Text

- Screenshots
- Settings
- Data
- Template usage

import Screenshots from '@site/src/components/Screenshots';

The text field creates a simple text input. You can enter any value or select a value from a predefined list.

:::info Input types

This docs is also used for **URL** and **email** field types as they share the same settings. If you want to have full text editing abilities, checkout the WYSIWYG field.

:::

Screenshots

<Screenshots name="text" col1={[['https://imgur.elightup.com/WnedDrd.png', 'The text field interface']]} />

Settings

Besides the common settings, this field has the following specific settings, the keys are for use with code:

Name	Key	Description
Type	`type`	Allowed values: `text`, `url` or `email`. If the value is `url` or `email`, the field value will be validated accordingly.

Name	Key	Description
Placeholder	`placeholder`	Placeholder for the input. Optional.
Size of the input box	`size`	Size of the input box. Without this setting, the input box is full-width.
Prepend text	`prepend`	Prepend text for the input. The prepended text should be considered as an informative label. It's not saved in the database with the input value.
Append text	`append`	Append text for the input. The appended text should be considered as an informative label. It's not saved in the database with the input value.
Predefined values	`datalist`	Predefined values that users can select from (users still can enter text if they want). If you use code, this parameter has the following sub-parameters - `id`: ID of the div that stores the options. Usually not used and auto-generated as `{$field['id']_list`. Useful if you have several inputs with same datalist. - `options`: Array of predefined values to select from.

This is a sample field settings array when creating this field with code:

```
[
    'name'        => 'Text',
    'id'          => 'text',
    'desc'        => 'Please enter some text above',
    'type'        => 'text',
    'std'         => 'Default text value',
    'placeholder' => 'Enter something here',
    'datalist'    => [
        'id'      => 'text_datalist',
        'options' => [
            'What',
            'When',
            'Where',
            'Why',
            'Who',
        ],
    ],
```

```
],
```

:::tip "type" is optional

You can omit the "type" setting. The plugin automatically sets it to "text" if missing.

```
add_filter( 'rwmb_meta_boxes', function ( $meta_boxes ) {
    $meta_boxes[] = [
        'title'      => 'Event details',
        'post_types' => 'event',
        'fields'     => [
            [
                'name' => 'Date and time',
                'id'   => 'datetime',
                'type' => 'datetime',
            ],
            // highlight-start
            [
                'name' => 'Location',
                'id'   => 'location',
            ],
            // highlight-end
            [
                'name'          => 'Map',
                'id'            => 'map',
                'type'          => 'osm',
                'address_field' => 'location',
            ],
        ],
    ];

    return $meta_boxes;
} );
```

:::

:::tip Quick define a text field

If your text field only need a label, then you can define it with code simply by the label text. In this case, the field id is auto generated from the label:

```
add_filter( 'rwmb_meta_boxes', function ( $meta_boxes ) {
    $meta_boxes[] = [
        'title'      => 'Event details',
        'post_types' => 'event',
        'fields'     => [
            [
                'name' => 'Date and time',
                'id'   => 'datetime',
                'type' => 'datetime',
            ],
            // highlight-next-line
            'Location', // id is auto-generated to "location"
            [
                'name'          => 'Map',
                'id'            => 'map',
                'type'          => 'osm',
                'address_field' => 'location',
            ],
        ],
    ];

    return $meta_boxes;
} );
```

:::

Data

This field saves the entered value into the database.

If the field is cloneable, then the value is stored as a serialized array in a single row in the database.

Template usage

Displaying the value:

```
<p>Entered: <?php rwmb_the_value( 'my_field_id' ) ?></p>
```

Getting the value:

```
<?php $value = rwmb_meta( 'my_field_id' ) ?>
<p>Entered: <?= $value ?></p>
```

Displaying cloneable values:

```
<?php $values = rwmb_meta( 'my_field_id' ) ?>
<?php foreach ( $values as $value ) : ?>
    <p><?= $value ?></p>
<?php endforeach ?>
```

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:17 UTC
□ □□□ □□□ 2026-08-29 15:36:17 UTC