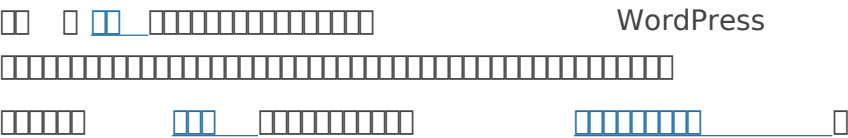
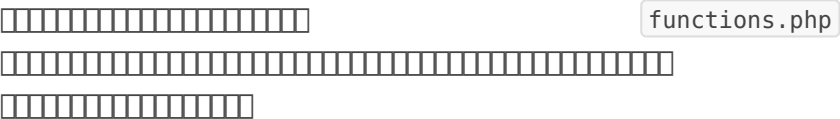




- [add\\_action\(\)](#)
  - [add\\_action\(\)](#)
  - [Assign \(hook\) your callback function](#)
  - [add\\_action\(\)](#)
    - [add\\_action\(\)](#)
    - [Number of Arguments](#)



# Assign (*hook*) your callback function

Second, add your callback function to the action. This is called *hooking* and tells the action to run your callback function when the action is run.

When your callback function is ready, use [add\\_action\(\)](#) to hook it to the action you have selected. At a minimum, `add_action()` requires two parameters:

1. `string $hook_name` which is the name of the action you're hooking to, and
2. `callable $callback` the name of your callback function.

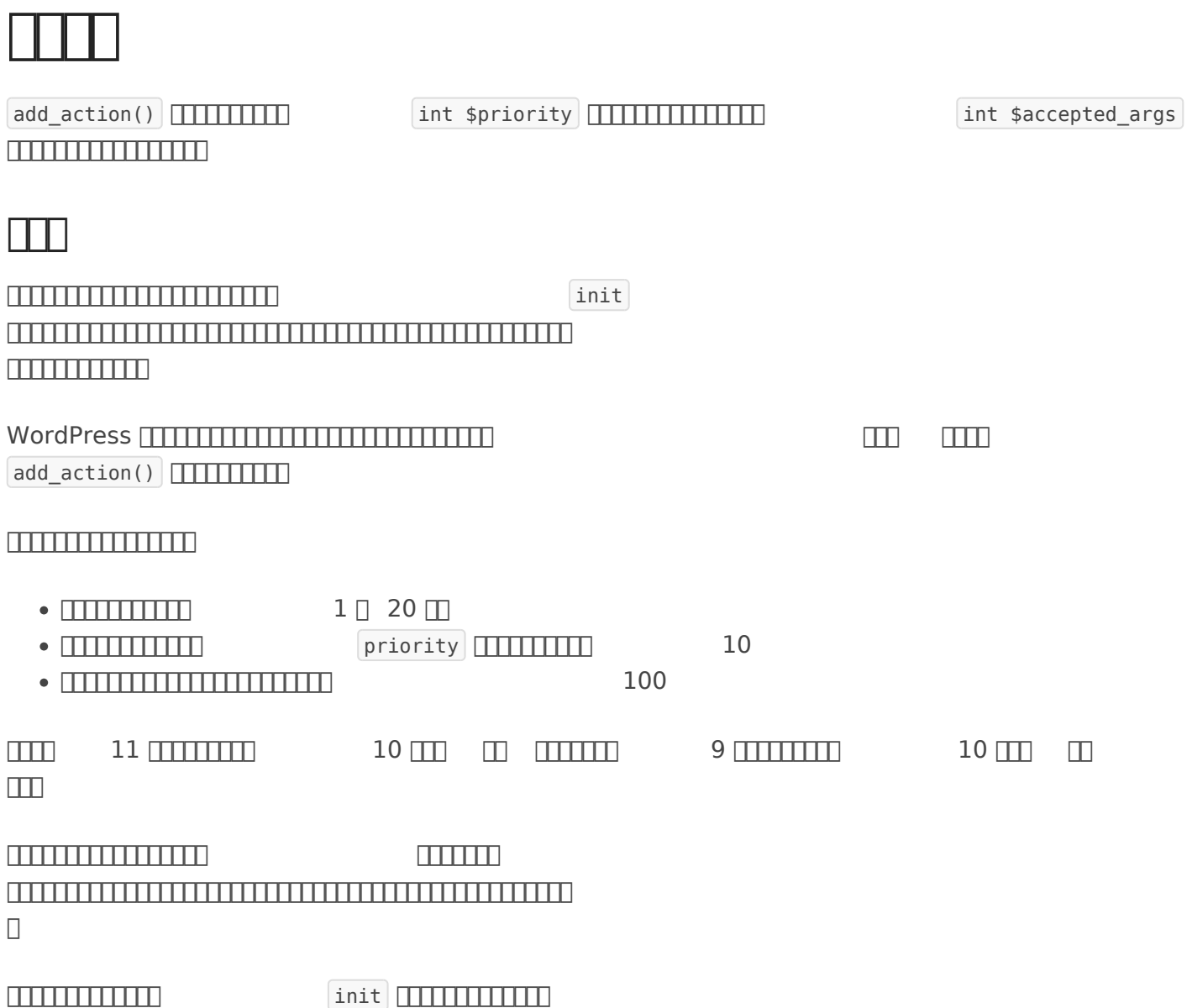
The example below will run `wporg_callback()` when the `init` hook is executed:

```
function wporg_callback() {
    // do something
}

add_action( 'init', 'wporg_callback' );
```

You can refer to the [Hooks](#) chapter for a list of available hooks.

As you gain more experience, looking through WordPress Core source code will allow you to find the most appropriate hook.



```
add_action( 'init', 'wporg_callback_run_me_late', 11 );
add_action( 'init', 'wporg_callback_run_me_normal' );
add_action( 'init', 'wporg_callback_run_me_early', 9 );
add_action( 'init', 'wporg_callback_run_me_later', 11 );
```

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

- `wporg_callback_run_me_early()` 9
- `wporg_callback_run_me_normal()` 10
- `wporg_callback_run_me_late()` 11
- `wporg_callback_run_me_later()` 11

## Number of Arguments

Sometimes it's desirable for a callback function to receive some extra data related to the action being hooked to.

For example, when WordPress saves a post and runs the `save_post` hook, it passes two parameters to the callback function: the ID of the post being saved, and the post object itself:

```
do_action( 'save_post', $post->ID, $post );
```

When a callback function is registered for the `save_post` hook, it can specify that it wants to receive those two parameters. It does so by telling `add_action` to expect them by (in this case) putting `2` as the fourth argument:

```
add_action( 'save_post', 'wporg_custom', 10, 2 );
```

In order to actually receive those parameters in your callback function, modify the parameters your callback function will accept, like this:

```
function wporg_custom( $post_id, $post ) {
    // do something
}
```

[tip]It's good practice to give your callback function parameters the same name as the passed parameters, or as close as you can.[/tip]

III #2

2026-03-27 01:08:39 UTC

2026-03-27 01:08:39 UTC