

Tooling prerequisites for theme designers

Tooling prerequisites for theme designers

The way 1.7 themes are created is significantly different from the way they were with PrestaShop 1.6. By taking the freedom to rewrite the way themes work according to the latest best-practices, we chose to also work with modern tools. These tools are widely accepted as best-of-breed in the current web development; hence, learning to use them will not only make you more productive in PrestaShop development, but more importantly in web development in general.

While those tools will help all designers and developers in the long run, there is a first stop of knowing how to use them, particularly if you have never used such a chain of tools through the command line.

This page is here to help! It will browse through the toolset, so that you know what the rest of the documentation is talking about :)

- [Tooling prerequisites for theme designers](#)
 - [Which tools are we talking about?](#)
 - [Before we get started: using the command line](#)
 - [Using Git to manage your project files](#)
 - [Why you should use it for your PrestaShop theme?](#)
 - [How can you install it?](#)
 - [Using Composer to manage your PHP dependencies](#)
 - [Why you should use it for your PrestaShop theme?](#)
 - [How can you install it?](#)
 - [Using npm to automate compilation from third-party package](#)
 - [Why you should use it for your PrestaShop theme?](#)
 - [How can you install it?](#)
 - [Using Webpack to compile and minify your asset files](#)
 - [Why you should use it for your PrestaShop theme?](#)
 - [How can you install it?](#)

Which tools are we talking about?

The PrestaShop developers chose to rely on these tools for the development of the Classic theme:

- Git
- Composer
- npm
- Webpack

Along with these tools, PrestaShop 1.7 introduces frameworks and design tools that are must-know:

- Bootstrap 4
- Symfony and its Twig template engine
- Sass
- Bourbon

This document presents each tool, and how they can get you to become a more productive web developer -- not just a more productive PrestaShop theme developer!

Before we get started: using the command line

Many of the tools cited above are to be used through the command line -- launching commands in a text-based environment, line after line. While it has the advantage of being very powerful and easy to automate, it is also not intuitive and hard to grasp. Luckily, it only take a handful of commands to understand the power of the command line, and to adopt it for your everyday processes.

Now, as a web-designer, you might be more used to work with graphical interfaces, such as Dreamweaver (for the more WYSIWYG-inclined), or a full-featured text editor, such as Sublime Text, Atom or Brackets (among many other possibilities). Using the command line can supplement this.

All operating systems give you access to a command line interface (CLI).

- Windows: It is called the "command prompt".
 - You can open it this way:
 1. Open the Start menu, for instance by pressing the Windows key of your keyboard.
 2. In the Search field, type `cmd` and press Enter. This will start the `cmd.exe` program.
 - Try out a few basic commands:
 - `dir`: lists the files in the current folder.
 - `cd`: change the current directory. For instance, "`cd Downloads`" to enter a Downloads directory in the current folder; or `cd ..` to go to the parent folder.
 - `ping`: to see the response time for a website. For instance, `ping google.com`.
 - Note: Windows 10 also includes the Linux-
- OS X: It is called the "terminal".
 - You can open it this way:
 1. Open Spotlight, for instance by pressing the Command key and Space-bar at the same time.
 2. In the Search field, start typing "terminal" until the real "Terminal" is suggested, then press Enter. This will start the Terminal app program.
 - Try out a few basic commands: * `ls`: displays the content of the current folder. * `cd`: change the current directory. For instance, `cd Downloads` to enter a Downloads directory in the current folder; or `cd ..` to go to the parent folder. * `ping`: to see the response time for a website. For instance, `ping google.com`.
- Unix/Linux: Oh come on, don't tell me you don't already use bash, tcsh, zsh or any other POSIX variation of the concept! :)

There are many default tools and commands which you can use and even combine, but we're going to use some non-standards tools: npm, Git, Composer and Webpack. Because they are non-standards, you will have to install and configure them first.

Note that you do not HAVE TO use these exactly: you can choose alternatives such as Gulp or Grunt if you're more familiar with them. We simply use Webpack :)

Using Git to manage your project files

Git is a version control system, designed to manage decentralized projects with speed and efficiency. In plain English: you can use Git to save the current state of your files (through individual snapshot), in order to return to a previous state should the need arise, among other things.

As a theme designer, tracking all your CSS and JavaScript changes can help you return to a previous snapshot of a file or of the whole project, add new code without breaking the code that's already working, collaborate with others, and in general keep you from losing your project because of a bad Ctrl-S.

If you have already used the Subversion system, think of Git as a decentralized Subversion (to keep it simple): any machine hosts all versions of all the files, and therefore there is no centralized repository. In effect, the PrestaShop project is mainly hosted on GitHub -- but developers also work on it on their own machine.

Why you should use it for your PrestaShop theme?

The PrestaShop Open Source project makes heavy use of Git in order to keep versions of its files. The community can access the official public repository on GitHub: <https://github.com/PrestaShop/PrestaShop>

As a theme designer, you will need Git in order to retrieve the latest version of PrestaShop 1.7 and its default theme, "Classic".

Note: You do not need to install or use Git if you are not interested in the latest development of PrestaShop or the Starter Theme.

How can you install it?

Simply download the archive for the latest version of Git (2.10 as of this writing) and install it just like you would do for any other software:

- Windows: <https://git-for-windows.github.io/>
 - User interfaces are available, for instance <https://tortoisegit.org/> or <https://www.gitkraken.com/>
- OS X: <https://git-scm.com/download/mac>
 - User interfaces are available, for instance <https://git-fork.com/> or <https://www.gitkraken.com/>
- Linux: type `apt-get install git` (or your distribution's equivalent).

You can then open your command line and type `git --version` to check that Git is indeed installed. It should display the version number. You're good to go!

Using Composer to manage your PHP dependencies

The PHP ecosystem is made of code packages which can be embedded into bigger projects, or even into other code packages which, in turn, can be embedded, etc. These packages are called dependencies.

Being able to rely on such code packages is great, because it prevents from having to reinvent the wheel every time a well-known feature is needed. On the other hand, dependencies can get problematic: the more a project depends on third-party package, the more it must manage dependency resolution (determining which package to use), autoloading solution (finding the right package and making it available automatically), and keeping all packages up to date (or not, if backward compatibility is an issue).

In short, the modern PHP ecosystem can get complex quite quickly, and Composer is the main way PHP developers manage their dependencies.

Why you should use it for your PrestaShop theme?

Now, why should you care about PHP files when working on a PrestaShop theme? Since you mostly work with theme files (.tpl, .css and .js), .php files are few in your editor, and "PHP dependencies" is something out there, and it seems you'd be better off avoiding them, right?

Thing is, PrestaShop uses Composer to build its own package dependencies. While the Zip archive available for public download is packed with all the required dependencies, the Git-hosted files do not automatically download and install all those dependencies: PrestaShop developers rely on Composer for that, through a `composer.json` file located at the root of the official Git folder.

In short, you need to use Composer when working with the latest development version from GitHub, in order to have a complete set of packages.

Yes, that means that if you choose not to retrieve the latest PrestaShop (or Starter Theme) files using Git, but to simply install the latest public archive, then you do not need to worry about Composer. But not using the latest Git version also means that you cannot work with the development version of PrestaShop, and that you rely on the Core developers to release upgraded packages, thus giving you no head start in developing with new features. Your call!

How can you install it?

Download the archive for the latest version of Composer (1.2.2 as of this writing) and install it just like you would do for any other software:

Note: you need to already have the PHP tool installed on your machine. You probably already have it if you're building websites locally. If not, install WampServer, EasyPHP, XAMPP or any other Windows Apache+PHP+MySQL package there is.

- Windows: <https://getcomposer.org/Composer-Setup.exe>
- OS X and Linux: in your command line, type these commands:
 - `curl -s https://getcomposer.org/installer | php`
 - `sudo mv composer.phar /usr/local/bin/composer`

Testing it requires using the command line; there is no graphical interface for this tool. Simply type `composer --version` to check that it is indeed installed.

Using npm to automate compilation from third-party package

npm is a popular package manager, which originates from the JavaScript ecosystem -- most precisely, from the Node.js JS runtime environment.

npm is both a command line tool and an online registry (located at <https://www.npmjs.com/>): you can use it to manage the dependencies (yes, just like Composer), or simply to work with useful packages. It is an extremely useful command to have when building websites, even if you do not use JavaScript or Node.js.

Why you should use it for your PrestaShop theme?

In the context of building PrestaShop themes, npm is mostly used in order to automate tasks -- namely, building assets automatically so that you don't have to.

PrestaShop 1.7 themes are built around "assets": CSS, JavaScript and image files, which are (or can be) generated from easier-to-manager formats:

- CSS files are built from Sass files (.scss).
- Some image files are built from SVG files (.svg).
- JavaScript files are from several files (.js).

All the source files are located in the theme's `_dev` folder. The generated files are built using Webpack, a JavaScript module bundler. See below for more information about Webpack.

How can you install it?

npm cannot be directly downloaded and installed. It is an integral part of the Node.js tool, therefore you need to download and install Node.js, which in turn will install npm for you.

To download Node.js, head to <https://nodejs.org/en/download/> and choose the LTS version for your platform. LTS means Long Term Support (currently, 6.9.1): this version contains less features than the Current version (currently, v7.2.0), but is very stable and will be supported until April 2018.

So, install Node.js on your machine, then test that npm is available:

- Open your command line interface.
- Type `node -v` to check that Node.js is installed.
- Type `npm -v` to check that npm is indeed available.

npm is updated much more frequently than Node.js, so chances are that there's a more recent version available than the one from the Node.js package.

To update your npm:

- Open your command line interface.
- Type `npm install npm -g`: this tells npm to install npm as global package.
- Type `npm -v` to check if the version has indeed changed.

Using Webpack to compile and minify your asset files

Modern website are getting more and more complex, and JavaScript becomes more prominent than even in the web-development world. As a result there is a lot of code on the client side!

Webpack was built in order to make your life easier, most notably by organizing your code into JavaScript modules. It takes a whole lot of work off your shoulders: you have better things to do than to edit configuration files in order to adjust media files, fonts or URLs.

Before Webpack, many were using task-runners such as Grunt or Gulp in order to organize their code. That lead to a patchwork of configuration, and you had to pay a lot of attention to any change in order to not break everything. Webpack fixes this in an elegant way.

Why you should use it for your PrestaShop theme?

Let's see where we're at. So far we have installed:

- Git: A better way to make snapshot of your codebase.
- Composer: A better way to manage your PHP dependencies.
- npm: A better way to automate tasks (among many other things).

It's all fine and dandy, but all of this remains very developer-centric, and there comes a time when you have to think about the user, and optimize for the browser. Fear not, for Webpack is here to save the day! Webpack is a "module bundler", meaning that it turns your assets into JavaScript modules, and packs them into static assets.

So, the main interest of using Webpack is that it will compile all your styles into a single CSS file. This way, your theme will make only one HTTP request for this single file, and since your browser will cache it for later re-use, it will even download this file only once.

The same goes with your JavaScript code. Instead of loading jQuery along with its community plugins, your own custom plugins and any extra code you might need, Webpack compiles and minifies all this JavaScript code into a single file, which will be loaded once - and cached.

How can you install it?

From the moment you have npm installed (see above), Webpack can be installed in a few seconds:

- Open your command line interface.
- Type `npm install webpack -g`