


Advanced Actions & Filters



```
... string $tag The name of the action.  
... param mixed $arg,... Options for the action.  
... action.  
* @return null Will return null if no actions are found.  
*/  
function do_action($tag, $arg = '') {  
    global $wp_filter, $wp_actions, $wp_current_filter;  
  
    if ( ! isset($wp_actions) )  
        $wp_actions = array();  
  
    if ( ! isset($wp_actions[$tag]) )  
        $wp_actions[$tag] = 1;  
    else  
        ++$wp_actions[$tag];  
  
    // Do 'all' actions first  
    if ( isset($wp_filter['all']) ) {  
        $wp_current_filter[] = $tag;  
        $all_args = func_get_args();  
        _wp_call_all_hook($all_args, $tag);  
    }  
}
```



Alison Barrett

WordPress.com VIP Team,
Automattic

@alisothegeek

<http://alisothegeek.com>

actions + filters =
hooks

how hooks work

action

“do something”

filter

“do something to something”

action

do_action

filter

apply_filters

action

add_action & remove_action

filter

add_filter & remove_filter

...but actions *are* filters.

```
function add_action($tag, $function_to_add, $priority = 10,  
$accepted_args = 1) {  
    return add_filter($tag, $function_to_add, $priority,  
$accepted_args);  
}
```


action

“do something **to nothing**”

filter

“do something to something”

```
function add_filter($tag, $function_to_add, $priority = 10, $accepted_args = 1) {  
    global $wp_filter, $merged_filters;  
  
    $idx = _wp_filter_build_unique_id($tag, $function_to_add, $priority);  
    $wp_filter[$tag][$priority][$idx] = array('function' => $function_to_add, 'accepted_args' => $accepted_args);  
    unset( $merged_filters[ $tag ] );  
    return true;  
}
```

add_filter()

understanding add_filter()

put code in
functions.php



WordPress
does magic



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

```
add_filter( 'the_content', 'show_author_bio', 10, 1 );
```

priority

no. of arguments

```
function add_filter(  
    $tag, $function_to_add, $priority = 10,  
    $accepted_args = 1  
) {
```

```
$tag = 'the_content' $priority = 10 (default)
```

```
$function_to_add = 'show_author_bio' $accepted_args = 1 (default)
```

```
global $wp_filter, $merged_filters;
```

```
$idx = _wp_filter_build_unique_id(  
    $tag, $function_to_add, $priority  
);
```

```
global $wp_filter, $merged_filters;
```

```
$idx = _wp_filter_build_unique_id(  
    $tag, $function_to_add, $priority  
);
```

```
function _wp_filter_build_unique_id(...) {  
    ...  
    if ( is_string($function) )  
        return $function;  
    ...  
}
```

```
$wp_filter[$tag][$priority][$idx] = array(
    'function'      => $function_to_add,
    'accepted_args' => $accepted_args
);
unset( $merged_filters[ $tag ] );
return true;
}
```

```
$wp_filter['the_content'][10]['show_author_bio']
= array(
    'function'      => 'show_author_bio',
    'accepted_args' => 1
);
```



```

function apply_filters($tag, $value) {
    global $wp_filter, $merged_filters, $wp_current_filter;

    $args = array();

    // Do 'all' actions first
    if ( isset($wp_filter['all']) ) {
        $wp_current_filter[] = $tag;
        $args = func_get_args();
        _wp_call_all_hook($args);
    }

    if ( !isset($wp_filter[$tag]) ) {
        if ( isset($wp_filter['all']) )
            array_pop($wp_current_filter);
        return $value;
    }

    if ( !isset($wp_filter['all']) )
        $wp_current_filter[] = $tag;

    // Sort
    if ( !isset( $merged_filters[ $tag ] ) ) {
        ksort($wp_filter[$tag]);
        $merged_filters[ $tag ] = true;
    }

    reset( $wp_filter[ $tag ] );

    if ( empty($args) )
        $args = func_get_args();

    do {
        foreach( (array) current($wp_filter[$tag]) as $the_ )
            if ( !is_null($the_['function']) ){
                $args[1] = $value;
                $value = call_user_func_array($the_['function'], array_slice($args, 1, (int) $the_['accepted_args']));
            }
    } while ( next($wp_filter[$tag]) !== false );

    array_pop( $wp_current_filter );

    return $value;
}

```

apply_filters()

```
apply_filters( 'the_content', $content );
```

```
apply_filters( 'the_category', $thelist, $separator, $parents );
```

what to filter

extra information

```
function apply_filters($tag, $value) {  
    global $wp_filter, $merged_filters,  
    $wp_current_filter;
```

```
    $args = array();
```

```
// Do 'all' actions first
if ( isset($wp_filter['all']) ) {
    $wp_current_filter[] = $tag;
    $args = func_get_args();
    _wp_call_all_hook($args);
}
```

```
if ( !isset($wp_filter[$tag]) ) {
    if ( isset($wp_filter['all']) )
        array_pop($wp_current_filter);
    return $value;
}
```

```
if ( !isset($wp_filter['all']) )  
    $wp_current_filter[] = $tag;
```

```
// Sort
```

```
if ( !isset( $merged_filters[ $tag ] ) ) {  
    ksort($wp_filter[$tag]);  
    $merged_filters[ $tag ] = true;  
}
```

```
reset( $wp_filter[ $tag ] );
```

```
if ( empty($args) )  
    $args = func_get_args();
```

stores all arguments
passed to apply_filters

```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )  
        if ( !is_null($the_['function']) ){  
            $args[1] = $value;  
            $value = call_user_func_array(  
                $the_['function'],  
                array_slice(  
                    $args, 1, (int) $the_['accepted_args']  
                )  
            );  
        }  
    }
```

```
} while ( next($wp_filter[$tag]) !== false );
```

```
array_pop( $wp_current_filter );
```

```
return $value;
```

```
}
```

```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )
```

Looping through `$wp_filter['the_content'][10]`

```
$wp_filter['the_content'][10]['show_author_bio']
```

```
$wp_filter['the_content'][10]['wp_texturize']
```

```
$wp_filter['the_content'][10]['convert_smilies']
```

```
$wp_filter['the_content'][10]['convert_chars']
```

```
$wp_filter['the_content'][10]['wpautop']
```

```
$wp_filter['the_content'][10]['shortcode_unautop']
```

```
$wp_filter['the_content'][10]['prepend_attachment']
```

```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )  
        if ( !is_null($the_['function']) ){
```

```
$the_ = $wp_filter['the_content'][10]['show_author_bio']
```

```
$wp_filter['the_content'][10]['show_author_bio'] = array(  
    'function'      => 'show_author_bio',  
    'accepted_args' => 1  
);
```

```
$the_['function']: 'show_author_bio'
```

```
$the_['accepted_args']: 1
```



```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )  
        if ( !is_null($the_['function']) ){  
            $args[1] = $value;
```

```
$args = func_get_args();
```

```
$args = array( $tag, $value [, $param1, $param2, ...] );
```

```
apply_filters( 'the_content', $content );
```

```
$args = array( 'the_content', 'This is an example of a page.' );
```

```
apply_filters( 'the_category', $thelist, $separator, $parents );
```

```
$args = array( 'the_category', $thelist, $separator, $parents );
```

```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )  
        if ( !is_null($the_['function']) ){
```

```
            $args[1] = $value;
```

first time through loop:
original post content

```
do {  
    foreach( (array) current($wp_filter[$tag]) as $the_ )  
        if ( !is_null($the_['function']) ){  
            $args[1] = $value;  
            $value = call_user_func_array(  
                $the_['function'],  
                array_slice(  
                    $args, 1, (int) $the_['accepted_args']  
                )  
            );  
        }  
}
```

subsequent loops: \$value is updated each time

```
do {
  foreach( (array) current($wp_filter[$tag]) as $the_ )
    if ( !is_null($the_['function']) ){
      $args[1] = $value;
      $value = call_user_func_array(
        $the_['function'] — call this function
        array_slice(
          $args, 1, (int) $the_['accepted_args']
        )
      );
    }
  }
}
```

with these arguments

```
do {
    foreach( (array) current($wp_filter[$tag]) as $the_ )
        if ( !is_null($the_['function']) ){
            $args[1] = $value;
            $value = call_user_func_array(
                $the_['function'],
                array_slice(
                    $args, 1, (int) $the_['accepted_args']
                )
            );
        }
    }
}
```

```
} while ( next($wp_filter[$tag]) !== false );
```

```
array_pop( $wp_current_filter );
```

```
return $value;
```

```
}
```

```

function do_action($tag, $arg = '') {
    global $wp_filter, $wp_actions, $merged_filters, $wp_current_filter;

    if ( ! isset($wp_actions) )
        $wp_actions = array();

    if ( ! isset($wp_actions[$tag]) )
        $wp_actions[$tag] = 1;
    else
        ++$wp_actions[$tag];

    // Do 'all' actions first
    if ( isset($wp_filter['all']) ) {
        $wp_current_filter[] = $tag;
        $all_args = func_get_args();
        _wp_call_all_hook($all_args);
    }

    if ( !isset($wp_filter[$tag]) ) {
        if ( isset($wp_filter['all']) )
            array_pop($wp_current_filter);
        return;
    }

    if ( !isset($wp_filter['all']) )
        $wp_current_filter[] = $tag;

    $args = array();
    if ( is_array($arg) && 1 == count($arg) && isset($arg[0]) && is_object($arg[0]) ) // array(&$this)
        $args[] =& $arg[0];
    else
        $args[] = $arg;
    for ( $a = 2; $a < func_num_args(); $a++ )
        $args[] = func_get_arg($a);

    // Sort
    if ( !isset( $merged_filters[ $tag ] ) ) {
        ksort($wp_filter[$tag]);
        $merged_filters[ $tag ] = true;
    }

    reset( $wp_filter[ $tag ] );

    do {
        foreach ( (array) current($wp_filter[$tag]) as $the_ )
            if ( !is_null($the_['function']) )
                call_user_func_array($the_['function'], array_slice($args, 0, (int) $the_['accepted_args']));
    } while ( next($wp_filter[$tag]) !== false );

    array_pop($wp_current_filter);
}

```

do_action()

differences in do_action()

- ▶ does not keep track of `$value`
- ▶ does not return anything
- ▶ stores number of times each tag is fired in `$wp_actions[$tag]` (global)
- ▶ ...that's it.

hook spotlight:

template_redirect

wp-includes/template-loader.php

```
if ( defined( 'WP_USE_THEMES' ) && WP_USE_THEMES )  
    do_action( 'template_redirect' );
```

fired before **all** template decisions

```

if ( is_robots() ) :
    do_action('do_robots');
    return;
elseif ( is_feed() ) :
    do_feed();
    return;
elseif ( is_trackback() ) :
    include( ABSPATH . 'wp-trackback.php' );
    return;
endif;

if ( defined('WP_USE_THEMES') && WP_USE_THEMES ) :
    $template = false;
    if ( is_404() && $template = get_404_template() ) :
    elseif ( is_search() && $template = get_search_template() ) :
    elseif ( is_tax() && $template = get_taxonomy_template() ) :
    elseif ( is_front_page() && $template = get_front_page_template() ) :
    elseif ( is_home() && $template = get_home_template() ) :
    elseif ( is_attachment() && $template = get_attachment_template() ) :
        remove_filter('the_content', 'prepend_attachment');
    elseif ( is_single() && $template = get_single_template() ) :
    elseif ( is_page() && $template = get_page_template() ) :
    elseif ( is_category() && $template = get_category_template() ) :
    elseif ( is_tag() && $template = get_tag_template() ) :
    elseif ( is_author() && $template = get_author_template() ) :
    elseif ( is_date() && $template = get_date_template() ) :
    elseif ( is_archive() && $template = get_archive_template() ) :
    elseif ( is_comments_popup() && $template = get_comments_popup_template() ) :
    elseif ( is_paged() && $template = get_paged_template() ) :
    else :
        $template = get_index_template();
    endif;
    if ( $template = apply_filters( 'template_include', $template ) )
        include( $template );
    return;
endif;

```

where `template_redirect` is used

- ▶ `redirect_canonical`
- ▶ `wp_redirect_admin_locations`
- ▶ `wp_shortlink_header`
- ▶ `wp_old_slug_redirect`
- ▶ `maybe_redirect_404` (multisite only)

what you can do with `template_redirect`

- ▶ log search queries
- ▶ bypass template decision process and use custom theme structure
- ▶ custom URLs: load template before 404 is chosen

good hooks to know

you might have heard of:

- ▶ `init`
- ▶ `wp_enqueue_scripts`
- ▶ `wp_head` & `wp_footer`
- ▶ `the_content` & `the_excerpt`
- ▶ `post_class` & `body_class`

you might not have heard of:

- ▶ `posts_where`
- ▶ `style_loader_tag`
- ▶ `login_message`
- ▶ `post_updated`
- ▶ `wp_mail_from` & `wp_mail_from_name`

making your own hooks

you should make hooks when:

- ▶ making a theme framework
- ▶ releasing a public plugin
- ▶ making something big or complex

you should make hooks because:

- ▶ easier for other developers
- ▶ keeps code organized
- ▶ it's really, really easy

footer.php

```
<footer id="site-footer">  
    <p>Powered by  
        <a href="http://wordpress.org">WordPress</a>.</p>  
  
    <?php do_action( 'copyright_message' ); ?>  
</footer>
```

functions.php

```
add_action( 'copyright_message', 'my_groovy_copyright' );  
  
function my_groovy_copyright() {  
    echo '<p>&copy;2013 Groovy Inc. All rights reserved.</p>';  
}
```

home.php

```
echo apply_filters( 'archive_excerpt', get_the_excerpt() );
```

functions.php

```
add_filter( 'archive_excerpt', 'auto_link_blog_title' );
```

```
function auto_link_blog_title( $text ) {  
    $search = get_bloginfo('name');  
    $replace = '<a href="' . home_url() . '">  
        . get_bloginfo('name') . '</a>';  
  
    $new_text = str_replace( $search, $replace, $text );  
    return $new_text;  
}
```

~fin~