



Open Source Software Development

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



Content Management System

WordPress



Contents

01

Introduction to WordPress Theme

02

Theme Basics

03

Template Hierarchy

04

Template Tags

05

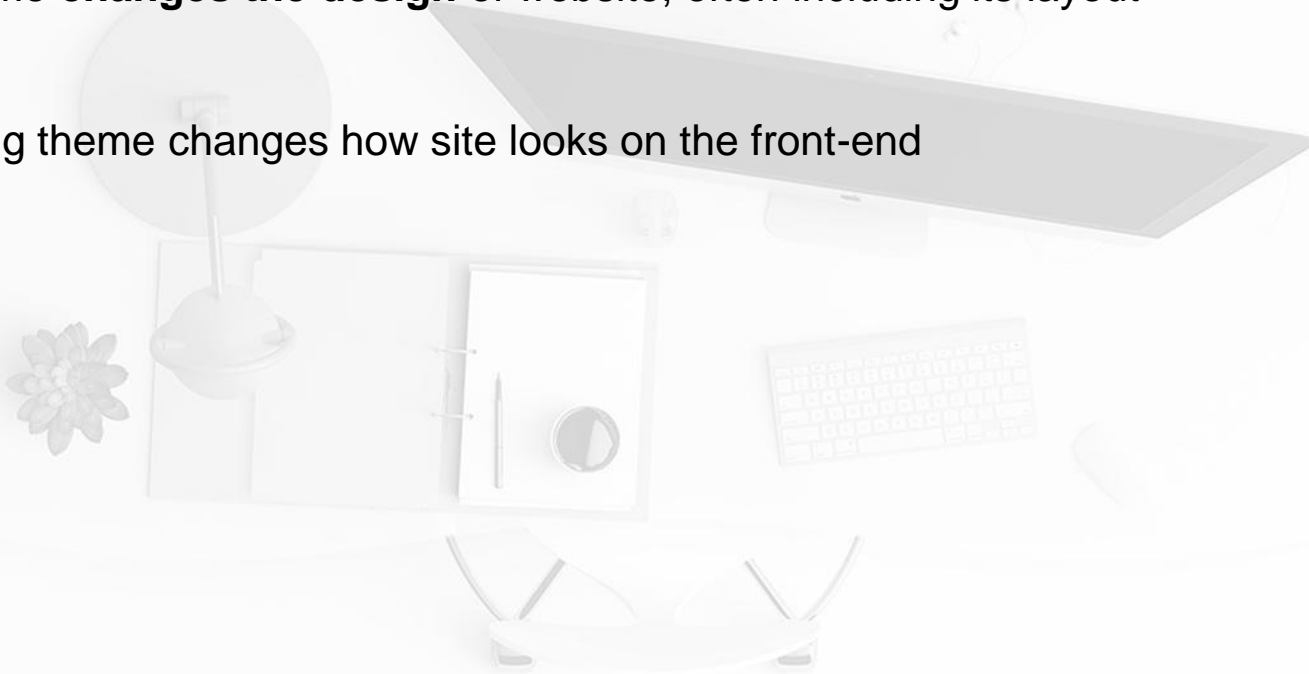
The Loop



WordPress Theme

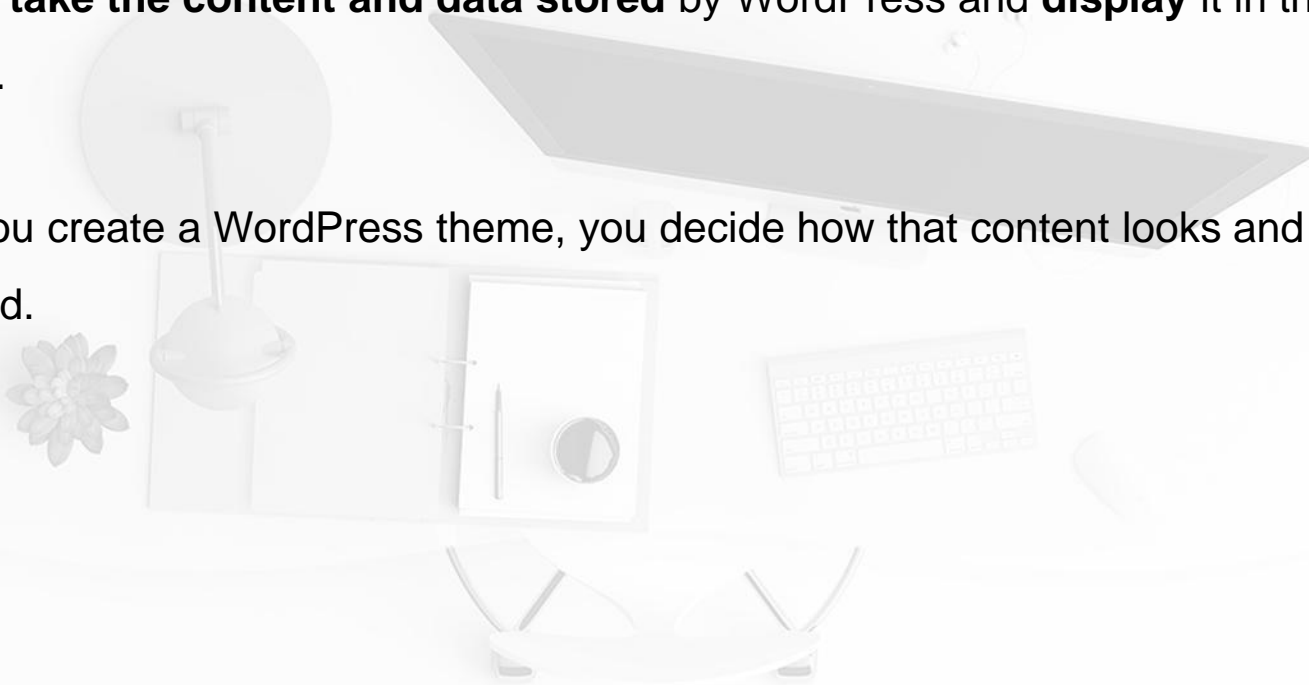
What is a Theme?

- The theme **changes the design** of website, often including its layout
- Changing theme changes how site looks on the front-end



What can themes do?

- Themes **take the content and data stored** by WordPress and **display** it in the browser.
- When you create a WordPress theme, you decide how that content looks and is displayed.



What are themes made of?

Required files

only **two** files absolutely required

- **index.php** – the main template file
- **style.css** – the main style file

Though not required, you may see additional files:

- PHP files – including template files
- Localization files
- CSS files
- Graphics
- JavaScript
- Text files – usually license info, readme.txt instructions, and a changelog file

What is the difference between a theme and a plugin?

- A theme controls the **presentation** of content;
- A plugin is used to control the **behavior and features** of WordPress site

Any theme you create should not add critical functionality → move critical features to plugins



Theme Basics

Theme Basics

- Learn **how to build** a theme properly
- The **anatomy** of a theme
- **Understand the building blocks** of a theme by looking at **theme files and post types**
- Learn **how to keep files organized** within theme
- Look at **The Loop**, which is responsible for **pulling content out** of the WordPress database
- Learn more about **adding features** to theme by using theme **functions**, including **CSS & JavaScript**,...

Template Files

▪ Template Terminology

- **Templates files** exist within a theme and express how site is displayed.
- **Page Templates** are those that **apply only to pages** to change look and feel. A page template can be applied to a single page, or a class of pages.
- **Template Tags** are built-in **functions** you can use inside a template file to retrieve and display data such as **the_title()** and **the_content()**
- **Template Hierarchy** is the logic WordPress uses to **decide which theme template file(s) to use**, depending on the content being requested.

Template Files

▪ Template files

- These are PHP files that **contain a mixture of HTML, Template Tags, and PHP code.**
- When someone visits a page on your website, WordPress loads a template based on the request
- The **most critical template file** is **index.php**, which is the catch-all template if a more-specific template can not be found in the template hierarchy

Template Files

▪ Template partials

- A piece of a template that is **included as a part of another template**, such as a site header.
- Template partials can be embedded in multiple templates, simplifying theme creation.
- Common template partials include:
 - ✓ **header.php** for generating the site's header
 - ✓ **footer.php** for generating the footer
 - ✓ **sidebar.php** for generating the sidebar

Template Files

- **Common WordPress template files**

- **index.php**

- The main template file. It is **required** in all themes.

- **style.css**

- The main stylesheet. It is **required** in all themes and contains the information header for theme.

- **comments.php**

- The comments template.

- **header.php**

- The header template file usually contains site's document type, meta information, links to stylesheets and scripts, and other data.

Template Files

- **Common WordPress template files**

- **singular.php**

- ✓ The singular template is used for posts when **single.php** is not found, or for pages when **page.php** are not found.
- ✓ If singular.php is not found, index.php is used.

- **single.php**

The single post template is used when a visitor requests a single post.

- **single-{post-type}.php**

- ✓ The single post template used when a visitor requests a single post from a custom post type.
- ✓ The index.php is used if a specific query template for the custom post type is not present.

Template Files

▪ Common WordPress template files

- **page.php**

The page template is used when visitors request individual pages, which are a built-in template.

- **page-{slug}.php**

The page slug template is used when visitors request a specific page, for example one with the “about” slug (page-about.php).

- **category.php**

The category template is used when visitors request posts by category.

- **archive.php**

✓ The archive template is used when visitors request posts by category, author, or date.

✓ **Note:** this template will be overridden if more specific templates are present like category.php, author.php, and date.php.

Template Files

- **Common WordPress template files**

- **search.php**

- The search results template is used to display a visitor's search results.

- **attachment.php**

- The attachment template is used when viewing a single attachment like an image, pdf, or other media file.

- **404.php**

- The 404 template is used when WordPress cannot find a post, page, or other content that matches the visitor's request.

Template Files

- **Using template files**

- Use Template Tags to display information dynamically
- For example, in **index.php** you can include other files in final generated page
 - ✓ To include the header, use **get_header()**
 - ✓ To include the sidebar, use **get_sidebar()**
 - ✓ To include the footer, use **get_footer()**
 - ✓ To include the search form, use **get_search_form()**
 - ✓ To include custom theme files, use **get_template_part()**

Template Files

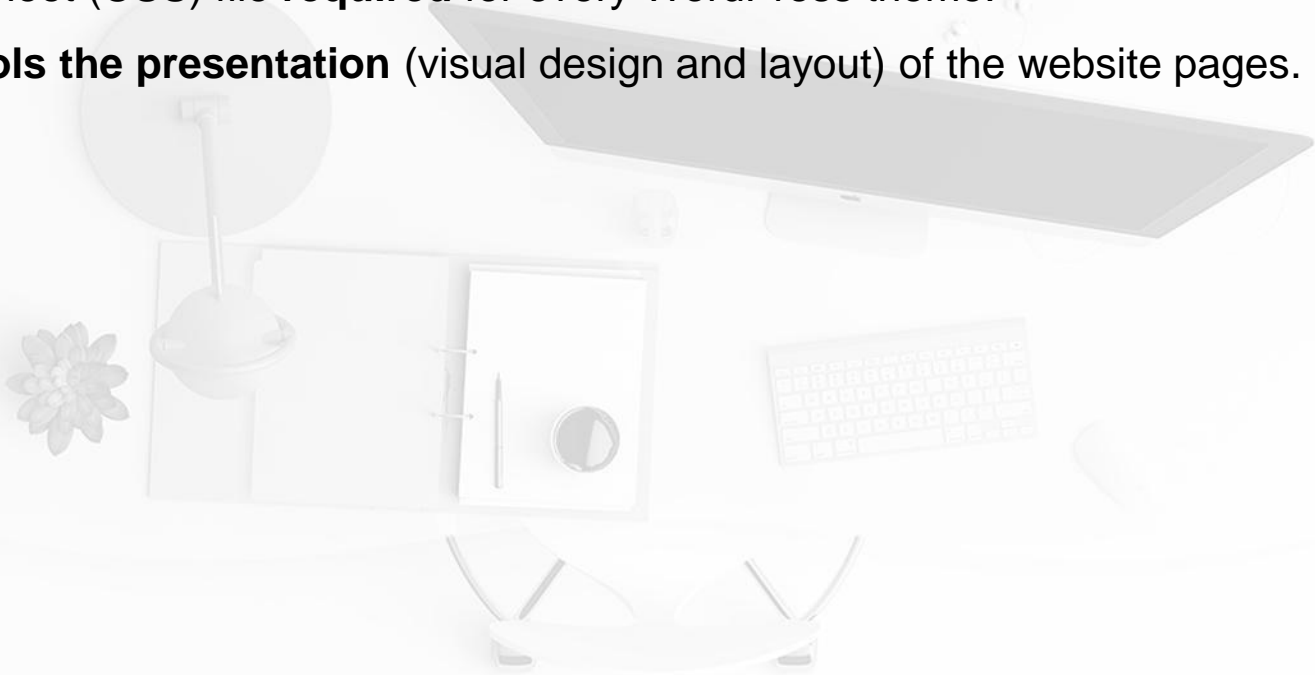
- **Using template files**

- An example of template tags to include specific templates:

```
<?php get_sidebar(); ?>  
<?php get_template_part( 'featured-content' ); ?>  
<?php get_footer(); ?>
```

Main Stylesheet (style.css)

- A stylesheet (CSS) file **required** for every WordPress theme.
- It **controls the presentation** (visual design and layout) of the website pages.



Main Stylesheet (style.css)

- **Location**

To recognize the set of theme template files as a valid theme, the style.css file needs to be located in the **root directory** of theme, not a subdirectory.

- **Basic Structure**

WordPress uses the **header comment section** of a style.css to **display information about the theme** in the Appearance (Themes) dashboard panel.

Main Stylesheet (style.css)

```
/*
Theme Name: Twenty Twenty
Theme URI: https://wordpress.org/themes/twentytwenty/
Author: the WordPress team
Author URI: https://wordpress.org/
Description: Our default theme for 2020 is designed to take full advantage of the latest WordPress features.
Tags: blog, one-column, custom-background, custom-colors, custom-logo, custom-menu, custom-header, e-commerce, featured-images, flexible-header, full-width-template, rtl-language-support, translation-ready, two-column, wide-blocks
Version: 1.3
Requires at least: 5.0
Tested up to: 5.4
Requires PHP: 7.0
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Text Domain: twentytwenty
This theme, like WordPress, is licensed under the GPL.
Use it to make something cool, have fun, and share what you've learned with others.
*/
```

Main Stylesheet (style.css)

▪ Explanations

- **Theme Name (*)**: Name of the theme.
- **Theme URI**: The URL of a public web page where users can find more information about the theme.
- **Author (*)**: The name of the individual or organization who developed the theme.
Using the Theme Author's wordpress.org username is recommended.
- **Author URI**: The URL of the authoring individual or organization.
- **Description (*)**: A short description of the theme.
- **Version (*)**: The version of the theme, written in X.X or X.X.X format.
- **Requires at least (*)**: The oldest main WordPress version the theme will work with, written in X.X format. Themes are only required to support the three last versions.

Main Stylesheet (style.css)

▪ Explanations

- **Tested up to (*)**: The last main WordPress version the theme has been tested up to, i.e., 5.4. Write only the number, in X.X format.
- **Requires PHP (*)**: The oldest PHP version supported, in X.X format, only the number
- **License (*)**: The license of the theme.
- **License URI (*)**: The URL of the theme license.
- **Text Domain (*)**: The string used for textdomain for translation.
- **Tags**: Words or phrases that allow users to find the theme using the tag filter. A full list of tags is in the Theme Review Handbook.
- **Domain Path**: Used so that WordPress knows where to find the translation when the theme is disabled. Defaults to /languages.



Template Hierarchy

How WordPress determines which template file(s) to use
on individual pages

The Template File Hierarchy

▪ Overview

- WordPress **uses the query string to decide which template or set of templates** should be used to display the page
- The query string is information that is contained in the link to each part of your website

The Template File Hierarchy

▪ Overview

- To determine which template file to use, WordPress:
 - ✓ **Matches every query string to a query type** to decide which page is being requested (for example, a search page, a category page, etc.);
 - ✓ **Selects the template in the order determined** by the template hierarchy;
 - ✓ **Looks for template files with specific names** in the current theme's directory and uses the first matching template file as specified by the hierarchy.
- If WordPress cannot find a template file with a matching name, it will skip to the next file in the hierarchy.
- If WordPress cannot find any matching template file, the theme's index.php file will be used.

The Template File Hierarchy

Visual Overview



The Template File Hierarchy

The Template Hierarchy In Detail

▪ Home Page display

- By default, WordPress sets your site's home page to display latest blog posts. This page is called the **blog posts index**.
- You can also set blog posts to display on a separate static page.
- The template file **home.php** is used to render the blog posts index. If home.php does not exist, WordPress will use index.php.
 - ✓ home.php
 - ✓ index.php
 - ✓ If **front-page.php** exists, it will override the **home.php** template.

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Front Page display

- The **front-page.php** template file is used to render site's front page
- **front-page.php** – Used for both “**your latest posts**” or “**a static page**” as set in the front page displays section of Settings → Reading.
- **home.php** – If WordPress cannot find **front-page.php** and “**your latest posts**” is set in the front page displays section, it will look for **home.php**.
- **page.php** – When “**front page**” is set in the **front page displays** section.
- **index.php** – When “**your latest posts**” is set in the **front page displays** section but **home.php** does not exist or when **front page** is set but **page.php** does not exist.

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Single Post

- The single post template file is used to render a single post
- WordPress uses the following path:
 - ✓ **single-{post-type}.php** – If the post type is product, WordPress would look for single-product.php.
 - ✓ **single.php** – WordPress then falls back to single.php.
 - ✓ **singular.php** – Then it falls back to singular.php.
 - ✓ **index.php** – Finally, as mentioned above, WordPress ultimately falls back to index.php.

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Single Page

- The template file used to render a static page (**page** post-type).
- Note that unlike other post-types, **page** is special to WordPress and uses the following path:
 - ✓ **page-{slug}.php** – If the page slug is recent-news, WordPress will look to use page-recent-news.php.
 - ✓ **page-{id}.php** – If the page ID is 6, WordPress will look to use page-6.php.
 - ✓ **page.php**
 - ✓ **singular.php**
 - ✓ **index.php**

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Category

Rendering category archive index pages uses the following path in WordPress:

- **category-{slug}.php** – If the category's slug is news, WordPress will look for category-news.php.
- **category-{id}.php** – If the category's ID is 6, WordPress will look for category-6.php.
- **category.php**
- **archive.php**
- **index.php**

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Search Result

- Search results follow the same pattern as other template types:
- **search.php**
- **index.php**

▪ 404 (Not Found)

- 404 template files are called in this order:
- **404.php**
- **index.php**

The Template File Hierarchy

The Template Hierarchy In Detail

▪ Attachment

Rendering an attachment page (**attachment** post-type) uses the following path:

- **{MIME-type}.php** – can be any MIME type (For example: image.php, video.php, pdf.php). For text/plain, the following path is used (in order):
 - ✓ **text-plain.php**
 - ✓ **plain.php**
 - ✓ **text.php**
- **attachment.php**
- **single-attachment-{slug}.php** – For example, if the attachment slug is holiday, WordPress would look for single-attachment-holiday.php.
- **single-attachment.php**
- **single.php**
- **singular.php**
- **index.php**

Template Tags

- Template tags are used within themes to **retrieve content from database**
- Template tags are the preferred method to pull content into your theme because:
 - they can print dynamic content;
 - they can be used in multiple theme files;
 - they separate the theme into smaller, more understandable, sections.

What is a Template Tag?

- A template tag is **simply a piece of code** that tells WordPress to get something from the database.
- It is broken up into three components:
 - A PHP code tag
 - A WordPress function
 - Optional parameters
- You can use a template tag to **call another theme file** or some **information from the database**.

What is a Template Tag?

- For example:
 - **get_header()** to **get** the **header.php** file and **include** it in the current theme file.
 - **get_footer()** to get the **footer.php** file.
- There are also other kinds of template tags:
 - **the_title()** to get the title of the page or post from the database and include it.
 - **bloginfo('name')** to get the blog title out of the database and include it in the template file.

Why Use Template Tags

- By encapsulating all of the code for a particular chunk of content, template tags make it very easy to include various pieces of a template in a theme file and also to maintain the theme.
- Another reason to use template tags is to display dynamic data, i.e., data from the database

How to Use Template Tags

- Using template tags is very simple.
- In any template file you can use a template tag by simply printing one line of php code to call the template tag.
- Printing the header.php file is as simple as:

get_header();



Parameters

- Some template tags let you pass parameters.
- Parameters are extra pieces of information that determine what is retrieved from the database.
- For example:
 - `bloginfo('name');`
 - `bloginfo('version');`

Using Template Tags Within the Loop

- Many template tags work within the WordPress Loop.
- This means that they are included in the template files as part of the php “loop” that generates the pages users see based upon the instructions inside of the loop.
- The WordPress loop begins with:

```
if ( have_posts() ) :  
    while ( have_posts() ) :  
        the_post();
```

Using Template Tags Within the Loop

- Template tags that **work within the loop must be in the middle area**, before the ending section of the loop below:

```
    endwhile;  
else :  
    _e( 'Sorry, no posts matched your criteria.', 'devhub' );  
endif;
```

- Some of template tags that need to be inside of the loop include:
 - the_title()
 - the_ID()
 - the_permalink()
 - the_content()
 - the_post_thumbnail("thumbnail" | "medium" | "large" | "full");
 - the_excerpt()
 - next_post_link() / previous_post_link()

The Loop

- The Loop is the default mechanism WordPress **uses for outputting posts** through a theme's template files
- Within the Loop, WordPress retrieves each post to be displayed on the current page and formats it according to theme's instructions.
- The Loop **extracts the data for each post** from the WordPress database and **inserts the appropriate information in place of each template tag**

The Loop

- You can use the Loop for a number of different things, for example to:
 - display post titles and excerpts on blog's homepage;
 - display the content and comments on a single post;
 - display the content on an individual page using template tags;
 - display data from Custom Post Types and Custom Fields.

The Loop in Detail

The basic loop is:

```
<?php
if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        // Display post content
    endwhile;
endif;
?>
```

This loop says that when there are posts, loop through and display the posts

Using The Loop

- An extremely simple index.php file would look like:

```
<?php
get_header();

if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        the_content();
    endwhile;
else :
    _e( 'Sorry, no posts matched your criteria.', 'textdomain' );
endif;

get_sidebar();
get_footer();
?>
```

What the Loop Can Display

- The Loop can display a number of different elements for each post.
- Some common template tags used in many themes are:
 - **the_title()** – the title of the post or page
 - **the_content()** – the main content for a post or page
 - **the_ID()** – the ID for the post or page
 - **the_author()** – the author of the post or page
 - **the_excerpt()** – the first 55 words of a post's main content followed by an ellipsis (...) or read more link that goes to the full post. You may also use the “Excerpt” field of a post to customize the length of a particular excerpt.
 - **the_permalink()** – displays the permalink for the current post

What the Loop Can Display

- Some common template tags used in many themes are:
 - **the_post_thumbnail()**([string \$size, [string|array \$attr]]) – displays the post thumbnail
E.g., the_post_thumbnail('full', array("class" => "center middle", "alt" => "Logo"))
 - **the_time()** – the time or date for the post or page. This can be customized using standard php date function formatting. E.g., the_time("d-m-Y"). Formatting Date and Time:
<https://wordpress.org/support/article/formatting-date-and-time/>
 - **the_category()** – the category or categories associated with the post or page being viewed
 - **next_post_link()** – a link to the post published chronologically after the current post
 - **previous_post_link()** – a link to the post published chronologically before the current post
 - **the_meta()** – the custom fields associated with the post or page
 - **the_tags()** – the tag or tags associated with the post

Basic Examples

Blog Archive

- Most blogs have a blog archive page, which can show a number of things including the post title, thumbnail, and excerpt.

```
<?php
if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        the_title( '<h2>', '</h2>' );
        the_post_thumbnail();
        the_excerpt();
    endwhile;
else:
    _e( 'Sorry, no posts matched your criteria.', 'textdomain' );
endif;
?>
```

Basic Examples

Individual Post

- Each post has its own page, which displays the relevant information for that post.

```
<?php
if ( have_posts() ) :
    while ( have_posts() ) : the_post();
        the_title( '<h1>', '</h1>' );
        the_content();
    endwhile;
else:
    _e( 'Sorry, no pages matched your criteria.', 'textdomain' );
endif;
?>
```

Creating secondary queries and loops

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

```
<?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();
?>
```



Q&A

References

- **Theme Development**

https://codex.wordpress.org/Theme_Development

- **Using Themes**

<https://wordpress.org/support/article/using-themes/>