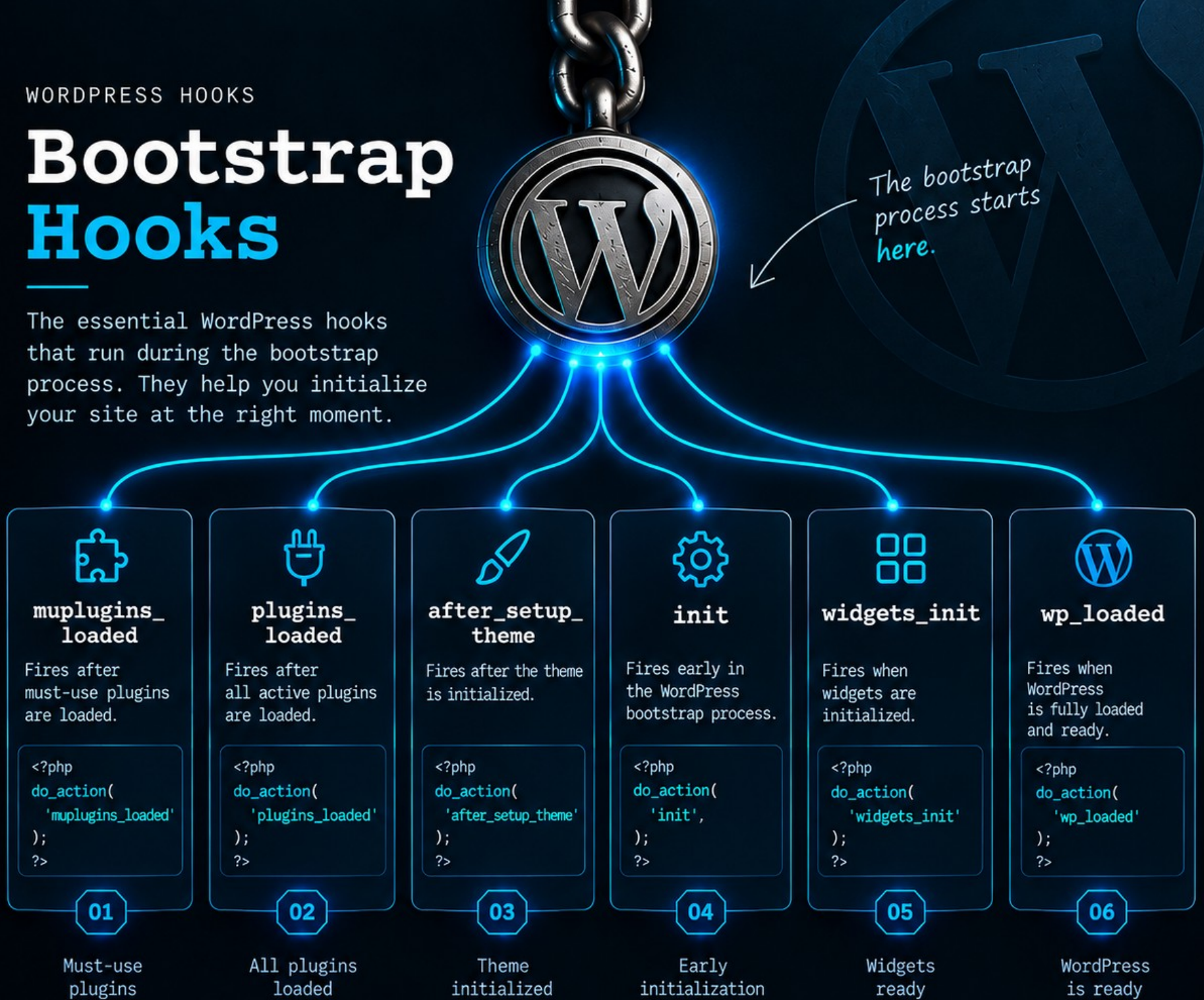


Bootstrap Hooks

The essential WordPress hooks that run during the bootstrap process. They help you initialize your site at the right moment.



4wp.dev/hooks/actions/bootstrap/

Build **better**. Hook **earlier**.

Understand the bootstrap hooks and create more powerful WordPress solutions.

muplugins_loaded()

The `muplugins_loaded` action hook is fired after all must-use plugins have been loaded, but before regular plugins are loaded.

Fired after *must-use plugins* are loaded and before regular plugins.



```
add_action( 'muplugins_loaded',  
    'my_init_plugin' );
```

Initialize your plugin



```
add_action( 'muplugins_loaded',  
    'load_textdomain' );
```

Load translation files



```
add_action( 'muplugins_loaded',  
    'setup_features' );
```

Setup features and constants



```
add_action( 'muplugins_loaded',  
    'register_custom_types' );
```

Register custom post types or taxonomies



```
<!DOCTYPE html>  
<html <?php language_attributes(); ?>>  
<head>  
    ...  
    do_action( 'muplugins_loaded' );  
    ...  
</head>  
<body <?php body_class(); ?>>  
    ...  
</body>  
</html>
```



4wp.dev/hooks/actions/muplugins_loaded/

Run code after must-use plugins are loaded.

Use the `muplugins_loaded` action hook to initialize your plugin, load translations, or set up features before regular plugins are loaded.

plugins_loaded()

The `plugins_loaded` action hook is fired **after all must-use and regular plugins** have been loaded and **activated**, but **before** WordPress starts setting up the environment.

*Fired after all plugins are loaded and activated, but **before** WordPress is set up.*



```
add_action( 'plugins_loaded',  
    'my_define_constants' );
```

Define Constants



```
add_action( 'plugins_loaded',  
    'my_init_dependencies' );
```

Initialize Dependencies



```
add_action( 'plugins_loaded',  
    'my_load_textdomain' );
```

Load Textdomain



```
add_action( 'plugins_loaded',  
    'my_plugin_integrations' );
```

Plugin Integrations

```
<?php
```

```
add_action( 'plugins_loaded',  
    'my_initialize' );
```

```
function my_initialize() {
```

```
    // All plugins are loaded and active.  
    // Safe to use most plugin functions  
    // and classes.
```

```
    // Your code here.
```

```
}
```

```
?>
```

Work with loaded plugins.

Use `plugins_loaded` to safely interact with other plugins, define constants, load textdomains, and more.

after_setup_theme()

The `after_setup_theme` action hook is fired **after** WordPress has finished initializing the theme but **before** any headers are sent. It's the ideal place to run theme setup code.

Fired **after** the theme is set up but **before** headers are sent.



```
add_action( 'after_setup_theme',  
    'my_theme_setup' );
```

Theme Setup



```
add_action( 'after_setup_theme',  
    'my_register_menus' );
```

Register Menus



```
add_action( 'after_setup_theme',  
    'my_add_theme_supports' );
```

Add Theme Supports



```
add_action( 'after_setup_theme',  
    'my_custom_logo_setup' );
```

Custom Logo Setup

```
<?php  
add_action( 'after_setup_theme',  
    'my_theme_setup' );  
  
function my_theme_setup() {  
    // Make theme available for translation  
    load_theme_textdomain( ... );  
    // Add theme supports  
    add_theme_support( ... );  
    // Register menus  
    register_nav_menus( ... );  
    // And more...  
}  
?>
```



4wp.dev/hooks/actions/after_setup_theme/

Set up your theme the right way.

Use `after_setup_theme` to register **features**, add **supports**, set up **menus**, and prepare your theme for success.

WORDPRESS HOOKS

init()

The `init` action hook is fired after WordPress has finished loading but before any headers are sent.

Fired after WP loads, before headers are sent.



```
add_action( 'init',  
'my_register_post_types' );  
Register Post Types
```



```
add_action( 'init',  
'my_register_taxonomies' );  
Register Taxonomies
```



```
add_action( 'init',  
'my_custom_textdomain' );  
Load Textdomain
```



```
add_action( 'init',  
'my_rewrite_rules' );  
Rewrite Rules
```

```
<!DOCTYPE html>  
<html <?php language_attributes(); ?>>  
<head>  
...  
do_action( 'init' );  
...  
</head>  
<body <?php body_class(); ?>>  
...  
</body>  
</html>
```



4wp.dev/hooks/actions/init/

Initialize your site.

Use `init` to set up post types, taxonomies, textdomains, rewrite rules, and more.

widgets_init()

The `widgets_init` action hook is fired **after** WordPress has finished loading but **before** the widgets screen is loaded in the admin. It's the perfect time to register and set up widgets.

*Fired after widgets are registered but **before** the Widgets screen is loaded.*



```
add_action( 'widgets_init',  
    'my_register_widgets' );
```

Register Widgets



```
add_action( 'widgets_init',  
    'my_widget_areas' );
```

Register Widget Areas



```
add_action( 'widgets_init',  
    'my_widget_defaults' );
```

Set Widget Defaults



```
add_action( 'widgets_init',  
    'my_custom_widget_supports' );
```

Add Widget Supports



```
<?php  
add_action( 'widgets_init', 'my_register_widgets' );  
  
function my_register_widgets() {  
    // Register a custom widget.  
    register_widget( 'My_Custom_Widget' );  
  
    // Or register multiple widgets at once.  
    register_widget( [  
        'My_First_Widget',  
        'My_Second_Widget',  
    ] );  
}  
?>
```


wp_loaded()

The `wp_loaded` action hook is fired **after** WordPress has finished loading all files, loaded plugins, set up the environment, and is ready to process the current request.

*Fired after WordPress has finished loading but **before** the request is fully processed.*



```
add_action( 'wp_loaded',  
    'my_security_checks' );
```

Run Security Checks



```
add_action( 'wp_loaded',  
    'my_init_features' );
```

Initialize Features



```
add_action( 'wp_loaded',  
    'my_load_textdomain' );
```

Load Textdomain



```
add_action( 'wp_loaded',  
    'my_custom_bootstrap' );
```

Custom Bootstrap



```
<?php
```

```
add_action( 'wp_loaded', 'my_setup' );
```

```
function my_setup() {
```

```
    // WordPress is fully loaded.
```

```
    // The environment is ready.
```

```
    // Safe to run most tasks.
```

```
    // Your code here.
```

```
}
```

```
?>
```

Ready when WordPress is.

Use `wp_loaded` to run code after WordPress has fully loaded but before the request is handled.