










- 
- 
- 
- 
- 
- 
- 



title 00000000

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

sanitize\_text\_field() 00000000000000000000000000000000

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

sanitize\_text\_field() 000000

1. 0000 UTF-8
2. 0000 (<) 0000
3. 0000
4. 0000000000000000
5. 0000



- [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()`
- `wpkses()`
- `wpkses_post()`



- 
- 
- 
- 



```
$untrusted_input = '1 malicious string'; // will evaluate to integer 1 during loose comparisons

if ( 1 === $untrusted_input ) { // == would have evaluated to true, but === evaluates to false
    echo '<p>Valid data';
} else {
    wp_die( 'Invalid data' );
}
```

in\_array()

```

$untrusted_input = '1 malicious string'; // will evaluate to integer 1 during loose comparisons
$safe_values     = array( 1, 5, 7 );

if ( in_array( $untrusted_input, $safe_values, true ) ) { // `true` enables strict type checking
    echo '<p>Valid data';
} else {
    wp_die( 'Invalid data' );
}

```

## switch()

```

$untrusted_input = '1 malicious string'; // will evaluate to integer 1 during loose comparisons

switch ( true ) {
    case 1 === $untrusted_input: // do your own strict comparison instead of relying on switch()'s loose
comparison
        echo '<p>Valid data';
        break;

    default:
        wp_die( 'Invalid data' );
}

```



```

if ( ! ctype_alnum( $data ) ) {
    wp_die( "Invalid format" );
}

if ( preg_match( "/^[^0-9.-]"/, $data ) ) {
    wp_die( "Invalid format" );
}

```



XXXXXXXXXXXXXXXXXX

```
$trusted_integer = (int) $untrusted_integer;
$trusted_alpha = preg_replace( '/[^a-z]/i', "", $untrusted_alpha );
$trusted_slug = sanitize_title( $untrusted_slug );
```

XXX

XXXXXXXXXXXXXXXXXX

```
<input type="text" id="wporg_zip_code" name="my-zipcode" maxlength="10" />
```

XXX11221eval()XXXXX.....XXXXXX

XX

my-zipcode XXXXXXXXX

```
/**
 * Validate a US zip code.
 *
 * @param string $zip_code RAW zip code to check.
 *
 * @return bool true if valid, false otherwise.
 */
function wporg_is_valid_us_zip_code( string $zip_code ):bool {
    // Scenario 1: empty.
    if ( empty( $zip_code ) ) {
        return false;
    }

    // Scenario 2: more than 10 characters.
    // The `maxlength` attribute is only enforced by
    // the browser, so we still need to validate the
    // length of the input on the server to protect
    // against a manual submission.
    if ( 10 < strlen( trim( $zip_code ) ) ) {
        return false;
    }

    // Scenario 3: incorrect format.
```



- `sanitize_html_class( $class, $fallback )` - `html` `AZ,az,0-9,-'`
- `tag_escape( $html_tag_name )` - `HTML`
- `term_exists()`
- `username_exists()`
- `validate_file()`

**WordPress** `*_exists()` `*_validate()` `is_*`





WordPress HTML

WordPress



WordPress

WordPress HTML

- `esc_html()` - HTML HTML

```
<h4><?php echo esc_html( $title ); ?></h4>
```

- `esc_js()` - Javascript

```
<div onclick='<?php echo esc_js( $value ); ?>' />
```

- `esc_url()` - URL HTML `src` `href` URL

```

```

- `esc_url_raw()` - URL URL
- `esc_xml()` - XML
- `esc_attr()` - HTML

```
<ul class="<?php echo esc_attr( $stored_class ); ?>">
```

- `esc_textarea()` -
- `wp_kses()` - HTML HTML
- `wp_kses_post()` - `wp_kses()` HTML
- `wp_kses_data()` - `wp_kses()` HTML



`wp_kses()` "Kisses"

HTML HTML

```
<?php
echo wp_kses_post( $partial_html );
echo wp_kses(
    $another_partial_html,
    array(
        'a' => array(
            'href' => array(),
            'title' => array(),
        ),
        'br' => array(),
        'em' => array(),
        'strong' => array(),
    )
); ?>
```

`<a>` `<br>` `<em>` `<strong>` `<a>` href the title

## escape

- 
- 
- 
- 
- 

```
// Okay, but not great.
$url = esc_url( $url );
$text = esc_html( $text );
echo '<a href="'. $url . '"'>' . $text . '</a>';

// Much better!
echo '<a href="'. esc_url( $url ) . '"'>' . esc_html( $text ) . '</a>';
```

.....

`wp_kses()`

`_escaped` ☐ `_safe` ☐ `_clean` ☐ `$variable`, `$variable_escaped` or `$variable_safe` ☐

`echo my_custom_script_code();` ☒ `wp_kses()`

☐ WordPress `echo _e()` `_()`

```
esc_html_e( 'Hello World', 'text_domain' );  
// Same as  
echo esc_html( __( 'Hello World', 'text_domain' ) );
```

- [esc\\_html\\_\\_\(\)](#)
- [esc\\_html\\_e\(\)](#)
- [esc\\_html\\_x\(\)](#)
- [esc\\_attr\\_\\_\(\)](#)
- [esc\\_attr\\_e\(\)](#)
- [esc\\_attr\\_x\(\)](#)

```
echo $int;
```

☐ (int) `absint()` ☐ (float)

`number_format()` ☐ `number_format_i18n()`

`intval()` ☐ `floatval()`  (PHP4)

HTML

```
echo '<div id="", $prefix, '-box', $id, "">';
```

`esc_attr()`

url

```
echo '<div id="", esc_attr( $prefix . '-box' . $id ), "">';
```

```
echo '<div id="", esc_attr( $prefix ), '-box', esc_attr( $id ), "">';
```

`wp_create_nonce()` ☐ HTML ☐

□ HTML □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ URL

```
echo '<a href="", $url, "">';
```

esc\_url()

--	--	--	--

```
echo '<a href="", esc_url( $url ), "">';
```

111

```
echo '<a href="", esc_attr( $url ), "">';  
echo '<a href="", esc_attr( esc_url( $url ) ), "">';
```

JavaScript wp\_localize\_script()

```
wp_localize_script( 'handle', 'name',
    array(
        'prefix_nonce' => wp_create_nonce( 'plugin-name' ),
        'ajaxurl'      => admin_url( 'admin-ajax.php' ),
        'errorMsg'      => __( 'An error occurred', 'plugin-name' ),
    )
);
```

WordPress

JavaScript

```
<script type="text/javascript">
    var myVar = <?php echo $my_var; ?>
</script>
```

```
$my_var  [] esc_js() []
```

[[[[]]]]

```
<script type="text/javascript">
    var myVar = <?php echo esc_js( $my_var ); ?>
</script>
```

## [[[[]]]] JavaScript [[[[[[]]]]]]

```
<a href="#" onclick="do_something(<?php echo $var; ?>); return false;">
```

```
$var  [] esc_js() []
```

[[[[]]]]

```
<a href="#" onclick="do_something(<?php echo esc_js( $var ); ?>); return false;">
```

## [[[]] HTML [[[[[[]]]]]] JavaScript [[[]]

```
<a href="#" data-json="<?php echo $var; ?>">>
```

```
$var  [] esc_js(), json_encode() [] wp_json_encode() []
```

[[[[]]]]

```
<a href="#" data-json="<?php echo esc_js( $var ); ?>">
```

## [] HTML [[[[[[]]]]]]

```
echo '<textarea>', $data, '</textarea>';
```

```
$data  [] esc_textarea() []
```

[[[[]]]]

```
echo '<textarea>', esc_textarea( $data ), '</textarea>';
```

## HTML 过滤器

```
echo '<div>', $phrase, '</div>';
```

`$phrase` HTML

- `esc_html()`
- HTML `wp_kses_post()`, `wp_kses_allowed_html()` `wp_kses()` HTML

## XML 和 XSL 过滤器

```
echo '<loc>', $var, '</loc>';
```

```
echo '<loc>', ent2ncr( $var ), '</loc>';
```



“ ” URL

WordPress “ ”

WordPress “current\_user\_can()” CSRF



URL URL 123

```
http://example.com/wp-admin/post.php?post=123&action=trash
```

URL WordPress cookie URL

```

```

WordPress cookie WordPress

WordPress URL

```
http://example.com/wp-admin/post.php?post=123&action=trash&_wpnonce=b192fc4204
```

WordPress 123 WordPress “403 Forbidden”



URL

AJAX JavaScript



# URL

wp\_nonce\_url() URL

```
$complete_url = wp_nonce_url( $bare_url, 'trash-post_'. $post->ID );
```

wp\_nonce\_url() \_wpnonce

```
$complete_url = wp_nonce_url( $bare_url, 'trash-post_'. $post->ID, 'my_nonce' );
```

wp\_nonce\_field() wp\_nonce\_field() URL

```
wp_nonce_field( 'delete-comment_'. $comment_id );
```

```
<input type="hidden" id="_wpnonce" name="_wpnonce" value="796c7766b1" />
<input type="hidden" name="_wp_http_referer" value="/wp-admin/edit-comments.php" />
```

no wp\_nonce\_field()

wp\_create\_nonce()

```
$nonce = wp_create_nonce( 'my-action_'. $post->ID );
```

295a686963

URLAJAX



check\_admin\_referer()

--	--	--

```
check_admin_referer( 'delete-comment_'.$comment_id );
```

“403 Forbidden”

```

    wpnonce )

```

--	--	--

```
check_admin_referer( 'delete-comment_'.$comment_id, 'my_nonce' );
```

[illegible]

check\_ajax\_referer()

```
check_ajax_referer( 'process-comment' );
```

[illegible]

\_wpnonce \_ajax\_nonce check\_ajax\_referer()

wp\_verify\_nonce()

--	--	--

```
wp_verify_nonce( $_REQUEST['my_nonce'], 'process-comment'.$comment_id );
```

wp\_nonce\_ays() "403 Forbidden"

## nonce

--	--	--	--	--	--	--	--

nonce\_life

```
add_filter( 'nonce_life', function () { return 4 * HOUR_IN_SECONDS; } );
```

check\_admin\_referrer() check\_admin\_referer

```
function wporg_additional_check ( $action, $result ) {  
    ...  
}  
add_action( 'check_admin_referer', 'wporg_additional_check', 10, 2 );
```

check\_ajax\_referer() check\_ajax\_referer

```
function my_nonce_message ($translation) {  
    if ($translation === 'Are you sure you want to do this?') {  
        return 'No! No! No!';  
    }  
  
    return $translation;  
}  
  
add_filter('gettext', 'my_nonce_message');
```

WordPress

WordPress “”WordPress

1 2

WcNONCE\_KEY NONCE\_SALT wp-config.php

current\_user\_can()

check\_admin\_referrer() check\_ajax\_referrer()

wp\_create\_nonce() wp\_verify\_nonce() wp\_nonce\_tick()

wp\_nonce\_ays(), wp\_nonce\_field(), wp\_nonce\_url(), wp\_verify\_nonce(), wp\_create\_nonce(),  
check\_admin\_referer(), check\_ajax\_referer(), wp\_referer\_field()

nonce\_life, nonce\_user\_logged\_out, explain\_nonce\_verb-noun, check\_admin\_referer

--	--	--	--	--	--	--

--	--	--	--	--	--	--

[illegible]

--	--	--	--	--	--	--

WordPress

WordPress

[illegible]

```
00000000 "manage_options" 000000000000000000000000000000000000000000000000000
```

WordPress

[illegible]

--	--	--	--

[illegible][illegible]

--	--

--	--	--

/\*\*

\* Generate a Delete link based on the homepage url.

```

*
* @param string $content Existing content.
*
* @return string|null
*/
function wporg_generate_delete_link( $content ) {
    // Run only for single post page.
    if ( is_single() && in_the_loop() && is_main_query() ) {
        // Add query arguments: action, post.
        $url = add_query_arg(
            [
                'action' => 'wporg_frontend_delete',
                'post' => get_the_ID(),
            ], home_url()
        );

        return $content . ' <a href="' . esc_url( $url ) . '"' > . esc_html__( 'Delete Post', 'wporg' ) . '</a>';
    }

    return null;
}

/**
 * Request handler
 */
function wporg_delete_post() {
    if ( isset( $_GET['action'] ) && 'wporg_frontend_delete' === $_GET['action'] ) {

        // Verify we have a post id.
        $post_id = ( isset( $_GET['post'] ) ) ? ( $_GET['post'] ) : ( null );

        // Verify there is a post with such a number.
        $post = get_post( (int) $post_id );
        if ( empty( $post ) ) {
            return;
        }

        // Delete the post.
        wp_trash_post( $post_id );
    }
}

```

```

// Redirect to admin page.
$redirect = admin_url( 'edit.php' );
wp_safe_redirect( $redirect );

```

□□// We are done.

```
    die;
```

□ }

}

/ \*\*

\* Add the delete link to the end of the post content.

 $\ast/$ 

```
add_filter( 'the_content', 'wporg_generate_delete_link' );
```

/ \*\*

- \* Register our request handler with the init hook.

 $\ast/$ 

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

--	--	--	--	--	--	--

[illegible]

edit\_others\_posts ☐

/ \*\*

\* Generate a Delete link based on the homepage url.

\*

\* @param string \$content Existing content.

\*

```
* @return string|null
```

 $\ast/$ 

```
function wporg_generate_delete_link( $content ) {
```

☐ // Run only for single post page.

```
if ( is_single() && in_the_loop() && is_main_query() ) {
```

```
// Add query arguments: action, post.
```

```
url = add_query_arg(
```

```

        'action' => 'wporg_frontend_delete',
        'post' => get_the_ID(),
    ], home_url()
    );

    return $content . ' <a href="' . esc_url( $url ) . '"' . esc_html__( 'Delete Post', 'wporg' ) . '</a>';
}

return null;
}

/**
 * Request handler
 */
function wporg_delete_post() {
    if ( isset( $_GET['action'] ) && 'wporg_frontend_delete' === $_GET['action'] ) {

        // Verify we have a post id.
        $post_id = ( isset( $_GET['post'] ) ) ? ( $_GET['post'] ) : ( null );

        // Verify there is a post with such a number.
        $post = get_post( (int) $post_id );
        if ( empty( $post ) ) {
            return;
        }

        // Delete the post.
        wp_trash_post( $post_id );

        // Redirect to admin page.
        $redirect = admin_url( 'edit.php' );
        wp_safe_redirect( $redirect );

        // We are done.
        die;
    }
}

```

```
/**
 * Add delete post ability
 */
add_action('plugins_loaded', 'wporg_add_delete_post_ability');

function wporg_add_delete_post_ability() {
    if ( current_user_can( 'edit_others_posts' ) ) {
        /**
         * Add the delete link to the end of the post content.
         */
        add_filter( 'the_content', 'wporg_generate_delete_link' );

        /**
         * Register our request handler with the init hook.
         */
        add_action( 'init', 'wporg_delete_post' );
    }
}
```





# SQL

SQL `add_post_meta();` `INSERT INTO wp_postmeta...`

`table` not found or type unknown

xkcd

SQL WordPress

`$wpdb`

SQL `$wpdb->prepare()` `sqlf()` `sqlf()`

```
$wpdb->get_var( $wpdb->prepare(
    "SELECT something FROM table WHERE foo = %s and status = %d",
    $name, // an unescaped string (function will do the sanitization for you)
    $status // an untrusted integer (function will do the sanitization for you)
));
```

# (XSS)

JavaScript (XSS)

XSS

URL

```

```

HTML HTML

```
$allowed_html = array(
    'a' => array(
        'href' => array(),
        'title' => array()
    ),
    'br' => array(),
    'em' => array(),
    'strong' => array(),
);

echo wp_kses( $custom_content, $allowed_html );
```

(CSRF)

CSRF sea-surf Web Work

HTML HTTP

```
<form method="post">
    <!-- some inputs here ... -->
    <?php wp_nonce_field( 'name_of_my_action', 'name_of_nonce_field' ); ?>
</form>
```

- WordPress
- WordPress
- Web (OWASP) 10



```
/**
 * Generate a Delete link based on the homepage url.
 *
 * @param string $content Existing content.
 *
 * @return string|null
 */
function wporg_generate_delete_link( $content ) {
    // Run only for single post page.
    if ( is_single() && in_the_loop() && is_main_query() ) {
        // Add query arguments: action, post, nonce
        $url = add_query_arg(
            [
                'action' => 'wporg_frontend_delete',
                'post' => get_the_ID(),
                'nonce' => wp_create_nonce( 'wporg_frontend_delete' ),
            ], home_url()
        );

        return $content . ' <a href="' . esc_url( $url ) . '"> . esc_html__( 'Delete Post', 'wporg' ) . '</a>';
    }

    return null;
}

/**
 * Request handler
 */
```

```
function wporg_delete_post() {  
    if ( isset( $_GET['action'] )  
        && isset( $_GET['nonce'] )  
        && 'wporg_frontend_delete' === $_GET['action']  
        && wp_verify_nonce( $_GET['nonce'], 'wporg_frontend_delete' ) ) {
```

```
        // Verify we have a post id.
```

```
        $post_id = ( isset( $_GET['post'] ) ) ? ( $_GET['post'] ) : ( null );
```

```
        // Verify there is a post with such a number.
```

```
        $post = get_post( (int) $post_id );
```

```
        if ( empty( $post ) ) {
```

```
            return;
```

```
        }
```

```
        // Delete the post.
```

```
        wp_trash_post( $post_id );
```

```
        // Redirect to admin page.
```

```
        $redirect = admin_url( 'edit.php' );
```

```
        wp_safe_redirect( $redirect );
```

```
        // We are done.
```

```
        die;
```

```
    }
```

```
}
```

```
/**
```

```
 * Add delete post ability
```

```
 */
```

```
add_action('plugins_loaded', 'wporg_add_delete_post_ability');
```

```
function wporg_add_delete_post_ability() {
```

```
    if ( current_user_can( 'edit_others_posts' ) ) {
```

```
        /**
```

```
         * Add the delete link to the end of the post content.
```

```
         */
```

```
        add_filter( 'the_content', 'wporg_generate_delete_link' );
```

```
/**  
 * Register our request handler with the init hook.  
 */  
add_action( 'init', 'wporg_delete_post' );  
}  
}
```