Shortcodes Available in Custom Checkout Fields

Shortcodes · https://wpfactory.com/docs/using-the-shortcode/

Custom Checkout Fields for WooCommerce registers three shortcodes you use *inside* field options such as the label, default value, placeholder, and datepicker min/max date. They do not create fields on their own. They output a value that the field option then uses.

> **Important:** these shortcodes are meant for the plugin's own settings fields, not for placement in page or template files. To add an actual field to checkout, set a field's **Type** on its tab, as described in How to add a custom field to WooCommerce checkout.

Which field options run shortcodes

These field options pass their content through the shortcode engine, so you can use the shortcodes below in them: Label, Placeholder, Default value, Description, Fee title, checkbox Value for ON / Value for OFF, and datepicker Min date / Max date.

1. Current date and time

`[alg_wc_ccf_datetime]` outputs the current date and time as text. It does not create a date field. Use it to pre-fill or display a live value.

Attribute	Effect	Default
`format`	A PHP date format string	Your WordPress date format

Example: set a field's **Default value** to `[alg_wc_ccf_datetime format="Y-m-d"]` to pre-fill today's date as `2026-07-21`.

2. Language-specific text

`[alg_wc_ccf_translate]` shows different text depending on the active language (it reads WPML's `ICL_LANGUAGE_CODE`). It has two forms.

Enclosing form, one block per language set:

```
[alg_wc_ccf_translate lang="EN,DE"]Text for English and German[/alg_wc_ccf_translate]
[alg_wc_ccf_translate not_lang="EN,DE"]Text for other languages[/alg_wc_ccf_translate]
```

Self-contained form, both texts in one tag:

```
[alg_wc_ccf_translate lang="EN,DE" lang_text="Text for English and German" not_lang_text="Text for other languages"]
```

Use `lang` for an inclusion list and `not_lang` for an exclusion list. Language codes are comma-separated and case-insensitive.

3. Conditional value

`[alg_wc_ccf_if]` returns one value or another based on a comparison. Inside it, wrap nested shortcodes in curly braces so they resolve correctly.

```
[alg_wc_ccf_if value1="{alg_wc_ccf_datetime format='Gi'}" operator="greater" value2="1300" then="1" else="0"]
```

Attribute	Meaning
`value1`, `value2`	The two values to compare
`operator`	`equal`, `not_equal`, `less`, `less_or_equal`, `greater`, `greater_or_equal`
`then`	Returned when the comparison is true (or use the enclosed content)
`else`	Returned when the comparison is false



CUSTOM CHECKOUT FIELDS FOR WOOCOMMERCE

See every feature and the pricing on the Custom Checkout Fields for WooCommerce plugin page.

View plugin →

The example above returns `1` after 1:00 PM and `0` before it, which you could feed into a datepicker Min date to shift ordering cut-offs by time of day.

Need a hand?

Our team answers every ticket on the Custom Checkout Fields support forum.

Related articles

- Where Custom Checkout Field Data Is Stored (Order Meta Reference)
- Custom Checkout Fields in Emails and Order Pages: Templates and Placeholders
- General Troubleshooting for Custom Checkout Fields
- What You Get with Custom Checkout Fields Pro
- Adding a Checkout Fee Based on a Custom Field