# Beginners Guide: Implement jQuery Scripts into WordPress Websites

Tutorial · https://mhthemes.com/support/knb/implementing-jquery-scripts-wordpress/

jQuery scripts come in all kinds of shapes and sizes. They can do a variety of awesome things, but implementing them into your WordPress website can be a challenge for beginners. In particular, if the documentation is littered with hard-to-understand jargon and unclear instructions. In this article we shall walk you through the process of correctly implementing jQuery scripts into WordPress sites.

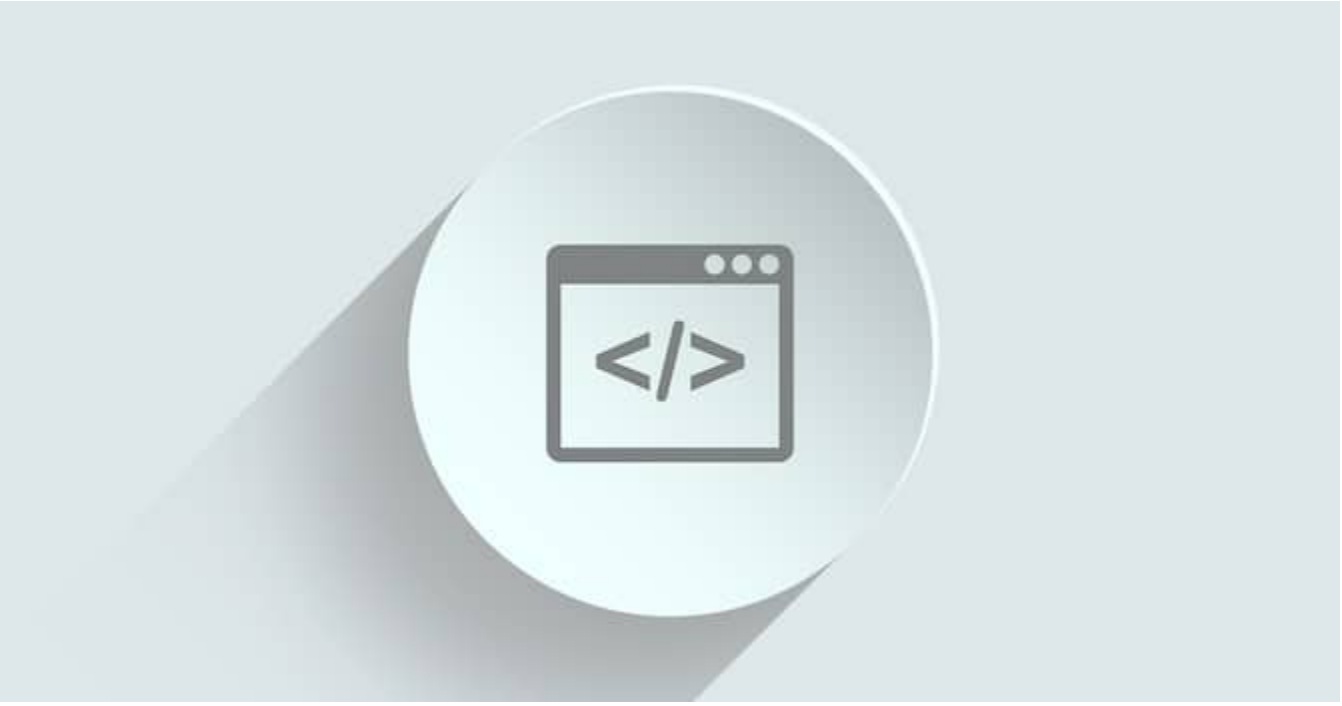

Image Source: ivke32 – Pixabay.com / License: CC0 Public Domain

## Using a Child Theme

Firstly, we're going to be using a child theme to add scripts to a WordPress website. If you're using a premium WordPress theme then chances are, that you have been provided one with your purchase. If not, you can easily create your own child theme. With your child theme installed and activated on your WordPress site, you're ready to start adding scripts to it. The child theme will ensure that your implementations won't get lost in case you update the parent theme in the future.

## Choosing a jQuery Script

Let's use a *zoom on hover* script called Drift as an example. By visiting the documentation for the script, you will be greeted with some installation information, let's go through this for a moment.

> **NPM:** npm install drift-zoom
>
> **Bower:** bower install drift
>
> **Manual:** Download and use dist/Drift.min.js or dist/Drift.js
>
> If your build process will re-run *dist/Drift.js* or *dist/Drift.min.js* through Browserify, you'll need to add *noParse: ['drift-zoom']* to your Browserify config. If you skip this, Browserify will attempt to re-require Drift's dependencies which have already been inlined.

To a beginner, the above installation text won't be of much help at all, but all that is required is that you obtain the *Drift.js* file that the documentation refers to, which you're able to do so via the download link you can see in the above image. This will download a ZIP archive that contains a variety of files, the *Drift.js* that you require is located within the folder structure as shown below.

```
drift-master/src/js/Drift.js
```

After you've downloaded the file, you can simply upload this JS file to your child theme folder using your preferred FTP client, or web host control panel.

## Enqueuing the Script in WordPress

Let's start by opening the child theme's *functions.php* file and adding the following code to enqueue the script in WordPress:

```
function my_custom_scripts() {
    wp_enqueue_script('drift', get_stylesheet_directory_uri() . '/Drift.js');
}
add_action('wp_enqueue_scripts', 'my_custom_scripts');
```

This will create a new PHP function called *my_custom_scripts()*. It instructs WordPress to load the *Drift.js* file from the correct location within your child theme. You can enqueue multiple JS files within

the same function just by referencing each JS file individually. Since each script usually requires some configuring you need to enqueue another JS file for this purpose, so please add the *scripts.js* file to the function as well, which should now look like this:

```
function my_custom_scripts() {
    wp_enqueue_script('drift', get_stylesheet_directory_uri() . '/Drift.js');
    wp_enqueue_script('myscripts', get_stylesheet_directory_uri() . '/scripts.js',
array('jquery'));
}
add_action('wp_enqueue_scripts', 'my_custom_scripts');
```

Once you have enqueued both JS files, you can save and close the *functions.php* file.

## Creating the scripts.js File

You may now create the *scripts.js* file that will contain the configuration code for the script you're implementing. Create a new file within your child theme and save it as *scripts.js*, then paste the following code within so it's ready for the configuration code.

```
jQuery(document).ready(function($){
});
```

This creates a new JavaScript function that makes use of jQuery. Our configuration code will be added within this function, much like the PHP function you created earlier.

## Configuring the Script

Now take a look at the usage information which is provided by the Drift script documentation.

> Once you've installed Drift via one of the above methods, you're ready to get started. There are no dependencies, so you can just start making cool stuff. Here's an example of the most basic possible implementation:

```
<img src="https://assets.imgix.net/dog.png?w=400" data-
zoom="https://assets.imgix.net/dog.png?w=1200">
```

```
<p>This is a simple description of the dog picture.</p>

new Drift(document.querySelector('img'), {
    paneContainer: document.querySelector('p')
});
```

Again, the above information provided may not be of much use to a complete beginner, but some configuration is being given that you can add to the function within *scripts.js*, like so:

```
jQuery(document).ready(function($){
    new Drift(document.querySelector('img'), {
        paneContainer: document.querySelector('p')
    });
});
```

Once you have added the configuration code, you can save and close the *scripts.js* file.

## Adding Script Options

Additionally, you are usually provided with a set of options that you're able to use to modify the scripts behaviour in a variety of ways. You can make use of these options again by using the *scripts.js* file. Let's take a closer look at the options available for the Drift script shall we? The script offers several options you can use and add to your *scripts.js* file like so:

```
jQuery(document).ready(function($){
    new Drift(document.querySelector('img'), {
        paneContainer: document.querySelector('p'),
        containInline: true,
        hoverBoundingBox: true,
        hoverDelay: 100
    });
});
```

The particular options of course will vary from script to script however, but all options can be contained within the single function you created earlier, making everything easy to manage.

## So far, so good

You can use the above steps to implement almost any jQuery script into your WordPress website. Simply enqueue the script using *functions.php* and add all configuration and options code via the *scripts.js* file. This is the point where scripts will begin to vary as you must now implement the scripts use into your website. And that usually entails modifying your HTML and CSS code in whichever way is detailed within the relevant documentation of the particular jQuery script.

In this example we implemented a script that allows your website visitors to magnify any area of an image using their mouse cursor. Via the usage information above, it shows that we need to do this by modifying our chosen *img*-tag. Which image tag you choose is entirely dependent on your needs and requirements and also dependent on the WordPress theme you're currently using. If you're looking for other jQuery scripts, have a look here: Best jQuery Scripts to Jazz Up your WordPress Website.