Open Source Software Development

Ung Văn Giàu
Email: giau.ung@eiu.edu.vn

Creating secondary queries and loops

You will often want to create a secondary query to display different content on the template

```php
<?php
// The main query.
if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        the_title();
        the_content();
    endwhile;
else :
    // When no posts are found, output this text.
    _e( 'Sorry, no posts matched your criteria.' );
endif;
wp_reset_postdata();

/*
 * The secondary query. Note that you can use any category name here. In our example,
 * we use "example-category".
 */
$secondary_query = new WP_Query( 'category_name=example-category' );

// The second loop.
if ( $secondary_query->have_posts() )
    echo '<ul>';
    while ( $secondary_query->have_posts() ) : $secondary_query->the_post();
        the_title( '<li>', '</li>' );
    endwhile;
    echo '</ul>';
endif;
wp_reset_postdata();
?>
```

WP_Query

https://developer.wordpress.org/reference/classes/wp_query/

- WP_Query provides numerous functions for common tasks within The Loop

- **Category Parameters**

  Show posts associated with certain categories.

  - cat (int) – use category id.

  - category_name (string) – use category slug.

  - category__and (array) – use category id.

  - category__in (array) – use category id.

  - category__not_in (array) – use category id.

WP_Query

- **Pagination Parameters**
  - **posts_per_page** (int) – number of post to show per page. Use 'posts_per_page'=>-1 to show all posts.
  - **offset** (int) – number of post to displace or pass over. It is ignored when 'posts_per_page'=>-1 is used.
  - Example:

    Display 5 posts per page which follow the 3 most recent posts:

    ```
    $query = new WP_Query( array( 'posts_per_page' => 5, 'offset' => 3 ) );
    ```

WP_Query

https://developer.wordpress.org/reference/classes/wp_query/

- **Order & Orderby Parameters**

  Sort retrieved posts

  - **order** (string) – Designates the ascending or descending order of the 'orderby' parameter.

    - ✓ 'ASC' – ascending order from lowest to highest values (1, 2, 3; a, b, c).

    - ✓ 'DESC' – descending order from highest to lowest values (3, 2, 1; c, b, a). (Defaults)

  - **orderby** (string) – Sort retrieved posts by parameter. Defaults to 'date (post_date)'.

    - ✓ 'ID' – Order by post id

    - ✓ 'title' – Order by title

    - ✓ 'name' – Order by post name (post slug)

    - ✓ 'date' – Order by date

WP_Query

- **Order & Orderby Parameters**

  - Example

    Display posts sorted by post 'title' in a descending order:

```php
$args = array(
    'orderby' => 'title',
    'order'   => 'DESC',
);
$query = new WP_Query( $args );
```

WP_Query

https://developer.wordpress.org/reference/classes/wp_query/

- Display posts that have this category (and any children of that category), using category slug:

  $query = new WP_Query(array('category_name' => 'staff'));

```php
// The Query
$query = new WP_Query( $args );

// The Loop
while ( $query->have_posts() ) {
    $query->the_post();
    echo '<li>' . get_the_title() . '</li>';
}

/* Restore original Post Data
 * NB: Because we are using new WP_Query we aren't stomping on the
 * original $wp_query and it does not need to be reset with
 * wp_reset_query(). We just need to set the post data back up with
 * wp_reset_postdata().
 */
wp_reset_postdata();
```

Function Reference

https://codex.wordpress.org/Function_Reference

Functions by category

Post, Custom Post Type, Page, Attachment and Bookmarks Functions	Category, Tag and Taxonomy Functions
Posts	**Categories**

Posts

- get_adjacent_post
- get_boundary_post
- get_children
- get_extended
- get_next_post
- get_next_posts_link
- next_posts_link
- get_permalink
- the_permalink
- get_the_excerpt
- the_excerpt
- get_the_post_thumbnail
- get_post
- get_post_field
- get_post_ancestors
- get_post_mime_type
- get_post_status
- get_post_format

Categories

- cat_is_ancestor_of
- get_all_category_ids (deprecated)
- get_ancestors
- get_cat_ID
- get_cat_name
- get_categories
- get_category
- get_category_by_path
- get_category_by_slug
- get_the_category_by_ID
- get_the_category_list
- get_category_link
- get_category_parents
- get_the_category
- single_cat_title
- in_category
- is_category
- the_category
- wp_category_checklist

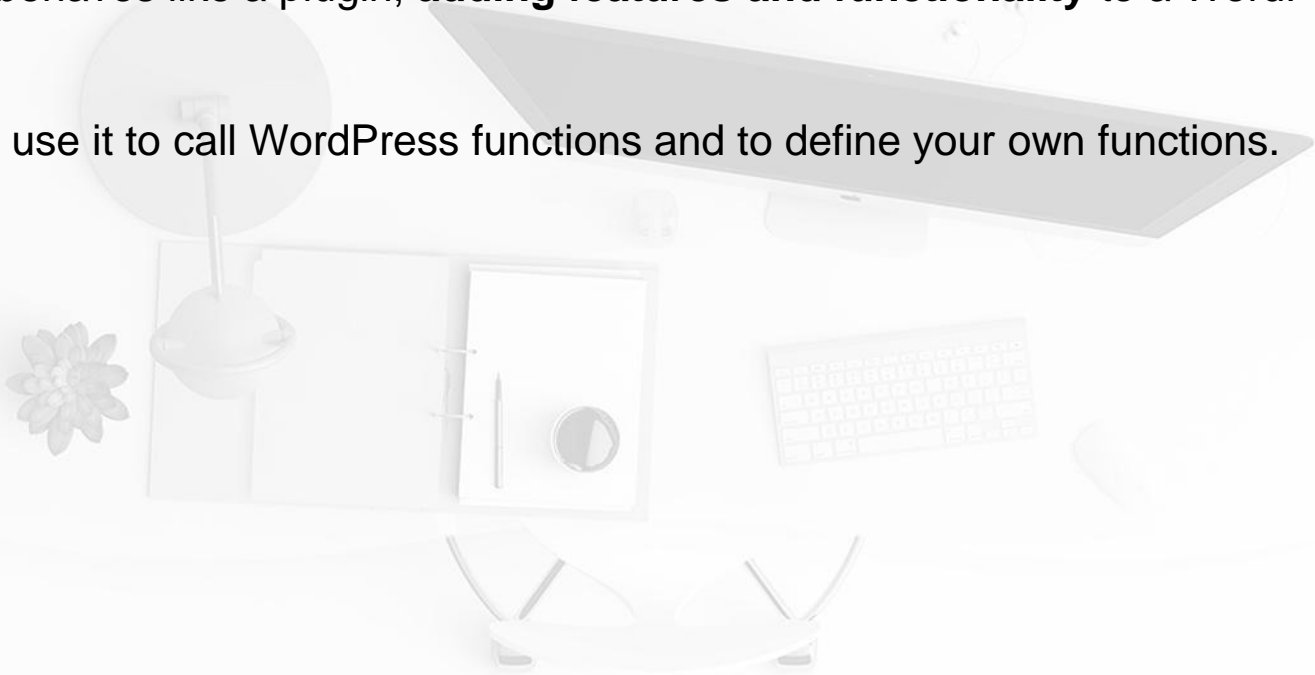# Content Management System

WordPress

Contents

Theme Functions

The **functions.php** file is where you add unique features to WordPress theme.

It can be used to hook into the core functions of WordPress to make the theme more modular, extensible, and functional.

What is functions.php?

- The file behaves like a plugin, **adding features and functionality** to a WordPress site.

- You can use it to call WordPress functions and to define your own functions.

What is functions.php?

There are advantages and tradeoffs to either using a WordPress plugin or using functions.php.

- **A WordPress plugin:**
  - only executes on page load when activated;
  - applies to all themes
- **A functions.php file:**
  - executes only when in the active theme's directory;
  - applies only to that theme (if the theme is changed, the features can no longer be used)

What is functions.php?

- With **functions.php** you can:

  - Use WordPress hooks

  - Enable WordPress features with add_theme_support()

  - Define functions you wish to reuse in multiple theme template files.

Navigation Menus

Examples

- Custom navigation menus allow users to **edit and customize menus** in the Menus admin panel, giving users a drag-and-drop interface to edit the various menus in theme

Navigation Menus

Examples

- You can set up multiple menus in **functions.php**.

  They can be added using **register_nav_menus()** and inserted into a theme using

wp_nav_menu()

- **Short:**

```php
register_nav_menus(
    array(
        'header-menu' => __( 'Header Menu' ),
        'extra-menu' => __( 'Extra Menu' )
    )
);
```

- **Full:**

```php
function register_my_menus() {
    register_nav_menus(
        array(
            'header-menu' => __( 'Header Menu' ),
            'extra-menu' => __( 'Extra Menu' )
        )
    );
}
add_action( 'init', 'register_my_menus' );
```

Navigation Menus

Examples

- Each of the menus you define can be called later using **wp_nav_menu()** and using the name assigned (i.e., primary) as the **theme_location** parameter.

```
wp_nav_menu( array( 'theme_location' => 'header-menu' ) );
```

- More details about wp_nav_menu():

  https://developer.wordpress.org/reference/functions/wp_nav_menu/

Post Thumbnails

Examples

- **Post thumbnails and featured images** allow users to choose an image to represent their post.

- Your theme can decide how to display them, depending on its design.

```
add_theme_support( 'post-thumbnails' );
```

Add menu page

Examples

- Add a custom menu item to the WordPress admin menu

- **Syntax:**

```
add_menu_page( string $page_title, string $menu_title, string $capability,
        string $menu_slug, callable $function = '', string $icon_url = '',
        int $position = null )
```

- Example:

```
add_action( 'admin_menu', 'register_my_custom_menu_page' );
function register_my_custom_menu_page() {
    add_menu_page( 'Custom Menu Page Title', 'Custom Menu Page',
        'manage_options', 'custom.php', '', 'dashicons-welcome-widgets-menus', 90 );
}
```

- **More details:**

  https://developer.wordpress.org/reference/functions/add_menu_page/

Linking Theme Files & Directories

Linking to Core Theme Files

- WordPress themes are built from a number of different template files.

- This will usually include a sidebar.php, header.php and footer.php

  - get_header();

  - get_footer();

  - get_sidebar();

- You can create custom versions of these files by naming the file

  - sidebar-{your_custom_template}.php,

  - header-{your_custom_template}.php

  - and footer-{your_custom_template}.php

Linking to Core Theme Files

- You can then use Template Tags with the custom template name as the only parameter, like this:

```
get_header( 'your_custom_template' );
get_footer( 'your_custom_template' );
get_sidebar( 'your_custom_template' );
```

- You can create custom template files and call them at any location in the page using **get_template_part()**

```
get_template_part( 'content', 'product' );
```

```
get_template_part( 'content-templates/content', 'location' );
get_template_part( 'content-templates/content', 'product' );
get_template_part( 'content-templates/content', 'profile' );
```

Linking to Theme Directories

- To link to the theme's directory:

  get_theme_file_uri();

- Example:

  echo get_theme_file_uri('images/logo.png');

- To access the path to a file in theme's directories:

  get_theme_file_path();

Dynamic Linking in Templates

You can link to a page or post dynamically by referring to its unique numerical ID

```
<a href="<?php echo get_permalink($ID); ?>">This is a link</a>
```

Q&A