add_shortcode(string $tag, callable $callback)

Adds a new shortcode.

Description

Care should be taken through prefixing or other means to ensure that the shortcode tag being added is unique and will not conflict with other, already-added shortcode tags. In the event of a duplicated tag, the tag loaded last will take precedence.

Parameters

$tag string required
Shortcode tag to be searched in post content.

$callback callable required
The callback function to run when the shortcode is found.
Every shortcode callback is passed three parameters by default, including an array of attributes (`$atts`), the shortcode content or null if not set (`$content`), and finally the shortcode tag itself (`$shortcode_tag`), in that order.

More Information

The shortcode callback will be passed three arguments: the shortcode attributes, the shortcode content (if any), and the name of the shortcode.

There can only be one hook for each shortcode. This means that if another plugin has a similar shortcode, it will override yours, or yours will override theirs depending on which order the plugins are included and/or ran.

Shortcode attribute names are always converted to lowercase before they are passed into the handler function. Values are untouched.

Note that the function called by the shortcode should never produce an output of any kind. Shortcode functions should return the text that is to be used to replace the shortcode. Producing the output directly will lead to unexpected results. This is similar to the way filter functions should behave, in that they should not produce unexpected side effects from the call since you cannot control when and where they are called from.

Source

wp-includes/shortcodes.php

Expand code Copy

```
function add_shortcode( $tag, $callback ) {
    global $shortcode_tags;
```

```
    if ( '' === trim( $tag ) ) {
        _doing_it_wrong(
            __FUNCTION__,
            __( 'Invalid shortcode name: Empty name given.' ),
            '4.4.0'
        );
        return;
    }

    if ( 0 !== preg_match( '@[<>&/\[\]\x00-\x20=]@', $tag ) ) {
```

View all references·View on Trac·View on GitHub

Related

Uses

__()
wp-includes/l10n.php

_doing_it_wrong()
wp-includes/functions.php

Used by

WP_Embed::__construct()
wp-includes/class-wp-embed.php

WP_Embed::run_shortcode()
wp-includes/class-wp-embed.php

Changelog

Version	Description
2.5.0	Introduced.

User Contributed Notes

Codex9 years ago 16

Examples

Simplest example of a shortcode tag using the API: `[footag foo="bar"]`

Copy

```
add_shortcode( 'footag', 'wpdocs_footag_func' );
function wpdocs_footag_func( $atts ) {
    return "foo = {$atts['foo']}";
}
```

Example with nice attribute defaults: `[bartag foo="bar"]`

Copy

```
add_shortcode( 'bartag', 'wpdocs_bartag_func' );
function wpdocs_bartag_func( $atts ) {
    $atts = shortcode_atts( array(
        'foo' => 'no foo',
        'baz' => 'default baz'
    ), $atts, 'bartag' );

    return "foo = {$atts['foo']}";
}
```

Example with enclosed content: `[baztag]content[/baztag]`

Copy

```
add_shortcode( 'baztag', 'wpdocs_baztag_func' );
function wpdocs_baztag_func( $atts, $content = "" ) {
    return "content = $content";
}
```

If your plugin is designed as a class write as follows:

Copy

```
add_shortcode( 'baztag', array( 'MyPlugin', 'wpdocs_baztag_func' ) );
class MyPlugin {
    public static function wpdocs_baztag_func( $atts, $content = "" ) {
        return "content = $content";
    }
}
```

Log in to add feedback

Denis Žoljom4 years ago 11

When adding `add_shortcode() ` function in a plugin, it's good to add it in a function that is hooked to `init` hook. So that WordPress has time to initialize properly.

Copy

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

function wpdocs_add_custom_shortcode() {
    add_shortcode( 'footag', 'wpdocs_footag_func' );
}
```

As described in the plugins handbook.

Show feedback (1)Log in to add feedback

nasifshuvo1233 years ago 5

For wppb plugin boilerplate I add code to define_public_hooks()

```
$this->loader->add_action( 'init', $plugin_public, 'register_shortcodes' );
```

Then in public folder, in class_public file I added:

Copy

```
public function register_shortcodes(){

        add_shortcode( 'sample-shortcode','shortcode_function'  );
        function shortcode_function(  ) {
            return "Hello Shortcode";
        }
    }
```

Log in to add feedback

KittMedia2 years ago 5

Beware that the first argument passed to the callback is an empty string if no attributes are set in the shortcode. So a type declaration will fail:

“Fatal error: Uncaught Error: Argument 1 passed to shortcode_callback() must be of the type array, string given”

Use a check whether `$arguments` is not an array and if so, replace it with an empty array:

Copy

```
function shortcode_callback( $attributes, string $content, string $shortcode ) {
    if ( ! is_array( $attributes ) ) {
        $attributes = [];
    }

    // do stuff
}
```

Show feedback (2)Log in to add feedback

Ahmad Karim3 years ago 3

Example of `add_shortcode` function that uses `get_template_part` function to include a template.

Expand code Copy

```
function wpdocs_the_shortcode_func( $atts ) {
    $attributes = shortcode_atts( array(
```

```
        'title' => false,
        'limit' => 4,
    ), $atts );

    ob_start();

    // include template with the arguments (The $args parameter was added in v5.5.0)
    get_template_part( 'template-parts/wpdocs-the-shortcode-template', null, $attributes );

    return ob_get_clean();
```

Log in to add feedback

Patrick Johanneson6 years ago 2

Notes (from the Codex — https://codex.wordpress.org/Function_Reference/add_shortcode#Notes)

- The shortcode callback will be passed three arguments: the shortcode attributes, the shortcode content (if any), and the name of the shortcode.
- There can only be one hook for each shortcode. Which means that if another plugin has a similar shortcode, it will override yours or yours will override theirs depending on which order the plugins are included and/or ran.
- Shortcode attribute names are always converted to lowercase before they are passed into the handler function. Values are untouched.
- Note that the function called by the shortcode should never produce output of any kind. Shortcode functions should return the text that is to be used to replace the shortcode. Producing the output directly will lead to unexpected results. This is similar to the way filter functions should behave, in that they should not produce expected side effects from the call, since you cannot control when and where they are called from.

Log in to add feedback

Sabbir Mia2 years ago 1

$atts: an associative array of attributes, you can access the associative array anywhere.
$content: the enclosed content (if the shortcode is used in its enclosing form)

Expand code Copy

```
/**
 * Appends a <a> tag with URL to the content.
 *
 * @param array $atts the shortcode attributes.
 * @param sring $content the editor enclosed content.
 * @return string
 */
function button_callback( $atts, $content ) {
    // It's a default value.
    $default = array(
        'url' => '',
    );
    // You will pass default value after that user define values.
```

Log in to add feedback

123host1 year ago 1

When passing values via the shortcode e.g. `[myshortcode FooBar="one" SomethingElse="two"]` the keys in the array are transformed to lowercase

The array passed to the callback function will be: `Array( foobar => "one", somethingelse=> "two" )`

Don't be like me and spend hours trying to figure out why the values weren't being passed.

Log in to add feedback

Xedin Unknown7 months ago 1

⚠️⚠️

Currently, and for some time now, viewing a post with a shortcode in the editor caused the shortcode to get rendered. This is very unexpected behaviour described in an old issue. It can, for example, cause the post to become uneditable via the admin if there's an error in the shortcode, such as some visitor-facing area variables or functions not being declared, etc.

To go around this, as mentioned in a comment, use the `is_admin()` function to prevent computation on admin pages.

Show feedback (1)Log in to add feedback

You must log in before being able to contribute a note or feedback.

bbPress ↗

BuddyPress ↗

CODE IS POETRY