Author
John Hughes

How to Add RTL Support to Your Custom WordPress Theme

Filed Under
Beginners

Published
March 12, 2018

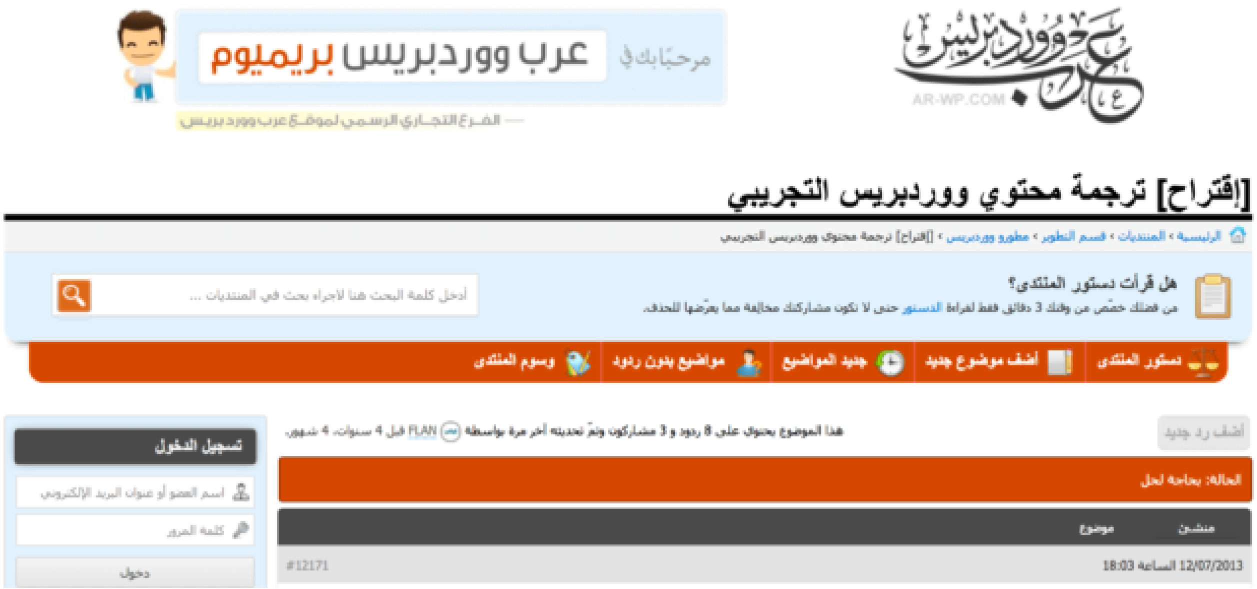
7 Comments

About **75% of the internet** speaks a language other than English. Some of these other languages read Right-to-Left (RTL), which can completely ruin the layout of many English-only designs.

Thankfully, having a theme that flows well for RTL languages (such as Arabic) is largely a matter of updating your CSS. With a few design considerations and style tweaks, you can adapt any WordPress theme to work with RTL languages.

In this article, you'll learn how RTL languages function in the browser, and how to work with them in your stylesheets. First, though, let’s delve into why you should add support in the first place!

Why It’s Worth Adding RTL Support to Your Theme


مرحبًابك في
عرب ووردبريس بريميوم
— الفرع التجاري الرسمي لموقع عرب ووردبريس
AR-WP.COM
الرئيسية › المنتديات › قسم التطوير › مطورو ووردبريس › [إقتراح] ترجمة محتوي ووردبريس التجريبي
أدخل كلمة البحث هنا لاجراء بحث في المنتديات ...
دستور المنتدى
أضف موضوع جديد
جديد المواضيع
مواضيع بدون ردود
وسوم المنتدى
أضف رد جديد
تسجيل الدخول
اسم العضو أو عنوان البريد الإلكتروني
كلمة المرور
دخول
منشئ
موضوع

[إقتراح] ترجمة محتوي ووردبريس التجريبي

هل قرأت دستور المنتدى؟

من فضلك خصّص من وقتك 3 دقائق فقط لقراءة الدستور حتى لا تكون مشاركتك مخالفة مما يعرّضها للحذف.

هذا الموضوع يحتوي على 8 ردود و 3 مشاركون وتمّ تحديثه آخر مرة بواسطة FLAN قبل 4 سنوات، 4 شهور.

الحالة: بحاجة لحل

12/07/2013 الساعة 18:03 #12171

Translating a website to another language can be an onerous task, though relatively simplified with plugins. However, while many languages read left to right (like English, of course), there are plenty of languages that are read from right to left. Simply translating these sites won’t be enough – the design of the site needs to be updated to match as well.

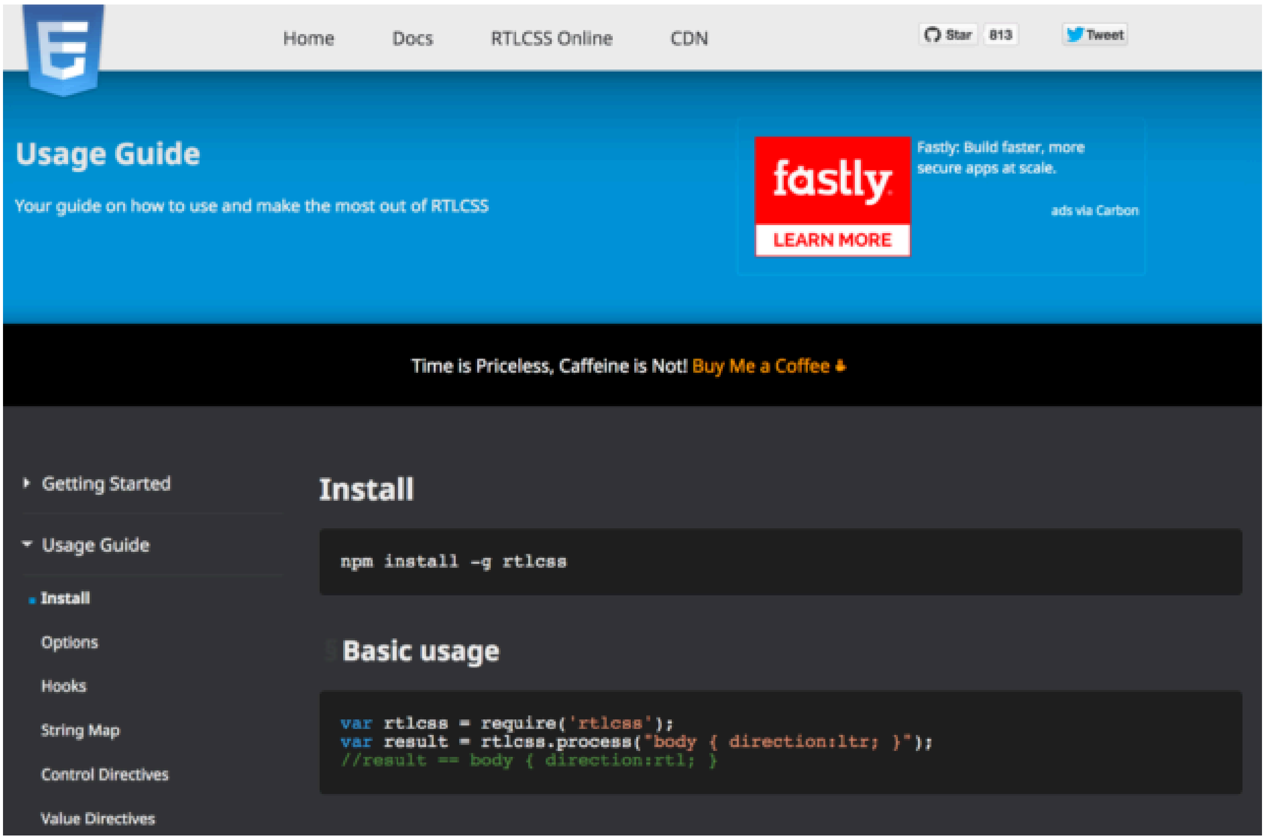
One example of a popularly spoken right to left language is Arabic. It’s the sixth most spoken language in the world, with over 420 million speakers, and around **40% of them** have access to the internet. This is a large niche market to consider. Keep in mind that this is only *one* of several popularly spoken right to left languages.

Opening up your theme to support all language styles, no matter which direction they read, makes your theme translation-ready for every situation. Fortunately, updating your theme to support right to left languages is relatively straightforward.

How to Add RTL Support to Your Custom WordPress Theme (In 3 Steps)

Since you're essentially designing a theme, we'll assume you're **developing locally** on your computer. If you're not, you definitely *should* be. Also, before making any changes, **create a backup** of your site to ensure you can recover everything if something goes wrong. When you're ready, let's dive in!

Step 1: Create the RTL CSS for Your Theme



Install

```
npm install -g rtlcss
```

Basic usage

```
var rtlcss = require('rtlcss');
var result = rtlcss.process("body { direction:ltr; }");
//result == body { direction:rtl; }
```

An RTL CSS converter is likely your most direct route to easily implemented RTL languages.

The first step is to create the CSS for displaying RTL languages correctly on your website. There are two methods for doing this. The first is to simply feed your existing stylesheet into a CSS RTL generator. This tool will examine your CSS and attempt to make a copy, but switching all the left and right alignments. This effectively creates a mirrored version of your website.

Here are two sites we recommend:

01 **RTLCSS:** This framework has a very **detailed set of instructions**, and at current writing is also beta testing a live converter tool.

02 **CSSJanus:** This tool is simplicity itself – type your Left-to-Right (LTR) CSS in the left-hand panel, click the *Submit* button, and see your RTL CSS on the right.

This is by far the simplest way to go. Once you've generated the new CSS, save the resulting styles as a file named *style-rtl.css*, ready for use later.

The second method is a little more tedious because it involves an 'override' stylesheet. In this version, you'll go through each line of your CSS and manually move all horizontal positioning elements into a separate file named *rtl.css*, and mirror them. This results in a smaller file, but ultimately takes a lot more work to keep up with, especially as you continue to make new updates and changes to your theme.

To do this, first add the following code to the top of your *rtl.css* file:

```
body {
   direction: rtl;
   unicode-bidi: embed;
}
```

Then, for the rest of your selectors, ensure you switch each left or right positioning to the opposite. Look for *text-align*, *float*, and *clear* in particular. However, the exact selectors you switch will be based on your own (and your client's) needs.

If you have any background images indicating a specific direction, you'll need to provide a flipped version and update the reference to it in the *rtl.css* stylesheet as well. Meanwhile, all of these attributes need to be 'zeroed' for the original image, before setting a new value:

- *margin*
- *padding*
- *borders*
- *background-position*
- *right* and *left* positioning

Finally, any buttons pushed off screen using negative text-indents should be switched to positive instead. This ensures that your functionality stays in place regardless of the device.

Once you've made these changes, save the new file as *rtl.css* in the same directory as *style.css*. You can now move onto ensuring WordPress is able to load your file when necessary.

Step 2: Ensure WordPress 'Sees' the RTL Styles

If you're using a CSS generator tool, you'll now need to **enqueue your new *style-rtl.css*** stylesheet so WordPress can load it in at the appropriate time. This is just a matter of adding a snippet to your theme's *functions.php* file:

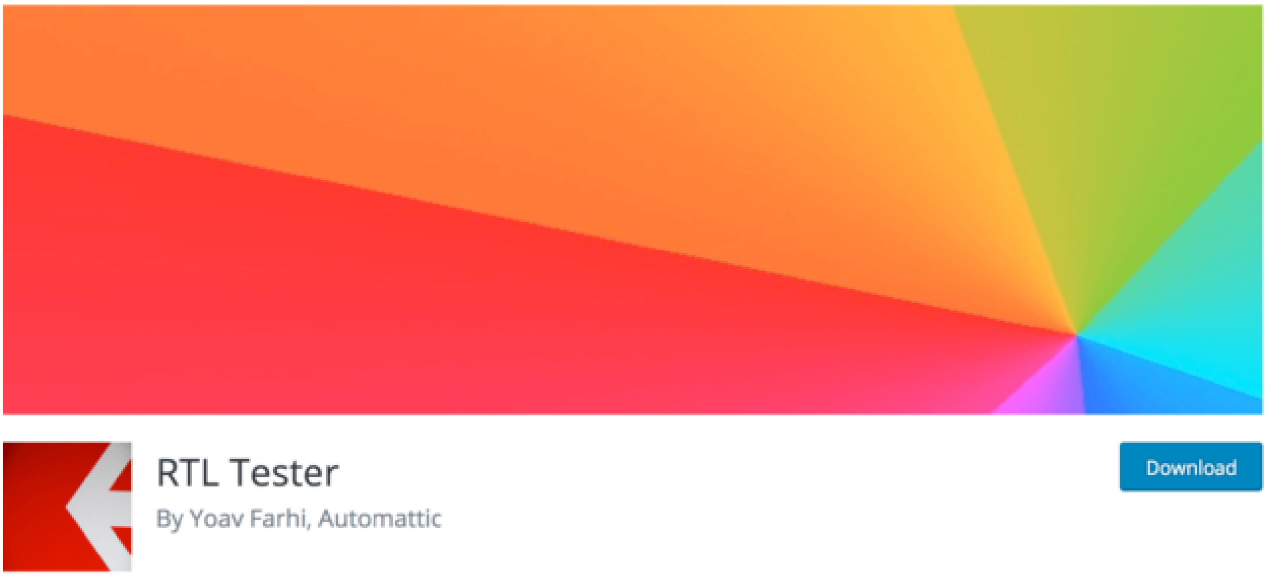
```
function enqueue_theme_files() {
    wp_enqueue_style( 'themeslug-style', get_styles
    wp_style_add_data( 'themeslug-style', 'rtl', 'r
}
add_action( 'wp_enqueue_scripts', 'enqueue_theme_fi
```

This tells WordPress to load the correct stylesheet when the website is set to an RTL language. Assuming you're already using *enqueue* properly to load in your regular *styles.css* stylesheet, you can simply add the *wp_style_add_data()* information into your existing enqueued styles function.

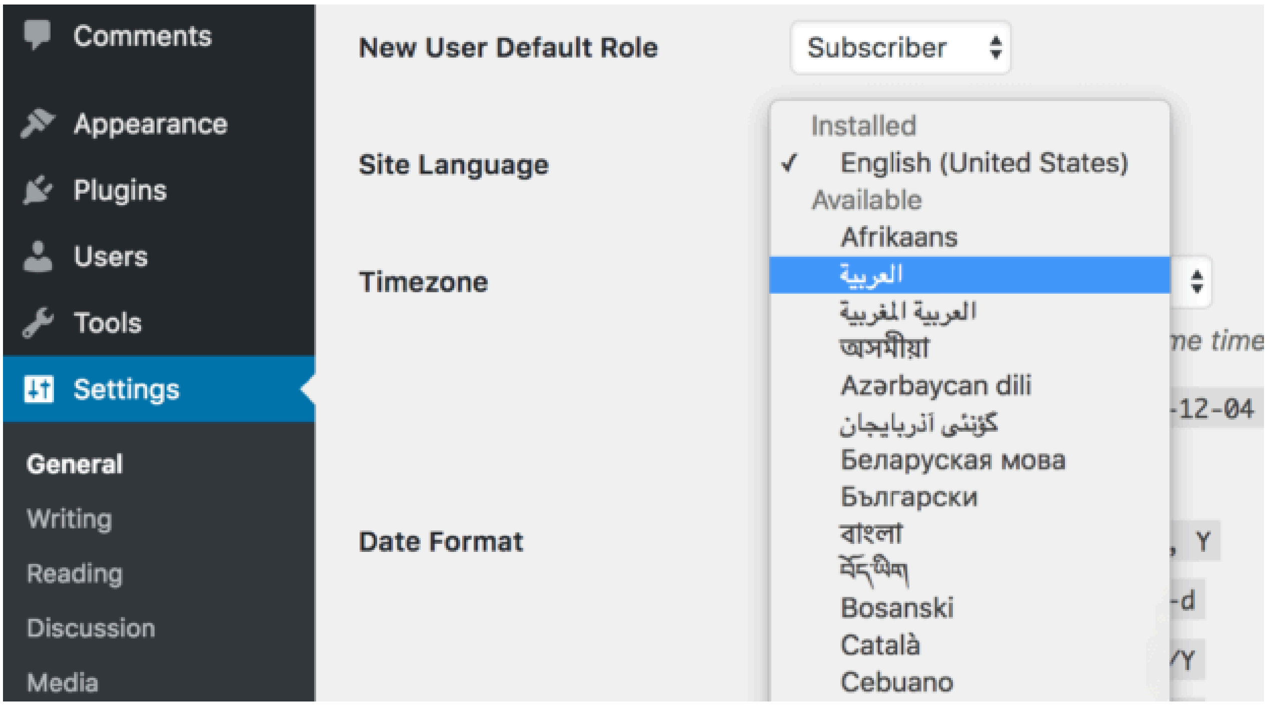
However, when using a dedicated override stylesheet, you won't need to load in any special functions. WordPress will simply load in the *rtl.css* file after your regular *styles.css*, which will overwrite directional cues using the secondary file.

Once your chosen method is ready to go, it’s time to test whether or not everything is working correctly in WordPress. Let’s take a look at how you can test your new RTL-supported theme design. Seeing it in action will enable you to ensure that everything looks as expected, and take any necessary manual fixes into account.

Step 3: Test Your RTL Styles in WordPress



You have a few options for testing out RTL languages in your WordPress installation. The first is to simply switch WordPress to an RTL language. To do this, go to the *Settings > General* page in your WordPress dashboard. Then, select the first script language you see in the *Language* drop-down box:



New User Default Role: Subscriber

Site Language

Timezone

Date Format

After saving, you'll see that everything is now right to left. However, unless you're bilingual of course, you'll struggle understand any of the text. To work around this, you can use developer tools to enable RTL styles while keeping English as the primary language.

There are two plugins that could fit the bill here. First, there is the general **RTL Tester** plugin. Once **installed and activated**, this essentially flips the switch on your entire website, telling it to load any and all RTL-specific styles. Second, there is **WP-RTL**, which lets you switch between different directional languages within a single post.

While the functionality of these plugins is good, bear in mind that they haven't been updated recently. Given their simple and esoteric functionality, that's understandable. Regardless, you'll likely need to make sure any additions work with your current WordPress installation, although they'll likely be surplus to requirements when pushing the site live.

Between these two plugins, you should be able to quickly test what various layouts look like with different RTL or LTR content. Finally, we recommend using WordPress' **Theme Unit Test Data** to make sure you're covering practically every permutation of text styles and formatting when adding RTL styles.

Conclusion

At least 200 million internet users speak a language that is written from right to left. Offering RTL styles in your themes opens up your design so it can be used by these markets. Without them, that's a good chunk of people currently missing out on your products and services (or your clients').

In this article, you've learned the importance of RTL styles and why you shouldn't overlook them. Next, we've taught you how to add them to your theme in three steps:

01 Create the RTL CSS for your theme.

02 Ensure that WordPress will see the RTL styles.

03 Test your RTL styles to make sure everything looks right.

What questions do you have about RTL styles in WordPress? Let us know in the comments section below!

*Image credit: **Christine Roy**.*

Author
John Hughes

John is a blogging addict, WordPress fanatic, and a staff writer for WordCandy.

More articles by John Hughes

Tagged:

rtl languages

WordPress insights delivered directly to your inbox every Monday morning.

TOONS

Quirky webcomics that WordPress professionals can relate to.

Recent articles

View all ›

Local vs XAMPP: Which Should You Use for Local Development?

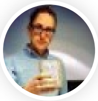
Nick Schäferhoff

WordPress File Permissions: Everything You Need to Know

Nick Schäferhoff

Google's E-E-A-T Guidelines: How to Show Experience in Content

Nick Schäferhoff



Press This: Data Liberation

Doc Pop

Advanced Custom Fields is Your 2024 Plugin Madness Champion!

Torque Staff

There are 7 comments

July 29, 2018 #
Segen

Hi there,
I have a single language website (Hebrew), it's possible to overwrite the main style.css with RTL directions?
I think in this way, I can have my WP english installation and at the same time have an RTL theme because WP just load one CSS file that is overwrited with RTL direction. It's correct? if not please help me.

October 23, 2018 #
Hamed

Hi,

This is undoubtedly the best article I have seen in months. It works perfectly. Everything is fixed the way they are supposed to. Thanks a lot John!

October 27, 2018 #
hajer

what about javascript. I think this is the hardest thing. when you have to rtl javascript transitions. how can we override it?

August 4, 2019 #
Somnath Jadhav

Hello John,

I am working on my first RTL website. You post is really helpful me.

Thank you for sharing.

September 23, 2019 #
Hussam

Thank you for the information,
Actually I added a plug in which helped me to write RTL but still has an issue with the titles, which did not change.
How can I apply changes to the titles and the main menu bar also?
Thank you

October 30, 2019 #
sfa

Maaaaaan! your saved my site... Thanks for the article. I am bilingual and RTL Tester did its magic.

Really appreciate your help

April 21, 2020 #
James Donegan

I am working with RTL for the first time, and I have created a new starter theme using underscores.me for my client. (I use underscores all the time, but this is the first time I've tried to incorporate RTL and Sass).

The RTL style is enqueued exactly as you suggest, however, its font choices overriding my style.css and .scss files even though the language is set to American English:

This is from style-rtl.css

`body, button, input, select, optgroup, textarea {
color: #404040;
font-family: sans-serif;
font-size: 16px;
font-size: 1rem;
line-height: 1.5;
}`

And it is overriding this from _typography.scss

`body, button, input, select, optgroup, textarea {
color: #404040;
font-family: ‘Lato’, sans-serif;
font-size: 16px;
font-size: 1rem;
line-height: 1.5;
}`

How do you stay on top of your WordPress game?

A good way to start is to sign up for our weekly newsletter. We keep you up to date by bringing you the freshest WordPress content from the brightest minds in the industry.

Email*

Subscribe

Create your own free forms to generate leads from your website.

Browse Topics

- Agency •
- Archives •
- Beginners •
- Business •
- Community •
- Development
- E-Books •
- Events •
- Featured •
- News •
- News Drop •
- Plugins •
- Security
- Themes •

- Tools
- Videos
- Torque Toons

About Torque

- About Torque
- Contact
- Contribute
- Privacy Policy

[1] WP Engine is a proud member and supporter of the community of WordPress® users. The WordPress® trademarks are the intellectual property of the WordPress Foundation, and the Woo® and WooCommerce® trademarks are the intellectual property of WooCommerce, Inc. Uses of the WordPress®, Woo®, and WooCommerce® names in this website are for identification purposes only and do not imply an endorsement by WordPress Foundation or WooCommerce, Inc. WP Engine is not endorsed or owned by, or affiliated with, the WordPress Foundation or WooCommerce, Inc.