



Open Source Software Development

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



Content Management System

WordPress



Contents

01

Introduction to Widget

02

Built-in versus stand-alone widgets

03

Anatomy of a Widget

04

Developing Widgets



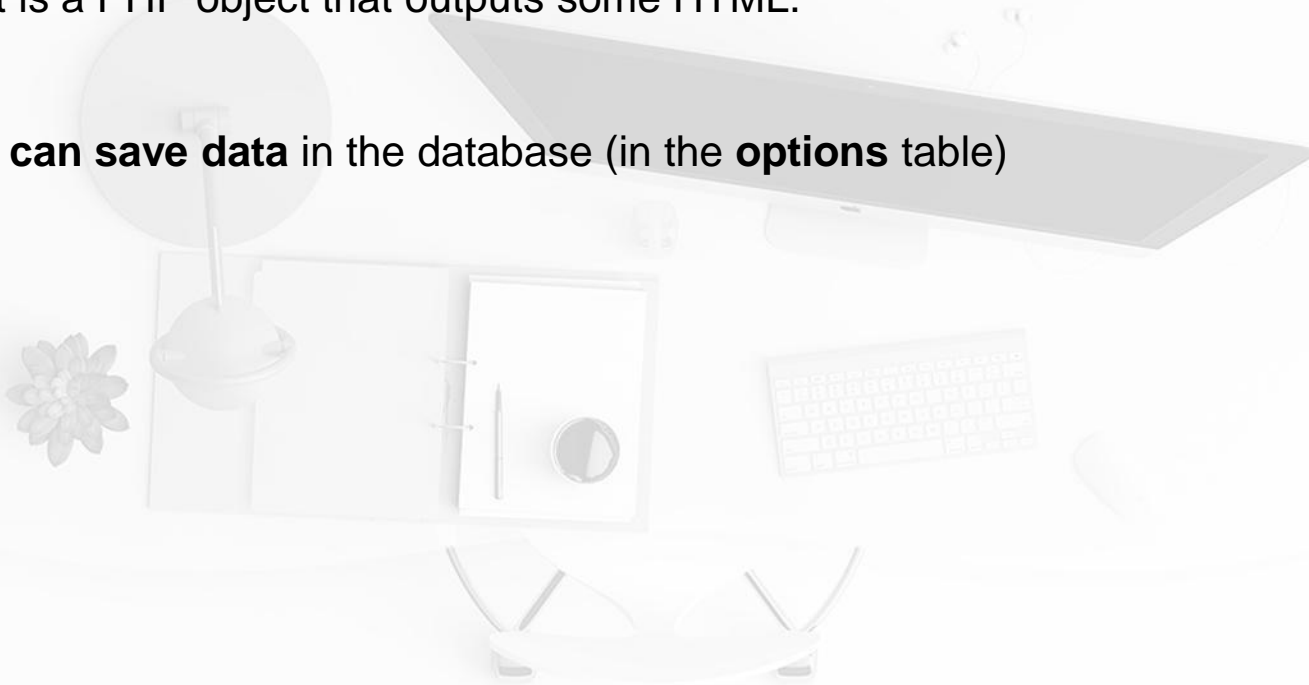
Widgets

Widgets

- A widget **adds content and features to a widget area** (also called a sidebar)
- A widget area can appear on multiple pages or on only one page
- Examples:
 - Lay out a homepage using widgets → this allows site owners to decide what should appear in each section
 - Create a footer that users can customize with their own content
 - Add a customizable sidebar

Widgets

- A widget is a PHP object that outputs some HTML.
- Widgets **can save data** in the database (in the **options** table)

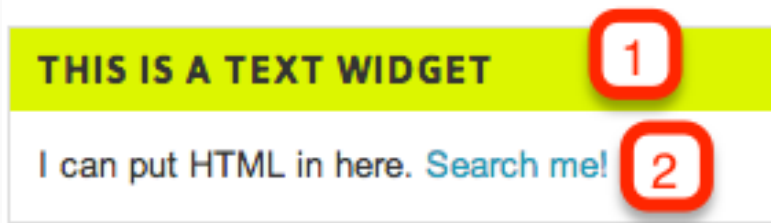


Built-in versus stand-alone widgets

- A set of widgets is included with the default WordPress installation
- In addition to these standard widgets, extra widgets can be included by themes or plugins
- An **advantage** of widgets built into themes or plugins is to provide extra features and increase the number of widgets
- A **disadvantage** is that if a theme is changed or a plugin is deactivated, the plugin or theme widget's functionality will be lost

Anatomy of a Widget

- Visually, a widget comprises two areas:
 - Title Area
 - Widget Options



A widget as it appears to a **site visitor**.

Text

Title:
This is a text widget

I can put HTML in here.
Search me!

☐ Automatically add paragraphs

Delete | Close Save

A widget form in the **admin area**

Developing Widgets

To create and display a widget, you need to:

1. Create your widget's class by extending the standard **WP_Widget class** and some of its functions
2. Register your widget so that it's made available in the **Widgets screen**
3. Make sure that theme has at least one **widget area** in which to add the widgets.

Create your Widget Class

```
<?php

class My_Widget extends WP_Widget {

    public function __construct() {
        // actual widget processes
    }

    public function widget( $args, $instance ) {
        // outputs the content of the widget
    }

    public function form( $instance ) {
        // outputs the options form in the admin
    }

    public function update( $new_instance, $old_instance ) {
        // processes widget options to be saved
    }
}

?>
```

Create your Widget Class

- **construct**

Set up a description, name, and display width in admin

- **widget**

Process the widget options and display the HTML

- **form**

- Display the form to set the options for widget.
- If widget doesn't have any options, skip this function

- **update**

- Save the widget options to the database.
- If widget doesn't have any options, skip this function

Create your Widget Class

Example

```
class Foo_Widget extends WP_Widget {
    // Register widget with WordPress
    public function __construct() {
        parent::__construct(
            'foo_widget', // Base ID
            'Foo_Widget', // Name
            array( 'description' => 'A Foo Widget' ) // Args
        );
    }

    /**
     * Front-end display of widget.
     * @param array $args Widget arguments.
     * @param array $instance Saved values from database.
     */
    public function widget( $args, $instance ) {
        extract( $args );
        $title = apply_filters( 'widget_title', $instance['title'] );

        echo $before_widget;
        if ( ! empty( $title ) ) {
            echo $before_title . $title . $after_title;
        }
        echo 'Hello, World!';
        echo $after_widget;
    }
}
```

Create your Widget Class

Example

```
/**
 * Back-end widget form.
 * @param array $instance Previously saved values from database.
 */
public function form( $instance ) {
    if ( isset( $instance[ 'title' ] ) ) {
        $title = $instance[ 'title' ];
    }
    else {
        $title = 'New title';
    }
    ?>
    <p>
        <label for="<?php echo $this->get_field_name( 'title' ); ?>"><?php _e( 'Title:' ); ?>
        </label>
        <input class="widefat" id="<?php echo $this->get_field_id( 'title' ); ?>" name="<?php
            echo $this->get_field_name( 'title' ); ?>" type="text" value="<?php echo
            esc_attr( $title ); ?>" />
    </p>
    <?php
}
```

Create your Widget Class

Example

```
/**
 * Sanitize widget form values as they are saved.
 * @see WP_Widget::update()
 * @param array $new_instance Values just sent to be saved.
 * @param array $old_instance Previously saved values from database.
 * @return array Updated safe values to be saved.
 */
public function update( $new_instance, $old_instance ) {
    $instance = array();
    $instance['title'] = ( !empty( $new_instance['title'] ) ) ? strip_tags( $new_instance['title'] ) : '';

    return $instance;
}
```

PHP extract() Function

- The extract() function imports variables into the local symbol table from an array.
- This function **uses array keys as variable names and values as variable values**.
- This function returns the number of variables extracted on success.
- Example: Assign the values “Cat”, “Dog” and “Horse” to the variables \$a, \$b and \$c

```
<?php
$a = "Original";
$my_array = array("a" => "Cat", "b" => "Dog", "c" => "Horse");
extract($my_array);
echo "\$a = $a; \$b = $b; \$c = $c";
?>
```

Registering a Widget

- The **register_widget()** function is used to register a widget
- Call this function using the **widgets_init** hook:

```
<?php
add_action( 'widgets_init', 'wpdocs_register_widgets' );

function wpdocs_register_widgets() {
    register_widget( 'My_Widget' );
}
?>
```


Register the widget area / Sidebar

- A sidebar is any widgetized area of your theme.
- Widget areas are places in theme where users can add their own widgets.
- You do not need to include a sidebar in theme, but including a sidebar means users can add content to the widget areas through the **Customizer** or the **Widgets Admin Panel**

Register the widget area / Sidebar

- To use sidebars, you must register them in **functions.php**
- **register_sidebar()** has several parameters
 - **name** – the sidebar name users will see in the Widgets panel
 - **id** – must be lowercase; call this in theme using the **dynamic_sidebar** function
 - **description** – the sidebar description will also be shown in the admin Widgets panel.
 - **class** – The CSS class name to assign to the widget's HTML.

Register the widget area / Sidebar

- To use sidebars, you must register them in **functions.php**
- **register_sidebar()** has several parameters
 - **before_widget** – HTML that is placed before every widget
 - **after_widget** – HTML that is placed after every widget.
 - **before_title** – HTML that is placed before the title of each widget
 - **after_title** – HTML that is placed after every title

Register the widget area / Sidebar

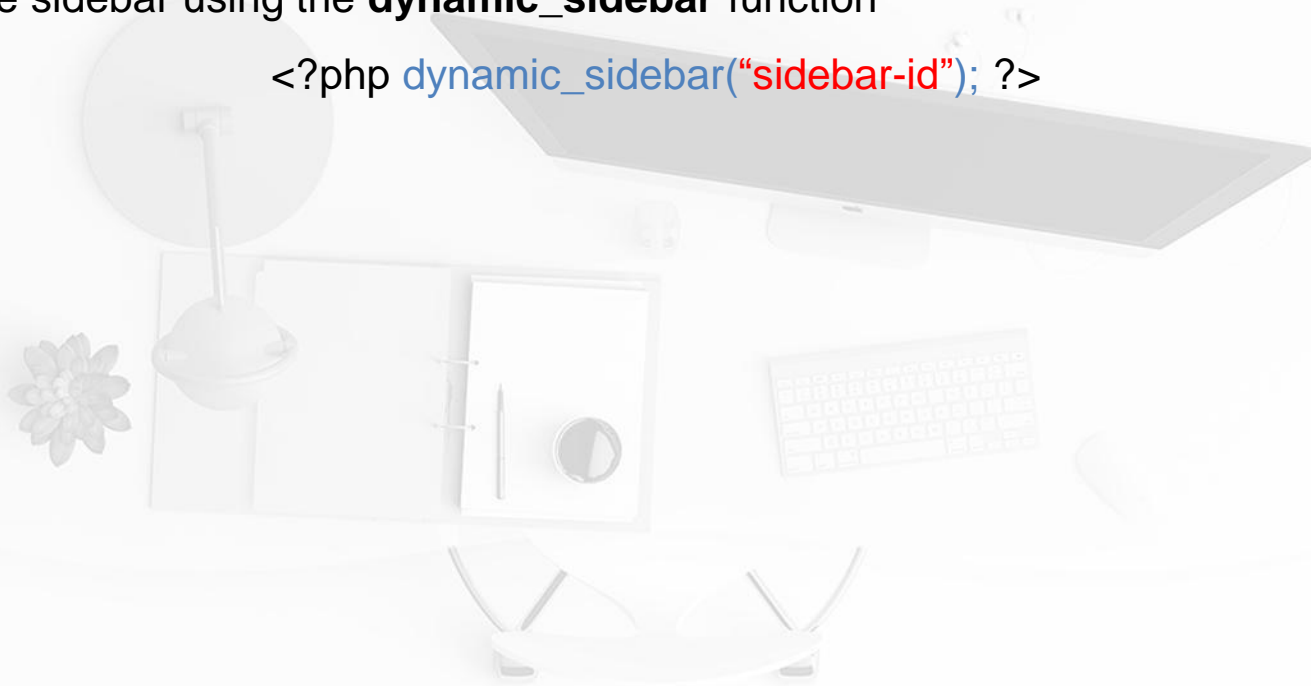
Add the following code to functions.php to registers a sidebar:


```
add_action( 'widgets_init', 'my_register_sidebars' );
function my_register_sidebars() {
    /* Register the 'primary' sidebar. */
    register_sidebar(
        array(
            'id'          => 'primary',
            'name'         => __( 'Primary Sidebar' ),
            'description'  => __( 'A short description of the sidebar.' ),
            'before_widget' => '<div id="%1$s" class="widget %2$s">',
            'after_widget'  => '</div>',
            'before_title'  => '<h3 class="widget-title">',
            'after_title'   => '</h3>',
        )
    );
    /* Repeat register_sidebar() code for additional sidebars. */
}
```

Displaying Sidebars in your Theme

Display the sidebar using the **dynamic_sidebar** function

```
<?php dynamic_sidebar("sidebar-id"); ?>
```





Q&A

Topics

▪ wpdb

- WordPress database access abstraction class.
- <https://developer.wordpress.org/reference/classes/wpdb/#select-generic-results>

▪ Fontawesome.com/v4.7.0/cheatsheet/

<https://fontawesome.com/v4.7.0/cheatsheet/>

Topics

- **Basic Usage**

<https://wordpress.org/support/category/basic-usage/>

- **First Steps with WordPress (Classic Editor)**

<https://wordpress.org/support/article/first-steps-with-wordpress-classic/>

- **Writing Posts**

<https://wordpress.org/support/article/writing-posts/>

- **Using Permalinks**

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

Topics

- **Basic Administration**

<https://wordpress.org/support/category/basic-administration/>

- **Administration Screens**

<https://wordpress.org/support/article/administration-screens/#top>

Topics

- **Customizing**

<https://wordpress.org/support/category/customizing/>

- **Using Themes**

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

- **Appearance Menu Screen**

<https://wordpress.org/support/article/appearance-menus-screen/>

Topics

- **Maintenance**

<https://wordpress.org/support/category/maintenance/>

- **Moving WordPress**

<https://wordpress.org/support/article/moving-wordpress/>

- **Security**

<https://wordpress.org/support/category/security/>

- **Advanced Topics**

<https://wordpress.org/support/category/advanced-topics/>

- **Plugin Handbook**

<https://developer.wordpress.org/plugins/>

Topics

- **WordPress Support**

<https://wordpress.org/support/>

- **Developer Resources**

<https://developer.wordpress.org/>

- **Main Page**

<https://codex.wordpress.org/>