Sanitizing Data

- Example
- Sanitization functions

Untrusted data comes from many sources (users, third party sites, even your own database!) and all of it needs to be checked before it's used.

Remember: Even admins are users, and users will enter incorrect data, either on purpose or accidentally. It's your job to protect them from themselves.

Sanitizing input is the process of securing/cleaning/filtering input data. Validation is preferred over sanitization because validation is more specific. But when "more specific" isn't possible, sanitization is the next best thing.

Example

Let's say we have an input field named `title`:

```
<input id="title" type="text" name="title">
```

We can't use Validation here because the text field is too general: it can be anything at all. So we sanitize the input data with the `sanitize_text_field()` function:

```
$title = sanitize_text_field( $_POST['title'] );
update_post_meta( $post->ID, 'title', $title );
```

Behind the scenes, `sanitize_text_field()` does the following:

1. Checks for invalid UTF-8
2. Converts single less-than characters (<) to entity
3. Strips all tags
4. Removes line breaks, tabs and extra white space
5. Strips octets

Sanitization functions

There are many functions that will help you sanitize your data.

- `sanitize_email()`
- `sanitize_file_name()`
- `sanitize_hex_color()`
- `sanitize_hex_color_no_hash()`
- `sanitize_html_class()`
- `sanitize_key()`
- `sanitize_meta()`
- `sanitize_mime_type()`
- `sanitize_option()`
- `sanitize_sql_orderby()`
- `sanitize_term()`
- `sanitize_term_field()`
- `sanitize_text_field()`
- `sanitize_textarea_field()`
- `sanitize_title()`
- `sanitize_title_for_query()`
- `sanitize_title_with_dashes()`
- `sanitize_user()`
- `sanitize_url()`
- `wp_kses()`
- `wp_kses_post()`

□□□ #2
□ □□□ □□□ 2026-10-31 18:07:50 UTC
□ □□□ □□□ 2026-10-31 18:07:51 UTC