

# WordPress & Hosting Techniques



# Table of Contents

<b>About</b> .....	<b>1</b>
<b>Chapter 1: Getting started with WordPress</b> .....	<b>2</b>
Remarks .....	2
Versions.....	3
Examples.....	4
Introduction to WordPress.....	4
WordPress Themes.....	5
<b>Mapping URLs to specific templates</b> .....	<b>5</b>
<b>Basic Theme Directory Structure</b> .....	<b>5</b>
<b>Example of a "Single" (template for an individual post)</b> .....	<b>6</b>
<b>Example of an "Archive" (template for a list of multiple posts)</b> .....	<b>7</b>
<b>Posts, Pages, Custom Post Types, and Custom Fields</b> .....	<b>7</b>
<b>Chapter 2: Actions and Filters</b> .....	<b>8</b>
Syntax.....	8
Parameters .....	8
Remarks.....	8
Examples.....	11
add_action - init.....	11
add_action - init - anonymous function.....	11
add_action - init - within class object.....	11
add_action - init - within static class.....	12
<b>Chapter 3: Add Shortcode</b> .....	<b>13</b>
Syntax.....	13
Parameters.....	13
Remarks.....	13
Examples.....	13
Simple shortcode for recent post.....	13
Advanced shortcode for recent posts.....	14
<b>Chapter 4: Add/remove contact info for users with user_contactmethods filter hook</b> .....	<b>15</b>

Examples.....	15
Enabling most popular social networks.....	15
Removing contact method.....	16
<b>Chapter 5: add_action().....</b>	<b>17</b>
Syntax.....	17
Parameters.....	17
Examples.....	17
Direct function callback.....	17
Function name reference callback.....	17
Class static method callback.....	18
Object method callback.....	18
<b>Chapter 6: add_editor_style().....</b>	<b>19</b>
Introduction.....	19
Syntax.....	19
Parameters.....	19
Examples.....	19
Loading a single css file.....	19
<b>Chapter 7: add_menu_page().....</b>	<b>20</b>
Introduction.....	20
Syntax.....	20
Parameters.....	20
Remarks.....	20
Examples.....	21
Adding the "Theme page title" item to the nav bar.....	21
OOP & how to load scripts/styles on menu page.....	22
<b>Chapter 8: add_submenu_page().....</b>	<b>24</b>
Introduction.....	24
Syntax.....	24
Parameters.....	24
Remarks.....	24
Examples.....	25
Adding the "Submenu Page" as a sub-page of "Tools" to the nav bar.....	25

<b>Chapter 9: add_theme_support()</b> .....	<b>27</b>
Introduction.....	27
Syntax.....	27
Parameters.....	27
Remarks.....	27
Examples.....	27
Adding theme support for post formats.....	27
Adding theme support for post thumbnails to posts.....	27
<b>Chapter 10: Admin Dashboard Widgets</b> .....	<b>29</b>
Introduction.....	29
Syntax.....	29
Parameters.....	29
Examples.....	29
Simple widget (displays text).....	29
<b>Chapter 11: AJAX</b> .....	<b>31</b>
Examples.....	31
AJAX request with a JSON response.....	31
AJAX with .ajax() and WordPress Nonce.....	32
wp_ajax - core functionality + _wpnonce check.....	33
OOP ajax submission using a simple class with nonce.....	34
<b>Chapter 12: Alternating main loop (pre_get_posts filter)</b> .....	<b>38</b>
Syntax.....	38
Parameters.....	38
Remarks.....	38
Examples.....	38
Even more specific loop targeting.....	38
Show posts from only one category.....	39
Pre get posts filter basic usage.....	39
Exclude category from posts list edit share.....	39
Change posts_per_page for main loop.....	40
Targeting only main WordPress loop.....	40
<b>Chapter 13: Child Theme Basics</b> .....	<b>41</b>

Syntax.....	41
Remarks	41
Examples.....	41
2) The purpose.....	41
Definition and requirements.....	42
3) Template overwriting.....	42
Assets replacement.....	43
<b>Chapter 14: Create a Post Programmatically.....</b>	<b>45</b>
Syntax.....	45
Parameters.....	45
Remarks.....	45
<b>Arguments.....</b>	<b>45</b>
<b>Avoid Duplicated Posts.....</b>	<b>46</b>
Explanation .....	47
Examples.....	47
Introduction.....	47
Create a Basic Post.....	47
Create a Basic Page.....	47
<b>Chapter 15: Create Template for Custom Post Type.....</b>	<b>49</b>
Examples.....	49
Creating a custom template for Custom Post type book.....	49
Custom Post Type Templates.....	49
<b>Custom Post Type Archive:.....</b>	<b>49</b>
<b>Custom Post Type Single template:.....</b>	<b>51</b>
<b>Chapter 16: Creating a custom template.....</b>	<b>53</b>
Examples .....	53
Creating basic blank template.....	53
Including header and footer in our template.....	54
Custom template with content.....	54
<b>Chapter 17: Custom excerpts with excerpt_length and excerpt_more.....</b>	<b>56</b>
Examples.....	56

Limit excerpt length to 50 words	56
Adding a Read More link at the end of the excerpt	56
Adding a few dots at the end of the excerpt	57
<b>Chapter 18: Custom Post Types</b>	<b>59</b>
Syntax	59
Parameters	59
Examples	59
Registering a Custom Post Type	59
Add Custom Post Types to Main Query	60
Adding Custom Post Types to Main RSS Feed	61
Register Custom Post Type	61
Custom Post Type using Twenty Fifteen WordPress Theme	62
Custom post type in default search	63
<b>Chapter 19: Customizer Basics (Add Panel, Section, Setting, Control)</b>	<b>64</b>
Examples	64
Add a Customizer Panel	64
Add a Customizer Section With Basic Settings and their Controls	64
<b>Chapter 20: Customizer Hello World</b>	<b>68</b>
Parameters	68
Examples	68
Hello World Example	68
<b>Chapter 21: Debugging</b>	<b>70</b>
Introduction	70
Remarks	70
Examples	70
WP_DEBUG	70
WP_DEBUG_LOG	70
WP_DEBUG_DISPLAY	70
SCRIPT_DEBUG	71
SAVEQUERIES	71
Example wp-config.php and good practices for Debugging	71
See logs in a separate file	72

<b>Chapter 22: Enqueuing scripts</b>	<b>73</b>
Syntax	73
Parameters	73
Examples	73
Enqueuing scripts in functions.php	73
Enqueue scripts for IE only	74
Enqueuing Scripts conditionally for specific pages	74
<b>Chapter 23: Enqueuing Styles</b>	<b>76</b>
Syntax	76
Parameters	76
Examples	76
Including internal css file with another css file as a dependency	76
Including internal css file	76
Including external css file	76
Enqueue stylesheets for IE only	77
Including internal css file for your Plugin class	77
Add Alternative Stylesheets	77
<b>Chapter 24: Function : wp_trim_words()</b>	<b>79</b>
Syntax	79
Parameters	79
Examples	79
Trimming post content	79
<b>Chapter 25: Function: add_action()</b>	<b>80</b>
Syntax	80
Parameters	80
Remarks	80
Examples	80
Basic Action Hook	80
Action Hook Priority	81
Hooking Class & Object Methods to Actions	81
<b>Object Method Action Hooks</b>	<b>82</b>
<b>Class Method Action Hooks</b>	<b>82</b>

<b>Chapter 26: get_bloginfo()</b> .....	<b>84</b>
Introduction.....	84
Syntax.....	84
Parameters.....	84
Remarks.....	84
Examples.....	86
Getting the site title.....	86
Getting the site tagline.....	87
Getting the active theme URL.....	88
Get site url.....	89
Get Email Address of Site Administrator.....	89
<b>Chapter 27: get_home_path()</b> .....	<b>90</b>
Introduction.....	90
Parameters.....	90
Remarks.....	90
Important difference between get_home_path() and ABSPATH.....	90
Using it in your code.....	91
Examples.....	91
Usage.....	91
<b>Chapter 28: get_option()</b> .....	<b>92</b>
Introduction.....	92
Syntax.....	92
Parameters.....	92
Remarks.....	92
Examples.....	92
Show blog title.....	92
<b>Name of the blog in H1 style</b> .....	<b>93</b>
Show Character Set.....	93
Handling non-existing options.....	93
<b>Chapter 29: get_permalink()</b> .....	<b>94</b>
Introduction.....	94

Syntax.....	94
Parameters.....	94
Remarks.....	94
Examples.....	94
Simple use of get_parmalink.....	94
Specifying the post to get the link.....	94
Get the link without the post's name.....	95
<b>Chapter 30: get_template_part().....</b>	<b>96</b>
Syntax.....	96
Parameters.....	96
Examples.....	96
Load a template part from a subfolder.....	96
Get a specific file.....	96
<b>Chapter 31: get_template_part().....</b>	<b>97</b>
Introduction.....	97
Syntax.....	97
Parameters.....	97
Examples.....	97
Including a custom template.....	97
Including a custom template with a dash-separated filename.....	97
Including a custom template from inside a directory.....	98
Including a custom template with a dash-separated filename located inside a directory.....	98
Passing variable to custom template scope.....	98
<b>Chapter 32: get_template_part().....</b>	<b>99</b>
Syntax.....	99
Parameters.....	99
Examples.....	99
Loading Template Part.....	99
<b>Chapter 33: get_the_category().....</b>	<b>100</b>
Introduction.....	100
Syntax.....	100
Parameters.....	100

Remarks.....	100
Examples.....	100
Get all names of categories of the post.....	101
Get all ids of categories of the post.....	101
<b>Chapter 34: get_the_title().....</b>	<b>102</b>
Introduction.....	102
Syntax.....	102
Parameters.....	102
Remarks.....	102
Examples.....	102
Simple use of get_the_title.....	102
Get the title of a specified post id.....	102
<b>Chapter 35: home_url().....</b>	<b>104</b>
Syntax.....	104
Parameters.....	104
Examples.....	104
Getting the Home URL.....	104
Site url.....	104
<b>Chapter 36: How Can I integrate Markdown editor with Advance Custom Field's repeater Add-o</b>	<b>105</b>
Examples.....	105
Add MarkDown Editor.....	105
<b>Chapter 37: init.....</b>	<b>106</b>
Syntax.....	106
Remarks.....	106
Examples.....	106
Processing \$_POST request data.....	106
Processing \$_GET request data.....	106
Registering a custom post type.....	106
<b>Chapter 38: Installation and Configuration.....</b>	<b>107</b>
Examples.....	107
Wordpress on LAMP.....	107

Installation WP in MAMP .....	108
<b>Chapter 39: Making network requests with HTTP API .....</b>	<b>110</b>
Syntax .....	110
Parameters .....	110
Remarks .....	110
Returns .....	110
Examples .....	110
GET a remote JSON resource .....	110
<b>Chapter 40: Meta Box .....</b>	<b>111</b>
Introduction .....	111
Syntax .....	111
Remarks .....	111
Examples .....	111
A simple example with a regular input and a select input .....	111
<b>Chapter 41: Options API .....</b>	<b>114</b>
Introduction .....	114
Syntax .....	114
Remarks .....	114
Examples .....	114
get_option .....	114
add_option .....	115
delete_option .....	115
update_option .....	115
<b>Chapter 42: Plugin development .....</b>	<b>116</b>
Syntax .....	116
Parameters .....	116
Remarks .....	116
Examples .....	117
Filter .....	117
Action .....	117
Plugin development examples : Favorite Song Widget .....	117

<b>Chapter 43: Post Formats</b> .....	<b>119</b>
Remarks .....	119
Examples.....	119
Adding post type to Theme.....	119
Add post-formats to post_type 'page'.....	119
Register custom post type 'my_custom_post_type'.....	119
Add post-formats to post_type 'my_custom_post_type'.....	120
Register custom post type 'my_custom_post_type' with 'supports' parameter.....	120
Add Theme Support for post.....	120
Function Call.....	120
<b>Chapter 44: Querying posts</b> .....	<b>121</b>
Syntax.....	121
Parameters.....	121
Remarks.....	121
Never use query_posts().....	121
Examples.....	122
Using WP_Query() object.....	122
Using get_posts().....	122
<b>Chapter 45: Remove Auto Line Breaks From Content and Excerpt</b> .....	<b>124</b>
Introduction.....	124
Remarks.....	124
Examples.....	124
Remove the Filters.....	124
Function to remove the filters.....	124
<b>Chapter 46: Remove Version from Wordpress and Stylesheets</b> .....	<b>126</b>
Introduction.....	126
Syntax.....	126
Parameters.....	126
Remarks.....	126
Remove version numbers from css/js.....	126
Remove version numbers from WordPress.....	127
<b>Chapter 47: REST API</b> .....	<b>128</b>

Introduction	128
Remarks.....	128
Examples.....	128
Complete working example.....	129
<b>Chapter 48: Run WordPress local with XAMPP.....</b>	<b>130</b>
Introduction.....	130
Examples.....	130
1. Installing XAMPP.....	130
2. Setup a database after installing XAMPP.....	130
3. Installing WordPress after setup the database.....	130
<b>Chapter 49: Secure your installation.....</b>	<b>132</b>
Remarks.....	132
Examples.....	132
Disable File Editor.....	132
Move wp-config.php.....	132
Set a custom prefix for WordPress tables.....	133
<b>Chapter 50: Security in WordPress - Escaping.....</b>	<b>138</b>
Syntax.....	138
Remarks.....	138
Examples.....	138
escape data in HTML code.....	138
escape a url.....	138
escape data in js code.....	138
escape attributes.....	139
escape data in textarea.....	139
<b>Chapter 51: Security in WordPress - Sanitization.....</b>	<b>140</b>
Syntax.....	140
Remarks.....	140
Sanitize text field.....	140
Sanitize title.....	140
Sanitize email.....	141
Sanitize html class.....	141

Sanitize file name .....	141
Sanitize user name.....	141
<b>Chapter 52: Shortcode .....</b>	<b>142</b>
Examples.....	142
Registering shortcode.....	142
Using Shortcodes in WordPress Backend.....	142
Adding New Shortcodes.....	142
Using Shortcodes Inside PHP Code (themes and plugins).....	143
Using Shortcodes in Widgets.....	143
<b>Chapter 53: Shortcode with attribute .....</b>	<b>144</b>
Syntax.....	144
Parameters.....	144
Remarks.....	144
Examples.....	144
Examples of Shortcodes.....	144
Creating a self-closing shortcode.....	144
Creating a self-closing shortcode with parameters.....	145
Creating an enclosing shortcode.....	145
Shortcodes in Widgets.....	146
<b>Chapter 54: Shortcodes .....</b>	<b>147</b>
Examples.....	147
Shortcode introduction.....	147
Button shortcode.....	147
<b>Chapter 55: Sidebars .....</b>	<b>149</b>
Syntax.....	149
Parameters.....	149
Remarks.....	149
Registering sidebars.....	149
Get Sidebar.....	150
<b>Chapter 56: Site Migration .....</b>	<b>151</b>
Syntax.....	151
Examples.....	151

Updating the tables with a new URL .....	151
<b>Chapter 57: Taxonomies .....</b>	<b>152</b>
Syntax .....	152
Parameters .....	152
Examples .....	152
Example of registering a taxonomy for genres .....	152
Add category in page .....	153
Add Categories and Tags to Pages and insert them as class into .....	153
Add Categories and Tags to Pages and insert as class into .....	154
<b>Chapter 58: Template hierarchy .....</b>	<b>156</b>
Remarks .....	156
Examples .....	156
Introduction .....	156
Debugging .....	158
<b>Chapter 59: template_include .....</b>	<b>159</b>
Parameters .....	159
Remarks .....	159
Examples .....	159
Simple example .....	159
More Adv example .....	160
<b>Chapter 60: The \$wpdb Object .....</b>	<b>161</b>
Remarks .....	161
Examples .....	161
Selecting a variable .....	161
Selecting multiple rows .....	161
<b>Chapter 61: The Admin Bar (aka "The Toolbar") .....</b>	<b>163</b>
Remarks .....	163
Examples .....	163
Removing the Admin Toolbar from all except Administrators .....	163
Removing the Admin toolbar using filters .....	163
How to Remove WordPress Logo From Admin Bar .....	163
Add your custom logo and custom link on admin login page .....	164

<b>Chapter 62: The Loop (main WordPress loop)</b>	<b>165</b>
Examples	165
Basic WordPress loop structure	165
Alternative loop syntax	165
Handling no items in the loop	165
<b>Chapter 63: the_title()</b>	<b>167</b>
Introduction	167
Syntax	167
Parameters	167
Remarks	167
Examples	167
Simple use of the_title	167
Printing the title with code before and after	167
<b>Chapter 64: Themes</b>	<b>169</b>
Introduction	169
Examples	169
WordPress Themes	169
How To Choose A Theme	169
Update Available	169
Install Themes	170
Creating A Custom Theme	171
<b>Chapter 65: Update WordPress Manually</b>	<b>172</b>
Examples	172
VIA FTP	172
<b>Chapter 66: WordPress Plugin creation</b>	<b>173</b>
Introduction	173
Examples	173
Minimal Setup of a Plugin Folder and Files	173
<b>Chapter 67: Wordpress theme and child-theme development</b>	<b>175</b>
Introduction	175
Examples	175

Developing your own theme .....	175
<b>Chapter 68: wp_get_current_user()</b> .....	<b>178</b>
Syntax.....	178
Examples.....	178
Getting the current user.....	178
Use foreach loop to get user info from wp_get_current_user().....	178
<b>Chapter 69: wp_get_current_user()</b> .....	<b>179</b>
Examples.....	179
Get current loggedin user information.....	179
<b>Chapter 70: WP_Query() Loop</b> .....	<b>180</b>
Introduction.....	180
Examples.....	180
Retrieve latest 10 posts.....	180
<b>Chapter 71: WP-CLI</b> .....	<b>181</b>
Introduction.....	181
Examples.....	181
Manage themes.....	181
Manage plugins.....	181
Manage WP-CLI itself.....	182
Download, install, update and manage a WordPress install.....	183
Manage users.....	183
Perform basic database operations using credentials stored in wp-config.php.....	183
<b>Chapter 72: WP-Cron</b> .....	<b>185</b>
Examples.....	185
wp_schedule_event() example.....	185
custom recurrence interval in wp_schedule_event().....	185



---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [wordpress](#)

It is an unofficial and free WordPress ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official WordPress.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with WordPress

## Remarks



ଓର୍ଡ଼ପ୍ରେସ୍ ହେଉଛି ଏକ ମୁକ୍ତ ଉତ୍ସ ବିଷୟବସ୍ତୁ ପରିଚାଳନା ସିଷ୍ଟମ୍ (CMS) ଯାହା ୱେବସାଇଟ୍ ଗଠନ ଏବଂ ପରିଚାଳନା ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ଓର୍ଡ଼ପ୍ରେସ୍ ହେଉଛି ଏକ ଦେଶ ମାଲିକ୍ ଦ୍ୱାରା ଇଣ୍ଟରନେଟରେ ସବୁଠାରୁ ଲୋକପ୍ରିୟ CMS, ଲେଖିବା ସମୟରେ ସମସ୍ତ CMS ୱେବସାଇଟର ପ୍ରାୟ ଅଧା ଏବଂ ଇଣ୍ଟରନେଟରେ ସମସ୍ତ ୱେବସାଇଟ୍ ର ଏକ ଚତୁର୍ଥାଂଶ ଶକ୍ତି ପ୍ରଦାନ କରିଥାଏ ।

ଓର୍ଡ଼ପ୍ରେସ୍ ବ୍ଲଗିଂ ପାଇଁ ଏକ ପ୍ଲାଟଫର୍ମ ଭାବରେ ଜୀବନ ଆରମ୍ଭ କରିଥିଲା କିନ୍ତୁ ଅଧିକାଂଶ ପ୍ରକାରର ୱେବସାଇଟ୍ ପାଇଁ ଉପଯୁକ୍ତ ହେବା ପାଇଁ ବର୍ଷ ବର୍ଷ ଧରି ବିକଶିତ ହୋଇଛି । ଜ୍ଞାନକୁ କୋଡିଂ ନକରି ଇଣ୍ଟରଫେସ୍ ବ୍ୟବହାର କରାଯାଇପାରିବ ଏବଂ ଏହାକୁ ନୂତନ ତଥା ବିକାଶକାରୀଙ୍କ ପାଇଁ ଲୋକପ୍ରିୟ କରାଏ, ଯେଉଁମାନେ ନିଜ ଗ୍ରାହକମାନଙ୍କୁ ନିଜ ୱେବସାଇଟ୍ ପରିଚାଳନା କରିବାକୁ ସଶକ୍ତ କରିବାକୁ ଚାହୁଁଛନ୍ତି ।

ଓର୍ଡ଼ପ୍ରେସର ଲୋକପ୍ରିୟତାର ଅନ୍ୟ ଏକ ବଡ଼ କାରଣ ହେଉଛି ଏହା ନମନୀୟତା, ମୁଖ୍ୟତଃ the କୋରର ପ୍ଲଗିନ୍ ଏବଂ ଥିମ୍ ସିଷ୍ଟମ୍ ଯୋଗୁଁ । ପ୍ଲଗିନ୍ ସିଷ୍ଟମ୍ କୋର କୋଡ୍ ପରିବର୍ତ୍ତନ ନକରି ମୂଳ କାର୍ଯ୍ୟକାରିତାକୁ ବିସ୍ତାର କରିବା ସହଜ କରିଥାଏ । ସମାନ manner ଙ୍କରେ, ଥିମ୍ ସିଷ୍ଟମ୍ ୱେବସାଇଟ୍ ର ଲେଆଉଟ୍ ଏବଂ ସ est କାର୍ଯ୍ୟକୁ ପରିବର୍ତ୍ତନ କରିବା ସହଜ କରିଥାଏ । ବର୍ତ୍ତମାନ ସେଠାରେ ହଜାର ହଜାର ମାଗଣା ଏବଂ ପ୍ରିମିୟମ୍ ଓର୍ଡ଼ପ୍ରେସ୍ ପ୍ଲଗିନ୍ ଏବଂ ଥିମ୍ ଉପଲବ୍ଧ । ଏଥିମଧ୍ୟରୁ ଅନେକ ଯଥାକ୍ରମେ ଓର୍ଡ଼ପ୍ରେସ୍ ଡେଭଲପର ପ୍ଲଗିନ୍ ରେପୋଜିଟୋରୀ ଏବଂ ଥିମ୍ ରେପୋଜିଟୋରୀରେ ଅବସ୍ଥିତ ।

ଓର୍ଡ଼ପ୍ରେସ୍ ଏହାର ନିଜସ୍ୱ ସମ୍ପ୍ରଦାୟ ଦ୍ୱାରା ବିକଶିତ ହୋଇଛି, କିନ୍ତୁ ଅଟୋମାଟିକ୍ କମ୍ୟୁନିଟି ସହିତ ଦୃ strongly ଭାବରେ ଜଡ଼ିତ, ଯାହା ଅନେକ ଓର୍ଡ଼ପ୍ରେସର ମୂଳ ବିକାଶକାରୀଙ୍କୁ ନିୟୋଜିତ କରିଥାଏ ।

## Code

ଝାଡ଼ପ୍ରେସ୍ PHP ସର୍ଭର ସ୍କ୍ରିପ୍ଟ ଭାଷା ଏବଂ MySQL ଜିଜ୍ଞାସା ଭାଷା ଉପରେ ନିର୍ମିତ । ଝାଡ଼ପ୍ରେସ୍ MySQL କୁ ଉପଭୋକ୍ତା ବିଷୟବସ୍ତୁ ଏବଂ ବିନ୍ୟାସ ପାଇଁ ତାତାକାଷ୍ଟ୍ରେ ଭାବରେ ବ୍ୟବହାର କରେ । PHP ସମସ୍ତ ଆବଶ୍ୟକୀୟ ସମ୍ପୂର୍ଣ୍ଣ ସହିତ ବିଷୟବସ୍ତୁ ତଥ୍ୟକୁ HTML ଝାଡ଼ପ୍ରେସ୍ରେ ବିବାଦ କରିଥାଏ ।

## wordpress.com vs wordpress.org

ଅଟୋମେଟିକ୍ ର ଝାଡ଼ପ୍ରେସ୍ ଡଟ୍ ସେବା ପାଇଁ ସାଇନ୍ ଅପ୍ କରି ଏବଂ ଆପଣଙ୍କର ସର୍ଭରରେ ଆପଣଙ୍କର ଝାଡ଼ପ୍ରେସ୍ ହୋଷ୍ଟ କରି ଆପଣ ଝାଡ଼ପ୍ରେସ୍ ବ୍ୟବହାର କରିପାରିବେ, କିମ୍ବା ଆପଣ ଝାଡ଼ପ୍ରେସ୍ ସଫ୍ଟୱେର୍ କୁ ଝାଡ଼ପ୍ରେସ୍ ଡଟ୍ ରୁ ତାଉନଲୋଡ୍ କରିପାରିବେ ଏବଂ ଆପଣଙ୍କ ନିୟନ୍ତ୍ରଣରେ ଥିବା ଏକ ସର୍ଭରରେ ଆପଣଙ୍କର ଝାଡ଼ପ୍ରେସ୍ ହୋଷ୍ଟ କରିପାରିବେ । ପ୍ରଥମ ବିକଳ୍ପଟି ସହଜ କିନ୍ତୁ ଆପଣ କି site ଶସି ସାଇଟ୍ କୋଡ୍ ସଂପାଦନ କରିପାରିବେ ନାହିଁ । ଆପଣ କେବଳ ଝାଡ଼ପ୍ରେସ୍ ଇଣ୍ଟରଫେସ୍ ମାଧ୍ୟମରେ ପରିବର୍ତ୍ତନ କରିପାରିବେ । ଦ୍ୱିତୀୟ ବିକଳ୍ପ ଅଧିକ କାର୍ଯ୍ୟ ଆବଶ୍ୟକ କରେ କିନ୍ତୁ ତୁମର ଝାଡ଼ପ୍ରେସ୍ କୋଡ୍ ସହିତ ଯାହା ଇଚ୍ଛା ତାହା କରିବାକୁ ତୁମକୁ ନମନୀୟତା ପ୍ରଦାନ କରେ । ଯଦି ଆପଣ ଏକ ଷ୍ଟାକଅଭରକ୍ଲୋ ଉପଭୋକ୍ତା, ଆପଣ ବୋଧହୁଏ ଦ୍ୱିତୀୟ ବିକଳ୍ପ ସହିତ ଯିବେ ।

## Open Source

ଝାଡ଼ପ୍ରେସ୍ ହେଉଛି ମୁକ୍ତ ଉତ୍ସ ସଫ୍ଟୱେର୍ ଅର୍ଥାତ୍ ଏହା ବ୍ୟବହାର କରିବା ମାଗଣା ଏବଂ ଯେକେହି ଉତ୍ସ କୋଡ୍ ଦେଖିପାରିବେ ଏବଂ ଏଥିରେ ସହଯୋଗ କରିପାରିବେ । ଝାଡ଼ପ୍ରେସ୍ କୋଡ୍ରେକ୍ଟର ଅବଦାନ ପୃଷ୍ଠା ପ reading ି ସମ୍ଭାବ୍ୟ ସହଯୋଗୀମାନେ ଆରମ୍ଭ କରିପାରିବେ ..

ଝାଡ଼ପ୍ରେସ୍ ଟିକେଟ୍ ଟ୍ରାକରରେ ଏକ ବଗ୍ ଦାଖଲ କରି ବଗ୍ ରିପୋର୍ଟ କରାଯାଇପାରେ ।( Bugs can be reported by submitting a bug on the WordPress ticket tracker.)

## Documentation

ଝାଡ଼ପ୍ରେସ୍ ଆନୁଷ୍ଠାନିକ ଭାବରେ ଝାଡ଼ପ୍ରେସ୍ କୋଡ୍ରେକ୍ଟର ଝାଡ଼ପ୍ରେସ୍ ଡଟ୍ । ଝାଡ଼ପ୍ରେସ୍ ସହିତ କାର୍ଯ୍ୟ କରୁଥିବା ଡେଭଲପର୍ମାନେ ଡେଭଲପର୍ କୋଡ୍ରେକ୍ଟ ବିଭାଗ ଏବଂ ଝାଡ଼ପ୍ରେସ୍ ଡଟ୍ ର ଡେଭଲପର୍ ରେଫରେନ୍ସ ବିଭାଗରେ ବିଶେଷ ଆଗ୍ରହୀ ହେବେ ।( WordPress is officially documented in the WordPress Codex at WordPress.org. Developers working with WordPress will be particularly interested in the Developer Codex section and Developer Reference section of wordpress.org)

## Versions

Version	Release Date
1.0	2004-01-03
1.2	2004-05-22
1.5	2005-02-17
2.0	2005-12-26
2.1	2007-01-22
2.2	2007-05-16
2.3	2007-09-24
2.5	2008-03-29
2.6	2008-07-15
2.7	2008-12-10

Version	Release Date
2.8	2009-06-10
2.9	2009-12-18
3.0	2010-06-17
3.1	2011-02-23
3.2	2011-07-04
3.3	2011-12-12
3.4	2012-06-13
3.5	2012-12-11
3.6	2013-08-01
3.7	2013-10-24
3.8	2013-12-12
3.9	2014-04-16
4.0	2014-09-04
4.1	2014-12-17
4.2	2015-04-23
4.3	2015-08-18
4.4	2015-12-08
4.5	2016-04-12
4.6	2016-08-16
4.7	2016-12-06
4.8	2017-06-08

## Examples

### Introduction to WordPress

ଆପଣ, ଖେବସାଇଟ୍ ଏବଂ ବ୍ଲଗ୍ ନିର୍ମାଣ ପାଇଁ ଖର୍ଚ୍ଚପ୍ରେସ୍ [WP] ଏକ ଫୁଲ୍ ଉତ୍ତ ବିଷୟବସ୍ତୁ ପରିଚାଳନା ବ୍ୟବସ୍ଥା | WP PHP ରେ ଲେଖା ହୋଇଛି ଏବଂ MySQL କୁ ଉପଭୋକ୍ତା ବିଷୟବସ୍ତୁ ଏବଂ ବିନ୍ୟାସ ପାଇଁ ତାତା ଷ୍ଟୋର ଭାବରେ ବ୍ୟବହାର କରେ |

ଏଥିରେ ପୁନର୍ଗଠନ ଏବଂ ଥିମର ଏକ ସମୂହ ଇକୋସିଷ୍ଟମ୍ ଅଛି ଏବଂ ଏକ ଜୀବନ୍ତ ଉତ୍ସ ଉତ୍ସ ସମ୍ପ୍ରଦାୟ, ଭଲ ଡକ୍ୟୁମେଣ୍ଟେସନ୍ ଏବଂ ପ୍ରବେଶ ପାଇଁ କମ୍ ପ୍ରତିବନ୍ଧକ ଉପଭୋଗ କରେ | WP କୋଡେକ୍ସରେ ବ୍ୟବହାର ଯୋଗ୍ୟତା ଏବଂ ବିକାଶକାରୀ ଡକ୍ୟୁମେଣ୍ଟେସନ୍ ମିଳିପାରିବ |.

ୱାଉଚପ୍ରେସର ଏକ ଅଂଶ ଯାହା ଏହାକୁ ଅନ୍ୟାନ୍ୟ CMS ଉପାଦାନରୁ ଭିନ୍ନ କରିଥାଏ ଏହାର ଇଭେଣ୍ଟୁ ଡ୍ରାଇଭ ପ୍ରୋଗ୍ରାମିଂ | ପ୍ରୋଗ୍ରାମିଂ ଏବଂ ଲଜିକ୍ ପ୍ରତିନିଧିତ୍ୱ This ର ଏହା ଏକ ଭିନ୍ନ ଉପାୟ ତାପରେ MVC (ମଡେଲ ଡ୍ୟୁ କଣ୍ଟ୍ରୋଲର୍) ସ୍ଥାପତ୍ୟ ଯାହା ଅଧିକାଂଶ CMS ସିଷ୍ଟମ୍ ଦ୍ୱାରା ବ୍ୟବହୃତ ହୁଏ | ୱାଉଚପ୍ରେସ୍ ଆକ୍ସନ୍ ଏବଂ ଫିଲ୍ଡ୍ ର ଧାରଣା ବ୍ୟବହାର କରେ | ସେମାନେ ଇଭେଣ୍ଟୁଗୁଡ଼ିକର ଏକ ଧାଡ଼ି ଗଠନ କରନ୍ତି ଯାହା ପୁନର୍ଗଠନ ଏବଂ ଥିମଗୁଡ଼ିକୁ ଅକ୍ରିମ ଖେପ୍ ଆପ୍ଲିକେସନ୍ ପୃଷ୍ଠା ଏବଂ / କିମ୍ବା ଅଂଶଗୁଡ଼ିକୁ ସମ୍ବିବେଶ, ରୂପାନ୍ତର କିମ୍ବା ଅପସାରଣ କରିବାକୁ ଅନୁମତି ଦିଏ | ସମାନ ଧାରଣା ହେଉଛି JIT କିମ୍ବା ଜଷ୍ଟ-ଇନ୍-ଟାଇମ୍ କମ୍ପାଇଲ୍ |

Histor ତିହାସିକ ଭାବରେ ୱାଉଚପ୍ରେସ୍ ଏକ ବ୍ଲଗିଂ ପ୍ଲାଟଫର୍ମ ଭାବରେ ଜଣାଶୁଣା, ଏବଂ ଏହା କଦାପି ଏହି କଳଙ୍କ ହରାଇପାରେ ନାହିଁ, ମୂଳ ୱାଉଚପ୍ରେସ୍ ଦଳର ଧାନ ସ୍ୱଷ୍ଟ ଭାବରେ ବଦଳି ଯାଇଛି | ପ୍ରତିଷ୍ଠାତା ମାଥ୍ୟୁ ମଲେନଖେଗଙ୍କ ଦ୍ୱ 2016 ାରା 2016 ସ୍ପେଟ୍ ଅଫ୍ ୱାଉଚ ସହିତ ଆମେ ଲକ୍ଷ୍ୟ, ଦୃଷ୍ଟିକୋଣ ଏବଂ ପ୍ରୟାସରେ ଏକ ସ୍ୱଷ୍ଟ ପରିବର୍ତ୍ତନ ଦେଖିପାରିବା | 2016 ରେ, ଆମେ ଆଶ୍ଚର୍ଯ୍ୟଜନକ ଅଗ୍ରଗତି ଦେଖୁଲୁ ଯେତେବେଳେ ୱାଉଚପ୍ରେସ୍ କୋର୍ ବହୁ ଲୋକପ୍ରିୟ REST API ପୁନର୍ଗଠନ ଗ୍ରହଣ କଲା | ଏହା ସ୍ୱଷ୍ଟ ଭାବରେ ମୂଳ ଦଳର ଏକ ଉଦ୍ଦେଶ୍ୟ ଥିଲା ଯେତେବେଳେ ସେମାନେ ଏକ ଫ୍ରଣ୍ଟ-ଏଣ୍ଡ ଜାଭାସ୍କ୍ରିପ୍ଟ CMS ଆଡମିନି ପ୍ୟାନେଲ ନିର୍ମାଣ କରିବାର ଏକ ସାହସିକ ପ୍ରୟାସ ଆରମ୍ଭ କରିଥିଲେ, ଯାହା ଆମେ ଏତେ ବର୍ଷ ଧରି ଦେଖୁଥିବା ସୁବର୍ଣ୍ଣ ମାନକଠାରୁ ଦୂରେଇ ଯାଇଥାଏ; ସେମାନେ ଏହାକୁ କଲପିସୋ ବୋଲି ଡାକିଲେ |

### WordPress Themes

## Mapping URLs to specific templates

To fully grasp WordPress themes, you must understand two primary concepts:

1. Permalinks(ପର୍ମାଲିଙ୍କ୍ )
2. The Template Hierarchy(ଟେମ୍ପଲେଟ୍ ହାଇରାର୍କି )

A permalink is a permanent, non-changing URL (or link, to a specific resource. For instance (ଏକ ପର୍ମାଲିଙ୍କ୍ ହେଉଛି ଏକ ସ୍ଥାୟୀ, ପରିବର୍ତ୍ତନଶୀଳ URL (କିମ୍ବା ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ଉତ୍ସ ସହିତ ଲିଙ୍କ୍ | ଉଦାହରଣ ସ୍ୱରୂପ:)):

- example.com/about-us/ (a Page in WP)
- example.com/services/ (a listing of multiple items, also called an "archive" in WP lingo)
- example.com/services/we-can-do-that-for-you/ (an individual item)

ଯେତେବେଳେ ଜଣେ ଉପଭୋକ୍ତା ଏକ URL ଅନୁରୋଧ କରନ୍ତି, କେଉଁ ଟେମ୍ପଲେଟ୍ ଏହାର ଲେଆଉଟ୍ ନିୟନ୍ତ୍ରଣ କରିବା ଉଚିତ ତାହା ଜାଣିବା ପାଇଁ ୱାଉଚପ୍ରେସ୍ ରିଭର୍ସ-ଇଞ୍ଜିନିୟର୍ସ ପର୍ମାଲିଙ୍କ୍ | ୱାଉଚପ୍ରେସ୍ ବିଭିନ୍ନ ଟେମ୍ପଲେଟ୍ ଫାଇଲଗୁଡ଼ିକ ଖୋଜେ ଯାହା ଏହି ନିର୍ଦ୍ଦିଷ୍ଟ ବିଷୟବସ୍ତୁକୁ ନିୟନ୍ତ୍ରଣ କରିପାରିବ ଏବଂ ଶେଷରେ ଏହା ପାଇଥିବା ନିର୍ଦ୍ଦିଷ୍ଟକୁ ଅଗ୍ରାଧିକାର ଦେଇଥାଏ | ଏହା ଟେମ୍ପଲେଟ୍ ହାଇରାର୍କି ଭାବରେ ଜଣାଶୁଣା |

Once WP finds the matching view template in the hierarchy, it uses that file to process and render the page.( ଥରେ WP କ୍ରମରେ ମେଲ ଖାଉଥିବା ଭୂୟ ଟେମ୍ପଲେଟ୍ ପାଇଲେ, ଯେଉଁ ପ୍ରକ୍ରିୟାକରଣ ଏବଂ ରେଣ୍ଡର୍ କରିବା ପାଇଁ ଏହା ସେହି ଫାଇଲ ବ୍ୟବହାର କରେ )

For example: index.php (the default, "catch-all" template) will be overridden by archive.php (the default template for list-based content), which will in turn be overridden by archive-services.php (a template file specifically for the archive named "services").

[Here is a great visual reference for the Template Hierarchy](#)

# Basic Theme Directory Structure

A simple theme looks something like this:

```
// Theme CSS
style.css

// Custom functionality for your theme
functions.php

// Partials to include in subsequent theme files
header.php
footer.php
sidebar.php
comments.php

// "Archives", (listing views that contain multiple posts)
archive.php
author.php
date.php
taxonomy.php
tag.php
category.php

// Individual content pages
// Note that home and frontpage templates are not recommended
// and they should be replaced by page templates
singular.php
single.php
page.php
front-page.php
home.php

// Misc. Utility Pages
index.php (a catch-all if nothing else matches)
search.php
attachment.php
image.php
404.php
```

---

## Example of a "Single" (template for an individual post)

```
<?php get_header(); ?>

<?php if ( have_posts() ) while ( have_posts() ) : the_post(); ?>
    <h1><?php the_title(); ?></h1>
    <?php the_content(); ?>
    <?php comments_template( '', true ); ?>
<?php endwhile; ?>

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

What's happening here? First, it loads `header.php` (similar to a PHP include or require), sets up The Loop, displays `the_title` and `the_content`, then includes `comments.php`, `sidebar.php`, and `footer.php`. The Loop does the heavy lifting, setting up a `post` object, which contains all the information for the currently-viewed content.

## Example of an "Archive" (template for a list of multiple posts)

```
<?php get_header(); ?>

<?php if ( have_posts() ) while ( have_posts() ) : the_post(); ?>
  <a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
  <?php the_excerpt(); ?>
<?php endwhile; ?>

<?php
  next_posts_link( 'Older Entries', $the_query->max_num_pages );
  previous_posts_link( 'Newer Entries' );
?>

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

First, it includes `header.php`, sets up The Loop, and includes `sidebar.php`, and `footer.php`. But in this case there are multiple posts in the loop, so instead an excerpt is shown with a link to the individual post. `next_posts_link` and `previous_posts_link` are also included so the archive can paginate results.

## Posts, Pages, Custom Post Types, and Custom Fields

ବାକ୍ସ ବାହାରେ, ଖର୍ଚ୍ଚପ୍ରେସ୍ ଦୁଇ ପ୍ରକାରର ବିଷୟବସ୍ତୁକୁ ସମର୍ଥନ କରେ: ପୋଷ୍ଟ ଏବଂ ପୃଷ୍ଠାଗୁଡ଼ିକ । ପୋଷ୍ଟଗୁଡ଼ିକ ସାଧାରଣତଃ blog ବ୍ଲଗ୍ ପୋଷ୍ଟ ପରି ଅଣ-ହାଇରାଙ୍କିକାଲ୍ ବିଷୟବସ୍ତୁ ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ପୃଷ୍ଠାଗୁଡ଼ିକ ଷ୍ଟାଟିକ୍, ଷ୍ଟାଣ୍ଡାଲୋନ୍ ବିଷୟବସ୍ତୁ ପାଇଁ ଆମ ବିଷୟରେ ପୃଷ୍ଠା ପରି, କିମ୍ବା ତଳେ ଥିବା ସର୍ବ-ପେଜ୍ ସହିତ ଏକ କମ୍ପାନୀର ସେବା ପୃଷ୍ଠା ପାଇଁ ବ୍ୟବହୃତ ହୁଏ ।

ସଂସ୍କରଣ 3.0.0 ଅନୁଯାୟୀ, ଡେଭଲପର୍ମାନେ କେବଳ ନିଜସ୍ୱ ମ Word ଲିକିତା ବ୍ୟତୀତ ଖର୍ଚ୍ଚପ୍ରେସର କାର୍ଯ୍ୟକାରୀତାକୁ ବ extend ାଇବା ପାଇଁ ସେମାନଙ୍କର ନିଜସ୍ୱ କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରକୁ ବ୍ୟାଖ୍ୟା କରିପାରିବେ । କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରଗୁଡ଼ିକ ସହିତ, ତୁମେ ତୁମର ଟେମ୍ପଲେଟ୍ ମଧ୍ୟରେ ମୋଡାଡାଟା ଯୋଡ଼ିବା ଏବଂ ଆକ୍ସେସ୍ କରିବାର ଏକ ଗଠନମୂଳକ ଉପାୟ ପ୍ରଦାନ କରିବାକୁ ଅନୁମତି ଦେଇ ତୁମର ପୋଷ୍ଟ / ପୃଷ୍ଠା / କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାର ସହିତ ସଂଲଗ୍ନ କରିବା ପାଇଁ ତୁମର ନିଜସ୍ୱ କଷ୍ଟମ୍ ଫିଲ୍ଡ ମଧ୍ୟ ସୃଷ୍ଟି କରିପାରିବ । ଦେଖନ୍ତୁ: ଉନ୍ନତ କଷ୍ଟମ୍ ଫିଲ୍ଡସ୍ ।

# Chapter 2: Actions and Filters

## Syntax

- `add_action( tag, function_to_call, priority, num_of_args );`
- `add_filter( tag, function_to_call, priority, num_of_args );`

## Parameters

Parameter	Explanation
\$tag	(string) (Required) The name of the action to which the \$function is hooked.
\$function	(callable) (Required) Requires a string containing the function name or anonymous function. See examples for adding functions within classes.
\$priority	(int) default = 10. Functions attached to hooks/ filters will run in the priority assigned. You may have a situation where you want to work with code before any other actions, set priority =1 or after all other attached functions priority = 100 etc. As with all php functions, you can use the function without passing a value for a variable where a default value has been set, but if you wish to change the number of parameters returned, you must specify!
\$parameters	(int) default = 1. The number of parameters returned to your attached function. The parameters returned will depend on the number attached where the hook was created. See <code>apply_filters()</code> and <code>do_action()</code> for more details.

## Remarks

### Wordpress Hooks

Something that often confuses developers when starting to work with WordPress are the use of `apply_filters()` and `add_action()`. You will often see plugins/themes making use of these in code and if you don't understand the concept, you will find it hard to work with them. ( ଖାର୍ଡପ୍ରେମ୍ ସହିତ କାମ କରିବା ଆରମ୍ଭ କରିବାବେଳେ ପ୍ରାୟତଃ develop ଡେଭଲପମେଣ୍ଟକୁ ବ୍ୟବହାର କରିବା କିଛି ହେଉଛି ପ୍ରୟୋଗ ଫିଲ୍ଡର () ଏବଂ ଆଡ୍\_ଆକ୍ସନ୍ () ର ବ୍ୟବହାର । ଆପଣ ପ୍ରାୟତଃ plug ପ୍ଲଗିନ୍ / ଥିମ୍ ଗୁଡ଼ିକ କୋଡ୍ ରେ ବ୍ୟବହାର କରୁଥିବା ଦେଖିବେ ଏବଂ ଯଦି ଆପଣ ଧାରଣାକୁ ବୁଝି ନାହିଁ, ତେବେ ସେମାନଙ୍କ ସହିତ କାମ କରିବା ଆପଣଙ୍କୁ କଷ୍ଟକର ହେବ ।)

In brief (very brief, look up WordPress load flowchart for process in detail), WordPress loads in the following way:

1. wp-load.php - functions etc
2. mu-plugins - any files found in the mu-plugins folder - often used to serve cached objects
3. Plugins - no particular order, any installed and activated plugins will be loaded
4. Active child theme / parent theme
5. init - rest of data

6. template

ଯଦି ଆପଣ ଜଣେ ଡେଭଲପର ଏବଂ ଏକ ଫଙ୍କସନ୍ ଫାଇଲ୍ ସହିତ କାମ କରୁଛନ୍ତି, ଆପଣ ଯେଉଁ ଫାଇଲଗୁଡ଼ିକ ସହିତ କାମ କରୁଛନ୍ତି ତାହା ଅପେକ୍ଷା ଆପଣ ଉଭୟ ପ୍ରକ୍ରିୟାରେ ପୂର୍ବରୁ ଲୋଡ୍ ହୋଇଥିବାର ଦେଖିପାରିବେ । ଅର୍ଥାତ୍ ଆପଣ ପ୍ରକ୍ରିୟାଗୁଡ଼ିକୁ ପରିବର୍ତ୍ତନ କରିପାରିବେ ନାହିଁ (ଧ୍ୟାନ ଦିଅନ୍ତୁ ଯେ ଆପଣ ଫଙ୍କସନ୍ଗୁଡ଼ିକୁ ଓଭର ରାଇଟ୍ କରିପାରିବେ ନାହିଁ) କିମ୍ବା ଭେରିଏବଲ୍ ଯାହା ପରେ ଚାଲିବ କିମ୍ବା ଏପର୍ଯ୍ୟନ୍ତ ବ୍ୟାଖ୍ୟା କରାଯାଇ ନାହିଁ । ଥିମ୍ ଡେଭଲପର୍ମାନେ ପ୍ଲଗଇନ୍ ଗୁଡ଼ିକୁ ହୁକ୍ କରିବାକୁ ଅନୁମତି ଦେବା ପାଇଁ କିମ୍ବା ସେମାନଙ୍କ ପ୍ଲଗଇନ୍ ଅନ୍ୟ ଭେରିଏବଲ୍ଗୁଡ଼ିକୁ ଓଭର ରାଇଟ୍ କରିବାକୁ ଅନୁମତି ଦେଇପାରନ୍ତି । ବର୍ତ୍ତମାନ ଏହା ଏପର୍ଯ୍ୟନ୍ତ ବ୍ଲକ୍ସ୍ ହୋଇପାରେ, କିନ୍ତୁ ସେଠାରେ hang ଠୁଲିକ୍ତ ।

To understand `add_filter()` and `add_action()` we need to look at how the hooks are created in the first place.

```
$arga= 'hello';
do_action('im_a_hook', $arga );
```

When you encounter the above in WordPress, it will call any functions attached to the hook `im_a_hook` (look up `$wp_filter` for information on the process). In your attached function `$arga` will be available for the attached function to work with.

```
add_action('im_a_hook', 'attached_function');

function attached_function($arga){
    echo $arga;
}
```

```
do_action( 'woocommerce_after_add_to_cart_button' );
```

ଲୋଡ୍ ପ୍ରକ୍ରିୟାର କିଛି ବିନ୍ଦୁରେ ଭେରିଏବଲ୍ ପରିବର୍ତ୍ତନ କରିବାକୁ ଏହା ଶକ୍ତିଶାଳୀ ନୂତନ ସୁଯୋଗ ଖୋଲିଥାଏ । ମନେରଖ ଯେ ଆମେ ପୂର୍ବରୁ କହିଥିଲୁ ଯେ ପ୍ଲଗଇନ୍ / ଥିମ୍ ପରେ ଟେମ୍ପଲେଟ୍ ଲୋଡ୍ ହୋଇଛି? ଗୋଟିଏ ସାଧାରଣ ପ୍ଲଗଇନ୍ ହେଉଛି WooCommerce ଯାହା ପରବର୍ତ୍ତୀ ସମୟରେ ସ୍କିନ୍ ସୃଷ୍ଟି କରେ, ମୁଁ ତତ୍ତ୍ୱମେଣ୍ଟ୍ କରିବାକୁ ଯାଉନାହିଁ କିନ୍ତୁ ପ୍ଲଗଇନ୍ ରେ `do_action` ର ଏକ ଉଦାହରଣ କିପରି ମିଳିବ ।

Here we have a hook created that passes no variables back, but we can still have fun with it:

```
add_action( 'woocommerce_after_add_to_cart_button', 'special_offer');

function special_offer(){
    echo '<h1>Special Offer!</h1>';
}
```

The above `add_action` will echo a heading of special offer where `do_action('woocommerce_after_add_to_cart_button')` is located which is when creating a WooCommerce screen. So we can use this hook to insert html. Other uses might include redirecting to a different screen altogether, etc.

```
add_action('custom_hook', 'attached_function', 10, 3);

function attached_function($a,$b,$c){
```

ଫଙ୍କସନ୍ କୁ ଏକାଧିକ ଭେରିଏବଲ୍ ମଧ୍ୟ ପାସ୍ ହୋଇପାରେ । ତୁମର ଥିମ୍ ଫଙ୍କସନ୍ ଗୁଡ଼ିକରେ ଏହାକୁ ଚେଷ୍ଟା କର । ଆମେ 3 କୁ ସେଟିଂ କରୁଥିବା ଶେଷ ପାରାମିଟରକୁ ଧ୍ୟାନ ଦିଅନ୍ତୁ, କାରଣ ଆମେ 3 ଉପଲବ୍ଧ ପାରାମିଟର ସହିତ କାମ କରିବାକୁ ଚାହୁଁ । ଯଦି ଆମେ ଏହାକୁ 2 କୁ ବଦଳାଇଥାଉ, କେବଳ 2 ଫେରସ୍ତ ହେବ ଏବଂ ଆମେ ଏକ ଅଜ୍ଞାତ ତ୍ରୁଟି ପାଇବୁ ।

```

var_dump($a);
var_dump($b);
var_dump($c);

}

$args = 1;
$argb = 2;
$argc = 3;

do_action('custom_hook', $arga, $argb, $argc);
exit;

```

```
$filter_me= apply_filters('im_a_filter', $variable_to_filter);
```

ଫିଲ୍ଟର ନାମକ ଆଉ ଏକ WP ହୁକ୍ ପ୍ରକାର ଅଛି | ଏହାର ବ୍ୟବହାରରେ ଏକ କ୍ରିୟା ଠାରୁ ଏକ ଫିଲ୍ଟର ଭିନ୍ନ, ଏକ କ୍ରିୟା କେବଳ ଭେରିଏବଲ୍ ଗ୍ରହଣ କରିପାରିବ, ନିଶ୍ଚିତ ଭାବରେ ଏହି ଭେରିଏବଲ୍ ଗୁଡ଼ିକ ଫଙ୍କସନ୍ ପରିସର ମଧ୍ୟରେ ଅଛି (ଯଦି ଆପଣ ଗୁଗୁଲ୍ ନୁହଁନ୍ତି php ପରିସର କ'ଣ ତାହା ଜାଣିବା ଉଚିତ୍) | ଫିଲ୍ଟରଗୁଡ଼ିକ ଫେରସ୍ତ ହୋଇଥିବା ତଥ୍ୟକୁ ପାସ୍ କରେ, ତେଣୁ ଆପଣ ଭେରିଏବଲ୍ ପରିବର୍ତ୍ତନ କରିବାକୁ ବ୍ୟବହାର କରିପାରିବେ |

Where you see the above, you can modify the value of `$filter_me` as any data you return will be the value stored in the variable. So for example (note we are changing `$variable_to_filter` to `$filter_me` in the example):

```

add_filter('im_a_filter', 'attached_function', 100);

function attached_function($filter_me){

    $filter_me= 'ray';

    return $filter_me;

}

$filter_me = 'bob';
$filter_me= apply_filters('im_a_filter', $filter_me);

```

The `$filter_me` variable will now contain `'ray'` rather than `'bob'`, we have set a priority of 100 so we are reasonably confident no one is changing the value after use (there can be multiple filters running on the same hook) So we can now change variables used later in the process if `apply_filters()` is present.

```

add_filter('im_a_filter', 'attached_function', 100, 3);

function attached_function($filter_me, $arga, $argb){

    $filter_me= 'ray'.$arga.$argb;

```

ଆପଣ ଏକାଧିକ ପାରାମିଟର ମଧ୍ୟ ପାସ୍ କରିପାରିବେ, କିନ୍ତୁ ଆପଣ କେବଳ ଗୋଟିଏର ମୂଲ୍ୟ ପରିବର୍ତ୍ତନ କରିପାରିବେ | ଆପଣ ନିଶ୍ଚିତ ଭାବରେ ଏକ ମୂଲ୍ୟ ଫେରସ୍ତ କରିବେ, ନଚେତ୍ ଆପଣଙ୍କର ଭେରିଏବଲ୍ କିଛି ଧାରଣା କରିବ ନାହିଁ | ଭେରିଏବଲ୍ ଗୁଡ଼ିକରେ ଭାଲ୍ୟୁ / ଆରେ / ବସ୍ତୁ ନ୍ୟସ୍ତ କରିବାକୁ ଆପଣ କିପରି php ବ୍ୟବହାର କରନ୍ତି ତାହା ଯଦି ଆପଣ ବୁଝିପାରନ୍ତି ତେବେ ଏହା ଆପଣଙ୍କ ପାଇଁ ସ୍ପଷ୍ଟ ହେବ, ଯଥା:

```

    $arga= 'you fool';

    return $filter_me;
}

$filter_me = 'bob';

$arga = ' middlename';
$argb = ' surname';

$filter_me= apply_filters('im_a_filter', $filter_me, $arga, $argb);

```

The `$filter_me` variable now contains *'ray middlename surname'*. But what about `$arga`? This still contains *'middlename'*, changing an `$arga` to *'you fool'* within our function has no effect on the defined value outside of its scope (there are ways, google globals etc.)

**`add_action($hook_name, $function, $priority, $parameters)`**

**`add_filter($hook_name, $function, $priority, $parameters);`**

## Examples

### add\_action - init

```

add_action('init', 'process_post');

function process_post(){
    if($_POST)
        var_dump($_POST);
}

```

### add\_action - init - anonymous function

```

add_action('init' , function(){
    echo 'i did something';
});

```

### add\_action - init - within class object

```

class sample{

    public function __construct(){
        add_action('init', array($this, 'samp') );
    }

    public function samp(){ // must be public!!
        echo 'i did something';
    }
}

new sample();

```

## add\_action - init - within static class

```
class sample{

    public static function add_action_func(){
        //note __CLASS__ will also include any namespacing
        add_action('init', array(__CLASS__, 'samp') );
    }

    public static function samp(){
        echo 'i did something';
    }

}

sample::add_action_func();
```

# Chapter 3: Add Shortcode

## Syntax

- `add_shortcode( $tag , $func );`

## Parameters

Parameter	Details
\$tag	(string) (required) Shortcode tag to be searched in post content
\$func	(callable) (required) Hook to run when shortcode is found

## Remarks

- ସର୍ଟକୋଡ୍ କଲବ୍ୟାକ୍ ଡିନୋଟି ଆର୍ଗୁମେଣ୍ଟ୍ ପାସ୍ ହେବ: ସର୍ଟକୋଡ୍ ଆକ୍ଟିଭ୍‌ସ୍, ସର୍ଟକୋଡ୍ ବିଷୟବସ୍ତୁ (ଯଦି ଥାଏ), ଏବଂ ସର୍ଟକୋଡ୍ ନାମ ।
- ପ୍ରତ୍ୟେକ ସର୍ଟକୋଡ୍ ପାଇଁ କେବଳ ଗୋଟିଏ ହୁକ୍ ହୋଇପାରେ । ଯାହାର ଅର୍ଥ ହେଉଛି ଯଦି ଅନ୍ୟ ଏକ ପ୍ଲଗିନ୍ ସମାନ ସର୍ଟକୋଡ୍ ଥାଏ, ତେବେ ଏହା ଆପଣଙ୍କର ଓଭରରାଇଡ୍ କରିବ କିମ୍ବା ପ୍ଲଗିନ୍ ଗୁଡ଼ିକ କେଉଁ କ୍ରମରେ ଅନ୍ତର୍ଭୁକ୍ତ ହୋଇଛି ଏବଂ / କିମ୍ବା ଚାଲୁଛି ତାହା ଉପରେ ନିର୍ଭର କରି ଆପଣଙ୍କର ସେଗୁଡ଼ିକ ନବଲିଖନ କରିବ ।
- ସର୍ଟକୋଡ୍ ଆକ୍ଟିଭ୍‌ସ୍ ନାମଗୁଡ଼ିକ ହ୍ୟାଣ୍ଡଲର୍ ଫଙ୍କସନ୍ସ ପାସ୍ ହେବା ପୂର୍ବରୁ ସର୍ବଦା ଲୋୟର କେସରେ ରୁପାନ୍ତରିତ ହୋଇଥାଏ । ମୂଲ୍ୟଗୁଡ଼ିକ ଅସ୍ପଷ୍ଟ ।
- ଧ୍ୟାନ ଦିଅନ୍ତୁ ଯେ ସର୍ଟକୋଡ୍ ଦ୍ୱାରା କୁହାଯାଇଥିବା କାର୍ଯ୍ୟ କ `never` ଶସି ପ୍ରକାରର ଆଉଟପୁଟ୍ ଉତ୍ପାଦନ କରିବା ଉଚିତ୍ ନୁହେଁ । ସର୍ଟକୋଡ୍ ଫଙ୍କସନ୍ ଗୁଡ଼ିକ ଶର୍ଟକୋଡ୍ ବଦଳାଇବା ପାଇଁ ବ୍ୟବହୃତ ହେବାକୁ ଥିବା ପାଠ୍ୟକୁ ଫେରାଇବା ଉଚିତ୍ । ସିଧାସଳଖ ଆଉଟପୁଟ୍ ଉତ୍ପାଦନ କରିବା ଅପ୍ରତ୍ୟାଶିତ ଫଳାଫଳକୁ ନେଇଯିବ । ଏହା ଫିଲ୍ଡର ଫଙ୍କସନ୍‌ଗୁଡ଼ିକର ଆଚରଣ ସହିତ ସମାନ, ଯେହେତୁ ସେମାନେ କଲ୍ ଠାରୁ ଆଶା କରାଯାଉଥିବା ପାର୍ଶ୍ୱ ପ୍ରତିକ୍ରିୟା ଉତ୍ପାଦନ କରିବା ଉଚିତ୍ ନୁହେଁ, ଯେହେତୁ ସେମାନେ କେବେ ଏବଂ କେଉଁଠାରୁ ଡାକନ୍ତି ଆପଣ ନିୟନ୍ତ୍ରଣ କରିପାରିବେ ନାହିଁ ।

## Examples

### Simple shortcode for recent post

`add_shortcode` is wp keyword.

```
// recent-posts is going to be our shortcode.
add_shortcode('recent-posts', 'recent_posts_function');

// This function is taking action when recent post shortcode is called.
function recent_posts_function() {
    query_posts(array('orderby' => 'date', 'order' => 'DESC' , 'showposts' => 1));
    if (have_posts()) :
        while (have_posts()) : the_post();
            $return_string = '<a href="'.get_permalink().'">'.get_the_title().'</a>';
        endwhile;
```

```

endif;
wp_reset_query();
return $return_string;
}

```

This snippet can be placed in your theme `functions.php`.

[recent-posts] This is our shortcode for recent post. We can apply this shortcode in backend (such as pages, post, widgets ).

We can also use the same shortcode inside our code. with the help of `do_shortcode`.

**Eg.** `echo do_shortcode( '[recent-posts]' );`

## Advanced shortcode for recent posts

ଆପଣ କେତେ ସାମ୍ପ୍ରତିକ ପୋଷ୍ଟଗୁଡ଼ିକ ପ୍ରଦର୍ଶନ କରିବାକୁ ଚାହୁଁଛନ୍ତି, ଏହି କାର୍ଯ୍ୟଗୁଡ଼ିକ ପାରାମିଟର ନେଇଥାଏ ।

Ex : you want to display only five recent posts. Just passed the arguments with `posts="5"` (you can pass any number of recent posts that you want to display).

Function fetch only five recent posts from database.

```

// recent-posts is going to be our shortcode.
add_shortcode('recent-posts', 'recent_posts_function');

// Functions takes parameter such as posts="5".
function recent_posts_function($atts){
    extract(shortcode_atts(array(
        'posts' => 1,
    ), $atts));

    $return_string = '<ul>';
    query_posts(array('orderby' => 'date', 'order' => 'DESC' , 'showposts' => $posts));
    if (have_posts()) :
        while (have_posts()) : the_post();
            $return_string .= '<li><a href="'.get_permalink().'">'.get_the_title().</a></li>';
        endwhile;
    endif;
    $return_string .= '</ul>';

    wp_reset_query();
    return $return_string;
}

```

**Eg.** `echo do_shortcode( '[recent-posts posts="5"]' );`

---

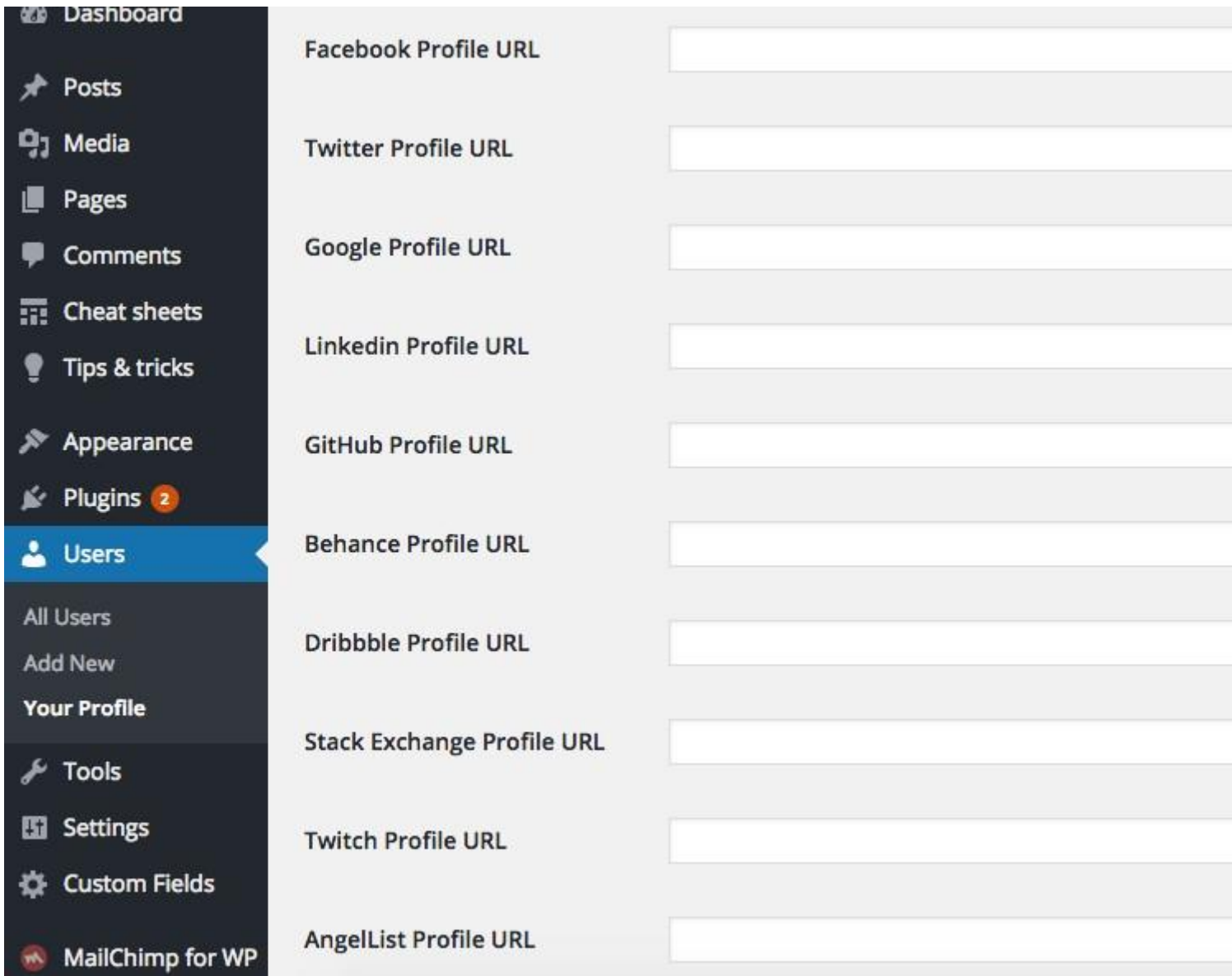
## Chapter 4: Add/remove contact info for users with `user_contactmethods` filter hook

### Examples

#### Enabling most popular social networks

```
function social_profiles( $contactmethods ) {  
  
    $contactmethods['facebook_profile'] = 'Facebook Profile URL';  
    $contactmethods['twitter_profile'] = 'Twitter Profile URL';  
    $contactmethods['google_profile'] = 'Google Profile URL';  
    $contactmethods['linkedin_profile'] = 'Linkedin Profile URL';  
    $contactmethods['github_profile'] = 'GitHub Profile URL';  
    $contactmethods['behance_profile'] = 'Behance Profile URL';  
    $contactmethods['dribbble_profile'] = 'Dribbble Profile URL';  
    $contactmethods['stack_profile'] = 'Stack Exchange Profile URL';  
    $contactmethods['twitch_profile'] = 'Twitch Profile URL';  
    $contactmethods['angellist_profile'] = 'AngelList Profile URL';  
  
    return $contactmethods;  
}  
  
add_filter( 'user_contactmethods', 'social_profiles', 10, 1);
```

You will get this fields in your dashboard:



And this is how you retrieve it in code

```
<?php $user_stack_exchange = get_the_author_meta( 'stack_profile' ); ?>
```

## Removing contact method

```
function remove_contact_methods( $contactmethods ) {  
  
    unset($contactmethods['facebook_profile']);  
    unset($contactmethods['twitter_profile']);  
  
    return $contactmethods;  
}  
  
add_filter( 'user_contactmethods', 'remove_contact_methods', 10, 1);
```

---

# Chapter 5: add\_action()

## Syntax

- `add_action( $tag, $function_to_add )`
- `add_action( $tag, $function_to_add, $priority )`
- `add_action( $tag, $function_to_add, $priority, $accepted_args )`

## Parameters

Parameter	Details
<code>\$tag</code>	<i>(string)</i> The name of the action to which the procedure <code>\$function_to_add</code> will be hooked.
<code>\$function_to_add</code>	<i>(callable)</i> The callable function/procedure you want to be called.
<code>\$priority</code>	<i>(int)</i> The priority level to which the <code>\$function_to_add</code> will be executed. Optional. Default 10.
<code>\$accepted_args</code>	<i>(int)</i> The number of arguments that the callable function <code>\$function_to_add</code> accepts. Optional. Default 1.

## Examples

### Direct function callback

```
add_action( 'init', function() {  
    // do something here  
} );
```

Using a function block to hook a set of instructions. With the `init` hook, the set of instructions will be executed right after wordpress has finished loading the necessary components.

### Function name reference callback

```
function my_init_function() {  
    // do something here  
}  
  
add_action( 'init', 'my_init_function' );
```

Using the name of the function to hook a set of instructions. With the `init` hook, the set of instructions will be executed right after wordpress has finished loading the necessary components.

## Class static method callback

```
class MyClass {
    static function my_init_method() {
        // do something here
    }
}

add_action( 'init', array( 'MyClass', 'my_init_method' ) );
```

Using a static method of a class to hook a set of instructions. With the `init` hook, the set of instructions will be executed right after wordpress has finished loading the necessary components.

## Object method callback

```
class MyClass {
    function my_init_method() {
        // do something here
    }
}

$obj = new MyClass();

add_action( 'init', array( $obj, 'my_init_method' ) );
```

Using a method of an object to hook a set of instructions. With the `init` hook, the set of instructions will be executed right after wordpress has finished loading the necessary components.

---

# Chapter 6: add\_editor\_style()

## Introduction

The function allows user to load stylesheets for the TinyMCE editor

## Syntax

- add\_editor\_style( \$stylesheet )

## Parameters

Parameter	Details
\$stylesheet	(array or string) (Optional) Stylesheet name or array thereof, relative to theme root. Defaults to 'editor-style.css'

## Examples

### Loading a single css file

#### Code

```
function add_new_style() {  
    add_editor_style( 'file-name-here.css' );  
}  
add_action( 'admin_init', 'add_new_style' );
```

#### Explanation

In the above code, we used add\_editor\_style to load the css file. We also used add\_action to make sure that WordPress runs our function. ( ଉପରୋକ୍ତ କୋଡ଼ ରେ, css ଫାଇଲ୍ ଲୋଡ଼ କରିବାକୁ ଆମେ add\_editor\_style ବ୍ୟବହାର କଲୁ । ଏଡ଼ିସନ୍ତରା ଆମର ଫଙ୍କସନ୍ ଚଳାଇଛି କି ନାହିଁ ନିଶ୍ଚିତ କରିବାକୁ ଆମେ add\_action ମଧ୍ୟ ବ୍ୟବହାର କରିଥିଲୁ । )

---

# Chapter 7: add\_menu\_page()

## Introduction

This function is to add an item in the admin panel's nav bar.

## Syntax

- `add_menu_page( $page_title, $menu_title, $capability, $menu_slug, $function, $icon_url, $position)`

## Parameters

Parameter	Details
<code>\$page_title</code>	(string) The text to be displayed in the title tags of the page when the menu is selected.
<code>\$menu_title</code>	(string) The text to be used for the menu.
<code>\$capability</code>	(string) The capability required for this menu to be displayed to the user.
<code>\$menu_slug</code>	(string) The slug name to refer to this menu by (should be unique for this menu).
<code>\$function</code>	(callable) (optional) The function to be called to output the content for this page.
<code>\$icon_url</code>	(string) (optional) The URL to the icon to be used for this menu.
<code>\$position</code>	(int) (optional) The position in the menu order this one should appear.

## Remarks

Here is a list of the default positions (for `$position`)

- 2 – Dashboard
- 4 – Separator
- 5 – Posts
- 10 – Media
- 15 – Links
- 20 – Pages
- 25 – Comments
- 59 – Separator

- 60 – Appearance
- 65 – Plugins
- 70 – Users
- 75 – Tools
- 80 – Settings
- 99 – Separator

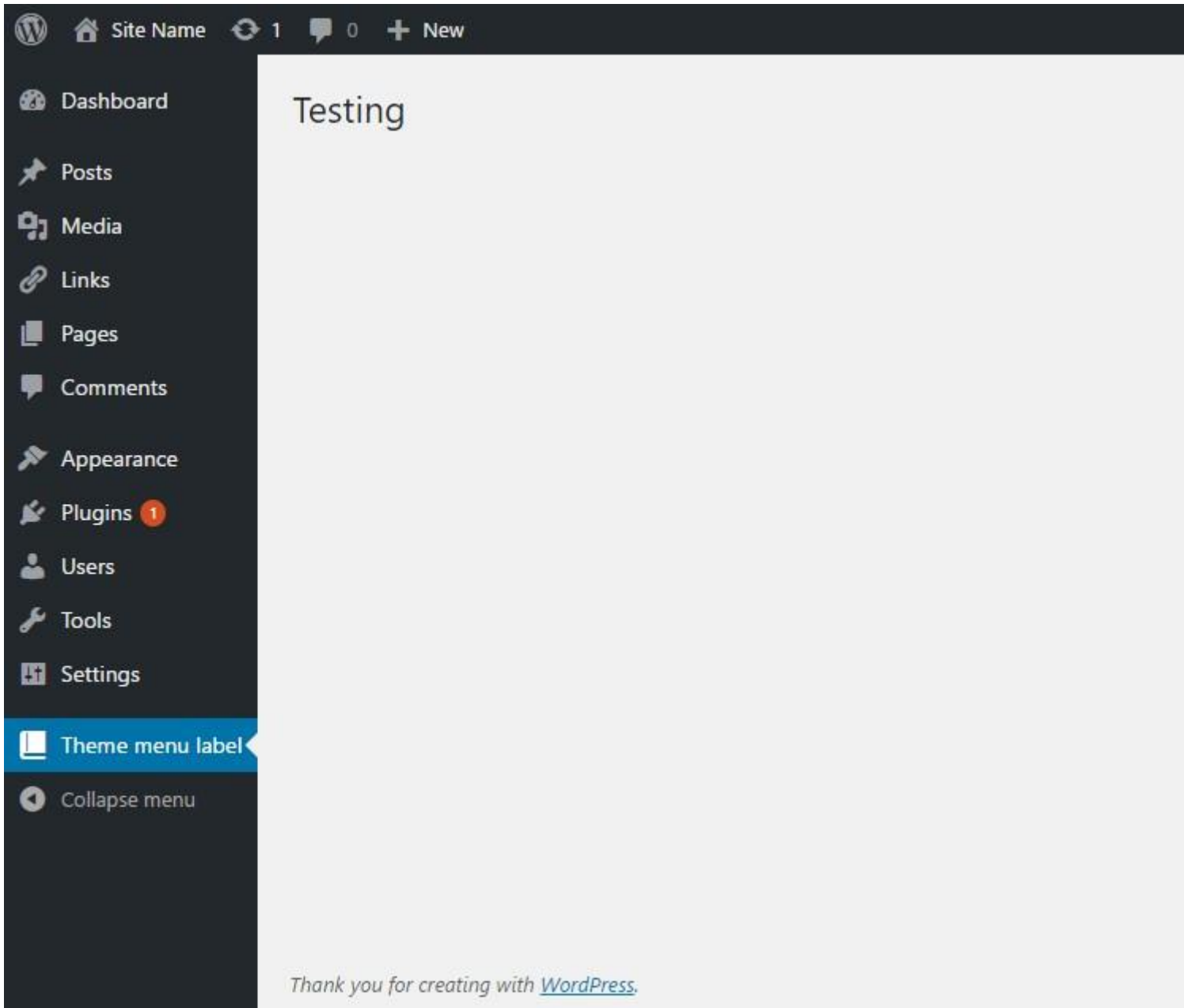
## Examples

### Adding the "Theme page title" item to the nav bar

#### Code

```
function add_the_theme_page(){
    add_menu_page('Theme page title', 'Theme menu label', 'manage_options', 'theme-options',
'page_content', 'dashicons-book-alt');
}
add_action('admin_menu', 'add_the_theme_page');
function page_content(){
    echo '<div class="wrap"><h2>Testing</h2></div>';
}
```

#### Output



## Explanation

In the code, we created a function named `add_the_theme_page` and we used `add_menu_page` to add the item to the navbar. Please check the parameters part in this page to know about the arguments we passed in. Then we used `add_action` to run our `add_the_theme_page` function. Finally, we created the function `page_content` to display contents in the page. (କୋଡ଼ ରେ, ଆମେ `add_the_theme_page` ନାମକ ଏକ ଫଙ୍କସନ୍ ତିଆରି କଲୁ ଏବଂ ନାଭରରେ ରେ ଆଇଟମ୍ ଯୋଡ଼ିବା ପାଇଁ ଆମେ `add_menu_page` ବ୍ୟବହାର କଲୁ । ଆମେ ପାସ୍ କରିଥିବା ଆର୍ଗୁମେଣ୍ଟସ୍ ବିଷୟରେ ଜାଣିବା ପାଇଁ ଦୟାକରି ଏହି ପୃଷ୍ଠାର ପାରାମିଟର ଅଂଶ ଯାଞ୍ଚ କରନ୍ତୁ । ତା'ପରେ ଆମେ ଆମର `add_the_theme_page` ଫଙ୍କସନ୍ ଚଲାାଇବା ପାଇଁ `add_action` ବ୍ୟବହାର କଲୁ । ଶେଷରେ, ପୃଷ୍ଠାରେ ବିଷୟବସ୍ତୁ ପ୍ରଦର୍ଶନ କରିବାକୁ ଆମେ `page_content` ଫଙ୍କସନ୍ ସୃଷ୍ଟି କଲୁ ।)

## OOP & how to load scripts/styles on menu page

```
<?php
/*
 * Plugin Name: Custom Admin Menu
 */

class SO_WP_Menu {

    private $plugin_url;
```

```

public function __construct() {
    $this->plugin_url = plugins_url( '/', __FILE__ );
    add_action( 'plugins_loaded', array( $this, 'init' ) );
}

public function init() {
    add_action( 'admin_menu', array( $this, 'add_menu' ) );
}

public function add_menu() {
    $hook = add_menu_page(
        'My Menu', // Title, html meta tag
        '<span style="color:#e57300;">My Menu</span>', // Menu title, hardcoded style
        'edit_pages', // capability
        'dummy-page-slug', // URL
        array( $this, 'content' ), // output
        null, // icon, uses default
        1 // position, showing on top of all others
    );
    add_action( "admin_print_scripts-$hook", array( $this, 'scripts' ) );
    add_action( "admin_print_styles-$hook", array( $this, 'styles' ) );
}

public function content() {
    ?>
    <div id="icon-post" class="icon32"></div>
    <h2>Dummy Page</h2>
    <p> Lorem ipsum</p>
    <?php
}

# Printing directly, could be wp_enqueue_script
public function scripts() {
    ?><script>alert('My page');</script><?php
}

# Enqueing from a CSS file on plugin directory
public function styles() {
    wp_enqueue_style( 'my-menu', $this->plugin_url . 'my-menu.css' );
}
}

new SO_WP_Menu();

```

What's important to note in this example is that, when using `add_menu_page()`, it returns a hook that can be used to target our exact page and load Styles and Scripts there.

A common mistake is to enqueue without targeting and that spills scripts and styles all over `/wp-admin`.

Using OOP we can store common variables to be used among internal methods.

(ଏହି ଉଦାହରଣରେ ଧ୍ୟାନ ଦେବା ଜରୁରୀ ହେଉଛି ଯେ, `add_menu_page()` ବ୍ୟବହାର କରିବାବେଳେ, ଏହା ଏକ ହୁକ୍ ରିଟର୍ନ କରେ ଯାହା ଆମର ସଠିକ୍ ପୃଷ୍ଠାକୁ ଟାର୍ଗେଟ୍ କରିବା ପାଇଁ ଏବଂ ସେଠାରେ ସ୍ଟାଇଲ୍ ଏବଂ ସ୍କ୍ରିପ୍ଟ୍ ଲୋଡ୍ କରିବା ପାଇଁ ବ୍ୟବହୃତ ହୋଇପାରେ । ଏକ ସାଧାରଣ ତ୍ରୁଟି ହେଉଛି ଟାର୍ଗେଟ୍ ନକରି ଏନକ୍ୟୁଏ କରିବା ଏବଂ ଏହା `/wp-` ଆଡମିନିରେ ସ୍କ୍ରିପ୍ଟ୍ ଏବଂ ସ୍ଟାଇଲ୍ ଛିଞ୍ଚିଥାଏ । OOP ବ୍ୟବହାର କରି ଆମେ ଆଭ୍ୟନ୍ତରୀଣ ପଦ୍ଧତି ମଧ୍ୟରେ ବ୍ୟବହୃତ ହେବାକୁ ଥିବା ସାଧାରଣ ଭେରିଏବଲ୍ ଗଠିତ କରିପାରିବା ।)

---

# Chapter 8: add\_submenu\_page()

## Introduction

This function is to add a sub-item to an existing item in the admin panels nav bar.

## Syntax

- add\_submenu\_page( \$parent\_slug, \$page\_title, \$menu\_title, \$capability, \$menu\_slug, \$function )

## Parameters

Parameter	Details
\$parent_slug	(string) The slug name for the parent menu (or the file name of a standard WordPress admin page).
\$page_title	(string) The text to be displayed in the title tags of the page when the menu is selected.
\$menu_title	(string) The text to be used for the menu.
\$capability	(string) The capability required for this menu to be displayed to the user.
\$menu_slug	(string) The slug name to refer to this menu by (should be unique for this menu).
\$function	(callable) (Optional) The function to be called to output the content for this page.

## Remarks

Here are a list of slugs for \$parent\_slug

- Dashboard: 'index.php'
- Posts: 'edit.php'
- Media: 'upload.php'
- Pages: 'edit.php?post\_type=page'
- Comments: 'edit-comments.php'
- Custom Post Types: 'edit.php?post\_type=your\_post\_type'
- Appearance: 'themes.php'
- Plugins: 'plugins.php'
- Users: 'users.php'

- Tools: 'tools.php'
- Settings: 'options-general.php'
- Network Settings: 'settings.php'

## Examples

Adding the "Submenu Page" as a sub-page of "Tools" to the nav bar

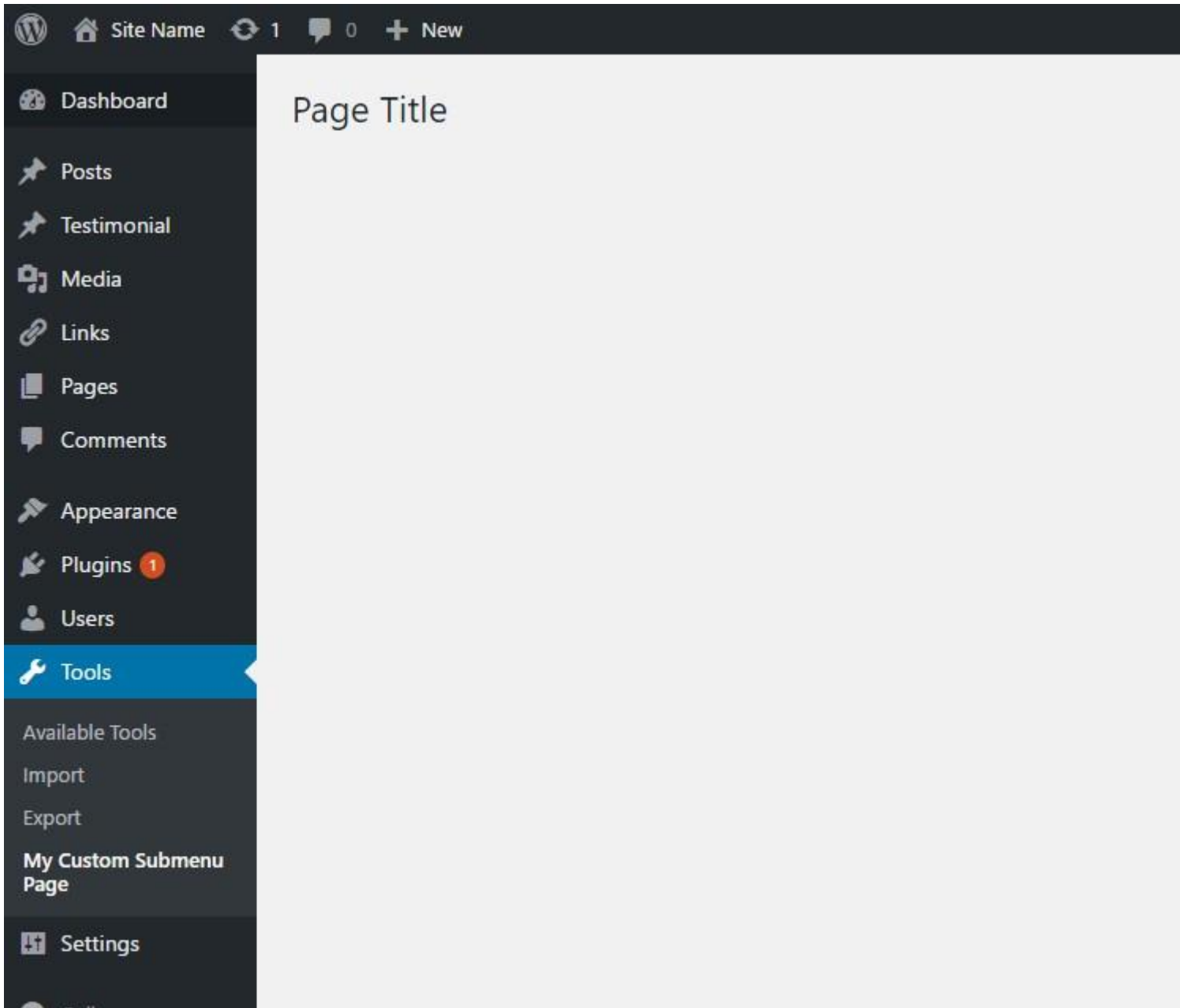
### Code

```
add_action('admin_menu', 'register_my_custom_submenu_page');

function register_my_custom_submenu_page() {
    add_submenu_page(
        'tools.php',
        'Submenu Page',
        'My Custom Submenu Page',
        'manage_options',
        'my-custom-submenu-page',
        'my_custom_submenu_page_content' );
}

function my_custom_submenu_page_content() {
    echo '<div class="wrap">';
    echo '<h2>Page Title</h2>';
    echo '</div>';
}
```

### Output



## Explanation

In the code, we created a function named `register_my_custom_submenu_page` and we used `add_submenu_page` to add the item to the navbar as a child of `tools.php`, which is the Tools page.

Please check the parameters part in this page to know about the arguments we passed in. Then we used `add_action` to run our `register_my_custom_submenu_page` function. Finally, we created the function `my_custom_submenu_page_content` to display contents in the page.

(କୋଡ୍ ରେ, ଆମେ ରେଜିଷ୍ଟର `my_custom_submenu_page` ନାମକ ଏକ ଫଙ୍କସନ୍ ତିଆରି କଲୁ ଏବଂ ଆମେ ବ୍ୟବହାର କଲୁ | `add_submenu_page` ଫଙ୍କସନ୍ କୁ `tools.php` ର ପିଲା ଭାବରେ ନାଭରରେ ଆଇଟମ୍ ଯୋଡ଼ିବାକୁ, ଯାହା ଉପକରଣ ପୃଷ୍ଠା |

ଆମେ ପାସ୍ କରିଥିବା ଆର୍ଗୁମେଣ୍ଟସ୍ ବିଷୟରେ ଜାଣିବା ପାଇଁ ଦୟାକରି ଏହି ପୃଷ୍ଠାର ପାରାମିଟର ଅଂଶ ଯାଞ୍ଚ କରନ୍ତୁ | ତା'ପରେ ଆମେ ଆମ ରେଜିଷ୍ଟର `my_custom_submenu_page_content` ଫଙ୍କସନ୍ ଚଳାଇବା ପାଇଁ `add_action` ବ୍ୟବହାର କଲୁ | ଶେଷରେ, ପୃଷ୍ଠାରେ ବିଷୟବସ୍ତୁ ପ୍ରଦର୍ଶନ କରିବାକୁ ଆମେ `my_custom_submenu_page_content` ଫଙ୍କସନ୍ ସୃଷ୍ଟି କଲୁ |)

---

# Chapter 9: add\_theme\_support()

## Introduction

This function registers features that the theme supports.

## Syntax

- `add_theme_support( $feature )`

## Parameters

Parameter	Details
<code>\$feature</code>	(string) The feature being added.

## Remarks

List of features to be used in `$feature`:

- 'post-formats'
- 'post-thumbnails'
- 'html5'
- 'custom-logo'
- 'custom-header-uploads'
- 'custom-header'
- 'custom-background'
- 'title-tag'
- 'starter-content'

## Examples

### Adding theme support for post formats

```
add_theme_support( 'post-formats', array( 'formatone', 'formattwo' ) );
```

### Adding theme support for post thumbnails to posts

```
add_theme_support( 'post-thumbnails', array( 'post' ) );
```

The above code only allows post thumbnails on all posts. To enable the feature on all post types, do:

```
add_theme_support( 'post-thumbnails' );
```

# Chapter 10: Admin Dashboard Widgets

## Introduction

(With an admin dashboard widget you are able to display any kind of information at the admin dashboard. You can make multiple widgets if you want. You can add the code to the functions.php of your theme or to your plugin. )ଏକ ଆଡମିନ୍ ଡ୍ୟାସବୋର୍ଡ ଓଜେଟ୍ ସହିତ ଆପଣ ଆଡମିନ୍ ଡ୍ୟାସବୋର୍ଡରେ ଯେକିନ୍ଦ ଶସି ପ୍ରକାରର ସୂଚନା ପ୍ରଦର୍ଶନ କରିବାକୁ ସକ୍ଷମ ଅଟନ୍ତି । ଆପଣ ଚାହଁଲେ ଏକାଧିକ ଓଜେଟ୍ ତିଆରି କରିପାରିବେ । ତୁମେ ତୁମର ଥିମ୍ ର ଫଙ୍କସନ୍ସ୍ । Php କିମ୍ବା ତୁମର ପ୍ଲଗିନ୍ ରେ କୋଡ୍ ଯୋଡ଼ି ପାରିବ ।

## Syntax

- `add_action($tag, $function_to_add, $priority, $accepted_args);`
- `wp_add_dashboard_widget($widget_id, $widget_name, $callback, $control_callback, $callback_args);`

## Parameters

Parameter	Details
<code>\$tag</code>	<i>(string required)</i> Name of the action where <code>\$function_to_add</code> is hooked
<code>\$function_to_add</code>	<i>(callable required)</i> Name of the function you want to call.
<code>\$priority</code>	<i>(int optional)</i> Place of the function call in all functions of action (default = 10)
<code>\$accepted_args</code>	<i>(int optional)</i> Number of parameters the function accepts (default = 1)
<code>\$widget_id</code>	<i>(string required)</i> Unique slug for your widget
<code>\$widget_name</code>	<i>(string required)</i> Name of your widget (displayed in the head)
<code>\$callback</code>	<i>(callable required)</i> Name of the function which displays the content of your widget
<code>\$control_callback</code>	<i>(callable optional)</i> Name of the function which handles the widget options forms
<code>\$callback_args</code>	<i>(array optional)</i> Parameters of the <code>\$control_callback</code> function

## Examples

### Simple widget (displays text)

This will add a simple widget which displays just a small message.  
(ଏହା ଏକ ସରଳ ୱିଜେଟ୍ ଯୋଗ କରିବ ଯାହା କେବଳ ଏକ ଛୋଟ ବାର୍ତ୍ତା ପ୍ରଦର୍ଶନ କରେ ।)

```
add_action('wp_dashboard_setup', 'register_my_dashboard_widgets');

function register_my_dashboard_widgets() {
    wp_add_dashboard_widget('myInfo_widget', 'Important Information', 'display_infowidget');
}

function display_infowidget() {
    echo '<p>At the first of february this site gets a new design.
    Therefore is wont be available this day. To see the current progress you can visit
    <a href="http://www.justanexample.com" >this site</a></p>';
}
```

---

# Chapter 11: AJAX

## Examples

### AJAX request with a JSON response

#### functions.php:

```
// We add the action twice, once for logged in users and once for non logged in users.
add_action( 'wp_ajax_my_action', 'my_action_callback' );
add_action( 'wp_ajax_nopriv_my_action', 'my_action_callback' );

// Enqueue the script on the front end.
add_action( 'wp_enqueue_scripts', 'enqueue_my_action_script' );
// Enqueue the script on the back end (wp-admin)
add_action( 'admin_enqueue_scripts', 'enqueue_my_action_script' );

function my_action_callback() {
    $json = array();

    if ( isset( $_REQUEST['field2'] ) ) {
        $json['message'] = 'Success!';
        wp_send_json_success( $json );
    } else {
        $json['message'] = 'Field 2 was not set!';
        wp_send_json_error( $json );
    }
}

function enqueue_my_action_script() {
    wp_enqueue_script( 'my-action-script', 'path/to/my-action-script.js', array( 'jquery' ),
    null, true );
    wp_localize_script( 'my-action-script', 'my_action_data', array(
        'ajaxurl' => admin_url( 'admin-ajax.php' ),
    ) );
}
```

#### my-action-script.js:

```
(function( $ ) {
    'use strict';

    $( document ).on( 'ready', function() {
        var data = {
            action: 'my_action',
            field2: 'Hello world',
            field3: 3
        };

        $.getJSON( my_action_data.ajaxurl, data, function( json ) {
            if ( json.success ) {
                alert( 'yes!' );
            } else {
                alert( json.data.message );
            }
        }
    )
}
```

```

    } );
  } );
})( jQuery );

```

## AJAX with .ajax() and WordPress Nonce

### functions.php

```

//Localize the AJAX URL and Nonce
add_action('wp_enqueue_scripts', 'example_localize_ajax');
function example_localize_ajax(){
    wp_localize_script('jquery', 'ajax', array(
        'url' => admin_url('admin-ajax.php'),
        'nonce' => wp_create_nonce('example_ajax_nonce'),
    ));
}

//Example AJAX Function
add_action('wp_ajax_example_function', 'example_function');
add_action('wp_ajax_nopriv_example_function', 'example_function');
function example_function(){
    if ( !wp_verify_nonce($_POST['nonce'], 'example_ajax_nonce') ){
        die('Permission Denied.');
```

```

    }

    $firstname = sanitize_text_field($_POST['data']['firstname']);
    $lastname = sanitize_text_field($_POST['data']['lastname']);

```

```

    //Do something with data here
    echo $firstname . ' ' . $lastname; //Echo for response
    wp_die(); // this is required to terminate immediately and return a proper response:-
https://codex.wordpress.org/AJAX\_in\_Plugins
}

```

### example.js

```

jQuery(document).on('click touch tap', '.example-selector', function(){
    jQuery.ajax({
        type: "POST",
        url: ajax.url,
        data: {
            nonce: ajax.nonce,
            action: 'example_function',
            data: {
                firstname: 'John',
                lastname: 'Doe'
            },
        },
        success: function(response){
            //Success
        },
        error: function(XMLHttpRequest, textStatus, errorThrown){
            //Error
        },
        timeout: 60000
    });
});

```

```
    return false;
});
```

## wp\_ajax - core functionality + \_wpnonce check

### functions.php:

```
function rm_init_js() {
    wp_enqueue_script( 'custom-ajax-script', get_template_directory_uri() . '/js/ajax.js',
array( 'jquery', 'wp-util' ), '1.0', true );
    // pass custom variables to JS
    wp_localize_script( 'custom-ajax-script', 'BEJS', array(
        'action' => 'custom_action',
        'nonce' => wp_create_nonce( 'test-nonce' )
    ) );
}

add_action( 'wp_enqueue_scripts', 'rm_init_js' );

function rm_ajax_handler() {
    check_ajax_referer( 'test-nonce' );

    extract( $_POST );
    $data = compact( 'first_name', 'last_name', 'email' );

    foreach ( $data as $name => $value ) {
        switch ( $name ) {
            case 'first_name':
            case 'last_name':
                $data[ $name ] = ucfirst( sanitize_user( $value ) );
                break;
            case 'email':
                $data[ $name ] = sanitize_email( $value );
                break;
        }
    }

    $userID = email_exists( $data['email'] );

    if ( ! $userID ) {
        wp_send_json_error( sprintf(__( 'Something went wrong! %s try again!', 'textdomain'
), $data['first_name'] . ' ' . $data['last_name'] ) );
    }

    wp_update_user( array(
        'ID'           => $userID,
        'display_name' => $data['first_name'] . ' ' . $data['last_name'],
        'first_name'   => $data['first_name'],
        'last_name'    => $data['last_name'],
    ) );

    wp_send_json_success( sprintf(__( 'welcome Back %s', 'textdomain' ), $data['first_name']
, ' ' . $data['last_name'] ) );
}

add_action( 'wp_ajax_custom_action', 'rm_ajax_handler' );
add_action( 'wp_ajax_nopriv_custom_action', 'rm_ajax_handler' );
```

## ajax.js

```
;(function() {
    wp.ajax.post(BEJS.action, {
        first_name: 'john',
        last_name: '%65doe',
        email: 'john.doe@example.com',
        _ajax_nonce: BEJS.nonce
    }).done( function( response ) {
        alert(`Success: ${response}`);
    }).fail( function( response ) {
        alert(`Error: ${response}`);
    });
})();
```

## OOP ajax submission using a simple class with nonce

You can copy and paste this whole plugin to try it. The class skeleton is used from [here](#).

### class-oop-ajax.cpp

```
<?php

/**
 * The plugin bootstrap file
 *
 * This file is read by WordPress to generate the plugin information in the plugin
 * Dashboard. This file defines a function that starts the plugin.
 *
 * @wordpress-plugin
 * Plugin Name:      Oop Ajax
 * Plugin URI:       http://
 * Description:      A simple example of using OOP principles to submit a form from the
 * front end.
 * Version:          1.0.0
 * Author:           Digvijay Naruka
 * Author URI:       http://
 * License:          GPL-2.0+
 * License URI:      http://www.gnu.org/licenses/gpl-2.0.txt
 * Text Domain:      oop-ajax
 * Domain Path:     /languages
 */

// If this file is called directly, abort.
if ( ! defined( 'WPINC' ) ) {
    die;
}

class Oop_Ajax {

    // Put all your add_action, add_shortcode, add_filter functions in __construct()
    // For the callback name, use this: array($this, '<function name>')
    // <function name> is the name of the function within this class, so need not be globally
```

```

unique
// Some sample commonly used functions are included below
public function __construct() {

    // Add Javascript and CSS for front-end display
    add_action('wp_enqueue_scripts', array($this, 'enqueue'));

    // Add the shortcode for front-end form display
    add_action( 'init', array( $this, 'add_form_shortcode' ) );
    // Add ajax function that will receive the call back for logged in users
    add_action( 'wp_ajax_my_action', array( $this, 'my_action_callback' ) );
    // Add ajax function that will receive the call back for guest or not logged in users
    add_action( 'wp_ajax_nopriv_my_action', array( $this, 'my_action_callback' ) );

}

// This is an example of enqueueing a JavaScript file and a CSS file for use on the front
end display
public function enqueue() {
    // Actual enqueues, note the files are in the js and css folders
    // For scripts, make sure you are including the relevant dependencies (jquery in this
case)
    wp_enqueue_script('my-ajax-script',    plugins_url('js/oop-ajax.js', __FILE__),
array('jquery'), '1.0', true);

    // Sometimes you want to have access to data on the front end in your Javascript file
// Getting that requires this call. Always go ahead and include ajaxurl. Any other
variables,
// add to the array.
// Then in the Javascript file, you can refer to it like this:
my_php_variables.ajaxurl
    wp_localize_script( 'my-ajax-script', 'my_php_variables', array(
        'ajaxurl' => admin_url('admin-ajax.php'),
        'nonce' => wp_create_nonce('_wponce')
    ));

}

/**
 * Registers the shortcode for the form.
 */
public function add_form_shortcode() {

    add_shortcode( "oop-ajax-add-form", array( $this, "add_form_front_end" ) );

}

/**
 * Processes shortcode oop-ajax-add-form
 *
 * @param array $atts The attributes from the shortcode
 *
 * @return mixed $output Output of the buffer
 */
function add_form_front_end($atts, $content) {

    echo "<form id='my_form'>";

        echo "<label for='name'>Name: </label>";
        echo "<input id='name' type='text' name='name' ";

```

```

    echo "<br>" ;

    echo "<label id='email' for='email'>Email: </label>" ;
    echo "<input type='text' name='email'>";

    echo "<br>" ;

    echo "<input type='hidden' name='action' value='my_action' >" ;
    echo "<input id='submit_btn' type='submit' name='submit' value='submit'> ";

    echo "</form><br><br>";
    echo "<div id='response'>ajax response will be here</div> ";
}

/**
 * Callback function for the my_action used in the form.
 *
 * Processes the data recieved from the form, and you can do whatever you want with it.
 *
 * @return echo response string about the completion of the ajax call.
 */
function my_action_callback() {
    // echo wp_die('<pre>' . print_r($_REQUEST) . "<pre>");

    check_ajax_referer( '_wpnonce', 'security');

    if( ! empty( $_POST ) ){

        if ( isset( $_POST['name'] ) ) {

            $name = sanitize_text_field( $_POST['name'] ) ;
        }

        if( isset( $_POST['email'] ) ) {

            $email = sanitize_text_field( $_POST['email'] ) ;
        }

        ////////////////////////////////////////////////////////////////////
        // do stuff with values
        // example : validate and save in database
        //           process and output
        ////////////////////////////////////////////////////////////////////

        $response = "wow <strong style= 'color:red'>". $name . "!</style></strong> you
rock, you just made ajax work with oop.";
        //this will send data back to the js function:
        echo $response;

    } else {

        echo "Uh oh! It seems I didn't eat today";
    }

    wp_die(); // required to terminate the call so, otherwise wordpress initiates the
termination and outputs weird '0' at the end.

}

}
//initialize our plugin

```

```
global $plugin;

// Create an instance of our class to kick off the whole thing
$plugin = new Oop_Ajax();
```

## oop-ajax.js

Put the js file inside the js directory i.e oop-ajax/js/ooP-ajax.js

```
(function($) {
    'use strict';

    $("#submit_btn").on('click', function() {
        // set the data
        var data = {
            action: 'my_action',
            security: my_php_variables.nonce,
            name: $("#name").val(),
            email: $("#email").val()
        }

        $.ajax({
            type: 'post',
            url: my_php_variables.ajaxurl,
            data: data,
            success: function(response) {
                //output the response on success
                $("#response").html(response);
            },
            error: function(err) {
                console.log(err);
            }
        });

        return false;
    });
})(jQuery);
```

---

# Chapter 12: Alternating main loop (pre\_get\_posts filter)

## Syntax

- `add_action( 'pre_get_posts', 'callback_function_name' );`
- `function callback_function_name( $query ) {}`
- `// for PHP 5.3.0 or above`
- `add_action( 'pre_get_posts', function( $query ){ } );`

## Parameters

Parameter	Details
<code>\$query</code>	( <i>WP_Query</i> ) Loop object

## Remarks

If you are using PHP 5.3.0 or above, you can use closures ([anonymous functions](#))

```
add_action( 'pre_get_posts', function( $query ) {  
    if( !$query->is_main_query() || is_admin() ) return;  
  
    // this code will run only if  
    // - this query is main query  
    // - and this is not admin screen  
});
```

## Examples

### Even more specific loop targeting

Let's say we want to change *main loop*, only for specific taxonomy, or post type. (ଆସକ୍ତ କହିବା ଯେ

ଆମେ କେବଳ ନିର୍ଦ୍ଦିଷ୍ଟ ଟ୍ୟାକ୍ସୋନୋମି କିମ୍ବା ପୋଷ୍ଟ ପ୍ରକାର ପାଇଁ ମୁଖ୍ୟ ଲୁପ୍ ପରିବର୍ତ୍ତନ କରିବାକୁ ଚାହୁଁ

)

Targeting only main loop on book post type archive page.

```
add_action( 'pre_get_posts', 'my_callback_function' );

function my_callback_function( $query ) {
    if( !$query->is_main_query() || is_admin() ) return;
    if( !is_post_type_archive( 'book' ) ) return;

    // this code will run only if
    // - this query is main query
    // - and this is not admin screen
```

```
// - and we are on 'book' post type archive page
}
```

You can also check for category, tag or custom taxonomy archive page using `is_category()`, `is_tag()` and `is_tax()`.

You can use any *conditional tag* available in WordPress.

## Show posts from only one category

```
add_action( 'pre_get_posts', 'single_category' );

function single_category( $query ) {
    if( !$query->is_main_query() || is_admin() ) return;

    $query->set( 'cat', '1' );
    return;
}
```

## Pre get posts filter basic usage

Sometimes you would like to change main WordPress query.

Filter `pre_get_posts` is the way to go.

For example using `pre_get_posts` you can tell [main loop](#) to show only 5 posts. Or to show posts only from one category, or excluding any category etc.

```
add_action( 'pre_get_posts', 'my_callback_function' );

function my_callback_function( $query ) {
    // here goes logic of your filter
}
```

As you can see, we are passing [main loop](#) query object into our callback function argument.

Important note here: **we are passing argument as a reference.** ଆମେ ଏକ ରେଫରେନ୍ସ ଭାବରେ ଆର୍ଗୁମେଣ୍ଟ ପାସ୍ କରୁଛୁ । ଏହାର ଅର୍ଥ ହେଉଛି ଯେ ଏହା କାର୍ଯ୍ୟ କରିବା ପାଇଁ ଆମକୁ ଜିଜ୍ଞାସା ଫେରାଇବା କିମ୍ବା କି ob ଶି ସି ଗ୍ଲୋବାଲ୍ ସେଟ୍ କରିବା ଆବଶ୍ୟକ ନାହିଁ । ଯେହେତୁ \$ ଜିଜ୍ଞାସା ହେଉଛି ମୁଖ୍ୟ ଜିଜ୍ଞାସା ବସ୍ତୁର ଏକ ରେଫରେନ୍ସ, ଆମ ବସ୍ତୁରେ ଆମେ କରୁଥିବା ସମସ୍ତ ପରିବର୍ତ୍ତନ ତୁରନ୍ତ ମୁଖ୍ୟ ଲୁପ୍ ଅବଜେକ୍ଟରେ ପ୍ରତିଫଳିତ ହୁଏ । (It means that we do not need to return query or set any globals to get it working. As \$query is a reference to the main query object, all changes we make on our object are immediately reflected in the main loop object.)

## Exclude category from posts list edit share

```
add_action( 'pre_get_posts', 'single_category_exclude' );

function single_category_exclude( $query ) {
    if( !$query->is_main_query() || is_admin() ) return;

    $query->set( 'cat', '-1' );
    return;
}
```

## Change posts\_per\_page for main loop

All we need to do is to use `set()` method of `$query` object.

It takes two arguments, first what we want to set and second what value to set.

```
add_action( 'pre_get_posts', 'change_posts_per_page' );

function change_posts_per_page( $query ) {
    if( !$query->is_main_query() || is_admin() ) return;

    $query->set( 'posts_per_page', 5 );
    return;
}
```

## Targeting only main WordPress loop

WordPress is applying `pre_get_posts` filter to literally any loop it generates. It means that all changes we are making in our callback function are applied to all existing loops. Obviously it is not what we want in most scenarios.

ଓର୍ଡ଼ପ୍ରେସ୍ ଏହା ସୃଷ୍ଟି କରୁଥିବା ଯେକ `any` ଶସି ଲୁପ୍ ପାଇଁ `pre_get_posts` ଫିଲ୍ଟର ପ୍ରୟୋଗ କରୁଛି | ଏହାର ଅର୍ଥ ହେଉଛି ଯେ ଆମେ ଆମର କଲବ୍ୟାକ୍ ଫଙ୍କସନ୍ ରେ କରୁଥିବା ସମସ୍ତ ପରିବର୍ତ୍ତନ ସମସ୍ତ ପ୍ରସ୍ଥାନ ଲୁପ୍ ଉପରେ ପ୍ରୟୋଗ କରାଯାଏ |

ଆଉ `vious` ା ହୁଁ, ଆମେ ଅଧିକାଂଶ ପରିସ୍ଥିତିରେ ଏହା ଚାହୁଁନାହୁଁ |

In most cases we would like to target only [main loop](#), and only for non-admin screens. ( ଅଧିକାଂଶ କ୍ଷେତ୍ରରେ ଆମେ କେବଳ ମୁଖ୍ୟ ଲୁପ୍, ଏବଂ କେବଳ ଅଣ-ଆଡମିନ୍ ସ୍କ୍ରିନ୍ ପାଇଁ ଟାର୍ଗେଟ୍ କରିବାକୁ ଚାହୁଁବୁ | )

We can use `is_main_query()` method and `is_admin()` global function to check if we are in the right place.

```
add_action( 'pre_get_posts', 'my_callback_function' );

function my_callback_function( $query ) {
    if( !$query->is_main_query() || is_admin() ) return;

    // this code will run only if
    // - this query is main query
    // - and this is not admin screen
}
```

# Chapter 13: Child Theme Basics

## Syntax

- **template** — is the name of the main WordPress theme, the parent.
- **child-theme** — is the package which overrides the **template**.

## Remarks

ମୁଁ ବିଜ୍ଞାପନ କରିଆସୁଛି ଯେ ଏକ ଶିଶୁ ଥିମର ବ୍ୟବହାର ସର୍ବଦା ଏକ ଭଲ ଜିନିଷ କିନ୍ତୁ ସର୍ବଦା ଏକ କିଛି |  
...

ଆମର ଟେମ୍ପଲେଟ୍ ଓଭର ରାଇଟ୍ ଉଦାହରଣରେ ଆସନ୍ତୁ କଳ୍ପନା କରିବା ଯେ ଏକ ଥିମର ଲେଖକ ପାର୍ଶ୍ୱ ଦଣ୍ଡ ଟେମ୍ପଲେଟ୍ରେ ନିଜର ଉନ୍ନତି ଯୋଗ କରୁଛନ୍ତି ଏବଂ ସେଠାରେ ଏକ ନୂତନ ହେବ |

I've been advertising that the use of a child theme is always a good thing but there is always a *But*  
...In our Template overwriting example let's imagine that the author of a theme is adding his own improvements to the sidebar template and there will be a new one at

/themes/template/sidebar.php

```
<?php
/**
 * The template for the sidebar containing the main widget area
 *
 * @link https://developer.wordpress.org/themes/basics/template-files/#template-partials
 */

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

Now our website won't benefit from the new `role="complementary"` spec because our child theme is still overwriting the **template** with its own file at `/themes/child-theme/sidebar.php`

ୱେବସାଇଟ୍ ରକ୍ଷକ ଭାବରେ ଏହା ଆମର କର୍ତ୍ତବ୍ୟ ଯେ ଆମେ କେଉଁ ଟେମ୍ପଲେଟ୍ ଗୁଡ଼ିକୁ ଓଭର ରାଇଟ୍ କରୁ ଏବଂ ଏକ ଅପଡେଟ୍ ର ଶୀଘ୍ର, ଟେମ୍ପଲେଟ୍ କୁ ଭଲ ଭାବରେ ଦେଖନ୍ତୁ ତେଣୁ ଆବଶ୍ୟକ ହେଲେ ଆପଣ ଶିଶୁ ଥିମ୍ ଫାଇଲଗୁଡ଼ିକୁ ଅପଡେଟ୍ କରନ୍ତୁ |

## Examples

### 2) The purpose

ଥିମ୍ ଅପଡେଟ୍ ରେ ସେଗୁଡ଼ିକ ହରାଇବାକୁ ଭୟ ନକରି ମୁଖ୍ୟ ଟେମ୍ପଲେଟର କଷ୍ଟମାଇଜେସନ୍ ରଖିବା ପାଇଁ ଶିଶୁ ଥିମ୍ଗୁଡ଼ିକ ଏକ ନିରାପଦ ଉପାୟ ଭାବରେ ଉଦ୍ଦିଷ୍ଟ | ମିଳିତ ଭାବରେ, ଯେତେବେଳେ ଆପଣ ନିଜ ୱେବସାଇଟରୁ ସକ୍ରିୟ ଟେମ୍ପଲେଟ୍ ଭିତରେ ଏକ ଫାଇଲ୍ ଏଡିଟ୍ କରିବାକୁ ଚାହାଁନ୍ତି, ଆପଣଙ୍କୁ ନିଜକୁ ପଚାରିବାକୁ ପଡିବ “କ’ଣ? ଯେତେବେଳେ ମୁଁ ଥିମ୍ ଅପଡେଟ୍ କରିବି ତାହା ଘଟିବାକୁ ଯାଉଛି କି? ଏବଂ ଉତ୍ତରଟି ସରଳ: ତୁମେ ସେଗୁଡ଼ିକୁ ହରାଇବ କାରଣ ଅପଡେଟ୍ ଏକ ସମ୍ପୂର୍ଣ୍ଣ ନୂଆ ଥିମ୍ ଆଣିବ | ଫୋଲ୍ଡର.

So let's look at a child theme as a folder with files which will overwrite the files with the same path in the parent theme. Now let's bring some real examples:

### Definition and requirements

A child theme is identified by WordPress when there is a directory (for example `child-theme`) inside `/wp-content/themes/` with the following files:

- `style.css`

This file must start with a comment template like this:

```
/*
Theme Name: Example child
Author: Me
Author URI: https://example.com/
Template: example
Text Domain: example-child-theme
Domain Path: /languages/
*/
```

The most important things to consider here are:

- The `Template` name must be exactly the folder name which holds the parent theme (aka the parent theme slug)
- Name your child theme in such a way you can easily identify it in Dashboard (usually just append `child` to the parent's name, like `Example child`)

- `index.php`
- `functions.php`

### 3) Template overwriting

The most common usage of a child theme is to override template parts. For example, a sidebar, if we have a theme with the following file at

`/themes/template/sidebar.php`

```
<?php
/**
 * The sidebar containing the main widget area.
 *
 * @link https://developer.wordpress.org/themes/basics/template-files/#template-partials
 */

if ( ! is_active_sidebar( 'sidebar-1' ) ) {
    return;
}
?>
<div id="sidebar" class="widget-area">
    <?php dynamic_sidebar( 'sidebar-1' ); ?>
```

```
</div>
```

We can definitely add our own file in child theme (with the specifications from the first example) with the following file

```
/themes/child-theme/sidebar.php
```

```
<?php
/**
 * The sidebar containing the main widget area.
 */

if ( ! is_active_sidebar( 'sidebar-1' ) ) {
    return;
}
?>
<div id="my-sidebar" class="my-own-awesome-class widget-area">
    <div class="my-wrapper">
        <?php dynamic_sidebar( 'sidebar-1' ); ?>
    </div>
</div>
```

Now `my-own-awesome-class` is safe in **child theme** and it won't be removed at any theme update and WordPress will always choose a template from child themes when it does find one on the same path.

## Assets replacement

ଯଦିଓ ଏହା ଏକ ସର୍ବୋତ୍ତମ ଅଭ୍ୟାସ ନୁହେଁ, ବେଳେବେଳେ ଆପଣଙ୍କୁ CSS କିମ୍ବା JS ଫାଇଲ୍ କିମ୍ବା ଲାଇବ୍ରେରୀ ପରି ସମ୍ପତ୍ତି ବଦଳାଇବାକୁ ପଡ଼ିବ ।

ଧାନ ଦିଅନ୍ତୁ ଯେ ଓଡ଼ିଆ ଟେମ୍ପଲେଟ୍ ଓଭର ରାଇଟ୍ ସିଷ୍ଟମ୍ .php ବ୍ୟତୀତ ଅନ୍ୟ କିଛି ସହିତ କାମ କରେ ନାହିଁ ।

ଫାଇଲଗୁଡ଼ିକ, ତେଣୁ ଯେତେବେଳେ ଆମେ ସମ୍ପତ୍ତି ବିଷୟରେ କଥାବାର୍ତ୍ତା କରୁ ଆମେ ପଞ୍ଜୀକୃତ ସମ୍ପତ୍ତିକୁ ସୁଚୀତ କରୁ ।

One example could be the replacement of jQuery library with your desired version. In our child theme `functions.php` file we need to add a function that removes the current jQuery version and add our own from CDN(remember is just an example).

```
/**
 * Dequeue the jQuery script and add our own version.
 *
 * Hooked to the wp_print_scripts action, with a late priority (100),
 * so that it is after the script was enqueued.
 */
function my_own_theme_scripts() {
    // remove the current version
    wp_dequeue_script( 'jquery' );
    // register my desired version
    wp_register_script( 'jquery', 'https://code.jquery.com/jquery-3.1.0.min.js', false,
'3.1.0' );
    // load my version, here or somewhere else
    wp_enqueue_script( 'jquery' );
}
add_action( 'wp_print_scripts', 'my_own_theme_scripts' );
```



---

# Chapter 14: Create a Post Programmatically

## Syntax

- `wp_insert_post(array $args, bool $wp_error);`

## Parameters

Parameter	Description
\$args (Array Required)	A Key Value Array of the below elements.
\$wp_error (Boolean Optional)	Return a WP_Error in case of failure.

## Remarks

---

---

## Arguments

The next table shows you a list of elements that you can use inside of the first parameter (Array).

Parameter	Description
ID	(Int) The post ID. If equal to something other than 0, the post with that ID will be updated. Default 0.
post_author	(Int) The ID of the user who added the post. Default is the current user ID.
post_date	(String) The date of the post. Default is the current time.
post_date_gmt	(String) The date of the post in the GMT timezone. Default is the value of \$post_date.
post_content	(Mixed) The post content. Default empty.
post_content_filtered	(String) The filtered post content. Default empty.
post_title	(String) The post title. Default empty.
post_category	(Array) Array of post category values.
post_excerpt (String)	The post excerpt. Default empty.

Parameter	Description
post_status	(String) The post status. Default draft.
post_type	(String) The post type. Default post.
comment_status	(String) Whether the post can accept comments. Accepts open or closed. Default is the value of default_comment_status option.
ping_status	(String) Whether the post can accept pings. Accepts open or closed. Default is the value of default_ping_status option.
post_password	(String) The password to access the post. Default empty.
post_name	(String) The post name or slug. Default is the sanitized post title when creating a new post.
to_ping	(String) Space or carriage return-separated list of URLs to ping. Default empty.
pinged	(String) Space or carriage return-separated list of URLs that have been pinged. Default empty.
post_modified	(String) The date when the post was last modified. Default is the current time.
post_modified_gmt	(String) The date when the post was last modified in the GMT timezone. Default is the current time.
post_parent	(Int) Set this for the post it belongs to, if any. Default 0.
menu_order	(Int) The order the post should be displayed in. Default 0.
post_mime_type	(String) The mime type of the post. Default empty.
guid	(String) Global Unique ID for referencing the post. Default empty.
tax_input	(Array) Array of taxonomy terms keyed by their taxonomy name. Default empty.
meta_input	(Array) Array of post meta values keyed by their post meta key. Default empty.

---

## Avoid Duplicated Posts

When you execute this function, you could probably get a duplicated post, at least that happened to me. (You can check it into the Post WordPress Section)

I found a [solution](#):

```
if( !get_page_by_title( $title, 'OBJECT', 'post' ) ){
    $my_post = array('post_title' => $title,
        'post_content' => 'Content',
        'tags_input' => $tags,
        'post_category' => array(2),
        'post_status' => 'publish'
    );

    $result = wp_insert_post( $my_post );
}
```

## Explanation

ଆପଣ ଏକ ନୂତନ ପୋଷ୍ଟ ସୂଚନା କରିବା ପୂର୍ବରୁ, ବ title ଧ କରନ୍ତୁ ଯଦି ନୂତନ ପୋଷ୍ଟ ପାରାମିଟର ଭାବରେ ପୋଷ୍ଟ ଟାଇଟଲ ବ୍ୟବହାର କରି ବିଦ୍ୟମାନ ଅଛି, ଯଦି କ post ଶିକ୍ଷିତ ପୋଷ୍ଟ ଟାଇଟଲ ନାହିଁ, ତେବେ ଆପଣ ଆପଣଙ୍କର ନୂତନ ପୋଷ୍ଟକୁ ସଂରକ୍ଷଣ କରିପାରିବେ ।

Check `get_page_by_title`'s documentation [here](#).

## Examples

### Introduction

Sometimes we have another editor somewhere else instead of TinyMCE (Wordpress Default Editor). That happen when we are creating our own Theme, Plugin or something specific; and we need to write and manipulate a type of post and save it into our WP Database. ( ବେଳେବେଳେ TinyMCE (ୱାଉର୍ଡପ୍ରେସ୍ ଡିଫଲ୍ଟ ଏଡିଟର) ବଦଳରେ ଆମର ଅନ୍ୟ ଏକ ସମ୍ପାଦକ ଥାଏ । ଯେତେବେଳେ ଆମେ ଆମର ନିଜସ୍ୱ ଥିମ୍, ପ୍ଲଗିନ୍ କିମ୍ବା କିଛି ନିର୍ଦ୍ଦିଷ୍ଟ ସ୍ୱିଷ୍ଟ୍ କରୁ, ତାହା ଘଟେ; ଏବଂ ଆମକୁ ଏକ ପ୍ରକାର ପୋଷ୍ଟ ଲେଖିବା ଏବଂ ପରିଚାଳନା କରିବା ଏବଂ ଏହାକୁ ଆମର WP ଡାଟାବେସରେ ସେଭ୍ କରିବା ଆବଶ୍ୟକ ।)

So, if you are on that situation, you can use a Wordpress Function called:

```
wp_insert_post( array $args, bool $wp_error );
```

### Create a Basic Post

```
$basic_post_args = array(
    'post_title' => 'My Basic Post',
    'post_content' => 'This is a basic content',
    'post_status' => 'publish',
    'post_author' => 1,
    'post_category' => array(8, 59)
);

wp_insert_post( $basic_post_args );
```

### Create a Basic Page

```
$basic_page_args = array(
    'post_title' => 'My Basic Page',
    'post_content' => 'This is a basic content',
    'post_type' => 'page',
    'post_status' => 'publish',
    'post_author' => 1
);

wp_insert_post( $basic_page_args );
```

---

# Chapter 15: Create Template for Custom Post Type

## Examples

### Creating a custom template for Custom Post type book

ଆମର କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରର ସିଙ୍ଗଲ୍ ପୋଷ୍ଟଗୁଡ଼ିକ ପାଇଁ ଏକ ଟେମ୍ପଲେଟ୍ ସୃଷ୍ଟି କରିବାକୁ, ଆମକୁ ସିଙ୍ଗଲ୍-ପୋଷ୍ଟ\_ଟାଇପ୍\_ନାମ.ଫାଇଲ୍ ନାମକ ଏକ ଫାଇଲ୍ ତିଆରି କରିବାକୁ ପଡ଼ିବ ଯେଉଁଠାରେ `post_type_name` ହେଉଛି ଆମର କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରର ନାମ |( we need to create a file called `single-post_type_name.php` where `post_type_name` is the name of our custom post ty)

ଉଦାହରଣ ସ୍ୱରୂପ, ଯଦି ଆମର କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରକୁ “ବୁକ୍” କୁହାଯାଏ, ଆମକୁ ସିଙ୍ଗଲ୍ ନାମକ ଏକ PHP ଫାଇଲ୍ ତିଆରି କରିବାକୁ ପଡ଼ିବ |

`book.php` | ଧ୍ୟାନ ଦିଅନ୍ତୁ ଯେ ଆମେ ଆମର କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାରର ଏକକ ନାମ ବ୍ୟବହାର କରିଛୁ |

ଅନୁ ଫୋଲ୍ଡରରୁ `single.php` ର ବିଷୟବସ୍ତୁକୁ କପି କରନ୍ତୁ ଏବଂ ଏହାକୁ ନୂତନ ଟେମ୍ପଲେଟରେ ଲେପନ କରନ୍ତୁ ଏବଂ ଏହାକୁ ସଂରକ୍ଷଣ କରନ୍ତୁ ତେବେ ଟେମ୍ପଲେଟ୍ କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାର ବ୍ୟକ୍ତିଗତ ପୃଷ୍ଠା ପାଇଁ ପ୍ରୟୋଗ ହେବ |

### Custom Post Type Templates

---

---

## Custom Post Type Archive:

To create an archive template for a custom post type you have to set the `has_archive` argument equal to `true` in your `register_post_type()` function. In the example below a custom post type is created for an Event post type.

```
add_action( 'init', 'create_events_post_type' );
function create_events_post_type() {
    register_post_type( 'event',
        array(
            'labels' => array(
                'name' =>__( 'Events' ),
                'singular_name' =>__( 'Event' )
            ),
            'public' => true,
            'has_archive' => true,
        )
    );
}
```

To [create a template](#) for new custom post types you will have to create a new template file. To create a template for the single post pages you would name it `single-{post_type}.php` and `archive-{post_type}.php` for the archive.

The filename for our archive template will be `archive-event.php` and for the event page it would be `single-event.php`. Both files should be in your themes root directory.

An example archive template would look like this. Pulled from the [twentyseventeen theme](#).

```
<?php
/**
 * The template for displaying archive pages
 *
 * @link https://codex.wordpress.org/Template_Hierarchy
 *
 * @package WordPress
 * @subpackage Twenty_Seventeen
 * @since 1.0
 * @version 1.0
 */

get_header(); ?>

<div class="wrap">

    <?php if ( have_posts() ) : ?>
        <header class="page-header">
            <?php
                the_archive_title( '<h1 class="page-title">', '</h1>' );
                the_archive_description( '<div class="taxonomy-description">', '</div>' );
            ?>
        </header><!-- .page-header -->
    <?php endif; ?>

    <div id="primary" class="content-area">
        <main id="main" class="site-main" role="main">

            <?php
                if ( have_posts() ) : ?>
                    <?php
                        /* Start the Loop */
                        while ( have_posts() ) : the_post();

                            /*
                             * Include the Post-Format-specific template for the content.
                             * If you want to override this in a child theme, then include a file
                             * called content-___.php (where___is the Post Format name) and that will be
                             used instead.
                             */
                            get_template_part( 'template-parts/post/content', get_post_format() );

                        endwhile;

                        the_posts_pagination( array(
                            'prev_text' => twentyseventeen_get_svg( array( 'icon' => 'arrow-left' ) ) .
                            '<span class="screen-reader-text">' . __( 'Previous page', 'twentyseventeen' ) . '</span>',
                            'next_text' => '<span class="screen-reader-text">' . __( 'Next page',
                            'twentyseventeen' ) . '</span>' . twentyseventeen_get_svg( array( 'icon' => 'arrow-right' ) ),
                            'before_page_number' => '<span class="meta-nav screen-reader-text">' . __(
                            'Page', 'twentyseventeen' ) . ' </span>',
                            ) );

                    else :

                        get_template_part( 'template-parts/post/content', 'none' );

                    endif; ?>
            ?>
        </main>
    </div>
</div>
```

```

        </main><!-- #main -->
    </div><!-- #primary -->
    <?php get_sidebar(); ?>
</div><!-- .wrap -->

<?php get_footer();

```

## Custom Post Type Single template:

Here is an example of a single template. Pulled from the [twentyseventeen theme](#).

```

<?php
/**
 * The template for displaying all single posts
 *
 * @link https://developer.wordpress.org/themes/basics/template-hierarchy/#single-post
 *
 * @package WordPress
 * @subpackage Twenty_Seventeen
 * @since 1.0
 * @version 1.0
 */

get_header(); ?>

<div class="wrap">
    <div id="primary" class="content-area">
        <main id="main" class="site-main" role="main">

            <?php
                /* Start the Loop */
                while ( have_posts() ) : the_post();

                    get_template_part( 'template-parts/post/content', get_post_format() );

                    // If comments are open or we have at least one comment, load up the
comment template.
                    if ( comments_open() || get_comments_number() ) :
                        comments_template();
                    endif;

                    the_post_navigation( array(
                        'prev_text' => '<span class="screen-reader-text">' . __( 'Previous
Post', 'twentyseventeen' ) . '</span><span aria-hidden="true" class="nav-subtitle">' . __(
'Previous', 'twentyseventeen' ) . '</span> <span class="nav-title"><span class="nav-title-
icon-wrapper">' . twentyseventeen_get_svg( array( 'icon' => 'arrow-left' ) ) .
'</span>%title</span>',
                        'next_text' => '<span class="screen-reader-text">' . __( 'Next Post',
'twentyseventeen' ) . '</span><span aria-hidden="true" class="nav-subtitle">' . __( 'Next',
'twentyseventeen' ) . '</span> <span class="nav-title">%title<span class="nav-title-
icon-wrapper">' . twentyseventeen_get_svg( array( 'icon' => 'arrow-right' ) ) . '</span></span>',
                    ) );

                    endwhile; // End of the loop.
                ?>

        </main><!-- #main -->

```

```
</div><!-- #primary -->
<?php get_sidebar(); ?>
</div><!-- .wrap -->

<?php get_footer();
```

**Both template examples are pulling in [partials](#) to display the inner content.**

If your child/parent theme has a single/archive template you should use that code as boilerplate for your new templates.

# Chapter 16: Creating a custom template

## Examples

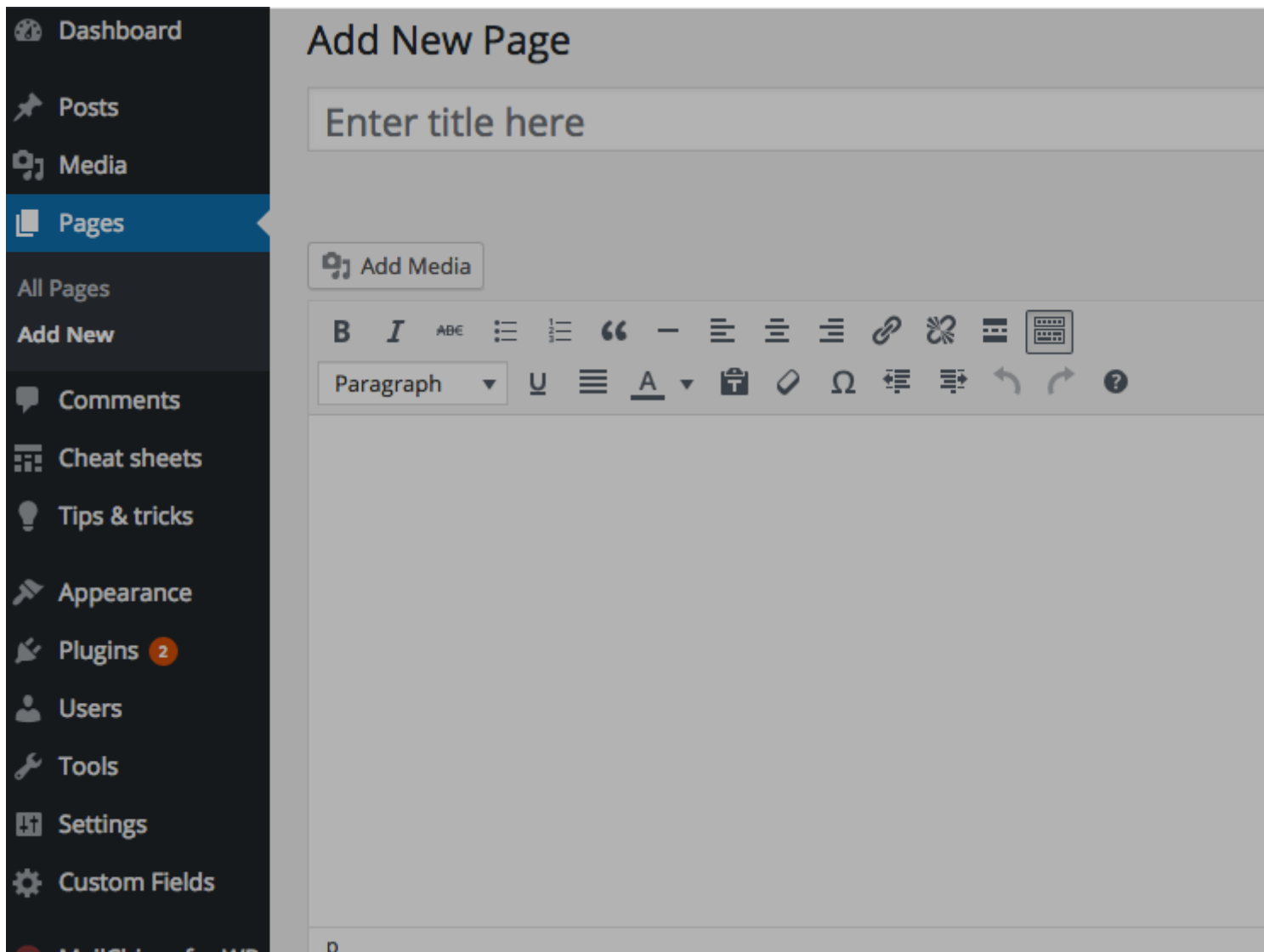
### Creating basic blank template

ଏକ କଷ୍ଟମ୍ ଟେମ୍ପଲେଟ୍ ସୃଷ୍ଟି କରିବାକୁ ଆମକୁ ପ୍ରଥମେ ଏକ ଥିମ୍ ଡିରେକ୍ଟୋରୀରେ php ଫାଇଲ୍ ତିଆରି କରିବାକୁ ପଡ଼ିବ । ଆପଣ ଚାହୁଁଥିବା ଉପାୟରେ ଆପଣ ଏହାର ନାମକରଣ କରିପାରିବେ । ଏହି ଉଦାହରଣ ପାଇଁ ଆମେ example.php ସୃଷ୍ଟି କରିବା ।

```
<?php
/*
Template Name: Example
*/
?>
```

ଓର୍ଡ଼ପ୍ରେସ୍ ଦ୍ୱାରା ଏକ ଟେମ୍ପଲେଟ୍ ଭାବରେ ସ୍ୱୀକୃତି ପାଇବାକୁ ଆମ example.php ଭିତରେ ବ୍ୟାଖ୍ୟା କରିବାକୁ ଥିବା ଗୋଟିଏ ଏବଂ କେବଳ ଜିନିଷ ହେଉଛି ଟେମ୍ପଲେଟ୍ ନାମ । ଏହିପରି ଆମେ ଏକ ଫାଇଲ୍ ର ଶୀର୍ଷରେ ବିଶେଷ ମନ୍ତବ୍ୟ ଦେଇ କିଣିବା କିଣୁ:

And now when we should see our template listed in **Template dropdown** in **Page Attributes Box**



## Including header and footer in our template

Let's extend our template from above and include content from **header.php** and **footer.php**

### Including header:

ଟେମ୍ପଲେଟ୍ ନାମ ମନେପାରି ପରେ ଆମେ ହେଡର୍ ଅନ୍ତର୍ଭୁକ୍ତ କରିବୁ ।

ଏହା କରିବା ପାଇଁ ଦୁଇଟି ସାଧାରଣ ଉପାୟ ଅଛି । ଉଭୟ ଠିକ୍ ଏବଂ ସମାନ କାମ କରନ୍ତି, ଏହା କେବଳ ଆପଣଙ୍କ ଶ style ନାମ ଏବଂ କୋଡ୍ କିପରି ଦେଖାଯାଏ ।

First way:

```
<?php
/*
Template Name: Example
*/
get_header();
?>
```

Second way:

```
<?php
/*
Template Name: Example
*/
?>
<?php get_header(); ?>
```

### Including footer:

ଫୁଟର୍ ଅନ୍ତର୍ଭୁକ୍ତ କରି ସମାନ ଭାବରେ କାର୍ଯ୍ୟ କରେ, କେବଳ ଗୋଟିଏ ଜିନିଷ ଅଛି ଯାହାକୁ ଆମେ ଧ୍ୟାନ ଦେବା ଆବଶ୍ୟକ, ଏବଂ ତାହା ହେଉଛି ଯେ ଆମେ ହେଡର୍ ଅନ୍ତର୍ଭୁକ୍ତ କରିବା ପରେ ଫୁଟର୍ ଅନ୍ତର୍ଭୁକ୍ତ କରୁ । ଟେଣ୍ଟ ଅନ୍ତିମ ଟେମ୍ପଲେଟ୍ ଏହିପରି କିଛି ଦେଖାଯିବା ଉଚିତ୍ । (Including footer works the same way, there is only one thing we need to care about, and that is that we include footer after we included header. So the final template should look something likethis.)

```
<?php
/*
Template Name: Example
*/
get_header();
?>

<?php get_footer(); ?>
```

## Custom template with content

We will further extend our template and include title of the page and a content

```
<?php
/*
Template Name: Example
*/
```

```
get_header();

the_title();
the_content();

get_footer();
```

And if you want you can wrap them with HTML elements like this

```
<?php
/*
Template Name: Example
*/
get_header();

echo '<h1>' . the_title() . '</h1>';
echo '<section>' . the_content() . '</section>';

get_footer();
```

Or if you prefer working like normal HTML file, without using echo

```
<?php
/*
Template Name: Example
*/
get_header();
?>

<h1><?php the_title(); ?></h1>
<section><?php the_content(); ?></section>

<?php get_footer(); ?>
```

---

# Chapter 17: Custom excerpts with `excerpt_length` and `excerpt_more`

## Examples

### Limit excerpt length to 50 words

Put the following code in **functions.php**:

```
function themify_custom_excerpt_length( $length ) {  
    return 50;  
}  
add_filter( 'excerpt_length', 'themify_custom_excerpt_length', 999 );
```

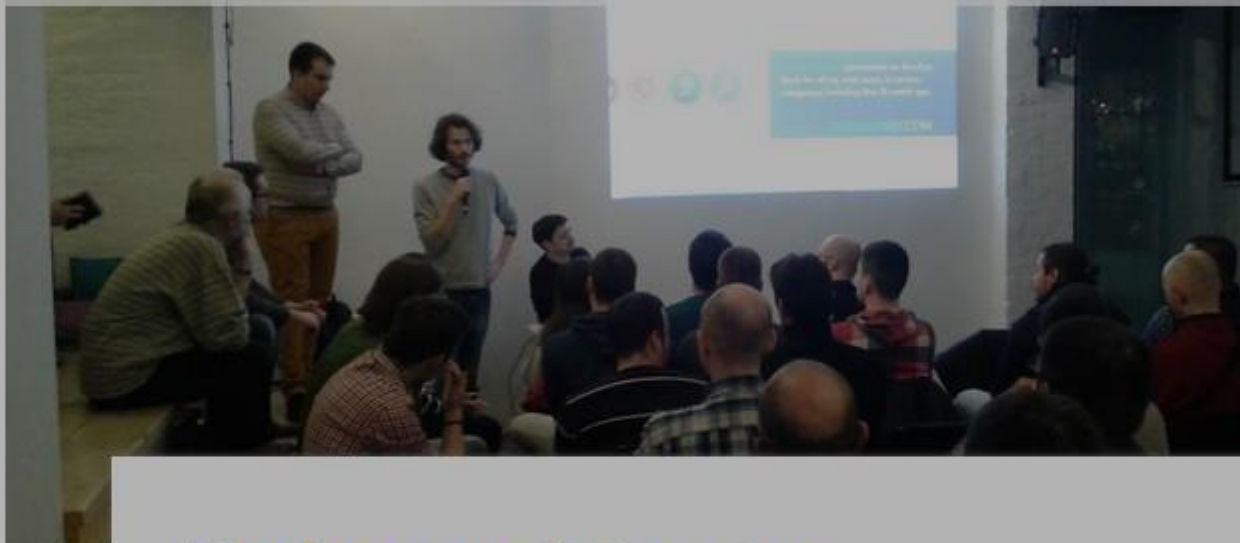
Use 999 as the priority to ensure that the function runs after the default WordPress filter, otherwise it would override what is set here.

### Adding a Read More link at the end of the excerpt

To do this, put the following code in **functions.php**:

```
function custom_excerpt_more($more) {  
    return '<a href="'. get_permalink($post->ID) . '">Read More</a>';  
}  
add_filter('excerpt_more', 'custom_excerpt_more');
```

The results should look like this:



## Utisci sa prvog iOS meetupa

Mar 4, 2016

Utisci sa prvog iOS meetupa su fantastični, što bi u našem narodu rekli-prvo pa muško! Ovo okupljanje dokazalo je da iOS zajednica u Srbiji i te kako postoji i da uopšte nije mala kao što se mislilo. Na događaju je bilo nešto više od 100 ljudi, a predavanja su bila izuzetno zanimljiva i korisna. Organizatori → [Read More](#)

[Ostavi komentar](#)



## Adding a few dots at the end of the excerpt

In our **functions.php**

```
function new_excerpt_more( $more ) {  
    return '.....';  
}  
add_filter('excerpt_more', 'new_excerpt_more');
```

We should get this:



## Utisci sa prvog iOS meetupa

Mar 4, 2016

Utisci sa prvog iOS meetupa su fantastični, što bi u našem narodu rekli-prvo pa muško! Ovo okupljanje dokazalo je da iOS zajednica u Srbiji i te kako postoji i da uopšte nije mala kao što se mislilo. Na događaju je bilo nešto više od 100 ljudi, a predavanja su bila izuzetno zanimljiva i korisna. Organizatori.....

[Ostavi komentar](#)



---

# Chapter 18: Custom Post Types

## Syntax

- `register_post_type( $post_type, $args );`

## Parameters

Parameter	Details
<code>\$post_type</code>	(string) (Required)
<code>\$args</code>	(array/string) (Optional)

## Examples

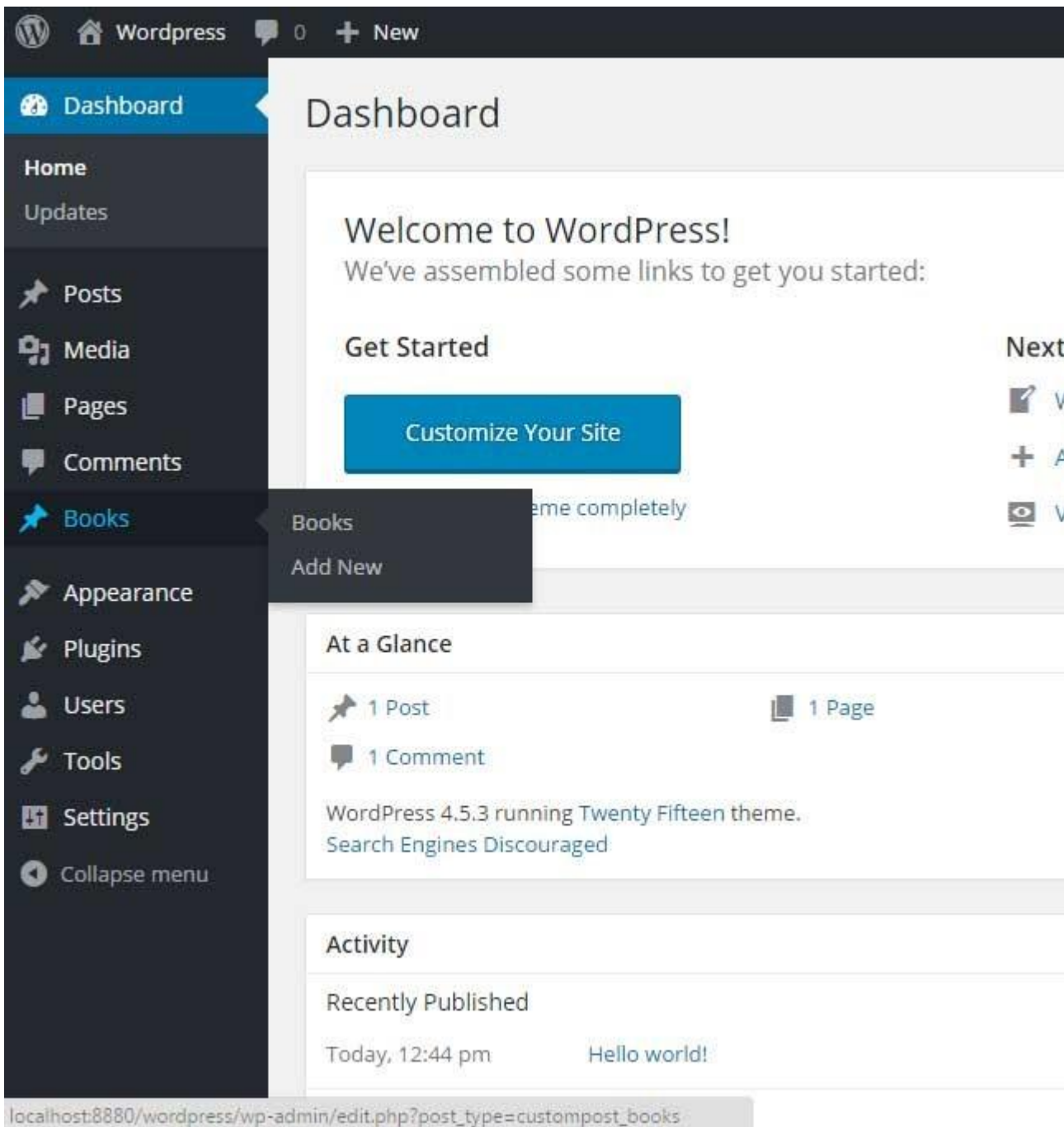
### Registering a Custom Post Type

Say you have a library website, and you want to have a custom post type named *Books*. It can be registered as

```
function create_bookposttype() {
    $args = array(
        'public' => true,
        'labels' => array(
            'name' =>__( 'Books' ),
            'singular_name' =>__( 'Book' )
        ),
    );
    register_post_type( 'custompost_books', $args );
}

add_action( 'init', 'create_bookposttype' );
```

and, as simple as that is, you now have a custom post type registered.



This snippet can be placed in your theme `functions.php` file, or within a plugin structure.

## Add Custom Post Types to Main Query

Registering a custom post type does not mean it gets added to the main query automatically. You need to use `pre_get_posts` filter to add custom post types to main query.

```
// Show posts of 'post' and 'book' custom post types on home page
add_action( 'pre_get_posts', 'add_my_post_types_to_query' );

function add_my_post_types_to_query( $query ) {
    if ( is_home() && $query->is_main_query() )
        $query->set( 'post_type', array( 'post', 'book' ) );
    return $query;
}
```

## Adding Custom Post Types to Main RSS Feed

Registering a custom post type does not mean it gets added to the main RSS feed automatically. You need to use `request filter` to add custom post types to main RSS feed.

```
// Add 'books' custom post types on main RSS feed
function add_book_post_types_to_rss($qv) {
    if (isset($qv['feed']) && !isset($qv['post_type']))
        $qv['post_type'] = array('post', 'books', );
    return $qv;
}
add_filter('request', 'add_book_post_types_to_rss');
```

## Register Custom Post Type

```
if ( ! function_exists('products_post_type') ) {

function products_post_type() {

    $labels = array(
        'name'                => _x( 'Products', 'Post Type General Name', 'text_domain' ),
        'singular_name'       => _x( 'Product', 'Post Type Singular Name', 'text_domain' ),
        'menu_name'           => __( 'Products', 'text_domain' ),
        'name_admin_bar'      => __( 'Product', 'text_domain' ),
        'archives'            => __( 'Item Archives', 'text_domain' ),
        'attributes'          => __( 'Item Attributes', 'text_domain' ),
        'parent_item_colon'   => __( 'Parent Product:', 'text_domain' ),
        'all_items'           => __( 'All Products', 'text_domain' ),
        'add_new_item'        => __( 'Add New Product', 'text_domain' ),
        'add_new'              => __( 'New Product', 'text_domain' ),
        'new_item'            => __( 'New Item', 'text_domain' ),
        'edit_item'           => __( 'Edit Product', 'text_domain' ),
        'update_item'         => __( 'Update Product', 'text_domain' ),
        'view_item'          => __( 'View Product', 'text_domain' ),
        'view_items'         => __( 'View Items', 'text_domain' ),
        'search_items'        => __( 'Search products', 'text_domain' ),
        'not_found'           => __( 'No products found', 'text_domain' ),
        'not_found_in_trash'  => __( 'No products found in Trash', 'text_domain' ),
        'featured_image'      => __( 'Featured Image', 'text_domain' ),
        'set_featured_image'  => __( 'Set featured image', 'text_domain' ),
        'remove_featured_image' => __( 'Remove featured image', 'text_domain' ),
        'use_featured_image'  => __( 'Use as featured image', 'text_domain' ),
        'insert_into_item'    => __( 'Insert into item', 'text_domain' ),
        'uploaded_to_this_item' => __( 'Uploaded to this item', 'text_domain' ),
        'items_list'          => __( 'Items list', 'text_domain' ),
        'items_list_navigation' => __( 'Items list navigation', 'text_domain' ),
        'filter_items_list'  => __( 'Filter items list', 'text_domain' ),
    );

    $args = array(
        'label'                => __( 'Product', 'text_domain' ),
        'description'         => __( 'Product information pages.', 'text_domain' ),
        'labels'               => $labels,
        'supports'             => array( 'title', 'editor', 'excerpt', 'author', 'thumbnail',
'comments', 'custom-fields', ),
        'taxonomies'          => array( 'category', 'post_tag' ),
        'hierarchical'        => false,
        'public'               => true,
        'show_ui'              => true,
    );
}
}
```

```

        'show_in_menu'           => true,
        'menu_position'        => 5,
        'menu_icon'            => 'dashicons-products',
        'show_in_admin_bar'    => true,
        'show_in_nav_menus'    => true,
        'can_export'           => true,
        'has_archive'          => true,
        'exclude_from_search'  => false,
        'publicly_queryable'   => true,
        'capability_type'      => 'page',
        'show_in_rest'         => true,
    );
    register_post_type( 'product', $args );
}
add_action( 'init', 'products_post_type', 0 );
}

```

## Custom Post Type using Twenty Fifteen WordPress Theme

You can use any name for the function.

```

function custom_postype(){
    register_post_type('cus_post',array(

        'labels'=>array(
            'name'=>'khaiyam'// Use any name you want to show in menu for your users
        ),
        'public'=>true,// **Must required
        'supports'=>array('title','editor','thumbnail')// Features you want to provide on your
posts
    ));
}
add_action('after_setup_theme','custom_postytp');

```

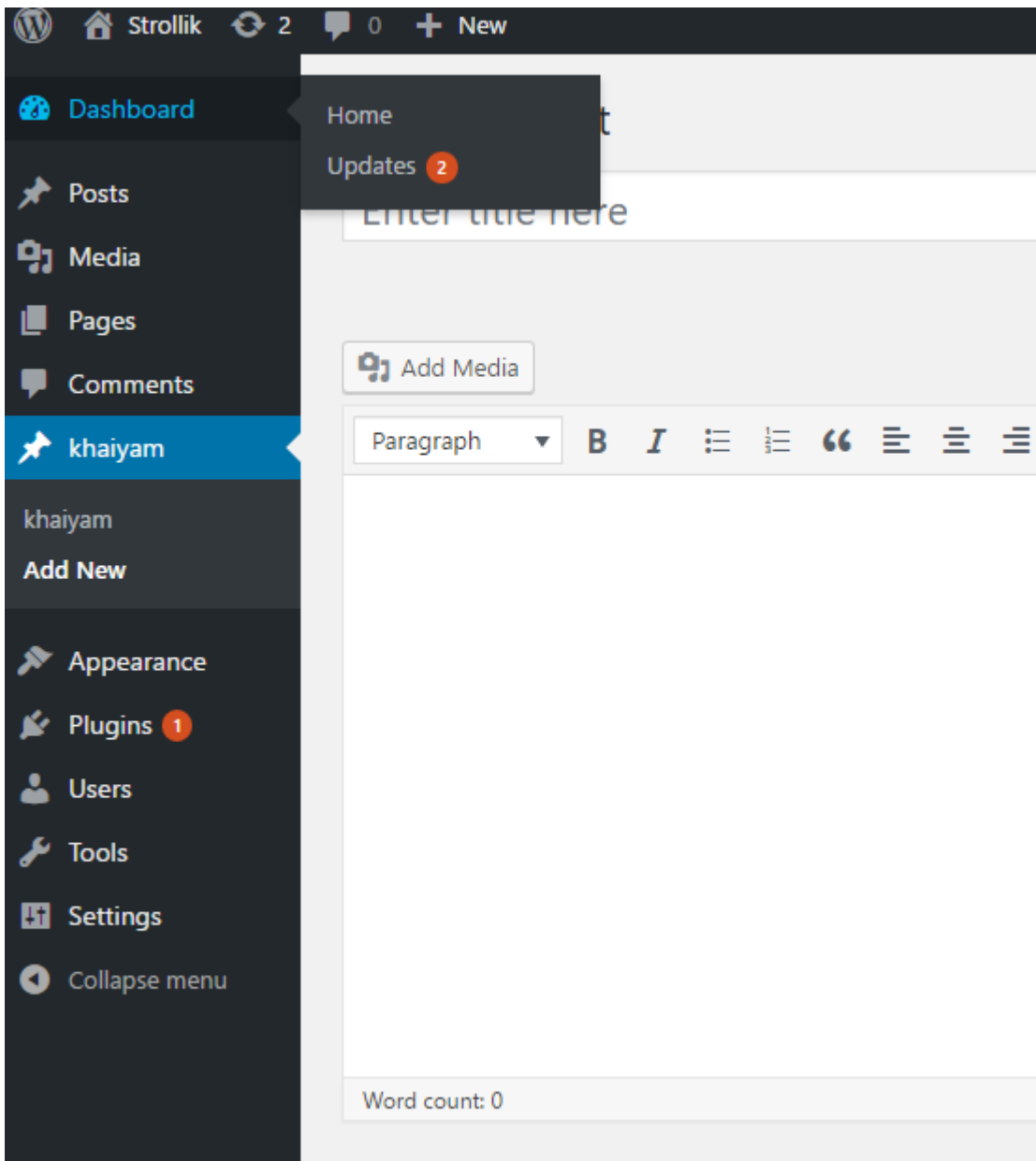
or

```

add_action('init','custom_postytp');

```

You can use any of the hooks you want but of course they have different meaning and uses.



## Custom post type in default search

You can add custom post type posts on default wordpress search, Add below code in theme functions.php

```
function my_search_filter($query) {
    if ( !is_admin() && $query->is_main_query() ) {
        if ($query->is_search) {
            $query->set('post_type', array( 'news','post','article' ) );
        }
    }
}
add_action('pre_get_posts', 'my_search_filter');
```

---

# Chapter 19: Customizer Basics (Add Panel, Section, Setting, Control)

## Examples

### Add a Customizer Panel

```
<?php
/**
 * Panel: WPCustomize
 *
 * Basic Customizer panel with basic controls.
 *
 * @since 1.0.0
 * @package WPC
 */

// Exit if accessed directly.
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

// Customize function.
if ( ! function_exists( 'wpc_panel_wpcustomize' ) ) {
    // Customize Register action.
    add_action( 'customize_register', 'wpc_panel_wpcustomize' );
    /**
     * Customize Panel.
     *
     * Adds a Panel, Section with basic controls.
     *
     * @param object WP_Customize $wp_customize Instance of the WP_Customize_Manager class.
     * @since 1.0.0
     */
    function wpc_panel_wpcustomize( $wp_customize ) {
        // Panel: Basic.
        $wp_customize->add_panel( 'wpc_panel_wpcustomize', array(
            'priority'      => 10,
            'title'         =>__( 'WPCustomize Panel Title', 'WPC' ),
            'description'   =>__( 'Panel Description', 'WPC' ),
            'capability'    => 'edit_theme_options'
        ) );
    }
}
```

### Add a Customizer Section With Basic Settings and their Controls

Panels can have sections, sections can have settings, and settings can have controls. Settings are saved in the database, while the controls for particular settings are only used to display their corresponding setting to the user.

This code creates a basic section in the panel from above. Inside are a few basic settings with controls

attached.

```
<?php
/**
 * Section: Basic
 *
 * Basic Customizer section with basic controls.
 *
 * @since 1.0.0
 * @package WPC
 */

// Exit if accessed directly.
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

// Customize function.
if ( ! function_exists( 'wpc_customize_panel_basic' ) ) {
    // Customize Register action.
    add_action( 'customize_register', 'wpc_customize_panel_basic' );

    /**
     * Customize Panel.
     *
     * Adds a Panel, Section with basic controls.
     *
     * @param object WP_Customize $wp_customize Instance of the WP_Customize_Manager class.
     * @since 1.0.0
     */
    function wpc_customize_panel_basic( $wp_customize ) {
        // Section: Basic.
        $wp_customize->add_section( 'wpc_section_basic', array(
            'priority'      => 10,
            'panel'         => 'wpc_panel_wpcustomize',
            'title'         => __( 'Basic Section Title', 'WPC' ),
            'description'   => __( 'Section Description.', 'WPC' ),
            'capability'    => 'edit_theme_options'
        ) );

        // Setting: Text.
        $wp_customize->add_setting( 'wpc_text', array(
            'type'          => 'theme_mod',
            'default'       => 'Placeholder.',
            'transport'     => 'refresh', // Options: refresh or postMessage.
            'capability'    => 'edit_theme_options',
            'sanitize_callback' => 'esc_attr'
        ) );

        // Control: Text.
        $wp_customize->add_control( 'wpc_text', array(
            'label'         => __( 'Text', 'WPC' ),
            'description'   => __( 'Description', 'WPC' ),
            'section'       => 'wpc_section_basic',
            'type'          => 'text'
        ) );

        // Setting: Textarea.
        $wp_customize->add_setting( 'wpc_textarea', array(
            'type'          => 'theme_mod',
            'default'       => 'Placeholder textarea.',
        ) );
    }
}
```

```

        'transport'           => 'refresh', // Options: refresh or postMessage.
        'capability'         => 'edit_theme_options',
        'sanitize_callback'  => 'exc_textarea'
    ) );

    // Control: Textarea.
    $wp_customize->add_control( 'wpc_textarea', array(
        'label'           =>__( 'Textarea', 'WPC' ),
        'description'    =>__( 'Description', 'WPC' ),
        'section'        => 'wpc_section_basic',
        'type'           => 'textarea'
    ) );

    // Setting: Checkbox.
    $wp_customize->add_setting( 'wpc_checkbox', array(
        'type'           => 'theme_mod',
        'default'        => 'enable',
        'transport'      => 'refresh', // Options: refresh or postMessage.
        'capability'     => 'edit_theme_options',
        'sanitize_callback' => 'wpc_sanitize_checkbox' // Custom function in
customizer-sanitization.php file.
    ) );

    // Control: Checkbox.
    $wp_customize->add_control( 'wpc_checkbox', array(
        'label'           =>__( 'Checkbox', 'WPC' ),
        'description'    =>__( 'Description', 'WPC' ),
        'section'        => 'wpc_section_basic',
        'type'           => 'checkbox'
    ) );

    // Setting: Radio.
    $wp_customize->add_setting( 'wpc_radio', array(
        'type'           => 'theme_mod',
        'default'        => 'on',
        'transport'      => 'refresh', // Options: refresh or postMessage.
        'capability'     => 'edit_theme_options',
        'sanitize_callback' => 'wpc_sanitize_select', // Custom function in customizer-
sanitization.php file.
    ) );

    // Control: Radio.
    $wp_customize->add_control( 'wpc_radio', array(
        'label'           =>__( 'Radio', 'WPC' ),
        'description'    =>__( 'Description', 'WPC' ),
        'section'        => 'wpc_section_basic',
        'type'           => 'radio',
        'choices'        => array(
            'enable' => 'Enable',
            'disable' => 'Disable'
        )
    ) );

    // Setting: Select.
    $wp_customize->add_setting( 'wpc_select', array(
        'type'           => 'theme_mod',
        'default'        => 'enable',
        'transport'      => 'refresh', // Options: refresh or postMessage.
        'capability'     => 'edit_theme_options',
        'sanitize_callback' => 'wpc_sanitize_select' // Custom function in customizer-
sanitization.php file.
    ) );

```

```
) );  
  
// Control: Select.  
$wp_customize->add_control( 'wpc_select', array(  
    'label'      =>_( 'Select', 'WPC' ),  
    'description' =>__( 'Description', 'WPC' ),  
    'section'    => 'wpc_section_basic',  
    'type'       => 'select',  
    'choices'    => array(  
        'enable' => 'Enable',  
        'disable' => 'Disable'  
    )  
) );  
}  
}
```

# Chapter 20: Customizer Hello World

## Parameters

Parameter	Details
mytheme	A unique identifier for your theme (or child theme). This can be your theme slug

## Examples

### Hello World Example

କଷ୍ଟମାଲଜରର ମ fundamental ଲିକ ଧାରଣା ହେଉଛି ଯେ ଆଡମିନମାନେ ସେମାନଙ୍କ ସାଇଟରେ ପୂର୍ବାବଲୋକନ ପରିବର୍ତ୍ତନଗୁଡ଼ିକୁ ଜୀବନ୍ତ କରିପାରିବେ, ଏବଂ ତାପରେ ସେମାନଙ୍କୁ ସ୍ଥାୟୀ ଭାବରେ ଯୋଡ଼ିପାରିବେ |

The fundamental concept of the customizer is that admins can live preview changes to their site, and then permanently add them.

The following can be copied and pasted into a theme's `functions.php` file to

- Add a customizer section called `my first section`
- Add a customizer setting called `hello world color` allowing the admin to choose a color
- Add a css rule for `.hello-world` that will correspond with the color chosen and default to `#000000` if nothing is chosen. The setting will be put in a `<style>` tag at the end of the `<head>`.

```
function mytheme_customize_register( $wp_customize ) {

    $wp_customize->add_section( 'my_first_section_id' , array(
        'title'      =>__( 'My First Section', 'mytheme' ),
        'priority'   => 30,
    ) );

    $wp_customize->add_setting( 'hello_world_color' , array(
        'default'    => '#000000',
        'transport'  => 'refresh',
    ) );

    $wp_customize->add_control( new WP_Customize_Color_Control( $wp_customize, 'link_color',
array(
        'label'      =>__( 'Hello world color', 'mytheme' ),
        'section'    => 'my_first_section_id',
        'settings'   => 'hello_world_color',
    ) ) );

}
add_action( 'customize_register', 'mytheme_customize_register' );

function mytheme_customize_css()
{
    ?>
    <style type="text/css">
```

```
        .hello-world { color: #<?php echo get_theme_mod('hello_world_color', '000000'); ?>; }  
    </style>  
    <?php  
}  
add_action( 'wp_head', 'mytheme_customize_css');
```

# Chapter 21: Debugging

## Introduction

[https://codex.wordpress.org/Debugging\\_in\\_WordPress](https://codex.wordpress.org/Debugging_in_WordPress)

PHP କୋଡ୍ ଡିବଗ୍ କରିବା ଯେକ *any* ଶସି ପ୍ରକଳ୍ପର ଅଂଶ, କିନ୍ତୁ ଓଡ଼ିଏସ୍ ନିର୍ଦ୍ଦିଷ୍ଟ ଡିବଗ୍ ସିଷ୍ଟମ୍ ସହିତ ଆସିଥାଏ ଯାହା ପ୍ରକ୍ରିୟାକୁ ସରଳ କରିବା ସହିତ କୋର୍, ପ୍ଲଗିନ୍ ଏବଂ ଥିମ୍ ମଧ୍ୟରେ କୋଡ୍ ମାନକ କରିବା ପାଇଁ ପରିକଳ୍ପିତ |(Debugging PHP code is part of any project, but WordPress comes with specific debug systems designed to simplify the process as well as standardize code across the core, plugins and themes.)

## Remarks

Plugins for debugging in WordPress:

- <https://wordpress.org/plugins/query-monitor/>
- <https://wordpress.org/plugins/debug-bar/>
- <https://wordpress.org/plugins/debug-bar-console/>
- <https://wordpress.org/plugins/kint-debugger/>
- <https://wordpress.org/plugins/rest-api-console/>

## Examples

### WP\_DEBUG

`WP_DEBUG` is a PHP constant (a permanent global variable) that can be used to trigger the "debug" mode throughout WordPress. It is assumed to be false by default and is usually set to true in the `wp-config.php` file on development copies of WordPress.

```
define( 'WP_DEBUG', true );  
define( 'WP_DEBUG', false );
```

### WP\_DEBUG\_LOG

`WP_DEBUG_LOG` is a companion to `WP_DEBUG` that causes all errors to also be saved to a `debug.log` log file inside the `/wp-content/` directory. This is useful if you want to review all notices later or need to view notices generated off-screen (e.g. during an AJAX request or `wp-cron` run).

```
//enable  
define( 'WP_DEBUG_LOG', true );  
  
//disable  
define( 'WP_DEBUG_LOG', false );
```

### WP\_DEBUG\_DISPLAY

`WP_DEBUG_DISPLAY` is another companion to `WP_DEBUG` that controls whether debug messages are

shown inside the HTML of pages or not. The default is 'true' which shows errors and warnings as they are generated. Setting this to false will hide all errors. This should be used in conjunction with WP\_DEBUG\_LOG so that errors can be reviewed later. Note: for WP\_DEBUG\_DISPLAY to do anything, WP\_DEBUG must be enabled (true).

```
//enable
define( 'WP_DEBUG_DISPLAY', true );

//disable
define( 'WP_DEBUG_DISPLAY', false );
```

## SCRIPT\_DEBUG

**SCRIPT\_DEBUG** is a related constant that will force WordPress to use the "dev" versions of core CSS and JavaScript files rather than the minified versions that are normally loaded. This is useful when you are testing modifications to any built-in .js or .css files. Default is false.

**SCRIPT\_DEBUG** ହେଉଛି ଏକ ସମ୍ବନ୍ଧୀୟ ସ୍ଥିର ଯାହାକି ସାଧାରଣତଃ load ଲୋଡ୍ ହୋଇଥିବା କ୍ଷୁଦ୍ର ସଂସ୍କରଣ ଅପେକ୍ଷା କୋର୍ଡ୍ CSS ଏବଂ ଜାଭାସ୍କ୍ରିପ୍ଟ ଫାଇଲଗୁଡ଼ିକର "dev" ସଂସ୍କରଣ ବ୍ୟବହାର କରିବାକୁ ଓଡ଼ିପ୍ରେସ୍ କୁ ବାଧ୍ୟ କରିବ । ଯେତେବେଳେ ଆପଣ ଯେକି any ଶିସି ବିଲ୍ଡ-ଇନ୍ .js କିମ୍ବା .css ଫାଇଲଗୁଡ଼ିକରେ ପରିବର୍ତ୍ତନଗୁଡ଼ିକ ପରୀକ୍ଷା କରୁଛନ୍ତି, ଏହା ଉପଯୋଗୀ । ଡିଫଲ୍ଟ ମିଥ୍ୟା ଅଟେ ।)

```
//enable
define( 'SCRIPT_DEBUG', true );

//disable
define( 'SCRIPT_DEBUG', false );
```

## SAVEQUERIES

(The **SAVEQUERIES** definition saves the database queries to an array and that array can be displayed to help analyze those queries. The constant defined as true causes each query to be saved, how long that query took to execute, and what function called it. NOTE: This will have a performance impact on your site, so make sure to turn this off when you aren't debugging. )**SAVEQUERIES** ସଂଜ୍ଞା ତାତାବେସ୍ ଜିଜ୍ଞାସାଗୁଡ଼ିକୁ ଏକ ଆରେରେ ସଂରକ୍ଷଣ କରେ ଏବଂ ସେହି ପ୍ରଶ୍ନଗୁଡ଼ିକୁ ବିଶ୍ଳେଷଣ କରିବାରେ ସାହାଯ୍ୟ କରିବାକୁ ସେହି ଆରେ ପ୍ରଦର୍ଶିତ ହୋଇପାରିବ । ସତ୍ୟ ଭାବରେ ବ୍ୟାଖ୍ୟା କରାଯାଇଥିବା ପ୍ରତ୍ୟେକ ଜିଜ୍ଞାସା ସଞ୍ଚୟ ହୁଏ, ସେହି ଜିଜ୍ଞାସା କେତେ ସମୟ କାର୍ଯ୍ୟକାରୀ କରିବାକୁ ଲାଗିଲା, ଏବଂ କେଉଁ କାର୍ଯ୍ୟ ଏହାକୁ ଡାକିଲା । ଚିତ୍ରଣୀ: ଏହା ଆପଣଙ୍କ ସାଇଟରେ ଏକ କାର୍ଯ୍ୟଦକ୍ଷତା ପ୍ରଭାବ ପକାଇବ, ତେଣୁ ଯେତେବେଳେ ଆପଣ ତ୍ରୁଟି ନିବାରଣ କରୁନାହାଁନ୍ତି ଏହାକୁ ବନ୍ଦ କରିବାକୁ ନିଶ୍ଚିତ କରନ୍ତୁ ।

```
define( 'SAVEQUERIES', true );
```

The array is stored in the

```
global $wpdb->queries;
```

## Example wp-config.php and good practices for Debugging

The following code, inserted in your wp-config.php file, will log all errors, notices, and warnings to a file called debug.log in the wp-content directory. It will also hide the errors so they do not interrupt page generation. ନିମ୍ନଲିଖିତ କୋଡ୍, ଆପଣଙ୍କର wp-config.php ଫାଇଲରେ ସଂକ୍ଷିପ୍ତ ହୋଇଛି, wp- ବିଷୟବସ୍ତୁ ଡିରେକ୍ଟୋରୀରେ debug.log ନାମକ ଏକ ଫାଇଲରେ ସମସ୍ତ ତ୍ରୁଟି, ନୋଟିସ୍ ଏବଂ ଚେତାବନୀ ଲଗ୍ କରିବ । ଏହା ତ୍ରୁଟିଗୁଡ଼ିକୁ ମଧ୍ୟ ଲୁଚାଇବ ଯାହା ଦ୍ୱାରା page ାରା ସେମାନେ ପୃଷ୍ଠା ସୃଷ୍ଟିରେ ବାଧା ସୃଷ୍ଟି କରିବେ ନାହିଁ ।

```
// Enable WP_DEBUG mode
define( 'WP_DEBUG', true );

// Enable Debug logging to the /wp-content/debug.log file
define( 'WP_DEBUG_LOG', true );

// Disable display of errors and warnings
```

```
define( 'WP_DEBUG_DISPLAY', false );
@ini_set( 'display_errors', 0 );

// Use dev versions of core JS and CSS files (only needed if you are modifying these core files)
define( 'SCRIPT_DEBUG', true );
```

```
//Checking is function_exists
if ( !function_exists( 'print_to_log' ) ) {
    //function writes a message to debug.log if debugging is turned on.
    function print_to_log( $message )
    {
        if ( true === WP_DEBUG ) {
            if ( is_array( $message ) || is_object( $message ) ) {
                error_log( print_r( $message, true ) );
            } else {
                error_log( $message );
            }
        }
    }
}
```

Good practice If you want add custom messages to debug log add following code in your plugin or theme. ( ଭଲ ଅଭ୍ୟାସ ଯଦି ଆପଣ ଡିବଗ୍ ଲଗ୍ ରେ କଷ୍ଟମ୍ ମେସେଜ୍ ଯୋଡ଼ିବାକୁ ଚାହୁଁଛନ୍ତି ତେବେ ଆପଣଙ୍କର ପ୍ଲଗିନ୍ କିମ୍ବା ଥିମ୍ ରେ ନିମ୍ନ କୋଡ୍ ଯୋଡ଼ନ୍ତୁ ।)

## See logs in a separate file

When you have an ajax call, it's extremely difficult to get a log from inside of the callback function. But if you enable the debugging (ଯେତେବେଳେ ତୁମର ଆଜାକ୍ସ କଲ୍ ଆସି, କଲବ୍ୟାକ୍ ଫଙ୍କସନ୍ ଭିତରୁ ଏକ ଲଗ୍ ପାଇବା ଅତ୍ୟନ୍ତ କଷ୍ଟକର । କିନ୍ତୁ ଯଦି ଆପଣ ତ୍ରୁଟି ନିବାରଣକୁ ସକ୍ଷମ କରନ୍ତି ।)

```
define('WP_DEBUG', true);
```

and then after that add

```
ini_set('log_errors',TRUE);
ini_set('error_reporting', E_ALL);
ini_set('error_log', dirname(__FILE__) . '/error_log.txt');
```

you will have an error.log.txt file in your root folder where all your logs are located. you can even log them with

```
error_log( print_r( 'what I want to check goes here', true ) );
```

inside your code. This will make your life a lot easier.

# Chapter 22: Enqueuing scripts

## Syntax

- `wp_enqueue_script( $handle, $src, $deps, $ver, $in_footer)`

## Parameters

Parameter	Details
<code>\$handle</code>	<i>(string)</i> (Required) Name of the script. Should be unique.
<code>\$src</code>	<i>(string)</i> (Optional) Full URL of the script, or path of the script relative to the WordPress root directory. <i>Default value: false</i>
<code>\$deps</code>	<i>(array)</i> (Optional) An array of registered script handles this script depends on. <i>Default value: array()</i>
<code>\$ver</code>	<i>(string   bool   null)</i> (Optional) String specifying script version number, if it has one, which is added to the URL as a query string for cache busting purposes. If version is set to false, a version number is automatically added equal to current installed WordPress version. If set to null, no version is added. <i>Default value: false</i>
<code>\$in_footer</code>	<i>(bool)</i> (Optional) Whether to enqueue the script before <code>&lt;/body&gt;</code> instead of in the <code>&lt;head&gt;</code> . <i>Default value: false</i>

## Examples

### Enqueuing scripts in functions.php

If you want to add `custom.js` script that is located in the `js/` folder of your theme, you'll need to enqueue it. In `functions.php` add

```
<?php
add_action( 'after_setup_theme', 'yourtheme_theme_setup' );

if ( ! function_exists( 'yourtheme_theme_setup' ) ) {
    function yourtheme_theme_setup() {

        add_action( 'wp_enqueue_scripts', 'yourtheme_scripts' );
        add_action( 'admin_enqueue_scripts', 'yourtheme_admin_scripts' );

    }
}
```

```

if ( ! function_exists( 'yourtheme_scripts' ) ) {
    function yourtheme_scripts() {

        wp_enqueue_script( 'yourtheme_custom', get_template_directory_uri().'/js/custom.js',
array( 'jquery' ), '1.0.0', true );

    }
}

if ( ! function_exists( 'yourtheme_admin_scripts' ) ) {
    function yourtheme_admin_scripts() {

        wp_enqueue_script( 'yourtheme_custom', get_template_directory_uri().'/js/custom.js',
array( 'jquery-ui-autocomplete', 'jquery' ), '1.0.0', true );

    }
}

```

## Enqueue scripts for IE only

```

add_action( 'wp_enqueue_scripts', 'enqueue_my_styles_and_scripts' );

/**
 * Enqueue scripts (or styles) conditionally.
 *
 * Load scripts (or stylesheets) specifically for IE. IE10 and above does
 * not support conditional comments in standards mode.
 *
 * @link https://gist.github.com/wpscholar/4947518
 * @link https://msdn.microsoft.com/en-us/library/ms537512(v=vs.85).aspx
 */
function enqueue_my_styles_and_scripts() {

    // Internet Explorer HTML5 support
    wp_enqueue_script( 'html5shiv', get_template_directory_uri().'/js/html5shiv.js', array(),
'3.7.3', false);
    wp_script_add_data( 'html5shiv', 'conditional', 'lt IE 9' );

    // Internet Explorer 8 media query support
    wp_enqueue_script( 'respond', get_template_directory_uri().'/js/respond.js', array(),
'1.4.2', false);
    wp_script_add_data( 'respond', 'conditional', 'lt IE 9' );

}

```

## Enqueuing Scripts conditionally for specific pages

You can use conditional operators in WordPress to enqueue scripts on specific pages of your Website.

```

function load_script_for_single_post(){
    if(is_single()){
        wp_enqueue_script(
            'some',
            get_template_directory_uri().'/js/some.js',
            array('jquery'),
            '1.0.0',

```

```
        );  
        }  
    }  
    add_action('wp_enqueue_scripts', 'load_script_for_single_post');
```

In the above example, if the current webpage is single post, script will be enqueued. Otherwise *wp\_enqueue\_script* function will not be executed.

---

# Chapter 23: Enqueuing Styles

## Syntax

1. `wp_enqueue_style($handle, $src, $dependency, $version, $media);`

## Parameters

Parameter	Details
<code>\$handle</code>	(String) (Required) Unique name for the stylesheet.
<code>\$src</code>	(String) (Optional) URL of stylesheet which will be used inside <b>link</b> tag's <code>src</code> attribute.
<code>\$deps</code>	(String) (Optional) An array of stylesheet handles this stylesheet depends on.
<code>\$ver</code>	(String) (Optional) String specifying stylesheet version of stylesheet.
<code>\$media</code>	(String) (Optional) The media for which this stylesheet is created. e.g 'all', 'print', 'screen' etc

## Examples

### Including internal css file with another css file as a dependency

```
function themeslug_enqueue_scripts() {
    wp_enqueue_style( 'themeslug-reset', get_template_directory_uri() .'/css/reset.css',
    '1.0.0' );
    wp_enqueue_style( 'themeslug-style', get_template_directory_uri() .'/style.css',
    'themeslug-reset', '1.0.0');
}
add_action('wp_enqueue_scripts', 'themeslug_enqueue_scripts');
```

### Including internal css file

In this case `style.css` is located in root of the theme's folder

```
function themeslug_enqueue_scripts() {
    wp_enqueue_style( 'themeslug-style', get_template_directory_uri() .'/style.css', '1.0.0');
}
add_action('wp_enqueue_scripts', 'themeslug_enqueue_scripts');
```

### Including external css file

In this example we want to include font awesome icon font

```
function themeslug_enqueue_scripts() {
    wp_enqueue_style( 'font-awesome', '//cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.6.3/css/font-awesome.css');
}
add_action('wp_enqueue_scripts', 'themeslug_enqueue_scripts');
```

## Enqueue stylesheets for IE only

```
add_action( 'wp_enqueue_scripts', 'enqueue_my_styles_and_scripts' );

/**
 * Enqueue styles (or scripts) conditionally.
 *
 * Load stylesheets (or scripts) specifically for IE. IE10 and above does
 * not support conditional comments in standards mode.
 *
 * @link https://gist.github.com/wpscholar/4947518
 * @link https://msdn.microsoft.com/en-us/library/ms537512(v=vs.85).aspx
 */
function enqueue_my_styles_and_scripts() {

    // Internet Explorer specific stylesheet.
    wp_enqueue_style( 'themename-ie', get_stylesheet_directory_uri() . '/css/ie.css', array(
'twentyfifteen-style' ), '20141010' );
    wp_style_add_data( 'themename-ie', 'conditional', 'lte IE 9' );

    // Internet Explorer 7 specific stylesheet.
    wp_enqueue_style( 'themename-ie7', get_stylesheet_directory_uri() . '/css/ie7.css', array(
'twentyfifteen-style' ), '20141010' );
    wp_style_add_data( 'themename-ie7', 'conditional', 'lt IE 8' );

}

```

## Including internal css file for your Plugin class

```
class My_Plugin() {
    function __construct() {
        add_action( 'wp_enqueue_scripts', array( $this, 'init_fe_assets' ) );
    }

    public function init_fe_assests() {
        wp_enqueue_style( 'my-plugin', plugin_dir_url( __FILE__ ) .
'assets/css/frontend/plugin.css', array(), '0.0.1', true );
    }
}

new My_Plugin();
```

## Add Alternative Stylesheets

```
<?php wp_enqueue_style('theme-five', get_template_directory_uri() .
'/path/to/additional/css');
wp_style_add_data('theme-five', 'alt', true);
wp_style_add_data('theme-five', 'title', __('theme-five.css', 'your-theme-name')); ?>
```



---

# Chapter 24: Function : wp\_trim\_words()

## Syntax

- `<?php $trimmed_text = wp_trim_words( $text, $num_words = 55, $more = null ); ?>`

## Parameters

Parameter	Details
\$text	(String) (Required) Text that will be shortened or trimmed.
\$num_words	(Integer) (Required) Number of words to which text will be restricted.
\$more	(String) (Optional) What to append if \$text needs to be trimmed.

## Examples

### Trimming post content

This function shortens the text to a specified number of words and returns the shortened text.

```
<?php echo wp_trim_words( get_the_content(), 40, '...' ); ?>
```

In the above example we are passing the post content to the function. It will restrict the length of the content to 40 words and will trim rest of the words.

(ଉପରୋକ୍ତ ଉଦାହରଣରେ ଆମେ ପୋଷ୍ଟ ବିଷୟବସ୍ତୁକୁ ଫଙ୍କସନ୍ କୁ ପଠାଉଛୁ । ଏହା ବିଷୟବସ୍ତୁର ଦ length ଧ୍ୟକୁ 40 ଶବ୍ଦରେ ସୀମିତ କରିବ ଏବଂ ବାକି ଶବ୍ଦଗୁଡ଼ିକୁ କ୍ଷେପନ କରିବ ।)

# Chapter 25: Function: add\_action()

## Syntax

- add\_action( \$tag, \$function\_to\_add )
- add\_action( \$tag, \$function\_to\_add, \$priority )
- add\_action( \$tag, \$function\_to\_add, \$priority, \$accepted\_args )

## Parameters

Parameter	Details
\$tag	(string) (Required) The name of the action to which the \$function_to_add is hooked
\$function_to_add	(callable) (Required) The function which should be called when the action indicated by \$tag is executed
\$priority	(int) (Optional) Default value: 10 Used to specify the order in which the functions associated with a particular action are executed. Lower numbers correspond with earlier execution, and functions with the same priority are executed in the order in which they were added to the action.
\$accepted_args	(int) (Optional) Default value: 1 The number of arguments the function accepts.

## Remarks

Add\_action () ଫଙ୍କସନ୍ ଏକ ଆକ୍ସନ୍ ହୁଏ ସୃଷ୍ଟି କରେ, ଏକ PHP ଫଙ୍କସନ୍ କୁ ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ କ୍ରିୟା “ଟ୍ୟାଗ” କିମ୍ବା ନାମ ସହିତ ଯୋଡ଼ିଥାଏ । ଯେତେବେଳେ ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ଟ୍ୟାଗ୍ ସହିତ do\_action () (କିମ୍ବା do\_action\_ref\_array ()) କୁ କଲ୍ ଦିଆଯାଏ କିମ୍ବା “ଟ୍ରିଗର” ହୁଏ, ସେହି ଟ୍ୟାଗ୍ ସହିତ “ହୁକ୍” ହୋଇଥିବା ସମସ୍ତ କାର୍ଯ୍ୟ ଏକତ୍ରେ କରାଯାଏ । (The add\_action() function creates an Action Hook, associating a PHP function with a particular action "tag" or name. When the action is "triggered" by a call to do\_action() (or do\_action\_ref\_array()) with a specific tag, all functions "hooked" to that tag will be executed.)

In most cases, this function should be used in a theme's functions.php file or a plugin file - or another source file loaded by either.

This function is a part of the **Plugin API**

## Examples

### Basic Action Hook

The most basic application of add\_action() is to add custom code to be executed at a certain location in a WordPress installation's source-code - either using actions supplied by the **Core** of WordPress, or ones created by third-party code such as plugins and themes.

Add\_action () ର ସବୁଠାରୁ ମୂଳିକ ଲିକ ପ୍ରୟୋଗ ହେଉଛି ଏକ ଉପଯୋଗୀ ସଂସ୍ଥାର ଉପ-କୋଡ଼ରେ ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ସ୍ଥାନରେ ଏକତ୍ରେ କରାଯାଏ କିମ୍ବା କଣ୍ଟ୍ରୋଲ କୋଡ଼ ଯୋଡ଼ିବା - ଉପଯୋଗୀ କୋର ଦ୍ୱାରା ପ୍ରଦତ୍ତ କ୍ରିୟାଗୁଡ଼ିକୁ ବ୍ୟବହାର କରି, କିମ୍ବା ପ୍ଲଗିନ୍ ପରି ତୃତୀୟ-ପକ୍ଷ କୋଡ଼ ଦ୍ୱାରା ସୃଷ୍ଟି ହୋଇଥିବା କାର୍ଯ୍ୟଗୁଡ଼ିକ ବ୍ୟବହାର କରି । ଥିମ୍

To add content to the `<head></head>` section of the site - say to add a `<link>` meta element to indicate where copyright information for the site can be found - `add_action()` can be used to attach a function that prints the appropriate markup to the `'wp_head'` action (which "triggers" when WordPress builds the `<head>` section):

```
function add_copyright_meta_link() {
    echo( '<link rel="copyright" href="' . get_home_url() . '/copyright">' );
}

add_action( 'wp_head', 'add_copyright_meta_link' );
```

## Action Hook Priority

Any number of functions may be "hooked" to any given action. In some instances it is important for a hooked function to execute before or after others, which is where the third parameter to `add_action()`, `$priority` comes into play.

If the `$priority` argument is omitted, the function will be attached with the default priority of 10. When the action is "triggered", the "hooked" functions will be called starting with those added with the smallest `$priority`, and progressing to the functions with the largest `$priority`. Any hooked functions that share the same priority will be called in the order that they were added (the order in which their respective `add_action()` calls were executed).

For instance, say a third-party plugin is using a function hooked to the `'template_redirect'` action in order to forward visitors to the `daily-deal` page to an affiliate link for an external e-commerce site, but you'd like the redirection to only occur for logged-in users. You would need to use your own `'template_redirect'` hook to send logged-out visitors to the sign-in page. After determining that the third-party plugin attaches its function with the default `$priority` of 10, you could hook your function with a priority of 9 to ensure that your logged-in check happens first:

```
function redirect_deal_visitors_to_login() {
    if( is_page( 'daily-deal' ) && !user_is_logged_in() ) {
        wp_redirect( wp_login_url() );
        exit();
    }
}

add_action( 'template_redirect', 'redirect_deal_visitors_to_login', 9 );
```

## Hooking Class & Object Methods to Actions

[PHP Classes](#) are a powerful tool for improving code organization and minimizing naming collisions. At some point or another, the question of how to create an action hook for a class method inevitably arises.

The `$function_to_add` argument is often shown as a string containing the function's name, however the data-type of the argument is actually a "[callable](#)", which for our purposes can be summed up as "a reference to a function or method".

There are a number of callable formats that can be used to reference methods on classes and

objects. In all cases however, the referenced method *must* be publicly [visible](#). A method is public when it is either prefixed with the `public` keyword, or no visibility keyword at all (in which case the method defaults to public).

---

## Object Method Action Hooks

Object methods are executed on a particular instance of a class.

```
class My_Class {
    // Constructor
    function My_Class() {
        // (Instantiation logic)
    }

    // Initialization function
    public function initialize() {
        // (Initialization logic)
    }
}
```

After instantiating the above class as follows,

```
$my_class_instance = new My_Class();
```

the `initialize()` method would normally be invoked on the object by calling `$my_class_instance->initialize()`. Hooking the method to the 'init' WordPress action is done by passing an array containing a reference to the instance and a string containing the object method's name:

```
add_action( 'init', [ $my_class_instance, 'initialize' ] );
```

If `add_action()` is called within an object method, the `$this` pseudo-variable can also be used:

```
class My_Class {
    // Constructor
    function My_Class() {
        // (Instantiation logic)
        add_action( 'init', [ $this, 'initialize' ] );
    }

    // Initialization function
    public function initialize() {
        // (Initialization logic)
    }
}
```

---

## Class Method Action Hooks

Class methods are executed statically on a class rather than any particular instance. Given the

following class,

```
class My_Class {
    // Initialization function
    public static function initialize() {
        // (Initialization logic)
    }
}
```

the `initialize()` method would normally be invoked by using the `::` scope-resolution operator, i.e. `My_Class::initialize()`;. Hooking a static class method to a WordPress can be done in a couple different ways:

- Using an array composed of a string containing the class name, and a string containing the method name:

```
add_action( 'init', [ 'My_Class', 'initialize' ] );
```

- Passing a string containing a full reference to the method, including the `::` operator:

```
add_action( 'init', 'My_Class::initialize' );
```

- If `add_action()` is called within a static class method, the `self` keyword or the `__CLASS__` magic-constant can be used in place of the class name. Note that this is generally inadvisable as the values of these items become somewhat counter-intuitive in the case of class inheritance.

```
class My_Class {
    // Setup function
    public static function setup_actions() {
        add_action( 'init', 'self::initialize' );
    }

    // Initialization function
    public static function initialize() {
        // (Initialization logic)
    }
}
```

# Chapter 26: get\_bloginfo()

## Introduction

Retrieves information about the current site.

## Syntax

- `get_bloginfo( $show , $filter )`

## Parameters

Parameter	Details
<code>\$show</code>	(string) The site setting information to retrieve.
<code>\$filter</code>	(string) The specification on whether to return a filtered value or not.

## Remarks

### `$show`

Values	Description	Example
'name' (Default)	Site title	'Matt Mullenweg'
'description'	Site tagline	'Just another WordPress site'
'wpurl'	URL of the WordPress installation. Same as the <code>site_url()</code> function	'http://example.com' , 'http://localhost/wordpress'
'url'	URL of the site. Same as the <code>home_url()</code> function	'http://example.com' , 'http://localhost/wordpress'
'admin_email'	Email address of the main Administrator	'matt@mullenweg.com'
'charset'	Character encoding of the pages and feeds	'UTF-8'
'version'	Current version of the WordPress installation	'4.5'
'html_type'	content-type value of the	'text/html'

Values	Description	Example
	HTML	
'text_direction'	Text direction determined by the site's language	'ltr'
'language'	ISO 639-1 based language code	'en-us'
'stylesheet_url'	URL of the stylesheet of the activated theme. Value priority: <b>Child theme</b> » Parent theme.	'http://example.com/wp-content/themes/twenty-sixteen/style.css'
'stylesheet_directory'	Resource location of the activated theme. Value priority: <b>Child theme</b> » Parent theme.	'http://example.com/wp-content/themes/twenty-sixteen'
'template_url'	URL directory of the activated theme. Value priority: <b>Parent theme</b> » Child theme.	'http://example.com/wp-content/themes/twenty-sixteen'
'template_directory'	Same as 'template_url'	
'pingback_url'	Pingback XML-RPC file	'http://example/xmlrpc.php'
'atom_url'	Atom feed URL	'http://example/feed/atom/'
'rdf_url'	RDF/RSS 1.0 feed URL	'http://example/feed/rdf/'
'rss_url'	RSS 0.92 feed URL	'http://example/feed/rss/'
'rss2_url'	RSS 2.0 feed URL	'http://example/feed/'
'comments_atom_url'	Comments Atom feed URL	'http://example/comments/feed/atom/'
'siteurl'	<i>(deprecated)</i> Use 'url' instead	
'home'	<i>(deprecated)</i> Use 'url' instead	

## \$filter

Values	Description	Example
'raw' (Default)	No filters will be applied	<i>raw data</i>
'display'	Filters will be applied to the return value if \$show is neither 'url' , 'directory' , 'home'	<i>filtered data</i>

## Examples

### Getting the site title

```
<?php echo get_bloginfo( 'name' ); ?>
```

or

```
<?php echo get_bloginfo(); ?>
```

### Output

```
Matt Mullenweg
```

Based on these sample settings

WordPress dashboard header: Matt Mullenweg 0 + New

Left sidebar menu: Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, **Settings**, General, Writing, Reading, Discussion, Media, Permalinks, Collapse menu

## General Settings

Site Title: Matt Mullenweg

Tagline: Just another WordPress site  
*In a few words, explain what this site is about.*

WordPress Address (URL): http://localhost/wordpress

Site Address (URL): http://localhost/wordpress  
*Enter the address here if you [want your site home page](#)*

Email Address: matt@mullenweg.com  
*This address is used for admin purposes, like new user r*

Membership:  Anyone can register

New User Default Role: Subscriber

Timezone: UTC+0

## Getting the site tagline

```
<?php echo get_bloginfo( 'description' ); ?>
```

## Output

```
Just another WordPress site
```

Based on these sample settings

The screenshot shows the WordPress administration interface. The top navigation bar includes the WordPress logo, a home icon, the user name 'Matt Mullenweg', a comment count of '0', and a '+ New' button. The left sidebar contains a menu with items: Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings (highlighted in blue), General, Writing, Reading, Discussion, Media, Permalinks, and Collapse menu. The main content area is titled 'General Settings' and contains the following fields:

- Site Title: Matt Mullenweg
- Tagline: Just another WordPress site  
*In a few words, explain what this site is about.*
- WordPress Address (URL): http://localhost/wordpress
- Site Address (URL): http://localhost/wordpress  
*Enter the address here if you [want your site home page](#)*
- Email Address: matt@mullenweg.com  
*This address is used for admin purposes, like new user r*
- Membership:  Anyone can register
- New User Default Role: Subscriber
- Timezone: UTC+0

## Getting the active theme URL

```
<?php echo esc_url( get_bloginfo( 'stylesheet_directory' ) ); ?>
```

### Output

```
http://example.com/wp-content/themes/twenty十六teen
```

### Alternatives

Internally, `get_bloginfo( 'stylesheet_directory' )` calls `get_stylesheet_directory_uri()`, so you may want to use that instead:

```
<?php echo esc_url( get_stylesheet_directory_uri() ); ?>
```

Many developers prefer to use these dedicated functions because of inconsistent naming conventions between them and `get_bloginfo()`. For example, `get_stylesheet_directory()` returns the child theme path; however, as our previous example illustrates, `get_bloginfo('stylesheet_directory')` returns the child theme URL. If you use `get_stylesheet_directory_uri()` instead, there's less chance of confusion over whether you're retrieving a path or a URL.

## Get site url

```
<?php echo esc_url(get_bloginfo('url')); ?>
```

or if you needed to link to a sub page

```
<?php echo esc_url(get_bloginfo('url') . '/some-sub-page'); ?>
```

## Get Email Address of Site Administrator

We can use the `get_bloginfo` function to retrieve the email address of the site administrator.

```
<?php echo get_bloginfo('admin_email'); ?>
```

---

# Chapter 27: get\_home\_path()

## Introduction

Get the absolute filesystem path to the root of the WordPress installation.

## Parameters

Parameter	Details
<i>None</i>	This function does not accept any parameters.

## Remarks

### Important difference between `get_home_path()` and `ABSPATH`

Please keep in mind the difference between `ABSPATH` and `get_home_path()` if you have WordPress installed in a subfolder.

The `get_home_path()` function will always return a path **without** the subfolder:

- <http://www.example.com> - `/var/www/html/example`
- <http://www.example.com/wp> - `/var/www/html/example`

This is how it differs from `ABSPATH`, which will return different values:

- <http://www.example.com> - `/var/www/html/example`
- <http://www.example.com/wp> - `/var/www/html/example/wp`

`ABSPATH` is first defined in `wp-load.php` which will be located at `/var/www/html/example/wp/wp-load.php` hence this is where `ABSPATH` will take its definition from.

`get_home_path()` checks if the `site_url` and `home_url` differ, and removes the substring from the path. Otherwise it returns `ABSPATH` value:

```
function get_home_path() {
    $home      = set_url_scheme( get_option( 'home' ), 'http' );
    $siteurl   = set_url_scheme( get_option( 'siteurl' ), 'http' );
    if ( ! empty( $home ) && 0 !== strcasecmp( $home, $siteurl ) ) {
        $wp_path_rel_to_home = str_ireplace( $home, '', $siteurl ); /* $siteurl - $home */
        $pos = stripos( str_replace( '\\', '/', $_SERVER['SCRIPT_FILENAME'] ),
trailing_slashit( $wp_path_rel_to_home ) );
        $home_path = substr( $_SERVER['SCRIPT_FILENAME'], 0, $pos );
        $home_path = trailing_slashit( $home_path );
    } else {
        $home_path = ABSPATH;
    }
}
```

```
return str_replace( '\\', '/', $home_path );  
}
```

## Using it in your code

Calling `get_home_path()` must be done in a context where `wp-admin/includes/file.php` has already been included.

For example using `get_home_path()` within the `admin_init` hook is fine, but using it within the `init` is not and will result in a PHP fatal error:

```
Call to undefined function get_home_path()
```

This file only gets included from within the admin (dashboard) context, if you absolutely need it outside of this context you will need to include the file yourself before calling the function:

```
require_once(ABSPATH . 'wp-admin/includes/file.php');
```

## Examples

### Usage

```
$path = get_home_path();
```

### Return value:

string

Full filesystem path to the root of the WordPress installation, even if it's installed in a subfolder.

### Example:

```
/var/www/htdocs/example
```

---

# Chapter 28: get\_option()

## Introduction

Retrieves an option value based on an option name.

## Syntax

- `get_option( $option, $default )`

## Parameters

Parameter	Details
<code>\$option</code>	(string) Name of option to retrieve. Expected to not be SQL-escaped.
<code>\$default</code>	(mixed) (Optional) Default value to return if the option does not exist.

## Remarks

List of arguments for `$option`

- 'admin\_email'
- 'blogname'
- 'blogdescription'
- 'blog\_charset'
- 'date\_format'
- 'default\_category'
- 'home'
- 'siteurl'
- 'template'
- 'start\_of\_week'
- 'upload\_path'
- 'users\_can\_register'
- 'posts\_per\_page'
- 'posts\_per\_rss'

## Examples

Show blog title

Code

```
<h1><?php echo get_option( 'blogname' ); ?></h1>
```

## Output

---

# Name of the blog in H1 style

## Show Character Set

### Code

```
<p><?php echo esc_html( sprintf(__( 'Character set: %s', 'textdomain' ), get_option( 'blog_charset' ) ) ); ?></p>
```

## Output

Character set: UTF-8

## Handling non-existing options

### Code

```
<?php echo get_option( 'something_bla_bla_bla' ); ?>
```

## Output

false

---

### Code

```
<?php echo get_option( 'something_bla_bla_bla', 'Oh no!' ); ?>
```

## Output

Oh no!

---

# Chapter 29: get\_permalink()

## Introduction

This function returns the full paralink of the current post or the designated post.

## Syntax

- `get_permalink( $post, $leavename )`

## Parameters

Parameter	Details
<code>\$post</code>	(int) (optional) Post ID or post object. Default is the the current post's id.
<code>\$leavename</code>	(bool) (optional) Whether to keep post name or page name.

## Remarks

For the parameter `$leavename`, it is false by default.

## Examples

### Simple use of get\_parmalink

#### Code

```
echo get_permalink();
```

#### Output

The link of the current page, for example: <http://website.com/category/name-of-post/>

### Specifying the post to get the link

#### Code

```
echo get_permalink( 44 );
```

#### Output

The link of the post id:44, for example: <http://website.com/category/name-of-post/>

## Get the link without the post's name

### Code

```
echo get_permalink( 44 , false );
```

### Output

The link of the post id:44 without the name of the post, for example:

<http://website.com/category/%postname%/>

---

# Chapter 30: get\_template\_part()

## Syntax

- `get_template_part( $slug, $name );`
- `get_template_part( $slug );`

## Parameters

Parameter	Details
<i>\$slug</i>	<i>(string)</i> Generic template slug name
<i>\$name</i>	<i>(string)</i> Specialized template name

## Examples

### Load a template part from a subfolder

```
get_template_part( 'foo/bar', 'page' );
```

will require 'bar-page.php' from the directory 'foo'.

### Get a specific file

You can get specific file using this function.

```
get_template_part('template-parts/layout');
```

Include layout.php file from template-parts subdirectory placed in the root of your theme folder.

---

# Chapter 31: get\_template\_part()

## Introduction

The purpose of this function is to standardize the way import partials or components of a theme into the main theme template. You could use a standard PHP SSI (server side includes), however, there are some benefits to using `get_template_part()`. Using this function reduces errors prone to less experienced developers trying to identify fully qualified path on the server. Also, it fails gracefully when files don't exist, and handles a custom hierarchy fallback system aka "fuzzy template searching".

## Syntax

- `get_template_part( $slug )`
- `get_template_part( $slug, $name )`

## Parameters

Parameter	Details
<code>\$slug</code>	<i>(string)</i> The slug name of the custom template.
<code>\$name</code>	<i>(string)</i> The name of the specialized template. Optional

## Examples

### Including a custom template

```
<?php get_template_part( 'foo' ); ?>
```

Includes

```
../wp-content/themes/your-theme-slug/foo.php
```

### Including a custom template with a dash-separated filename

```
<?php get_template_part( 'foo', 'bar' ); ?>
```

Includes

```
../wp-content/themes/your-theme-slug/foo-bar.php
```

## Including a custom template from inside a directory

```
<?php get_template_part( 'dir/foo' ); ?>
```

### Includes

```
../wp-content/themes/your-theme-slug/dir/foo.php
```

## Including a custom template with a dash-separated filename located inside a directory

```
<?php get_template_part( 'dir/foo', 'bar' ); ?>
```

### Includes

```
../wp-content/themes/your-theme-slug/dir/foo-bar.php
```

## Passing variable to custom template scope

```
<?php  
set_query_var( 'passed_var', $my_var );  
get_template_part( 'foo', 'bar' );  
?>
```

### Access it in foo-bar.php

```
<?php echo $passed_var; ?>
```

---

# Chapter 32: get\_template\_part()

## Syntax

- `get_template_part('file-name-no-extension');`

## Parameters

Parameter	Description
file-name-no-extension	The name of the template part with no extension. E.g. 'foo' instead of 'foo.php'

## Examples

### Loading Template Part

Pulls the code from a certain specified file into another file where the call was made.

E.g. inside example.php

```
<h1>Hello world!</h1>
```

Inside page.php

```
// header code  
get_template_part('example');  
// rest of page code
```

Output:

```
// header code  
<h1>Hello world!</h1>  
// rest of page code
```

---

# Chapter 33: `get_the_category()`

## Introduction

This function returns all categories as an array of the current post or page or the designated post or page.

## Syntax

- `get_the_category( $id )`

## Parameters

Parameter	Details
<code>\$id</code>	(int) (Optional) default to current post ID. The post ID.

## Remarks

Please note that `get_the_category()` returns an array, which means that you can't directly echo the value returned.

Here is a list of objects of each category that you can print:

- `term_id`
- `name`
- `slug`
- `term_group`
- `term_taxonomy_id`
- `taxonomy`
- `description`
- `parent`
- `count`
- `object_id`
- `filter`
- `cat_ID`
- `category_count`
- `category_description`
- `cat_name`
- `category_nicename`
- `category_parent`

## Examples

## Get all names of categories of the post

### Code

```
$categories = get_the_category();  
foreach( $categories as $category ) {  
    echo $category->name . '<br />';  
}
```

### Output

All names of categories of the current page, one on each line.

## Get all ids of categories of the post

### Code

```
$categories = get_the_category();  
foreach( $categories as $category ) {  
    echo $category->term_id . '<br />';  
}
```

### Output

All ids of categories of the current page, one on each line.

---

# Chapter 34: get\_the\_title()

## Introduction

This function returns the title of the current post or the designated post.

## Syntax

- `get_the_title( $post )`

## Parameters

Parameter	Details
<code>\$post</code>	(int) (optional) Post ID or post object. Default is the the current post's id.

## Remarks

If you plan to get the title of a post or page using a post loop, it is suggested to use `the_title()` instead.

## Examples

### Simple use of `get_the_title`

#### Code

```
get_the_title();
```

#### Output

The title of the current post or page

### Get the title of a specified post id

#### Code

```
get_the_title( 44 );
```

#### Output

The title of the post id:44.



---

# Chapter 35: home\_url()

## Syntax

- `home_url( $path, $scheme );`

## Parameters

Parameter	Details
<code>\$path</code>	<i>(String, Optional)</i> To adding more segment after the home url.
<code>\$scheme</code>	<i>(String, Optional)</i> Scheme to give the home url context. Accepts 'http', 'https', 'relative', 'rest', or null.

## Examples

### Getting the Home URL

`home_url()` is used for retrieving the current site home url.

```
<?php echo esc_url( home_url( '/' ) ) ; ?>
```

### Output

```
http://www.example.com
```

### Site url

Returns the 'site\_url' option with the appropriate protocol (<https://>)

```
<?php echo site_url(); ?>
```

### Output

```
http://www.example.com
```

---

# Chapter 36: How Can I integrate Markdown editor with Advance Custom Field's repeater Add-on.

## Examples

### Add MarkDown Editor

I found the solution. Please consider below mention steps.

Install [wp Markdown Editor](#) plugin.

Then Install "[acf-wp-wysiwyg](#)" for repeater field. Now you have to update in some files. Open this file and go to line number "180" or go to "create\_field" function add

```
echo '<script> var simplemde = new SimpleMDE({element:
document.getElementById("'.$id.'"});jQuery(".quicktags-
toolbar").css("display","none");</script>';
```

Now under "acf-repeater" plugin open "input.js" file, line number "142"

replace

```
new_field_html = this.$el.find('> table > tbody > tr.row-clone').html().replace(/(=["]*[\w-
\[\\]]*?)(acfcloneindex)/g, '$1' + new_id),
```

With

```
new_field_html = this.$el.find('> table > tbody > tr.row-clone').html().replace(/(["]*[\w-
\[\\]]*?)(acfcloneindex)/g, '$1' + new_id),
```

---

# Chapter 37: init

## Syntax

1. `add_action( 'init', callable $function )`

## Remarks

`init` is an action hook that gets fired after WordPress has finished loading but before any HTTP headers are sent.

## Examples

### Processing `$_POST` request data

```
add_action('init', 'process_post_data');
```

```
function process_post_data() {
    if( isset( $_POST ) ) {
        // process $_POST data here
    }
}
```

### Processing `$_GET` request data

```
add_action('init', 'process_get_data');

function process_get_data() {
    if( isset( $_GET ) ) {
        // process $_GET data here
    }
}
```

### Registering a custom post type

```
add_action( 'init', function() {
    register_post_type( 'event', array(
        'public' => true,
        'label' => 'Events'
    ) );
});
```

Registers a new custom post type with a label `Events` and a slug `event`

---

# Chapter 38: Installation and Configuration

## Examples

### Wordpress on LAMP

I have tested the following steps on Amazon EC2 with Ubuntu 14.04

Here is a summary of command lines to install LAMP technology stack (before installing wordpress):

#### 1: Install LAMP technology stack

##### Update linux

```
#> sudo apt-get update -> update package repositories information
```

```
#> sudo apt-get upgrade -> install package upgrades
```

##### Install apache2

```
#> sudo apt-get install apache2
```

##### Install php

```
#> sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

##### Install mysql

```
#> sudo apt-get install mysql-server php5-mysql
```

##### Install phpmyadmin

```
#> sudo apt-get install phpmyadmin
```

*In case thing went wrong, remove installed package and its configuration file(s)*

```
#> sudo apt-get purge package_name
```

Once the stack is installed here is the steps to complete wordpress installation:

#### 2: Install WordPress

1. `wget https://wordpress.org/latest.zip`
2. `sudo mkdir /var/www/wordpress`
3. Move zip file into `/var/www/wordpress` and extract.
4. `cd /etc/apache2/sites-available/`
5. `sudo cp 000-default.conf wordpress.conf`

6. Modify wordpress.conf so apache2 knows to forward any request or your domain to the wordpress app folder.
7. `sudo a2ensite wordpress.conf`
8. `sudo service apache2 restart`
9. Go to PhpMyAdmin via your browser by `yourdomain.com/phpmyadmin` . Create a new user "wordpress" and check to create the corresponding database. `cd /var/www/wordpress`
10. `sudo cp wp-config-example.php wp-config.php`
11. Modify config file to add your MySQL wordpress database information.
12. `sudo chown -R www-data:www-data /var/www/wordpress`
13. `sudo chmod -R 755 /var/www/wordpress/`
14. Open browser and type in your domain. You should see the WordPress installation page. Follow the instruction and you are done!

## Installation WP in MAMP

It's quite simple to install wordpress in MAMP.

You have to follow a few simple steps:

- 1 MAMP ରେ ଖର୍ଚ୍ଚପ୍ରେସ୍ ସଂସ୍ଥାପନ କରିବା ବହୁତ ସରଳ | ଆପଣଙ୍କୁ କିଛି ସରଳ ପଦକ୍ଷେପ ଅନୁସରଣ କରିବାକୁ ପଡ଼ିବ:
  - 1 - ଏଠାରୁ MAMP ଡାଉନଲୋଡ୍ କରନ୍ତୁ, ଏହା ମାଗଣା ଏବଂ ଆପଣ ବୋଧହୁଏ ପ୍ରୋ ଭର୍ସନ୍ ଆବଶ୍ୟକ କରନ୍ତି ନାହିଁ | 2 - ଆପଣଙ୍କର PC କିମ୍ବା Mac ରେ ସଂସ୍ଥାପନ କରନ୍ତୁ |
  - 3 - ପ୍ରୋଗ୍ରାମ୍ ଚଲାନ୍ତୁ -> ଆପଣ ଚିକିଏ ଓଷ୍ଟୋ ଖୋଲା ଦେଖିବେ, ସେଠାରୁ ଆପଣ MAMP ସେଟ୍ କରିପାରିବେ | ଏହି ମୁହୂର୍ତ୍ତରେ ସମସ୍ତ ପୂର୍ବ-ସେଟ୍ ମୂଲ୍ୟଗୁଡ଼ିକୁ ଛାଡ଼ିଦିଅ, ପ୍ରଥମ ସ୍ଥାପନ ପାଇଁ ତୁମର ଜୀବନକୁ ଜଟିଳ କରିବା ଆବଶ୍ୟକ ନାହିଁ! ଭବିଷ୍ୟତରେ ମନେରଖନ୍ତୁ MySQL ଡାଟାବେସ୍ ପାଇଁ ଆପଣଙ୍କର ପାସୱାର୍ଡ୍ ଏବଂ ବ୍ୟବହାରକାରୀ ନାମ ପରିବର୍ତ୍ତନ କରିବା ଭଲ ଅଭ୍ୟାସ | ଡିଫଲ୍ଟ ଭାବରେ ଏହା ମୂଳ ଅଟେ |
  - 4 - "ୱେବ୍ ୟୁରଲ୍ ପୃଷ୍ଠା ଖୋଲନ୍ତୁ" ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ - ଏଠାରେ ଆପଣ ଆପଣଙ୍କର ଡାଟା ସୂଚନା ପାସୱାର୍ଡ୍, ଆଡମିନି ନାମ ଏବଂ MAMP ବିଷୟରେ ସୂଚନା ମଧ୍ୟ ଦେଖିପାରିବେ |
  - 5 - ଉପକରଣଗୁଡ଼ିକ ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ -> phpMyAdmin ଏବଂ ଆପଣ ଡାଟାବେସ୍ ପୃଷ୍ଠାକୁ ପୁନଃ ନିର୍ଦ୍ଦେଶିତ ହେବେ |
  - 6 - ଏକ ନୂତନ ଡାଟାବେସ୍ ପ୍ରସ୍ତୁତ କରନ୍ତୁ, ନୂତନ ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ ଏବଂ ଆପଣ ଚାହୁଁଥିବା ନାମ ଦିଅନ୍ତୁ, ପରେ ଆପଣଙ୍କୁ ଏହି ନାମ ଦରକାର |
  - 7 - ବର୍ତ୍ତମାନ "htdocs" ଏକ ଫୋଲ୍ଡର କଲ୍ ଖୋଜ, ଏହା ତୁମର PC କିମ୍ବା Mac ରେ MAMP ଫୋଲ୍ଡର ଭିତରେ | ଯଦି ଆପଣ ଏକ ମ୍ୟାକ୍ ବ୍ୟବହାର କରନ୍ତି ତେବେ ଆପଣଙ୍କୁ ଫାଇଲ୍‌ସ୍ତରେ ଏକ ପ୍ରୟୋଗ କରିବାକୁ ପଡ଼ିବ ଏବଂ "ପ୍ୟାକେଜ୍ ବିଷୟବସ୍ତୁ ଦେଖାନ୍ତୁ" ବ୍ୟବହାର କରି ଆପ୍ ଖୋଲନ୍ତୁ | ଭିତରେ ଆପଣ htdocs ଫୋଲ୍ଡର ପାଇବେ |
  - 8 - ଖର୍ଚ୍ଚପ୍ରେସ୍ ଫୋଲ୍ଡର ନିଅ ଏବଂ ଏହାକୁ htdocs ଭିତରେ କପି କର, ତୁମକୁ ସମସ୍ତ ଫୋଲ୍ଡର ଭିତରେ ରଖିବା ଆବଶ୍ୟକ, ଜିପ୍ ଫାଇଲ୍ ରଖ ନାହିଁ | ଫୋଲ୍ଡରର ନାମ ପରିବର୍ତ୍ତନ କର, ତୁମେ ତୁମର ପ୍ରୋଜେକ୍ଟ ନାମ ସହିତ କଲ୍ କରିପାରିବ |
  - 9 - ଆପଣଙ୍କ ବ୍ରାଉଜରରେ MAMP ର ଓଷ୍ଟୋକୁ ଫେରିବା, ଏବଂ "ମୋ ସାଇଟ୍" ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ - ଏହା ଏକ ଖୋଲିବ |

URL କଲ୍ <http://localhost:8888> (8888 ହେଉଛି ଡିଫଲ୍ଟ ପୋର୍ଟ, ଆପଣ ଚାହଁଲେ ଏହାକୁ ପରିବର୍ତ୍ତନ କରିପାରିବେ) | ଏଠାରେ ଆପଣ htdocs ଫୋଲ୍ଡର ଭିତରେ ରଖିଥିବା ଫୋଲ୍ଡରକୁ ଦେଖିପାରିବେ, ଆପଣ ଫୋଲ୍ଡରକୁ ଦେଇଥିବା ନାମ ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ |

3 - ବର୍ତ୍ତମାନ ଖର୍ଚ୍ଚପ୍ରେସର ଏକ ସାଧାରଣ ସଂସ୍ଥାପନ ଆରମ୍ଭ କରନ୍ତୁ, ଆପଣଙ୍କୁ ଆଡମିନି ଏବଂ ପାସୱାର୍ଡ ବ୍ୟବହାର କରିବାକୁ ପଡ଼ିବ | ଡିଫଲ୍ଟ ଭାବରେ ଏହା ମୂଳ ଏବଂ ମୂଳ, ଏବଂ ଆପଣ ପୂର୍ବରୁ ସୃଷ୍ଟି କରିଥିବା ଡାଟାବେସର ନାମ ବ୍ୟବହାର କରନ୍ତୁ |

4 - ସଂସ୍ଥାପନ ଚଲାନ୍ତୁ ଏବଂ ଶେଷ କରନ୍ତୁ!

ଆପଣ ଆପଣଙ୍କର ୱେବସାଇଟ୍ କୁ [http://localhost:8888 / namefolder](http://localhost:8888/namefolder) ରେ ଦେଖିବେ ଅବଶ୍ୟ ଆପଣଙ୍କୁ MAMP ଚାଲିବା ଆବଶ୍ୟକ |

---

# Chapter 39: Making network requests with HTTP API

## Syntax

- `$response = wp_remote_get( $url, $args );`
- `$response = wp_remote_post( $url, $args );`
- `$response = wp_safe_remote_post( $url, $args );`

## Parameters

Parameter	Details
<code>\$url</code>	(string) (Required) Site URL to retrieve.
<code>\$args</code>	(array) (Optional) Request arguments.

## Remarks

## Returns

(*WP\_Error* | array) The response as an array, or *WP\_Error* on failure.

## Examples

### GET a remote JSON resource

This snippet will grab a JSON formatted resource, decode it and print it in PHP array format.

```
// Fetch
$response = wp_remote_get( 'http://www.example.com/resource.json' );

if ( ! is_wp_error( $response ) ) {
    $headers = wp_remote_retrieve_headers( $response );

    if ( isset( $headers[ 'content-type' ] ) && 'application/json' === $headers[ 'content-type' ] ) {
        print_r( json_decode( wp_remote_retrieve_body( $response ) ) );
    }
}
```

---

# Chapter 40: Meta Box

## Introduction

Simple usage of a Meta Box in the wp-admin content editors

## Syntax

- `_x( 'Text', 'Description', 'textdomain')` is used to add a description for the translation service instead of `__( 'Text', 'textdomain')` which is just the translation
- `_ex( 'Text', 'Description', 'textdomain')` is used to echo translated text with a description

## Remarks

ରେଣ୍ଡର ମେଟା ବାକ୍ସ ଭିତରେ ଥିବା ବିଷୟବସ୍ତୁ କିଛି ହୋଇପାରେ । ମୂଲ୍ୟଗୁଡ଼ିକ ସିଧାସଳଖ ଏକାଭିତ ହେବା ପରିବର୍ତ୍ତେ, ଆପଣ ଏକ PHP ଟେମ୍ପଲେଟ୍ ସହିତ ଏକ ଅନ୍ତର୍ଭୁକ୍ତ ମଧ୍ୟ ବ୍ୟବହାର କରିପାରିବେ ଏବଂ ଏହାକୁ ତଥ୍ୟ ପଠାଇବା ପାଇଁ `set_query_var` ପଦ୍ଧତି ବ୍ୟବହାର କରିପାରିବେ । ସେହି ମଧ୍ୟ ସମାନ ଭାବରେ କାମ କରିବ ।

## Examples

### A simple example with a regular input and a select input

```
/**
 * Add meta box to post types.
 *
 * @since 1.0.0
 */
function myplugin_add_meta_box() {
    // Set up the default post types/
    $types = array(
        'post',
    );

    // Optional filter for adding the meta box to more types. Uncomment to use.
    // $types = apply_filters( 'myplugin_meta_box_types', $types );

    // Add the meta box to the page
    add_meta_box(
        'myplugin-meta-box', // Meta Box Id. Can be anything.
        _x( 'Custom Meta', 'Custom Meta Box', 'myplugin' ), // The title of the meta box.
        // Translation is optional. Can just be string.
        'myplugin_render_meta_box', // The render meta box function.
        $types, // Add the post types to which to add the meta box.
        'side', // Show on the side of edit.
        'high' // Show at top of edit.
    );
}

add_action( 'add_meta_boxes', 'myplugin_add_meta_box' );
```

```

/**
 * Render the meta box.
 *
 * This shows examples of a basic input and a select inside a meta box. These can be anything.
 *
 * @since 1.0.0
 *
 * @param $post WP_Post The post being edited.
 */
function myplugin_render_meta_box( $post ) {
    // Get the current post meta values for our custom meta box.
    $city = get_post_meta( $post->ID, 'city', true ); // True is for returning a single
value.
    $country = get_post_meta( $post->ID, 'country', true ); // True is for returning a single
value.

    // Add the WP Nonce field for security.
    wp_nonce_field( plugin_basename(__FILE__), 'myplugin_meta_nonce' );
?>

<p>
<label for="city">
    <?php _ex( 'City', 'Custom Meta Box Template', 'myplugin' ); ?>
</label>
<input name="city" id="city" value="<?php echo $city; ?>" />
</p>
<p>
<label for="country">
    <?php _ex( 'Country', 'Custom Meta Box Template', 'myplugin' ); ?>
</label>
<select name="country" id="country">
    <option value="United States" <?php selected( $country, 'United States' ); ?><?php
_ex( 'United States', 'Custom Meta Box Template', 'myplugin' ); ?></option>
    <option value="Mexico" <?php selected( $country, 'Mexico' ); ?><?php _ex( 'Mexico',
'Custom Meta Box Template', 'myplugin' ); ?></option>
    <option value="Canada" <?php selected( $country, 'Canada' ); ?><?php _ex( 'Canada',
'Custom Meta Box Template', 'myplugin' ); ?></option>
</select>
</p>

<?php
}

/**
 * Save meta box data.
 *
 * @since 1.0.0
 *
 * @param $post_id int The Id of the Post being saved.
 */
function myplugin_save_meta_data( $post_id ) {
    // Verify this is not an auto save.
    if ( defined( 'DOING_AUTOSAVE' ) && DOING_AUTOSAVE ) {
        return;
    }

    // validate the meta box nonce for security.
    if ( ! isset( $_POST['myplugin_meta_nonce'] ) || ! wp_verify_nonce(
$_POST['myplugin_meta_nonce'], plugin_basename(__FILE__) ) ) {
        return;
    }
}

```

```

// Get the new values from the form.
$city    = $_POST['city'];
$country = $_POST['country'];

// update_post_meta will add the value if it doesn't exist or update it if it does.
update_post_meta( $post_id, 'city', $city );
update_post_meta( $post_id, 'country', $country );

/*
 * OPTIONAL ALTERNATIVE
 *
 * Instead of just using update_post_meta, you could also check the values and
 * issue create/update/delete on the post meta value.
 */
// $current_city_value = get_post_meta( $post_id, 'city', true ); // True is for returning
a single value.
//
// // Add the post meta if it doesn't exist.
// if ( $city && '' === $city ) {
//     add_post_meta( $post_id, 'city', $city, true ); // True means the key is unique to
the post. False is default and more than one can be added.
// }
// // Edit the post meta if it does exist and there is a new value.
// elseif ( $city && $city !== $current_city_value ) {
//     update_post_meta( $post_id, 'city', $city );
// }
// // Delete the post meta if there is no new value.
// elseif ( '' === $city && $current_city_value ) {
//     delete_post_meta( $post_id, 'city', $current_city_value ); // $current_city_value
is optional and is used to differentiate between other values with the same key.
// }
}

add_action( 'save_post', 'myplugin_save_meta_data' );

```

---

# Chapter 41: Options API

## Introduction

ବିକଳ୍ପଗୁଡ଼ିକ ହେଉଛି ଡାଟାବେସ୍ ଖଣ୍ଡ ଯାହା ବିଭିନ୍ନ ପସନ୍ଦ ଏବଂ ବିନ୍ୟାସ ସେଟିଂସ୍ ମୁହଁ ସଂରକ୍ଷଣ କରିବାକୁ ଉଦ୍ଦେଶ୍ୟ ବ୍ୟବହାର କରେ | ଡାଟାବେସ୍ରେ ଡାଟା ଗଠିତ କରିବାର ବିକଳ୍ପ API ହେଉଛି ଏକ ସରଳ ଏବଂ ମାନକ ଉପାୟ | API ବିକଳ୍ପଗୁଡ଼ିକ ସୃଷ୍ଟି, ପ୍ରବେଶ, ଅନ୍ୟତନ ଏବଂ ବିଲୋପ କରିବା ସହଜ କରିଥାଏ |

## Syntax

- // Create new option within WordPress  
add\_option( \$option, \$value = , \$deprecated = , \$autoload = 'yes' );
- // Removes an option from the database.  
delete\_option( \$option );
- // Retrieve a saved option  
get\_option( \$option, \$default = false );
- // Update the value of an option that was already added.  
update\_option( \$option, \$newvalue );
- // There are also \*\_site\_option() versions of these functions,  
// to manipulate network-wide options in WordPress Multisite
- // Create new network option  
add\_site\_option( \$option, \$value = , \$deprecated = , \$autoload = 'yes' );
- // Removes a network option  
delete\_site\_option( \$option );
- // Retrieve a saved network option  
get\_site\_option( \$option, \$default = false );
- // Update the value of an option that was already added.  
update\_site\_option( \$option, \$newvalue );

## Remarks

MySQL ଡାଟାବେସ୍ରେ ବିକଳ୍ପ ସାରଣୀରେ ସଂରକ୍ଷିତ ଡାଟା ସହଜ କାର୍ଯ୍ୟ କରିବାର ବିକଳ୍ପଗୁଡ଼ିକ ହେଉଛି ଏକ ସରଳ ଏବଂ ମାନକ ଉପାୟ | API ବିକଳ୍ପଗୁଡ଼ିକ ସୃଷ୍ଟି, ପଢ଼ିବା, ଅନ୍ୟତନ ଏବଂ ବିଲୋପ କରିବା ସହଜ କରିଥାଏ |

## Examples

### get\_option

`get_option` function is used to retrieve a value from from options table based on option name.

You can use the following code to get email address of a WordPress site administrator.

```
<?php echo get_option('admin_email'); ?>
```

`get_option()` has an optional 2nd argument, which allows you to set a default value to return in the case that the requested option isn't set. By default, this argument is `false`.

To retrieve a text string, and use a boilerplate string if the text isn't set in the options table, you could do this `get_option()` (ର ଏକ ବା `add_option()` କଳ୍ପିତ ଦ୍ୱିତୀୟ ଯୁକ୍ତି ଅଛି, ଯାହା ଆପଣଙ୍କୁ ଅନୁରୋଧ କରାଯାଇଥିବା ବିକଳ ସେଟ୍ ହୋଇନଥିବା କ୍ଷେତ୍ରରେ ଫେରିବା ପାଇଁ ଏକ ଡିଫଲ୍ଟ ମୂଲ୍ୟ ସେଟ୍ କରିବାକୁ ଅନୁମତି ଦେଇଥାଏ | ଡିଫଲ୍ଟ ଭାବରେ, ଏହି ଯୁକ୍ତି ମିଥ୍ୟା ଅଟେ |

ଏକ ଟେକ୍ସଟ୍ ସ୍ତମ୍ଭ ପୁନରୁଦ୍ଧାର କରିବାକୁ, ଏବଂ ଯଦି ଏକ ବ୍ୟଲେପ୍ଲେଟ୍ ସ୍ତମ୍ଭ ବ୍ୟବହାର କରନ୍ତୁ ଯଦି ପାଠ୍ୟ ବିକଳ ସାରଣୀରେ ସେଟ୍ ହୋଇନାହିଁ, ଆପଣ ଏହା କରିପାରିବେ:)

```
<?php get_option( 'my_text', "I don't have anything written. Yet." ); ?>
```

## add\_option

`add_option` function is used to insert new row into options table.

This will insert a new row in options table with option name `some_option_name` and value as `some_option_value`

```
<?php add_option( 'some_option_name', 'some_option_value' ); ?>
```

## delete\_option

`delete_option` function is used to delete an option from the options table.

This will delete `my_custom_option` from the options table.

```
<?php delete_option( 'my_custom_option' ); ?>
```

## update\_option

`update_option` function is used to update a value that already exists in the options table. If the option does not exist, then the option will be added with the option value. ( ଅପଡେଟ୍\_ପସନ୍ ଫଙ୍କସନ୍ ଏକ ଭାଲ୍ୟୁ ଅପଡେଟ୍ କରିବାକୁ ବ୍ୟବହୃତ ହୁଏ ଯାହା ଅସ୍ପନ୍ ଟେବୁଲ୍ ରେ ପୂର୍ବରୁ ବିଦ୍ୟମାନ | ଯଦି ଅସ୍ପନ୍ ବିଦ୍ୟମାନ ନାହିଁ, ତେବେ ଅସ୍ପନ୍ ଭାଲ୍ୟୁ ସହିତ ଅସ୍ପନ୍ ଯୋଡାଯିବ | )

This will set the default comment status to 'closed':

```
update_option( 'default_comment_status', 'closed' );
```

# Chapter 42: Plugin development

## Syntax

- `add_action(string $tag, callable $function_to_add, int $priority = 10, int $accepted_args = 1)`
- `add_filter(string $tag, callable $function_to_add, int $priority = 10, int $accepted_args = 1)`

## Parameters

Parameter	Detail
\$tag	(string) (Required) The name of the filter to hook the \$function_to_add callback to.
\$function_to_add	(callable) (Required) The callback to be run when the filter is applied.
\$priority	(int) (Optional) Used to specify the order in which the functions associated with a particular action are executed. Lower numbers correspond with earlier execution, and functions with the same priority are executed in the order in which they were added to the action. Default value: 10
\$accepted_args	(int) (Optional) The number of arguments the function accepts. Default value: 1

## Remarks

ପ୍ଲଗିନ୍ ଡେଭେଲପର କାର୍ଯ୍ୟ କରିବାର ଉପାୟ ହେଉଛି ଯେ ବିଭିନ୍ନ ସମୟରେ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଚାଲୁଥିବାବେଳେ, ଖର୍ଚ୍ଚପ୍ରେସ୍ ଯାଞ୍ଚ କରେ ଯେ ସେହି ସମୟରେ ଚାଲିବା ପାଇଁ କି ପ୍ଲଗିନ୍ ଶାସି ପ୍ଲଗିନ୍ କାର୍ଯ୍ୟ ପଞ୍ଜିକୃତ ହୋଇଛି କି ନାହିଁ, ଏବଂ ଯଦି ତାହା ହୁଏ, କାର୍ଯ୍ୟଗୁଡ଼ିକ ଚାଲିଥାଏ । ଏହି କାର୍ଯ୍ୟଗୁଡ଼ିକ ଖର୍ଚ୍ଚପ୍ରେସର ଡିଫଲ୍ଟ ଆଚରଣକୁ ପରିବର୍ତ୍ତନ କରେ ।

There are two kinds of hooks:

ଖର୍ଚ୍ଚପ୍ରେସ୍ ର ଏକଜେକ୍ୟୁଶନ୍ ସମୟରେ ଫିଲ୍ଟରଗୁଡ଼ିକ ଆପଣଙ୍କୁ ଏକ ଖଣ୍ଡର ମୂଲ୍ୟ ପରିବର୍ତ୍ତନ କରିବାର କ୍ଷମତା ଦେଇଥାଏ । ଫିଲ୍ଟରଗୁଡ଼ିକ ପାଇଁ କଲବ୍ୟାକ୍ ଫଙ୍କସନ୍ ଗୁଡ଼ିକ ଏକ ଭେରିଏବଲ୍, ପରିବର୍ତ୍ତନ ଏବଂ ପରେ ଫେରସ୍ତ ହେବ । ସେଗୁଡ଼ିକ ଏକ ପୃଥକ work ଜ୍ଞରେ କାର୍ଯ୍ୟ କରିବା ପାଇଁ ଭିନ୍ନ, ଏବଂ ଗ୍ଲୋବାଲ୍ ଭେରିଏବଲ୍ କିମ୍ବା ଫଙ୍କସନ୍ ବାହାରେ ଅନ୍ୟ କି affect ଶାସି ଜିନିଷକୁ ପ୍ରଭାବିତ କରିବା ଉଚିତ୍ ନୁହେଁ ।

କାର୍ଯ୍ୟଗୁଡ଼ିକ, ଏହାର ବିପରୀତରେ, ଖର୍ଚ୍ଚପ୍ରେସ୍ କିପରି କାର୍ଯ୍ୟ କରେ ତାହା ଯୋଡ଼ିବାକୁ କିମ୍ବା ପରିବର୍ତ୍ତନ କରିବାକୁ ଅନୁମତି ଦିଏ । ଆପଣଙ୍କର କଲବ୍ୟାକ୍ ଫଙ୍କସନ୍ ଖର୍ଚ୍ଚପ୍ରେସ୍ ର ଏକଜେକ୍ୟୁଶନ୍ ରେ ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ସମୟରେ ଚାଲିବ ଏବଂ ଏକ ପ୍ରକାର କାର୍ଯ୍ୟ କରିପାରିବ, ଯେପରି ଉପଭୋକ୍ତାଙ୍କୁ ଆଉଟପୁଟ୍ ଇକୋ କରିବା କିମ୍ବା ତାଗାବେସରେ କିଛି ଭର୍ତ୍ତି କରିବା ।

[Filter Reference](#)

[Action Reference](#)

## Examples

### Filter

```
add_filter('comment_text','before_comment');
add_filter('comment_text','after_comment');
function before_comment($comment_text){
    return 'input before comment'.$comment_text;
}
function after_comment($comment_text){
    return $comment_text.'input after comment';
}
```

### Action

```
add_action('wp_head','hook_javascript');

function hook_javascript() {
    $output="<script> alert('Page is loading...'); </script>";
    echo $output;
}
```

## Plugin development examples : Favorite Song Widget

```
<?php
function wpshout_register_widgets() {
    register_widget( 'Favorite_Song_Widget' );
}

add_action( 'widgets_init', 'wpshout_register_widgets' );

class Favorite_Song_Widget extends WP_Widget {

function Favorite_Song_Widget() {
    // Instantiate the parent object
    parent::__construct(
        'favorite_song_widget', // Base ID
        __('Favorite Song', 'text_domain'), // Name
        array( 'description' =>__( 'widget for playable favorite song', 'text_domain' ),
    ) // Args
    );
}

function widget( $args, $instance ) {
    echo $args['before_widget'];
    echo '<h3>Favorite Song Lists:</h3>';
    echo $instance['songinfo'];
    echo '<a href="' . $instance['link'] . '">Download it</a><br>';
}
```

```

    echo $instance['description'];
        echo $args['after_widget'];
}

function update($new_abc,$old_abc) {
    $instance = $old_abc;
    // Fields
    $instance['link'] = strip_tags($new_abc['link']);
    $instance['songinfo'] = strip_tags($new_abc['songinfo']);
        $instance['description'] = strip_tags($new_abc['description']);
    return $instance;
}

// widget form creation
function form($instance) {
    $link = '';
    $songinfo = '';
        $description = '';
    // Check values
    if( $instance) {
        $link = esc_attr($instance['link']);
        $songinfo = esc_textarea($instance['songinfo']);
            $description = esc_textarea($instance['description']);
    } ?>

    <p>
        <label for="<?php echo $this->get_field_id('link'); ?>"><?php _e('Link',
'wp_widget_plugin'); ?></label>
        <input class="widefat" id="<?php echo $this->get_field_id('link'); ?>" name="<?php
echo $this->get_field_name('link'); ?>" type="text" value="<?php echo $link; ?>" />
    </p>

    <p>
        <label for="<?php echo $this->get_field_id('songinfo'); ?>"><?php _e('Song Info:',
'wp_widget_plugin'); ?></label>
        <input class="widefat" id="<?php echo $this->get_field_id('songinfo'); ?>" name="<?php
echo $this->get_field_name('songinfo'); ?>" type="text" value="<?php echo $songinfo; ?>" />
    </p>

        <p>
            <label for="<?php echo $this->get_field_id('description'); ?>"><?php
_e('Description:', 'wp_widget_plugin'); ?></label>
            <textarea class="widefat" id="<?php echo $this->get_field_id('description'); ?>"
name="<?php echo $this->get_field_name('description'); ?>" type="text" value="<?php echo
$description; ?>"></textarea>
        </p>

        <p><a href="#" id="add-more-tabs"><?php _e('Add More Tabs', 'wp_widget_plugin');
?></a></p>

<?php }
}

```

# Chapter 43: Post Formats

## Remarks

The following Post Formats are available for users to choose from, if the theme enables support for them.

ଧାନ ଦିଅନ୍ତୁ ଯେ ଯେତେବେଳେ ପ୍ରକୃତ ପୋଷ୍ଟ ବିଷୟବସ୍ତୁ ପ୍ରବେଶ ପରିବର୍ତ୍ତନ ହେବ ନାହିଁ, ଥିମ୍ ନମୋନୀତ ଫର୍ମାଟ୍ ଉପରେ ଆଧାର କରି ପୋଷ୍ଟକୁ ଭିନ୍ନ ଭାବରେ ପ୍ରଦର୍ଶନ କରିବାକୁ ଏହି ଉପଭୋକ୍ତା ପଦକ୍ଷେପକୁ ବ୍ୟବହାର କରିପାରିବ । ଉଦାହରଣ ସ୍ୱରୂପ, ଏକ ଥିମ୍ “ସ୍ଲିଡି” ପୋଷ୍ଟ ପାଇଁ ଟାଇଟଲ୍ ପ୍ରଦର୍ଶନକୁ ଛାଡ଼ିପାରେ । ଜିନିଷଗୁଡ଼ିକ କିପରି ପ୍ରଦର୍ଶିତ ହୁଏ ତାହା ସମ୍ପୂର୍ଣ୍ଣ ବିଷୟବସ୍ତୁ ଉପରେ ନିର୍ଭର କରେ, କିନ୍ତୁ ଏଠାରେ କିଛି ସାଧାରଣ ନିର୍ଦ୍ଦେଶାବଳୀ ଅଛି ।

- ଅଲଗା - ସାଧାରଣତଃ a ଏକ ଆଖ୍ୟା ବିନା ଷ୍ଟାଇଲ୍ । ଏକ ଫେସବୁକ୍ ନୋଟ୍ ଅପଡେଟ୍ ପରି ।
- ଗ୍ୟାଲେରୀ - ଚିତ୍ରଗୁଡ଼ିକର ଏକ ଗ୍ୟାଲେରୀ । ପୋଷ୍ଟ ସମ୍ବନ୍ଧିତ a ଏକ ଗ୍ୟାଲେରୀ ସର୍ଚକୋଡ୍ ଧାରଣ କରିବ ଏବଂ ଏଥିରେ ପ୍ରତିଛବି ସଂଲଗ୍ନକ ରହିବ ।
- ଲିଙ୍କ - ଅନ୍ୟ ଏକ ସାଇଟ୍ ସହିତ ଏକ ଲିଙ୍କ । ଥିମ୍ ଗୁଡ଼ିକ ପୋଷ୍ଟ ବିଷୟବସ୍ତୁରେ ପ୍ରଥମ ଟ୍ୟାଗ୍ ସେହି ପୋଷ୍ଟ ପାଇଁ ବାହ୍ୟ ଲିଙ୍କ ଭାବରେ ବ୍ୟବହାର କରିବାକୁ ଇଚ୍ଛା କରିପାରନ୍ତି । ଏକ ବିକଳ୍ପ ପଛା ହୋଇପାରେ ଯଦି ପୋଷ୍ଟଟି କେବଳ ଏକ URL ଧାରଣ କରେ, ତେବେ ତାହା URL ହେବ ଏବଂ ଟାଇଟଲ୍ (ପୋଷ୍ଟ\_ ଟାଇଟଲ୍) ଏହା ପାଇଁ ଆଙ୍କର୍ ସହିତ ସଂଲଗ୍ନ ହେବ ।
- ପ୍ରତିଛବି - ଗୋଟିଏ ପ୍ରତିଛବି । ପୋଷ୍ଟର ପ୍ରଥମ ଟ୍ୟାଗ୍ ପ୍ରତିଛବି ଭାବରେ ବିବେଚନା କରାଯାଇପାରେ । ବ ly କଳ୍ପିକ ଭାବରେ, ଯଦି ପୋଷ୍ଟଟି କେବଳ ଏକ URL ଧାରଣ କରେ, ତାହା ଇମେଜ୍ URL ହେବ ଏବଂ ପୋଷ୍ଟର ଟାଇଟଲ୍ (ପୋଷ୍ଟ\_ ଟାଇଟଲ୍) ପ୍ରତିଛବି ପାଇଁ ଟାଇଟଲ୍ ଆଟ୍ରିବ୍ୟୁଟ୍ ହେବ ।
- ଉଦ୍ଧୃତ - ଏକ ଉଦ୍ଧୃତି । ସମ୍ବନ୍ଧିତ the କୋଟ ବିଷୟବସ୍ତୁ ଧାରଣ କରିଥିବା ଏକ ବ୍ଲକକୋଟ ଧାରଣ କରିବ । ବ ly କଳ୍ପିକ ଭାବରେ, ଉତ୍ସ / ଲେଖକ ଟାଇଟଲ୍ ସହିତ କୋଟ୍ କେବଳ ବିଷୟବସ୍ତୁ ହୋଇପାରେ ।
- ସ୍ଲିଡି - ଏକ ବିଚାର ସ୍ଲିଡି ଅପଡେଟ୍ ପରି ଏକ କ୍ଷୁଦ୍ର ସ୍ଲିଡି ଅପଡେଟ୍ ।
- ଭିଡିଓ - ଗୋଟିଏ ଭିଡିଓ କିମ୍ବା ଭିଡିଓ ପ୍ଲେଲିଷ୍ଟ । ପୋଷ୍ଟ ବିଷୟବସ୍ତୁରେ ପ୍ରଥମ ଟ୍ୟାଗ୍ କିମ୍ବା ବସ୍ତୁ / ଏମ୍ବେଡ୍ ଭିଡିଓ ଭାବରେ ବିବେଚନା କରାଯାଇପାରେ । ବ ly କଳ୍ପିକ ଭାବରେ, ଯଦି ପୋଷ୍ଟଟି କେବଳ ଏକ URL ଧାରଣ କରେ, ତାହା ହେଉଛି ଭିଡିଓ URL । ପୋଷ୍ଟରେ ଏକ ସଂଲଗ୍ନକ ଭାବରେ ଭିଡିଓ ମଧ୍ୟ ଧାରଣ କରିପାରେ, ଯଦି ବ୍ଲଗ୍ରେ ଭିଡିଓ ସପୋର୍ଟ ସମ୍ପନ୍ନ ହୁଏ (ଯେପରି ଏକ ପ୍ଲଗିନ୍ ମାଧ୍ୟମରେ) । ଅତିରିକ୍ତ ଏକ ଅତିରିକ୍ତ ଫାଇଲ୍ କିମ୍ବା ପ୍ଲେଲିଷ୍ଟ । ପୋଡକାଷ୍ଟିଂ ପାଇଁ ବ୍ୟବହୃତ ହୋଇପାରେ ।
- ଚାର୍ଟ - ଏକ ଚାର୍ଟ ଗ୍ରାଫିକ୍ସ ।

### Examples

#### Adding post type to Theme

##### Add post-formats to post\_type 'page'

```
add_post_type_support( 'page', 'post-formats' );
```

Next example registers custom post type 'my\_custom\_post\_type', and add Post Formats.

##### Register custom post type 'my\_custom\_post\_type'

```

add_action( 'init', 'create_my_post_type' );
function create_my_post_type() {
    register_post_type( 'my_custom_post_type',
        array(
            'labels' => array( 'name' =>__( 'Products' ) ),
            'public' => true
        )
    );
}

```

## Add post-formats to post\_type 'my\_custom\_post\_type'

```

add_post_type_support( 'my_custom_post_type', 'post-formats' );

```

Or in the function `register_post_type()`, add 'post-formats', in 'supports' parameter array. Next example is equivalent to above one.

## Register custom post type 'my\_custom\_post\_type' with 'supports' parameter

```

add_action( 'init', 'create_my_post_type' );
function create_my_post_type() {
    register_post_type( 'my_custom_post_type',
        array(
            'labels' => array( 'name' =>__( 'Products' ) ),
            'public' => true,
            'supports' => array('title', 'editor', 'post-formats')
        )
    );
}

```

## Add Theme Support for post

### Function Call

```

add_theme_support( 'post-formats' )

```

---

# Chapter 44: Querying posts

## Syntax

- `$the_query = new WP_Query( $args );`
- `$posts_array = get_posts( $args );`

## Parameters

Parameter	Description
<code>\$args</code>	<i>(array)</i> An array of needed arguments for a query - can be custom tailored to your needs, e.g. querying posts from only one category, from custom post type or even querying certain taxonomy

## Remarks

Query arguments are numerous. [WP\\_Query\(\) codex](#) page has a list of parameters. Some of them are

- [Author Parameters](#)
- [Category Parameters](#)
- [Tag Parameters](#)
- [Taxonomy Parameters](#)
- [Search Parameter](#)
- [Post & Page Parameters](#)
- [Password Parameters](#)
- [Type Parameters](#)
- [Status Parameters](#)
- [Pagination Parameters](#)
- [Order & Orderby Parameters](#)
- [Date Parameters](#)
- [Custom Field Parameters](#)
- [Permission Parameters](#)
- [Mime Type Parameters](#)
- [Caching Parameters](#)
- [Return Fields Parameter](#)

One of the most important thing to have in mind is:

## Never use `query_posts()`

`query_posts()` overrides the main query, and can cause problems in the rest of your theme. Any

time you need to modify the main query (or any query for that matter) is to use [pre\\_get\\_posts](#) filter. This will allow you to modify the query before it ran.

Also when you are querying posts, you should always reset it using [wp\\_reset\\_postdata\(\)](#). This will restore the global `$post` variable of the main query loop, and you won't have any issues later on (such as categories being excluded, because in your secondary loop you've excluded them and forgot to reset the query).

## Examples

### Using WP\_Query() object

Creating a separate instance of the `WP_Query` object is easy:

```
$query_args = array(
    'post_type' => 'post',
    'post_per_page' => 10
);

$my_query = new WP_Query($query_args);

if( $my_query->have_posts() ):
    while( $my_query->have_posts() ): $my_query->the_post();
        //My custom query loop
    endwhile;
endif;

wp_reset_postdata();
```

Notice that you need to build the query arguments array to your specification. For more details, look at [WP\\_Query codex page](#).

### Using get\_posts()

`get_posts()` is a wrapper for a separate instance of a `WP_Query` object. The returned value is an array of post object.

```
global $post;

$args = array(
    'numberposts' => 5,
    'offset'=> 1,
    'category' => 1
);

$myposts = get_posts( $args );

foreach( $myposts as $post ) :
    setup_postdata($post); ?>
    <h2><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
<?php endforeach;
wp_reset_postdata(); ?>
```



# Chapter 45: Remove Auto Line Breaks From Content and Excerpt

## Introduction

(ସଂପାଦକ କିମ୍ବା ଉତ୍ତର ଅଂଶରେ HTML ଉପରେ ନିର୍ଭର କରୁଥିବା ସାଇଟଗୁଡ଼ିକ ପାଇଁ, ଆପଣ ନିଜେ କୋଡ୍ କରିବାକୁ ଚାହୁଁଛନ୍ତି, ଅଟୋ ଲାଇନ୍ ବ୍ରେକ୍ ଏକ ବିରକ୍ତିକର ହୋଇପାରେ । ଏହି ଫିଲ୍ଟରଗୁଡ଼ିକ ଅପସାରଣ କରି ଆପଣ ସେଗୁଡ଼ିକୁ ଅକ୍ତିକରିପାରିବେ ।) For sites that rely on HTML by hand in the editor or excerpts, ones you want to code yourself, the auto line breaks can be an annoyance. You can disable them by removing these filters.

## Remarks

ଏଗୁଡ଼ିକ ଏକ ଅକ୍ତିକୃତ ଫାଇଲ୍ ରେ ସିଧାସଳଖ ଏକଜେକ୍ୟୁଟ୍ ହେବା ଜରୁରୀ । ଏହା function.php ରେ ହେଉ କିମ୍ବା ଅନ୍ୟ ଏକ ଫାଇଲ୍ ଅକ୍ତିକୃତ କରେ, ଏଗୁଡ଼ିକ ଏକ ସୁକ୍ରେ ଗୁଡ଼ାଯାଇପାରିବ ନାହିଁ । ସେମାନେ init କିମ୍ବା ମୁଁ ଏପର୍ଯ୍ୟନ୍ତ ପାଇଥିବା ଅନ୍ୟ କ on ଶସି କାର୍ଯ୍ୟରେ କାମ କରିବେ ନାହିଁ । କେବଳ ସେହି ଟେମ୍ପଲେଟ୍ ପାଇଁ ଏକଜେକ୍ୟୁଟ୍ କରିବାକୁ ସେଗୁଡ଼ିକ page.php ପରି ଏକ ଟେମ୍ପଲେଟ୍ରେ ସିଧାସଳଖ ଅକ୍ତିକୃତ କରାଯାଇପାରେ । ଟିପ୍ପଣୀ: ଏହାକୁ ଏକ ବିଭାଜିତ ଥିମ୍ କିମ୍ବା ପ୍ଲଗିନରେ ଅକ୍ତିକୃତ କରନ୍ତୁ ନାହିଁ (ଯଦି ଏହା ଅକ୍ତିକୃତ ନହୁଏ । ଡିଫଲ୍ଟ ଭାବରେ, ଯେପରି ଉପଭୋକ୍ତା ନିର୍ଦ୍ଦେଶ ନକରନ୍ତି ଏଥିରେ ଥିବା ଫାଇଲ୍ ଅକ୍ତିକୃତ ନକରିବା ପରି) ।

ଏକ ସାଇଟରେ ଅକ୍ତିକୃତ କରିବା ଏହା ଏକ ଖରାପ ଅଭ୍ୟାସ, କାରଣ ଏହା ଅନ୍ୟ କ them ଶସି ଥିମ୍ କିମ୍ବା ପ୍ଲଗିନ ର ଫଳାଫଳକୁ ଭାଙ୍ଗିପାରେ ।

## Examples

### Remove the Filters

```
// Remove the auto-paragraph and auto-line-break from the content
remove_filter( 'the_content', 'wpautop' );

// Remove the auto-paragraph and auto-line-break from the excerpt
remove_filter( 'the_excerpt', 'wpautop' );
```

### Function to remove the filters

```
/**
 * Remove the automatic line breaks from content and excerpts.
 *
 * @since 1.0.0
 */
function remove_content_auto_line_breaks() {
    // Remove the auto-paragraph and auto-line-break from the content
    remove_filter( 'the_content', 'wpautop' );

    // Remove the auto-paragraph and auto-line-break from the excerpt
    remove_filter( 'the_excerpt', 'wpautop' );
}
```

```
// Execute the function  
remove_content_auto_line_breaks();
```

# Chapter 46: Remove Version from Wordpress and Stylesheets

## Introduction

ଅନ୍ୟମାନଙ୍କ ପାଇଁ ଆପଣଙ୍କର ଝେବସାଇଟ୍ ହ୍ୟାକ୍ କରିବା ଅଧିକ କଷ୍ଟକର କରିବା ପାଇଁ ଆପଣ ଆପଣଙ୍କର ସାଇଟ୍, ଆପଣଙ୍କର css ଏବଂ js ରୁ ଝଡ଼ିପ୍ରେସ୍ ଭର୍ସନ୍ ନମ୍ବର ଅପସାରଣ କରିପାରିବେ । ସେହି ସଂଖ୍ୟା ବିନା ପୁରାତନ ସଂସ୍କରଣରୁ ତ୍ରୁଟି ଶୋଷଣ କରିବା ପାଇଁ ଆପଣ ବର୍ତ୍ତମାନର ସଂସ୍କରଣ ଚଳାଉ ନାହିଁ ତାହା ଦେଖିବା ସମ୍ଭବ ନୁହେଁ ।

ଅତିରିକ୍ତ ଭାବରେ, ଏହା ଆପଣଙ୍କ ସାଇଟର ଲୋଡ଼ିଙ୍ଗ୍ ସ୍ପିଡ୍ କୁ ଉନ୍ନତ କରିପାରିବ, କାରଣ URL ରେ ଜିଜ୍ଞାସା ଷ୍ଟ୍ରିଙ୍ଗ୍ ବିନା css ଏବଂ js ଫାଇଲଗୁଡ଼ିକ କ୍ୟାଚ୍ ହୋଇପାରିବ ।

## Syntax

- `add_filter( $tag, $function_to_add, $priority, $accepted_args)`

## Parameters

Parameter	Details
<code>\$tag</code>	<i>(string required)</i> Name of the filter where <code>\$function_to_add</code> is hooked to
<code>\$function_to_add</code>	<i>(callable required)</i> Name of the function which runs when the filter is applied
<code>\$priority</code>	<i>(int optional)</i> place of <code>\$function_to_add</code> between other functions in one action (default = 10)
<code>\$accepted_args</code>	<i>(int optional)</i> Number of parameters which <code>\$function_to_add</code> accepts (default = 1)

## Remarks

Intended to improve site speed and safety.

## Examples

### Remove version numbers from css/js

Just add this function to your `functions.php`. It will remove the version from all enqueued js and css files.

```
function remove_cssjs_ver( $src ) {
    if( strpos( $src, '?ver=' ) )
        $src = remove_query_arg( 'ver', $src );
    return $src;
}

add_filter( 'style_loader_src', 'remove_cssjs_ver', 999 );
add_filter( 'script_loader_src', 'remove_cssjs_ver', 999 );
```

## Remove version numbers from WordPress

If you add this to your functions.php it removes the WordPress version number from the RSS feed and the header.

```
function remove_wordpress_ver() {
    return '';
}

add_filter('the_generator', 'remove_wordpress_ver', 999);
```

# Chapter 47: REST API

## Introduction

ଓର୍ଡ଼ପ୍ରେସ୍ REST API ଓର୍ଡ଼ପ୍ରେସ୍ ତାଟା ପ୍ରକାରଗୁଡ଼ିକ ପାଇଁ API ଏଣ୍ଡପଏଣ୍ଟ୍ସ ପ୍ରଦାନ କରେ ଯାହା ଡେଭଲପର୍ମାନଙ୍କୁ JSON (ଜାଭାସ୍କ୍ରିପ୍ଟ ଅବଜେକ୍ଟ ନୋଟେସନ୍) ବସ୍ତୁ ପଠାଇବା ଏବଂ ଗ୍ରହଣ କରି ଦୂରଦୂରାନ୍ତରେ ଯୋଗାଯୋଗ କରିବାକୁ ଅନୁମତି ଦେଇଥାଏ ।

ଯେତେବେଳେ ଆପଣ API କୁ ବିଷୟବସ୍ତୁ ପଠାନ୍ତି କିମ୍ବା ଅନୁରୋଧ କରନ୍ତି, ପ୍ରତିକ୍ରିୟା JSON ରେ ଫେରସ୍ତ ହେବ । ଏହା ଡେଭଲପର୍ମାନଙ୍କୁ କ୍ଲାଏଣ୍ଟ-ସାଇଡ୍ ଜାଭାସ୍କ୍ରିପ୍ଟରୁ କିମ୍ବା ବାହ୍ୟ ପ୍ରୟୋଗଗୁଡ଼ିକରୁ ଓର୍ଡ଼ପ୍ରେସ୍ ବିଷୟବସ୍ତୁ ସୃଷ୍ଟି, ପଢ଼ିବା ଏବଂ ଅନ୍ୟତନ କରିବାକୁ ସମ୍ଭବ କରିଥାଏ, ଏପରିକି PHP ବାହାରେ ଭାଷାରେ ଲିଖିତ ।

## Remarks

To get this WordPress REST API simple example to work for you, you need to learn how it works in more detail. Official documentation recommends learning about:

1. ମାର୍ଗ / ଏଣ୍ଡପଏଣ୍ଟ୍ସ(Routes/Endpoints) - ଯାହାକି “ଏଣ୍ଡପଏଣ୍ଟ୍ସ” ଭାବରେ ଜଣାଶୁଣା ମାର୍ଗକୁ ବ୍ୟକ୍ତିଗତ HTTP ପଦ୍ଧତିର ମ୍ୟାପିଂ ଅଟେ - ଆପଣ ଏହାକୁ ରେଜିଷ୍ଟର\_ରେଷ୍ଟ\_ରୋଡ୍ () ଫଙ୍କସନ୍ ବ୍ୟବହାର କରି କରନ୍ତି, ଏବଂ ଏଠାରେ ଆପଣ ମାର୍ଗ ଏବଂ ଏଣ୍ଡପଏଣ୍ଟ୍ସ ବିଷୟରେ ଅଧିକ ପାଇପାରିବେ ।
2. ଅନୁରୋଧ (Requests -) - ଓର୍ଡ଼ପ୍ରେସ୍ REST API WP\_REST\_Request ଶ୍ରେଣୀକୁ ବ୍ୟାଖ୍ୟା କରେ ଯାହା ସାମ୍ପ୍ରତିକ ଅନୁରୋଧ ପାଇଁ ସୂଚନା ସଂରକ୍ଷଣ ଏବଂ ପୁନରୁଦ୍ଧାର ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ଯେତେବେଳେ ଆପଣ ଏକ ପଞ୍ଜୀକୃତ ମାର୍ଗରେ ଏକ HTTP ଅନୁରୋଧ କରନ୍ତି WP\_REST\_Request ବସ୍ତୁଗୁଡ଼ିକ ସ୍ୱୟଂଚାଳିତ ଭାବରେ ଆପଣଙ୍କ ପାଇଁ ସୃଷ୍ଟି ହୋଇଥାଏ । ଅନୁରୋଧରେ ନିର୍ଦ୍ଦିଷ୍ଟ ହୋଇଥିବା ତଥ୍ୟ ନିର୍ଣ୍ଣୟ କରିବ ଯେ ଆପଣ API ରୁ କେଉଁ ପ୍ରତିକ୍ରିୟା ପାଇବେ । WP\_REST\_Request ଶ୍ରେଣୀ ବିଷୟରେ ଏଠାରେ ଅଧିକ ଜାଣିପାରିବେ ।
3. ପ୍ରତିକ୍ରିୟା ()Responses- ଆପଣ API ରୁ ଫେରିଥିବା ତଥ୍ୟ । WP\_REST\_Response ଶ୍ରେଣୀ ପଏଣ୍ଟ୍ସ ଦ୍ୱାରା ଫେରସ୍ତ କରାଯାଇଥିବା ପ୍ରତିକ୍ରିୟା ତଥ୍ୟ ସହିତ ଯୋଗାଯୋଗ କରିବାର ଏକ ଉପାୟ ପ୍ରଦାନ କରିଥାଏ । ତୁମ୍ଭର ଏଣ୍ଡପଏଣ୍ଟ୍ସ ପରିଭାଷାରେ ତୁମ୍ଭେ ପାରମ୍ପରିକ କାର୍ଯ୍ୟକଳାପକୁ ସେବା କରିବା ପାଇଁ କଲବ୍ୟାକ୍ (ପ୍ରତିକ୍ରିୟା) ଫଙ୍କସନ୍ ନାମ ଦିଅ । WP\_REST\_Response ଶ୍ରେଣୀ ବିଷୟରେ ଏଠାରେ ଅଧିକ ଜାଣିପାରିବେ ।
4. ସ୍କିମା(Schema) - ପ୍ରତ୍ୟେକ ଶ୍ରେଣୀ ପଏଣ୍ଟ୍ସ ସାମାନ୍ୟ ଭିନ୍ନ ତାଟା ସଂରଚନା ଆବଶ୍ୟକ କରେ ଏବଂ ପ୍ରଦାନ କରିଥାଏ, ଏବଂ ସେହି ସଂରଚନାଗୁଡ଼ିକ API ସ୍କିମାରେ ବ୍ୟାଖ୍ୟା କରାଯାଇଥାଏ । ଯଦି ଆପଣ ରକ୍ଷଣାବେକ୍ଷଣ, ଆବିଷ୍କାରଯୋଗ୍ୟ ଏବଂ ସହଜରେ ବିସ୍ତାରିତ ଶ୍ରେଣୀ ପଏଣ୍ଟ୍ସ ଚାହୁଁଛନ୍ତି ତେବେ ଏହି ସ୍କିମା ବ୍ୟବହାର କରିବାକୁ ପରାମର୍ଶ ଦିଆଯାଇଛି । ଏଠାରେ ଆପଣ ସ୍କିମା ବିଷୟରେ ଅଧିକ ଜାଣିପାରିବେ ।
5. ନିୟନ୍ତ୍ରଣ ଶ୍ରେଣୀ (Control Access)- ସେମାନେ ସମସ୍ତ ଉପାଦାନକୁ ଗୋଟିଏ ସ୍ଥାନରେ ଏକାଠି କରନ୍ତି । ଏକ ନିୟନ୍ତ୍ରଣ ଶ୍ରେଣୀ ସହିତ ଆପଣ ରୁଡ୍ ଏବଂ ଏଣ୍ଡପଏଣ୍ଟ୍ସଗୁଡ଼ିକର ପଞ୍ଜୀକରଣ ପରିଚାଳନା କରିପାରିବେ, ଅନୁରୋଧଗୁଡ଼ିକ ପରିଚାଳନା କରିପାରିବେ, ସ୍କିମା ବ୍ୟବହାର କରିପାରିବେ ଏବଂ API ପ୍ରତିକ୍ରିୟା ସୃଷ୍ଟି କରିପାରିବେ । ଆପଣ ଦୁଇଟି ନିୟନ୍ତ୍ରଣ ଶ୍ରେଣୀ ବିଷୟରେ ପୂର୍ବରୁ ଜାଣିସାରିଛନ୍ତି: WP\_REST\_Request ଏବଂ WP\_REST\_Response |Here you can learn more about the [Controller Classes](#)

Note: Some of this information is taken from the official [Wordpress REST APi Handbook](#)

# Examples

## Complete working example

```
add_action('rest_api_init', 'my_rest_validate_endpoint' );
function my_rest_validate_endpoint() {

    // Declare our namespace
    $namespace = 'myrest/v1';

    // Register the route
    // Example URL matching this route:
    // http://yourdomain/wp-json/myrest/v1/guides/tag=europe/price=29
    register_rest_route($namespace,
        // Using regular expressions we can initially validate the input
        '/guides/tag=(?P<tag>[a-zA-Z0-9-]+)/price=(?P<price>[0-9]+)',
        // We can have multiple endpoints for one route
        array(
            array(
                'methods' => 'GET',
                'callback' => 'my_get_guides_handler'
            )
        ),
        // We can register our schema callback
        // 'schema' => 'my_get_guide_schema',

    );

    // You can register another route here the same way
}

// The callback handler for the endpoint
function my_get_guides_handler(WP_REST_Request $request) {

    // Get the parameters:
    $tag = $request->get_param('tag');
    $price = $request->get_param('price');

    // Do something with the parameters
    // for instance: get matching guides from the DB into an array - $results
    // ...

    // Prepare the response
    $message = "we've found " . count($results) . " guides ";
    $message .= "(searching for a tag: " . $tag . ", with a price tag: " . $price . ")";

    // The response gets automatically converted into a JSON format
    return new WP_REST_Response(
        array(
            'results' => $results,
            'message' => $message,
            'status' => 'OK'
        ),
        200 );
}
```

---

# Chapter 48: Run WordPress local with XAMPP

## Introduction

XAMPP ସହିତ ଆପଣ ଆପଣଙ୍କର ସ୍ଥାନୀୟ PC ରେ ଏକ ୱେବ୍ ସର୍ଭର ସଂସ୍ଥାପନ କରିପାରିବେ । ଏଥିରେ ଏକ ଆପାଚେ ୱେବ୍ ସର୍ଭର, ଡାଟାବେସ୍ ମାରିଆଡିବି (MySQL) ଅଛି ଏବଂ Perl ଏବଂ PHP ସହିତ କାମ କରେ । ସଂସ୍ଥାପନ ପରେ ଆପଣ ଚଲାଉବାକୁ ଏବଂ ଡିବଗ୍ କରିବାକୁ ସକ୍ଷମ ଅଟନ୍ତି । ଆପଣଙ୍କର ସ୍ଥାନୀୟ PC ରେ WordPress ପରି ବିଷୟବସ୍ତୁ ପରିଚାଳନା ପ୍ରଣାଳୀ । ଆପଣ ଏହାକୁ ଓପେନସି, ଲିନକ୍ସ, ମାକ୍ OS X ଏବଂ ସୋଲାରିସ୍ ସହିତ ବ୍ୟବହାର କରିପାରିବେ ।

### Examples

#### 1. Installing XAMPP

1. XAMPP ର ସାମ୍ପ୍ରତିକ ସଂସ୍କରଣ ପାଇଁ ସଂସ୍ଥାପକକୁ ଡାଉନଲୋଡ୍ କରନ୍ତୁ ।
2. ସଂସ୍ଥାପନ ଓଜାର୍ଡ୍ ଦେଇ ଯାଆନ୍ତୁ ଏବଂ ଯଦି ଆପଣ କ'ଣ ପରିବର୍ତ୍ତନ କରୁଛନ୍ତି ତାହା ନିଶ୍ଚିତ ଦୁଇଟି ଡେବେ କିଛି ପରିବର୍ତ୍ତନ କରନ୍ତୁ ନାହିଁ ।
3. ସଂସ୍ଥାପନ ପରେ ଆପଣ XAMPP କଣ୍ଟ୍ରୋଲ୍ ପ୍ୟାନେଲ୍ ଦେଖନ୍ତି । ଉଭୟ ମା basic ଲିକ ବିନ୍ୟାସ ସହିତ ଚଲାଉବା ପାଇଁ କେବଳ ଆପାଚେ ଏବଂ MySQL ର ଆରମ୍ଭ ବଟନ୍ କ୍ଲିକ୍ କରନ୍ତୁ ।
4. ଯଦି ଆପଣଙ୍କର କମ୍ପ୍ୟୁଟର ଏକ ସ୍ଥାନୀୟ ନେଟୱାର୍କ ସହିତ ସଂଯୁକ୍ତ, ଏବଂ ଆପଣ ଅନ୍ୟ କମ୍ପ୍ୟୁଟରରୁ ୱେବ୍ ସର୍ଭରକୁ ପ୍ରବେଶ କରିବାକୁ ଚାହୁଁଛନ୍ତି, ନିଶ୍ଚିତ କରନ୍ତୁ ଯେ ଆପଣଙ୍କର ଫାୟାରୱାଲ୍ ଏହାକୁ ଅନୁମତି ଦିଏ ।

#### 2. Setup a database after installing XAMPP

To run WordPress on your computer you need to configurate a database first. Be sure that Apache and MySQL are running (see [Installing XAMPP](#) step 3)

1. ଆପଣଙ୍କ ପସନ୍ଦର ଏକ ୱେବ୍ ବ୍ରାଉଜର ଆରମ୍ଭ କରନ୍ତୁ ଏବଂ ଠିକଣାରେ "ଲୋକାଲ ହୋଷ୍ଟ" ପ୍ରବେଶ କରନ୍ତୁ ।
2. ଆପଣଙ୍କର ପସନ୍ଦିତ ଭାଷା ବାଛନ୍ତୁ ଏବଂ "ଉପକରଣ" -> phpMyAdmin ଶ୍ରେଣୀରେ ବାଛନ୍ତୁ ।
3. ଡାଟାବେସ୍ ଟ୍ୟାବ୍ ଚୟନ କରନ୍ତୁ ।
4. ଆପଣଙ୍କର ଡାଟାବେସ୍ ର ନାମ ପ୍ରବେଶ କରନ୍ତୁ (ଆପଣ ପରେ ଏହି ନାମ ଆବଶ୍ୟକ କରିବେ) ତଳେ "ଡାଟାବେସ୍ ସୃଷ୍ଟି କରନ୍ତୁ" ଏବଂ ତ୍ରୁପ୍ ଡାଉନ୍ "ସଂଗ୍ରହ" ରେ utf8\_general\_ci ବାଛନ୍ତୁ ।
5. ବାମ ପାର୍ଶ୍ୱରେ ଆପଣଙ୍କର ସୃଷ୍ଟି ହୋଇଥିବା ଡାଟାବେସ୍ରେ କ୍ଲିକ୍ କରନ୍ତୁ ଏବଂ ବ୍ୟବହାରକାରୀ ଟ୍ୟାବ୍ ଚୟନ କରନ୍ତୁ ।
6. ଯୁଜର୍ ଆଡ୍ କ୍ଲିକ୍ କରି ଏକ ନୂତନ ଯୁଜର୍ ଯୋଡନ୍ତୁ ।
7. ଆପଣଙ୍କର ଉପଯୋଗକର୍ତ୍ତା ନାମ ପ୍ରବେଶ କରନ୍ତୁ, "ହୋଷ୍ଟ" ତ୍ରୁପ୍ ଡାଉନ୍ରେ ସ୍ଥାନୀୟ ବାଛନ୍ତୁ (ଯଦି ଆପଣ ନେଟୱାର୍କର ଅନ୍ୟ କମ୍ପ୍ୟୁଟରରୁ ଡାଟାବେସ୍ ପ୍ରବେଶ କରିବାକୁ ଚାହୁଁଛନ୍ତି ଯେକି Any ଶସି ହୋଷ୍ଟ ବାଛନ୍ତୁ) ଏବଂ ଆପଣଙ୍କର ପାସୱାର୍ଡ୍ ପ୍ରବେଶ କରନ୍ତୁ (ଆପଣଙ୍କୁ ଏହା ପରେ ମଧ୍ୟ ଆବଶ୍ୟକ ହେବ) ।
8. ନିମ୍ନରେ "ଉପଭୋକ୍ତା ପାଇଁ ଡାଟାବେସ୍" ଖାଲକୁ ନାମ ଉପରେ ସମସ୍ତ ସୁବିଧା ଯାଞ୍ଚ କରାଯାଇଛି କି ନାହିଁ ଯାଞ୍ଚ କରନ୍ତୁ ।
9. ଗୋ ଦବାନ୍ତୁ ।

#### 3. Installing WordPress after setup the database

After you installed XAMPP and setup the MySQL database you can install WordPress.

1. ଝଡ଼ପ୍ରେସର ସର୍ବଶେଷ ସଂସ୍କରଣ ଡାଉନଲୋଡ଼ କରନ୍ତୁ ।
2. ଫାଇଲ୍ ଅପ୍ଲାଇଡ୍ କରନ୍ତୁ ଏବଂ ଫୋଲ୍ଡରକୁ ଆପଣଙ୍କର ୱେବ୍ ସାଇଟର ମୂଳ ଡିରେକ୍ଟୋରୀରେ ସବ୍‌ସେଟ୍ କରନ୍ତୁ (ଯଦି ଆପଣ XAMPP ସେଟଅପ୍ ସମୟରେ ପରାମର୍ଶିତ ପଥ ବ୍ୟବହାର କରନ୍ତି ତେବେ ଏହା "c: \xampp \htdocs") ।
3. ସମସ୍ତ ଫାଇଲ୍ ଯାହା ଦୁଇଥର ବିଦ୍ୟମାନ ଅଛି ତାହା ବଦଳାଇପାରିବ ।
4. ଆପଣଙ୍କର ୱେବ୍ ବ୍ରାଉଜର "ଲୋକାଲ୍ ହୋଷ୍ଟ / ୱାଡ଼ମିନ୍ / install.php" ର ଠିକଣା କ୍ଷେତ୍ରରେ ପ୍ରବେଶ କରନ୍ତୁ ।
5. ଆପଣଙ୍କର ଭାଷା ବାଛିବା ପରେ ଜାରି ରଖନ୍ତୁ କ୍ଲିକ୍ କରନ୍ତୁ ଏବଂ ତାପରେ ଚାଲି ଯିବ ।
6. ଆପଣଙ୍କର ମନୋନୀତ ସଂରଚନା ସହିତ ସମସ୍ତ ଡିଫଲ୍ଟ ପାଠ୍ୟ କ୍ଷେତ୍ରଗୁଡ଼ିକୁ ବଦଳାନ୍ତୁ ("ଏକ ଡାଟାବେସ୍ ସେଟଅପ୍" ଦେଖନ୍ତୁ) । ଆପଣ ଆପଣଙ୍କର ସ୍ଥାନୀୟ ଝଡ଼ପ୍ରେସ୍ ସଂସ୍ଥାପନକୁ ଅନୁଲୋଚନାରେ ସ୍ଥାନାନ୍ତର କରିବାକୁ ଚାହୁଁଥିବା ବ୍ୟତୀତ "ଡାଟାବେସ୍ ହୋଷ୍ଟ" ଏବଂ "ଟେବୁଲ୍ ପ୍ରିଫିକ୍ସ" ପରିବର୍ତ୍ତନ କରନ୍ତୁ ନାହିଁ । ତା'ପରେ ମନେରଖନ୍ତୁ ଯେ ସୁରକ୍ଷା ଗୁରୁତ୍ୱପୂର୍ଣ୍ଣ । ଆପଣ ନୂତନ ଝଡ଼ପ୍ରେସ୍ ସଂସ୍ଥାପନାରେ ସେଟ୍ ପ୍ରିଫିକ୍ସରେ ଅଧିକ ସୁଚନା ପାଇପାରିବେ ।
7. ସର୍ଭିସ୍ ବଦଳନ୍ତୁ ଏବଂ ଇନଷ୍ଟାଲ୍ ଚଳାଇବା ଅପେକ୍ଷା ।
8. ବର୍ତ୍ତମାନ ଆପଣଙ୍କୁ ଆପଣଙ୍କର ସାଇଟର ନାମ, ଏକ ଉପଯୋଗକର୍ତ୍ତା ନାମ, ପାସୱାର୍ଡ୍ ଏବଂ ଏକ ବ valid ଧ ଇ-ମେଲ୍ ପ୍ରବେଶ କରିବାକୁ ପଡ଼ିବ ।
9. ସେଟଅପ୍ ଶେଷ କରିବାକୁ ଏବଂ ଆପଣଙ୍କର ନୂତନ ସ୍ଥାନୀୟ ଝଡ଼ପ୍ରେସ୍ ସାଇଟରେ ଲଗ୍ କରିବାକୁ ଝଡ଼ପ୍ରେସ୍ ସଂସ୍ଥାପନ କ୍ଲିକ୍ କରନ୍ତୁ ।

# Chapter 49: Secure your installation

## Remarks

ଓର୍ଡ଼ପ୍ରେସ୍ ଖେବସାଇଟ୍ ଗୁଡ଼ିକ ବାରମ୍ବାର ହ୍ୟାକ୍ ହୋଇଥାଏ | ଏହି ବିଷୟଟି କି technic ଶିଳ ଏବଂ ଅଭ୍ୟାସ ପାଇଁ, ଯାହା ଆପଣଙ୍କର ଓର୍ଡ଼ପ୍ରେସ୍ ସଂସ୍ଥାର ସୁରକ୍ଷାକୁ ବଢ଼ାଇଥାଏ ଯାହା ଏକ ଆଧାର ସଂସ୍ଥାପନରେ ହାସଲ କରାଯାଇଥାଏ |

Apart from this topic, another good place to read about securing a WordPress installation is the [Hardening WordPress Codex page](#).

## Examples

### Disable File Editor

ଓର୍ଡ଼ପ୍ରେସ୍ ସହିତ ପଠାଉଥିବା ଫାଇଲ୍ ଏଡିଟର୍ ହେଉଛି ଏକ ସୁରକ୍ଷା ବିପଦ | ଯଦି ଜଣେ ଆକ୍ରମଣକାରୀ ଆପଣଙ୍କର ଓର୍ଡ଼ପ୍ରେସ୍ ଖେବସାଇଟ୍ କୁ ଆଡମିନି ଆକ୍ସେସ୍ ଲାଭ କରନ୍ତି ତେବେ ସେମାନେ ସହଜରେ ଥିମ୍ ଏବଂ ପ୍ଲଗିନ୍ ଫାଇଲରେ ଦୁଷ୍ଟ କୋଡ୍ ସମ୍ପର୍କିତ କରିବାକୁ ସମର୍ଥ ହେବେ | ଗ୍ରାହକମାନଙ୍କ ସହିତ ଏହା ମଧ୍ୟ ଏକ ବିପଦ, ଯେଉଁମାନେ ଜାଣନ୍ତି ନାହିଁ ସେମାନେ କଣ କରୁଛନ୍ତି | ଥରେ ଫାଇଲ୍ ଏଡିଟର୍ ରେ ଭୁଲ୍ ସ୍ଥାନିତ କୋଲନ୍ ଏକ ସାଇଟ୍ ଭାଙ୍ଗି ବ୍ରାଉଜରରୁ ଏହାକୁ ଅସମ୍ଭବ କରିପାରେ | In your WordPress wp-config.php file, disable the file editor by adding the following line of code.

```
define( 'DISALLOW_FILE_EDIT', true );
```

That line will have the desired effect when added to your theme's functions.php file too but it is better to add to wp-config.php.

If you are using [WordPress CLI](#) to install WordPress you can use the following command to create a wp-config.php file with file editing disabled.

```
/* declare variables beforehand or substitute strings in */
wp core config --dbname="$MYSQL_DBNAME" --dbuser="$MYSQL_USERNAME" --dbpass="$MYSQL_PASS" --
dbprefix="$WP_DBPREFIX" --locale=en_AU --extra-php <<PHP
define( 'DISALLOW_FILE_EDIT', true );
PHP
```

This method is useful if you install WordPress with a script.

### Move wp-config.php

ଓର୍ଡ଼ପ୍ରେସ୍ ଇନଷ୍ଟଲ୍ ର ସବୁଠାରୁ ସମ୍ପେଦନଶୀଳ ସୂଚନା wp-config.php ଫାଇଲରେ ଗଢ଼ିତ | ଯଦି ଜଣେ ହ୍ୟାକର୍ ଏହି ଫାଇଲ୍ କୁ ପ୍ରବେଶ କରେ ତେବେ ତୁମର ଖେବସାଇଟ୍ ଉପରେ ସେମାନଙ୍କର ସମ୍ପୂର୍ଣ୍ଣ ନିୟନ୍ତ୍ରଣ ଅଛି |

ଡିଫଲ୍ଟ ଭାବରେ wp-config.php ଓର୍ଡ଼ପ୍ରେସ୍ ଇନଷ୍ଟଲ୍ ଫୋଲ୍ଡରରେ ଗଢ଼ିତ | ଏହି ଫାଇଲ୍ ଚୋରି କରିବା କଷ୍ଟକର କରିବାକୁ ଆପଣ ଏହାକୁ ଖେବ୍ ଆକ୍ସେସିବଲ୍ ଫୋଲ୍ଡରରୁ ବାହାର କରିପାରିବେ | ଯଦି ଆପଣ ଏହାକୁ ଉପରେ କେବଳ ଗୋଟିଏ ଫୋଲ୍ଡର ଘୁଞ୍ଚାନ୍ତି, ଓର୍ଡ଼ପ୍ରେସ୍ ସ୍ୱୟଂଚାଳିତ ଭାବରେ ଏହାକୁ ପାଇବ | ଯଦି ଆପଣ wp-config.php କୁ ଏକ ଭିନ୍ନ ସ୍ଥାନକୁ ସ୍ଥାନାନ୍ତର କରନ୍ତି, ତେବେ ଓର୍ଡ଼ପ୍ରେସ୍ ସଂସ୍ଥାପନ ଫୋଲ୍ଡରରେ wp-config.php ନାମକ ଏକ ଖାଲି ଫାଇଲ୍ ସୃଷ୍ଟି କରନ୍ତୁ | ତାପରେ ନିମ୍ନଲିଖିତ ଯୋଗ କରନ୍ତୁ:

```
define('ABSPATH', dirname(__FILE__) . '/');
// '../..wp-config.php' defines location two folders above installation folder.
// Substitute with actual location of wp-config.php file as necessary.
require_once(ABSPATH . '../..wp-config.php');
```

You may need to make `php` executable in the folder you place `wp-config.php` in. You should make `php` executable in as few folders as possible. A good system puts the WordPress install in `/path/to/wordpress/install/` and the config in `/path/to/wordpress/config`. You'd make sure the config folder is not web accessible and don't place any other sensitive information would be placed in `/path/to/` or higher in the folder hierarchy. In that case you'd write a line similar to the following in your `php.ini`:

```
open_basedir = "/path/to/wordpress/install/;/path/to/wordpress/config"
```

This technique is controversial and some people don't think it enhances security. Extensive discussion on the topic can be read at [this WordPress StackExchange question](#).

## Set a custom prefix for WordPress tables

ଯେତେବେଳେ ତୁମେ ତୁମର ସର୍ଭରରେ ଓଡ଼ିଏସ୍ ସଂସ୍ଥାପନ କର, ସଂସ୍ଥାପନ ସ୍ତ୍ରୀ ସମସ୍ତ ଓଡ଼ିଏସ୍ MySQL ଟେବୁଲ୍ ନାମଗୁଡ଼ିକ ଆଗରେ ଏକ ଉପସର୍ଗ ରଖିବ । ଏହି ଉପସର୍ଗଟି ପୂର୍ବନିର୍ଦ୍ଧାରିତ ଭାବରେ 'wp\_' କୁ ସେଟ୍ ହୋଇଛି । ଉଦାହରଣ ସ୍ୱରୂପ ଓଡ଼ିଏସ୍ ପୋଷ୍ଟ୍ ସାରଣୀକୁ `wp_posts` କୁହାଯିବ । ଟେବୁଲ୍ ଉପସର୍ଗ ପରିବର୍ତ୍ତନ କରି ଆପଣ ଅନ୍ଧକାର ଦ୍ୱାରା କିଛି ସୁରକ୍ଷା ସୃଷ୍ଟି କରିପାରିବେ । ଏହି ଉପାୟରେ ଯେତେବେଳେ ଜଣେ ହ୍ୟାକର୍ SQL ଇଞ୍ଜିନ୍ୟାନ୍ ଆକ୍ରମଣ କରିବାକୁ ଚେଷ୍ଟା କରେ, ସେମାନଙ୍କୁ କେବଳ 'wp\_' ବ୍ୟବହାର କରିବା ପରିବର୍ତ୍ତେ ତୁମର ଟେବୁଲର ଉପସର୍ଗ ଅନୁମାନ କରିବାକୁ ପଡ଼ିବ । ଆପଣ ଯାହା ପସନ୍ଦ କରନ୍ତି ତାହା କରିବାକୁ ଆପଣ ଏହି ଉପସର୍ଗ ସେଟ୍ କରିପାରିବେ ।

### Set Prefix in New WordPress Installation

If using famous 5 minute installation Change prefix in field during installation.



Below you should enter your database connection details. If you're not sure about these, contact your host.

<b>Database Name</b>	<input type="text" value="wordpress"/>	The name of the database you want to use with WordPress.
<b>Username</b>	<input type="text" value="username"/>	Your database username.
<b>Password</b>	<input type="text" value="password"/>	Your database password.
<b>Database Host</b>	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
<b>Table Prefix</b>	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Submit

If installing via WordPress CLI use the following command:

```
// set other variables above, or substitute your strings in.  
WP_DBPREFIX=foo  
wp core config --dbname="$MYSQL_DBNAME" --dbuser="$MYSQL_USERNAME" --dbpass="$MYSQL_PASS" --  
dbprefix="$WP_DBPREFIX_" --locale=en_AU
```

### Change Prefix in Existing Installation

Changing the prefix is a little more difficult. Firstly use a FTP program like FileZilla to edit the wp-config.php file. Change the entry `$table_prefix = 'wp_';` to `$table_prefix = 'foo_';` substituting 'foo' for your desired prefix.

Next we'll need to edit the database. If you have access to phpMyAdmin, login and do the following:

- Select the WordPress database

<input type="checkbox"/> wp_links	Browse	Structure	
<input type="checkbox"/> wp_options	Browse	Structure	
<input type="checkbox"/> wp_postmeta	Browse	Structure	
<input type="checkbox"/> wp_posts	Browse	Structure	

- Select all tables and in the dropdown select replace table prefix.

Check All
   
 Print view Data Dict
   
 Create table
   
 Name:

With selected:
 

- With selected:
- Export
- Print view
- Empty
- Drop
- Check table
- Optimize table
- Repair table
- Analyze table
- Add prefix to table
- Replace table prefix
- Copy table with prefix

- In "From" type 'wp\_'. In "To" type your prefix, 'foo\_' in this example and press "Submit".

**Replace table prefix:**

From

To

- Tables should now look like this:

<input type="checkbox"/> foo_links	Browse	Structure	
<input type="checkbox"/> foo_options	Browse	Structure	
<input type="checkbox"/> foo_postmeta	Browse	Structure	
<input type="checkbox"/> foo_posts	Browse	Structure	

If you can't use phpMyAdmin then use the following MySQL command:

```
RENAME table `wp_comments` TO `foo_comments`
```



































You'll need to run that command for each table, substituting 'comments' for the other table names.

Next we need to change a few entries in some tables. Run this query on the 'foo\_options' table

```
SELECT * FROM foo_options WHERE option_name LIKE '%user_roles%'
```

A entry with option\_name of 'wp\_user\_roles' should appear. In that entry change the 'option\_name' entry from wp\_user\_roles to foo\_user\_roles.

Then open up 'foo\_usermeta' table and find every entry with 'wp\_' at the front.

 Edit  Copy  Delete	10	1	wp_capabilities	a:1:{s
 Edit  Copy  Delete	26	2	wp_capabilities	a:1:{s
 Edit  Copy  Delete	75	3	wp_capabilities	a:1:{s
 Edit  Copy  Delete	14	1	wp_dashboard_quick_press_last_post_id	2595
 Edit  Copy  Delete	29	2	wp_dashboard_quick_press_last_post_id	1283
 Edit  Copy  Delete	11	1	wp_user_level	10
 Edit  Copy  Delete	27	2	wp_user_level	10
 Edit  Copy  Delete	76	3	wp_user_level	0
 Edit  Copy  Delete	15	1	wp_user-settings	advl
 Edit  Copy  Delete	41	2	wp_user-settings	editor
 Edit  Copy  Delete	16	1	wp_user-settings-time	14753
 Edit  Copy  Delete	42	2	wp_user-settings-time	14134

and change it to 'foo\_'. The number of entries you must change will depend on how many users you have.

<input type="checkbox"/>	 Edit	 Copy	 Delete	10	1	foo_capabilities	a:1
<input type="checkbox"/>	 Edit	 Copy	 Delete	26	2	foo_capabilities	a:1
<input type="checkbox"/>	 Edit	 Copy	 Delete	75	3	foo_capabilities	a:1
<input type="checkbox"/>	 Edit	 Copy	 Delete	14	1	foo_dashboard_quick_press_last_post_id	25
<input type="checkbox"/>	 Edit	 Copy	 Delete	29	2	foo_dashboard_quick_press_last_post_id	12
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	1	foo_user_level	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	27	2	foo_user_level	10
<input type="checkbox"/>	 Edit	 Copy	 Delete	76	3	foo_user_level	0
<input type="checkbox"/>	 Edit	 Copy	 Delete	15	1	foo_user-settings	ad
<input type="checkbox"/>	 Edit	 Copy	 Delete	41	2	foo_user-settings	ed
<input type="checkbox"/>	 Edit	 Copy	 Delete	16	1	foo_user-settings-time	14
<input type="checkbox"/>	 Edit	 Copy	 Delete	42	2	foo_user-settings-time	14

That should be all you need to change the prefix in an existing installation

# Chapter 50: Security in WordPress - Escaping

## Syntax

- `esc_html( string $text )`
- `esc_url( string $url, array $protocols, string $_context )`
- `esc_js( string $text )`
- `wp_json_encode( mixed $data, int $options, int $depth = 512 )`
- `esc_attr( string $text )`
- `esc_textarea( string $text )`

## Remarks

ବିକାଶ ସମୟରେ ସୁରକ୍ଷା ସର୍ବଦା ଧ୍ୟାନରେ ରଖିବା ଉଚିତ୍ । ସୁରକ୍ଷା ବିନା ଏକ ଆପ୍ ବିଭିନ୍ନ ଆକ୍ରମଣ ପାଇଁ ଖୋଲା ଅଛି ଯେପରିକି SQL ଇଞ୍ଜେକ୍ସନ୍, XSS, CSRF, RFI ଇତ୍ୟାଦି ଯାହା ଗୁରୁତର ସମସ୍ୟାର କାରଣ ହୋଇପାରେ ।

ଅବିଶ୍ୱାସନୀୟ ତଥ୍ୟ ଅନେକ ଉତ୍ସରୁ ଆସିଥାଏ (ଉପଭୋକ୍ତା, ତୃତୀୟ ପକ୍ଷ ସାଇଟ୍, ଆପଣଙ୍କର ନିଜ ତାତାବେସ୍, ...) ଏବଂ ଏହାର ସମସ୍ତ ଇନପୁଟ୍ ଏବଂ ଆଉଟପୁଟ୍ ଉପରେ ବ `valid` ଧ ହେବା ଆବଶ୍ୟକ । (ଉତ୍ସ: ଓର୍ଡ଼ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ବ୍ୟବହାର ଏବଂ ଉଦ୍ଦେଶ୍ୟ ଉପରେ ନିର୍ଭର କରି ତଥ୍ୟ ବ `valid` ଧ, ପରିମଳ କିମ୍ବା ପଲାର୍ନ କରାଯିବା ଉଚିତ୍ ।

ବ `valid` ଧତା ହେଉଛି ଉପଭୋକ୍ତାଙ୍କୁ ଆପଣ ଅନୁରୋଧ କରିଥିବା ତଥ୍ୟ ଯାହା ସେମାନେ ଦାଖଲ କରିଛନ୍ତି ତାହା ମେଳ ଖାଉଛି । (ଉତ୍ସ: ଓର୍ଡ଼ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ଉପଭୋକ୍ତା ତଥ୍ୟ ଗ୍ରହଣ କରିବା ପାଇଁ ସାନ୍ନିଚାଇଜେସନ୍ ଟିକେ ଅଧିକ ଉଦାର ଅଟେ । ଯେତେବେଳେ ଗ୍ରହଣୀୟ ଇନପୁଟ୍ ପରିସର ଅଛି, ଆମେ ଏହି ପଦ୍ଧତିଗୁଡ଼ିକ ବ୍ୟବହାର କରିବାକୁ ଫେରିଯିବା । (ଉତ୍ସ: ଓର୍ଡ଼ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ଖସିଯିବା ହେଉଛି ଆପଣଙ୍କ ପାଖରେ ଥିବା ତଥ୍ୟକୁ ନେବା ଏବଂ ଶେଷ ଉପଭୋକ୍ତା ପାଇଁ ଏହାକୁ ପ୍ରଦାନ କରିବା ପୂର୍ବରୁ ଏହାକୁ ସୁରକ୍ଷିତ କରିବାରେ ସାହାଯ୍ୟ କରିବା । (ଉତ୍ସ: ଓର୍ଡ଼ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

## Examples

### escape data in HTML code

`esc_html` should be used anytime we're outputting data inside HTML code.

```
<h4><?php echo esc_html( $title ); ?></h4>
```

### escape a url

```
<a href="<?php echo esc_url( home_url( '/' ) ); ?>">Home</a>
```

```

```

### escape data in js code

`esc_js()` is intended to be used for inline JS, inside a tag attribute.

For data inside a `<script>` tag use `wp_json_encode()`.

```
<input type="text" onfocus="if( this.value == '<?php echo esc_js( $fields['input_text'] ); ?>' ) { this.value = ''; }" name="name">
```

`wp_json_encode()` encodes a variable into JSON, with some sanity checks.

Note that `wp_json_encode()` includes the string-delimiting quotes automatically.

```
<?php
$book = array(
    "title" => "JavaScript: The Definitive Guide",
    "author" => "Stack Overflow",
);
?>
<script type="text/javascript">
var book = <?php echo wp_json_encode($book) ?>;
/* var book = {
    "title": "Security in WordPress",
    "author" => "Stack Overflow",
}; */
</script>
```

or

```
<script type="text/javascript">
    var title = <?php echo wp_json_encode( $title ); ?>;
    var content = <?php echo wp_json_encode( $content ); ?>;
    var comment_count = <?php echo wp_json_encode( $comment_count ); ?>;
</script>
```

## escape attributes

```
<input type="text" value="<?php echo esc_attr($_POST['username']); ?>" />
```

## escape data in textarea

```
<textarea><?php echo esc_textarea( $text ); ?></textarea>
```

---

# Chapter 51: Security in WordPress - Sanitization

## Syntax

- `sanitize_text_field( string $str )`
- `sanitize_title( string $title, string $fallback_title, string $context )`
- `sanitize_email( string $email )`
- `sanitize_html_class( string $class, string $fallback )`
- `sanitize_file_name( string $name )`
- `sanitize_user( string $username, boolean $strict )`

## Remarks

ବିକାଶ ସମୟରେ ସୁରକ୍ଷା ସର୍ବଦା ଧ୍ୟାନରେ ରଖିବା ଉଚିତ୍ । ସୁରକ୍ଷା ବିନା ଏକ ଆପ୍ ବିଭିନ୍ନ ଆକ୍ରମଣ ପାଇଁ ଖୋଲା ଅଛି ଯେପରିକି SQL ଇଞ୍ଜେକ୍ସନ୍, XSS, CSRF, RFI ଇତ୍ୟାଦି ଯାହା ଗୁରୁତର ସମସ୍ୟାର କାରଣ ହୋଇପାରେ ।

ଅବିଶ୍ୱାସନୀୟ ତଥ୍ୟ ଅନେକ ଉତ୍ସରୁ ଆସିଥାଏ (ଉପଭୋକ୍ତା, ତୃତୀୟ ପକ୍ଷ ସାଇଟ୍, ଆପଣଙ୍କର ନିଜ ତାଟାବେସ୍!, ...) ଏବଂ ଏହାର ସମସ୍ତ ଇନପୁଟ୍ ଏବଂ ଆଉଟପୁଟ୍ ଉପରେ ବ `valid` ଧ ହେବା ଆବଶ୍ୟକ । (ସୋର୍ସ: ୱାଉଚ୍ ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ବ୍ୟବହାର ଏବଂ ଉଦ୍ଦେଶ୍ୟ ଉପରେ ନିର୍ଭର କରି ତଥ୍ୟ ବ `valid` ଧ, ପରିମଳ କିମ୍ବା ପଲାର୍ସନ କରାଯିବା ଉଚିତ୍ ।

ବ `valid` ଧତା ହେଉଛି ଉପଭୋକ୍ତାଙ୍କୁ ଆପଣ ଅନୁରୋଧ କରିଥିବା ତଥ୍ୟ ଯାହା ସେମାନେ ଦାଖଲ କରିଛନ୍ତି ତାହା ମେଳ ଖାଉଛି । (ସୋର୍ସ: ୱାଉଚ୍ ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ଉପଭୋକ୍ତା ତଥ୍ୟ ଗ୍ରହଣ କରିବା ପାଇଁ ସାନ୍ନିତ୍ୟାଭେଦ ସନ୍ ଟିକେ ଅଧିକ ଉଦାର ଅଟେ । ଯେତେବେଳେ ଗ୍ରହଣୀୟ ଇନପୁଟ୍ ପରିସର ଅଛି, ଆମେ ଏହି ପଦ୍ଧତିଗୁଡ଼ିକ ବ୍ୟବହାର କରିବାକୁ ଫେରିଯିବା । (ସୋର୍ସ: ୱାଉଚ୍ ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

ଖସିଯିବା ହେଉଛି ଆପଣଙ୍କ ପାଖରେ ଥିବା ତଥ୍ୟକୁ ନେବା ଏବଂ ଶେଷ ଉପଭୋକ୍ତା ପାଇଁ ଏହାକୁ ପ୍ରଦାନ କରିବା ପୂର୍ବରୁ ଏହାକୁ ସୁରକ୍ଷିତ କରିବାରେ ସାହାଯ୍ୟ କରିବା । (ସୋର୍ସ: ୱାଉଚ୍ ପ୍ରେସ୍ କୋଡେକ୍ସ୍)

## Examples

### Sanitize text field

```
$title = sanitize_text_field( $_POST['title'] );
```

### Sanitize title

The returned value is intended to be suitable for use in a URL, not as a human-readable title. Use `sanitize_text_field` instead.

```
$new_url = sanitize_title($title);
```

## Sanitize email

```
$sanitized_email = sanitize_email(' admin@example.com! ');
```

## Sanitize html class

```
$post_class = sanitize_html_class( $post->post_title );  
echo '<div class="' . $post_class . '>';
```

## Sanitize file name

```
$incfile = sanitize_file_name($_REQUEST["file"]);  
include($incfile . ".php");
```

Without sanitizing the file name an attacker could simple pass [http://attacker\\_site/malicious\\_page](http://attacker_site/malicious_page) as input and execute whatever code in your server.

## Sanitize user name

```
$user = sanitize_user("attacker username<script>console.log(document.cookie)</script>");
```

\$user value after sanitize is "attacker username"

# Chapter 52: Shortcode

## Examples

### Registering shortcode

ସର୍ଟକୋଡ୍ ହେଉଛି ଏକ ଛୋଟ ସଂକେତ ଯାହାକି ସ୍ୱତନ୍ତ୍ର ସ୍ୱତନ୍ତ୍ର ରେ ଯୋଡ଼ିହେବ ଏବଂ ପୃଷ୍ଠା ପ୍ରକାଶିତ ହେବା କିମ୍ବା ପୂର୍ବାବଲୋକନ ହେବା ପରେ କିଛି ଅଲଗା ହେବ ।

ବାରମ୍ବାର, ଥୁମ୍ ଫଙ୍କସନ୍ । Php ଫାଇଲରେ ସର୍ଟକୋଡ୍ ଯୋଡ଼ା ଯାଇଥାଏ, କିନ୍ତୁ ଏହା ଏକ ଭଲ ଅଭ୍ୟାସ ନୁହେଁ କାରଣ ଥୁମ୍ ବଦଳାଇବା ପରେ ସର୍ଟକୋଡ୍ କାର୍ଯ୍ୟ ଜାରି ରଖିବ ବୋଲି ଆଶା କରାଯାଏ । ଏହା ପରିବର୍ତ୍ତେ, ଏହି କାର୍ଯ୍ୟକାରୀତାକୁ ଯୋଡ଼ିବା ପାଇଁ ଏକ ସ୍ୱଗତ୍ତ୍ୱ ଲେଖନ୍ତୁ ।

The structure for registering shortcode is:

```
function new_shortcode($atts, $content = null){
    // if parameters are needed in the shortcode
    // parameters can be set to default to something
    extract( shortcode_atts( array(
        'param_one' => 'h1'
    ), $atts ) );
    $shortcode = '<'. $param_one '>'. $content . '</'. $param_one . '>';
    return $shortcode;
}
// this is what registers the shortcode with wordpress
add_shortcode('demo-shortcode', 'new_shortcode');
```

Inside the WordPress editor, you can type:

```
[demo-shortcode param_one="h2"]Demo[/demo-shortcode]
// you don't need to insert param_one into the editor if it has a default value.
// having it in the editor will override the default
```

Once the page is published, this will turn into

```
<h2>Demo</h2>
```

### Using Shortcodes in WordPress Backend

```
[footag foo="value of 1" attribute-2="value of 2"]
```

In wordpress admin we use pre defined shortcodes by writing the shortcode name inside square brackets and optionally adding attributes to it separating by space.

### Adding New Shortcodes

```
function footag_func( $atts ) {
    return "foo = {$atts['foo']}";
}
```

```
}  
add_shortcode( 'footag', 'footag_func' );
```

In plugins we can add shortcodes using the `add_shortcode` function.

The shortcode can be used in any Wordpress page or post just by enclosing it in square brackets.

```
[footag]
```

## Using Shortcodes Inside PHP Code (themes and plugins)

```
<?php echo do_shortcode("[footag foo='Hi! I am a foo output']"); ?>
```

To print a shortcode using php use the `do_shortcode` function and echo the returned value.

## Using Shortcodes in Widgets

```
add_filter( 'widget_text', 'shortcode_unautop' );  
add_filter( 'widget_text', 'do_shortcode' );
```

Add this to a plugin or the `functions.php` file to enable shortcodes in widgets. The code first stops WordPress turning line breaks into paragraph tags and then lets shortcodes to parse for widgets. The order of the two lines is important.

---

# Chapter 53: Shortcode with attribute

## Syntax

- `add_shortcode('your_short_code', 'your_function_name');`

## Parameters

Parameters	Discription and usage
\$tag	(string) (required) Shortcode tag to be searched in post content Default: None
\$func	(callable) (required) Hook to run when shortcode is found Default: None

## Remarks

IMPORTANT – Don't use camelCase or UPPER-CASE for your attributes

You can generate a shortcode with attribute [Here](#)

## Examples

### Examples of Shortcodes

WordPress shortcodes were introduced in 2.5

Here is come example of shortcode

```
[button]
```

to use shortcode direct into theme you have to use `do_shortcode()`

```
<?php echo do_shortcode('[button]'); ?>
```

To customize the button, we could simply add something like:

```
[button type="twitter"]
```

Or to make it even better, we could use an enclosing shortcode:

```
[button type="twitter"]Follow me on Twitter![/button]
```

### Creating a self-closing shortcode

The simplest shortcode is the self-closing one. We're going to create a simple link to our Twitter account, and then add it in a blog post. All the code goes in `functions.php`, which is located in `/wp-content/themes/your-theme/`. If you don't have one, just create it and put the code in it.

```
<?php
function button_shortcode() {
return '<a href="http://twitter.com/rupomkhondaker" class="twitter-button">Follow me on
Twitter!</a>';
}
add_shortcode('button', 'button_shortcode');
?>
```

Usage: `[button]`

## Creating a self-closing shortcode with parameters

Creating a self-closing shortcode with parameters

```
<?php
function button_shortcode( $type ) {

    extract( shortcode_atts(
        array(
            'type' => 'value'
        ), $type ) );

    // check what type user entered
    switch ( $type ) {

        case 'twitter':
            return '<a href="http://twitter.com/rupomkhondaker" class="twitter-button">Follw
me on Twitter!</a>';
            break;

        case 'rss':
            return '<a href="http://example.com/rss" class="rss-button">Subscribe to the
feed!</a>';
            break;

    }
}
add_shortcode( 'button', 'button_shortcode' );
?>
```

Now you can choose what button to display by defining type in your shortcode.

```
[button type="twitter"]
[button type="rss"]
```

## Creating an enclosing shortcode

enclosing shortcode

The enclosing shortcode allows you to embed content within your shortcode, just like BBCode if you've ever used that.

```

<?php
function button_shortcode( $attr, $content = null ) {
return '<a href="http://twitter.com/filipstefansson" class="twitter-button">' . $content .
'</a>';
}
add_shortcode('button', 'button_shortcode');
?>

```

To use this shortcode, you embedd the text you want to use like this:

```
[button]Follow me on Twitter![/button]
```

To make this button even better, we could add parameters just like we did in the previous example.

```

<?php
function button_shortcode( $atts, $content = null ) {
extract( shortcode_atts( array(
'account' => 'account',
'style' => 'style'
), $atts ) );
return '<a href="http://twitter.com/' . esc_attr($account) . '" class="twitter-button ' .
esc_attr($style) . '">' . $content . '</a>';
}
add_shortcode('button', 'button_shortcode');
?>

```

Usage:

```
[button account="rupomkhondaker" style="simple"]Follow me on Twitter![/button]
```

## Shortcodes in Widgets

By default, WordPress does not support shortcodes within Sidebar Widgets. It only expands the shortcodes within the content of a Post, Page, or custom post type. To add shortcode support to sidebar widgets, you can install a plugin, or use the below code:

```

add_filter( 'widget_text', 'shortcode_unautop' );
add_filter( 'widget_text', 'do_shortcode' );

```

It is important that these lines be added in this order. The first line prevents WordPress from turning line breaks into paragraph tags, since this keeps shortcodes from working. The second line is the one that makes the shortcodes work.

# Chapter 54: Shortcodes

## Examples

### Shortcode introduction

ଯେତେବେଳେ ଆପଣ ସାଧାରଣ ବିଷୟବସ୍ତୁ ସମ୍ପାଦକରେ ଅଧିକ ଜଟିଳ ଉପାଦାନଗୁଡ଼ିକ ଇନଲାଇନ ଯୋଗ କରିବାକୁ ସମର୍ଥ ହେବାକୁ ଚାହାଁନ୍ତି, ସର୍ଟକୋଡ୍ ଉପଯୋଗୀ |(Shortcodes are useful when you want to be able add more complex elements inline into the normal content editor.)

A shortcode in it's simplest form would look like this:

```
function my_shortcode( ){
    return "This is a shortcode";
}
add_shortcode( 'my_shortcode', 'my_shortcode' );
```

It would output the text "This is a shortcode" and you would use it by writing [my\_shortcode] in the content editor.

### Button shortcode

Here is an example of a simple button short code:

```
<?php
function my_button_shortcode( $atts ) {

    // Parse the input attributes and assign default values for the
    // attributes that are not specified on the shortcode
    $a = shortcode_atts( array(
        'id' => '',
        'url' => '#',
        'class' => '',
        'text' => ''
    ), $atts );

    // Open the anchor tag and add role=button for better accessibility
    $btn_html = '<a role="button"';

    // Add the href(link) attribute
    $btn_html .= ' href="' . $a['url'] . '"';

    // Add id attribute to output if specified
    if ( !empty( $a['id'] ) ) {
        $btn_html .= ' id="' . $a['id'] . '"';
    }

    $btn_classes = 'button';

    // Add class attribute to output
    $btn_html .= ' class="button ' . ( !empty( $a['class'] ) ? $a['class'] : '' ) . '"';

    // Close opening anchor tag
```

```
$btn_html .= '>'.  
    // Add button text  
    $a['text'].  
    // Add closing anchor tag  
    '</a>'. "\n";  
  
    return $btn_html;  
}  
add_shortcode( 'button', 'my_button_shortcode' );
```

This shortcode can be used by typing `[button url="/my-other-page" id="my-other-page-button" class="dark" text="Click me!"]` into the editor and will output the following HTML:

```
<a role="button" href="/my-other-page" id="my-other-page-button"  
class="button dark">Click me!</a>
```

# Chapter 55: Sidebars

## Syntax

- register\_sidebar( \$args )
- get\_sidebar(string \$name = null)

## Parameters

Parameter	Details
\$args	(string   array) (Optional) Builds sidebar based on the name and id values
\$name	*(string) (Optional) The name of the specialised sidebar. Default value: null

## Remarks

Argument options are:

- ନାମ - ସାଇଡ୍ ବାର ନାମ (ଡିଫଲ୍ଟ: ଲୋକାଲ 'Sidebar' ଏବଂ ନ୍ୟୁମେରିକ ID) |
- id - ସାଇଡ୍ ବାର id - କ space ଶସି ସ୍ପେସ୍ ବିନା (ଡିଫଲ୍ଟ: ଏକ ସାଂଖ୍ୟିକ ଅଟୋ-ଇନ୍କ୍ରିମେଣ୍ଟେଡ ID) ଲୋକାଲ କେସରେ ରହିବା ଜରୁରୀ | ଯଦି ଆପଣ id ଆର୍ଗୁମେଣ୍ଟ୍ ମୂଲ୍ୟ ସେଟ୍ କରନ୍ତି ନାହିଁ, ତେବେ ଆପଣ ସଂସ୍କରଣ 4.2 ଠାରୁ ଆରମ୍ଭ କରି ଡିବଗ୍ ମୋଡ୍ ରେ E\_USER\_NOTICE ବାର୍ତ୍ତା ପାଇବେ |
- ବର୍ଣ୍ଣନା - ପାର୍ଶ୍ଵ ପଟି କ'ଣ / କେଉଁଠାରେ ପାଠ୍ୟ ବର୍ଣ୍ଣନା | ଝିଜେଟ୍ ପରିଚାଳନା ସ୍କ୍ରିନରେ ପ୍ରଦର୍ଶିତ | (2.9 ପରଠାରୁ) (ଡିଫଲ୍ଟ: ଖାଲି)
- ଶ୍ରେଣୀ - ଦୃଶ୍ୟପତ୍ତରେ ସାଇଡ୍ ବାର କୁ ନ୍ୟସ୍ତ କରିବାକୁ CSS ଶ୍ରେଣୀ -> ଝିଜେଟ୍ ଆଡମିନ୍ ପୃଷ୍ଠାରେ | ଏହି ଶ୍ରେଣୀ କେବଳ ଝିଜେଟ୍ ପୃଷ୍ଠାରେ ଆଡମିନ୍ ପୃଷ୍ଠାର ଉପରେ ଦେଖାଯିବ | ଏହା ଆପଣଙ୍କର ଝେବସାଇଡ୍ ର ଆଗ ଭାଗରେ ଅନ୍ତର୍ଭୁକ୍ତ ହେବ ନାହିଁ | ଟିପ୍ପଣୀ: ଭାଲ୍ୟୁ ସାଇଡ୍ ବାର ଶ୍ରେଣୀ ମୂଲ୍ୟ ସହିତ ପ୍ରିପେଣ୍ଡ୍ ହେବ | ଉଦାହରଣ ସ୍ଵରୂପ, ଟାଲ୍ ର ଏକ ଶ୍ରେଣୀ ସାଇଡ୍ ବାର-ଟାଲ୍ ର ଏକ ଶ୍ରେଣୀ ମୂଲ୍ୟ ସୃଷ୍ଟି କରିବ | (ଡିଫଲ୍ଟ: ଖାଲି)
- ପୂର୍ବରୁ\_ଝିଜେଟ୍ - ପ୍ରତ୍ୟେକ ଝିଜେଟ୍ ପୂର୍ବରୁ ରଖିବାକୁ HTML (ଡିଫଲ୍ଟ: <li id = "% 1 \$ s" ଶ୍ରେଣୀ = "ଝିଜେଟ୍ | % 2 \$ s ">) ଟିପ୍ପଣୀ: ଭେରିଏବଲ୍ ପ୍ରତିସ୍ଥାପନ ପାଇଁ ସ୍ପ୍ରିଣ୍ଟ୍ ବ୍ୟବହାର କରେ |
- ପରେ\_ଝିଜେଟ୍ - ପ୍ରତ୍ୟେକ ଝିଜେଟ୍ ପରେ ସ୍ଥାନିତ କରିବାକୁ HTML (ଡିଫଲ୍ଟ: </li> \ n) |
- ପୂର୍ବରୁ\_ଟାଇଟଲ୍ - ପ୍ରତ୍ୟେକ ଟାଇଟଲ୍ ପୂର୍ବରୁ HTML ରଖିବାକୁ (ଡିଫଲ୍ଟ: <h2 ଶ୍ରେଣୀ = "ଝିଜେଟ୍ ଟାଇଟଲ୍">) |
- ପରେ\_ଟାଇଟଲ୍ - ପ୍ରତ୍ୟେକ ଟାଇଟଲ୍ ପରେ ସ୍ଥାନିତ କରିବାକୁ HTML (ଡିଫଲ୍ଟ: </h2> \ n)

name - Sidebar name (default: localized 'Sidebar' and numeric ID).

- id - Sidebar id - Must be all in lowercase, with no spaces (default: a numeric auto-incremented ID). If you do not set the id argument value, you will get E\_USER\_NOTICE messages in debug mode, starting with version 4.2.
- description - Text description of what/where the sidebar is. Shown on widget management screen. (Since 2.9) (default: empty)
- class - CSS class to assign to the Sidebar in the Appearance -> Widget admin page. This class will only appear in the source of the WordPress Widget admin page. It will not be included in the front end of your website. Note: The value sidebar will be prepended to the class value. For example, a class of tal will result in a class value of sidebar-tal. (default: empty).
- before\_widget - HTML to place before every widget (default: <li id="%1\$s" class="widget %2\$s">) Note: uses sprintf for variable substitution
- after\_widget - HTML to place after every widget (default: </li>\n).
- before\_title - HTML to place before every title (default: <h2 class="widgettitle">).
- after\_title - HTML to place after every title (default: </h2>\n).

|Examples

## Registering sidebars

In your functions.php you can register new sidebars with this code

/\*\*

```

* Registers sidebars
*
* @param array Array with default or specified array values
* @since      1.0.0
*/
if ( function_exists( 'register_sidebar' ) ) {
    register_sidebar( array (
        'name'          => esc_html__( 'Primary Sidebar', 'mytheme' ),
        'id'            => 'primary-widget-area',
        'description'   => esc_html__( 'The Primary widget Area', 'mytheme' ),
        'before_widget' => '<div id="%1$s" class="widget %2$s">',
        'after_widget'  => '</div>',
        'before_title'  => '<div class="sidebar-widget-heading"><h3>',
        'after_title'   => '</h3></div>',
    ) );

    register_sidebar( array (
        'name'          => esc_html__( 'Secondary Sidebar', 'mytheme' ),
        'id'            => 'secondary-widget-area',
        'description'   => esc_html__( 'The Secondary widget Area', 'mytheme' ),
        'before_widget' => '<div id="%1$s" class="widget %2$s">',
        'after_widget'  => '</div>',
        'before_title'  => '<div class="sidebar-widget-heading"><h3>',
        'after_title'   => '</h3></div>',
    ) );
}

```

You can add as many sidebars as you want to.

## Get Sidebar

You can also create your own sidebar file in the theme to call it on different templates. Copy and paste sidebar.php of current theme and change the name (i.e. sidebar-book.php)

In the template you can call this sidebar using `get_sidebar('book')`. Using this you can call different sidebars on different pages.

(ଏହାକୁ ବିଭିନ୍ନ ଟେମ୍ପଲେଟରେ ଡାକିବା ପାଇଁ ଥିମ୍ ରେ ଆପଣ ନିଜର ସାଇଡ୍ ବାର୍ ଫାଇଲ୍ ମଧ୍ୟ ସୃଷ୍ଟି କରିପାରିବେ । ସାମ୍ପ୍ରତିକ ଥିମ୍ ର sidebar.php କୁ କପି ଏବଂ ଲେଉଟନ କରନ୍ତୁ ଏବଂ ନାମ ପରିବର୍ତ୍ତନ କରନ୍ତୁ (ଯଥା sidebar-book.php)

ଟେମ୍ପଲେଟରେ ଆପଣ `get_sidebar('ବୁକ୍')` ବ୍ୟବହାର କରି ଏହି ପାର୍ଶ୍ୱ ଦଣ୍ଡିକାକୁ କଲ କରିପାରିବେ । ଏହାକୁ ବ୍ୟବହାର କରି ଆପଣ ବିଭିନ୍ନ ପୃଷ୍ଠାରେ ବିଭିନ୍ନ ପାର୍ଶ୍ୱ ଦଣ୍ଡିକାକୁ କଲ କରିପାରିବେ ।)

---

# Chapter 56: Site Migration

## Syntax

- OLD\_SITE - The old site being migrated (eg: <http://localhost/example>)
- NEW\_SITE - The new site to which to migrate (eg: <https://example.com>)

## Examples

### Updating the tables with a new URL

```
UPDATE wp_options SET option_value = replace(option_value, 'OLD_SITE', 'NEW_SITE') WHERE
option_name = 'home' OR option_name = 'siteurl';
UPDATE wp_posts SET guid = replace(guid, 'OLD_SITE','NEW_SITE');
UPDATE wp_posts SET post_content = replace(post_content, 'OLD_SITE', 'NEW_SITE');
```

# Chapter 57: Taxonomies

## Syntax

- `register_taxonomy( $taxonomy, $object_type, $args );`

## Parameters

Parameter	Details
<code>\$taxonomy</code>	<i>(string) (required)</i> The name of the taxonomy. Name should only contain lowercase letters and the underscore character, and not be more than 32 characters long (database structure restriction).
<code>\$object_type</code>	<i>(array/string) (required)</i> Name of the object type for the taxonomy object. Object-types can be built-in Post Type or any Custom Post Type that may be registered.
<code>\$args</code>	<i>(array/string) (optional)</i> An array of Arguments.

## Examples

### Example of registering a taxonomy for genres

```
<?php
// hook into the init action and call create_book_taxonomies when it fires
add_action( 'init', 'create_book_taxonomies', 0 );

// create taxonomy genres 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' ),
        'singular_name' => _x( 'Genre', 'taxonomy singular name' ),
        'search_items'  => __( 'Search Genres' ),
        'all_items'     => __( 'All Genres' ),
        'parent_item'   => __( 'Parent Genre' ),
        'parent_item_colon' => __( 'Parent Genre:' ),
        'edit_item'     => __( 'Edit Genre' ),
        'update_item'   => __( 'Update Genre' ),
        'add_new_item'  => __( 'Add New Genre' ),
        'new_item_name' => __( 'New Genre Name' ),
        'menu_name'     => __( 'Genre' ),
    );

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

```

        'show_admin_column' => true,
        'query_var'         => true,
        'rewrite'           => array( 'slug' => 'genre' ),
    );

    register_taxonomy( 'genre', array( 'book' ), $args );
}
?>

```

You can define custom taxonomies in a theme's `functions.php` template file:

## Add category in page

You can also add some custom created taxonomy into post type page using below code.

```

function add_taxonomies_to_pages() {
    register_taxonomy_for_object_type( 'genre', 'page' );
}
add_action( 'init', 'add_taxonomies_to_pages' );

```

Add above code into your theme's `functions.php` file. Same way you can add custom or default `post_tag` into post type page.

To get pages using custom taxonomy query need to add below code in same file.

```

if ( ! is_admin() ) {
    add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

function category_and_tag_archives( $wp_query ) {
    $my_post_array = array( 'page' );
    if ( $wp_query->get( 'category_name' ) || $wp_query->get( 'cat' ) )
        $wp_query->set( 'post_type', $my_post_array );
}

```

## Add Categories and Tags to Pages and insert them as class into

```

// add tags and categories to pages

function add_taxonomies_to_pages() {
    register_taxonomy_for_object_type( 'post_tag', 'page' );
    register_taxonomy_for_object_type( 'category', 'page' );
}
add_action( 'init', 'add_taxonomies_to_pages' );
if ( ! is_admin() ) {
    add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

function category_and_tag_archives( $wp_query ) {
    $my_post_array = array( 'post', 'page' );

    if ( $wp_query->get( 'category_name' ) || $wp_query->get( 'cat' ) )
        $wp_query->set( 'post_type', $my_post_array );
}

```

```

if ( $wp_query->get( 'tag' ) )
$wp_query->set( 'post_type', $my_post_array );
}

// add tags and categorys as class to <body>

function add_categories_and_tags( $classes = '' ) {
    if( is_page() ) {
        $categories = get_the_category();
        foreach( $categories as $category ) {
            $classes[] = 'category-'. $category->slug;
        }
        $tags = get_the_tags();
        foreach( $tags as $tag ) {
            $classes[] = 'tag-'. $tag->slug;
        }
    }
    return $classes;
}
add_filter( 'body_class', 'add_categories_and_tags' );

```

## Add Categories and Tags to Pages and insert as class into

You can add this code to your custom functions.php file:

```

// add tags and categories to pages

function add_taxonomies_to_pages() {
    register_taxonomy_for_object_type( 'post_tag', 'page' );
    register_taxonomy_for_object_type( 'category', 'page' );
}
add_action( 'init', 'add_taxonomies_to_pages' );

if ( ! is_admin() ) {
    add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

function category_and_tag_archives( $wp_query ) {
    $my_post_array = array('post','page');

    if ( $wp_query->get( 'category_name' ) || $wp_query->get( 'cat' ) )
        $wp_query->set( 'post_type', $my_post_array );

    if ( $wp_query->get( 'tag' ) )
        $wp_query->set( 'post_type', $my_post_array );
}

// add tags and categorys as class to <body>

function add_categories_and_tags( $classes = '' ) {
    if( is_page() ) {
        $categories = get_the_category();
        foreach( $categories as $category ) {
            $classes[] = 'category-'. $category->slug;
        }
        $tags = get_the_tags();
        foreach( $tags as $tag ) {
            $classes[] = 'tag-'. $tag->slug;
        }
    }
    return $classes;
}
add_filter( 'body_class', 'add_categories_and_tags' );

```

```
    }  
  }  
  return $classes;  
}  
add_filter( 'body_class', 'add_categories_and_tags' );
```

Works perfect, tested in **WordPress 4.8**

---

# Chapter 58: Template hierarchy

## Remarks

Plugins for debugging in WordPress:

- <https://wordpress.org/plugins/query-monitor/>
- <https://wordpress.org/plugins/debug-bar/>
- <https://wordpress.org/plugins/debug-bar-console/>
- <https://wordpress.org/plugins/kint-debugger/>
- <https://wordpress.org/plugins/rest-api-console/>

## Examples

### Introduction

ଯେତେବେଳେ ଆପଣ ଏକ ଓର୍ଡ଼ର୍ ପ୍ରେସ୍ ଥିବା ଡିଭାଇସ୍ କରୁଛନ୍ତି, ତାହା ଶିଖିବା ପାଇଁ ଏକ ଗୁରୁତ୍ୱପୂର୍ଣ୍ଣ ଜିନିଷ ହେଉଛି ଥିମ୍ ପାଇଁ ଓର୍ଡ଼ର୍ ପ୍ରେସ୍ ଟେମ୍ପଲେଟ୍ ହାଇରାର୍କି | ଟେମ୍ପଲେଟ୍ ହାଇରାର୍କି ବ୍ୟାଖ୍ୟା କରେ କେଉଁ ଟେମ୍ପଲେଟ୍ ଫାଇଲ୍ ଯାହା ପ୍ରତ୍ୟେକ ଅନୁରୋଧ ପାଇଁ ଲୋଡ୍ ହେବ ଏବଂ କେଉଁ କ୍ରମରେ | ଯଦି ହାଇରାର୍କିରେ ପ୍ରଥମ ଟେମ୍ପଲେଟ୍ ବିଦ୍ୟମାନ ନାହିଁ, ଓର୍ଡ଼ର୍ ପ୍ରେସ୍ ପରବର୍ତ୍ତୀକୁ ଲୋଡ୍ କରିବାକୁ ଚେଷ୍ଟା କରିବ ଏବଂ ଯେପର୍ଯ୍ୟନ୍ତ ଆପଣ index.php ରେ ଶେଷ ନ ହୁଅନ୍ତି |

ଟେମ୍ପଲେଟ୍ କ୍ରମକୁ ବିସ୍ତୃତ ଭାବରେ ବର୍ଣ୍ଣନା କରିବା ପାଇଁ ସର୍ବୋତ୍ତମ ଉପାୟ ହେଉଛି ସମ୍ପୂର୍ଣ୍ଣ ସଂରଚନା ସହିତ ଏକ ପ୍ରତିଛବି ବ୍ୟବହାର କରିବା:

Archive Page



Singular Page

Single Post Page

Static Page



Site Front Page

front-page.php



category-books.php would be used only for the category with the slug book. Another example is page-\$id.php that only targets a page with a specific ID, for example page-41.php would target only the page with the ID 41.

ଟେମ୍ପଲେଟ୍ ପରେ ଯାହା ନିର୍ଦ୍ଦିଷ୍ଟ ପ୍ରକାର କିମ୍ବା ପୋଷ୍ଟକୁ ଟାର୍ଗେଟ୍ କରେ ଆମେ ଜେନେରିକ୍ ଟାଇପ୍ ଟେମ୍ପଲେଟ୍ ଗୁଡ଼ିକୁ ପାଇଥାଉ, ଯେପରି ସମସ୍ତ ଆର୍ଟାଇକ୍ଲ ପେଜ୍ ପାଇଁ archive.php କିମ୍ବା ସମସ୍ତ ପେଜ୍ ପାଇଁ page.php | କିଛି ମନେରଖନ୍ତୁ, ସେଗୁଡ଼ିକ କେବଳ ବ୍ୟବହୃତ ହେବ ଯଦି ସାମ୍ପ୍ରତିକ ପୃଷ୍ଠା କ any ଶସି ଟେମ୍ପଲେଟ୍ ସହିତ ମେଲ ଖାଉ ନାହିଁ ଯାହା କ୍ରମରେ ଉଚ୍ଚ ଅଟେ |

ଶେଷରେ, ଯଦି ଟେମ୍ପଲେଟ୍ ଡିରେକ୍ଟୋରୀରେ ଓର୍ଡ଼ରେସ୍ କ match ଶସି ମେଲକ ଟେମ୍ପଲେଟ୍ ଖୋଜି ପାଇଲା ନାହିଁ, ତେବେ ଶେଷ ଫାଲବ୍ୟାକ୍ ସର୍ବଦା index.php ଅଟେ ଯାହାକି ଓର୍ଡ଼ରେସ୍ ଥିଲେ ଏକମାତ୍ର ଆବଶ୍ୟକୀୟ ଟେମ୍ପଲେଟ୍ ଫାଇଲ୍ |

## Debugging

Its easy to get lost while debugging the hierarchy. You can use PHP's built in command debug\_backtrace.

Put the next snippet inside any template you want to debug and view the generated page:

```
<!--  
<?php print_r( debug_backtrace() ) ?>  
-->
```

# Chapter 59: template\_include

## Parameters

Parameter	Explanation
<code>\$template</code>	Passes one parameter to the filter, <code>\$template</code> is the current path to the appropriate file for the post type as found in the active child theme or parent theme (if no child theme in place or child theme has lower ranked templates. See wordpress template hierarchy for more details).

## Remarks

You must return `$template` even if not modifying. If this confuses you, look at examples where `apply_filter()` has been used in code

You should not set up variables here for use later, there are better hooks for this.

A useful program flow for this filter is:

1. Check `$template` includes our custom post type name --> template hierarchy!!
2. if not, search our plugin for suitable files --> Its better to point to specific files rather than searching through folders for files. More efficient. But completely up to the developer.
3. return the template.

## Examples

### Simple example

ଏହି ଫିଲ୍ଟର ବହୁତ ଉପଯୋଗୀ | ବିକାଶକାରୀଙ୍କ ପାଇଁ ଏକ ସାଧାରଣ ସମସ୍ୟା ହେଉଛି, ସେମାନେ ବିକାଶ କରୁଥିବା ପ୍ଲଗିନଗୁଡ଼ିକରେ ଟେମ୍ପଲେଟ୍ କିପରି ଅନ୍ତର୍ଭୁକ୍ତ କରିବେ |

ଖାତରପ୍ରସ୍ତୁତ wp ହାଇରାକି ବ୍ୟବହାର କରି ସକ୍ରିୟ ଶିଶୁ / ପ୍ୟାରେଣ୍ଟ ଥିମ୍ସ ଉପଯୁକ୍ତ ଟେମ୍ପଲେଟ୍ ଖୋଜିବା ପରେ ତୁରନ୍ତ ଫିଲ୍ଟର ପ୍ରୟୋଗ କରାଯାଏ | Be careful to define when you want to modify the template path. In the below example, the code checks to see if the current page is the single view of our custom post type `cpt`.

Simple enough example to get started with!

```
add_filter('template_include', 'custom_function');

function custom_function($template){

    //change a single post template...
```

```

if( is_singular('cpt') ){
    $template= 'path/to/another/template_file';
}

return $template;
}

```

## More Adv example

```

add_filter('template_include', 'custom_function');

function custom_function($template){

    /*
    *   This example is a little more advanced.
    *   It will check to see if $template contains our post-type in the path.
    *   If it does, the theme contains a high level template e.g. single-cpt.php
    *   If not we look in the plugin parent folder for the file. e.g. single-cpt.php
    */

    //check to see if the post type is in the filename (high priority file)
    //return template if it is!

    global $post;

    if( strpos($template, 'single-'. $post->post_type.'.php' ) !== false && strpos($template,
'archive-'. $post->post_type.'.php' ) !== false ){
        return $template;
    }

    $plugin_path = 'var/etc/wp-content/plugins/plugin'; //include own logic here...

    if( is_singular($post->post_type) && file_exists($plugin_path.'/single-'. $post->
post_type.'.php' ) ){
        $template= $plugin_path.'/single-'. $post->post_type.'.php';
    } elseif ( is_archive($post->post_type) && file_exists($plugin_path.'/archive-'. $post->
post_type.'.php' ) ) {
        $template= $plugin_path.'/archive-'. $post->post_type.'.php';
    }

    return $template;
}

```

---

# Chapter 60: The \$wpdb Object

## Remarks

There are two ways to reference the `$wpdb` object. The first is to use the PHP keyword `global` in order to act on the global instance of the object.

```
global $wpdb;
echo $wpdb->prefix;
// Outputs the prefix for the database
```

The second way to use the `$wpdb` object is to reference PHP's `$GLOBALS` super global variable.

```
echo $GLOBALS['wpdb']->prefix;
// This will also output the prefix for the database
```

The second way is discouraged as it may not be considered the best practice.

## Examples

### Selecting a variable

In the most basic form, it is possible to select a single variable from a table by calling the object's `get_var` method passing in an SQL query.

```
global $wpdb;
$user = $wpdb->get_var( "SELECT ID FROM $wpdb->users WHERE user_email='foo@bar.com'" );
```

It is very important to note that any untrusted values used in queries *must* be escaped in order to protect against attacks. This can be done using the object's `prepare` method.

```
global $wpdb;
$email = $_POST['email'];
$user = $wpdb->get_var(
    $wpdb->prepare( "SELECT ID FROM $wpdb->users WHERE user_email=%s", $email )
);
if( !is_null( $user ) ){
    echo $user;
} else {
    echo 'User not found';
}
```

### Selecting multiple rows

You can use `get_results` to get multiple rows from database.

```
global $wpdb;
```

```
$userTable = $wpdb->prefix."users";  
$selectUser = $wpdb->get_results("SELECT * FROM $userTable");
```

This will show all users list in array.

---

# Chapter 61: The Admin Bar (aka "The Toolbar")

## Remarks

ଝାଡ଼ପ୍ରେସ୍ ଆଡମିନ୍ ଟୁଲ୍ ବାର୍ ସଂସ୍କରଣ 3.1 ରେ ଯୋଡା ଯାଇଥିଲା ଏବଂ ସାଧାରଣ ପ୍ରଶାସନିକ କାର୍ଯ୍ୟଗୁଡ଼ିକର ଲିଙ୍କ ସହିତ ଉପଭୋକ୍ତାଙ୍କର ପ୍ରୋଫାଇଲ୍ ଏବଂ ଅନ୍ୟାନ୍ୟ ଝଡ଼ପ୍ରେସ୍ ସୂଚନା ସହିତ ଲିଙ୍କ ଧାରଣ କରିଥିଲା । ତଥାପି, ଅନେକ ସାଇଟ୍ ମାଲିକ ସମସ୍ତ ଲିଙ୍କ ଇନ୍ ବ୍ୟବହାରକାରୀଙ୍କୁ ଡିଫଲ୍ଟ ଭାବରେ ଟୁଲ୍ ବାର୍ ଦେଖାଇବାକୁ ନାପସନ୍ଦ କରନ୍ତି ଏବଂ / କିମ୍ବା ଏଥିରେ ନିଜସ୍ୱ ବିକଳ୍ପ ଯୋଡ଼ିବାକୁ ଚାହାଁନ୍ତି ।

## Examples

### Removing the Admin Toolbar from all except Administrators

Add the following code to `functions.php` to remove it from everyone except the Administrator user level:

```
add_action('after_setup_theme', 'no_admin_bar');

function no_admin_bar() {
    if (!current_user_can('administrator') && !is_admin()) {
        show_admin_bar(false);
    }
}
```

### Removing the Admin toolbar using filters

Another way to hide admin bar is to add

```
if ( !current_user_can( 'manage_options' ) ) {
    add_filter( 'show_admin_bar', '__return_false' , 1000 );
}
```

The users who don't have privileges to access Settings page, won't be able to see the admin bar.

### How to Remove WordPress Logo From Admin Bar

Developers can use **admin\_bar\_menu** action to remove items from WordPress admin bar or toolbar.

```
add_action('admin_bar_menu', 'remove_wp_logo_from_admin_bar', 999);
function remove_wp_logo_from_admin_bar( $wp_admin_bar ) {
    $wp_admin_bar->remove_node('wp-logo');
}
```

Above code removes WordPress logo from the admin bar. All you need to do is paste the code

inside your functions.php file.

```
<li id="wp-admin-bar-wp-logo" class="menupop"> ... </li>
```

Remove\_node ପଦ୍ଧତିକୁ ପାସ୍ ହୋଇଥିବା ପାରାମିଟର ହେଉଛି ଆପଣ ଅପସାରଣ କରିବାକୁ ଚାହୁଁଥିବା ନୋଡର ID | ଝାଡ଼଼ପ୍ରେସ୍ ପୃଷ୍ଠାର HTML ଉତ୍ତ କୋଡ୍ ରେ ଏକ ଟୁଲ୍ ବାର୍ ସହିତ ID ମିଳିପାରିବ | ଉଦାହରଣ ସ୍ଵରୂପ, ଟୁଲ୍ ବାର୍ ରେ ବାମ ପାର୍ଶ୍ଵରେ ଥିବା ଝାଡ଼଼ପ୍ରେସ୍ ଲୋଗୋ ପାଇଁ li ଉପାଦାନର ID ହେଉଛି “wp-admin-bar-wp-logo”:

Remove "wp-admin-bar-" from the li's ID to get the ID of node. From this example the node ID is "wp-logo".

You can use browser inspector tools to find out the node ID's of various items or nodes on your admin bar.

## Add your custom logo and custom link on admin login page

You can add below hooks to add your own logo and link to replace default wordpress logo.

### To add custom logo

```
function custom_login_logo() {
    echo '<style type="text/css">
    h1 a { background-image: url('.get_bloginfo('template_directory').'/images/custom-logo.png)
    !important; background-size : 100% !important; width: 300px !important; height : 100px
    !important;}
    </style>';
}
add_action('login_head', 'custom_login_logo');
```

### To add custom logo link

```
add_filter( 'login_headerurl', 'custom_loginlogo_url' );
function custom_loginlogo_url($url) {
    return home_url();
}
```

# Chapter 62: The Loop (main WordPress loop)

## Examples

### Basic WordPress loop structure

Each time WordPress loads the page, it will run *main loop*.

The loop is the way to iterate over all elements related to the page you are currently on.

ମୁଖ୍ୟ ଲୁପ୍ ଏକ ବିଶ୍ୱସ୍ତରୀୟ WP\_Query ବସ୍ତୁ ଉପରେ କାମ କରିବ । ଜିଜ୍ଞାସାର ଏକ ଜଗତୀକରଣ ପଦ୍ଧତି have\_posts () ଅଛି, ଯାହା ଆମକୁ ସମସ୍ତ ଫଳାଫଳ ମାଧ୍ୟମରେ ଲୁପ୍ କରିବାକୁ ଅନୁମତି ଦିଏ । ଶେଷରେ ଲୁପ୍ ଭିତରେ ଆପଣ\_ପୋଷ୍ଟ () ପଦ୍ଧତିକୁ (ଏକ ଗ୍ଲୋବାଲ୍ ଫଙ୍କସନ୍ ଭାବରେ) ମଧ୍ୟ କଲ୍ କରିପାରିବେ, ଯାହା ଲୁପ୍ ଭିତରେ ସାମ୍ପ୍ରତିକ ପୋଷ୍ଟକୁ ଗ୍ଲୋବାଲ୍ ପୋଷ୍ଟ ଅବଜେକ୍ଟ ସେଟ୍ କରେ ଏବଂ ପୋଷ୍ଟଡାଟାକୁ ସାମ୍ପ୍ରତିକ ପୋଷ୍ଟରେ ସେଟ୍ କରେ । ଏଥିପାଇଁ ଧନ୍ୟବାଦ ଆପଣ ସିଧାସଳଖ ଲୁପ୍ ଭିତରେ \_ ଟାଇଟଲ୍, \_କଣ୍ଟେଣ୍ଟ୍, \_ଆଉଟର୍ (ଟେମ୍ପଲେଟ୍ ଚ୍ୟାଗ) ପରି ଫଙ୍କସନ୍ କଲ୍ କରିପାରିବେ ।

ଉଦାହରଣ ସ୍ୱରୂପ ଯଦି ଆପଣ ପୋଷ୍ଟ ତାଲିକାରେ ଅଛନ୍ତି, ମୁଖ୍ୟ ଲୁପ୍ ସମସ୍ତ ପୋଷ୍ଟ ସହିତ ଏକ ଜିଜ୍ଞାସା ବସ୍ତୁ ଧାରଣ କରିବ ।  
If you are on single post (or page), it will contain a query with single post (page) you are currently on.

```
if ( have_posts() ) :
    while ( have_posts() ) :
        the_post();
        var_dump( $post );
    endwhile;
endif;
```

### Alternative loop syntax

You can also use loop with curly brackets like this:

```
if ( have_posts() ) {
    while ( have_posts() ) {

        the_post();
        var_dump( $post );

    }
}
```

### Handling no items in the loop

If you want to handle such scenario just add an if/else statement.

```
if ( have_posts() ) : while ( have_posts() ) :

    the_post();
    var_dump( $post );
```

```
endwhile; else :  
    __('This Query does not have any results');  
endif;
```

---

# Chapter 63: the\_title()

## Introduction

This function returns the title of the current post.

## Syntax

- `the_title( $before, $after, $echo );`

## Parameters

Parameter	Details
<code>\$before</code>	(string) (optional) Text to place before the title.
<code>\$after</code>	(string) (optional) Text to place after the title.
<code>\$echo</code>	(Boolean) (optional) Display the title or return it for use in PHP

## Remarks

ପାରାମିଟର \$ ଇକୋ ପାଇଁ, ଟାଇଟଲ୍ ପ୍ରଦର୍ଶନ କରିବାକୁ ସତ୍ୟ ବ୍ୟବହାର କରନ୍ତୁ ଏବଂ PHP ରେ ବ୍ୟବହାର ପାଇଁ ଏହାକୁ ଫେରାଇବା ପାଇଁ ମିଥ୍ୟା ବ୍ୟବହାର କରନ୍ତୁ |

• ଦୟାକରି ଧ୍ୟାନ ଦିଅନ୍ତୁ ଯେ\_ ଟାଇଟଲ୍ କେବଳ ଲୁପ୍ ରେ ବ୍ୟବହୃତ ହୋଇପାରିବ, ଯଦି ଆପଣ ଏହାକୁ ଲୁପ୍ ବାହାରେ ବ୍ୟବହାର କରିବାକୁ ଚାହୁଁଛନ୍ତି, ଦୟାକରି `get_the_title` ବ୍ୟବହାର କରନ୍ତୁ |

## Examples

### Simple use of the\_title

#### Code

```
the_title( );
```

#### Output

The title of the current post or page

### Printing the title with code before and after

#### Code

```
the_title( '<h1>', '</h1>' );
```

## Output

The title of the current post or page in h1 tags

# Chapter 64: Themes

## Introduction

ଓର୍ଡ଼ପ୍ରେସ୍ ଥିମ୍ ଗୁଡ଼ିକ ହେଉଛି ଆପଣଙ୍କର ୱେବସାଇଟ୍ ର ଆଗ ଭାଗ । ଲୋକମାନେ ଯେତେବେଳେ ସାଇଟ୍ ପରିଦର୍ଶନ କରନ୍ତି ସେମାନେ ତାହା ଦେଖନ୍ତି । ପେଡ଼ ଏବଂ ମାଗଣା ସଂସ୍କରଣରୁ ଚୟନ କରିବାକୁ ହଜାରେ ଥିମ୍ ଅଛି । ଆପଣ କେବଳ ଦୁଇଟି ଆବଶ୍ୟକୀୟ ଫାଇଲ୍ ସହିତ ନିଜର କଷ୍ଟମ୍ ଥିମ୍ ସୃଷ୍ଟି କରିପାରିବେ ।

## Examples

### WordPress Themes

## How To Choose A Theme

ପ୍ରତ୍ୟେକ ଓର୍ଡ଼ପ୍ରେସ୍ ଇନଷ୍ଟଲ୍ ଏକ ପୂର୍ବ-ସ୍ଥାପିତ ଥିମ୍ ସହିତ ଆସିଥାଏ । ଆପଣ ଡ୍ୟାସବୋର୍ଡ଼ରୁ ଆପଣଙ୍କର ଥିମ୍ ପରିଚାଳନା କରନ୍ତି । ଥିମ୍ ସଂସ୍ଥାପନ, ପୂର୍ବାବଲୋକନ, ବିଲୋପ, ସକ୍ରିୟ ଏବଂ ଅପଡେଟ୍ କରିବାକୁ ଦୃଶ୍ୟ > ଥିମ୍ କୁ ଯାଆନ୍ତୁ । ସାମ୍ପ୍ରତିକ ଥିମ୍ ଏହି ମେନୁର ଉପର ବାମ କୋଣରେ ମିଳିଥାଏ |(ppearance > Themes to install, preview, delete, activate, and update Themes)

ଥିମ୍ ପ୍ରତିଛବି ଉପରେ ଠିକ୍ ଠିକ୍ ଏକ 'ଥିମ୍ ବିବରଣୀ' ବଟନ୍ ପ୍ରକାଶ କରେ । ଏହି ବଟନ୍ ଥିମ୍ ବିଷୟରେ ସୂଚନା ପ୍ରଦାନ କରିଥାଏ, ଯେପରିକି ସଂସ୍କରଣ ଏବଂ ବର୍ଣ୍ଣନା । ସାମ୍ପ୍ରତିକ ଥିମ୍ ଇମେଜ୍ ଉପରେ କ୍ଲିକ୍ କରିବା ଦ୍ୱାରା ଟାଇଟଲ୍ ପରି ନିର୍ଦ୍ଦିଷ୍ଟ ଥିମ୍ ସେଟିଂସମୂହ କଷ୍ଟମାୟାଜ୍ କରିବାକୁ ଆପଣଙ୍କୁ ଆକ୍ସେସ୍ ମିଳିଥାଏ ।

## Update Available

ଯଦି ସଂସ୍ଥାପିତ ଥିମ୍ ପାଇଁ ଅପଡେଟ୍ ଉପଲବ୍ଧ, ତେବେ ଆପଣ ଏକ ସନ୍ଦେଶ ପାଇବେ ଯାହା ଆପଣଙ୍କୁ ଏକ ନୂତନ ସଂସ୍କରଣ ଉପଲବ୍ଧ ବୋଲି ସୂଚିତ କରେ । ଆପଣ ନୂତନ ସଂସ୍କରଣର ବିବରଣୀ ଦେଖିବାକୁ କିମ୍ବା ବର୍ତ୍ତମାନ ଅବସ୍ଥା କରାଯାଇଥିବା ସମ୍ପର୍କ ହେବା ଉଚିତ୍ ।

ସଂସ୍କରଣ ବିବରଣୀ ଦେଖନ୍ତୁ |( View Version Details)

ସଂସ୍କରଣ ବିବରଣୀ ଲିଙ୍କ୍ କ୍ଲିକ୍ କରିବା ଦ୍ୱାରା ଆପଣଙ୍କୁ ଓର୍ଡ଼ପ୍ରେସ୍ ଥିମ୍ ଡିଭିଜନ୍‌ରୁ ଏକ ପୃଷ୍ଠାକୁ ନେଇଗେଟ୍ ହେବ । ଅପଡେଟ୍ ସଂସ୍କରଣ ପାଇଁ ଏଠାରେ ଆପଣ ସବିଶେଷ ତଥ୍ୟ ପାଇବେ ।

ବର୍ତ୍ତମାନ ଅବସ୍ଥା କରନ୍ତୁ |( Update Now)

ବର୍ତ୍ତମାନ ଲିଙ୍କ୍ କ୍ଲିକ୍ କଲେ ଥିମ୍ ଅପଡେଟ୍ ସଂସ୍ଥାପିତ ହେବ । ଥିମ୍ ଗୁଡ଼ିକ ଆଡମିନିଷ୍ଟ୍ରେସନ୍ > ଡ୍ୟାସବୋର୍ଡ଼ > ଅପଡେଟ୍ ସ୍ଥିତ ରୁ ମଧ୍ୟ ଅପଡେଟ୍ ହୋଇପାରିବ ।

ତୁମର ସାମ୍ପ୍ରତିକ ଥିମ୍ ସହିତ, ଥିମ୍ ପରିଚାଳନା ସ୍ଥିତ ଅନ୍ୟ ଥିମ୍ଗୁଡ଼ିକୁ ମଧ୍ୟ ଦର୍ଶାଏ ଯାହା ସଂସ୍ଥାପିତ କିନ୍ତୁ ବର୍ତ୍ତମାନ ନିଷ୍କ୍ରିୟ । ପ୍ରତ୍ୟେକ ଥିମ୍ ଏକ ଛୋଟ ସ୍ଥିନସର୍ ଦ୍ୱାରା ଉପସ୍ଥାପିତ ହୁଏ । ଏହି ପ୍ରତିଛବିଗୁଡ଼ିକ ଉପରେ କ୍ଲିକ୍ ଦ୍ୱାରା 'ଥିମ୍ ବିବରଣୀ', 'ସକ୍ରିୟ', ଏବଂ 'ଲାଇଭ୍ ପୂର୍ବାବଲୋକନ' ବଟନ୍‌ଗୁଡ଼ିକ ପ୍ରକାଶ ପାଇଥାଏ । ଆପଣ ମଧ୍ୟ ଏହି ପୃଷ୍ଠାରୁ ନିଷ୍କ୍ରିୟ ଥିମ୍ଗୁଡ଼ିକୁ ଅପଡେଟ୍ କିମ୍ବା ଡିଲିଟ୍ କରିବାକୁ ସକ୍ଷମ ହେବେ । ଏହି ପରଦାରେ ଥିମ୍ ପ୍ରତ୍ୟେକ ପୃଷ୍ଠା ଏକ ସମୟରେ 15 ଟି ଥିମ୍ ସ୍ଥିନସର୍ ପ୍ରଦର୍ଶନ କରିବ ।

### • ସକ୍ରିୟ କରନ୍ତୁ |

- ଏହି ଲିଙ୍କ୍ କ୍ଲିକ୍ କରିବା ଦ୍ୱାରା ଏହା ସାମ୍ପ୍ରତିକ ଥିମ୍ ହୋଇଯାଏ ।

- ଲାଇଭ୍ ପୂର୍ବାବଲୋକନ |
- ଏହି ଲିଙ୍କ୍ କ୍ଲିକ୍ କରିବା ଦ୍ୱାରା ଏହି ନିର୍ଦ୍ଦେଶ ଥିବା ସଂସ୍କରଣ ସହିତ ବ୍ଲଗ୍ ର ଏକ ପୂର୍ବାବଲୋକନ ପ୍ରଦର୍ଶିତ ହୁଏ |
- ଡିଲିଟ୍ କରନ୍ତୁ |
- ଏହି ଲିଙ୍କ୍ କ୍ଲିକ୍ କରିବା ଦ୍ୱାରା ଏହି ଥିମ୍ ସଂପୂର୍ଣ୍ଣ ବିଲୋପ ହୁଏ, ସମସ୍ତ ଥିମ୍ ଫାଇଲ୍ ଏବଂ ଫୋଲ୍ଡର୍ ଅନ୍ତର୍ଭୁକ୍ତ କର | ବ୍ୟାକଅପ୍ ହୋଇନଥିବା କିଛି thing ଶିକ୍ଷା ଜିନିଷ ସବୁଦିନ ପାଇଁ ହଜିଯିବ |
- ଅଦ୍ୟତନ ଉପଲବ୍ଧ |

## ଉପରୋକ୍ତ ଅଦ୍ୟତନ ଉପଲବ୍ଧ ବିଭାଗକୁ ଅନୁସରଣ କରନ୍ତୁ | Install Themes

Listed below are several ways to install Themes:

- Automated Theme Installer(Administration > Appearance > Themes)

ଓଟପ୍ରେସ୍ ଥିମ୍ ଡିରେକ୍ଟୋରୀରୁ ଥିମ୍ ସଂସ୍ଥାପନ କରିବାକୁ ଏହା ବ୍ୟବହାର କରାଯାଇପାରିବ | ଦୃଶ୍ୟ ଥିମ୍ ସ୍ତମ୍ଭ ଖୋଜିବା ପାଇଁ ପ୍ରଶାସନ> ଦୃଶ୍ୟ> ଥିମ୍ କୁ ଯାଆନ୍ତୁ | ନୂତନ ଯୋଡ଼ା ବଟନ୍ କ୍ଲିକ୍ କରନ୍ତୁ | ଏଠାରୁ ଆପଣ ବ୍ୟବହାର କରିବାକୁ ଥିମ୍ ପାଇବେ ଯାହା ପରିବର୍ତ୍ତନଯୁକ୍ତ | ଏହି ପରଦାର ଶୀର୍ଷରେ ଏକ ନୂତନ ଥିମ୍ ଖୋଜିବା ପାଇଁ ଡିନୋଟି ଉପଲବ୍ଧ ପଦ୍ଧତି ସହିତ ସମ୍ପାନ ବ feature ଶିଷ୍ଟ୍ୟ ଅଛି; ଫିଲ୍ଡ୍, କାଣ୍ଡିଡ୍, ଏବଂ ଗୁଣବତ୍ତା ସମ୍ପାନ |

- **Using The Upload Method**

ଅପଲୋଡ୍ ପଦ୍ଧତି ଏକ ଜିପ୍ ଫାଇଲ୍ ମାଧ୍ୟମରେ ଏକ ଥିମ୍ ସଂସ୍ଥାପନ କରେ | ଓଟପ୍ରେସ୍ ଥିମ୍ ଡିରେକ୍ଟୋରୀରେ ଥିବା ସମସ୍ତ ଥିମ୍ ଏହି ଉପାୟରେ ସଂସ୍ଥାପିତ ହୋଇପାରିବ | ଜିପ୍ ଫାଇଲ୍ ଡାଉନଲୋଡ୍ କରିବା ପରେ, ପ୍ରଶାସନ> ଦୃଶ୍ୟ> ଥିମ୍ ପରିବର୍ତ୍ତନ କରନ୍ତୁ ଏବଂ ନୂତନ ଯୋଡ଼ା ବଟନ୍ କ୍ଲିକ୍ କରନ୍ତୁ | ଏହା ପରେ, ଥିମ୍ ଅପଲୋଡ୍ ଲିଙ୍କ୍ କ୍ଲିକ୍ କରନ୍ତୁ |

ZIP ଫାଇଲ୍ ପାଇଁ ବ୍ରାଉଜ୍ କରନ୍ତୁ ଏବଂ ବର୍ତ୍ତମାନ ସଂସ୍ଥାପନ କ୍ଲିକ୍ କରନ୍ତୁ | ଏହାକୁ ସାମ୍ପ୍ରତିକ ଥିମ୍ ପ୍ରସ୍ତୁତ କରିବା ପାଇଁ ଆକ୍ଟିଭେଟ୍ ଲିଙ୍କ୍ କ୍ଲିକ୍ କରନ୍ତୁ |(downloading the ZIP file, visit Administration > Appearance > Themes, and click the Add New button)

- **Using The FTP Method**

FTP ପଦ୍ଧତି ସହିତ ଏକ ଥିମ୍ ସଂସ୍ଥାପନ କରିବାକୁ ତୁମକୁ ପ୍ରଥମେ ସ୍ଥାନୀୟ କମ୍ପ୍ୟୁଟରରେ ଥିମ୍ ଫାଇଲଗୁଡ଼ିକୁ ଡାଉନଲୋଡ୍ କରିବାକୁ ପଡ଼ିବ | ଫାଇଲ୍ ସଂରଚନାକୁ ସଂରକ୍ଷଣ କରି, ଜିପ୍ ଫାଇଲ୍ ର ବିଷୟବସ୍ତୁ ବାହାର କର ଏବଂ ଏକ ନୂତନ ଫୋଲ୍ଡରରେ ଯୋଡ଼ | ଯଦି ଥିମ୍ ଲେଖକଙ୍କ ଠାରୁ ନିର୍ଦ୍ଦେଶ ଅଛି, ସେମାନଙ୍କୁ ଅନୁସରଣ କରିବାକୁ ନିଶ୍ଚିତ ହୁଅନ୍ତୁ |

ଆପଣଙ୍କ ସାଇଟର ଖେପ୍ ସର୍ଭରକୁ ପ୍ରବେଶ କରିବାକୁ ଏକ FTP କ୍ଲାଏଣ୍ଟ ବ୍ୟବହାର କରନ୍ତୁ | ଓଟପ୍ରେସ୍ ଦ୍ୱାରା ପ୍ରଦତ୍ତ ଆପଣଙ୍କର wp- ବିଷୟବସ୍ତୁ / ଥିମ୍ ଡିରେକ୍ଟୋରୀରେ ଅପଲୋଡ୍ ହୋଇଥିବା ଥିମ୍ ଫାଇଲଗୁଡ଼ିକୁ ଯୋଡ଼ନ୍ତୁ | ଯଦି ଆବଶ୍ୟକ ହୁଏ, wp- ବିଷୟବସ୍ତୁ / ଥିମ୍ ଡିରେକ୍ଟୋରୀ ଭିତରେ ଆପଣଙ୍କର ନୂତନ ଥିମ୍ ଧାରଣ କରିବାକୁ ଏକ ଫୋଲ୍ଡର୍ ସୃଷ୍ଟି କରନ୍ତୁ | ଏହାର ଏକ ଉଦାହରଣ ହେବ, ଯଦି ଆପଣଙ୍କର ଥିମ୍ ଟେମ୍ପ୍ ନାମିତ ହୁଏ ତେବେ ଏହା wp- ବିଷୟବସ୍ତୁ / ଥିମ୍ / ପରୀକ୍ଷଣରେ ରହିବା ଉଚିତ |

ପ୍ରଶାସନ> ଦୃଶ୍ୟ> ଥିମ୍ କୁ ଯାଆନ୍ତୁ, ଏବଂ ସକ୍ରିୟ ଲିଙ୍କ୍ କ୍ଲିକ୍ କରନ୍ତୁ ଥିମ୍ କୁ ଆପଣଙ୍କର ସାମ୍ପ୍ରତିକ ଥିମ୍ ଭାବରେ ବାନ୍ଧନ୍ତୁ |

CPanel ସହିତ ସଂସ୍ଥାପନ କରିବା |

cPanel କଣ୍ଟ୍ରୋଲ ପ୍ୟାନେଲଗୁଡ଼ିକ ଏକ ZIP କିମ୍ବା GZ ଫାଇଲ୍ ସହିତ ଥିବା ସଂସ୍ଥାପନ କରିବାକୁ ଅନ୍ୟ ଏକ ପଦ୍ଧତି ପ୍ରଦାନ କରେ । CPanel ମ୍ୟାନେଜର୍ ରେ, ଯଦି WordPress ସଂସ୍ଥାପିତ ହୋଇଛି, ତେବେ ଆପଣଙ୍କର ଥିମ୍ ଫୋଲ୍ଡରକୁ ଯାଆନ୍ତୁ । ପଥଟି 'public\_html / wp-content / ଥିମ୍' ପରି ଦେଖାଯିବ । ଅପଲୋଡ୍ ଫାଇଲ୍ (ଗୁଡ଼ିକ) ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ ଏବଂ ZIP ଫାଇଲ୍ ଅପଲୋଡ୍ କରନ୍ତୁ । CPanel ରେ ZIP ଫାଇଲ୍ ସିଲେକ୍ଟ କରନ୍ତୁ ଏବଂ ସେହି ଫାଇଲ୍ କୁ ସଙ୍କୋଚନ କରିବାର ଡାହାଣ ପାର୍ଶ୍ୱରେ ଥିବା ପ୍ୟାନେଲରେ ଥିବା ଏକ୍ସଟ୍ରାକ୍ଟ ଫାଇଲ୍ ବିକ୍ଷୟବସ୍ତୁ ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ ।

Go to Administration > Appearance > Themes, and click the Activate link select the Theme as your Current Theme.

All information listed above is according to the WordPress Codex. It has been shortened for brevity. The original source material can be found [here](#). Or for more information visit [codex.wordpress.org](http://codex.wordpress.org).

## Creating A Custom Theme

ଏହି ନିର୍ଦ୍ଦେଶଗୁଡ଼ିକ ଏକ ମା basic ଲିକ୍, ସର୍ବନିମ୍ନ ମାନକ ଅନୁରୂପ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ସୃଷ୍ଟି କରେ ।

ପ୍ରଥମ ପଦକ୍ଷେପ ହେଉଛି ଆପଣଙ୍କର ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ଡିରେକ୍ଟୋରୀ ଭିତରେ ଏକ ନୂତନ ଥିମ୍ ଫୋଲ୍ଡର ସୃଷ୍ଟି କରିବା । ସଠିକ୍ ପଥ ହେବ:-> wp-content> ଥିମ୍> ଏକ ବା valid ଧା ଥିମ୍ ସୃଷ୍ଟି କରିବାକୁ, ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ଗୁଡ଼ିକ ଅତି କମରେ ଏହି ଦୁଇଟି ଫାଇଲ୍ ସେଠାରେ ରହିବା ଆବଶ୍ୟକ କରନ୍ତି(correct path will be: > wp-content > themes > To create a valid theme)

- index.php
- style.css

Your stylesheet should contain a comment that alerts WordPress that a theme exists here.

```
/*
Theme Name: <theme name>
Author: <author name>
Description: <description goes here>
Version: <theme version #>
Tags: <tag to id theme>
*/
```

ତୁମର ଥିମ୍ ବର୍ତ୍ତମାନ ସୃଷ୍ଟି ହୋଇଛି । ଖର୍ଚ୍ଚପ୍ରେସ୍ ଡ୍ୟାସବୋର୍ଡକୁ ଯାଆନ୍ତୁ, ଏବଂ ଏହାକୁ ସକ୍ରିୟ କରିବାକୁ ଦୃଶ୍ୟ> ଥିମ୍ ଉପରେ କ୍ଲିକ୍ କରନ୍ତୁ ।

# Chapter 65: Update WordPress Manually

## Examples

### VIA FTP

1. ଖାତପ୍ରେସର ଇଚ୍ଛିତ ସଂସ୍କରଣକୁ [www.wordpress.org](http://www.wordpress.org) ରୁ ଆପଣଙ୍କର ସ୍ଥାନୀୟ କମ୍ପ୍ୟୁଟରକୁ ଡାଉନଲୋଡ୍ କରନ୍ତୁ ଏବଂ ଫାଇଲ୍ ଅପ୍ଲୋଡ୍ କରନ୍ତୁ |
  2. ତୁମର ସାମ୍ପ୍ରତିକ ସଂସ୍କରଣର ଏକ ବ୍ୟାକଅପ୍ ମଧ୍ୟ ରଖ ... ଠିକ୍ କ୍ଷେତ୍ରରେ |( backup of your current version)
  3. ଆପଣଙ୍କର ପ୍ରିୟ FTP କ୍ଲାଇଣ୍ଟ୍ ସହିତ ଆପଣଙ୍କର ଷ୍ଟେସାଇଟ୍ ସହିତ ସଂଯୋଗ କରନ୍ତୁ (FileZilla ଲୋକପ୍ରିୟ ଏବଂ ସହଜ, କିନ୍ତୁ ଯେକି any ଶସି FTP କ୍ଲାଇଣ୍ଟ୍ ଭଲ ରହିବ) |
    - ଏଥିପାଇଁ ନିର୍ଦ୍ଦେଶଗୁଡ଼ିକ ଖାତପ୍ରେସ୍ ପରିସର ବାହାରେ, କିନ୍ତୁ ପ୍ରସ୍ତାବିତ FTP ପ୍ରସଙ୍ଗରେ ଭବିଷ୍ୟତରେ ମିଳିପାରେ |
  4. ଆପଣଙ୍କ ସର୍ଭରରେ ସେମାନଙ୍କର ମେଳ ଖାଉଥିବା ଡିରେକ୍ଟୋରୀରେ “wp-admin” ଏବଂ “wp-includes” ଶୀର୍ଷକ ଫୋଲ୍ଡରଗୁଡ଼ିକୁ (ଏବଂ ସେମାନଙ୍କର ବିଷୟବସ୍ତୁ) ଅପଲୋଡ୍ କରନ୍ତୁ | ସାମ୍ପ୍ରତିକ ଫୋଲ୍ଡରଗୁଡ଼ିକୁ ଓଭର ରାଇଟ୍ କରିବାକୁ ନିଶ୍ଚିତ ହୁଅନ୍ତୁ |
    - ଆପଣ ଅନ୍ତର୍ଭୁକ୍ତ ହୋଇଥିବା ଥିମଗୁଡ଼ିକ ମଧ୍ୟରୁ ଗୋଟିଏକୁ ବ୍ୟବହାର ନକରିବା ପର୍ଯ୍ୟନ୍ତ ଆପଣ “wp-content” ଫୋଲ୍ଡର ଅପଲୋଡ୍ କରିବାକୁ ଛାଡ଼ି ପାରିବେ | ଯଦି ତୁମେ ବାଛିଥିବା ସଂସ୍କରଣ ସହିତ ଅନ୍ତର୍ଭୁକ୍ତ ହୋଇଥିବା ଡିଫଲ୍ଟ ଥିମଗୁଡ଼ିକୁ ଅପଡେଟ୍ / ଅପଲୋଡ୍ କରିବାକୁ ଚାହୁଁଛ, ତୁମେ ମଧ୍ୟ ଏହି ଫୋଲ୍ଡରକୁ ଅପଲୋଡ୍ କରିବା ଉଚିତ |
  5. ହୋମ ଫୋଲ୍ଡରରେ ବ୍ୟକ୍ତିଗତ ଫାଇଲଗୁଡ଼ିକୁ ଅପଲୋଡ୍ କରନ୍ତୁ (index.php, wp - \*। Php, ଇତ୍ୟାଦି) |
- ଆପଣ "license.txt" ଏବଂ "readme.html" ନାମକ ଫାଇଲଗୁଡ଼ିକୁ ଛାଡ଼ି ପାରିବେ କାରଣ ସେଗୁଡ଼ିକ କାର୍ଯ୍ୟ କରିବା ଆବଶ୍ୟକ ନୁହେଁ ଏବଂ ସେଗୁଡ଼ିକ ସୁରକ୍ଷା ଶୋଷଣ ପାଇଁ ଆପଣଙ୍କର WP ସଂସ୍କରଣ ନିର୍ଣ୍ଣୟ କରିବାର ପଦ୍ଧତି ଭାବରେ ବ୍ୟବହୃତ ହୋଇପାରିବ |
6. କି required ଶସି ଆବଶ୍ୟକୀୟ ଡାଟାବେସ୍ ଅପଡେଟ୍ କରିବାକୁ ଆପଣଙ୍କ ଷ୍ଟେସାଇଟ୍ ପରିଦର୍ଶନ କରନ୍ତୁ ଏବଂ ଲଗ୍ କରନ୍ତୁ |

ସମସ୍ତ WP ଅବ୍ୟତନଗୁଡ଼ିକରେ DB ପରିବର୍ତ୍ତନ ନାହିଁ, କିନ୍ତୁ କେତେକ କରନ୍ତି |

**Note:** This method will create orphaned files that can/will build up over time and may present security risks. Be sure to do a file comparison after completion and delete old files off your server from previous WP versions which are no longer in use.

# Chapter 66: WordPress Plugin creation

## Introduction

ଓର୍ଡ଼ପ୍ରେସ୍ ପ୍ଲଗିନ୍ ଗୁଡ଼ିକ ସର୍ଭର ଡର୍କ ଏବଂ / କିମ୍ବା ଆପଣଙ୍କର ୱେବସାଇଟ୍ ଆପ୍ଲିକେସନ୍ ର ଆଡମିନ୍ ଅଂଶ ଉପରେ ଧ୍ୟାନ ଦେବା ଉଚିତ୍ । ଭଲ ପ୍ଲଗିନ୍ ଭଲ ଆପ୍ ପରି, ସେମାନେ ଗୋଟିଏ ଜିନିଷ ପ୍ରକୃତରେ ଭଲ କରନ୍ତି । ସେଗୁଡ଼ିକ CMS ର ଅଂଶଗୁଡ଼ିକୁ ଏକ ମହୁଧାରୀ ଉପାୟରେ ବ enhance ାଇବା ଏବଂ ସ୍ୱୟଂଚାଳିତ କରିବା ପାଇଁ ଉଦ୍ଦିଷ୍ଟ, ଯେହେତୁ ଆପଣ ସେଗୁଡ଼ିକୁ ସକ୍ରିୟ ଏବଂ ନିଷ୍କ୍ରିୟ କରିପାରିବେ । ଭଲ ପ୍ଲଗିନ୍ ଗୁଡ଼ିକ ଓର୍ଡ଼ପ୍ରେସ୍ ମୂଳ କାର୍ଯ୍ୟ, ଫିଲ୍ଡ୍ ଏବଂ ବିଦ୍ୟମାନ ଜାଭାସ୍କ୍ରିପ୍ଟ ଏବଂ css framework ାଞ୍ଚାଗୁଡ଼ିକର ବ୍ୟବହାର କରେ ।

## Examples

### Minimal Setup of a Plugin Folder and Files

First step of creating a plugin is creating the folder and file which the plugin will load from.

Plugins are located in /wp-content/plugins/.

The WordPress standard is to create a folder and filename that mirror each other like so:

```
/wp-content/plugins/myplugin/  
/wp-content/plugins/myplugin/myplugin.php
```

ତୁମର ପ୍ଲଗିନ୍ ଫାଇଲ୍ ତିଆରି କରିବା ପରେ ତୁମ ପ୍ଲଗିନ୍ ହେଡର୍ ସହିତ ଆରମ୍ଭ କରିବାକୁ ପଡ଼ିବ । ଏହା ଓର୍ଡ଼ପ୍ରେସ୍ କୁ ଆପଣଙ୍କର ପ୍ଲଗିନ୍ ଫାଇଲ୍ ସ୍ଥାନ କରିବାକୁ ଏବଂ ପ୍ଲଗିନ୍ ବିଷୟରେ ମେଟା ତଥ୍ୟ ସଂରକ୍ଷଣ କରିବାକୁ ଅନୁମତି ଦିଏ, ଏବଂ ଉପଭୋକ୍ତାମାନଙ୍କୁ ଏହାକୁ ବ୍ୟବହାର କରିବାକୁ ଅନୁମତି ଦିଏ ଏବଂ ସେମାନେ ତୁମର ପ୍ଲଗିନ୍ ସକ୍ରିୟ କିମ୍ବା ନିଷ୍କ୍ରିୟ ଚାହୁଁଛନ୍ତି କି ନାହିଁ ତାହା ନିର୍ଣ୍ଣୟ କରିବାକୁ ଅନୁମତି ଦିଏ । ଆପଣ ସୃଷ୍ଟି କରିଥିବା ଆପଣଙ୍କର ମୁଖ୍ୟ ପ୍ଲଗିନ୍ ଫାଇଲ୍ ର ଶୀର୍ଷରେ ଏହି ଟେମ୍ପଲେଟ୍ କପି କରନ୍ତୁ ଏବଂ ଆବଶ୍ୟକ ଅନୁଯାୟୀ ଏହାକୁ ପରିବର୍ତ୍ତନ କରନ୍ତୁ:

```
<?php  
/**  
 * Plugin Name: PLUGIN-NAME  
 * Plugin URI: HTTP-LINK-TO-WEBSITE-PLUGIN-PAGE-OR-REPO  
 * Description: BREIF DESCRIPTION - KEEP IT SHORT  
 * Author: WORDPRESS-DOT-ORG-USERNAME  
 * Version: 0.0.1  
 * Author URI: HTTP-LINK-TO-MAINTAINER  
 * License: GNU General Public License v2 or later  
 * License URI: http://www.gnu.org/licenses/gpl-2.0.html  
 * Text Domain: short_prefix  
 */  
  
// Begin custom PHP WordPress plugin work
```

*Note that WordPress plugins should typically be licensed as GPL. Licensing should not be discussed as part of this topic though.*

ଏହି ସମୟରେ, ଆପଣ ଓର୍ଡ଼ପ୍ରେସ୍ ଆଡମିନ୍ ଅଞ୍ଚଳରେ ଆପଣଙ୍କର ନୂତନ ପ୍ଲଗିନ୍ ଦେଖିବାରେ ସକ୍ଷମ ହେବା ଉଚିତ୍ । ଏକ ମାନକ ସେଟଅପ୍ ରେ ଆପଣ ଏହି କ୍ଷେତ୍ରକୁ /wp-admin/plugins.php ରେ ଖୋଜିବେ । ଆଗକୁ ଯାଅ ଏବଂ ତୁମର ସକ୍ରିୟ କର ।

ପ୍ଲଗିନ୍, ଏବଂ ତୁମେ ତୁମର ପ୍ଲଗିନ୍ ନିର୍ମାଣର ପରବର୍ତ୍ତୀ ପଦକ୍ଷେପକୁ ଯିବାକୁ ପ୍ରସ୍ତୁତ!

କ୍ରିୟାଶୀଳ କିଛି ଉପରେ ଆମର ଉଦାହରଣ ଶେଷ କରିବାକୁ, ଆପଣ ବର୍ତ୍ତମାନ ଆପଣଙ୍କର ପ୍ଲଗିନ୍ ଫାଇଲ୍ ତଳେ ନିମ୍ନକୁ ଯୋଡ଼ିପାରିବେ::

```
die('My custom plugin is loaded : '.__FILE__);
```

Refreshing your site after this change should result in all pages printing this text. *Never do this in production (live) sites, and always remember to take this back out before continuing.*

# Chapter 67: Wordpress theme and child-theme development

## Introduction

ସରଳ ସୂଚନା ୱେବସାଇଟ୍ ସୃଷ୍ଟି କରିବା ପାଇଁ ୱାଡ଼ପ୍ରେସ୍ ଏକ ବହୁଳ ବ୍ୟବହୃତ CMS କିନ୍ତୁ ଅଧିକ ଅତ୍ୟାଧୁନିକ ୱେବସାଇଟ୍ ଏବଂ ଏପରିକି ଛୋଟ ୱେବସାଇଟ୍ ତିଆରି ପାଇଁ ।

ୱାଡ଼ପ୍ରେସ୍ ଥିବା ବ୍ୟବହାର କରେ । ଏହି ଥିମ୍ ଗୁଡ଼ିକ ୱାଡ଼ପ୍ରେସ୍ ୱେବସାଇଟ୍ ର ଲେ-ଆଉଟ୍ ଏବଂ ବିଷୟବସ୍ତୁ କାର୍ଯ୍ୟକାରୀତା ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ଥିମ୍ ଗୁଡ଼ିକ ସମଗ୍ର ଇଣ୍ଟରନେଟରେ ମିଳିପାରିବ ।

ପ୍ରତ୍ୟେକ ଥିମ୍ ନିଜସ୍ୱ ସ୍ୱତନ୍ତ୍ର କାର୍ଯ୍ୟକାରୀତା ଏବଂ ଲେ-ଆଉଟ୍ ଅଛି କିନ୍ତୁ ବେଳେବେଳେ ଏକ ୱେବସାଇଟ୍ ପାଇଁ କେବଳ ସଠିକ୍ ଥିମ୍ ଖୋଜିବା କଷ୍ଟକର । ସେ ଉପଯୋଗୀତା we ଆମେ ମଧ୍ୟ ଆମର ନିଜସ୍ୱ ଥିମ୍ ସୃଷ୍ଟି କରିବାକୁ ସକ୍ଷମ ଅଟୁ ।

Examples

## Developing your own theme

ଏକ ୱାଡ଼ପ୍ରେସ୍ ଥିମ୍ ଦୁଇ ପ୍ରକାରର ଫାଇଲ୍ ଧାରଣ କରେ । ପ୍ରତ୍ୟେକ ଥିମ୍ ଥିମ୍ ମା basic ଲିକ୍ ଫାଇଲ୍ଗୁଡ଼ିକ ଏବଂ ଥିମ୍ ଗୁଡ଼ିକର ଲେଆଉଟ୍ ଏବଂ କାର୍ଯ୍ୟକାରୀତାକୁ ବ୍ୟାଖ୍ୟା କରୁଥିବା ଫାଇଲ୍ଗୁଡ଼ିକ । ଏହି ଦ୍ୱିତୀୟ ଗୋଷ୍ଠୀ ମୁଁ ଥିମ୍ ନିର୍ଦ୍ଦେଶ ଫାଇଲ୍ କହିବାକୁ ଯାଉଛି ।

### The basic theme files

ମା basic ଲିକ୍ ଥିମ୍ ଫାଇଲ୍ଗୁଡ଼ିକ ହେଉଛି ଏକ ଫାଇଲ୍ ଯାହା ଏକ ଥିମ୍ ସେଟଅପ୍ ଏବଂ ପଞ୍ଜିକରଣ ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ନିମ୍ନରେ ଥିବା ତାଲିକାରେ ମୁଁ ପ୍ରତ୍ୟେକ ଫାଇଲ୍ ଏବଂ ଏହାର ବ୍ୟବହାର ବିଷୟରେ ବର୍ଣ୍ଣନା କରିବି । ପରେ ମୁଁ ସବୁଠାରୁ ମା basic ଲିକ୍ ଉଦାହରଣ ଫାଇଲ୍ଗୁଡ଼ିକୁ ଯୋଗ କରିବି ଯାହା ଆପଣଙ୍କର ନିଜସ୍ୱ ୱାଡ଼ପ୍ରେସ୍ ଥିମ୍ ସେଟ୍ କରିବାକୁ ଆବଶ୍ୟକ ।

- **functions.php:** function.php ଫାଇଲ୍ ସମସ୍ତ ଫଙ୍କସନ୍, ସାଇଡ୍ ବାର୍, ସ୍ଲିପ୍ ଏବଂ ଥିମ୍ ଅନ୍ତର୍ଭୁକ୍ତ କରିବା ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ଏହି ଫାଇଲ୍ରେ ଆପଣ ସକ୍ଷମ ଅଟନ୍ତି, ଉଦାହରଣ ସ୍ୱରୂପ, CSS ଫାଇଲ୍, JS ଫାଇଲ୍ ଇତ୍ୟାଦି ଅନ୍ତର୍ଭୁକ୍ତ କରନ୍ତୁ । ହେଡର୍ ଏବଂ ଫୁଟର୍: ହେଡର୍ ଏବଂ ଫୁଟର୍ ଫାଇଲ୍ (header.php ଏବଂ footer.php) ହେଉଛି ଫାଇଲ୍ ଯାହା ହେଡର୍ ଏବଂ ଫୁଟର୍ କୁ ଡାକିବା ପାଇଁ ବ୍ୟବହୃତ ହୁଏ । ହେଡର୍ ଏବଂ ଫୁଟର୍ ଫାଇଲ୍ ଉଦାହରଣ ସ୍ୱରୂପ ୱାଡ଼ପ୍ରେସ୍ ବ୍ୟାକ୍-ଏଣ୍ଡ ସିଷ୍ଟମର ଲିଙ୍କ୍ ଧରିଥାଏ ।
- **index.php:** index.php ଫାଇଲ୍ ହେଉଛି ଡିଫଲ୍ଟ-ପୃଷ୍ଠା ଟେମ୍ପ୍ଲେଟ୍ ସୃଷ୍ଟି କରୁଥିବା ଫାଇଲ୍ । ଏହି ଫାଇଲ୍ ରେ ଆପଣ ଏହି ଡିଫଲ୍ଟ-ଟେମ୍ପ୍ଲେଟ୍ ଲେ-ଆଉଟ୍ ର ଖଣ୍ଡଗୁଡ଼ିକୁ ଦେଖି, ସଂପାଦନ ଏବଂ ଅପସାରଣ କରିପାରିବେ ।
- **single.php:** single.php ଫାଇଲ୍ ହେଉଛି ଫାଇଲ୍ ଯାହା ଏକକ ପୋଷ୍ଟ ଟେମ୍ପ୍ଲେଟ୍ ପେଜ୍ ସୃଷ୍ଟି କରେ । ପୃଷ୍ଠାଗୁଡ଼ିକ ପାଇଁ ଡିଫଲ୍ଟ-ଟେମ୍ପ୍ଲେଟ୍ ପରି କିନ୍ତୁ ବର୍ତ୍ତମାନ ଏକକ ପୋଷ୍ଟ ପୃଷ୍ଠାଗୁଡ଼ିକ ପାଇଁ ।
- **format.php** ଫର୍ମାଟ୍ । Php ଫାଇଲ୍ ହେଉଛି ଏକ ଫାଇଲ୍ ଯାହା ଏକ ପୃଷ୍ଠାରୁ ବିଷୟବସ୍ତୁ-ଟେମ୍ପ୍ଲେଟ୍ ଟେମ୍ପ୍ଲେଟ୍ ନିର୍ମାଣ କରେ । ତେଣୁ ଯଦି ଆପଣଙ୍କର ଏକ ମୂଳପୃଷ୍ଠା ଅଛି ଏବଂ ଆପଣ ଏକ ପାଠ ଯୋଗ କରି ଏହାକୁ ବ୍ୟାକ୍-ଏଣ୍ଡରୁ ଏଡିଟ୍ କରିବେ । ଏହି ଫାଇଲ୍ ଏହି ପାଠ୍ୟର ମାନକ ମାର୍କଅପ୍ ସୃଷ୍ଟି କରେ ।
- **404.php** 404.php ଫାଇଲ୍ 404 ଟେମ୍ପ୍ଲେଟ୍ ସୃଷ୍ଟି କରେ । ଏହି ଫାଇଲ୍ ଏହି ପୃଷ୍ଠାର ମା basic ଲିକ୍ ଲେ-ଆଉଟ୍ କୁ ନେଇ ଗଠିତ ।
- **archive.php** archive.php ଫାଇଲ୍ ଆର୍କାଇଭ୍ ପେଜ୍ ର ଲେ-ଆଉଟ୍ ସୃଷ୍ଟି କରେ ।
- **style.css** ମା basic ଲିକ୍ ଷ୍ଟାଇଲ୍ ଶୀଟ୍ ଫାଇଲ୍ ।

ତେଣୁ ଏହି ଡାଲିକାରେ ଆପଣ ନିଜସ୍ୱ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ସେଟ୍ ଅପ୍ ପାଇଁ ସମସ୍ତ ଆବଶ୍ୟକୀୟ ଫାଇଲ୍ ଦେଖିପାରିବେ । ବର୍ତ୍ତମାନ କିଛି ଫାଇଲ୍ ଉପରେ ନଜର ପକାଇବା ଯାହାକୁ ଆପଣ ଚାହଁଲେ ସୃଷ୍ଟି କରିବାକୁ ସକ୍ଷମ କିଛି ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ପାଇଁ ଫାଇଲ୍ ଆବଶ୍ୟକ ନୁହେଁ । ଏହି ଫାଇଲଗୁଡ଼ିକ ମୁଖ୍ୟତଃ template ଟେମ୍ପଲେଟ୍ ଫାଇଲ୍ ଏବଂ ଅନ୍ୟାନ୍ୟ କାର୍ଯ୍ୟକ୍ଷମ ପରିମାଣ ।

### Custom page templates

```
<?php
/*
 * Template Name: Homepage Template
 */
get_header();
?>
```

ପୃଷ୍ଠା- <ଆପଣଙ୍କର ନିଜ ନାମ> .php: ଏକ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ରେ ଆପଣ ଏକାଧିକ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ସୃଷ୍ଟି କରିବାକୁ ସକ୍ଷମ ଅଟନ୍ତି । ନୂତନ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ଫାଇଲ୍ ସୃଷ୍ଟି କରି । ଏକ ମାନକ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ଫାଇଲ୍ ନିମ୍ନଲିଖିତ ନାମ ଗୁଣଗୁଡ଼ିକୁ ନେଇ ଗଠିତ । ଟେମ୍ପଲେଟର ପୃଷ୍ଠା ନାମ ଏବଂ .php ଯଦି ଉଦାହରଣ ସ୍ୱରୂପ ଆପଣ ଆପଣଙ୍କର କ୍ଲଗ୍ ପୃଷ୍ଠା ପାଇଁ ଏକ ନୂତନ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ସୃଷ୍ଟି କରିବାକୁ ଚାହାଁନ୍ତି ତେବେ ଆପଣ ଏହାକୁ ପେଜ୍-କ୍ଲଗ୍ ବୋଲି କହିପାରିବେ । ନିଶ୍ଚିତ କରନ୍ତୁ ଯେ ଆପଣ ଅତିକମରେ get\_header () ଏବଂ get\_footer () କାର୍ଯ୍ୟଗୁଡ଼ିକୁ ଅନ୍ତର୍ଭୁକ୍ତ କରିଛନ୍ତି । ନିମ୍ନଲିଖିତ ଉଦାହରଣ ଯୋଗ କରି ଫାଇଲ୍ ର ଶୀର୍ଷରେ ଥିବା ଏକ ମନ୍ତବ୍ୟରେ ତୁମର ଟେମ୍ପଲେଟ୍ ନାମକରଣ କର ।

### Custom single post page templates

```
<?php
/*
 * Template Name: Post Portfolio
 * Template Post Type: post, page
 */
?>
```

ସିଙ୍ଗଲ୍- <ଆପଣଙ୍କର ନିଜ ନାମ> .php: ଏକ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ଉପରେ ଯେପରି ଉପରୋକ୍ତ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ପରି ଆପଣ ମଧ୍ୟ ନିଜର ଏକକ ପୋଷ୍ଟ ପୃଷ୍ଠା ଟେମ୍ପଲେଟ୍ ସୃଷ୍ଟି କରିବାରେ ସକ୍ଷମ ଅଟନ୍ତି । ପେଜ୍ ଟେମ୍ପଲେଟ୍ ପରି ଫାଇଲ୍ ଏହାର ଏକ ପୋଷ୍ଟ ପୃଷ୍ଠା <ତୁମର ଟେମ୍ପଲେଟର ନାମ> ଏବଂ ଫାଇଲ୍ ଏକ୍ସଟେନ୍ସନ୍ .php ଘୋଷଣା କରିବା ପାଇଁ ତିନୋଟି ଅଂଶକୁ ନେଇ ଗଠିତ । ଯେପରି ପେଜ୍ ଟେମ୍ପଲେଟ୍ ସର୍ବନିମ୍ନ ଆବଶ୍ୟକତା ନିଶ୍ଚିତ କରେ ଯେ ଖର୍ଚ୍ଚପ୍ରେସ୍ ନୂତନ ଟେମ୍ପଲେଟ୍ ପ read ୍ରୁକ୍ତି ନିଶ୍ଚିତ କରିବାକୁ get\_header () ଏବଂ get\_footer () ଫଙ୍କସନ୍ସ ଯୋଗ କରୁଛି । ଏବଂ ନିମ୍ନଲିଖିତ ଉଦାହରଣ ପରି ତୁମର ଟେମ୍ପଲେଟ୍ ନାମକୁ ଯୋଡ଼ିବା ।

We also indicate the `template post type`: wich stands for the kind of template it is, in this case post and page.

### Custom post text templates

ଫର୍ମାଟ୍ - <ଆପଣଙ୍କର ନିଜ ନାମ> .php: ଏକ ଖର୍ଚ୍ଚପ୍ରେସ୍ ଥିମ୍ ରେ ଆପଣ ପୋଷ୍ଟ ଆଉଟପୁଟ୍ ଟେମ୍ପଲେଟ୍ ମଧ୍ୟ ସୃଷ୍ଟି କରିବାରେ ସକ୍ଷମ ଅଟନ୍ତି । ଏହି ଫର୍ମାଟ୍ ଟେମ୍ପଲେଟ୍ ଗୁଡ଼ିକ ହେଉଛି ଏକ ପୋଷ୍ଟର ଲେ-ଆଉଟ୍ ଏବଂ ବିଷୟବସ୍ତୁ । ଉଦାହରଣ ସ୍ୱରୂପ ଯଦି କେତେକ କ୍ଷେତ୍ରରେ ଆପଣ ପୋଷ୍ଟକୁ କେବଳ ବିଷୟବସ୍ତୁ କିମ୍ବା ପୋଷ୍ଟର ଆଖ୍ୟା ଦେଖାଇବାକୁ ଚାହାଁନ୍ତି ତେବେ ଆପଣ ସେହି ପ୍ରକାରର ଆଉଟପୁଟ୍ ସୃଷ୍ଟି କରିବାକୁ ଏହି ଟେମ୍ପଲେଟଗୁଡ଼ିକ ବ୍ୟବହାର କରିପାରିବେ । ଯେହେତୁ ଏହି ପ୍ରକାରର ଟେମ୍ପଲେଟଗୁଡ଼ିକ କେବଳ ପୋଷ୍ଟ ବ୍ୟାକ୍-ଏଣ୍ଡ ବିଷୟବସ୍ତୁକୁ ଫର୍ମାଟ୍ କରୁଛି ଯାହା ଏକ ଉପଭୋକ୍ତା ଦ୍ୱି created ାରା ସୃଷ୍ଟି ହୋଇଛି, ଆମକୁ get\_header () ଏବଂ get\_footer () ଅନ୍ତର୍ଭୁକ୍ତ କରିବାର ଆବଶ୍ୟକତା ନାହିଁ କାରଣ ଏହା ପୃଷ୍ଠା ଟେମ୍ପଲେଟରେ ପୂର୍ବରୁ ବ୍ୟାଖ୍ୟା କରାଯାଇଛି । ନିଶ୍ଚିତ କରନ୍ତୁ ଯେ ଆପଣଙ୍କର ଟେମ୍ପଲେଟ୍ ନିମ୍ନଲିଖିତ ମ basic ଲିକ ଉଦାହରଣ ବ୍ୟବହାର କରି ଏକ ପୋଷ୍ଟକୁ ଚିହ୍ନିବାରେ ସକ୍ଷମ ଅଟେ ।

```
<div>
```

```
<article id="post-<?php the_ID(); ?>" <?php post_class(); ?>>
</article>
</div>
```

ତେଣୁ ବର୍ତ୍ତମାନ ଆମେ ମା basic ଲିକ୍ ଫାଇଲଗୁଡ଼ିକ ବିଷୟରେ କିଛି ଜାଣିବା ଏବଂ ଅନେକ ଟେମ୍ପଲେଟ୍ ନିର୍ଦ୍ଦିଷ୍ଟ ଫାଇଲଗୁଡ଼ିକ ସାଇଡ୍ ବାର୍ ଏବଂ ଓଜେଟ୍ ବିଷୟରେ କଥାବାର୍ତ୍ତା ଆରମ୍ଭ କରିବାର ସମୟ ଆସିଛି । ଭବିଷ୍ୟତରେ ଏହା ଏକ ନିଜସ୍ୱ ଓର୍ଡ଼ରେସ୍ ଥିମ୍ ସୃଷ୍ଟି କରିବା ପାଇଁ ଷ୍ଟେପ୍ ଟୁଏଗୋରିଆଲ୍ ର ଷ୍ଟେପ୍ ସହିତ ଏକ ସ୍ମାର୍ଟ ସହିତ ମିଳିତ ହେବ ।

---

# Chapter 68: wp\_get\_current\_user()

## Syntax

- wp\_get\_current\_user()

## Examples

### Getting the current user

Getting all the information of current user in wordpress using the predefined function wp\_get\_current\_user();

```
<?php

$current_user = wp_get_current_user();

echo "Username :". $current_user->user_login;
echo "Username :". $current_user->ID;
echo "Username :". $current_user->user_pass;
echo "Username :". $current_user->user_nicename;
echo "Username :". $current_user->user_email;
echo "Username :". $current_user->user_url;
echo "Username :". $current_user->user_registered;
echo "Username :". $current_user->user_activation_key;
echo "Username :". $current_user->user_status;
echo "Username :". $current_user->display_name;

?>
```

### Use foreach loop to get user info from wp\_get\_current\_user()

```
$user = wp_get_current_user();

foreach($user->data as $key=>$user_data){
    if($key == 'user_pass' || $key == 'user_activation_key' || $key=='user_status'){
    }
    else{
        $nice_key = ucfirst(str_replace('_', ' ', $key));

        if($key == 'user_registered'){
            $user_data = date_i18n(get_option('date_format'), strtotime($user_data));
        }

        echo $nice_key . ' : ' . $user_data . '<br />';
    }
}
```

---

# Chapter 69: wp\_get\_current\_user()

## Examples

### Get current loggedin user information

Retrieves the information pertaining to the currently logged in user, and places it in the global variable `$current_user`

This function does not accept any parameters.

Usage:

```
<?php wp_get_current_user(); ?>
```

Example:

```
<?php
$current_user = wp_get_current_user();

echo 'Username: ' . $current_user->user_login . "\n";
echo 'User email: ' . $current_user->user_email . "\n";
echo 'User level: ' . $current_user->user_level . "\n";
echo 'User first name: ' . $current_user->user_firstname . "\n";
echo 'User last name: ' . $current_user->user_lastname . "\n";
echo 'User display name: ' . $current_user->display_name . "\n";
echo 'User ID: ' . $current_user->ID . "\n";
?>
```

To determine if a visitor is logged in or not, you can use `is_user_logged_in()` before, then get the current user info if the visitor it's logged in:

```
<?php
if ( is_user_logged_in() ) {
    $current_user = wp_get_current_user();

    echo 'welcome, ' . $current_user->user_login . '!';
} else {
    echo 'welcome, visitor!';
}
?>
```

---

# Chapter 70: WP\_Query() Loop

## Introduction

ପୋଷ୍ଟ, ପୃଷ୍ଠା ଏବଂ କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାର ପାଇଁ ଜିଜ୍ଞାସା କରିବାକୁ WP\_ ପ୍ରଶ୍ନ | ଆପଣ ନିର୍ଦ୍ଦିଷ୍ଟ ପୋଷ୍ଟ ଏବଂ ପୃଷ୍ଠା କିମ୍ବା କଷ୍ଟମ୍ ପୋଷ୍ଟ ପ୍ରକାର ପାଇଁ ତାଲିକା ପାଇବେ | WP\_Query ଛୁମ୍ବ ମାନବଶ୍ଚ ଅନୁଯାୟୀ ତାତାବେସ୍ ରୁ ପୋଷ୍ଟଗୁଡ଼ିକ ଚାଣିବାକୁ ଅନୁମତି ଦିଏ |

## Examples

### Retrieve latest 10 posts

```
$args = array(
    'post_type'=>'post',
    'posts_per_page' =>'10'
);
$latest_posts_query = new WP_Query( $args );
if($latest_posts_query->have_posts()) :
while ( $latest_posts_query-> have_posts() ) : $latest_posts_query->the_post();
//Get post details here
endwhile;
endif;
```

---

# Chapter 71: WP-CLI

## Introduction

ଓର୍ଡ଼ରେସ୍ ସଂସ୍ଥାପନ ପରିଚାଳନା ପାଇଁ WP-CLI ହେଉଛି କମାଣ୍ଡ ଲାଇନ୍ ଉପକରଣଗୁଡ଼ିକର ଏକ ସେଟ୍ । ଖେବ୍ ବ୍ରାଉଜର ବ୍ୟବହାର ନକରି ଆପଣ ସ୍ଥାନୀୟ ଅପଡେଟ୍ କରିପାରିବେ, ମଲ୍ଟିସାଇଟ୍ ଇନଷ୍ଟଲ୍ ଏବଂ ଅଧିକ କିଛି ବିନ୍ୟାସ କରିପାରିବେ ।

## Examples

### Manage themes

Get a list of themes.

```
$ wp theme list
```

Install the latest version from wordpress.org and activate

```
$ wp theme install twentysixteen --activate
```

Install from a local zip file

```
$ wp theme install ../my-theme.zip
```

Install from a remote zip file

```
$ wp theme install http://s3.amazonaws.com/bucketname/my-theme.zip?AWSAccessKeyId=123&Expires=456&Signature=abcdef
```

Get details of an installed theme

```
$ wp theme get twentysixteen --fields=name,title,version
```

Get status of theme

```
$ wp theme status twentysixteen
```

### Manage plugins

Get a list of plugins

```
$ wp plugin list
```

List active plugins on the site.

```
$ wp plugin list --status=active --format=json
```

List plugins on each site in a network.

```
$ wp site list --field=url | xargs -I % wp plugin list --url=%
```

Activate plugin

```
$ wp plugin activate hello-dolly
```

Deactivate plugin

```
$ wp plugin deactivate hello-dolly
```

Delete plugin

```
$ wp plugin delete hello-dolly
```

Install the latest version from wordpress.org and activate

```
$ wp plugin install bbpress --activate
```

## Manage WP-CLI itself

Display the version currently installed.

```
$ wp cli version
```

Check for updates to WP-CLI.

```
$ wp cli check-update
```

Update WP-CLI to the latest stable release.

```
$ wp cli update
```

List all available aliases.

```
$ wp cli alias
```

Print various details about the WP-CLI environment.

```
$ wp cli info
```

Dump the list of installed commands, as JSON.

```
$ wp cli cmd-dump
```

## Download, install, update and manage a WordPress install.

Download WordPress core

```
$ wp core download --locale=nl_NL
```

Install WordPress

```
$ wp core install --url=example.com --title=Example --admin_user=supervisor --admin_password=strongpassword --admin_email=info@example.com
```

Display the WordPress version

```
$ wp core version
```

Transform a single-site install into a WordPress multisite install.

```
$ wp core multisite-convert
```

Install WordPress multisite from scratch.

```
$ wp core multisite-install
```

## Manage users

List user IDs

```
$ wp user list --field=ID
```

Create a new user.

```
$ wp user create bob bob@example.com --role=author
```

Update an existing user.

```
$ wp user update 123 --display_name=Mary --user_pass=marypass
```

Delete user 123 and reassign posts to user 567

```
$ wp user delete 123 --reassign=567
```

## Perform basic database operations using credentials stored in wp-config.php

Create a new database.

```
$ wp db create
```

Drop an existing database.

```
$ wp db drop --yes
```

Reset the current database.

```
$ wp db reset --yes
```

Execute a SQL query stored in a file.

```
$ wp db query < debug.sql
```

# Chapter 72: WP-Cron

## Examples

### wp\_schedule\_event() example

```
// register activation hook
register_activation_hook(____FILE____, 'example_activation' );

// function for activation hook
function example_activation() {
    // check if scheduled hook exists
    if ( !wp_next_scheduled( 'my_event' ) ) {
        // Schedules a hook
        // time() - the first time of an event to run ( UNIX timestamp format )
        // 'hourly' - recurrence ('hourly', 'twicedaily', 'daily' )
        // 'my_event' - the name of an action hook to execute.
        wp_schedule_event( time(), 'hourly', 'my_event' );
    }
}

add_action( 'my_event', 'do_this_hourly' );

// the code of your hourly event
function do_this_hourly() {
    // put your code here
}

// register deactivation hook
register_deactivation_hook(____FILE____, 'example_deactivation');

// function for deactivation hook
function example_deactivation() {
    // clear scheduled hook
    wp_clear_scheduled_hook( 'my_event' );
}
```

**Important:** the WordPress cron runs only when some page of your website is hit. So, for website with low traffic you need to setup the cron on your hosting to hit pages. ( ଯେତେବେଳେ ଆପଣଙ୍କର ୱେବସାଇଟ୍ ର କିଛି ପୃଷ୍ଠା ହିଟ୍ ହୁଏ, ଏହିପ୍ରକାର କ୍ରୋନ୍ ଚାଲିବ | ତେଣୁ, କମ୍ ଟ୍ରାଫିକ୍ ଥିବା ୱେବସାଇଟ୍ ପାଇଁ ପୃଷ୍ଠାଗୁଡ଼ିକୁ ହିଟ୍ କରିବା ପାଇଁ ତୁମର ହୋଷ୍ଟିଂରେ କ୍ରୋନ୍ ସେଟ୍ ଅପ୍ କରିବାକୁ ପଡ଼ିବ । )

### custom recurrence interval in wp\_schedule\_event()

```
// this function add custom interval (5 minutes) to the $schedules
function five_minutes_interval( $schedules ) {
    $schedules['five_minutes'] = array(
        'interval' => 60 * 5,
        'display'   => '5 minutes';
    );
    return $schedules;
}

// add a custom interval filter
add_filter( 'cron_schedules', 'five_minutes_interval' );
```

```
// Schedules a hook  
wp_schedule_event( time(), 'five_minutes', 'my_event' );
```

# Credits

S. No	Chapters	Contributors
1	Getting started with WordPress	4444, A. Raza, Andrew, animuson, Anupam, Chris Fletcher, Ciprian, Community, Florida, James Jones, JonasCz, Leo F, Marc St Raymond, Mayank Gupta, Milap, nus, Panda, rap-2-h, Seth C., Shubham, Trevor Clarke, vajrasar
2	Actions and Filters	David, Ihor Vorotnov, Mrinal Haque, Trying Tobemyslf
3	Add Shortcode	purvik7373, RamenChef
4	Add/remove contact info for users with user_contactmethods filter hook	Petar Popovic, RamenChef
5	add_action()	Abel Melquiades Callejo, Waqas Bukhary
6	add_editor_style()	Gabriel Chi Hong Lee
7	add_menu_page()	brasofilo, Gabriel Chi Hong Lee
8	add_submenu_page()	Gabriel Chi Hong Lee, theoretisch
9	add_theme_support()	Gabriel Chi Hong Lee
10	Admin Dashboard Widgets	theoretisch
11	AJAX	Andy, Digvijayad, Gaurav Srivastava, GreatBlakes, Nisarg Patel, Ruslan Murarov, stweb
12	Alternating main loop (pre_get_posts filter)	Dawid Urbanski, Petar Popovic
13	Child Theme Basics	Andrei, Razvan Onofrei, Vlad Olaru
14	Create a Post Programmatically	RamenChef, Roel Magdaleno, RRikesh
15	Create Template for Custom Post Type	Ashok G, Egnaro, Joe Dooley, mnoronha
16	Creating a custom template	Petar Popovic

17	Custom excerpts with excerpt_length and excerpt_more	<a href="#">inkista</a> , <a href="#">Petar Popovic</a> , <a href="#">RamenChef</a>
18	Custom Post Types	<a href="#">Caio Felipe Pereira</a> , <a href="#">Dan Devine</a> , <a href="#">J.D.</a> , <a href="#">janw</a> , <a href="#">jgraup</a> , <a href="#">Kushal Shah</a> , <a href="#">Omar Khaiyam</a> , <a href="#">Ranuka</a> , <a href="#">theoretisch</a>
19	Customizer Basics (Add Panel, Section, Setting, Control)	<a href="#">4444</a> , <a href="#">Ahmad Awais</a> , <a href="#">RamenChef</a>
20	Customizer Hello World	<a href="#">Dan Green-Leipciger</a>
21	Debugging	<a href="#">barbocc</a> , <a href="#">dingo_d</a> , <a href="#">jgraup</a>
22	Enqueuing scripts	<a href="#">dingo_d</a> , <a href="#">Harshal Limaye</a> , <a href="#">J.D.</a> , <a href="#">mbacon40</a> , <a href="#">montrealist</a> , <a href="#">Pelmered</a> , <a href="#">Petar Popovic</a>
23	Enqueuing Styles	<a href="#">dingo_d</a> , <a href="#">Harshal Limaye</a> , <a href="#">Laxmana</a> , <a href="#">mnoronha</a> , <a href="#">montrealist</a> , <a href="#">Petar Popovic</a> , <a href="#">RamenChef</a> , <a href="#">Ruslan Murarov</a> , <a href="#">virtualLast</a>
24	Function : wp_trim_words()	<a href="#">Harshal Limaye</a>
25	Function: add_action()	<a href="#">bosco</a> , <a href="#">RamenChef</a>
26	get_bloginfo()	<a href="#">Abel Melquiades Callejo</a> , <a href="#">Harshal Limaye</a> , <a href="#">HeyCameron</a> , <a href="#">KenB</a> , <a href="#">Nate Beers</a> , <a href="#">RamenChef</a> , <a href="#">Tom J Nowell</a> , <a href="#">virtualLast</a> , <a href="#">Wes Moberly</a>
27	get_home_path()	<a href="#">Ihor Vorotnov</a>
28	get_option()	<a href="#">Gabriel Chi Hong Lee</a>
29	get_permalink()	<a href="#">Gabriel Chi Hong Lee</a>
30	get_template_part()	<a href="#">Dan Devine</a> , <a href="#">Kushal Shah</a>
31	get_the_category()	<a href="#">Gabriel Chi Hong Lee</a>
32	get_the_title()	<a href="#">Gabriel Chi Hong Lee</a>
33	home_url()	<a href="#">dingo_d</a> , <a href="#">Kushal Shah</a> , <a href="#">matthew</a> , <a href="#">Mr. Developer</a>
34	How Can I integrate Markdown editor with Advance Custom Field's repeater Add-on.	<a href="#">Fatbit</a>

35	init	<a href="#">Abel Melquiades Callejo</a> , <a href="#">barbocc</a>
36	Installation and Configuration	<a href="#">Kenyon</a> , <a href="#">Marco Romano</a> , <a href="#">Ping.Chen</a> , <a href="#">S.L. Barth</a> , <a href="#">stig-js</a> , <a href="#">theoretisch</a> , <a href="#">Yuan Lung Luo</a>
37	Making network requests with HTTP API	<a href="#">Jordan</a> , <a href="#">mjangda</a> , <a href="#">Rarst</a>
38	Meta Box	<a href="#">Austin Winstanley</a>
39	Options API	<a href="#">Harshal Limaye</a> , <a href="#">Pat J</a> , <a href="#">RamenChef</a>
40	Plugin development	<a href="#">Angle.R</a> , <a href="#">Ping.Chen</a>
41	Post Formats	<a href="#">Shashank Agarwal</a>
42	Querying posts	<a href="#">dingo_d</a>
43	Remove Auto Line Breaks From Content and Excerpt	<a href="#">Austin Winstanley</a>
44	Remove Version from Wordpress and Stylesheets	<a href="#">jay.jivani</a> , <a href="#">mnoronha</a> , <a href="#">theoretisch</a>
45	REST API	<a href="#">Picard</a>
46	Run WordPress local with XAMPP	<a href="#">Pierre.Vriens</a> , <a href="#">theoretisch</a>
47	Secure your installation	<a href="#">James Jones</a>
48	Security in WordPress - Escaping	<a href="#">Laxmana</a> , <a href="#">the4kman</a>
49	Security in WordPress - Sanitization	<a href="#">Laxmana</a>
50	Shortcode	<a href="#">Ad Wicks</a> , <a href="#">Adam Genshaft</a> , <a href="#">brasofilo</a> , <a href="#">John Slegers</a> , <a href="#">Kylar</a> , <a href="#">Shashank Agarwal</a>
51	Shortcode with attribute	<a href="#">Digvijayad</a> , <a href="#">Firefog</a> , <a href="#">RamenChef</a>
52	Shortcodes	<a href="#">Pelmered</a>
53	Sidebars	<a href="#">dingo_d</a> , <a href="#">Kushal Shah</a> , <a href="#">theoretisch</a>

54	Site Migration	<a href="#">Austin Winstanley</a>
55	Taxonomies	<a href="#">adifatz</a> , <a href="#">Kushal Shah</a> , <a href="#">purvik7373</a>
56	Template hierarchy	<a href="#">jgraup</a> , <a href="#">MarZab</a> , <a href="#">Pelmered</a> , <a href="#">theoretisch</a>
57	template_include	<a href="#">Abel Melquiades Callejo</a> , <a href="#">David</a> , <a href="#">RamenChef</a>
58	The \$wpdb Object	<a href="#">Kushal Shah</a> , <a href="#">mcon</a> , <a href="#">stweb</a>
59	The Admin Bar (aka "The Toolbar")	<a href="#">dingo_d</a> , <a href="#">Harshal Limaye</a> , <a href="#">JCL1178</a> , <a href="#">Kushal Shah</a>
60	The Loop (main WordPress loop)	<a href="#">anik4e</a> , <a href="#">Dawid Urbanski</a>
61	the_title()	<a href="#">Gabriel Chi Hong Lee</a>
62	Themes	<a href="#">Jef</a>
63	Update WordPress Manually	<a href="#">KnightHawk</a>
64	WordPress Plugin creation	<a href="#">Seth C.</a>
65	Wordpress theme and child-theme development	<a href="#">Deathstorm</a>
66	wp_get_current_user()	<a href="#">Abel Melquiades Callejo</a> , <a href="#">Benoti</a> , <a href="#">Muhammad Farrukh Faizy</a>
67	WP_Query() Loop	<a href="#">vrajesh</a>
68	WP-CLI	<a href="#">jgraup</a>
69	WP-Cron	<a href="#">stweb</a>