



Add inline scripts in WordPress

When working with WordPress there are times when you need to insert an inline script into a page or a template. This post will discuss two ways to do it

Method 1: wp_add_inline_script()

The first method, `wp_add_inline_script()` works with enqueued scripts and uses the same handle that we enqueued the script with to add the inline script to the page.

With this method we can be sure that the enqueued script will be added to the page before the inline script.

```
<?php
function mytheme_enqueue_typekit() {
    wp_enqueue_script( 'myth'.'mytheme-typekit'.'tps://use.typekit.net/.js',

    wp_add_inline_script(
        'mytheme-typekit',
        'try{Typekit.load({ async: true });}catch(e){}'
    );
}

add_action( 'wp_enqueue_scripts', 'mytheme_enqueue_typekit' );
```

Method 2: wp_footer()

Method 1 works best when the script we add is dependent on other scripts we've enqueued. There may be other situations where we need to add an independent script to the page that doesn't require any other scripts to be enqueued.

There may be times when I want to use a module (adding `type="module"` to the script). I am not aware of any way you can do so using `wp_add_inline_script()` or `wp_script_add_data()`.

Using [wp_footer](#) inserts the value of its second parameter into the document before the closing <html> tag.

You have to write the full script, including opening and closing <script> tags but it gives you more flexibility in how you write the script.

```
<?php
function script_init() {
    // This will work on browsers that support newer Javascript syntax
    echo '<script type="module">
    // Your code goes here.
    </script>';

}

add_action( 'wp_footer', 'scr"module"<script type="module">
    // Your code goes here.
    </script>'
```

Which method you choose depends on your needs. It may be that you use both on all your pages. As usual, test the code thoroughly before using it in production.