thing you'll be asked to check after updates. Do it before sending your query and you'll get an answer faster.

The standard spiel from theme developers is there are tens of thousands of WordPress plugins and inevitable some will conflict with themes and impossible to test compatibility with them all.

This is true, but what usually follows is a request to deactivate all your plugins and then re-enable them one at a time until you find the culprit. This is extremely annoying (and time consuming) to do. Here are some pointers to quickly identify culprit plugins:

- Plugins using jQuery (sliders, popup boxes) are most likely to conflict. If you're using any of these, deactivate them and test your site first.
- [WP Super Cache](#) or any other caching plugins aren't likely to cause conflicts but may hide changes from showing. Make sure you clear your cache to get the latest version of your site to show.
- Deactivate any plugins you have installed recently (before the problem happened) - perhaps they are incompatible.
- Switch to the default WordPress theme - at the time of writing that's TwentySixteen and see if the problem still happens, if so it's likely a problem with the plugin and not the theme.

Popular common plugins like [Yoast SEO](#), [Akismet](#) and [WooCommerce](#) are unlikely to be causing problems.

4. Debug the issue yourself

This is a little more technical but it can give you a really good idea of what the problem might be.

You'll find a file called wp-config.php in the root of your WordPress installation. Add `define( 'WP_DEBUG', true );` to the wp-config.php file and WordPress will then output errors, including the file that is causing the problem.

This can help narrow the issue down and, if it doesn't help you, sending the error to the developer can help them immensely. There's more info on using [WP_DEBUG on the WordPress Codex](#).

If you've looked through likely plugin conflicts, read the documentation and still not got a fix for your problem, it's time to reach out for support.

Efficient support requests

Most theme companies don't have the resources to keep large support staff, so you'll typically be kept waiting for a support response. If anything has to be clarified in your request, that adds time. **The way to get the fastest response is to include everything that's needed to give you an answer straight-off.**

There's some info you'll always want to mention:

- **The name of your theme** - the developer likely sells many themes so this will help them get started.
- **A reference number for your purchase** - just to prove you bought the theme.
- **You've got the latest version** of WordPress, the theme and all plugins.
- **You've tested for plugin conflicts** and had no results.
- **Your site URL**, and the url of the page having problems.
- **Screenshots of the problem** - particularly important if the issue is on a site that's not public (use [Awesome Screenshot](#) for this).

- **Temporary login details** (create an admin account and delete it after – nb obviously don't include this in any public forums).
- **You've checked relevant documentation.**

You then want to outline your problem. Do this in as much detail and as clearly as possible. Here's an example of a support request done badly:

I don't understand how to add the email form in the sidebar, it doesn't display?

And here's an example of a support request done well:

Hi, I purchased Exhibit on July 1st (reference 983478348) and am having some problems getting going.

Everything (WordPress, theme and plugins) is up-to-date and I've tested for plugin conflicts. My site is masterwp.co. I've created temporary login details: username – pass, password – wordpress. I'm attempting to use the Email Optin Widget in the Sidebar as per the theme demo but my Mailchimp API key is being rejected. I've followed the documentation but seem to be missing something. Could you take a look please?

Both are saying the same thing, but the second is packed full of relevant information that will make sure the issue is dealt with without the need for further (time consuming) correspondence.

Between us we've been answering WordPress support queries since 2008 and it makes such a difference having all the information to hand. Support people want to be able to answer your query as fast as possible, so give them the tools to do that.

Bonus tips for superb support requests

The above covers everything you need for a standard support request, but if you're unsure about what you can get away with asking see below.

Be polite - theme developers get a lot of support requests, and being polite is a lot more likely to get you a timely response. Being rude or aggressive will not encourage developers to help you - in fact I've heard of developers refunding rude customers and not giving them any help at all. They want their customers to be happy so be nice and they'll be nice in return.

Be patient - many theme shops are small teams (often just 1 or 2 people) and so they don't have the resources to reply immediately. In addition they may not be based in your timezone so might be asleep whilst you're working. If you've made sure to purchase a good theme then you will get a response!

Finally, it's worth making sure you have read the **terms of service**. Good theme developers will help fix bugs in their themes - but they may not support problems with plugin conflicts, or help with theme customizations. They are, after all, selling WordPress themes and not full customization services. Make sure that what you're asking for is something they will help with.

That said, if you think your request is borderline, follow the guidance we've offered here – be polite and patient and make it easy for the developer by offering lots of detail – and you can probably get away with it. No harm in asking.

Solving your own problems

Becoming a WordPress Master involves realising the limitations of your skills and using support to fix problems quickly. Being able to troubleshoot common support requests and send highly efficient support tickets is an exceptionally useful skill.

WordPress customization basics and plugins to edit your theme

Buying themes is all well and good, but you will nearly always need to customize them to make them fit your brand properly. Whether it's changing colours or fonts, or adding a logo - these things can make a big difference to the branding of the site helping to make it unique to you.

Why you shouldn't edit theme files

When you first start using WordPress it's tempting to just edit themes directly but **this is not recommended**. Initially it will be great as you can make changes quickly and conveniently – but then a theme update will be published – and you'll have to decide between updating (and making your changes again) and keeping the customized theme.

Unfortunately there's only really one option here.

You should (almost) always choose to update your theme. Not only will the theme update (probably) include new features but it will likely also include bug fixes, and sometimes security fixes. Security is more important than anything else - nobody wants to have to deal with a hacked website (we'll cover security in more detail in a later lesson).

To fix this we want to be able to make changes to how a theme works **without editing the theme directly**.

There's a few different ways to tackle this. In this email we're going to cover the easiest method, which is **how to make simple changes using plugins**. We'll equip you with the tools to make the changes you want.

Using the Customizer

Let's start with by far the easiest method of customizing your WordPress theme.

The Customizer is the future of WordPress. A lot of development will be happening here and most plugins we'll look at support the Customizer so it's worth starting with.

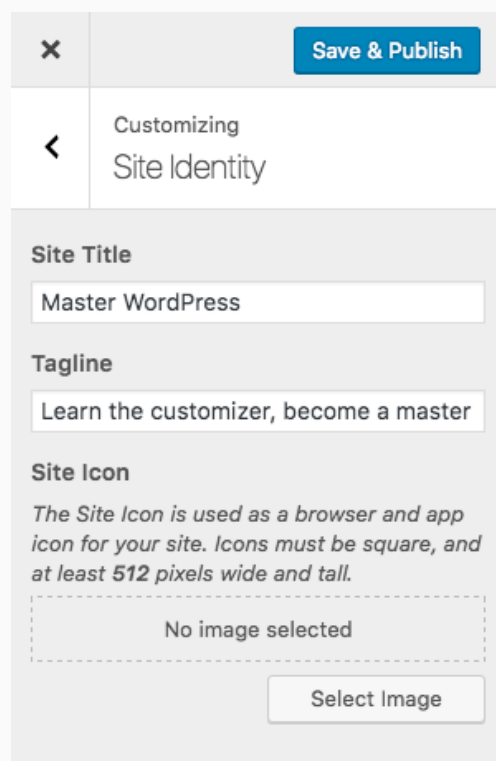
The Customizer is a place where you can edit theme options and see the changes you make apply to the website almost instantly. It's a great place to make changes to your site since you can edit things and see the effect quickly – but your users

won't see the changes until you press save. This means you can test things without the risk of your site breaking.

There's two ways to access the Customizer. From the WordPress Admin, you can go to the Appearance > Customize menu. Alternatively - when you're browsing your site itself, there's a Customize link in the header. Clicking this will take you straight to the customizer.

Before we look at plugins, these are all the areas you can customize with the Customizer (nb not all themes support all of these):

- **Site Identity** - this covers setting your site title and description, and a website logo. These are generally displayed in the header of your site. You can also set a site icon - this will be used in browser tabs to help people tell which website the tab is for.
- **Colors** - You can set a header text colour, and a site background colour here.
- **Header Image** - Set a site header image. If you like you can upload multiple images and set them to randomize - but I don't generally think this is a good idea. It's hard to make a memorable brand when your header image keeps changing.
- **Background image** - upload a background image, and set its properties (tiling and position). If you're going to do this then go for something subtle. Complex / bold images will detract from the content of your website.
- **Menus and Widgets** - control the menus and sidebar content that displays on your site. The widgets will only display the sidebars on the page you are viewing - so browse around the site to see the widgets you can change.
- **Static Front Page** - remove the blog posts from your site and set a page as the front page for your website. Great for sites that are about something, as opposed to a tradition blog/ news site.



The screenshot shows the WordPress Customizer interface for the 'Site Identity' section. At the top right is a blue 'Save & Publish' button. Below it is a breadcrumb trail: 'Customizing' followed by 'Site Identity'. The main content area is divided into three sections: 'Site Title' with a text input field containing 'Master WordPress'; 'Tagline' with a text input field containing 'Learn the customizer, become a master'; and 'Site Icon' which includes a descriptive paragraph: 'The Site Icon is used as a browser and app icon for your site. Icons must be square, and at least 512 pixels wide and tall.' Below this is a dashed rectangular box containing the text 'No image selected' and a 'Select Image' button at the bottom right.

It's fairly self explanatory to use all these. Use the menu down the left hand side to select the different areas and fill in the fields that pop up. Note that if something isn't visible on the page you're on you won't be able to edit it – so if you're on the homepage you won't be able to edit the post sidebar widget area (just click through to a post on the live preview to do that).

Once you're done making changes you can hit "Save & Publish" and your site will be immediately updated.

Tools for editing CSS

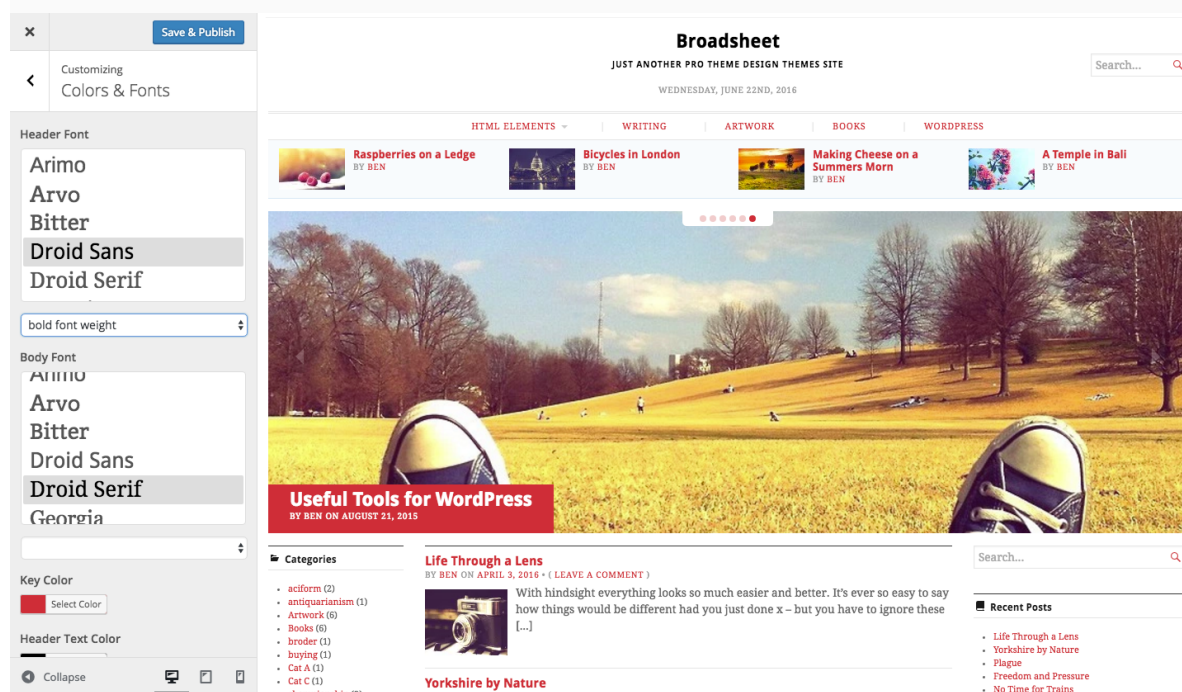
The Customizer is good for the basics, but to go beyond you'll need some extra plugin firepower. To really make progress you'll need to edit CSS. CSS is the code which styles your WordPress theme. To change how it looks, therefore, you need to change the code.

Or at least, in theory.

Learning CSS is complicated and whilst it may be worth it for some in the long run ([check here](#) for all the details you'll need), you can get a surprising amount done knowing very little code. The tools below will give you a lot of power with relatively little coding skill required.

Styleguide - Custom Fonts and Colours

The first tool we'll look at, and the easiest to use is [Styleguide](#). I made this plugin to help my customers at [Pro Theme Design](#) edit the fonts and colours in the themes they purchase.



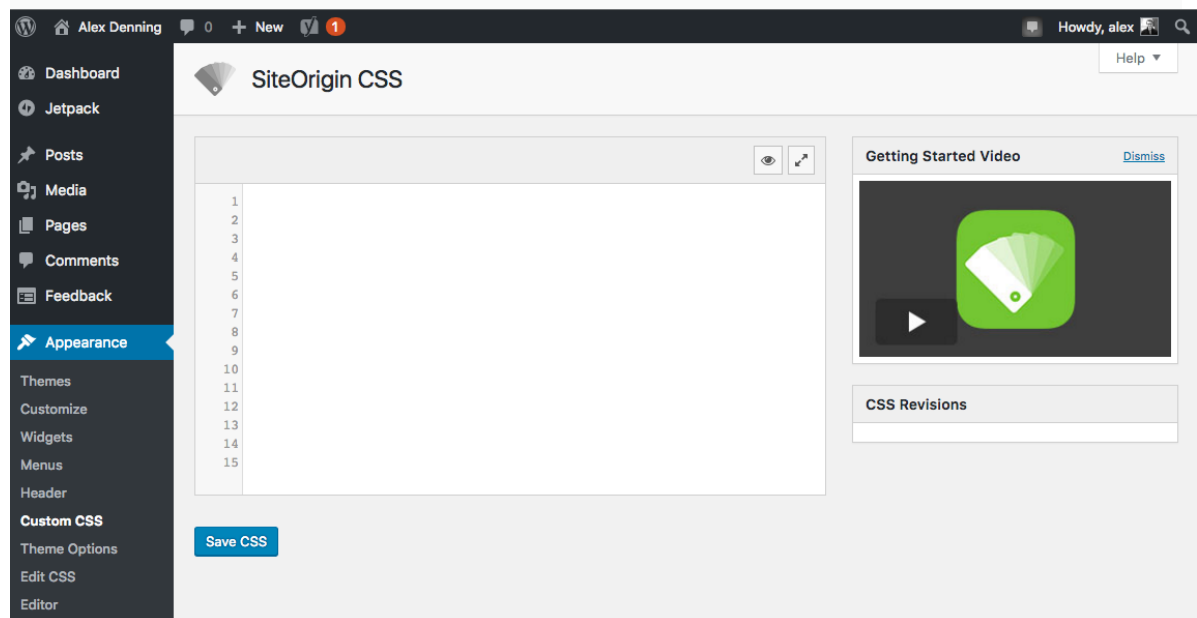
Styleguide adds an interface in the Customizer for selecting the fonts in use on your site. Colors can be edited too but only in themes which add support (so only Pro Theme Design themes which I make and the default WordPress themes support this).

The difference between Styleguide and other font plugins is that it should work with no additional setup. Most other plugins require selecting elements, or setting up css - whereas Styleguide 'Just Works' (most of the time ;)).

Very useful plugin for making basic typography changes to your site.

SiteOrigin Custom CSS

For more advanced changes (but still not needing any coding knowledge) we'll want [SiteOrigin CSS](#). This free plugin lets you edit CSS without knowing any CSS. You'll get a visual editor for editing code with an instant live preview.



It's really powerful and the best way of editing your site without learning CSS, but you will need to spend a little bit of time learning. Once you've mastered it, however, you'll be able to select any element on your site and change its properties - that includes both fonts and colours (and almost everything else).

There's an easy-to-follow video tutorial showing you the basics; it looks complicated but is relatively easy to pick up. You can install the plugin by heading to Plugins → Add New on the WordPress Dashboard, searching for SiteOrigin CSS, installing and activating.

Once the plugin is running you'll find a new menu, Custom CSS, under Appearance on the Dashboard. Head there and you'll find a blank screen.

Get my new free course: Becoming a WordPress Master

I wrote a couple of months ago **I was planning** on setting out freelancing full-time after my graduation (and **a trip to Japan**). This has now all happened and I'm entering my third week working for myself 100% of the time.

I'm excited to be writing regularly again, and over the next couple of weeks I'll be transitioning some content from **BlogBetr** to here and publishing a new post every fortnight. You can **grab a newsletter subscription** if you'd like to stay updated.

One of the first projects I've been working on is a free course for WordPress users to become WordPress *masters* and do everything they want, themselves.

There are a ton of WordPress courses for beginners and a second ton for experts, but nothing that bridges the gap – and that's where **MasterWP** comes in.

You'll then want to first watch the "getting started" video and second click the eyeball to load up a live preview of your site. You can now click on the element you want to edit and change the specifics on the left hand side (for example font size, width, colour).

Edit the values to get the look you want and you'll see a live preview. Press the tick once you're done and the plugin will generate all the code to make your changes live. Save and you're done. SiteOrigin CSS makes customization highly accessible and easy to use.

Properly editing CSS code with Jetpack

If with SiteOrigin you get to know CSS better and want to edit it code-only – or if you've got the skills – then [Jetpack's Custom CSS module](#) is the best tool to use. This allows you to edit the CSS on your theme without touching the code on the underlying theme.

There are a few benefits of using Jetpack Custom CSS over other CSS plugins.

1. **It's secure.** Most plugins just save the CSS to the database and then output it on the front end, however Jetpack makes sure there's no security vulnerabilities included. This might sound like a strange thing to worry about - but it's possible that if someone got access to your database or admin that they would be able to insert insecure code into every page of your site. With Jetpack CSS this code would not be displayed.
2. **It keeps versions of the CSS.** This means that if you make changes and don't like them - then you can go back through the old versions and restore the one that was working well.
3. **It has the option to use LESS or SASS.** These are more technical features for people already proficient in CSS, but they can be really handy if you know what you're doing.

Jetpack is a powerful plugin that we'll come back to plenty of times over this course (we've already mentioned it in this lesson), so it's worth having installed (if you don't already). You'll find info on [getting started with Jetpack here](#), info on [Custom CSS in Jetpack here](#) and [the plugin can be downloaded here](#).

If you're making very simple changes use Styleguide. If you want to make code changes to your site without knowing code, SiteOrigin CSS is the thing to use. If you want to change code, Jetpack has you covered.

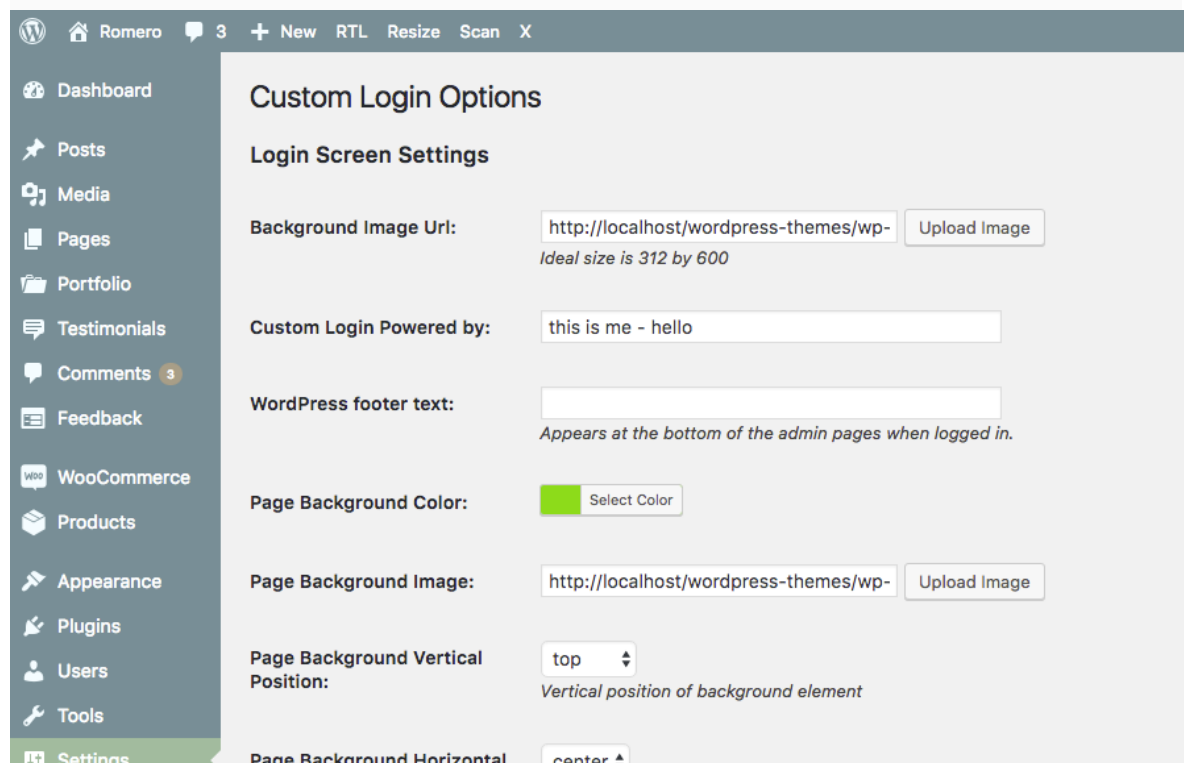
But what about customizations for WordPress that aren't just tweaking styling?

Plugins for extended customization

Finally, let's take a look at a series of plugins which let you make big changes to WordPress easily. The Customizer is good for the basics and CSS is perfect for tweaking styling, but these are the best plugins for taking WordPress one step further.

Custom Login

[BM Custom Login](#) lets you customize the WordPress login screen to match your brand. If you're making a site for clients, or where other users will login, then it's a nice customization touch which can add a layer of professionalism.



The plugin offers you a simple way to edit the images, and colours (and if you know how, the CSS) for your WordPress login page, giving you that extra little

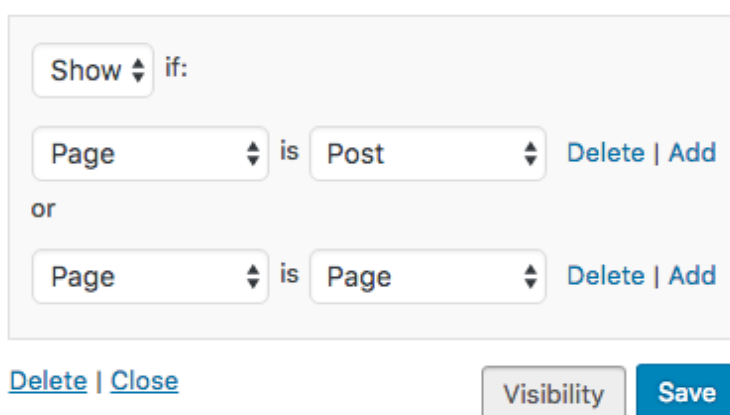
differentiator to make your website stand out. It's really well coded (I made it - Ben).

Get Custom Login - <https://wordpress.org/plugins/bm-custom-login/>

Custom Sidebars - Jetpack Widget visibility

A rather clever feature in the must-have [Jetpack plugin](#) is the ability to selectively display different widgets on different pages.

This can mean you can display some widgets on the homepage only or other widgets on the archives and not blog posts. This is a really powerful and useful way of customizing your site (especially since you can do so much with widgets anyway).

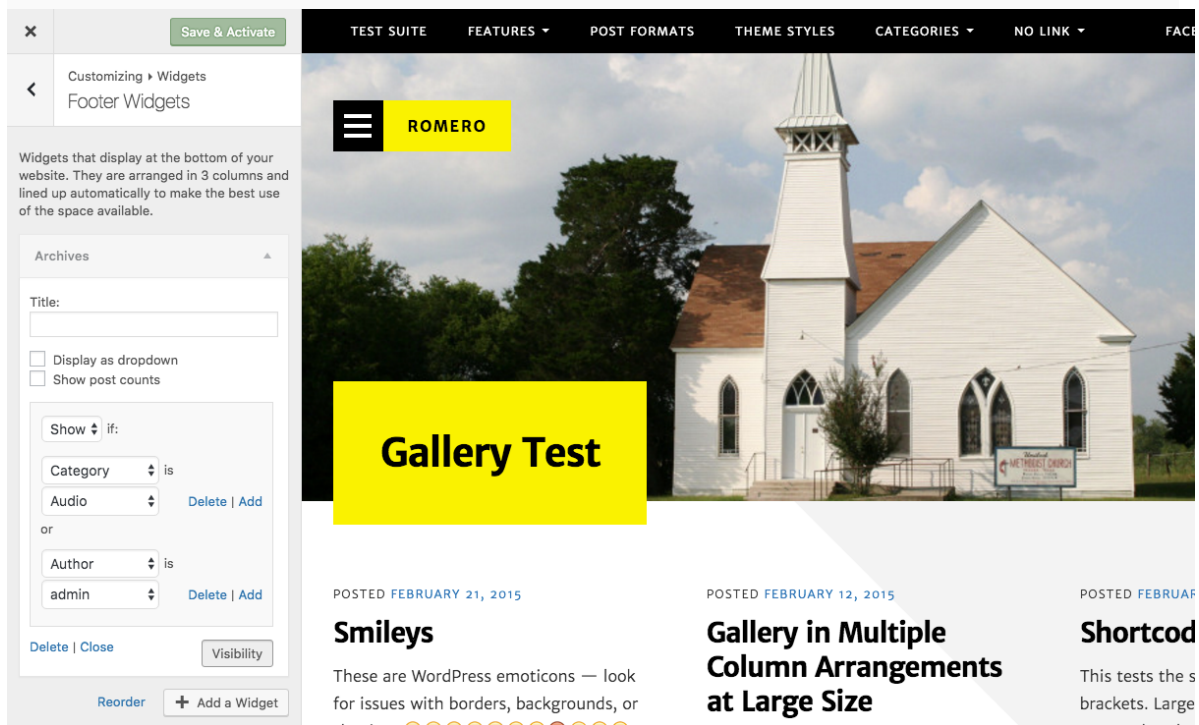
The screenshot shows the 'Widget Visibility' settings in the WordPress dashboard. At the top, there's a 'Show' dropdown with an up/down arrow, followed by 'if:'. Below this, there are two rows of settings. The first row has a 'Page' dropdown, 'is', a 'Post' dropdown, and 'Delete | Add' links. The second row has a 'Page' dropdown, 'is', a 'Page' dropdown, and 'Delete | Add' links. Below these rows is an 'or' label. At the bottom left, there are 'Delete | Close' links. At the bottom right, there are 'Visibility' and 'Save' buttons.

Widget visibility in action.

Widget visibility is incredibly flexible, but you have to be careful since complex rules can be both slow, and very confusing to work out. These are the steps for using it:

1. Head to Appearance → Widgets on the Dashboard.
2. Drag and drop widgets to the widget areas you want them to display.
3. Expand the widget with the visibility you want to change.
4. Click "visibility" (next to save).
5. Choose "show" or "hide" depending on your preference.
6. Add rules for the posts. For example, to display a widget only on posts and pages select "show" → "page" is "post" (did warn it was confusing).
7. Add extra rules as required. To display the widget on posts *and* pages you'd add a second rule of "page" is "page".
8. Save and test when you're done!

Widget Visibility is a module built into Jetpack. You can activate it by heading to Jetpack's settings, hovering over and pressing "activate".



Get Jetpack - <https://wordpress.org/plugins/jetpack/>

Unique Header Images

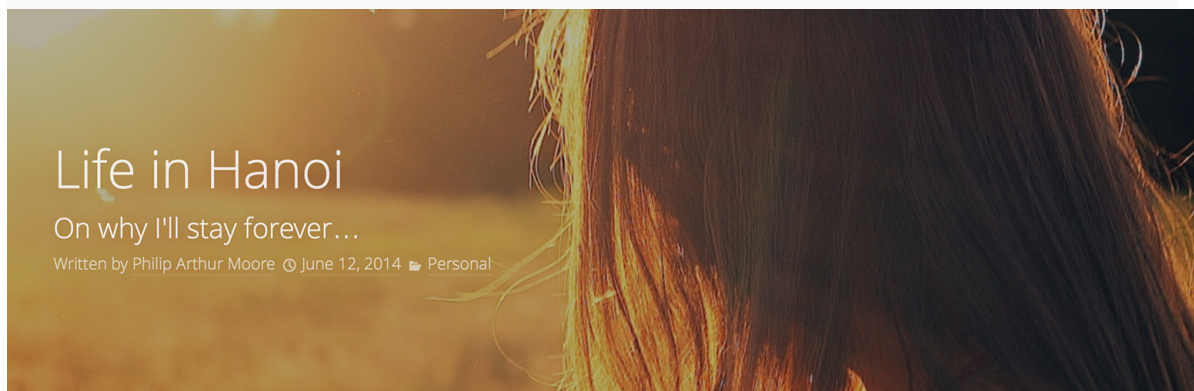
As we saw in the Customizer, you have the option to upload your own header image. [Unique Header Images](#) takes this further, giving you the option to upload a different header image for each post/page/category. Handy – and unique – plugin.

The plugin is easy to use. You'll find an extra box when editing posts and pages with the option to upload a header. Add one in and you're done. For categories, head to Posts → Categories and edit the category you want to add a unique header to.

Get Unique Headers - <https://wordpress.org/plugins/unique-headers/>

Add Subtitles

Subtitles are a nice way to add extra information about a blog post - and the subtitles plugin is a simple way to add this. It adds a 'subtitle' box to your post edit screen - where you can add the extra message. It then automatically inserts the subtitle beneath your blog post.



I've used this on a number of sites and themes and out of the box it tends to work surprisingly well, integrating nicely with a range of designs.

Get Subtitles - <https://wordpress.org/plugins/subtitles/>

Full customization control

Using plugins to customize your site is a relatively risk free method for making changes. If anything goes wrong you can revert your changes – or just deactivate the plugin.

Most of these plugins have visual interfaces for editing things, which makes it even easier for you to get started. Here's a recap of the tools we've mentioned:

- The Customizer, built into WordPress, for basic changes.
- [SiteOrigin CSS](#) for editing CSS without knowing CSS.
- [Jetpack's](#) Custom CSS module for editing CSS code.
- [BM Custom Login](#) for making a custom login page.
- [Jetpack](#) (again) for Widget Visibility.
- [Unique Header Images](#) for different header images across your site.
- [Subtitles](#) for subtitles on your posts.

The plugins we've introduced you to here don't let you do everything – you'd need to be using child themes to do that – but they let you do 80% of the changes you'd want to make, with 20% of the effort required. They're the perfect starting point, and as you use them you'll only get more proficient.

Make your site load really quickly

A fast website is very important. 47% of people expect a site to load in less than two seconds and 4/10 of your visitors will abandon your site if it takes more than three seconds to load ([source](#)). A slow website is something that's just not acceptable to visitors.

There's more data. [One usability study](#) (that is dated, but the theory remains legitimate) suggests users will tolerate a small delay, but will be equally unsatisfied with a medium delay as a long delay. You might not think of your site as "slow", but even if it takes a bit of time to load visitors may perceive it the same way as they would an exceptionally slow site.

There's a way of avoiding visitors leaving because your site is slow: have a really fast site.

This is hardly groundbreaking analysis, but part of becoming a WordPress master is making sure you can work with any WordPress installation – even on poor quality shared hosting – and make it load really quickly.

Fortunately, it's not especially difficult to make any WordPress site load quickly if you know how. We'll cover here everything you need to know.

Analyse your site's current loading speed

The first thing to do is find out how fast your site is already. We'll use two tools to profile your site.

[Pingdom Tools](#) is a great (free) tool that will do this for you and tell you:

- How long your site takes to load (lower is better)
- The PageSpeed score assigned by Google (higher/100 is better – you can [get the full test here](#); we'll come back to this later but for now the headline score will do)
- Your page's size in MB (lower is better)
- How many HTTP requests your site makes (fewer is better)

Test your site and make a note of the results. Screenshotting the page is probably easiest.

The second tool we'll use is [P3](#), a plugin which measures the loading time/impact of other plugins. The plugin is made by web host GoDaddy but is free, works on any host and does the job nicely.

Install the plugin and run the profiler. Head to the advanced breakdown and you'll be able to see what proportion of your load time plugins are responsible for. Again, make a note of this and we'll look at ways to reduce that impact later.

We now know what your site's current performance is and can get on with improving it.

Choosing a caching plugin

You need a caching plugin. Caching plugins work by generating static HTML files for your site when can be loaded much faster than a page which needs to be generated each time. The gains from doing this are very significant and if you only do one thing to improve your site's performance, this should be it.

Conventional WordPress wisdom says choose between [W3 Total Cache](#) and [WP Super Cache](#). Historically there's never been much difference between the two, although Total Cache did bundle more features for a time.

As a *WordPress Master* you're interested in the best option, not conventional wisdom. Here's where you can get an edge: a bit of digging shows Total Cache has been rarely updated in the last 24 months whereas Super Cache has been worked on by [Automattic](#), the quasi-WordPress-parent-company. In that time Super Cache has added a lot of new features; because of these new updates from the people who know WordPress best, you'll want to be using Super Cache.

If you're working on a site where you can make no compromises on page load speed (think eCommerce) then the best free plugin won't cut it; you'll want the best plugin outright. [These tests](#) show [WP Rocket](#) is the best option available.

You'll experience an immediate improvement in page load time just from installing a caching plugin, ticking all the boxes in the settings that are recommended and then re-running the Pingdom test. We've seen a 30%+ improvement in loading time on sites just from using the plugin.

Unless you're using non-shared hosting you won't have anything other than the recommended options available. If you do have extra options you can use, check out these notes on [how to get the most out of them](#).

Reduce page size

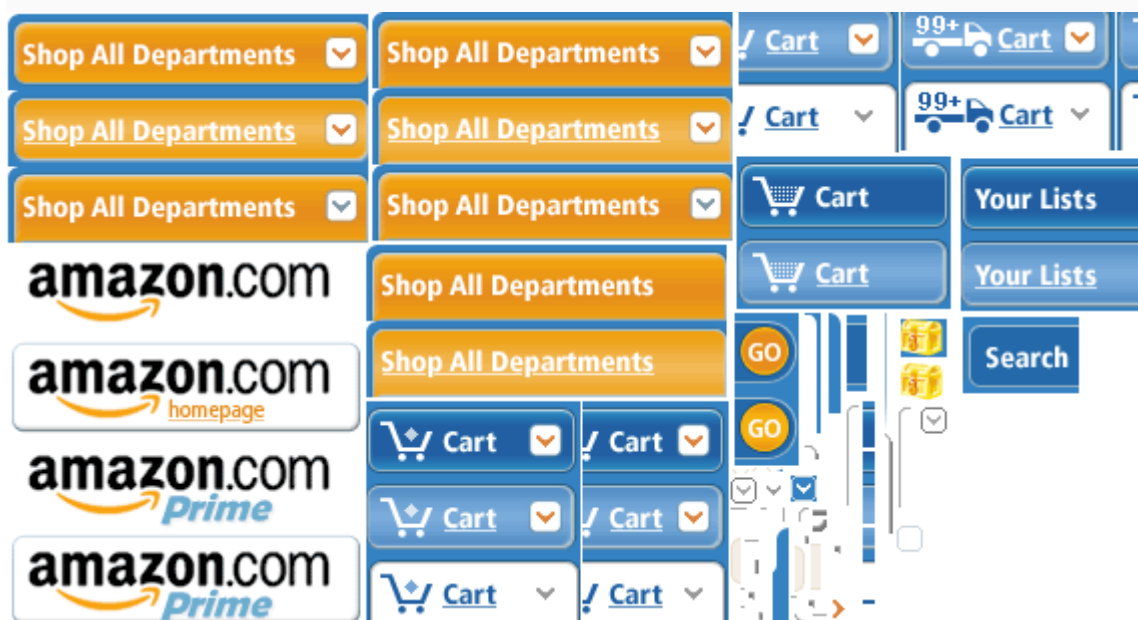
With caching sorted, the next target for making your site load really quickly is the page size. If less stuff (ie fewer MBs) is being loaded, your site loads faster. Simple, right?

An easy way of reducing the size of the files your site loads is by reducing the amount of characters in them. This is a process known as “minification” and it removes as many characters from source code as possible without changing its functionality.

Minification is a process that’s ripe for setting-and-forgetting whilst making a big improvement to your site.

[Autoptimize](#) has a very-difficult-to-pronounce name but handles minification excellently. Install the plugin and tick all the recommended settings and you should immediately see a fairly dramatic reduction in page size – we’ve immediately achieved a saving of around 30%. That won’t necessarily translate into an equivalent reduction in page loading time but it should make a reasonable difference.

CSS sprites work by combining all the images on your site into a single image, and then only displaying the segment of the combined image that fits the image which displayed originally. It’s then faster to load a single large image than lots of smaller images. Here's an example of Amazon's CSS sprite set up:



CSS sprites have historically been fiddly and time consuming to make. [SpriteMe](#) is a bookmarklet which will create a sprite for you and generate the code needed to put it into use, but it’s seven years old and it seems implausible nobody has found a better solution in that time. It’s also fiddly to use with WordPress as it requires editing the theme files directly, which will likely be overwritten every time you upgrade your theme.

Fortunately there’s a better although mildly temperamental solution available. The catchily named [CSS Sprite for Google PageSpeed plugin](#) will generate CSS

sprites for all the images on your site on the fly. The plugin author describes it as “in beta stage” but our testing found no problems. The plugin is free but uses the author’s external cloud service, so don’t rely on it forever. For now, though, it’s the only plugin available and does a good job.

Once you’ve made your page smaller, you can make it faster by utilising a “content delivery network” or CDN. We’ll cover these later on, but a quick tip is to install the Jetpack plugin, and enable Photon. Optimise for Google PageSpeed

Google PageSpeed rates sites by their theoretical possible loading time rather than their real-world speeds. PageSpeed [is a factor](#) in Google’s search algorithms. This is handy for you as it means you can just tick off a bunch of items and get a high PageSpeed.

The initial Pingdom Tools test gave you a PageSpeed score but you can [get the full thing from Google here](#).

Most of the suggestions Google makes will be covered by the other optimisations we’re making, but there’s one which you won’t cover without specifically looking for it.

Google requests you “remove query strings from static resources”. You can do this with another set-and-forget plugin, “[Remove Query Strings From Static Resources](#)”. It does literally what it says on the tin. There are no settings.

Install the plugin and you should see an immediate improvement in your PageSpeed score.

Analysing plugins

You’ll often read that “having too many plugins installed is bad”, but we ran the P3 test earlier which established this clearly isn’t the case. It’s not how many plugins you have installed that’s important, but what they’re doing.

This is good to know; indeed, all we’ve really done to make your site load really quickly is install a bunch of plugins. Run the P3 test again, however, and you’ll find the plugins we’ve added are very low impact. It’s likely you have a couple of plugins which make up the vast majority of your loading time. These are typically things like:

- Jetpack by WordPress.com
- Sliders or galleries
- Email capture

The aim, therefore, is not to arbitrarily reduce the number of plugins installed but look at how the plugins which have a disproportionate impact on loading time are set up and what can be done to make them load faster.

Most resource intensive plugins will have some options which you can utilise to reduce their impact. You want to be using these. If you have a high-impact plugin with no loading options you may wish to consider an alternative plugin.

If you have options to only load the plugin on specific pages where it's used then utilise those or if it's possible to disable features you're not using then also do this. Jetpack, for example, has a lot activated by default, so there's likely stuff you've never looked at and won't miss. Head to Jetpack's settings and deactivate the features you're not using.

A quick note on Jetpack and page speed. A lot of Jetpacks slowness only occurs when a user is logged in as an admin on the site. A regular site visitor will not see the slowness since a lot of the functionality is not available for them. This is something to consider for other plugins as well (such as WP SEO), as P3 is likely overestimating their load impact.

Try out the different performance options available with your high-impact plugins and run [Pingdom](#) again after making changes to see what makes a difference. If you're not able to move the needle then you may wish to consider using an alternative plugin.

Plugin recap

Put all these measures into place and you're going to have a significantly faster website. In half an hour you should be able to take a site on any host and get fast loading times and a high PageSpeed score with relatively little difficulty.

All the plugins we've recommended are free and can make a real difference to your site's loading time.

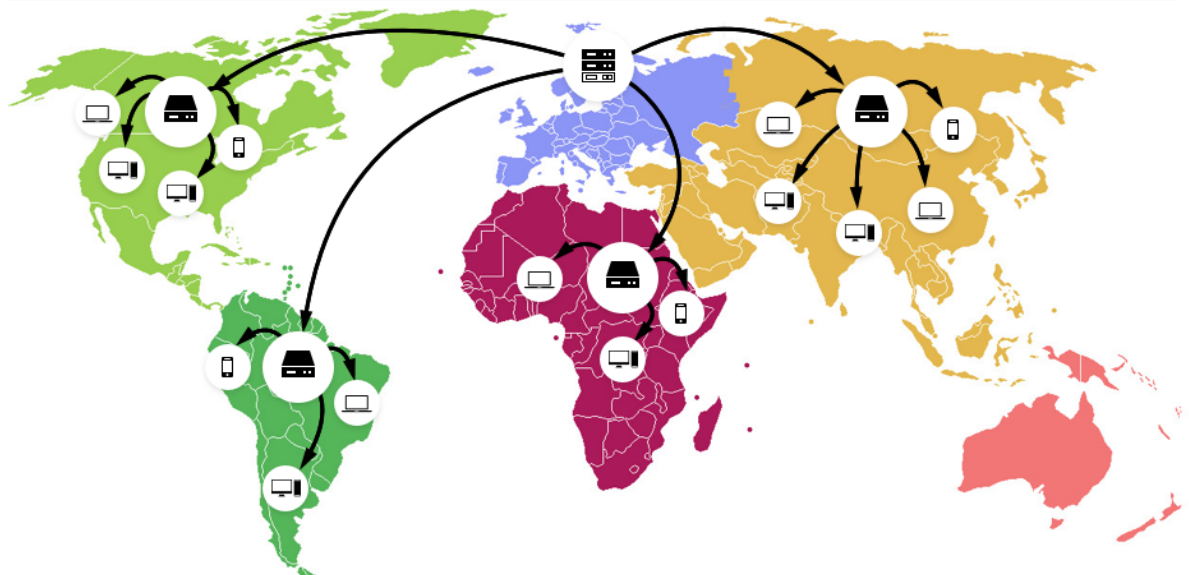
Using Photon and CDNs for Extra Speed

As we established yesterday, site speed is really important for a successful site.

One of the biggest hurdles you'll face when trying to make your site faster is images and other static files such as JS and CSS. Images especially can be large files and slow to load (especially on cheap shared servers). To improve this you can use a CDN.

What's a CDN?

CDN stands for Content Delivery Network. A CDN is made up from collection of servers that are spread around the world. They are generally optimised for serving static content quickly.



The way they work is for there to be an 'origin' server that stores your static content (images, javascript, css etc) and distribution servers that give the content to the users. The origin server is the parent server that stores the master copy of the images (in our case it will be the server that is running WordPress).

When an image is requested from the CDN it first calculates the server that is geographically closest to you. That server checks to see if it has the requested image. If it doesn't it loads it from the origin server. If it does it displays it.

The benefit of this is twofold:

1. Servers are optimized purely for serving images.
2. Servers are located as close to the user as possible.

These things combined mean that the image does not have far to travel and so can get to your users quickly.

This all sounds quite complex – and to be fair it is. Thankfully, you don't have to deal with the technical aspects of CDNs – you can just take the benefits. There are a number of services which handle everything for you and make it easy to get a speed boost from CDNs on your site.

First we'll look at what is by some distance the best WordPress CDN solution.

Photon makes your life really easy

Photon is a CDN that's really good. It's also free, which is handy.

There are a few reasons Jetpack's Photon service is good and we've separated it from the other CDN plugins:

1. It's part of the [Jetpack](#) plugin. This is a plugin we've recommended in previous lessons and you're probably already using. Just setup Jetpack and enable Photon and you're CDN ready. No more work needed. **This process is by far the simplest to set up.** Many of the other CDNs need some sort of technical changes to your server setup.
2. **Photon will automagically optimize the images for you.** Most CDNs will simply serve the images you give them, but Photon makes those images smaller. In fact they use a file format called webp. [Developed by Google](#), webp is *the best* compression algorithm around. Currently it's supported by Chrome and Opera web browsers – and Photon knows this, so it only serves webp images to Chrome and Opera users. That accounts for [nearly half of all browser usage](#), though, so it's not to be snubbed.
3. A further, more advanced function of Photon is that you can apply filters to images, and crop and resize them. The cropping and resizing Photon does by default (just use the standard WordPress `add_image_size` functions), and for more technical users you can add filters to change how the images look.

To set Photon up is the same as with any other Jetpack module. Head to Jetpack → Settings, scroll to find Photon and press "Activate". That's literally it (told you it was easy).

There's information about [Photon here](#) and [API documentation here](#).

Get Jetpack - <https://wordpress.org/plugins/jetpack/>

Alternative CDN options

So why would you want to use a CDN when Photon is so great? Three possible reasons:

1. **Privacy.** Photon is run by Automattic (the company that owns wordpress.com) and some users are suspicious of them. A bit tin foil hat? Maybe, but Automattic is a commercial company offering a free service and it's presumably not out of the goodness of the hearts.
2. **The cache does not invalidate.** This means you can't delete an image. If you want to replace an image you have to upload it with a new name.
3. **Photon only supports images,** so if you want to serve your Javascript and CSS from a CDN then you would need to use a separate service. The gains from doing this are marginal, though, so unless your site is really popular there's little reason to worry about this.

Unfortunately, try as we might, we've yet to find a CDN that is as easy to set up as Photon.

This isn't necessarily reason to fret, however. Loading images from Photon is going to do the majority of the work on your site's loading speed. The gains from loading other static files from a CDN are marginal, and unless your site is really popular it's unlikely make a difference such that it's worth the effort.

If you still want to go ahead with your CDN setup then below are the best commercial options for you to choose from.

The best option: Cloudflare

[Cloudflare](#) are a very good option. They do a lot more than work as a CDN, including enhancing security, so are well worth a look. Cloudflare are listed first as their generous free plan will be sufficient for most users (if you need more, pro plans start from \$20/site/month).

If you want to chase those marginal gains and pair Jetpack with another CDN service, Cloudflare is the one to go with.

The only downside is the setup is nowhere near as simple as Jetpack's. You'll need to sign up for an account and follow the instructions to update your DNS records. These may take a couple of hours upto a day to propagate across the web. Once that's done you can choose the free account and you'll be setup. WPBeginner [has a full tutorial](#) on the process.

The other commercial options

[Incapsula](#) provides a similar service to Cloudflare, giving you both security and speed benefits. Their free tier is quite generous and it's one of the easier services to setup (they have [a WordPress plugin](#) to ease integrationm however it's been a while since it has been updated).

The down side to Incapsula is that its premium plans are more expensive than Cloudflare's so if you should want to get some of the more advanced features then

you'd have to pay quite a lot more. Pro plans start from \$59/site/month.

[KeyCDN](#) is a relatively cost effective solution for powering a WordPress site, and they have [a WordPress plugin](#) that they have developed and maintain, so they may be worth a try. There's a free trial but no free plan. Paid plans start at US\$0.04 per Gigabyte, so you would need a very popular site to pay more than US\$1 a month.

[MaxCDN](#) is probably the best known. It's well used by some of the top blogging sites and frequently mentioned when you speak to other WordPress developers. You can add support for MaxCDN through plugins like W3 Total Cache ([see documentation here](#)). Prices start from \$9/month.

[Fastly](#) is a CDN service used by some of the bigger names in news media - such as the UK's Guardian newspaper. They recommend the use of the [Purgely plugin](#), which is developed by Conde Naste - for integrating with their service. They are not the cheapest CDN around but they have a trial option that gives you \$50 credit, so if you're serious about speed they may be worth trying out.

High End Options

If you want more control – or really know your stuff – then it may be worth considering using [Amazon Cloudfront](#) or [Google's Cloud CDN](#). They offer a lot more fine grained control and more precise pricing - however they also need a lot more technical knowledge.

It's probably worth paying a little more to have someone take care of all the technical challenges for you.

The best CDNs for the fastest site

The best free option is to use [Jetpack's Photon](#) for images and [Cloudflare](#) for everything else. They are relatively easy to setup and combined will deliver a significant speed boost for your site. A faster site means happier visitors and much happier search engines :)

If you have a large site you'll want to opt for premium service which offers more control. The services we looked at offer a wide range of features for all budgets and site sizes. The whole point of a CDN is to serve your content to your users more quickly. These are set-and-forget services, so it's well worth taking the time to make use of them and get your site loading faster.

A CDN is just one part of the puzzle in making your site quick. Installing a caching plugin, optimizing content, choosing a fast theme and selecting a good webhost all

contribute (as we covered yesterday). With today and yesterday's lessons, you should be well equipped to make *any* WordPress site fast.

The endgame is to combine optimization with the power of a CDN. They're a powerful mixture for a blazingly fast site.

Harnessing WordPress, WooCommerce and email marketing to sell more products

Effective email marketing is the one tool you can use to dramatically increase your sales that you're probably not utilizing. It's something that the majority of companies know the value of: email generates more than 10% of sales for 55% of companies, and 42% of businesses say email is one of their most effective lead generation channels (studies 2014 and 2012 [respectively](#)).

Email is an essential tool in the toolbox of any online business.

The good news is if you're rocking WordPress and WooCommerce, it's straightforward to making use of email and put a rocket booster under your sales.

The even better news is in this lesson we'll equip you with the tools you need to get going. We'll start off by covering the best ways of capturing email signups with signup boxes and then move on to look at how you can integrate with your WooCommerce installation.

Capturing email signups

The basic tenet of email marketing with WordPress is capturing email signups. You need signups so you have an email list to market to. For this you'll need an external service such as MailChimp ([free](#)) or ConvertKit ([expensive](#)). It's important to use an external service so the emails you send definitely arrive (if you do it yourself they may get stuck in spam filters). If you don't already have an account with one of these services, sign up for one now.

The easiest way of capturing signups is through signup boxes. There are two easy ways to add these to your WordPress site:

- Signup form widgets
- Signup form shortcodes (with a custom plugin)

We'll work through each of these to equip you with versatile solutions.

Using signup form widgets

Widgets are perfect for adding signup forms. Sadly there's no free, do-it-all email form solution. This means you need to be flexible: there are some good free plugins for specific email providers or a straightforward DIY solution (which can draw on some customisation knowledge from earlier in the course).

If your theme has a signup form built in then this should be your preference as the design will fit tightly with the rest of the theme. Moving forwards we'll assume that you either don't have an option built in to your theme or aren't using it.

Free signup form plugins

Plugins are the easiest way to add signup widgets (and hence forms) to your site. There are a couple of free plugins which can add a widget for you. You'll want to find one which works with your email provider and has the design options you want. These are ones we like:

- [MailChimp for WordPress](#): despite the name, not an official plugin. Generally regarded as the best signup option if you're using Mailchimp. Some addons available but the premium options below are better.
- [WP Subscribe](#): supports Mailchimp (but the above is better), Aweber and Feedburner. Minimal customisation options but the widget looks pretty good.
- [Newsletter Sign-Up](#): the predecessor to MailChimp for WordPress, this plugin works with CampaignMonitor, ConstantContact, YMLP, Aweber, iContact, PHPList and Feedblitz. If you're using any of those services (bar Aweber) this is your best free option.

All these plugins are free so try them out and see which works best for you. You can install any of these straight from your WordPress Dashboard by heading to Plugins → Add New and searching for the plugin name. Install and activate!

Get your API key here.'. At the bottom is a blue 'Save Changes' button." data-bbox="125 511 870 769"/>

General Settings

MailChimp returned the following error:
Missing API key..

MailChimp API Settings

Status **NOT CONNECTED**

API Key

The API key for connecting with your MailChimp account. [Get your API key here.](#)

Save Changes

You'll then need to add connect your account. For MailChimp for WordPress and Newsletter Sign-Up you'll have a new menu item added. Head to the general settings and connect your account by creating a new API key. You'll also want to check out the other settings to connect the plugin to a specific email list. For WP Subscribe you can connect your account on the widget.

You can use any of these new widgets by heading to Appearance → Widgets. You'll find a new widget added; drag and drop to any widget area to try it out.

Customize options as you want them, save and you're done!

Final step is to look at your site and enjoy the lovely new email signup form :)

Premium signup form plugins

Customer conversion is big business and there are a lot of premium plugins available which promise various levels of signup utopia.

Just be aware that these are marketing products and the most successful options are generally run by people really good at marketing. Bear in mind it may be possible they're overstating the benefits.

These premium options generally get you extra features such as exit-intent popup boxes (the ones that show up when you're about to leave the page), more design options (so you can better customize) and A/B testing (test two versions of your signup boxes and see which gets better results).

If you're selling a product or acquiring signups has a decent monetary value to you then these plugins are probably worthwhile.

There are a ton of these to choose from, but you'll find the best options below (pricing is for mid-level traffic website with comparable features you'd want at annual cost):

- [OptinMonster](#) (\$99): decent option and the admin panel is well laid out, but limited design customisation options and can get expensive if you need to upgrade.
- [SumoMe](#) (\$468): feature rich but gets very expensive very quickly (seriously it's so expensive compared to other options). A lot is available on the free plan, but you'll need to pay to remove branding. Probably not worth it.
- [ThriveLeads](#) (\$67): can be quite difficult to use but if you can deal with that, offers decent selection of features at a reasonable price.
- [Ninja Popups](#) (\$32.50): we've not used this, but it deserves a mention as a cheap option with plentiful designs to choose from. Worth taking a look at.

The competitive nature of the premium signup plugin business means all these options (except Ninja Popups) offer money-back guarantees, so there's no harm in trying out different options and sending back if they're not for you.

All come with extensive documentation and support for installation.

Easy build your own signup form!

There's a final option which doesn't involve any plugins. WordPress has a text widget which allows you to enter your own HTML to any widget area. If you're using an email service offering pre-designed signup boxes such as [ConvertKit](#) you can

just copy the box code and paste into a text widget. Save the widget and you'll have it on your site immediately.

There'll likely be a line of code to load a stylesheet or some inline styles so the box looks nice. You might want to copy these styles into one of the custom CSS solutions we discussed in the lesson on customization (Jetpack's CSS Editor is easiest). Having fewer CSS files to load will make your site load faster. Once you've copied the CSS over, delete the reference in your text widget to stop the styling loading twice.

Customizing your widgets

Nearly all the plugins we've looked at so far utilise widgets. Widgets are perfect for adding signup forms to a site. Most themes have lots of widget areas, so you can get your signup boxes displaying where you want and by default they should look pretty good too!

The *WordPress Master* obviously wants to do more, though. It'd be really handy to only display signup boxes on specific pages. For example:

- Display a sidebar signup widget everywhere but posts.
- Display an after-post signup widget.
- Hide all signup widgets on the contact page.

This is easy to do (and you may remember we've mentioned this in a previous lesson – this is a practical implementation). The trick here is **conditional widget display**. This is another feature built in to the [Jetpack plugin](#). We've talked about Jetpack so much we'll assume you've got it installed by now; you activate by going to Jetpack → Settings and activating Widget Visibility.

Now when adding widgets (under Appearance → Widgets) you're going to find an extra button: Visibility. Press this and you'll have the option to add conditionality for how your widget is displaying. You can choose to "show" or "hide" depending on the parameters you set. You can chain variables by pressing "Add".

The screenshot shows the 'Show if:' configuration for a widget's visibility. It features two rows of conditions separated by an 'or' label. The first row shows 'Page' selected from a dropdown, followed by 'is', and 'Post' selected from another dropdown, with 'Delete | Add' links. The second row shows 'Page' selected from a dropdown, followed by 'is', and 'Page' selected from another dropdown, also with 'Delete | Add' links. At the bottom left are 'Delete | Close' links, and at the bottom right are 'Visibility' and 'Save' buttons.

Note confusingly you'll find posts under "pages". For some reason the terminology is different here to the rest of WordPress.

Integrating email marketing with WooCommerce

Collecting emails throughout your site is one part of the email marketing puzzle. Once you've got emails you can start sending out notifications of new products, updates and promotions.

You should see some positive results straight away, but you'll get even better results if you can personalize your emails more. You want to be able to do things such as:

- Offer exclusive discounts to existing customers.
- Entice those who have signed up but not purchased with their own sales emails.
- Make sure you're not pitching to customers who've already purchased the product you're talking about.

With WordPress and WooCommerce all this is possible and pretty straightforward. You're only wanting to do a couple of things:

1. Add customers to your email list.
2. Make sure your email list knows what was purchased.
3. Segment purchases and customers on your email platform so you can send different emails to each.

This means you just need to connect WooCommerce and your email marketing platform. Again, there are a number of plugins which will let you do this and they're straightforward to use.

[Google Analytics](#) | [ShareThis](#) | [ShareYourCart](#) | **MailChimp**

MailChimp

Enter your MailChimp settings below to control how WooCommerce integrates with your mailchimp lists.

Enable/Disable ☒ Enable MailChimp

Event Occurs 
When will customers be added to lists?

API Key
[Login to mailchimp](#) to look up your api key.

Main List 
All customers will be added to this list.

Group Name
Optional: Enter the name of the group. Learn more about [Groups](#)

Groups
Optional: Comma delimited list of interest groups to add the email to.

Double Opt-In ☐ Enable Double Opt-In
If enabled, customers will receive an email prompting them to confirm their subscription to the list above.

[Save changes](#)

The informatively named [WooCommerce MailChimp plugin](#) is the best option if you're using MailChimp. This unofficial plugin is free and lets you have full and deep integration between the two platforms. As with the widgets, installation is straightforward but requires you create an API key. Make sure you set up interest groups so you can segment customers in MailChimp.

Other email marketing services typically have a similar plugin available. A search for WooCommerce + email service name will typically find you the plugin to use.

Once you've integrated purchases with email, you're in a position to massively extend what you can do with your email list – and the potential to generate revenue.

Going further with email

We're not quite done yet. You also have the option to go a lot further, but this involves more setup and paid products. These are some ideas to get you thinking

about what else you can do with email and WooCommerce:

- Reduce cart abandonment: automatically contact customers who've entered their email but failed to complete a purchase. This can [retain a huge number of sales](#) you would otherwise lose.
- Retain inactive customers: if and when customers go a long time without a purchase, get in touch and offer them the right deal to get them buying again.
- Make full usage of receipts: these are emails with huge open rates. Start using them as an opportunity to sell again to customers.

Take a look at [Receiptful](#), [AutomateWoo](#) and [ReMarketly](#) for the advanced options.

Bonus: build your own signup form shortcode plugin

If you're using your own code in a text widget there's a useful bonus you can apply: building your own custom shortcode plugin so you can drop a signup form into any post or page.

Building a simple custom plugin is actually really easy. For this you'll need FTP access to your site. You'll be able to get this from your hosting provider.

Log into to your FTP app and navigate to /wp-content/plugins/. Create a new file and call it signup.php. Edit this file and paste in the code below, adding in your form code where marked:

```
<?php
/*
 * Plugin Name: Signup shortcode
 * Version: 1.0
 * Author: masterwp.co
 */

function form_creation(){
?>

<!-- paste your form HTML here -->

<?php
}
add_shortcode(signup, 'form_creation');
?>
```

Save the file and navigate to Plugins on your WordPress Dashboard. You'll see the new plugin Signup shortcode is available to activate – do that, and you're done! You'll now be able to use the shortcode [signup] to drop your form into any post

or page. This is really useful for creating contextual signup forms which can increase your conversion rate.

One final tweak here: you may want to customise the styling of your in-post form. You can do this using the CSS customisation techniques we discussed in a previous lesson (see it [here](#) if you missed it). Setting width: 100%; and text-align: center; will probably do the job here.

Collecting so many emails

The techniques we've discussed here offer some powerful ways of connecting with visitors and turning them into loyal readers and subscribers. Especially with the premium plugins we looked at and the A/B testing they offer, there's scope to be exceptionally effective at this.

Being really good at email marketing is a hugely valuable skill and these are the tools which can get you 80% of the way there with 20% of the work. Put these into practice and there'll be a huge help on your journey to Mastering WordPress.

Your crash course in WordPress security

The security of your website is exceptionally important. WordPress powers [26% of all websites](#) and that doesn't make it vulnerable per se, but makes it a target.

WordPress is not impenetrable, but it's safe to assume the latest version is secure and any vulnerabilities that do exist will be fixed with updates within hours or days.

This means **the best way to keep your site secure is to make sure your WordPress installation is always up-to-date.**

Plugins and themes are a separate issue; there's not the same level of vigilance regarding their security and updates may be slower. The best practice is still to keep everything up-to-date all the time, but if a theme or plugin isn't updated to cover security vulnerabilities, having the latest version of an insecure theme or plugin is of little help.

The *better* practice, therefore, is to only use themes and plugins you can reasonably assume will be continually updated. A good rule of thumb is check when the last update was. If it's more than 6-12 months old, that's generally an indicator of infrequent updates. If the last update is more than 18-24 months ago, the plugin has likely been abandoned and should be avoided (although well-made plugins with simple functionality can be "old" but safe). If you're using a premium theme without automatic updates make sure you're signed up for notifications of available updates.

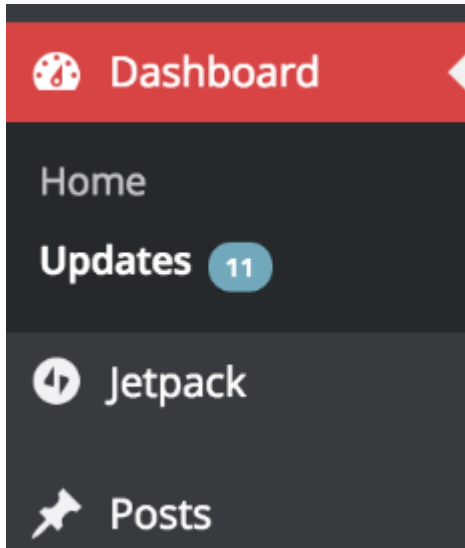
NB: A particular problem to be aware of is themes bundling plugins. Themes recommending you separately install plugins is fine, the dangerous thing is themes including plugins in their code. This means you can't get updates for the plugin without updating the entire theme – and it's unlikely there'll be a new version of the theme for every single release of the plugin.

The danger of this [was highlighted](#) by a critical vulnerability showing up in a plugin that was frequently bundled with themes on ThemeForest. The plugin needed updating to fix the problem. Three months later, over 100,000 sites were yet to be updated, with over 11,000 of those attacked.

Security is something to take very seriously. Keeping everything up-to-date is the basic requirement here. We'll run you through how to do this, common issues you'll run into and how to automate the entire process.

Updating WordPress, themes and plugins

WordPress makes it very easy to update your installation, themes and plugins all from one place. If any updates are available, a number will show on the WordPress Dashboard menu, as you'll see below:



Click through and you'll be able to see what can be updated. WordPress will automatically update minor releases of WordPress itself and themes and plugins in exceptional circumstances (if the WordPress.org team regards an update as necessary to fix a known security vulnerability). This means there'll likely be some updates to do. We'll cover how to make everything automatic later but walk you through manual updates first.

Before you update anything you'll be warned you need to do a backup first. This is true, but it's a pain to do and it's one of those things everybody knows they should do but doesn't. Instead of doing ad-hoc backups every time you do updates, it's better to have reliable automatic backups at a frequency that makes sense for however often you update your site (if you have frequent updates, once a day is good – infrequent, fortnightly or monthly is fine).

You can update WordPress by pressing "Update Now". For themes and plugins you'll want to tick "Select All" and then update. WordPress will handle the update and there shouldn't be any issues.

Troubleshooting common issues

There are two issues you'll likely come across:

1. Stuck in maintenance mode

When you run an update WordPress puts your site into “maintenance mode”. If for whatever reason the backup won't complete your site will be stuck in this mode, complete with the homepage being replaced with a message stating “Briefly unavailable for scheduled maintenance”.

Not pretty.

This is easy to fix. Log in to the root of your site via FTP and you'll find a new file: .maintenance. Delete this file and you'll get your site back. You can then run the update again and it usually works a second time.

2. Automatic updates not happening

The second common issue is automatic updates don't happen. If you're using a “managed” WordPress host that handles things like updates for you, this would explain it. Your host may just take a short time to check the new version for any bugs or issues and then apply the update a short time later. Contact support if there's any serious delay.

If you're expecting automatic updates that just aren't happening then install the [Background Update Tester](#) plugin. This will tell you what's not working and why, with steps to fix it. Your host's support will likely be able to help you out here (especially if they offer “WordPress compatibility”, which most hosts will).

Another possible cause of automatic updates not happening is if WordPress detects you are using Git or Svn to manage your website. Since these are quite technical things, WordPress assumes that people using them will update their site when required. If you're using a version control system then make sure you update your local version and deploy to your website.

Automating updates

Manual updates are fine, but a WordPress Master prefers to automate what doesn't need to be done manually. WordPress can do this for you (and we'll get to that), but there's an option that's easier to set up and scales a lot faster (so better if you're dealing with lots of sites).

Using ManageWP

[ManageWP](#) is one of many WordPress management platforms but its free account offers much more than the competition, including automatic updates and once-a-month cloud backups (NB ManageWP has [recently been bought](#) by GoDaddy, but they say the free accounts are here to stay for now).

They recently released a new dashboard, Orion, which is generally much easier to use, but tragically gets rid of the automatic updates feature seen in the old version. You used to be able to set-and-forget ManageWP to handle all updates across all your sites, but no more. I'm told it is a feature that's coming back, but there's no ETA at present.

It's worth having ManageWP installed anyway to keep an eye on your sites, and hopefully it'll soon again be the one-stop solution to site updates. In the meantime, we're in need a new solution to get updates done automatically.

Getting WordPress to update for you

In the meantime, WordPress has you covered. Built-in automatic updates is a little known but extremely useful feature.

To set this up you'll need to access your site using an FTP editor, navigate to the root of your WordPress installation (probably `yoursite.com/`) and edit the file `wp-config.php`. Add the lines:

```
define( 'WP_AUTO_UPDATE_CORE', true );  
add_filter( 'auto_update_plugin', '__return_true' );  
add_filter( 'auto_update_theme', '__return_true' );
```

Update the file and you're done.

For `WP_AUTO_UPDATE_CORE` you can change `true` to `false` to disable automatic updates entirely (not recommended but if you need to test everything first and will update manually promptly could be necessary) or change to `minor` to enable automatic updates for minor releases but not major releases (this is what happens by default).

You'll need to do this on every site you want automatic updates on. It'd be easier to use ManageWP on all your sites, but whilst there are no automatic updates available this is the best option.

Hardening WordPress security

You can keep WordPress secure by protecting yourself from widespread vulnerabilities in the software itself but you may still be vulnerable to weaknesses from users, plugins and themes. There are a number of things you can do to lock down the rest of your site.

Take advantage of Jetpack's security features

The ever-useful [Jetpack](#) plugin features some useful security features. You'll find these under Jetpack → Settings → Security. The essential feature to activate is Protect, which will prevent brute-force login attempts by locking out users with large number of wrong password entries. You can enter your IP address so you're not accidentally locked out your own site.

You can also make use of Jetpack's downtime monitoring so you know when something's up with your site. Security scanning is also available as a pro feature.

Use strong passwords and two factor authentication.

WordPress will warn you when creating a password if it's weak, but you'll get an even stronger password using a password manager such as [1Password](#) or [LastPass](#) will get you an even stronger password (which, with a password manager, you don't need to specifically remember).

You'll then want to pair your strong password with two factor authentication, so even if someone has your password, only you can login as you'll need a code from your phone. [Google Authenticator](#) is a free plugin which lets you use Google's secure app (which you may already have) to logon to your site. Installation is easy and adds a strong extra security layer to your site.

Lock down the plugin and theme editor

The little-loved `wp-config.php` file is probably the most important file in your WordPress installation. Without it, WordPress is missing basic configuration details and can't connect with your database.

If you use an automatic installation service for your WordPress install you'll find `wp-config.php` handled for you. Otherwise, you'll need to enter your details to `wp-config-sample.php` and rename the file once you're done.

Normally the config file is touched when doing the famous five minute install and at not at any other time. This is sad, because it lets you enable some really handy security features. Load up the file for editing and add the following two lines:

```
define( 'DISALLOW_FILE_EDIT', true );
define( 'DISALLOW_FILE_MODS', true );
```

This will tell WordPress to disable the plugin and theme editor, and disable the plugin and theme installer and updater respectively. Using these will prevent users (or clients) from interfering and making any changes which may compromise your site. Good to know.

Because wp-config is such a powerful file, always be extra careful when editing (and create a backup of the file). Here's what WordPress.org recommends:

Before you save the file, be sure to double-check that you have not accidentally deleted any of the single quotes around the parameter values. Be sure there is nothing after the closing PHP tag in the file. The last thing in the file should be ? > and nothing else. No spaces.

So – double check and then save. Enjoy the added safety.

Be careful with themes and plugins

Themes and plugins aren't dangerous per se, but as they're allowing code from an unknown source to run on your site you need to know you can trust them. Two easy rules to follow:

1. **Only install themes and plugins from reputable sources.** That means the WordPress theme and plugin repositories and well-established third-party premium sources. Avoid sites offering free versions of premium themes.
2. **Keep plugins to a minimum and delete plugins that are no longer needed.** The more plugins you have, the more likely there will be a security vulnerability on your site. Deactivate and delete those you're not using any more.

Keep an eye on users and permissions

You can have all the security in the world, but if you give the key to the wrong person it's not going to help. Make sure users only have the level of permission they need: be especially careful handing out administrator accounts, and keep an eye on who can publish content without you knowing (editor and author roles). Also make sure user registration is disabled (provided it's not needed), so you have control over which accounts exist.

If you need help from theme or plugin support create a temporary user rather than creating a new user, use this plugin to [create a temporary login](#) which can expire after a couple of days. After you're done with support, the account can be deactivated and there's no risk of continued access.

Security is important

Security is really not something to take lightly. WordPress generally gives you nothing to worry about – but only if you're always up-to-date. This lesson has given you the skills you need to handle WordPress' updates and the tools you need to automate the process. You're even equipped if anything goes wrong :)

This is something to set-and-forget, and something to set right now – so get to it!



The end.

Thanks for reading! If you have any queries feel free to reach out to us via email: hello@masterwp.co.

Still want more? [Sign up at MasterWP.co](https://masterwp.co).

– Alex and Ben