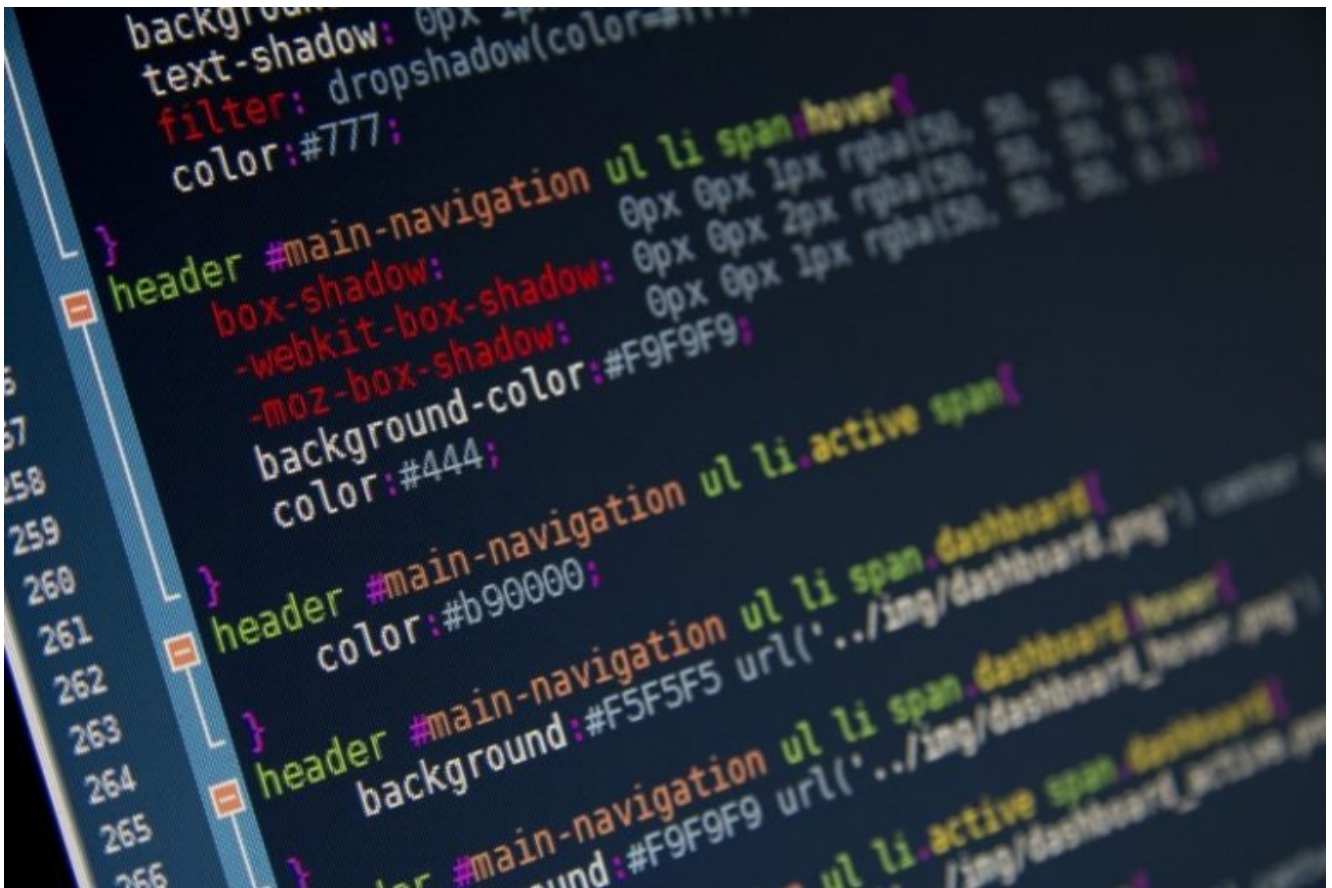**How to create the widget(Sidebar) in WordPress custom theme ?**

How to create the website menu in custom theme of WordPress ?

How to create custom Taxonomy in WordPress ?

To create the custom taxonomy in WordPress code is given below please copy this code and paste in your functions.php file of your activated theme.

// hook into the init action and call create_book_taxonomies when it fires

```php
add_action( 'init', 'create_book_taxonomies', 0 );

// create two taxonomies, genres and writers for the post type
"book"
function create_book_taxonomies() {
        // Add new taxonomy, make it hierarchical (like
categories)
    $labels = array(
            'name'                => _x( 'Genres', 'taxonomy
general name', 'textdomain' ),
            'singular_name'       => _x( 'Genre', 'taxonomy
singular name', 'textdomain' ),
            'search_items'        => __( 'Search Genres',
'textdomain' ),
            'all_items'           => __( 'All Genres',
'textdomain' ),
            'parent_item'         => __( 'Parent Genre',
'textdomain' ),
            'parent_item_colon' => __( 'Parent Genre:',
'textdomain' ),
            'edit_item'           => __( 'Edit Genre',
'textdomain' ),
            'update_item'         => __( 'Update Genre',
'textdomain' ),
            'add_new_item'        => __( 'Add New Genre',
'textdomain' ),
            'new_item_name'       => __( 'New Genre Name',
'textdomain' ),
            'menu_name'           => __( 'Genre',
'textdomain' ),
        );

    $args = array(
            'hierarchical'      => true,
            'labels'            => $labels,
            'show_ui'           => true,
            'show_admin_column' => true,
            'query_var'         => true,
            'rewrite'           => array( 'slug' =>
'genre' ),
        );
```

```php
        register_taxonomy( 'genre', array( 'book' ), $args );

        // Add new taxonomy, NOT hierarchical (like tags)
        $labels = array(
                'name'                          => _x( 'Writers',
'taxonomy general name', 'textdomain' ),
                'singular_name'                 => _x( 'Writer',
'taxonomy singular name', 'textdomain' ),
                'search_items'                  => __( 'Search
Writers', 'textdomain' ),
                'popular_items'                 => __( 'Popular
Writers', 'textdomain' ),
                'all_items'                     => __( 'All
Writers', 'textdomain' ),
                'parent_item'                   => null,
                'parent_item_colon'             => null,
                'edit_item'                     => __( 'Edit
Writer', 'textdomain' ),
                'update_item'                   => __( 'Update
Writer', 'textdomain' ),
                'add_new_item'                  => __( 'Add New
Writer', 'textdomain' ),
                'new_item_name'                 => __( 'New
Writer Name', 'textdomain' ),
                'separate_items_with_commas' => __( 'Separate
writers with commas', 'textdomain' ),
                'add_or_remove_items'           => __( 'Add or
remove writers', 'textdomain' ),
                'choose_from_most_used'         => __( 'Choose
from the most used writers', 'textdomain' ),
                'not_found'                     => __( 'No
writers found.', 'textdomain' ),
                'menu_name'                     => __( 'Writers',
'textdomain' ),
        );

        $args = array(
                'hierarchical'          => false,
                'labels'                => $labels,
                'show_ui'               => true,
                'show_admin_column'     => true,
```

```php
                              'update_count_callback' =>
'_update_post_term_count',
              'query_var'              => true,
              'rewrite'                => array( 'slug' =>
'writer' ),
        );

        register_taxonomy( 'writer', 'book', $args );
}
```

Basic Example :

```php
add_action( 'init', 'create_book_tax' );

function create_book_tax() {
        register_taxonomy(
                'genre',
                'book',
                array(
                        'label' => __( 'Genre' ),
                        'rewrite' => array( 'slug' => 'genre'
),
                        'hierarchical' => true,
                )
        );
}
```

How to create custom post type in WordPress?

To create the custom post types in WordPress code is given below please copy this code and paste in your functions.php file of your activated theme.

```
/**
 * Register a custom post type called "book".
 *
 * @see get_post_type_labels() for label keys.
 */
function wpdocs_codex_book_init() {
    $labels = array(
        'name'                  => _x( 'Books', 'Post type
general name', 'textdomain' ),
        'singular_name'         => _x( 'Book', 'Post type
singular name', 'textdomain' ),
        'menu_name'             => _x( 'Books', 'Admin Menu
text', 'textdomain' ),
        'name_admin_bar'        => _x( 'Book', 'Add New on
Toolbar', 'textdomain' ),
```

```php
        'add_new'                   => __( 'Add New', 'textdomain'
),
        'add_new_item'              => __( 'Add New Book',
'textdomain' ),
        'new_item'                  => __( 'New Book',
'textdomain' ),
        'edit_item'                 => __( 'Edit Book',
'textdomain' ),
        'view_item'                 => __( 'View Book',
'textdomain' ),
        'all_items'                 => __( 'All Books',
'textdomain' ),
        'search_items'              => __( 'Search Books',
'textdomain' ),
        'parent_item_colon'     => __( 'Parent Books:',
'textdomain' ),
        'not_found'                 => __( 'No books found.',
'textdomain' ),
        'not_found_in_trash'    => __( 'No books found in
Trash.', 'textdomain' ),
        'featured_image'            => _x( 'Book Cover Image',
'Overrides the "Featured Image" phrase for this post type.
Added in 4.3', 'textdomain' ),
        'set_featured_image'     => _x( 'Set cover image',
'Overrides the "Set featured image" phrase for this post type.
Added in 4.3', 'textdomain' ),
        'remove_featured_image' => _x( 'Remove cover image',
'Overrides the "Remove featured image" phrase for this post
type. Added in 4.3', 'textdomain' ),
        'use_featured_image'     => _x( 'Use as cover image',
'Overrides the "Use as featured image" phrase for this post
type. Added in 4.3', 'textdomain' ),
        'archives'                  => _x( 'Book archives', 'The
post type archive label used in nav menus. Default "Post
Archives". Added in 4.4', 'textdomain' ),
        'insert_into_item'       => _x( 'Insert into book',
'Overrides the "Insert into post"/"Insert into page" phrase
(used when inserting media into a post). Added in 4.4',
'textdomain' ),
        'uploaded_to_this_item' => _x( 'Uploaded to this
book', 'Overrides the "Uploaded to this post"/"Uploaded to
```

```
this page" phrase (used when viewing media attached to a
post). Added in 4.4', 'textdomain' ),
        'filter_items_list'     => _x( 'Filter books list',
'Screen reader text for the filter links heading on the post
type listing screen. Default "Filter posts list"/"Filter pages
list". Added in 4.4', 'textdomain' ),
        'items_list_navigation' => _x( 'Books list
navigation', 'Screen reader text for the pagination heading on
the post type listing screen. Default "Posts list
navigation"/"Pages list navigation". Added in 4.4',
'textdomain' ),
        'items_list'            => _x( 'Books list', 'Screen
reader text for the items list heading on the post type
listing screen. Default "Posts list"/"Pages list". Added in
4.4', 'textdomain' ),
    );
    $args = array(
        'labels'             => $labels,
        'public'             => true,
        'publicly_queryable' => true,
        'show_ui'            => true,
        'show_in_menu'       => true,
        'query_var'          => true,
        'rewrite'            => array( 'slug' => 'book' ),
        'capability_type'    => 'post',
        'has_archive'        => true,
        'hierarchical'       => false,
        'menu_position'      => null,
        'supports'           => array( 'title', 'editor',
'author', 'thumbnail', 'excerpt', 'comments' ),
    );
    register_post_type( 'book', $args );
}
add_action( 'init', 'wpdocs_codex_book_init' );
```

Which files are important to create the WordPress theme ?

1. style. css – First, a **WordPress theme** needs a style. css **file**
2. index. php – This **file** serves as the default **file**

In style.css **file** we will write the theme name, version, author name and theme description in comment section. For ex:

/*

Theme Name : My theme

Version : 1.0.0

Author : iTechSupport

Description : Very Simple WordPress theme

*/

In **Index.php file** we will write the HTML code this page contains the home page layout of the WordPress Theme.

How to start WordPress from scratch ?

WordPress is a Content Management System of PHP. Need to basic knowledge of PHP to start WordPress development. Now a days many premium themes and plugins are available in WordPress. IF any user do not want to learn WordPress as a development then they can use pre-build themes according to requirement.

WordPress structure is based on main three directories. Directories names are given below:
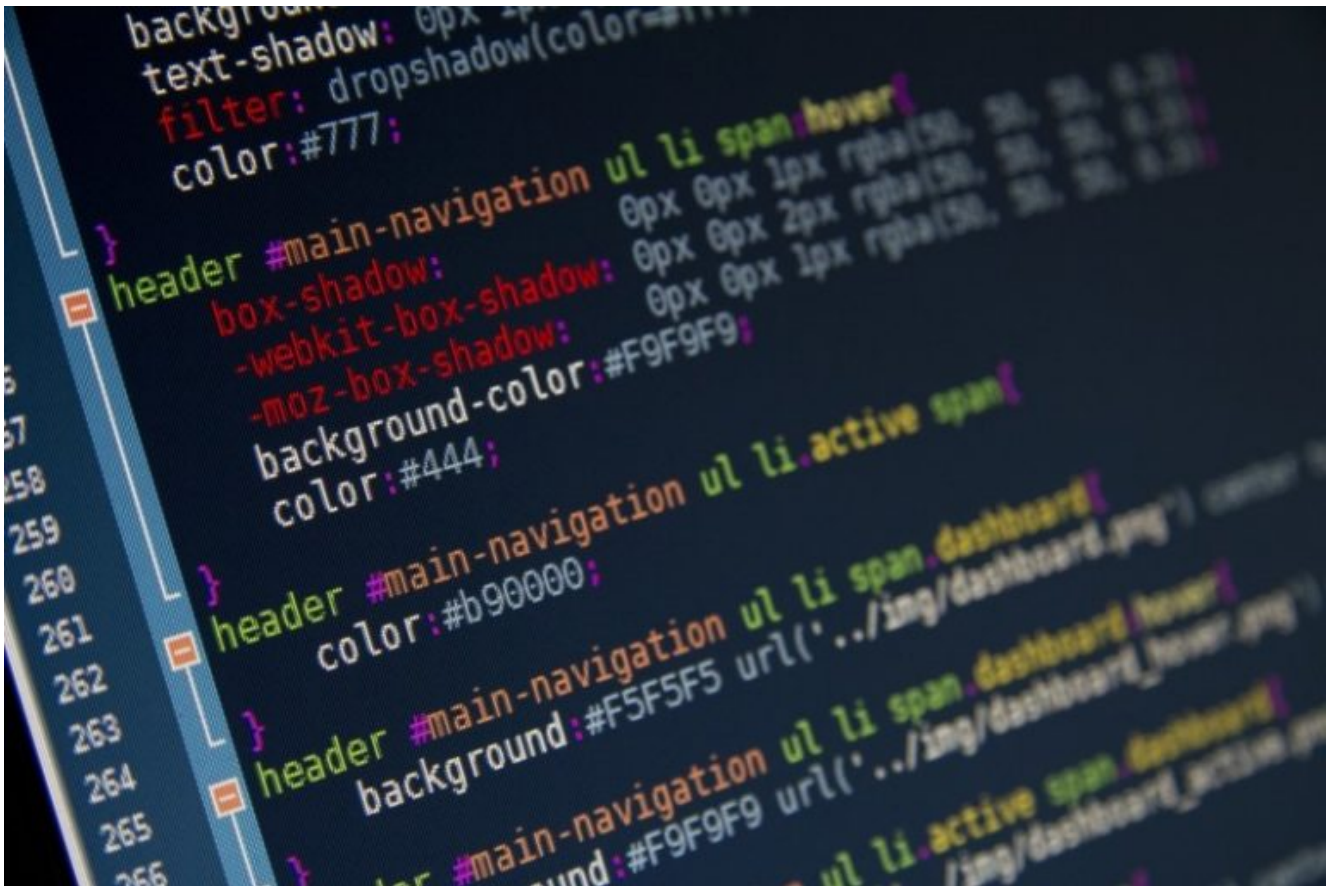1. wp-admin
2. wp-include
3. wp-content

wp-directory has mainly three sub directories.
1. themes
2. plugins
3. uploads

Themes directory has custom themes and premium themes. Same as Plugins directory has custom and premium plugins.

Uploads directory store the media images of the WordPress project.

If any user wants to learn [how to create the WordPress custom theme](#) so please look the next post where we will describe step by step to create custom theme in WordPress.

How to create the WordPress custom theme ?

Now in this post we will create the WordPress custom theme. First of all we will install WordPress on local machine or server. Now we will go inside the wp-content directory. Here we will see four directory with name of themes, plugins, uploads and upgrade. We will go inside the themes directory.

In the Themes directory we will create a folder with the theme name. Please look breadcrumb which is given below then you can understand better.

your project(directory) -> wp-content(directory) -> themes(directory) -> create custom theme directory with name(ex: mytheme).

How to activate the custom

theme in WordPress ?

After creating the custom theme in themes directory then you need to activate this theme. So first of all you need to go in WordPress Admin and then click on the Appearance menu and inside this you will see the Themes menu. Then you need to click on Themes menu and the you will look your custom theme here and then click on activate button. Finally you have activate your theme. Now your project will be run on your custom theme.

Appearance Menu -> Themes -> and then click on activate button of particular theme.