

Прописуємо мета-інформацію про Ваш плагін (наприклад наступне)

```
<?php
    /*Plugin Name: List Subpages Widget
    Description: This widget checks if the current page has parent or child pages
    and if so, outputs a list of the highest ancestor page.
    Version: 0.1
    Author: Anatoly Skidkin
    Author URI:
    License: GPLv2
    */
?>
```

Створюємо новий клас що буде розширенням для WP\_Widget

```
<?php
class List_Pages_Widget extends WP_Widget {
    function __construct() {
    }
    function form( $instance ) {
    }
    function update( $new_instance, $old_instance ) {
    }
    function widget( $args, $instance ) {
    }
}
?>
```

Реєструємо віджет у тому ж файлі

```
<?php
function register_list_pages_widget() {
    register_widget( 'List_Pages_Widget' );
}
add_action( 'widgets_init', 'register_list_pages_widget' );
?>
```

Відредагуйте функцію наступним чином

```
function __construct() {
    parent::__construct(
        // base ID of the widget
        'list_pages_widget',
        // name of the widget
        __('List Related Pages', 'themenname' ),
        // widget options
        array (
            'description' => __( 'Identifies where the current page is in the site structure
and displays a list of pages in the same section of the site. Only works on Pages.',
'themenname' )
        )
    );
}
```

Відредагуйте функцію `form` наступним чином:

```
function form( $instance ) {
    $defaults = array(
        'depth' => '-1'
    );
    $depth = $instance[ 'depth' ];
    // markup for form ?>
    <p>
        <label for="<?php echo $this->get_field_id( 'depth' ); ?>">Depth of list:</label>
        <input class="widefat" type="text" id="<?php echo $this->get_field_id( 'depth' );
    ?>" name="<?php echo $this->get_field_name( 'depth' ); ?>" value="<?php echo
    esc_attr( $depth ); ?>">
    </p>
    <?php
}
```

Відредагуйте функцію `update` наступним чином:

```
function update( $new_instance, $old_instance ) {
    $instance = $old_instance;
    $instance[ 'depth' ] = strip_tags( $new_instance[ 'depth' ] );
    return $instance;
}
```

Додаємо наступне вище класу `WP_widget`

```
<?php
function check_for_page_tree() {
    //start by checking if we're on a page
    if( is_page() ) {
        global $post;
        // next check if the page has parents
        if ( $post->post_parent ){
            // fetch the list of ancestors
            $parents = array_reverse( get_post_ancestors( $post->ID ) );
            // get the top level ancestor
            return $parents[0];
        }
        // return the id - this will be the topmost ancestor if there is one, or the
        current page if not
        return $post->ID;
    }
}
?>
```

Редагуємо функцію `widget`

```
function widget( $args, $instance ) {

    // kick things off
    extract( $args );
    echo $before_widget;
    echo $before_title . 'In this section:' . $after_title;
```

```

// run a query if on a page
if ( is_page() ) {

    // run the check_for_page_tree function to fetch top level page
    $ancestor = check_for_page_tree();

    // set the arguments for children of the ancestor page
    $args = array(
        'child_of' => $ancestor,
        'depth' => $instance[ 'depth' ],
        'title_li' => "",
    );

    // set a value for get_pages to check if it's empty
    $list_pages = get_pages( $args );

    // check if $list_pages has values
    if( $list_pages ) {

        // open a list with the ancestor page at the top
        ?>
        <ul class="page-tree">
            <?php // list ancestor page ?>
            <li class="ancestor">
                <a href="<?php echo get_permalink( $ancestor ); ?>"><?php echo
get_the_title( $ancestor ); ?></a>
            </li>

            <?php
            // use wp_list_pages to list subpages of ancestor or current page
            wp_list_pages( $args );

        // close the page-tree list
        ?>
        </ul>

        <?php
    }
}
}

```

## Функція реєстрації сайдбарів

```
function themename_widgets_init() {  
  
    register_sidebar( array(  
        'name' => __( 'Main Sidebar', 'themename' ),  
        'id' => 'sidebar-1',  
        'description' => __( 'The main sidebar appears on the right on each  
page except the front page template', 'themename' ),  
        'before_widget' => '<aside id="%1$s" class="widget %2$s">',  
        'after_widget' => '</aside>',  
        'before_title' => '<h3 class="widget-title">',  
        'after_title' => '</h3>',  
    ) );  
  
    register_sidebar( array(  
        'name' => __( 'Front page sidebar', 'themename' ),  
        'id' => 'sidebar-2',  
        'description' => __( 'Appears on the static front page template',  
'themename' ),  
        'before_widget' => '<aside id="%1$s" class="widget %2$s">',  
        'after_widget' => '</aside>',  
        'before_title' => '<h3 class="widget-title">',  
        'after_title' => '</h3>',  
    ) );  
}  
  
add_action( 'widgets_init', 'wpb_widgets_init' );
```

у файлах теми, там де нам потрібно відобразити певний сайдбар потрібно додати наступне:

```
<?php if ( is_active_sidebar( 'sidebar-1' ) ) : ?>  
    <div id="secondary" class="widget-area" role="complementary">  
        <?php dynamic_sidebar( 'sidebar-1' ); ?>  
    </div>  
<?php endif; ?>
```