

10* useful WordPress functions

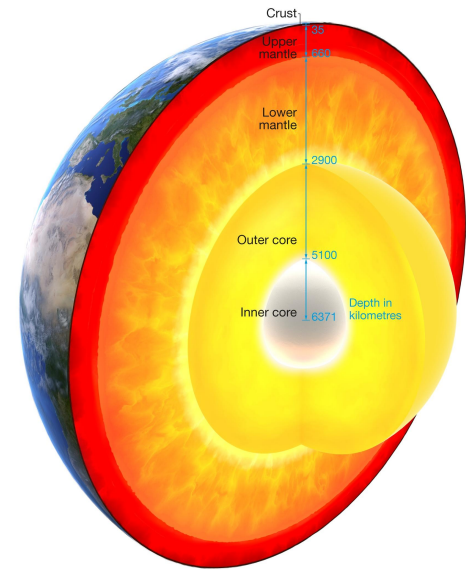
@jubstuff ~ borzacchiello.it
developer @ **DRIVEK**

A group of young people in a dimly lit club or bar. In the center, a man is shirtless and smoking a cigarette. To his left, another man wears a white tank top. To the right, a woman is wearing a dark bikini top. The background is dark with some blurred lights.

The #1 rule of WordPress development

DON'T TOUCH THE CORE

too hot!



The core

`/wp-admin`

Functions and templates to bootstrap the WordPress admin.

`/wp-includes`

Most WordPress functionality is here.

DON'T TOUCH THE CORE

... but you can read it!

checked()

Compares the first two arguments and if identical marks as checked.

checked()

```
<input name="my_control" type="radio"  
value="my_value"  
<?php  
echo ($value === 'my_value') ?  
'checked="checked"' :  
' ' ?> >
```


checked()

```
<input name="my_control" type="radio"  
value="my_value"  
<?php checked( 'my_value', $value ); ?> >
```

selected()

Compares the first two arguments and if identical marks as selected.

disabled()

Compares the first two arguments and if identical marks as disabled.

`esc_*`

Escaping functions.



`esc_*()`

Escaping functions.

`<` becomes **`<`**;

esc_*()

- `esc_sql`
- `esc_url`
- `esc_url_raw`
- `esc_js`
- `esc_html`
- `esc_attr`
- `esc_textarea`

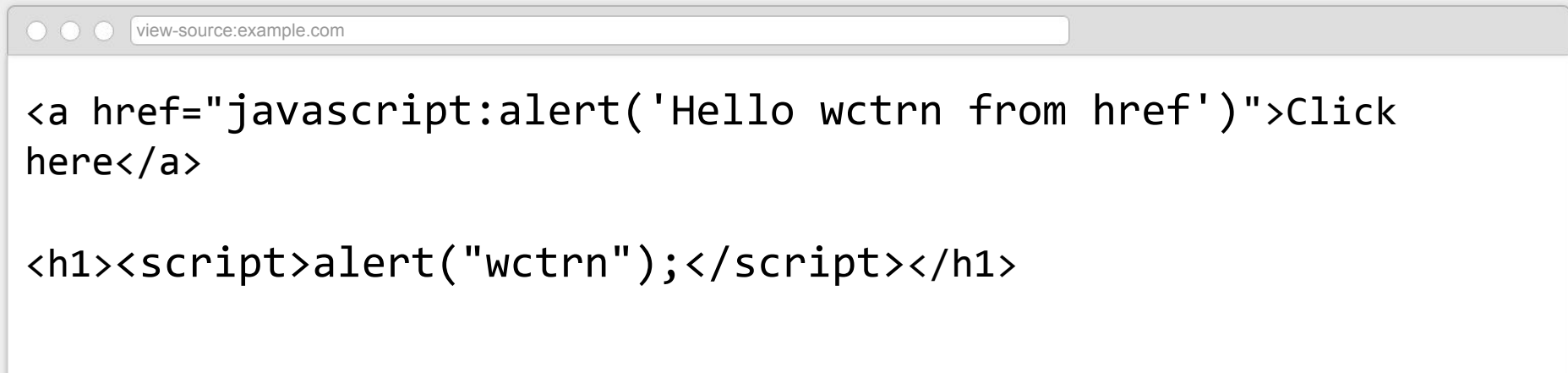
Example: `esc_attr()`

```
<?php // Fictional malicious input...
$href = "javascript:alert('Hello wctrn from href')";
$title = '<script>alert("wctrn");</script>';

// ...and displaying it ?>
<a href="<?php echo $href; ?>">Click here</a>
<h1><?php echo $title; ?></h1>
```

Example: `esc_attr()`

```
<?php // Fictional malicious input...  
$href = "javascript:alert('Hello wctrn from href')";  
$title = '<script>alert("wctrn");</script>';  
  
// ...and displaying it ?>  
<a href="<?php echo $href; ?>">Click here</a>  
<h1><?php echo $title; ?></h1>
```



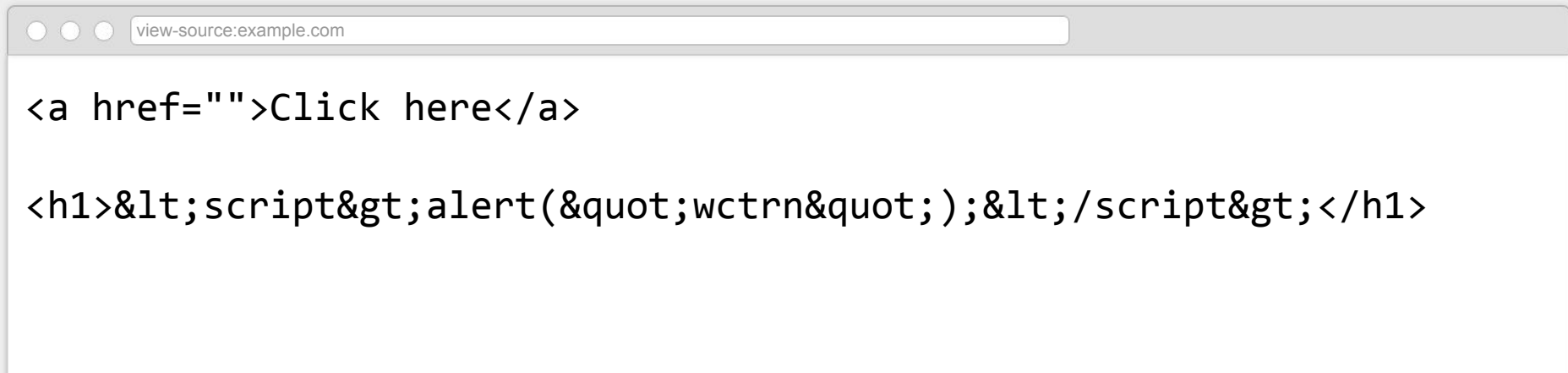
Example: `esc_attr()`

```
<?php // Fictional malicious input...
$href = "javascript:alert('Hello wctrn from href')";
$title = '<script>alert("wctrn");</script>';

// ...and displaying it ?>
<a href="<?php echo esc_url( $href ); ?>">Click here</a>
<h1><?php echo esc_html( $title ); ?></h1>
```

Example: `esc_attr()`

```
<?php // Fictional malicious input...  
$href = "javascript:alert('Hello wctrn from href')";  
$title = '<script>alert("wctrn");</script>';  
  
// ...and displaying it ?>  
<a href="<?php echo esc_url( $href ); ?>">Click here</a>  
<h1><?php echo esc_html( $title ); ?></h1>
```



The screenshot shows a browser window with the address bar containing "view-source:example.com". The main content area displays the rendered HTML code, where the PHP code has been converted to its output. The href attribute is now an empty string, and the title attribute contains the escaped script code.

```
<a href="">Click here</a>  
  
<h1>&lt;script&gt;alert(&quot;wctrn&quot;);&lt;/script&gt;</h1>
```

`wp_is_mobile()`

Test if the current browser runs on a mobile device (smart phone, tablet, etc.).

wp_is_mobile()

```
function add_my_cool_js_effect_on_desktop() {  
    if ( wp_is_mobile() ) {  
        return;  
    }  
  
    wp_enqueue_script( 'cool-js-effect',  
                      PATH_TO_MY_JS );  
}
```

wp_no_robots()

Display a noindex meta tag.

wp_no_robots()

```
/**
 * Display a noindex meta tag.
 * @since 3.3.0
 */
function wp_no_robots() {
    echo "<meta name='robots'
        content='noindex, follow' />\n";
}
```

wp_no_robots()

```
// Add this to block search engines on page named  
'no-search'  
add_action( 'init', function() {  
    if ( is_page( 'no-search' ) ) {  
        add_action( 'wp_head', 'wp_no_robots' );  
    }  
} );
```

wp_parse_args()

Merge user defined arguments
into defaults array.

wp_parse_args()

```
<?php
```

```
$defaults = [ 'count' => 5, 'orderby' => 'date',  
'order' => 'asc' ];
```

```
$params = [ 'orderby' => 'title',  
'order' => 'desc' ];
```

```
$options = wp_parse_args( $params, $defaults );
```

wp_parse_args()

```
<?php
```

```
$defaults = [ 'count' => 5, 'orderby' => 'date',  
'order' => 'asc' ];
```

```
$params = [ 'orderby' => 'title',  
'order' => 'desc' ];
```

```
$options = wp_parse_args( $params, $defaults );
```

```
$options = [  
    'count' => 5,  
    'orderby' => 'title',  
    'order' => 'desc',  
]
```

wp_parse_args()

```
<?php
```

```
$defaults = [ 'count' => 5, 'orderby' => 'date',  
'order' => 'asc' ];
```

```
$params = [ 'orderby' => 'title',  
'order' => 'desc' ];
```

```
$options = wp_parse_args( $params, $defaults );
```

```
$options = [  
    'count' => 5,  
    'orderby' => 'title',  
    'order' => 'desc',  
]
```

wp_parse_args()

```
<?php
```

```
$defaults = [ 'count' => 5, 'orderby' => 'date',  
'order' => 'asc' ];
```

```
$params = [ 'orderby' => 'title',  
'order' => 'desc' ];
```

```
$options = wp_parse_args( $params, $defaults );
```

```
$options = [  
    'count' => 5,  
    'orderby' => 'title',  
    'order' => 'desc',  
]
```

wp_parse_args()

```
// MyWidget.php
public function form( $instance ) {
    $defaults = array(
        'num_entries' => 10,
        'widget_title' => __('My title', 'my-domain'),
        'scale'       => 10,
    );

    $instance = wp_parse_args( $instance, $defaults);

    // display the form
}
```

wp_send_json_success()

Send a JSON response back to an Ajax request, indicating success.

wp_send_json_success()

```
jQuery(document).ready(function($) {  
    var data = {'action': 'my_action', 'post_id': 4};  
  
    jQuery.post(my_ajax_url, data, function(response){  
        // AJAX Callback  
    });  
});
```

wp_send_json_success()

```
add_action( 'wp_ajax_nopriv_my_action',  
'my_ajax_handler' );
```

```
function my_ajax_handler() {  
    $post_id = intval( $_POST['post_id'] );
```

```
//Retrieve some $output_data related to that post
```


```
wp_send_json_success( $output_data );  
}
```


wp_send_json_success()

```
{  
    success: true,  
    data: $output_data  
}
```

wp_send_json_success()

```
jQuery.post(my_ajax_url, data, function(response){  
    // AJAX Callback  
    if ( response.success ) {  
        alert( 'This is from PHP: ' + response.data );  
    }  
});
```



```
{  
    success: true,  
    data: $output_data  
}
```

`wp_send_json_error()`

Send a JSON response back to an Ajax request, indicating failure.

`wp_send_json()`

Send a JSON response back to an Ajax request.

Where do I start?

`/wp-includes/general-template.php`

Mostly template tags.

`/wp-includes/functions.php`

Functions, functions everywhere.

`/wp-includes/formatting.php`

Strings, dates, and general formatting functions.

`/wp-includes/pluggable.php`

Overwritable functions. You can be creative, if you know what you are doing.



Code Reference

Browse: [Home](#) / Reference

Want to know what's going on inside WordPress? Search the Code Reference for more information about WordPress' functions, classes, methods, and hooks.

Try it out:

Search

Filter by type: Functions | Hooks | Classes | Methods

Or browse through topics:

[Functions](#) | [Hooks](#) | [Classes](#) | [Methods](#)

NEW & UPDATED IN WORDPRESS 4.4:

API

Developer reference

<https://developer.wordpress.org/reference/>

takeaways

Read the Core

Understand

Contribute

Try to find these functions:

`wp_list_pluck()`

`wp_list_filter()`

`wp_extract_urls()`

`make_clickable()`

`add_query_arg()`

`_split_str_by_whitespace()`

`wp_remote_get()`

`wp_parse_id_list()`

Grazie :))

@jubstuff ~ borzacchiello.it