# Template Hierarchy

Template hierarchy is a fundamental concept in content management system like WordPress. It defines the order in which template files are loaded to display different types of content on a website. This system allows for flexibility and customization of the site’s appearance and functionality. Here’s an in-depth look at template hierarchy.

## What is Template Hierarchy?

Template hierarchy is a structure that determines which template file WordPress uses to display a particular page or post. When a page is requested, WordPress searches through the hierarchy of template files and uses the first one it finds.

## How Template Hierarchy Works?

The template hierarchy follows a specific order. WordPress looks for template files in the following order, falling back to the next file if the previous one is not found:

* Home Page:
* `home.php` (if set as a static front page)
* `index.php`
* Front Page:
* `front-page.php`
* Single Post:
* `single-{post-type}-{slug}.php` (for custom post types)
* `single-{post-type}.php` (for custom post types)
* `single.php`
* `singular.php`
* `index.php`
* Page:
* `custom-template.php` (if a custom template is assigned)
* `page-{slug}.php`
* `page-{ID}.php`
* `page.php`
* `singular.php`
* `index.php`
* Category:
* `category-{slug}.php`
* `category-{ID}.php`
* `category.php`
* `archive.php`
* `index.php`
* Tags:
* `tag-{slug}.php`
* `tag-{ID}.php`
* `tag.php`
* `archive.php`
* `index.php`
* Custom Taxonomy:
* `taxonomy-{taxonomy}-{term}.php`
* `taxonomy-{taxonomy}.php`
* `taxonomy.php`
* `archive.php`
* `index.php`
* Author
* `author-{name}.php`
* `author-{ID}.php`
* `author.php`
* `archive.php`
* `index.php`
* Date
* `date.php`
* `archive.php`
* `index.php`
* Archive
* `archive-{post-type}.php`
* `archive.php`
* `index.php`
* Search Results
* `search.php`
* `index.php`
* 404 Error Page:
* `404.php`
* `index.php`

## Understanding Template Hierarchy

* **Custom Templates:** You can create custom templates for specific pages or posts by naming the files appropriately. For example, **`page-contact.php`** will be used for a page with the slug **“contact”.**
* **Fallback System:** If a specific template file does not exist, WordPress falls back to a more general template file. For instance, if **`single-{post-type}.php`** does not exist, WordPress will use **`single.php`.**
* **Child Themes:** When using a child theme, the template hierarchy first checks for template files in the child theme’s directory before falling back to the parent theme’s directory.

## Example Scenario

Suppose you have a custom post type called “movie” and you want to create a custom template for individual movie posts:

* **Template File Order:** WordPress will look or the following files in this order:
* `single-movie.php`
* `single.php`
* `singular.php`
* `index.php`

If you create `single-movie.php`, WordPress will use this template to display all single posts of the custom post type “movie.”

**Practical Application**

Understanding the template hierarchy allows developers to customize their WordPress themes effectively, ensuring that different types of content are displayed using the most appropriate and specific templates available. This enhances the user experience by providing tailored layouts and designs for various sections of a website.

By leveraging the template hierarchy, you can create a well-structured, highly customizable website that caters to specific content needs without sacrificing flexibility or maintainability.