

वर्डप्रेस



विषयसूची

के बारे में	1
अध्याय 1: वर्डप्रेस के साथ शुरुआत करना	2
टिप्पणियां	2
संस्करणों	3
उदाहरण	4
वर्डप्रेस का परिचय	4
वर्डप्रेस थीम्स	5
यूआरएल को विशिष्ट टेम्पलेट्स में मैप करना	5
मूल थीम निर्देशिका संरचना	5
"एकल" का उदाहरण (व्यक्तिगत पोस्ट के लिए टेम्पलेट)	6
"संग्रह" का उदाहरण (एकाधिक पोस्ट की सूची के लिए टेम्पलेट)	7
पोस्ट, पेज, कस्टम पोस्ट प्रकार और कस्टम फ़ील्ड	7
अध्याय 2: क्रियाएँ और फ़िल्टर	8
वाक्य - विन्यास	8
पैरामीटर	8
टिप्पणियां	8
उदाहरण	11
add_action - init	11
add_action - init - अनाम फ़ंक्शन	11
add_action - init - क्लास ऑब्जेक्ट के भीतर	11
add_action - init - स्थिर वर्ग के भीतर	12
अध्याय 3: शोर्टकोड जोड़ें	13
वाक्य - विन्यास	13
पैरामीटर	13
टिप्पणियां	13
उदाहरण	13
हालिया पोस्ट के लिए सरल शॉर्टकोड	13
हाल की पोस्टों के लिए उन्नत शॉर्टकोड	14
अध्याय 4: user_contactmethods फ़िल्टर हुक वाले उपयोगकर्ताओं के लिए संपर्क जानकारी जोड़ें/हटाएँ	15

उदाहरण	15
सबसे लोकप्रिय सामाजिक नेटवर्क को सक्षम करना	15
संपर्क विधि हटाना	16
अध्याय 5: add_action()	17
वाक्य - विन्यास	17
पैरामीटर	17
उदाहरण	17
डायरेक्ट फ़ंक्शन कॉलबैक	17
फ़ंक्शन नाम संदर्भ कॉलबैक	17
क्लास स्टैटिक विधि कॉलबैक	18
ऑब्जेक्ट विधि कॉलबैक	18
अध्याय 6: add_editor_style()	19
परिचय	19
वाक्य - विन्यास	19
पैरामीटर	19
उदाहरण	19
एकल सीएसएस फ़ाइल लोड हो रही है	19
अध्याय 7: add_menu_page()	20
परिचय	20
वाक्य - विन्यास	20
पैरामीटर	20
टिप्पणियां	20
उदाहरण	21
नेव बार में "थीम पेज शीर्षक" आइटम जोड़ना	21
ओओपी और मेनू पेज पर स्क्रिप्ट/शीलियाँ कैसे लोड करें	22
अध्याय 8: add_submenu_page()	24
परिचय	24
वाक्य - विन्यास	24
पैरामीटर	24
टिप्पणियां	24
उदाहरण	25

"सबमेनू पेज" को नेव बार में "टूल्स" के उप-पेज के रूप में जोड़ना	25
अध्याय 9: add_theme_support()	27
परिचय	27
वाक्य - विन्यास	27
पैरामीटर	27
टिप्पणियां	27
उदाहरण	27
पोस्ट प्रारूपों के लिए थीम समर्थन जोड़ना	27
पोस्ट में पोस्ट थंबनेल के लिए थीम समर्थन जोड़ना	27
अध्याय 10: व्यवस्थापक डैशबोर्ड विजेट	29
परिचय	29
वाक्य - विन्यास	29
पैरामीटर	29
उदाहरण	29
सरल विजेट (पाठ प्रदर्शित करता है)	29
अध्याय 11: अजाक्स	31
उदाहरण	31
JSON प्रतिक्रिया के साथ AJAX अनुरोध	31
.ajax() और WordPress नॉन के साथ AJAX	32
wp_ajax - मुख्य कार्यक्षमता + _wpnonce जाँच	33
नॉन के साथ एक साधारण वर्ग का उपयोग करके ओओपी अजाक्स सबमिशन	34
अध्याय 12: वैकल्पिक मुख्य लूप (pre_get_posts फ़िल्टर)	38
वाक्य - विन्यास	38
पैरामीटर	38
टिप्पणियां	38
उदाहरण	38
और भी अधिक विशिष्ट लूप लक्ष्यीकरण	38
केवल एक श्रेणी से पोस्ट दिखाएँ	39
पोस्ट प्राप्त करने से पहले बुनियादी उपयोग को फ़िल्टर करें	39
पोस्ट सूची से श्रेणी को बाहर निकालें, शेयर संपादित करें	39
मुख्य लूप के लिए पोस्ट_पर्_पेज बदलें	40
केवल मुख्य वर्डप्रेस लूप को लक्षित करना	40

अध्याय 13: बाल विषय की मूल बातें	41
वाक्य - विन्यास	41
टिप्पणियां	41
उदाहरण	41
2) उद्देश्य	41
परिभाषा और आवश्यकताएँ	42
3) टेम्पलेट ओवरराइटिंग	42
संपत्ति प्रतिस्थापन	43
अध्याय 14: प्रोग्रामेटिक रूप से एक पोस्ट बनाएं	45
वाक्य - विन्यास	45
पैरामीटर	45
टिप्पणियां	45
बहस	45
डुप्लीकेट पोस्ट से बचें	46
व्याख्या	47
उदाहरण	47
परिचय	47
एक बेसिक पोस्ट बनाएं	47
एक बेसिक पेज बनाएं	47
अध्याय 15: कस्टम पोस्ट प्रकार के लिए टेम्पलेट बनाएं	49
उदाहरण	49
कस्टम पोस्ट प्रकार की पुस्तक के लिए एक कस्टम टेम्पलेट बनाना	49
कस्टम पोस्ट प्रकार टेम्पलेट्स	49
कस्टम पोस्ट प्रकार पुरालेख:	49
कस्टम पोस्ट प्रकार एकल टेम्पलेट:	51
अध्याय 16: एक कस्टम टेम्पलेट बनाना	53
उदाहरण	53
बुनियादी रिक्त टेम्पलेट बनाना	53
हमारे टेम्पलेट में शीर्ष लेख और पाद लेख शामिल करना	54
सामग्री के साथ कस्टम टेम्पलेट	54
अध्याय 17: अंश_लंबाई और अंश_अधिक के साथ कस्टम अंश	56

उदाहरण	56
अंश की लंबाई 50 शब्दों तक सीमित करें	56
अंश के अंत में और पढ़ें लिंक जोड़ रहा हूँ	56
अंश के अंत में कुछ बिंदु जोड़ना	57
अध्याय 18: कस्टम पोस्ट प्रकार	59
वाक्य - विन्यास	59
पैरामीटर	59
उदाहरण	59
एक कस्टम पोस्ट प्रकार पंजीकृत करना	59
मुख्य क्वेरी में कस्टम पोस्ट प्रकार जोड़ें	60
मुख्य RSS फ़ीड में कस्टम पोस्ट प्रकार जोड़ना	61
कस्टम पोस्ट प्रकार पंजीकृत करें	61
ट्वेंटी फिफ्थीन वर्डप्रेस थीम का उपयोग करके कस्टम पोस्ट प्रकार	62
डिफ़ॉल्ट खोज में कस्टम पोस्ट प्रकार	63
अध्याय 19: कस्टमाइज़र मूल बातें (पैनल, अनुभाग, सेटिंग, नियंत्रण जोड़ें)	64
उदाहरण	64
एक कस्टमाइज़र पैनल जोड़ें	64
बुनियादी सेटिंग्स और उनके नियंत्रण के साथ एक कस्टमाइज़र अनुभाग जोड़ें	64
अध्याय 20: कस्टमाइज़र हैलो वर्ल्ड	68
पैरामीटर	68
उदाहरण	68
हैलो वर्ल्ड उदाहरण	68
अध्याय 21: डिबर्गिंग	70
परिचय	70
टिप्पणियां	70
उदाहरण	70
WP_DEBUG	70
WP_DEBUG_LOG	70
WP_DEBUG_DISPLAY	70
SCRIPT_DEBUG	71
सहेजें प्रश्न	71
उदाहरण wp-config.php और डिबर्गिंग के लिए अच्छे अभ्यास	71

लॉग को एक अलग फ़ाइल में देखें	72
अध्याय 22: लिपियों को कतारबद्ध करना	73
वाक्य - विन्यास	73
पैरामीटर	73
उदाहरण	73
Functions.php में स्क्रिप्टों को कतारबद्ध करना	73
केवल IE के लिए स्क्रिप्टों को कतारबद्ध करें	74
विशिष्ट पृष्ठों के लिए सशर्त रूप से लिपियों को सूचीबद्ध करना	74
अध्याय 23: कतारबद्ध शैलियाँ	76
वाक्य - विन्यास	76
पैरामीटर	76
उदाहरण	76
निर्भरता के रूप में किसी अन्य सीएसएस फ़ाइल के साथ आंतरिक सीएसएस फ़ाइल शामिल करना	76
आंतरिक सीएसएस फ़ाइल सहित	76
बाहरी सीएसएस फ़ाइल सहित	76
केवल IE के लिए स्टाइलशीट संकलित करें	77
आपके प्लगइन क्लास के लिए आंतरिक सीएसएस फ़ाइल शामिल है	77
वैकल्पिक स्टाइलशीट जोड़ें	77
अध्याय 24: फ़ंक्शन: wp_trim_words()	79
वाक्य - विन्यास	79
पैरामीटर	79
उदाहरण	79
पोस्ट सामग्री को ट्रिम करना	79
अध्याय 25: फ़ंक्शन: add_action()	80
वाक्य - विन्यास	80
पैरामीटर	80
टिप्पणियाँ	80
उदाहरण	80
बेसिक एक्शन हुक	80
एक्शन हुक प्राथमिकता	81
हुकिंग क्लास और ऑब्जेक्ट मेथड्स टू एक्शन	81
ऑब्जेक्ट मेथड एक्शन हुक	82

क्लास मेथड एक्शन हुक	82
अध्याय 26: get_bloginfo()	84
परिचय	84
वाक्य - विन्यास	84
पैरामीटर	84
टिप्पणियां	84
उदाहरण	86
साइट का शीर्षक प्राप्त करना	86
साइट टैगलाइन प्राप्त करना	87
सक्रिय थीम URL प्राप्त करना	88
साइट यूआरएल प्राप्त करें	89
साइट प्रशासक का ईमेल पता प्राप्त करें	89
अध्याय 27: get_home_path()	90
परिचय	90
पैरामीटर	90
टिप्पणियां	90
Get_home_path() और ABSPATH के बीच महत्वपूर्ण अंतर	90
इसे अपने कोड में उपयोग करना	91
उदाहरण	91
प्रयोग	91
अध्याय 28: get_option()	92
परिचय	92
वाक्य - विन्यास	92
पैरामीटर	92
टिप्पणियां	92
उदाहरण	92
ब्लॉग शीर्षक दिखाएँ	92
ब्लॉग का नाम H1 शैली में	93
करेक्टर सेट दिखाएँ	93
गैर-मौजूदा विकल्पों को संभालना	93
अध्याय 29: get_permalink()	94

परिचय	94
वाक्य - विन्यास	94
पैरामीटर	94
टिप्पणियां	94
उदाहरण	94
Get_permalink का सरल उपयोग	94
लिंक पाने के लिए पोस्ट निर्दिष्ट करना	94
पोस्ट के नाम के बिना लिंक प्राप्त करें	95
अध्याय 30: get_template_part()	96
वाक्य - विन्यास	96
पैरामीटर	96
उदाहरण	96
एक सबफोल्डर से एक टेम्पलेट भाग लोड करें	96
एक विशिष्ट फ़ाइल प्राप्त करें	96
अध्याय 31: get_template_part()	97
परिचय	97
वाक्य - विन्यास	97
पैरामीटर	97
उदाहरण	97
एक कस्टम टेम्पलेट सहित	97
डैश-पृथक फ़ाइल नाम के साथ एक कस्टम टेम्पलेट शामिल करना	97
एक निर्देशिका के अंदर से एक कस्टम टेम्पलेट शामिल करना	98
एक निर्देशिका के अंदर स्थित डैश-पृथक फ़ाइल नाम के साथ एक कस्टम टेम्पलेट शामिल करना	98
वैरिएबल को कस्टम टेम्पलेट स्कोप में पास करना	98
अध्याय 32: get_template_part()	99
वाक्य - विन्यास	99
पैरामीटर	99
उदाहरण	99
टेम्पलेट भाग लोड हो रहा है	99
अध्याय 33: get_the_category()	100
परिचय	100

वाक्य - विन्यास	100
पैरामीटर	100
टिप्पणियां	100
उदाहरण	100
पोस्ट की सभी श्रेणियों के नाम प्राप्त करें	101
पोस्ट की श्रेणियों की सभी आईडी प्राप्त करें	101
अध्याय 34: <code>get_the_title()</code>	102
परिचय	102
वाक्य - विन्यास	102
पैरामीटर	102
टिप्पणियां	102
उदाहरण	102
<code>Get_the_title</code> का सरल उपयोग	102
किसी निर्दिष्ट पोस्ट आईडी का शीर्षक प्राप्त करें	102
अध्याय 35: <code>होम यूआरएल()</code>	104
वाक्य - विन्यास	104
पैरामीटर	104
उदाहरण	104
होम यूआरएल प्राप्त करना	104
साइट URL	104
अध्याय 36: मैं मार्कडाउन संपादक को एडवांस कस्टम फील्ड के रिपीटर ऐड-ओ के साथ कैसे एकीकृत कर सकता हूँ	105
उदाहरण	105
मार्कडाउन संपादक जोड़ें	105
अध्याय 37: <code>init</code>	106
वाक्य - विन्यास	106
टिप्पणियां	106
उदाहरण	106
<code>\$_POST</code> अनुरोध डेटा संसाधित किया जा रहा है	106
<code>\$_GET</code> अनुरोध डेटा संसाधित किया जा रहा है	106
एक कस्टम पोस्ट प्रकार पंजीकृत करना	106
अध्याय 38: स्थापना और विन्यास	107

उदाहरण	107
लैंप पर वर्डप्रेस	107
MAMP में स्थापना WP	108
अध्याय 39: HTTP एपीआई के साथ नेटवर्क अनुरोध करना	110
वाक्य - विन्यास	110
पैरामीटर	110
टिप्पणियां	110
रिटर्न	110
उदाहरण	110
एक दूरस्थ JSON संसाधन प्राप्त करें	110
अध्याय 40: मेटा बॉक्स	111
परिचय	111
वाक्य - विन्यास	111
टिप्पणियां	111
उदाहरण	111
नियमित इनपुट और चयनित इनपुट के साथ एक सरल उदाहरण	111
अध्याय 41: विकल्प एपीआई	114
परिचय	114
वाक्य - विन्यास	114
टिप्पणियां	114
उदाहरण	114
get_option	114
add_option	115
हटाएँ_विकल्प	115
अद्यतन_विकल्प	115
अध्याय 42: प्लगइन विकास	116
वाक्य - विन्यास	116
पैरामीटर	116
टिप्पणियां	116
उदाहरण	117
फ़िल्टर	117

कार्य	117
प्लगइन विकास उदाहरण: पसंदीदा गीत विजेट	117
अध्याय 43: पोस्ट प्रारूप	119
टिप्पणियां	119
उदाहरण	119
थीम में पोस्ट प्रकार जोड़ा जा रहा है	119
post_type 'पेज' में पोस्ट-फॉर्मेट जोड़ें	119
कस्टम पोस्ट प्रकार 'my_custom_post_type' पंजीकृत करें	119
पोस्ट टाइप 'my_custom_post_type' में पोस्ट-फॉर्मेट जोड़ें	120
कस्टम पोस्ट प्रकार 'my_custom_post_type' को 'समर्थन' पैरामीटर के साथ पंजीकृत करें	120
पोस्ट के लिए थीम समर्थन जोड़ें	120
फ़ंक्शन कॉल	120
अध्याय 44: पोस्ट पूछना	121
वाक्य - विन्यास	121
पैरामीटर	121
टिप्पणियां	121
कभी भी query_posts() का उपयोग न करें	121
उदाहरण	122
WP_Query() ऑब्जेक्ट का उपयोग करना	122
Get_posts() का उपयोग करना	122
अध्याय 45: सामग्री और अंश से ऑटो लाइन ब्रेक हटाएँ	124
परिचय	124
टिप्पणियां	124
उदाहरण	124
फ़िल्टर हटाएँ	124
फ़िल्टर हटाने का कार्य	124
अध्याय 46: वर्डप्रेस और स्टाइलशीट से संस्करण हटाएँ	126
परिचय	126
वाक्य - विन्यास	126
पैरामीटर	126
टिप्पणियां	126
उदाहरण	126

सीएसएस/जेएस से संस्करण संख्याएं हटाएं	126
वर्डप्रेस से संस्करण संख्याएँ हटाएँ	127
अध्याय 47: रेस्ट एपीआई	128
परिचय	128
टिप्पणियां	128
उदाहरण	128
पूर्ण कामकाजी उदाहरण	129
अध्याय 48: XAMPP के साथ वर्डप्रेस लोकल चलाएँ	130
परिचय	130
उदाहरण	130
1. एक्सएएमपीपी स्थापित करना	130
2. XAMPP इंस्टाल करने के बाद एक डेटाबेस सेटअप करें	130
3. डेटाबेस सेटअप करने के बाद वर्डप्रेस इंस्टाल करना	130
अध्याय 49: अपनी स्थापना सुरक्षित करें	132
टिप्पणियां	132
उदाहरण	132
फ़ाइल संपादक अक्षम करें	132
WP-config.php ले जाएँ	132
वर्डप्रेस तालिकाओं के लिए एक कस्टम उपसर्ग सेट करें	133
अध्याय 50: वर्डप्रेस में सुरक्षा - बचना	138
वाक्य - विन्यास	138
टिप्पणियां	138
उदाहरण	138
HTML कोड में डेटा से बचें	138
एक यूआरएल से बचें	138
जेएस कोड में डेटा से बचें	138
भागने के गुण	139
टेक्स्टएरिया में डेटा से बचें	139
अध्याय 51: वर्डप्रेस में सुरक्षा - स्वच्छता	140
वाक्य - विन्यास	140
टिप्पणियां	140
उदाहरण	140

टेक्स्ट फ़ील्ड को स्वच्छ करें	140
शीर्षक को स्वच्छ करें	140
ईमेल को सेमिटाइज़ करें	141
एचटीएमएल क्लास को सेमिटाइज़ करें	141
फ़ाइल नाम को स्वच्छ करें	141
उपयोगकर्ता नाम को स्वच्छ करें	141
अध्याय 52: शॉर्टकोड	142
उदाहरण	142
शॉर्टकोड पंजीकृत करना	142
वर्डप्रेस बैकएंड में शॉर्टकोड का उपयोग करना	142
नए शॉर्टकोड जोड़ना	142
PHP कोड के अंदर शॉर्टकोड का उपयोग करना (थीम और प्लगइन्स)	143
विजेट्स में शॉर्टकोड का उपयोग करना	143
अध्याय 53: विशेषता के साथ शॉर्टकोड	144
वाक्स - विन्यास	144
पैरामीटर	144
टिप्पणियां	144
उदाहरण	144
शॉर्टकोड के उदाहरण	144
एक स्व-समापन शॉर्टकोड बनाना	144
मापदंडों के साथ एक स्व-समापन शॉर्टकोड बनाना	145
एक संलग्न शॉर्टकोड बनाना	145
विजेट्स में शॉर्टकोड	146
अध्याय 54: लघुकोड	147
उदाहरण	147
शॉर्टकोड परिचय	147
बटन शॉर्टकोड	147
अध्याय 55: साइडबार	149
वाक्स - विन्यास	149
पैरामीटर	149
टिप्पणियां	149
उदाहरण	149

साइडबार पंजीकृत करना	149
साइडबार प्राप्त करें	150
अध्याय 56: साइट स्थानांतरण	151
वाक्य - विन्यास	151
उदाहरण	151
नए यूआरएल के साथ तालिकाओं को अपडेट किया जा रहा है	151
अध्याय 57: वर्गीकरण	152
वाक्य - विन्यास	152
पैरामीटर	152
उदाहरण	152
शैलियों के लिए वर्गीकरण दर्ज करने का उदाहरण	152
पेज में श्रेणी जोड़ें	153
पेजों में श्रेणियाँ और टैग जोड़ें और उन्हें क्लास के रूप में डालें	153
पेजों में श्रेणियाँ और टैग जोड़ें और क्लास के रूप में डालें	154
अध्याय 58: टेम्पलेट पदानुक्रम	156
टिप्पणियां	156
उदाहरण	156
परिचय	156
डिबगिंग	158
अध्याय 59: टेम्पलेट_शामिल	159
पैरामीटर	159
टिप्पणियां	159
उदाहरण	159
सरल उदाहरण	159
अधिक विज्ञापन उदाहरण	160
अध्याय 60: \$wpdb ऑब्जेक्ट	161
टिप्पणियां	161
उदाहरण	161
एक वेरिएबल का चयन करना	161
एकाधिक पंक्तियों का चयन करना	161
अध्याय 61: एडमिन बार (उर्फ "द टूलबार")	163

टिप्पणियां	163
उदाहरण	163
एडमिनिस्ट्रेटर को छोड़कर बाकी सभी से एडमिन टूलबार को हटाना	163
फ़िल्टर का उपयोग करके एडमिन टूलबार को हटाना	163
एडमिन बार से वर्डप्रेस लोगो कैसे हटाएं	163
एडमिन लॉगिन पेज पर अपना कस्टम लोगो और कस्टम लिंक जोड़ें	164
अध्याय 62: लूप (मुख्य वर्डप्रेस लूप)	165
उदाहरण	165
मूल वर्डप्रेस लूप संरचना	165
वैकल्पिक लूप सिंटैक्स	165
लूप में कोई आइटम नहीं संभालना	165
अध्याय 63: the_title()	167
परिचय	167
वाक्य - विन्यास	167
पैरामीटर	167
टिप्पणियां	167
उदाहरण	167
शीर्षक का सरल उपयोग	167
शीर्षक को पहले और बाद में कोड के साथ प्रिंट करना	167
अध्याय 64: विषय-वस्तु	169
परिचय	169
उदाहरण	169
वर्डप्रेस थीम्स	169
थीम कैसे चुनें	169
उपलब्ध अद्यतन	169
थीम्स स्थापित करें	170
एक कस्टम थीम बनाना	171
अध्याय 65: वर्डप्रेस को मैनुअल रूप से अपडेट करें	172
उदाहरण	172
एकटैबी के माध्यम से	172
अध्याय 66: वर्डप्रेस प्लगइन निर्माण	173

परिचय	173
उदाहरण	173
प्लगइन फ़ोल्डर और फ़ाइलों का न्यूनतम सेटअप	173
अध्याय 67: वर्डप्रेस थीम और बाल-थीम विकास	175
परिचय	175
उदाहरण	175
अपनी खुद की थीम विकसित करना	175
अध्याय 68: wp_get_current_user()	178
वाक्य - विन्यास	178
उदाहरण	178
वर्तमान उपयोगकर्ता प्राप्त करना	178
wp_get_current_user() से उपयोगकर्ता जानकारी प्राप्त करने के लिए फ़ोरैच लूप का उपयोग करें	178
अध्याय 69: wp_get_current_user()	179
उदाहरण	179
वर्तमान लॉगइन उपयोगकर्ता जानकारी प्राप्त करें	179
अध्याय 70: WP_Query() लूप	180
परिचय	180
उदाहरण	180
नवीनतम 10 पोस्ट पुनः प्राप्त करें	180
अध्याय 71: WP-CLI	181
परिचय	181
उदाहरण	181
थीम प्रबंधित करें	181
प्लगइन्स प्रबंधित करें	181
WP-CLI को स्वयं प्रबंधित करें	182
वर्डप्रेस इंस्टाल को डाउनलोड, इंस्टॉल, अपडेट और प्रबंधित करें।	183
उपयोगकर्ताओं को प्रबंधित करें	183
WP-config.php में संग्रहीत क्रेडेंशियल्स का उपयोग करके बुनियादी डेटाबेस संचालन करें	183
अध्याय 72: WP-क्रोन	185
उदाहरण	185
wp_schedule_event() उदाहरण	185
wp_schedule_event() में कस्टम पुनरावृत्ति अंतराल	185

के बारे में

आप इस पीडीएफ को किसी भी व्यक्ति के साथ साझा कर सकते हैं जिसके बारे में आपको लगता है कि इससे लाभ हो सकता है, नवीनतम संस्करण यहां से डाउनलोड किया गया है: वर्डप्रेस

यह शैक्षिक उद्देश्यों के लिए बनाई गई एक अनौपचारिक और मुफ्त वर्डप्रेस ईबुक है। सारी सामग्री स्टैक ओवरफ्लो दस्तावेज़ीकरण से निकाली गई है, जो स्टैक ओवरफ्लो पर कई मेहनती व्यक्तियों द्वारा लिखा गया है। यह न तो स्टैक ओवरफ्लो और न ही आधिकारिक वर्डप्रेस से संबद्ध है।

सामग्री क्रिएटिव कॉमन्स BY-SA के तहत जारी की गई है, और प्रत्येक अध्याय के योगदानकर्ताओं की सूची इस पुस्तक के अंत में क्रेडिट अनुभाग में प्रदान की गई है। जब तक अन्यथा निर्दिष्ट न किया जाए, छवियों पर उनके संबंधित स्वामियों का कॉपीराइट हो सकता है। सभी ट्रेडमार्क और पंजीकृत ट्रेडमार्क उनके संबंधित कंपनी मालिकों की संपत्ति हैं।

इस पुस्तक में प्रस्तुत सामग्री का उपयोग अपने जोखिम पर करें; इसके सही या सटीक होने की गारंटी नहीं है, कृपया अपनी प्रतिक्रिया और सुधार info@zzzprojects.com पर भेजें

अध्याय 1: वर्डप्रेस के साथ शुरुआत करना

टिप्पणियां



वर्डप्रेस एक ओपन सोर्स कंटेंट मैनेजमेंट सिस्टम (CMS) है जिसका उपयोग वेबसाइट बनाने और प्रबंधित करने के लिए किया जाता है। वर्डप्रेस इंटरनेट पर प्रति मील सबसे लोकप्रिय सीएमएस है, जो लगभग आधे को बिजली प्रदान करता है लेखन के समय सभी सीएमएस वेबसाइटों में से और सभी वेबसाइटों में से लगभग एक चौथाई इंटरनेट पर।

वर्डप्रेस ने ब्लॉगिंग के लिए एक मंच के रूप में शुरुआत की थी लेकिन पिछले कुछ वर्षों में यह अधिकांश प्रकार की वेबसाइटों के लिए उपयुक्त बन गया है। इंटरफ़ेस का उपयोग कोडिंग ज्ञान के बिना किया जा सकता है जो इसे शुरुआती और डेवलपर्स के लिए लोकप्रिय बनाता है जो अपने ग्राहकों को अपनी वेबसाइट प्रबंधित करने के लिए सशक्त बनाना चाहते हैं।

वर्डप्रेस की लोकप्रियता का एक और बड़ा कारक इसका लचीलापन है, जो मुख्य रूप से कोर के प्लगइन और थीम सिस्टम के कारण है। प्लगइन सिस्टम कोर कोड को संशोधित किए बिना कोर कार्यक्षमता का विस्तार करना आसान बनाता है। इसी तरह, थीम प्रणाली वेबसाइट के लेआउट और सौंदर्यशास्त्र को बदलना आसान बनाती है। अब हजारों निःशुल्क और प्रीमियम वर्डप्रेस प्लगइन्स और थीम उपलब्ध हैं। इनमें से कई WordPress.org प्लगइन रिपॉजिटरी पर स्थित हैं और थीम भंडार क्रमशः।

वर्डप्रेस को उसके अपने समुदाय द्वारा विकसित किया गया है, लेकिन यह ऑटोमेटिक कंपनी के साथ मजबूती से जुड़ा हुआ है, जो वर्डप्रेस के कई मुख्य डेवलपर्स को रोजगार देता है।

कोड

वर्डप्रेस PHP पर बनाया गया है सर्वर स्क्रिप्टिंग भाषा और MySQL प्रश्न पूछने की भाषा.

वर्डप्रेस उपयोगकर्ता सामग्री और कॉन्फिगरेशन के लिए डेटास्टोर के रूप में MySQL का उपयोग करता है। PHP इसमें गड़बड़ी करता है HTML में सामग्री डेटा सभी आवश्यक संपत्तियों के साथ वेबपेज।

WordPress.com बनाम WordPress.org

आप Automattic के WordPress.com पर साइन अप करके वर्डप्रेस का उपयोग कर सकते हैं सेवा और आपकी मेजबानी वेबसाइट को उनके सर्वर पर अपलोड करें, या आप WordPress.org से वर्डप्रेस सॉफ्टवेयर डाउनलोड कर सकते हैं और अपनी वेबसाइट को अपने नियंत्रण वाले सर्वर पर होस्ट करें। पहला विकल्प आसान है लेकिन आप कोई भी संपादन नहीं कर सकते स्थान कोड। आप केवल वर्डप्रेस इंटरफ़ेस के माध्यम से परिवर्तन कर सकते हैं। दूसरा विकल्प इसके लिए अधिक काम की आवश्यकता होती है, लेकिन यह आपको अपने वेबसाइट कोड के साथ जो चाहें करने की सुविधा देता है। अगर आप यदि आप एक StackOverflow उपयोगकर्ता हैं, तो आप संभवतः दूसरे विकल्प के साथ जा रहे होंगे।

खुला स्रोत

वर्डप्रेस एक ओपन सोर्स सॉफ्टवेयर है जिसका अर्थ है कि यह उपयोग करने के लिए मुफ्त है और कोई भी इसके स्रोत को देख सकता है कोड करें और इसमें योगदान दें। संभावित योगदानकर्ता योगदान पृष्ठ को पढ़कर शुरुआत कर सकते हैं [वर्डप्रेस कोडेक्स](#)..

[वर्डप्रेस टिकट ट्रैकर पर बग सबमिट करके बग की रिपोर्ट की जा सकती है](#) ।

प्रलेखन

वर्डप्रेस आधिकारिक तौर पर वर्डप्रेस कोडेक्स में प्रलेखित है WordPress.org पर . [डेवलपर्स](#) वर्डप्रेस के साथ काम करने से डेवलपर कोडेक्स में विशेष रुचि होगी अनुभाग और [डेवलपर संदर्भ WordPress.org का अनुभाग](#) ।

संस्करणों

संस्करण रिलीज़	दिनांक
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

संस्करण रिलीज़	दिनांक
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

उदाहरण

वर्डप्रेस का परिचय

WordPress के [WP] ऐप्स, वेबसाइट आदि बनाने के लिए एक खुला स्रोत सामग्री प्रबंधन प्रणाली है

ब्लॉग. WP PHP में लिखा गया है और उपयोगकर्ता सामग्री के लिए डेटा स्टोर के रूप में MySQL का उपयोग करता है

विन्यास। इसमें प्लगइन्स का एक समृद्ध पारिस्थितिकी तंत्र है और थीम और एक जीवंत ओपन सोर्स समुदाय, अच्छे दस्तावेज़ीकरण और प्रवेश के लिए कम बाधाओं का आनंद लेता है। प्रयोज्यता और डेवलपर दस्तावेज़ WP कोडेक्स में पाए जा सकते हैं।

वर्डप्रेस का एक हिस्सा जो इसे अधिकांश अन्य सीएमएस उत्पादों से अलग बनाता है वह इसकी इवेंट ड्रिवेन प्रोग्रामिंग है। यह एमवीसी (मॉडल व्यू कंट्रोलर) आर्किटेक्चर की तुलना में प्रोग्रामिंग और लॉजिक प्रतिनिधित्व का एक अलग तरीका है जिसका उपयोग अधिकांश सीएमएस सिस्टम द्वारा किया जाता है। वर्डप्रेस एक्शन और फिल्टर की अवधारणाओं का उपयोग करता है। वे घटनाओं की एक कतार बनाते हैं जो प्लगइन्स और थीम को अंतिम वेब एप्लिकेशन पेज और/या भागों को सम्मिलित करने, संशोधित करने या यहां तक कि हटाने की अनुमति देते हैं। एक समान अवधारणा JIT या जस्ट-इन-टाइम कंपाइलिंग है।

जबकि ऐतिहासिक रूप से वर्डप्रेस को एक ब्लॉगिंग प्लेटफॉर्म के रूप में जाना जाता है, और यह इस कलंक को कभी नहीं खो सकता है, कोर वर्डप्रेस टीम का फोकस स्पष्ट रूप से बदल गया है। 2016 की स्थिति के साथ, संस्थापक मैथ्यू मुलेनवेग द्वारा, हम लक्ष्य, दृष्टिकोण और प्रयास में स्पष्ट बदलाव देख सकते हैं। 2016 में, हमने आश्चर्यजनक प्रगति देखी जब वर्डप्रेस कोर ने अधिकांश लोकप्रिय REST API प्लगइन को अपनाया। यह स्पष्ट रूप से कोर टीम का इरादा था जब उन्होंने एक फ्रंट-एंड जावास्क्रिप्ट सीएमएस एडमिन पैनल बनाने का साहसिक प्रयास शुरू किया, जो कि कई वर्षों से देखे गए सुनहरे मानक से अलग है; उन्होंने इसे कैल्पिसो कहा।

वर्डप्रेस थीम्स

यूआरएल को विशिष्ट टेम्पलेट्स में मैप करना

वर्डप्रेस थीम को पूरी तरह से समझने के लिए, आपको दो प्राथमिक अवधारणाओं को समझना होगा:

1. पर्मालिंक्स 2.

टेम्पलेट पदानुक्रम

पर्मालिंक एक स्थायी, गैर-परिवर्तनशील यूआरएल (या किसी विशिष्ट संसाधन का लिंक) है। उदाहरण के लिए:

• example.com/about-us/ (WP में एक पेज) • example.com/services/ (एकाधिक वस्तुओं की सूची, जिसे WP भाषा में "संग्रह" भी कहा जाता है) • [example.com/services/we-can- वह-वह-तुम्हारे लिए/](http://example.com/services/we-can-वह-वह-तुम्हारे-लिए/) (एक व्यक्तिगत आइटम)

जब कोई उपयोगकर्ता यूआरएल का अनुरोध करता है, तो वर्डप्रेस यह पता लगाने के लिए पर्मालिंक को रिवर्स-इंजीनियर करता है कि किस टेम्पलेट को इसके लेआउट को नियंत्रित करना चाहिए। वर्डप्रेस विभिन्न टेम्पलेट फ़ाइलों की तलाश करता है जो सामग्री के इस विशेष भाग को नियंत्रित कर सकें, और अंततः जो सबसे विशिष्ट फ़ाइल मिलती है उसे प्राथमिकता देता है।

इसे टेम्पलेट पदानुक्रम के रूप में जाना जाता है।

एक बार जब WP को पदानुक्रम में मिलान दृश्य टेम्पलेट मिल जाता है, तो यह पृष्ठ को संसाधित करने और प्रस्तुत करने के लिए उस फ़ाइल का उपयोग करता है।

उदाहरण के लिए: [Index.php](#) (डिफ़ॉल्ट, "कैच-ऑल" टेम्पलेट) को [Archive.php](#) (सूची-आधारित सामग्री के लिए डिफ़ॉल्ट टेम्पलेट) द्वारा ओवरराइड किया जाएगा, जो बदले में [Archive-services.php](#) (एक टेम्पलेट) द्वारा ओवरराइड किया जाएगा विशेष रूप से "सेवाएँ" नामक संग्रह के लिए फ़ाइल)।

[यहां टेम्पलेट पदानुक्रम के लिए एक बेहतरीन दृश्य संदर्भ है](#)

मूल थीम निर्देशिका संरचना

एक साधारण विषय कुछ इस तरह दिखता है:

```
// थीम सीएसएस
स्टाइल.सीएसएस

// आपके थीम फ़ंक्शन के लिए कस्टम कार्यक्षमता.php

// बाद की थीम फ़ाइलों में शामिल करने के लिए आंशिक भाग हेडर.पीएचपी फूटर.पीएचपी साइडबार.पीएचपी
टिप्पणियां.पीएचपी

// "अभिलेख", (एकाधिक पोस्ट वाले दृश्यों की सूची) Archive.php लेखक.php date.php वर्गीकरण.php टैग.php श्रेणी.php

// व्यक्तिगत सामग्री पृष्ठ // ध्यान दें कि होम और फ्रंटपेज टेम्पलेट
अनुशंसित नहीं हैं // और उन्हें पेज टेम्पलेट सिंगल.php सिंगल.php पेज.php फ्रंट-पेज.php होम.php द्वारा प्रतिस्थापित किया जाना चाहिए

// विविध। यूटिलिटी पेज इंडेक्स.php (एक कैच-
ऑल अगर कुछ और मेल नहीं खाता) search.php अटैचमेंट.php इमेज.php 404.php
```

"एकल" का उदाहरण (व्यक्तिगत पोस्ट के लिए टेम्पलेट)

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

<?php if ( has_posts() ) while ( has_posts() ) <h1><?php the_title(); ?></h1> <?php : पोस्ट(); ?>
the_content(); ?> <?php comments_template('',
true ); ?> <?php एंडव्हाइल; ?>

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

यहाँ क्या हो रहा है? सबसे पहले, यह हेडर.php (PHP के समान या आवश्यकता के समान) लोड करता है, लूप सेट करता है, शीर्षक और सामग्री प्रदर्शित करता है, फिर टिप्पणियाँ.php, साइडबार.php और footer.php शामिल करता है।

लूप भारी भारोत्तोलन करता है, एक पोस्ट ऑब्जेक्ट सेट करता है, जिसमें वर्तमान में देखी गई सामग्री के लिए सभी जानकारी होती है।

"संग्रह" का उदाहरण (एकाधिक पोस्ट की सूची के लिए टेम्पलेट)

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

<?php अगर (हैव_पोस्ट्स()) जबकि (हैव_पोस्ट्स()) : पोस्ट(); ?> <a href='<?php
the_permalink(); ?>'<?php the_title(); ?></a> <?php the_excerpt(); ?> <?php एंडव्हाइल; ?>

<?php
Next_posts_link('पुरानी प्रविष्टियाँ', $the_query->max_num_pages ); पिछला_पोस्ट_लिंक ('नई प्रविष्टियाँ');

?>

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

सबसे पहले, इसमें हेडर.php शामिल है, लूप सेट करता है, और साइडबार.php, और footer.php शामिल है। लेकिन इस मामले में लूप में कई पोस्ट हैं, इसलिए इसके बजाय व्यक्तिगत पोस्ट के लिंक के साथ एक अंश दिखाया गया है। नेक्स्ट_पोस्ट्स_लिंक और प्रीवियस_पोस्ट्स_लिंक भी शामिल हैं ताकि संग्रह परिणामों को पृष्ठांकित कर सके।

पोस्ट, पेज, कस्टम पोस्ट प्रकार, और तटकर क्षेत्र

बॉक्स से बाहर, वर्डप्रेस दो प्रकार की सामग्री का समर्थन करता है: पोस्ट और पेज। पोस्ट का उपयोग आम तौर पर ब्लॉग पोस्ट जैसी गैर-पदानुक्रमित सामग्री के लिए किया जाता है। पेजों का उपयोग स्थिर, स्टैंडअलोन सामग्री के लिए किया जाता है जैसे हमारे बारे में पृष्ठ, या कंपनी के सेवा पृष्ठ जिसके नीचे नेस्टेड उप-पृष्ठ होते हैं।

संस्करण 3.0 के अनुसार, डेवलपर्स वर्डप्रेस की कार्यक्षमता को बुनियादी बातों से परे बढ़ाने के लिए अपने स्वयं के कस्टम पोस्ट प्रकारों को परिभाषित कर सकते हैं। कस्टम पोस्ट प्रकारों के अलावा, आप अपने पोस्ट/पेज/कस्टम पोस्ट प्रकारों से जुड़ने के लिए अपने स्वयं के कस्टम फ़ील्ड भी बना सकते हैं, जिससे आप अपने टेम्पलेट्स के भीतर मेटाडेटा को जोड़ने और एक्सेस करने का एक संरचित तरीका प्रदान कर सकते हैं। देखें: उन्नत कस्टम फ़ील्ड.

अध्याय 2: क्रियाएँ और फ़िल्टर

वाक्य - विन्यास

- `add_action` (टैग, `function_to_call`, प्राथमिकता, `num_of_args`);
- `add_filter` (टैग, `function_to_call`, प्राथमिकता, `num_of_args`);

पैरामीटर

पैरामीटर स्पष्टीकरण	
\$टैग	(स्ट्रिंग) (आवश्यक) उस क्रिया का नाम जिससे \$फ़ंक्शन जुड़ा हुआ है।
\$फ़ंक्शन	(कॉल करने योग्य) (आवश्यक) फ़ंक्शन नाम या अनाम फ़ंक्शन वाली एक स्ट्रिंग की आवश्यकता है। कक्षाओं के भीतर फ़ंक्शन जोड़ने के लिए उदाहरण देखें।
\$प्राथमिकता	(int) डिफ़ॉल्ट = 10. हुक/फ़िल्टर से जुड़े फ़ंक्शन निर्दिष्ट प्राथमिकता में चलेंगे। आपके पास ऐसी स्थिति हो सकती है जहां आप किसी भी अन्य कार्रवाई से पहले कोड के साथ काम करना चाहते हैं, प्राथमिकता = 1 या अन्य सभी संलग्न कार्यों के बाद प्राथमिकता = 100 आदि सेट करना चाहते हैं। सभी PHP फ़ंक्शंस की तरह, आप किसी वेरिएबल के लिए मान पास किए बिना फ़ंक्शन का उपयोग कर सकते हैं जहां एक डिफ़ॉल्ट मान सेट किया गया है, लेकिन यदि आप लौटाए गए पैरामीटर की संख्या बदलना चाहते हैं, तो आपको निर्दिष्ट करना होगा!
\$पैरामीटर	(int) डिफ़ॉल्ट = 1. आपके संलग्न फ़ंक्शन पर लौटाए गए पैरामीटरों की संख्या। लौटाए गए पैरामीटर उस संख्या पर निर्भर होंगे जहां हुक बनाया गया था। अधिक विवरण के लिए <code>apply_filters()</code> और <code>do_action()</code> देखें।

टिप्पणियां

वर्डप्रेस हुक

वर्डप्रेस के साथ काम करना शुरू करते समय कुछ ऐसी चीजें जो अक्सर डेवलपर्स को भ्रमित करती हैं, वह है `apply_filters()` और `add_action()` का उपयोग। आप अक्सर प्लगइन/थीम को कोड में इनका उपयोग करते हुए देखेंगे और यदि आप अवधारणा को नहीं समझते हैं, तो आपके लिए उनके साथ काम करना कठिन होगा।

संक्षेप में (बहुत संक्षेप में, प्रक्रिया के लिए वर्डप्रेस लोड फ़्लोचार्ट को विस्तार से देखें), वर्डप्रेस निम्नलिखित तरीके से लोड होता है:

1. `wp-load.php` - फ़ंक्शंस आदि
2. म्यू-प्लगइन्स - म्यू-प्लगइन्स फ़ोल्डर में पाई जाने वाली कोई भी फ़ाइल - अक्सर कैशड ऑब्जेक्ट्स को परोसने के लिए उपयोग की जाती है
3. प्लगइन्स - कोई विशेष ऑर्डर नहीं, कोई भी स्थापित और सक्रिय प्लगइन्स लोड किया जाएगा
4. सक्रिय चाइल्ड थीम/पैरेंट थीम
5. `init` - शेष डेटा
6. टेम्पलेट

यदि आप एक डेवलपर हैं और फ़ंक्शन फ़ाइल के साथ काम कर रहे हैं, तो आप देख सकते हैं कि जिन फ़ाइलों के साथ आप काम कर रहे हैं उनकी तुलना में दोनों प्रक्रिया में पहले लोड हो गए हैं। मतलब आप प्रक्रियाओं को संशोधित नहीं कर सकते (ध्यान दें कि आप फ़ंक्शंस को अधिलेखित नहीं कर सकते) या वेरिबल जो बाद में चलते हैं या जिन्हें अभी तक परिभाषित नहीं किया गया है। इसके अलावा थीम डेवलपर्स प्लगइन्स को हुक करने की अनुमति देने के लिए अपने कोड में हुक लगा सकते हैं या प्लगइन्स अन्य प्लगइन्स को उनके वेरिबल्स को ओवरराइट करने की अनुमति दे सकते हैं। अब यह अब तक भ्रमित करने वाला हो सकता है, लेकिन वहीं रुकें।

Add_filter() और add_action() को समझने के लिए हमें सबसे पहले यह देखना होगा कि हुक कैसे बनाए जाते हैं।

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

जब आप वर्डप्रेस में उपरोक्त का सामना करते हैं, तो यह हुक im_a_hook से जुड़े किसी भी फ़ंक्शन को कॉल करेगा (प्रक्रिया पर जानकारी के लिए \$wp_filter देखें)। आपके संलग्न फ़ंक्शन में \$arga संलग्न फ़ंक्शन के साथ काम करने के लिए उपलब्ध होगा।

```
add_action('im_a_hook', 'संलग्न_फ़ंक्शन');  
  
फ़ंक्शन संलग्न_फ़ंक्शन ($arga){ इको $arga;  
  
}
```

यह लोड प्रक्रिया के कुछ बिंदुओं पर चर को संशोधित करने के लिए शक्तिशाली नए अवसर खोलता है।

याद रखें हमने पहले कहा था कि टेम्पलेट प्लगइन्स/थीम के बाद लोड किए जाते हैं? एक सामान्य प्लगइन WooCommerce है जो प्रक्रिया में बाद में स्क्रीन बनाता है, मैं इसका दस्तावेजीकरण नहीं करने जा रहा हूँ कि कैसे, लेकिन do_action का एक उदाहरण प्लगइन में पाया जा सकता है।

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

यहां हमने एक हुक बनाया है जो किसी भी वेरिबल को पीछे नहीं छोड़ता है, लेकिन हम अभी भी इसका आनंद ले सकते हैं:

```
add_action('woocommerce_after_add_to_cart_button', 'special_offer');  
  
फ़ंक्शन विशेष_ऑफर(){ इको '<h1>विशेष ऑफर!</h1>';  
  
}
```

उपरोक्त ऐड_एक्शन विशेष ऑफर के शीर्षक को प्रतिध्वनित करेगा जहां

do_action('woocommerce_after_add_to_cart_button') स्थित है जो WooCommerce स्क्रीन बनाते समय होता है। तो हम html डालने के लिए इस हुक का उपयोग कर सकते हैं। अन्य उपयोगों में पूरी तरह से एक अलग स्क्रीन पर रीडायरेक्ट करना आदि शामिल हो सकता है।

इसके अलावा फ़ंक्शन में कई वेरिबल भी पास किए जा सकते हैं। इसे अपने थीम फ़ंक्शंस में आजमाएँ। अंतिम पैरामीटर पर ध्यान दें जिसे हम 3 पर सेट कर रहे हैं, क्योंकि हम 3 उपलब्ध पैरामीटर के साथ काम करना चाहते हैं। यदि हमने इसे 2 में बदल दिया, तो केवल 2 ही लौटाया जाएगा और हमें एक अपरिभाषित त्रुटि मिलेगी।

```
add_action('कस्टम_हुक', 'अटैच_फ़ंक्शन', 10, 3);  
  
फ़ंक्शन संलग्न_फ़ंक्शन($a,$b,$c){
```

```

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

}

$अर्ग = 1; $argb
= 2; $argc = 3;

do_action('custom_hook', $arga, $argb, $argc); बाहर निकलना;

```

एक अन्य WP हुक प्रकार है जिसे फ़िल्टर कहा जाता है। एक फ़िल्टर अपने उपयोग में एक क्रिया से भिन्न होता है, एक क्रिया केवल वेरिएबल प्राप्त कर सकती है, जाहिर है ये वेरिएबल फ़ंक्शंस के दायरे में हैं (आपको पता होना चाहिए कि php स्कोप क्या है, यदि Google नहीं है)। फ़िल्टर लौटाए गए डेटा को वापस भेजते हैं, ताकि आप वेरिएबल को संशोधित करने के लिए उपयोग कर सकें।

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

जहां आप ऊपर देखते हैं, आप \$filter_me के मान को संशोधित कर सकते हैं क्योंकि आपके द्वारा लौटाया गया कोई भी डेटा वेरिएबल में संग्रहीत मान होगा। तो उदाहरण के लिए (ध्यान दें कि हम उदाहरण में \$variable_to_filter को \$filter_me में बदल रहे हैं):

```

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

फ़ंक्शन संलग्न_फ़ंक्शन($फ़िल्टर_मी){

    $filter_me='रे';

    $filter_me लौटाएँ;

}

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

```

\$ filter_me वेरिएबल में अब 'बॉब' के बजाय 'रे' होगा, हमने 100 की प्राथमिकता निर्धारित की है, इसलिए हम काफी हद तक आश्वस्त हैं कि उपयोग के बाद कोई भी मूल्य नहीं बदल रहा है (एक ही हुक पर कई फ़िल्टर चल सकते हैं) इसलिए हम यदि apply_filters() मौजूद है तो अब प्रक्रिया में बाद में उपयोग किए गए वेरिएबल्स को बदल सकते हैं।

आप एकाधिक पैरामीटर भी पास कर सकते हैं, लेकिन आप केवल एक का मान बदल सकते हैं। आपको एक मान भी लौटाना होगा, अन्यथा आपके वेरिएबल में कुछ भी नहीं होगा। यदि आप समझते हैं कि आप वेरिएबल्स को मान/सरणी/ऑब्जेक्ट निर्दिष्ट करने के लिए PHP का उपयोग कैसे करते हैं तो यह आपके लिए स्पष्ट होगा, उदाहरण के लिए:

```

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

फ़ंक्शन संलग्न_फ़ंक्शन ($ फ़िल्टर_मी, $ arga, $ argb) {

    $filter_me= 'रे'.$arga.$argb;

```

```

    $arga= 'तुम मूर्ख';

    $filter_me लौटाएँ;

}

$filter_me = 'बॉब';

$arga =      ' मध्य नाम';
$argb =      ' उपनाम';

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

```

\$filter_me वेरिएबल में अब 'रे मिडिलनेम उपनाम' शामिल है। लेकिन \$arga के बारे में क्या? इसमें अभी भी 'मध्यनाम' शामिल है, हमारे फ़ंक्शन के भीतर \$arga को 'यू फूल' में बदलने से इसके दायरे के बाहर परिभाषित मूल्य पर कोई प्रभाव नहीं पड़ता है (इसके तरीके, Google ग्लोबल आदि हैं)

```
add_action($हुक_नाम, $फ़ंक्शन, $प्राथमिकता, $पैरामीटर)
```

```
add_filter($हुक_नाम, $फ़ंक्शन, $प्राथमिकता, $पैरामीटर);
```

उदाहरण

add_action - init

```

add_action('init', 'process_post');

फ़ंक्शन प्रोसेस_पोस्ट(){ if($_POST)
    var_dump($_POST);

}

```

add_action - init - अनाम फ़ंक्शन

```

add_action('init' echo 'मैंने कुछ , समारोह(){
    किया';
});

```

add_action - init - क्लास ऑब्जेक्ट के भीतर

```

कक्षा नमूना{

    सार्वजनिक फ़ंक्शन __construct(){ add_action('init',
        array($this, 'samp') );
    }

    सार्वजनिक फ़ंक्शन सैंप() { // सार्वजनिक होना चाहिए!! प्रतिध्वनि 'मैंने कुछ किया';

    }

}

नया नमूना();

```

add_action - init - स्थिर वर्ग के भीतर

```
कक्षा नमूना{
```

```
    सार्वजनिक स्थैतिक फ़ंक्शन add_action_func();//नोट __CLASS__ में कोई भी नेमस्पेसिंग  
        add_action('init', array(__CLASS__, 'samp') );
```

```
}
```

```
    सार्वजनिक स्थैतिक फ़ंक्शन सैंप(){ इको 'मैंने कुछ किया';
```

```
}
```

```
}
```

```
नमूना::add_action_func();
```

अध्याय 3: शॉर्टकोड जोड़ें

वाक्य - विन्यास

- `add_shortcode($टैग, $func);`

पैरामीटर

पैरामीटर विवरण	
\$टैग	(स्ट्रिंग) (आवश्यक) पोस्ट सामग्री में खोजा जाने वाला शॉर्टकोड टैग
\$func	(कॉल करने योग्य) (आवश्यक) शॉर्टकोड मिलने पर चलाने के लिए हुक

टिप्पणियां

- शॉर्टकोड कॉलबैक में तीन तर्क पारित किए जाएंगे: शॉर्टकोड विशेषताएँ, शॉर्टकोड सामग्री (यदि कोई हो), और शॉर्टकोड का नाम।
- प्रत्येक शॉर्टकोड के लिए केवल एक हुक हो सकता है। जिसका अर्थ है कि यदि किसी अन्य प्लगइन में एक समान शॉर्टकोड है, तो यह आपके या आपके प्लगइन को ओवरराइड कर देगा, यह इस पर निर्भर करता है कि प्लगइन किस क्रम में शामिल किए गए हैं और/या चलाए गए हैं।
- हैंडलर फ़ंक्शन में पास होने से पहले शॉर्टकोड विशेषता नाम हमेशा लोअरकेस में परिवर्तित किए जाते हैं। मूल्य अछूते हैं।
- ध्यान दें कि शॉर्टकोड द्वारा बुलाए गए फ़ंक्शन को कभी भी किसी भी प्रकार का आउटपुट नहीं देना चाहिए। शॉर्टकोड फ़ंक्शंस को वह टेक्स्ट वापस करना चाहिए जिसका उपयोग शॉर्टकोड को बदलने के लिए किया जाना है। सीधे आउटपुट उत्पन्न करने से अप्रत्याशित परिणाम प्राप्त होंगे। यह फ़िल्टर फ़ंक्शंस के व्यवहार के समान है, जिसमें उन्हें कॉल से अपेक्षित दुष्प्रभाव उत्पन्न नहीं करना चाहिए, क्योंकि आप यह नियंत्रित नहीं कर सकते कि उन्हें कब और कहाँ से कॉल किया जाता है।

उदाहरण

हालिया पोस्ट के लिए सरल शॉर्टकोड

`add_shortcode` wp कीवर्ड है।

```
//हाल ही की पोस्टें हमारा शॉर्टकोड बनने जा रही हैं। add_shortcode('recent-posts',
'recent_posts_function');

// यह फ़ंक्शन तब कार्यवाही कर रहा है जब हालिया पोस्ट शॉर्टकोड को कॉल किया जाता है। फ़ंक्शन हाल ही में पोस्ट्स_फ़ंक्शन() { query_posts(array('orderby'
=> 'date', 'order' => 'DESC' if(have_posts())) :
, 'शोपोस्ट' => 1));

जबकि (have_posts()) : the_post();
    $रिटर्न_स्ट्रिंग = '<a href="'.get_permalink().'">'.get_the_title().</a>';
    अंत तक;
```

```

अगर अंत;
wp_reset_query(); वापसी $
वापसी_स्ट्रिंग;
}

```

इस स्निपेट को आपके थीम function.php में रखा जा सकता है।

[हाल ही के पोस्ट] यह हालिया पोस्ट का शॉर्टकोड है। इस शॉर्टकोड को हम बैकएंड (जैसे पेज, पोस्ट, विजेट) में लागू कर सकते हैं।

हम अपने कोड के अंदर भी उसी शॉर्टकोड का उपयोग कर सकते हैं। do_shortcode की सहायता से।

जैसे. इको डू शॉर्टकोड ('[हालिया-पोस्ट]');

हाल की पोस्टों के लिए उन्नत शॉर्टकोड

यह फ़ंक्शन पैरामीटर लेता है कि आप कितने हालिया पोस्ट प्रदर्शित करना चाहते हैं।

उदाहरणार्थ: आप केवल पाँच हालिया पोस्ट प्रदर्शित करना चाहते हैं। अभी-अभी पोस्ट='5' के साथ तर्क पारित किये गये हैं (आप जितने भी हाल के पोस्ट प्रदर्शित करना चाहते हैं, उन्हें पारित कर सकते हैं)।

फ़ंक्शन डेटाबेस से केवल पांच हालिया पोस्ट लाता है।

```

//हाल ही की पोस्टें हमारा शॉर्टकोड बनने जा रही हैं। add_shortcode('recent-posts',
'recent_posts_function');

// फ़ंक्शन पोस्ट = "5" जैसे पैरामीटर लेता है। फ़ंक्शन हाल ही में_पोस्ट_फ़ंक्शन($atts){

    निकालें(shortcode_atts(संरणी(
        'पोस्ट' => 1,
    ), $atts));

    $रिटर्न_स्ट्रिंग = '<ul>';
    query_posts(array('orderby' => 'date', 'order' => 'DESC' if (have_posts()) :
        , 'शोपोस्ट' => $पोस्ट));

    जबकि (have_posts()) : the_post();
        $return_string .= '<li><a href="'.get_permalink().">'.get_the_title().</a></li>';
    अंत तक; अगर
अंत;
$return_string .= '</ul>';

wp_reset_query(); वापसी $
वापसी_स्ट्रिंग;
}

```

जैसे. इको do_shortcode('[हालिया-पोस्ट पोस्ट='5']');

अध्याय 4: user_contactmethods फ़िल्टर हुक वाले उपयोगकर्ताओं के लिए संपर्क जानकारी जोड़ें/हटाएँ

उदाहरण

सबसे लोकप्रिय सामाजिक नेटवर्क को सक्षम करना

```
फ़ंक्शन सोशल_प्रोफ़ाइल्स ($ संपर्क विधियां) {  
  
    $contactmethods['facebook_profile'] = 'फ़ेसबुक प्रोफ़ाइल यूआरएल'; $contactmethods['twitter_profile'] = 'ट्विटर प्रोफ़ाइल  
    URL'; $contactmethods['google_profile'] $contactmethods['linkedin_profile'] = 'LinkedIn प्रोफ़ाइल URL';  
    $contactmethods['github_profile'] = 'Google प्रोफ़ाइल URL';  
    $contactmethods['behance_profile'] = 'Behance प्रोफ़ाइल URL'; $contactmethods['dribbble_profile'] = 'ड्रिबल प्रोफ़ाइल  
    URL'; $contactmethods['stack_profile'] = 'गिटहब प्रोफ़ाइल यूआरएल';  
    $contactmethods['twitch_profile'] $contactmethods['angellist_profile'] = 'AngelList प्रोफ़ाइल URL';  
  
    = 'स्टैक एक्सचेंज प्रोफ़ाइल यूआरएल';  
    = 'ट्विच प्रोफ़ाइल यूआरएल';  
  
    $contactmethods वापस करें;  
}  
  
add_filter('user_contactmethods', 'social_profiles', 10, 1);
```

आपको ये फ़ाइलें आपके डैशबोर्ड में मिलेंगी:

The screenshot shows the WordPress 'Users' settings page. On the left is a dark sidebar menu with options: Dashboard, Posts, Media, Pages, Comments, Cheat sheets, Tips & tricks, Appearance, Plugins (with a '2' badge), Users (highlighted), All Users, Add New, Your Profile, Tools, Settings, Custom Fields, and MailChimp for WP. The main content area lists social media profile URL fields: Facebook Profile URL, Twitter Profile URL, Google Profile URL, LinkedIn Profile URL, GitHub Profile URL, Behance Profile URL, Dribbble Profile URL, Stack Exchange Profile URL, Twitch Profile URL, and AngelList Profile URL. Each field is represented by a white text input box on a light gray background.

और इस प्रकार आप इसे कोड में पुनर्प्राप्त करते हैं

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

संपर्क विधि हटाना

```

फ़ंक्शन रिमूव_कॉन्टैक्ट_मैथोड्स ($ संपर्कमैथोड्स) {
    अनसेट($contactmethods['facebook_profile']);
    अनसेट($contactmethods['twitter_profile']);

    $contactmethods वापस करें;
}

add_filter('user_contactmethods', 'remove_contact_methods', 10, 1);

```

अध्याय 5: add_action()

वाक्य - विन्यास

- ऐड_एक्शन (\$टैग, \$फंक्शन_टू_ऐड) • ऐड_एक्शन (\$टैग, \$फंक्शन_टू_ऐड, \$प्राथमिकता) • ऐड_एक्शन (\$टैग, \$फंक्शन_टू_ऐड, \$प्राथमिकता, \$स्वीकृत_आर्ग्स)

पैरामीटर

पैरामीटर	विवरण
\$टैग	(स्ट्रिंग) उस क्रिया का नाम जिससे प्रक्रिया \$function_to_add जुड़ी होगी।
\$function_to_add (कॉल करने योग्य)	वह कॉल करने योग्य फंक्शन/प्रक्रिया जिसे आप कॉल करना चाहते हैं।
\$प्राथमिकता	(int) प्राथमिकता स्तर जिस पर \$function_to_add निष्पादित किया जाएगा। वैकल्पिक। डिफ़ॉल्ट 10.
\$accepted_args	(int) कॉल करने योग्य फंक्शन \$function_to_add द्वारा स्वीकार किए जाने वाले तर्कों की संख्या। वैकल्पिक। डिफ़ॉल्ट 1.

उदाहरण

डायरेक्ट फंक्शन कॉलबैक

```
add_action('init', function() { //यहां कुछ करें  
});
```

निर्देशों के एक सेट को हुक करने के लिए फंक्शन ब्लॉक का उपयोग करना। इनिट हुक के साथ , निर्देशों का सेट वर्डप्रेस द्वारा आवश्यक घटकों को लोड करने के तुरंत बाद निष्पादित किया जाएगा।

फंक्शन नाम संदर्भ कॉलबैक

```
फंक्शन my_init_function() { // यहां कुछ करें  
}  
  
add_action('init', 'my_init_function');
```

निर्देशों के एक सेट को हुक करने के लिए फंक्शन के नाम का उपयोग करना। इनिट हुक के साथ , निर्देशों का सेट वर्डप्रेस द्वारा आवश्यक घटकों को लोड करने के तुरंत बाद निष्पादित किया जाएगा।

क्लास स्टैटिक विधि कॉलबैक

```
क्लास MyClass { स्टैटिक
    फ़ंक्शन my_init_method() { // यहां कुछ करें
    }
}

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

निर्देशों के एक सेट को हुक करने के लिए कक्षा की स्थिर विधि का उपयोग करना। इनिट हुक के साथ , निर्देशों का सेट वर्डप्रेस द्वारा आवश्यक घटकों को लोड करने के तुरंत बाद निष्पादित किया जाएगा।

ऑब्जेक्ट विधि कॉलबैक

```
कक्षा MyClass {फ़ंक्शन
    my_init_method() {
        //यहाँ कुछ करो
    }
}

$obj = नया MyClass();

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

निर्देशों के एक सेट को हुक करने के लिए किसी ऑब्जेक्ट की विधि का उपयोग करना। इनिट हुक के साथ , निर्देशों का सेट वर्डप्रेस द्वारा आवश्यक घटकों को लोड करने के तुरंत बाद निष्पादित किया जाएगा।

अध्याय 6: add_editor_style()

परिचय

फ़ंक्शन उपयोगकर्ता को TinyMCE संपादक के लिए स्टाइलशीट लोड करने की अनुमति देता है

वाक्य - विन्यास

- add_editor_style (\$ स्टाइलशीट)

पैरामीटर

पैरामीटर विवरण
(सरणी या स्ट्रिंग) (वैकल्पिक) थीम रूट के सापेक्ष स्टाइलशीट नाम या उसकी सरणी। 'editor-style.css' पर डिफ़ॉल्ट \$ स्टाइलशीट

उदाहरण

एकल सीएसएस फ़ाइल लोड हो रही है

कोड

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

व्याख्या

उपरोक्त कोड में, हमने सीएसएस फ़ाइल लोड करने के लिए add_editor_style का उपयोग किया। हमने यह सुनिश्चित करने के लिए add_action का भी उपयोग किया कि वर्डप्रेस हमारा फ़ंक्शन चलाता है।

अध्याय 7: add_menu_page()

परिचय

यह फ़ंक्शन एडमिन पैनल के नेव बार में एक आइटम जोड़ने के लिए है।

वाक्य - विन्यास

- `add_menu_page($ पेज_शीर्षक, $ मेनू_शीर्षक, $ क्षमता, $ मेनू_स्लग, $ फ़ंक्शन, $ आइकन_यूआरएल, $ स्थिति)`

पैरामीटर

पैरामीटर	विवरण
<code>\$पेज_शीर्षक</code>	(स्ट्रिंग) मेनू चयनित होने पर पृष्ठ के शीर्षक टैग में प्रदर्शित होने वाला पाठ।
<code>\$menu_title</code>	(स्ट्रिंग) मेनू के लिए उपयोग किया जाने वाला टेक्स्ट।
<code>\$क्षमता</code>	(स्ट्रिंग) इस मेनू को उपयोगकर्ता को प्रदर्शित करने के लिए आवश्यक क्षमता।
<code>\$menu_slug</code>	(स्ट्रिंग) इस मेनू को संदर्भित करने के लिए स्लग नाम (इस मेनू के लिए अद्वितीय होना चाहिए)।
<code>\$फ़ंक्शन</code>	(कॉल करने योग्य) (वैकल्पिक) इस पृष्ठ के लिए सामग्री को आउटपुट करने के लिए कॉल किया जाने वाला फ़ंक्शन।
<code>\$icon_url</code>	(स्ट्रिंग) (वैकल्पिक) इस मेनू के लिए उपयोग किए जाने वाले आइकन का यूआरएल।
<code>\$स्थिति</code>	(इंट) (वैकल्पिक) मेनू क्रम में यह स्थिति दिखाई देनी चाहिए।

टिप्पणियां

यहां डिफ़ॉल्ट स्थितियों की एक सूची दी गई है (\$स्थिति के लिए)

- 2 - डैशबोर्ड • 4 - विभाजक • 5 - पोस्ट • 10 - मीडिया • 15 - लिंक • 20 - पृष्ठ • 25 - टिप्पणियाँ • 59 - विभाजक

- 60 - उपस्थिति • 65 -
प्लगइन्स • 70 -
उपयोगकर्ता
- 75 - उपकरण
- 80 - सेटिंग्स • 99 -
विभाजक

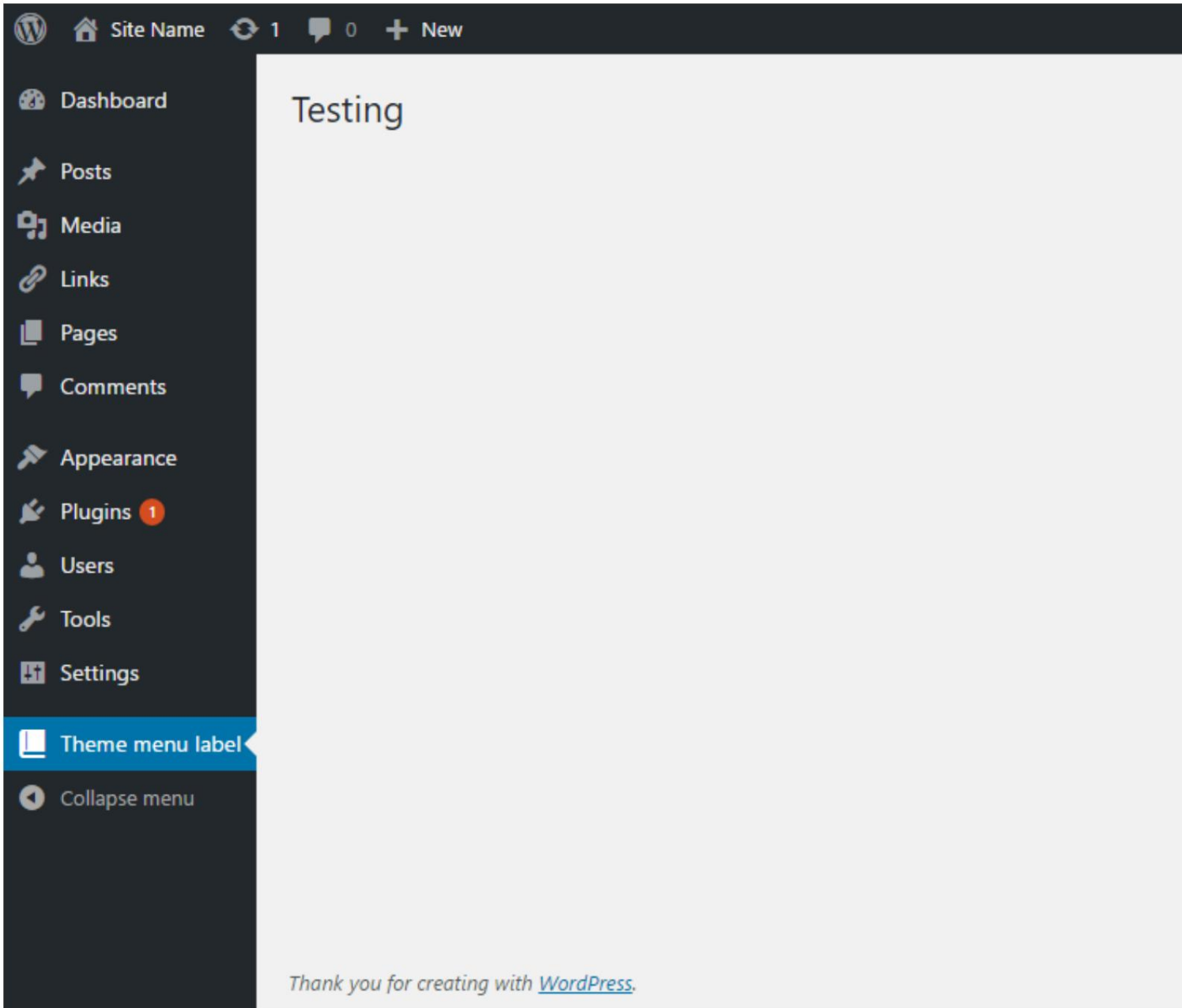
उदाहरण

नेव बार में "थीम पेज शीर्षक" आइटम जोड़ना

कोड

```
फ़ंक्शन add_the_theme_page(){  
    add_menu_page('थीम पृष्ठ शीर्षक', 'थीम मेनू लेबल', 'manage_options', 'थीम-विकल्प',  
'पेज_कंटेंट', 'डैशिकॉन्स-बुक-ऑल्ट'); } add_action('admin_menu',  
  
'add_the_theme_page'); फ़ंक्शन पृष्ठ_सामग्री(){  
  
    इको '<div class='wrap'><h2>परीक्षण</h2></div>';  
}
```

उत्पादन



व्याख्या

कोड में, हमने `add_the_theme_page` नामक एक फ़ंक्शन बनाया और आइटम को नेवबार में जोड़ने के लिए हमने `add_menu_page` का उपयोग किया। हमारे द्वारा पारित तर्कों के बारे में जानने के लिए कृपया इस पृष्ठ में पैरामीटर भाग की जांच करें। फिर हमने अपने `add_the_theme_page` फ़ंक्शन को चलाने के लिए `add_action` का उपयोग किया। अंत में, हमने पेज में सामग्री प्रदर्शित करने के लिए फ़ंक्शन `पेज_कंटेंट` बनाया।

ओओपी और मेनू पेज पर स्क्रिप्ट/शैलियाँ कैसे लोड करें

```
<?php /
*
* प्लगइन नाम: कस्टम एडमिन मेनू */

कक्षा SO_WP_Menu {

    निजी $plugin_url;
```

```

सार्वजनिक फ़ंक्शन __निर्माण() {
    $this->plugin_url = प्लगइन्स_url( '/', __FILE_ ); add_action('plugins_loaded', array($this,
        'init' ));
}

सार्वजनिक फ़ंक्शन init()
{ add_action( 'admin_menu', array($this, 'add_menu' ) );
}

सार्वजनिक फ़ंक्शन add_menu() { $hook =
    add_menu_page(
        'मेरा मेनू', 'मेरा // शीर्षक, html मेटा टैग
        मेनू', // मेनू शीर्षक, हार्डकोडेड शैली 'edit_pages', // क्षमता

        'डमी-पेज-स्लैग', सरणी ($ यह, // यूआरएल
        'सामग्री'), // आउटपुट // आइकन, डिफ़ॉल्ट // स्थिति का उपयोग करता है, अन्य
        व्यर्थ, सभी के शीर्ष पर दिख रहा है
        1
    );
    add_action('admin_print_scripts-$hook', array($this, 'scripts' )); add_action('admin_print_styles-$hook', array($this,
        'styles' ));
}

सार्वजनिक फ़ंक्शन सामग्री() {
    ?>
    <div id='icon-post' class='icon32'></div> <h2>डमी पेज</h2> <p> लोरेम
    इप्सम</p>

    <?php
}

# सीधे मुद्रण, wp_enqueue_script सार्वजनिक फ़ंक्शन स्क्रिप्ट() हो सकता है {

    ?><स्क्रिप्ट>चेतावनी('मेरा पृष्ठ');</स्क्रिप्ट><?php
}

# प्लगइन्स निर्देशिका सार्वजनिक फ़ंक्शन शैलियों() पर एक सीएसएस फ़ाइल से पूछताछ करना {

    wp_enqueue_style('my-menu', $this->plugin_url . 'my-menu.css' );
}
}

नया SO_WP_Menu();

```

इस उदाहरण में ध्यान देने योग्य बात यह है कि, `add_menu_page()` का उपयोग करते समय, यह एक हुक लौटाता है जिसका उपयोग हमारे सटीक पृष्ठ को लक्षित करने और वहां शैलियों और स्क्रिप्ट को लोड करने के लिए किया जा सकता है।

लक्ष्यकरण के बिना कतारबद्ध करना एक सामान्य गलती है और इससे स्क्रिप्ट और शैलियाँ /wp पर फैल जाती हैं व्यवस्थापक।

OOP का उपयोग करके हम आंतरिक तरीकों के बीच उपयोग किए जाने वाले सामान्य चर को संग्रहीत कर सकते हैं।

अध्याय 8: add_submenu_page()

परिचय

यह फ़ंक्शन एडमिन पैनल नेविगेशन बार में किसी मौजूदा आइटम में एक उप-आइटम जोड़ना है।

वाक्य - विन्यास

- `add_submenu_page ($parent_slug, $page_title, $menu_title, $capability, $menu_slug, $function)`

पैरामीटर

पैरामीटर विवरण	
<code>\$parent_slug</code>	(स्ट्रिंग) पैरेंट मेनू के लिए स्लग नाम (या मानक वर्डप्रेस एडमिन पेज का फ़ाइल नाम)।
<code>\$पेज_शीर्षक</code>	(स्ट्रिंग) मेनू चयनित होने पर पृष्ठ के शीर्षक टैग में प्रदर्शित होने वाला पाठ।
<code>\$menu_title</code>	(स्ट्रिंग) मेनू के लिए उपयोग किया जाने वाला टेक्स्ट।
<code>\$क्षमता</code>	(स्ट्रिंग) इस मेनू को उपयोगकर्ता को प्रदर्शित करने के लिए आवश्यक क्षमता।
<code>\$menu_slug</code>	(स्ट्रिंग) इस मेनू को संदर्भित करने के लिए स्लग नाम (इस मेनू के लिए अद्वितीय होना चाहिए)।
<code>\$फ़ंक्शन</code>	(कॉल करने योग्य) (वैकल्पिक) इस पेज के लिए सामग्री को आउटपुट करने के लिए कॉल किया जाने वाला फ़ंक्शन।

टिप्पणियां

यहां `$parent_slug` के लिए स्लग की एक सूची दी गई है

- डैशबोर्ड: 'index.php' • पोस्ट:
'edit.php' • मीडिया:
'upload.php' • पेज:
'edit.php?post_type=page' • टिप्पणियां: 'edit-
comments.php' • कस्टम पोस्ट प्रकार :
'edit.php?post_type=your_post_type' • उपस्थिति: 'themes.php' • प्लगइन्स:
'plugins.php' • उपयोगकर्ता:
'users.php'

- टूल्स: 'tools.php' •
- सेटिंग्स: 'options-general.php' • नेटवर्क
- सेटिंग्स: 'settings.php'

उदाहरण

"सबमेनू पेज" को नेव बार में "टूल्स" के उप-पेज के रूप में जोड़ना

कोड

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

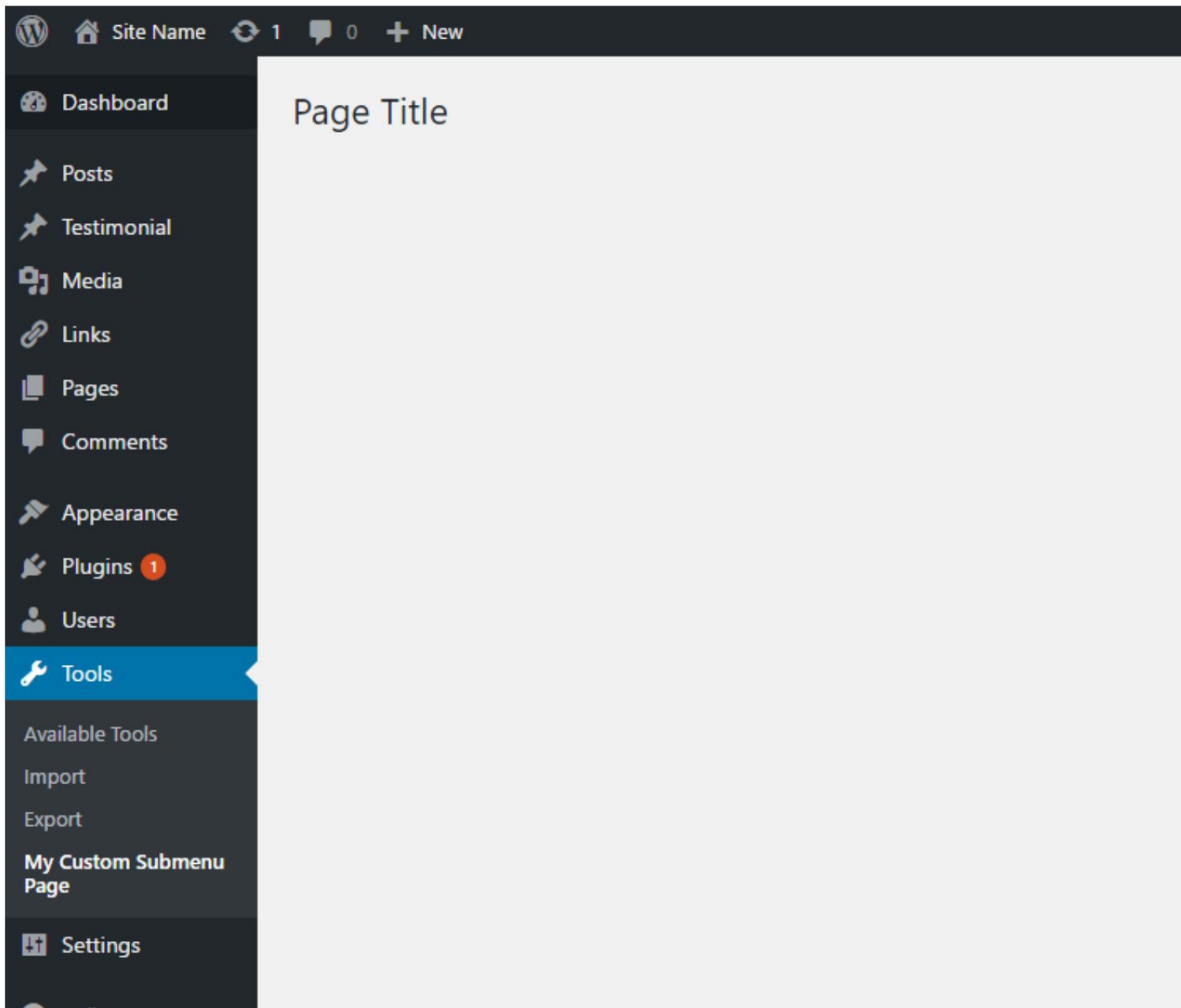
फ़ंक्शन रजिस्टर_माय_कस्टम_सबमेनू_पेज() {
    add_submenu_page('tools.php',
        'Submenu Page',
        'My Custom Submenu
        Page', 'manage_options', 'my-custom-
        submenu-page',
        'my_custom_submenu_page_content' );
}

फ़ंक्शन my_custom_submenu_page_content() { echo '<div class='wrap'>';

    इको '<h2>पेज शीर्षक</h2>'; प्रतिध्वनि '</div>';

}
```

उत्पादन



व्याख्या

कोड में, हमने रजिस्टर_माय_कस्टम_सबमेनू_पेज नामक एक फ़ंक्शन बनाया और हमने आइटम को टूल्स.php के चाइल्ड के रूप में नेवबार में जोड़ने के लिए add_submenu_page का उपयोग किया, जो कि टूल्स पेज है।

हमारे द्वारा पारित तर्कों के बारे में जानने के लिए कृपया इस पृष्ठ में पैरामीटर भाग की जांच करें। फिर हमने अपने रजिस्टर_माय_कस्टम_सबमेनू_पेज फ़ंक्शन को चलाने के लिए add_action का उपयोग किया। अंत में, हमने पेज में सामग्री प्रदर्शित करने के लिए फ़ंक्शन my_custom_submenu_page_content बनाया।

अध्याय 9: add_theme_support()

परिचय

यह फ़ंक्शन उन विशेषताओं को पंजीकृत करता है जिनका थीम समर्थन करता है।

वाक्य - विन्यास

- add_theme_support (\$फीचर)

पैरामीटर

पैरामीटर विवरण	
\$विशेषता	(स्ट्रिंग) सुविधा जोड़ी जा रही है।

टिप्पणियां

\$feature में उपयोग की जाने वाली सुविधाओं की सूची:

- 'पोस्ट-प्रारूप' • 'पोस्ट-थंबनेल' • 'html5' • 'कस्टम-लोगो' • 'कस्टम-हेडर-अपलोड' • 'कस्टम-हेडर' • 'कस्टम-पृष्ठभूमि' • 'शीर्षक-टैग' • 'स्टार्टर-सामग्री'

उदाहरण

पोस्ट प्रारूपों के लिए थीम समर्थन जोड़ना

```
add_theme_support('पोस्ट-फॉर्मेट', ऐसे('फॉर्मेटोन', 'फॉर्मेटो'));
```

पोस्ट में पोस्ट थंबनेल के लिए थीम समर्थन जोड़ना

```
add_theme_support('पोस्ट-थंबनेल', सरणी('पोस्ट'));
```

उपरोक्त कोड सभी पोस्ट पर केवल थंबनेल पोस्ट करने की अनुमति देता है। सभी पोस्ट प्रकारों पर सुविधा सक्षम करने के लिए, यह करें:

```
add_theme_support('पोस्ट-थंबनेल');
```

अध्याय 10: व्यवस्थापक डैशबोर्ड विजेट

परिचय

एडमिन डैशबोर्ड विजेट से आप एडमिन पर किसी भी प्रकार की जानकारी प्रदर्शित करने में सक्षम हैं डैशबोर्ड. आप चाहें तो एकाधिक विजेट बना सकते हैं। आप कोड को function.php में जोड़ सकते हैं आपकी थीम का या आपके प्लगइन का।

वाक्य - विन्यास

- add_action(\$टैग, \$function_to_add, \$प्राथमिकता, \$accepted_args);
- wp_add_dashboard_widget(\$विजेट_id, \$विजेट_name, \$callback, \$control_callback, \$कॉलबैक_आर्ग्स);

पैरामीटर

पैरामीटर	विवरण
\$टैग	(स्ट्रिंग आवश्यक) उस क्रिया का नाम जहां \$function_to_add जुड़ा हुआ है
\$function_to_add (कॉल करने योग्य आवश्यक)	उस फ़ंक्शन का नाम जिसे आप कॉल करना चाहते हैं।
\$प्राथमिकता	(int वैकल्पिक) कार्रवाई के सभी कार्यों में फ़ंक्शन कॉल का स्थान (डिफ़ॉल्ट = 10)
\$accepted_args (int वैकल्पिक)	फ़ंक्शन द्वारा स्वीकार किए जाने वाले पैरामीटरों की संख्या (डिफ़ॉल्ट = 1)
\$विजेट_आईडी	(स्ट्रिंग आवश्यक) आपके विजेट के लिए अनोखा स्लग
\$विजेट_नाम	(स्ट्रिंग आवश्यक) आपके विजेट का नाम (सिर में प्रदर्शित)
\$कॉलबैक	(कॉल करने योग्य आवश्यक) फ़ंक्शन का नाम जो डार्ड सामग्री प्रदर्शित करता है आपका विजेट
\$नियंत्रण_कॉलबैक	(कॉल करने योग्य वैकल्पिक) फ़ंक्शन का नाम जो विजेट विकल्पों को संभालता है फार्म
\$कॉलबैक_आर्ग्स	(सरणी वैकल्पिक) \$control_callback फ़ंक्शन के पैरामीटर

उदाहरण

सरल विजेट (पाठ प्रदर्शित करता है)

यह एक साधारण विजेट जोड़ देगा जो केवल एक छोटा संदेश प्रदर्शित करेगा।

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

```
फ़ंक्शन रजिस्टर_माय_डैशबोर्ड_विजेट्स() {  
    wp_add_dashboard_widget('myInfo_widget', 'महत्वपूर्ण जानकारी', 'display_infoWidget');  
}
```

```
फ़ंक्शन डिस्प्ले_इन्फोविजेट() {  
    इको '<p>फरवरी के पहले महीने में इस साइट को एक नया डिज़ाइन मिलता है।  
    इसलिए इस दिन उपलब्ध नहीं होगा. वर्तमान प्रगति देखने के लिए आप <a href="http://www.justanexample.com" >इस साइट</a></p>' पर जा सकते हैं;  
}
```

अध्याय 11: अजाक्स

उदाहरण

JSON प्रतिक्रिया के साथ AJAX अनुरोध

फ़ंक्शन.php:

```
// हम कार्रवाई को दो बार जोड़ते हैं, एक बार लॉग इन उपयोगकर्ताओं के लिए और एक बार गैर लॉग इन उपयोगकर्ताओं के लिए। add_action('wp_ajax_my_action', 'my_action_callback');
add_action('wp_ajax_nopriv_my_action', 'my_action_callback');

// स्क्रिप्ट को सामने के सिरे पर पंक्तिबद्ध करें। add_action( 'wp_enqueue_scripts',
'enqueue_my_action_script' ); // स्क्रिप्ट को बैक एंड पर एनक्वू करें (wp-admin) add_action( 'admin_enqueue_scripts',
'enqueue_my_action_script' );

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

    यदि (isset( $_REQUEST['field2'] ) ) {
        $json['सदेश'] = 'सफलता!'; wp_send_json_success( $json );
    } अन्य {
        $json['message'] = 'फील्ड 2 सेट नहीं था!'; wp_send_json_error( $json );
    }
}

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

माय-एक्शन-स्क्रिप्ट.जेएस:

```
(फ़ंक्शन ($) 'सख्त उपयोग {
    करें';

    $( दस्तावेज़ ).पर( 'तैयार', फ़ंक्शन() {
        वर डेटा = {
            क्रिया: 'my_action', फ़ील्ड2: 'हैलो वर्ल्ड',
            फ़ील्ड3:3
        };

        $.getJSON(my_action_data.ajaxurl, डेटा, फ़ंक्शन(json) {
            यदि (json.success) चेतावनी ('हाँ!');
        } अन्य {
            चेतावनी(json.data.message);
        }
    });
});
```

```

    });
});

})(jQuery);

```

.ajax() और WordPress नॉन के साथ AJAX

फ़ंक्शन.php

```

//AJAX URL को स्थानीयकृत करें और नॉन्स
add_action('wp_enqueue_scripts', 'example_localize_ajax'); फ़ंक्शन example_localize_ajax()
{ wp_localize_script('jquery', 'ajax', array( 'url' =>
    admin_url('admin-ajax.php'), 'nonce' => wp_create_nonce('example_ajax_nonce'),

));
}

// उदाहरण AJAX फ़ंक्शन
add_action('wp_ajax_example_function', 'example_function'); add_action('wp_ajax_nopriv_example_function',
'example_function'); फ़ंक्शन example_function(){

    यदि ( !wp_verify_nonce($_POST['nonce'], 'example_ajax_nonce') ){
        मरो ('अनुमति अस्वीकृत. ');
    }

    $पहला नाम = sanitize_text_field($_POST['डेटा']['पहला नाम']); $अंतिमनाम = sanitize_text_field($_POST['डेटा']['अंतिमनाम']);

    // यहां डेटा के साथ कुछ करें echo $firstname । . $अंतिमनाम; //
    प्रतिक्रिया के लिए प्रतिध्वनि wp_die(); // इसे तुरंत समाप्त करना और उचित प्रतिक्रिया देना आवश्यक है:-

https://codex.wordpress.org/AJAX\_in\_Plugins }

```

example.js

```

jQuery(document).on('क्लिक टच टैप', '.example-selector', function(){
    jQuery.ajax({ प्रकार:
        "POST", url: ajax.url,
        डेटा: {

            नॉन्स: ajax.nonce, क्रिया:
            'example_function', डेटा: {

                पहला नाम: 'जॉन',
                अंतिम नाम: 'डो'
            },
        },
    },
    सफलता: फ़ंक्शन(प्रतिक्रिया){
        //सफलता
    },
    त्रुटि: फ़ंक्शन(XMLHttpRequest, textStatus, errorThrown){
        //गलती
    },
    टाइमआउट: 60000
});

```

```
विवरण झूठा है;  
});
```

wp_ajax - मुख्य कार्यक्षमता + _wpnonce जाँच

फ़ंक्शन.php:

```
फ़ंक्शन rm_init_js() {  
    wp_enqueue_script('कस्टम-ajax-स्क्रिप्ट', get_template_directory_uri() . '/js/ajax.js',  
सरणी ('jquery', 'wp-util'), '1.0', सत्य);  
    // JS में कस्टम वेरिएबल पास करें  
    wp_localize_script('कस्टम-ajax-स्क्रिप्ट', 'BEJS', array(  
        'एक्शन' => 'कस्टम_एक्शन',  
        'नॉनस' => wp_create_nonce( 'टेस्ट-नॉनस' )  
    ));  
}  
  
add_action('wp_enqueue_scripts', 'rm_init_js');  
  
फ़ंक्शन rm_ajax_handler() {  
    check_ajax_referer('test-nonce');  
  
    निकालें($_POST);  
    $ डेटा = कॉम्पैक्ट ( 'पहला_नाम', 'अंतिम_नाम', 'ईमेल' );  
  
    foreach ( $डेटा $नाम के रूप में => $मूल्य ) {  
        स्विच ( $नाम ) {  
            मामला 'पहला_नाम':  
            मामला 'अंतिम_नाम':  
                $डेटा[ $नाम ] = ucfirst( sanitize_user( $value ) );  
            तोड़ना;  
            मामला 'ईमेल':  
                $डेटा[ $नाम ] = sanitize_email( $मूल्य );  
            तोड़ना;  
        }  
    }  
  
    $userID = email_exists( $data['email'] );  
  
    यदि ! $उपयोगकर्ताआईडी {  
        ( wp_send_json_error( sprintf( __( 'कुछ गलत हो गया! %s पुनः प्रयास करें!', 'textdomain'  
) , $डेटा['first_name'] . ' ' . $डेटा['last_name'] ) );  
    }  
  
    wp_update_user( सरणी(  
        'आईडी' => $ उपयोगकर्ता आईडी,  
        'display_name' => $data['first_name']। 'first_name' => $data['first_name'], ' ' . $डेटा['अंतिम_नाम'],  
        'उपनाम' => $डेटा['last_name'],  
    ));  
  
    wp_send_json_success( sprintf( __( 'बैंक बैक %s', 'textdomain' ) , $data['first_name']  
    . ' ' . $डेटा['last_name'] ) );  
}  
  
add_action('wp_ajax_custom_action', 'rm_ajax_handler');  
add_action('wp_ajax_nopriv_custom_action', 'rm_ajax_handler');
```

Ajax.js

```
;(समारोह()) {  
    wp.ajax.post(BEJS.action, {  
        प्रथम_नाम: 'जॉन',  
        अंतिम_नाम: '%65doe',  
        ईमेल: 'john.doe@example.com',  
        _ajax_nonce: BEJS.nonce  
    });  
    .किया(फ़ंक्शन(प्रतिक्रिया) {  
        चेतावनी('सफलता: ${प्रतिक्रिया}');  
    });  
    .असफल(फ़ंक्शन(प्रतिक्रिया) {  
        चेतावनी('त्रुटि: ${प्रतिक्रिया}');  
    });  
});
```

नॉन के साथ एक साधारण वर्ग का उपयोग करके ओओपी अजाक्स सबमिशन

आप इसे आजमाने के लिए इस संपूर्ण प्लगइन को कॉपी और पेस्ट कर सकते हैं। यहां से वर्ग कंकाल का उपयोग किया जाता है।

क्लास-ऊप-ajax.cpp

```
<?php  
  
/**  
 * प्लगइन बूटस्ट्रैप फ़ाइल  
 *  
 * प्लगइन में प्लगइन जानकारी उत्पन्न करने के लिए इस फ़ाइल को वर्डप्रेस द्वारा पढ़ा जाता है  
 * डैशबोर्ड. यह फ़ाइल एक फ़ंक्शन को परिभाषित करती है जो प्लगइन शुरू करती है।  
 *  
 * @वर्डप्रेस-प्लगइन  
 * प्लगइन का नाम: उफ़ अजाक्स  
 * प्लगइन यूआरआई: एचटीटीपी://  
 * विवरण: फ़ॉर्म सबमिट करने के लिए OOP सिद्धांतों का उपयोग करने का एक सरल उदाहरण  
 फ्रंट एंड।  
 * संस्करण: 1.0.0  
 * लेखक: दिग्विजय नरूका  
 * लेखक यूआरआई: एचटीटीपी://  
 * लाइसेंस: जीपीएल-2.0+  
 * लाइसेंस यूआरआई: http://www.gnu.org/licenses/gpl-2.0.txt  
 * टेक्स्ट डोमेन: उफ़-अजाक्स  
 * डोमेन पथ: /भाषाएँ  
 */  
  
// यदि इस फ़ाइल को सीधे कॉल किया जाता है, तो निरस्त करें।  
यदि ( ! परिभाषित( 'WPINC' ) मरो; ) {  
  
}  
  
कक्षा Oop_Ajax {  
  
    // अपने सभी add_action, add_shortcode, add_filter फ़ंक्शन को __construct() में रखें  
    // कॉलबैक नाम के लिए, इसका उपयोग करें: array($this, '<function name>')  
    // <फ़ंक्शन नाम> इस वर्ग के भीतर फ़ंक्शन का नाम है, इसलिए वैश्विक स्तर पर होने की आवश्यकता नहीं है
```

अद्वितीय

```
// आमतौर पर उपयोग किए जाने वाले कुछ नमूना फ़ंक्शन नीचे शामिल किए गए हैं
सार्वजनिक फ़ंक्शन _निर्माण() {

    // फ्रंट-एंड डिस्क्ले के लिए जावास्क्रिप्ट और सीएसएस जोड़ें add_action('wp_enqueue_scripts',
    array($this,'enqueue'));

    // फ्रंट-एंड फॉर्म डिस्क्ले add_action('init', array($this, 'add_form_shortcode' ) ) के लिए
    शॉर्टकोड जोड़ें;
    // अजाक्स फ़ंक्शन जोड़ें जो लॉग इन उपयोगकर्ताओं के लिए कॉल वापस प्राप्त करेगा add_action( 'wp_ajax_my_action', array($this,
    'my_action_callback' ) );
    // अजाक्स फ़ंक्शन जोड़ें जो अतिथि या लॉग इन नहीं किए गए उपयोगकर्ताओं के लिए कॉल वापस प्राप्त करेगा add_action( 'wp_ajax_nopriv_my_action', array( $this,
    'my_action_callback' ) );

}

// यह फ्रंट पर उपयोग के लिए एक जावास्क्रिप्ट फ़ाइल और एक सीएसएस फ़ाइल को सूचीबद्ध करने का एक उदाहरण है
अंत प्रदर्शन सार्वजनिक
समारोह enqueue() {
    // वास्तविक कतारें, ध्यान दें कि फ़ाइलें जेएस और सीएसएस फ़ोल्डरों में हैं
    // स्क्रिप्ट के लिए, सुनिश्चित करें कि आप प्रासंगिक निर्भरताएं (इसमें jquery) शामिल कर रहे हैं
मामला)
    wp_enqueue_script('my-ajax-script', प्लगइन्स_url('js/oop-ajax.js', __FILE__), array('jquery'), '1.0', true);

    // कभी-कभी आप अपनी जावास्क्रिप्ट फ़ाइल में फ्रंट एंड पर डेटा तक पहुंच चाहते हैं // इसे प्राप्त करने के लिए इस कॉल की आवश्यकता होती है। हमेशा आगे बढ़ें और Ajaxurl शामिल करें। कोई और
चर,
    // सरणी में जोड़ें।
    // फिर जावास्क्रिप्ट फ़ाइल में, आप इसे इस तरह संदर्भित कर सकते हैं: my_php_variables.ajaxurl wp_localize_script('my-ajax-
script', 'my_php_variables', array(

        'ajaxurl' => admin_url('admin-ajax.php'), 'nonce' =>
        wp_create_nonce('_wpnonce')
    ));

}

/**
 * फॉर्म के लिए शॉर्टकोड पंजीकृत करता है। */
सार्वजनिक फ़ंक्शन add_form_shortcode() {

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

}

/**
 * शॉर्टकोड oop-ajax-add-form को प्रोसेस करता है
 *
 * @परम सरणी          $atts          शॉर्टकोड से विशेषताएँ
 *
 * @वापस करना        मिला हुआ          $आउटपुट          बफ़र का आउटपुट
 */
फ़ंक्शन add_form_front_end($atts, $content) {

    इको "<फॉर्म आईडी='माय_फॉर्म'>";

    इको "<label for='name'>नाम: </label>"; इको "<इनपुट आईडी='नाम' प्रकार='टेक्स्ट'
नाम='नाम' ";
```

```

    प्रतिध्वनि "<br>" ;

इको "<लेबल आईडी='ईमेल' फॉर='ईमेल'>ईमेल: </लेबल>" ; इको "<इनपुट प्रकार='टेक्स्ट' नाम='ईमेल'>";

    प्रतिध्वनि "<br>" ;

    प्रतिध्वनि "<इनपुट प्रकार='छिपा हुआ' नाम='क्रिया' मान='my_action' >" ; इको "<इनपुट आईडी='सबमिट_बीटीएन' टाइप='सबमिट' नाम='सबमिट'
    वैल्यू='सबमिट'>";

    प्रतिध्वनि "</form><br><br>"; इको "<div
    id='response'>ajax रिस्पॉन्स यहां होगा</div>";
}

/**
 * फॉर्म में प्रयुक्त my_action के लिए कॉलबैक फ़ंक्शन।
 *
 * फॉर्म से प्राप्त डेटा को संसाधित करता है, और आप इसके साथ जो चाहें कर सकते हैं।
 *
 * @वापस करना */      अजाक्स कॉल के पूरा होने के बारे में प्रतिध्वनि प्रतिक्रिया स्ट्रिंग।

फ़ंक्शन my_action_callback() { // echo wp_die('<pre>' .
    print_r($_REQUEST) . "<pre>");

    check_ajax_referer('wpnonce', 'security');

    यदि( ! खाली( $_POST )){

        यदि ( जारीकर्ता( $_POST['नाम'] )          ) {

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

        यदि(जारी($_POST['ईमेल']))          ) {

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

        //////////////////////////////////////////////////////////////////////////////////////////////////////////////////

        उदाहरण: डेटाबेस में सत्यापित करें और सहजें // प्रक्रिया और
        आउटपुट //////////////////////////////////////////////////////////////////////////////////////////////////////////////////

        $प्रतिक्रिया = "वाह <मजबूत शैली= 'रंग:लाल'>"। $नाम. " !</style></strong>आप
रेंक, आपने अभी-अभी अजाक्स को ऊप के साथ काम कराया है।";
        //यह डेटा को js फ़ंक्शन पर वापस भेज देगा: echo $response;

        } अन्य {

            इको "उह ओह! ऐसा लगता है कि मैंने आज खाना नहीं खाय।";

        }

        wp_die(); // कॉल को समाप्त करना आवश्यक है, अन्यथा वर्डप्रेस आरंभ करता है
समाप्ति और अंत में अजीब '0' आउटपुट।

    }

}

// हमारे प्लगइन को इनिशियलाइज़ करें

```

वैश्विक \$प्लगइन;

```
// पूरी चीज़ शुरू करने के लिए हमारी कक्षा का एक उदाहरण बनाएं $plugin = new Oop_Ajax();
```

oop-ajax.js

js फ़ाइल को js निर्देशिका यानी oop-ajax/js/oop-ajax.js के अंदर रखें

```
(फ़ंक्शन ($) {'सख्त उपयोग करें';

    $('#submit_btn').on('क्लिक', फ़ंक्शन() {
        // डेटा सेट करें
        var डेटा = {
            क्रिया: 'my_action', सुरक्षा:
            my_php_variables.nonce, नाम: $("#name").val(), ईमेल: $
            ("#email").val()

        }

        $.ajax({ प्रकार:
            'पोस्ट', url:
            my_php_variables.ajaxurl,
            डेटा: डेटा,
            सफलता: फ़ंक्शन(प्रतिक्रिया) { //सफलता पर प्रतिक्रिया आउटपुट करें $
                ("#response").html(प्रतिक्रिया);

            },
            त्रुटि: फ़ंक्शन(त्रुटि) {
                कंसोल.लॉग(त्रुटि);
            }
        });

        विवरण झूठा है;
    });
})(jQuery);
```

अध्याय 12: वैकल्पिक मुख्य लूप (pre_get_posts फ़िल्टर)

वाक्य - विन्यास

- `add_action('pre_get_posts', 'callback_function_name');` • फ़ंक्शन कॉलबैक_फ़ंक्शन_नाम (\$ क्वेरी) {} • // PHP 5.3.0 या इसके बाद के संस्करण के लिए
- `add_action('pre_get_posts', function($query){});`

पैरामीटर

पैरामीटर विवरण	
\$क्वेरी	(WP_Query) लूप ऑब्जेक्ट

टिप्पणियां

यदि आप PHP 5.3.0 या उससे ऊपर का उपयोग कर रहे हैं, तो आप क्लोजर (अनाम फ़ंक्शन) का उपयोग कर सकते हैं

```
add_action('pre_get_posts', फ़ंक्शन($query) {
    अगर( !$query->is_main_query() || is_admin() ) वापसी;

    // यह कोड तभी चलेगा जब // - यह क्वेरी मुख्य क्वेरी है // - और
    यह एडमिन स्क्रीन नहीं है

});
```

उदाहरण

और भी अधिक विशिष्ट लूप लक्ष्यीकरण

मान लीजिए कि हम मुख्य लूप बदलना चाहते हैं, [केवल विशिष्ट वर्गीकरण, या पोस्ट प्रकार के लिए](#)।

पुस्तक पोस्ट प्रकार संग्रह पृष्ठ पर केवल मुख्य लूप को लक्षित करना ।

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

फ़ंक्शन my_callback_function ($ क्वेरी ) {
    अगर( !$query->is_main_query() || is_admin() ) वापसी; यदि( !is_post_type_archive( 'पुस्तक' ) )
    वापसी;

    // यह कोड तभी चलेगा जब // - यह क्वेरी मुख्य क्वेरी है

    // - और यह एडमिन स्क्रीन नहीं है
```

```
// - और हम 'पुस्तक' पोस्ट प्रकार संग्रह पृष्ठ पर हैं  
}
```

आप `is_category()`, `is_tag()` और `is_tax()` का उपयोग करके श्रेणी, टैग या कस्टम वर्गीकरण संग्रह पृष्ठ की भी जांच कर सकते हैं।

आप वर्डप्रेस में उपलब्ध किसी भी सशर्त टैग का उपयोग कर सकते हैं।

केवल एक श्रेणी से पोस्ट दिखाएँ

```
add_action('pre_get_posts', 'single_category');  
  
फ़ंक्शन सिंगल_कैटेगरी( $query ) if( !$query->is_main_query() {  
    || is_admin()) return;  
  
    $query->सेट('बिल्ली', '1');  
    वापस करना;  
}
```

पोस्ट प्राप्त करने से पहले बुनियादी उपयोग को फ़िल्टर करें

कभी-कभी आप मुख्य वर्डप्रेस क्वेरी को बदलना चाहेंगे।

`प्री_गेट_पोस्ट` फ़िल्टर करना ही रास्ता है।

उदाहरण के लिए `pre_get_posts` का उपयोग करके आप मुख्य लूप बता सकते हैं केवल 5 पोस्ट दिखाने के लिए. [या फिर केवल एक कैटेगरी से पोस्ट दिखाना](#), या किसी कैटेगरी को बाहर करना आदि।

```
add_action('pre_get_posts', 'my_callback_function');  
  
फ़ंक्शन my_callback_function( $query ) // यहां आपके फ़िल्टर का तर्क दिया {  
    गया है  
}
```

जैसा कि आप देख सकते हैं, हम मुख्य लूप पास कर रहे हैं हमारे कॉलबैक फ़ंक्शन [तर्क में क्वेरी ऑब्जेक्ट](#)।

यहां महत्वपूर्ण नोट: हम तर्क को संदर्भ के रूप में पारित कर रहे हैं। इसका मतलब है कि इसे काम करने के लिए हमें क्वेरी वापस करने या कोई ग्लोबल सेट करने की आवश्यकता नहीं है। चूंकि `$query` मुख्य क्वेरी ऑब्जेक्ट का संदर्भ है, हम अपने ऑब्जेक्ट पर जो भी परिवर्तन करते हैं वे तुरंत मुख्य लूप ऑब्जेक्ट में परिलक्षित होते हैं।

पोस्ट सूची से श्रेणी को बाहर निकालें, शेयर संपादित करें

```
add_action('pre_get_posts', 'single_category_exclude');  
  
फ़ंक्शन एकल_श्रेणी_बहिष्कृत( $क्वेरी ) {  
    अगर( !$क्वेरी->is_main_query() || is_admin() ) वापसी;  
  
    $क्वेरी->सेट('बिल्ली', '-1'); वापस करना;  
}
```

मुख्य लूप के लिए पोस्ट_पर_पेज बदलें

हमें बस \$query ऑब्जेक्ट की set() विधि का उपयोग करना है।

इसमें दो तर्क लगते हैं, पहला हम क्या सेट करना चाहते हैं और दूसरा कौन सा मान सेट करना है।

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

फ़ंक्शन परिवर्तन_पोस्ट_पर_पेज( $query ) if( !$query->is_main_query() ||
    is_admin() ) return;

    $query->सेट('पोस्ट_पर_पेज', 5);
    वापस करना;
}
```

केवल मुख्य वर्डप्रेस लूप को लक्षित करना

वर्डप्रेस अपने द्वारा उत्पन्न किसी भी लूप पर pre_get_posts फ़िल्टर लागू कर रहा है। इसका मतलब यह है कि हम अपने कॉलबैक फ़ंक्शन में जो भी बदलाव कर रहे हैं, वे सभी निकास लूपों पर लागू होते हैं।

जाहिर तौर पर यह वह नहीं है जो हम ज्यादातर परिदृश्यों में चाहते हैं।

अधिकांश मामलों में हम केवल मुख्य लूप को लक्षित करना चाहेंगे , और केवल गैर-व्यवस्थापक स्क्रीन के लिए।

यह जांचने के लिए कि हम सही जगह पर हैं या नहीं, हम is_main_query() विधि और is_admin() वैश्विक फ़ंक्शन का उपयोग कर सकते हैं।

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

फ़ंक्शन my_callback_function ( $ क्वेरी )
    अगर( !$query->is_main_query() || is_admin() ) वापसी;

    // यह कोड तभी चलेगा जब // - यह क्वेरी मुख्य क्वेरी है // -
    और यह एडमिन स्क्रीन नहीं है
}
```

अध्याय 13: बाल विषय की मूल बातें

वाक्य - विन्यास

- टेम्पलेट - मुख्य वर्डप्रेस थीम, पैरेंट का नाम है। • चाइल्ड-थीम --वह पैकेज है जो टेम्पलेट को ओवरराइड करता है।

टिप्पणियां

मैं विज्ञापन देता रहा हूँ कि चाइल्ड थीम का उपयोग हमेशा एक अच्छी बात है लेकिन इसमें हमेशा एक परंतु होता है

...

हमारे टेम्पलेट ओवरराइटिंग उदाहरण में आइए कल्पना करें कि किसी थीम का लेखक साइडबार टेम्पलेट में अपना सुधार जोड़ रहा है और वहां एक नया सुधार होगा

/themes/template/sidebar.php

```
<?php /
** *
मुख्य विजेट क्षेत्र वाले साइडबार के लिए टेम्पलेट
*
* @लिंक https://developer.wordpress.org/themes/basics/template-files/#template-partials */

यदि ( is_active_sidebar( 'साइडबार-1' ) ) :?>
<एक तरफ आईडी = "माध्यमिक" वर्ग = "साइडबार विजेट-क्षेत्र" भूमिका = "पूरक">
<?php डायनामिक साइडबार('साइडबार-1'); ?>
</एक तरफ><!-- .साइडबार .विजेट-क्षेत्र --> <?php एंडिफ; ?>
```

अब हमारी वेबसाइट को नई भूमिका = "पूरक" विशिष्टता से लाभ नहीं होगा क्योंकि हमारी चाइल्ड थीम अभी भी /themes/child-theme/sidebar.php पर अपनी फ़ाइल के साथ टेम्पलेट को ओवरराइड कर रही है।

वेबसाइट अनुरक्षक के रूप में यह हमारा कर्तव्य है कि हम इस बात पर नज़र रखें कि हम किन टेम्पलेट्स को ओवरराइड करते हैं और, अपडेट के आसन्न मामले में, चेंजलॉग को ध्यान से देखें ताकि यदि आवश्यक हो तो आप चाइल्ड थीम फ़ाइलों को अपडेट कर सकें।

उदाहरण

2) उद्देश्य

चाइल्ड थीम को थीम अपडेट पर खोने के डर के बिना मुख्य टेम्पलेट के अनुकूलन को बनाए रखने का एक सुरक्षित तरीका माना जाता है।

मूल रूप से, जब भी आप अपनी वेबसाइट से सक्रिय टेम्पलेट के अंदर किसी फ़ाइल को संपादित करना चाहते हैं तो आपको खुद से पूछना होगा "जब मैं थीम अपडेट करूंगा तो क्या होगा?"

और उत्तर सरल है: आप उन्हें खो देंगे क्योंकि अपडेट एक पूरी तरह से नई थीम लाएगा

फ़ोल्डर.

तो आइए चाइल्ड थीम को फ़ाइलों वाले एक फ़ोल्डर के रूप में देखें जो मूल थीम में समान पथ वाली फ़ाइलों को अधिलेखित कर देगा। आइए अब कुछ वास्तविक उदाहरण लाते हैं:

परिभाषा और आवश्यकताएँ

वर्डप्रेस द्वारा एक चाइल्ड थीम की पहचान तब की जाती है जब निम्नलिखित फ़ाइलों के साथ `/wp-content/themes/` के अंदर एक निर्देशिका (उदाहरण के लिए चाइल्ड-थीम) होती है:

- स्टाइल.सीएसएस

यह फ़ाइल इस प्रकार के टिप्पणी टेम्पलेट से प्रारंभ होनी चाहिए:

```
/*
थीम का नाम: उदाहरण बच्चा
लेखक: मैं
लेखक यूआरआई: https://example.com/ टेम्पलेट: उदाहरण टेक्स्ट
डोमेन: उदाहरण-बाल-थीम डोमेन पथ: /
भाषाएँ/ */
```

यहां विचार करने योग्य सबसे महत्वपूर्ण बातें हैं:

- टेम्पलेट का नाम बिल्कुल उस फ़ोल्डर का नाम होना चाहिए जिसमें मूल थीम (जिसे मूल थीम स्लग भी कहा जाता है) होती है
- अपने बच्चे की थीम को इस तरह नाम दें कि आप उसे डैशबोर्ड में आसानी से पहचान सकें (आमतौर पर केवल माता-पिता के नाम के साथ चाइल्ड जोड़ें, जैसे उदाहरण चाइल्ड)

• `Index.php`

• `फ़ंक्शन.php`

3) टेम्पलेट ओवरराइटिंग

चाइल्ड थीम का सबसे आम उपयोग टेम्पलेट भागों को ओवरराइट करना है। उदाहरण के लिए, एक साइडबार, यदि हमारे पास निम्नलिखित फ़ाइल के साथ एक थीम है

`/themes/template/sidebar.php`

```
<?php /
**
* मुख्य विजेट क्षेत्र वाला साइडबार।
*
* @लिंक https://developer.wordpress.org/themes/basics/template-files/#template-partials */

अगर ( ! is_active_sidebar('साइडबार-1') ) {
    वापस करना;
}??>
<div id='साइडबार' वर्ग='विजेट-क्षेत्र'>
    <?php डायनामिक_साइडबार('साइडबार-1'); ?>
```

```
</div>
```

हम निश्चित रूप से निम्नलिखित फ़ाइल के साथ चाइल्ड थीम में अपनी स्वयं की फ़ाइल (पहले उदाहरण से विशिष्टताओं के साथ) जोड़ सकते हैं

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

```
<?php /
** *
मुख्य विजेट क्षेत्र वाला साइडबार। */

अगर ( ! is_active_sidebar('साइडबार-1') ) {
    वापस करना;
}?>
<div id='my-sidebar' class='my-own-awesome-class Widget-area'> <div class='my-wrapper'> <?
    phpdynamic_sidebar( 'sidebar-1' ); ?> </
    div> </div>
```

अब मेरी-अपनी-विस्मयकारी-क्लास चाइल्ड थीम में सुरक्षित है और इसे किसी भी थीम अपडेट पर हटाया नहीं जाएगा और वर्डप्रेस हमेशा चाइल्ड थीम में से एक टेम्प्लेट चुनेगा जब उसे उसी पथ पर कोई मिलता है।

संपत्ति प्रतिस्थापन

भले ही यह सर्वोत्तम अभ्यास न हो, कभी-कभी आपको सीएसएस या जेएस फ़ाइलों या लाइब्रेरी जैसी संपत्तियों को बदलने की आवश्यकता होती है।

ध्यान दें कि वर्डप्रेस टेम्प्लेट ओवरराइटिंग सिस्टम .php फ़ाइलों के अलावा किसी अन्य चीज़ के साथ काम नहीं करता है , इसलिए जब हम संपत्तियों के बारे में बात करते हैं तो हम पंजीकृत का उल्लेख करते हैं संपत्ति

एक उदाहरण आपके इच्छित संस्करण के साथ jQuery लाइब्रेरी का प्रतिस्थापन हो सकता है। हमारी चाइल्ड थीम फ़ंक्शंस.php फ़ाइल में हमें एक फ़ंक्शन जोड़ने की ज़रूरत है जो वर्तमान jQuery संस्करण को हटा देती है और CDN से अपना स्वयं का संस्करण जोड़ती है (याद रखें कि यह केवल एक उदाहरण है)।

```
/**
 * jQuery स्क्रिप्ट को हटाएं और अपना स्वयं का संस्करण जोड़ें।
 *
 * देर से प्राथमिकता (100) के साथ, wp_print_scripts क्रिया से जुड़ा हुआ,
 * ताकि यह स्क्रिप्ट के कतारबद्ध होने के बाद हो।
 */
फ़ंक्शन my_own_theme_scripts() {
    // वर्तमान संस्करण को हटा दें
    wp_dequeue_script('jquery'); // मेरा वांछित संस्करण पंजीकृत
    करें wp_register_script('jquery', 'https://code.jquery.com/
    jquery-3.1.0.min.js', false,
    '3.1.0');
    // मेरा संस्करण यहां या कहीं और लोड करें wp_enqueue_script('jquery');
}
add_action('wp_print_scripts', 'my_own_theme_scripts');
```


अध्याय 14: प्रोग्रामेटिक रूप से एक पोस्ट बनाएं

वाक्य - विन्यास

- `wp_insert_post(सरणी $args, बूल $wp_error);`

पैरामीटर

पैरामीटर	विवरण
<code>\$args</code> (सरणी आवश्यक)	नीचे दिए गए तत्वों की एक मुख्य मान सारणी।
<code>\$wp_error</code> (बूलियन वैकल्पिक)	विफलता की स्थिति में <code>WP_Error</code> लौटाएँ।

टिप्पणियां

बहस

अगली तालिका आपको उन तत्वों की एक सूची दिखाती है जिनका उपयोग आप पहले पैरामीटर (एरे) के अंदर कर सकते हैं।

पैरामीटर	विवरण
<code>पहचान</code>	(इंट) पोस्ट आईडी. यदि 0 के अलावा किसी अन्य चीज़ के बराबर है, तो उस आईडी वाली पोस्ट अपडेट कर दी जाएगी. डिफ़ॉल्ट 0.
<code>पोस्ट_लेखक</code>	(इंट) पोस्ट जोड़ने वाले उपयोगकर्ता की आईडी। डिफ़ॉल्ट है वर्तमान उपयोगकर्ता आईडी.
<code>पोस्ट करने की तारीख</code>	(स्ट्रिंग) पोस्ट की तारीख. डिफ़ॉल्ट वर्तमान समय है.
<code>पोस्ट_डेट_जीएमटी</code>	(स्ट्रिंग) जीएमटी समयक्षेत्र में पोस्ट की तारीख। गलती करना <code>\$post_date</code> का मान है.
<code>पोस्ट सामग्री</code>	(मिश्रित) पोस्ट सामग्री. डिफ़ॉल्ट खाली.
<code>पोस्ट_सामग्री_फ़िल्टर किया गया</code>	(स्ट्रिंग) फ़िल्टर की गई पोस्ट सामग्री। डिफ़ॉल्ट खाली.
<code>शीर्षक पोस्ट करें</code>	(स्ट्रिंग) पोस्ट शीर्षक. डिफ़ॉल्ट खाली.
<code>पोस्ट_श्रेणी</code>	(सरणी) पोस्ट श्रेणी मानों की सारणी।
<code>पोस्ट_अंश (स्ट्रिंग) पोस्ट अंश.</code>	डिफ़ॉल्ट खाली.

पैरामीटर	विवरण
पोस्ट_स्थिति	(स्ट्रिंग) पोस्ट स्थिति. डिफ़ॉल्ट ड्राफ्ट.
पद प्रकार	(स्ट्रिंग) पोस्ट प्रकार. डिफ़ॉल्ट पोस्ट.
टिप्पणी_स्थिति	(स्ट्रिंग) क्या पोस्ट टिप्पणियाँ स्वीकार कर सकती है। स्वीकार खुला या बंद. डिफ़ॉल्ट का मान है default_comment_status विकल्प।
ping_status	(स्ट्रिंग) क्या पोस्ट पिंग स्वीकार कर सकती है। खुला स्वीकार करता है या बंद. डिफ़ॉल्ट default_ping_status का मान है विकल्प।
पोस्ट_पासवर्ड	(स्ट्रिंग) पोस्ट तक पहुंचने का पासवर्ड। डिफ़ॉल्ट खाली.
पोस्ट नाम	(स्ट्रिंग) पोस्ट नाम या स्लग। डिफ़ॉल्ट स्वच्छता है नई पोस्ट बनाते समय पोस्ट शीर्षक.
पिंग करने के लिए	(स्ट्रिंग) यूआरएल की स्पेस या कैरिज रिटर्न-पृथक सूची गुनगुनाहट। डिफ़ॉल्ट खाली.
पिंग किया	(स्ट्रिंग) यूआरएल की स्पेस या कैरिज रिटर्न-पृथक सूची जिसे पिंग कर दिया गया है. डिफ़ॉल्ट खाली.
पोस्ट_संशोधित	(स्ट्रिंग) वह तारीख जब पोस्ट को अंतिम बार संशोधित किया गया था। गलती करना वर्तमान समय है.
पोस्ट_संशोधित_जीएमटी	(स्ट्रिंग) वह तारीख जब पोस्ट को आखिरी बार संशोधित किया गया था जीएमटी समय क्षेत्र. डिफ़ॉल्ट वर्तमान समय है.
पोस्ट_पैरेंट	(इंट) इसे उस पोस्ट के लिए सेट करें जिससे यह संबंधित है, यदि कोई हो। डिफ़ॉल्ट 0.
मेनू_आदेश	(इंट) वह क्रम जिसमें पोस्ट प्रदर्शित होनी चाहिए। डिफ़ॉल्ट 0।
पोस्ट_माइम_प्रकार	(स्ट्रिंग) पोस्ट का माइम प्रकार. डिफ़ॉल्ट खाली.
GUID	(स्ट्रिंग) पोस्ट को संदर्भित करने के लिए ग्लोबल यूनिक आईडी। गलती करना खाली।
टैक्स_इनपुट	(सरणी) वर्गीकरण शब्दों की सारणी, जो उनके वर्गीकरण द्वारा कुंजीबद्ध है नाम। डिफ़ॉल्ट खाली.
मेटा_इनपुट	(सरणी) पोस्ट मेटा मानों की सारणी उनके पोस्ट मेटा द्वारा कुंजीबद्ध चाबी। डिफ़ॉल्ट खाली.

डुप्लीकेट पोस्ट से बचें

जब आप इस फ़ंक्शन को निष्पादित करते हैं, तो संभवतः आपको एक डुप्लिकेट पोस्ट प्राप्त हो सकती है, कम से कम मेरे साथ तो ऐसा हुआ है। (आप इसे पोस्ट वर्डप्रेस अनुभाग में देख सकते हैं)

मुझे एक समाधान मिला:

```
यदि(!  
  
    'post_content' => 'सामग्री', 'tags_input' =>  
    $tags, 'post_category' => array(2),  
    'post_status' => 'प्रकाशित करें'  
  
);  
  
$परिणाम = wp_insert_post($my_post);  
}
```

व्याख्या

किसी नई पोस्ट को सहेजने से पहले, पैरामीटर के रूप में पोस्ट शीर्षक का उपयोग करके सत्यापित करें कि क्या नई पोस्ट पहले से मौजूद है, यदि कोई पोस्ट शीर्षक नहीं है, तो आप अपनी नई पोस्ट को सहेज सकते हैं।

यहां [get_page_by_title](#) का दस्तावेज़ देखें ।

उदाहरण

परिचय

कभी-कभी हमारे पास TinyMCE (वर्डप्रेस डिफॉल्ट एडिटर) के बजाय कहीं और एक और संपादक होता है। ऐसा तब होता है जब हम अपनी खुद की थीम, प्लगइन या कुछ विशिष्ट बना रहे होते हैं; और हमें एक प्रकार की पोस्ट लिखने और हेरफेर करने और इसे अपने WP डेटाबेस में सहेजने की आवश्यकता है।

इसलिए, यदि आप उस स्थिति में हैं, तो आप एक वर्डप्रेस फ़ंक्शन का उपयोग कर सकते हैं जिसे कहा जाता है:

```
wp_insert_post(सरणी $args, बूल $wp_error);
```

एक बेसिक पोस्ट बनाएं

```
$basic_post_args = array( 'post_title' =>  
    'मेरा मूल पोस्ट', 'post_content' => 'यह एक बुनियादी सामग्री है',  
    'post_status' => 'प्रकाशित करें', 'post_author' => 1, 'post_category' =>  
    सरणी(8,59)  
  
);  
  
wp_insert_post($basic_post_args);
```

एक बेसिक पेज बनाएं

```
$basic_page_args = array( 'post_title' =>
    'मेरा मूल पृष्ठ', 'post_content' => 'यह एक मूल सामग्री है',
    'post_type' => 'पेज', 'post_status' => 'प्रकाशित करें', 'post_author' =>1

);

wp_insert_post($basic_page_args);
```

अध्याय 15: कस्टम पोस्ट के लिए टेम्पलेट बनाएं

प्रकार

उदाहरण

कस्टम पोस्ट प्रकार की पुस्तक के लिए एक कस्टम टेम्पलेट बनाना

हमारे कस्टम पोस्ट प्रकार के एकल पोस्ट के लिए एक टेम्पलेट बनाने के लिए, हमें सिंगल-पोस्ट_टाइप_नाम.php नामक एक फ़ाइल बनाने की आवश्यकता है जहां पोस्ट_टाइप_नाम हमारे कस्टम पोस्ट प्रकार का नाम है।

उदाहरण के लिए, यदि हमारे कस्टम पोस्ट प्रकार को "पुस्तकें" कहा जाता है, तो हमें सिंगल बुक.php नामक एक PHP फ़ाइल बनाने की आवश्यकता है। ध्यान दें कि हमने अपने कस्टम पोस्ट प्रकार के एकवचन नाम का उपयोग किया है।

थीम फ़ोल्डर से सिंगल.php की सामग्री को कॉपी करें और इसे नए टेम्पलेट में पेस्ट करें और इसे सेव करें, फिर टेम्पलेट को कस्टम पोस्ट प्रकार के व्यक्तिगत पेज के लिए लागू किया जाएगा।

कस्टम पोस्ट प्रकार टेम्पलेट्स

कस्टम पोस्ट प्रकार पुरालेख:

कस्टम पोस्ट प्रकार के लिए एक संग्रह टेम्पलेट बनाने के लिए आपको अपने रजिस्टर_पोस्ट_टाइप() में has_archive तर्क को सत्य के बराबर सेट करना होगा। समारोह। नीचे दिए गए उदाहरण में एक इवेंट पोस्ट प्रकार के लिए एक कस्टम पोस्ट प्रकार बनाया गया है।

```
add_action('init', 'create_events_post_type'); फ़ंक्शन create_events_post_type()
{ रजिस्टर_पोस्ट_टाइप( 'इवेंट',

    सारणी(
        'लेबल' => सारणी(
            'नाम' => __( 'घटनाएँ' ), 'एकवचन_नाम' =>
                __( 'घटना' )
        ),
        'सार्वजनिक' => सत्य,
        'has_archive' => सत्य,
    )
);
}
```

एक टेम्पलेट बनाने के लिए नए कस्टम पोस्ट प्रकारों के लिए आपको एक नई टेम्पलेट फ़ाइल बनानी होगी। एकल पोस्ट पेजों के लिए एक टेम्पलेट बनाने के लिए आप इसे सिंगल-{पोस्ट_टाइप}.php और आर्काइव के लिए आर्काइव- {पोस्ट_टाइप}.php नाम देंगे।

हमारे संग्रह टेम्पलेट का फ़ाइल नाम Archive-event.php होगा और इवेंट पृष्ठ के लिए यह सिंगल-इवेंट.php होगा। दोनों फ़ाइलें आपकी थीम रूट डायरेक्टरी में होनी चाहिए।

एक उदाहरण संग्रह टेम्पलेट इस तरह दिखेगा. इक्कीसवीं थीम से लिया गया ।

```
<?php /
** *
संग्रह पृष्ठों को प्रदर्शित करने के लिए टेम्पलेट
*
* @लिंक https://codex.wordpress.org/Template_Hierarchy
*
* @पैकेज वर्डप्रेस * @सबपैकेज ट्वेंटी_सेवेंटीन
* @1.0 से * @संस्करण 1.0 */

get_header(); ?>

<div class='रैप'>

    <?php if ( has_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='प्राथमिक' वर्ग='सामग्री-क्षेत्र'>
        <मुख्य आईडी='मुख्य' वर्ग='साइट-मुख्य' भूमिका='मुख्य'>

        <?php
        यदि ( has_posts() ) <?php /* लूप : ?>
            प्रारंभ करें
            * / जबकि ( has_posts() )
                : पोस्ट();

            /*
            * सामग्री के लिए पोस्ट-प्रारूप-विशिष्ट टेम्पलेट शामिल करें।
            * यदि आप इसे चाइल्ड थीम में ओवरराइड करना चाहते हैं, तो एक फ़ाइल शामिल करें * जिसे content-___.php कहा जाता है (जहां
            ___ पोस्ट फ़ॉर्मेट नाम है) और वह होगा

इसके स्थान पर उपयोग किया जाता है।

            * /
            get_template_part('टेम्पलेट-पार्स/पोस्ट/कंटेंट', get_post_format());

            अंत तक;

            the_posts_pagination( सारणी(
                'prev_text' => इक्कीसवीं_get_svg (सरणी ('आइकन' => 'तीर-बाएँ')
                    ) .
                '<span class='स्क्रीन-रीडर-टेक्स्ट'>'। __('पिछला पृष्ठ', 'सत्ताईसवें')
                    . ', 'next_text' =>
                '<span class='screen-reader-text'>' . __('अगला पृष्ठ',
                    . '' . ट्वेंटीसेवेंटीन_गेट_एसवीजी (सरणी ('आइकन' => 'तीर-दाएं')),
                'before_page_number' => '<span class='meta-nav screen-reader-text'>' . __(
                    'पेज', 'ट्वेंटीसेवेंटीन' )
                    . ' ',
                ) );

            अन्य :

            get_template_part('टेम्पलेट-पार्स/पोस्ट/कंटेंट', 'कोई नहीं');

            अगर अंत; ?>
```

```

</मुख्य><!-- #मुख्य -->
</div><!-- #प्राथमिक -->
<?php get_sidebar(); ?>
</div><!-- .wrap -->

<?php get_footer();

```

कस्टम पोस्ट प्रकार एकल टेम्पलेट:

यहां एकल टेम्पलेट का एक उदाहरण दिया गया है. इक्कीसवीं थीम से लिया गया ।

```

<?php
/**
 * सभी एकल पोस्ट प्रदर्शित करने के लिए टेम्पलेट
 *
 * @लिंक https://developer.wordpress.org/themes/basics/template-hierarchy/#single-post
 *
 * @पैकेज वर्डप्रेस
 * @सबपैकेज ट्वेंटी_सेवेंटीन
 * @1.0 से
 * @संस्करण 1.0
 */

get_header(); ?>

<div class='रैप'>
    <div id='प्राथमिक' वर्ग='सामग्री-क्षेत्र'>
        <मुख्य आईडी='मुख्य' वर्ग='साइट-मुख्य' भूमिका='मुख्य'>

            <?php
                /* लूप प्रारंभ करें */
                जबकि (have_posts()) : पोस्ट();

                    get_template_part('टेम्पलेट-पार्ट्स/पोस्ट/कंटेंट', get_post_format());

                    // यदि टिप्पणियाँ खुली हैं या हमारे पास कम से कम एक टिप्पणी है, तो इसे लोड करें
                    टिप्पणी टेम्पलेट.

                    यदि ( comments_open() || get_comments_number() ) comments_template();

                    अगर अंत;

                    the_post_नेविगेशन(सरणी(
                        'prev_text' => '<span class='स्क्रीन-रीडर-टेक्स्ट'>'। __('पहले का
                        पोस्ट', 'ट्वेंटीसेवेंटीन' )
                        . '<span aria-hidden='true' class='nav-subtitle'>' . __(
                        'पिछला', 'ट्वेंटीसेवेंटीन' ) आइकन-रैपर">'। ट्वेंटीसेवेंटीन_गेट_एसवीजी( एरे . ' <span class='nav-title'><span class='nav-title
                        ('आइकन' => 'एरो-लेफ्ट' ) '%title',
                        )
                        . ' ',

                        'next_text' => '<span class='screen-reader-text'>' . __('अगली पोस्ट',
                        'ट्वेंटीसेवेंटीन' ) 'ट्वेंटीसेवेंटीन' ) रैपर">' .
                        . '<span aria-hidden='true' class='nav-subtitle'>' . __( 'अगला',
                        ट्वेंटीसेवेंटीन_गेट_एसवीजी (सरणी
                        . ' <span class='nav-title'>%title<span class='nav-title-icon'
                        ('आइकन' => 'एरो-राइट')
                        . ' ',

                    ));

                    अंत तक; // लूप का अंत।
                ?>

            </मुख्य><!-- #मुख्य -->

```

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

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

दोनों टेम्पलेट उदाहरण आंशिक रूप से खींच रहे हैं आंतरिक सामग्री प्रदर्शित करने के लिए.

यदि आपके बच्चे/अभिभावक थीम में एकल/संग्रह टेम्पलेट है तो आपको उस कोड को अपने नए टेम्पलेट के लिए बॉयलरप्लेट के रूप में उपयोग करना चाहिए।

अध्याय 16: एक कस्टम टेम्पलेट बनाना

उदाहरण

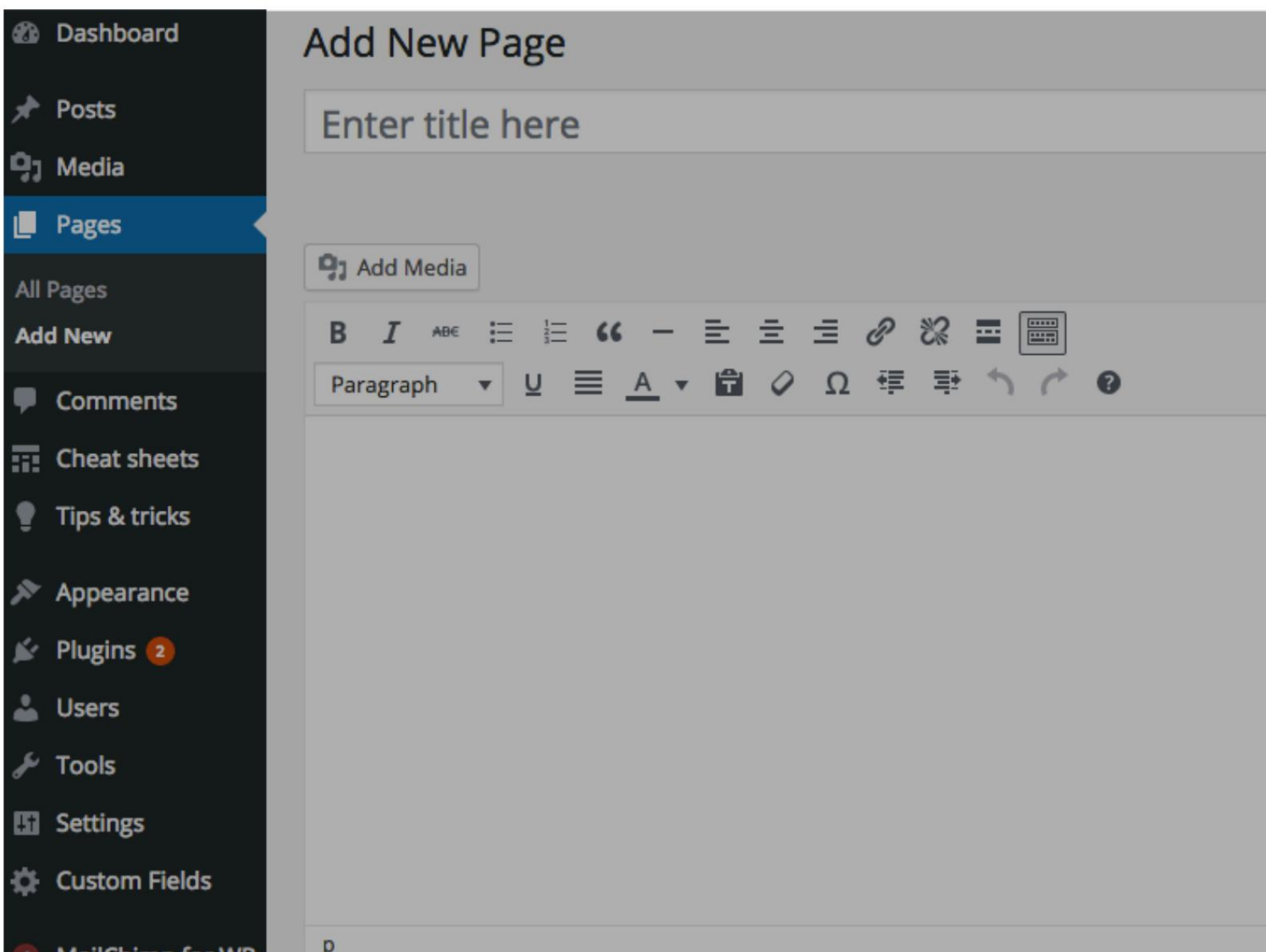
बुनियादी रिक्त टेम्पलेट बनाना

एक कस्टम टेम्पलेट बनाने के लिए हमें सबसे पहले थीम डायरेक्टरी में php फ़ाइल बनानी होगी। आप इसे लगभग अपनी इच्छानुसार कोई भी नाम दे सकते हैं। इस उदाहरण के लिए हम example.php बनाएंगे

एक और एकमात्र चीज जिसे हमें अपने example.php के अंदर परिभाषित करने की आवश्यकता है, वर्डप्रेस द्वारा एक टेम्पलेट के रूप में पहचाने जाने के लिए, वह टेम्पलेट नाम है। हम ऐसा करते हैं कि फ़ाइल के शीर्ष पर विशेष टिप्पणी डालते हैं, जैसे:

```
<?php /
*
टेम्पलेट नाम: उदाहरण */
?>
```

और अब जब हमें अपना टेम्पलेट पेज एट्रीब्यूट्स बॉक्स में टेम्पलेट ड्रॉपडाउन में सूचीबद्ध देखना चाहिए



हमारे टेम्पलेट में शीर्ष लेख और पाद लेख शामिल करना

आइए ऊपर से अपने टेम्पलेट का विस्तार करें और हेडर.php और footer.php से सामग्री शामिल करें

हेडर सहित:

हम टेम्पलेट नाम टिप्पणी के ठीक बाद हेडर शामिल करेंगे

ऐसा करने के दो सामान्य तरीके हैं। दोनों सही हैं और एक जैसे काम करते हैं, यह सिर्फ आपकी शैली और कोड कैसा दिखता है इसके बारे में है

पहला तरीका:

```
<?php
/*
टेम्पलेट का नाम: उदाहरण
*/
get_header();
?>
```

दूसरा तरीका:

```
<?php
/*
टेम्पलेट का नाम: उदाहरण
*/
?>
<?php get_header(); ?>
```

पाद लेख सहित:

पादलेख शामिल करना उसी तरह से काम करता है, केवल एक चीज है जिस पर हमें ध्यान देने की आवश्यकता है, और वह यह है कि हम शीर्षलेख शामिल करने के बाद पादलेख शामिल करते हैं। तो अंतिम टेम्पलेट कुछ इस तरह दिखना चाहिए।

```
<?php
/*
टेम्पलेट का नाम: उदाहरण
*/
get_header();
?>

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

सामग्री के साथ कस्टम टेम्पलेट

हम अपने टेम्पलेट को आगे बढ़ाएंगे और पेज का शीर्षक और एक सामग्री शामिल करेंगे

```
<?php /
*
टेम्पलेट का नाम: उदाहरण */
```

```
get_header();
```

```
शीर्षक();
```

```
यो विषय वस्तु();
```

```
get_footer();
```

और अगर आप चाहें तो इन्हें इस तरह HTML तत्वों के साथ लपेट सकते हैं

```
<?php /
```

```
*
```

```
टेम्पलेट का नाम: उदाहरण */
```

```
get_header();
```

```
प्रतिध्वनि '<h1>' . शीर्षक() | '</h1>'; प्रतिध्वनि '<अनुभाग>' | 'यो विषय
```

```
वस्तु() | '</अनुभाग>';
```

```
get_footer();
```

या यदि आप इको का उपयोग किए बिना सामान्य HTML फ़ाइल की तरह काम करना पसंद करते हैं

```
<?php /
```

```
*
```

```
टेम्पलेट का नाम: उदाहरण */
```

```
get_header();
```

```
?>
```

```
<h1><?php the_title(); ?></h1> <अनुभाग><?php
```

```
the_content(); ?></अनुभाग>
```

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

अध्याय 17: अंश_लंबाई और अंश_अधिक के साथ कस्टम अंश

उदाहरण

अंश की लंबाई 50 शब्दों तक सीमित करें

निम्नलिखित कोड को function.php में डालें:

```
फ़ंक्शन थीमिफ़ाई_कस्टम_एक्ससर्प्ट_लेंथ ($ लंबाई ) रिटर्न 50;
{

} add_filter('excerpt_length', 'themify_custom_excerpt_length', 999);
```

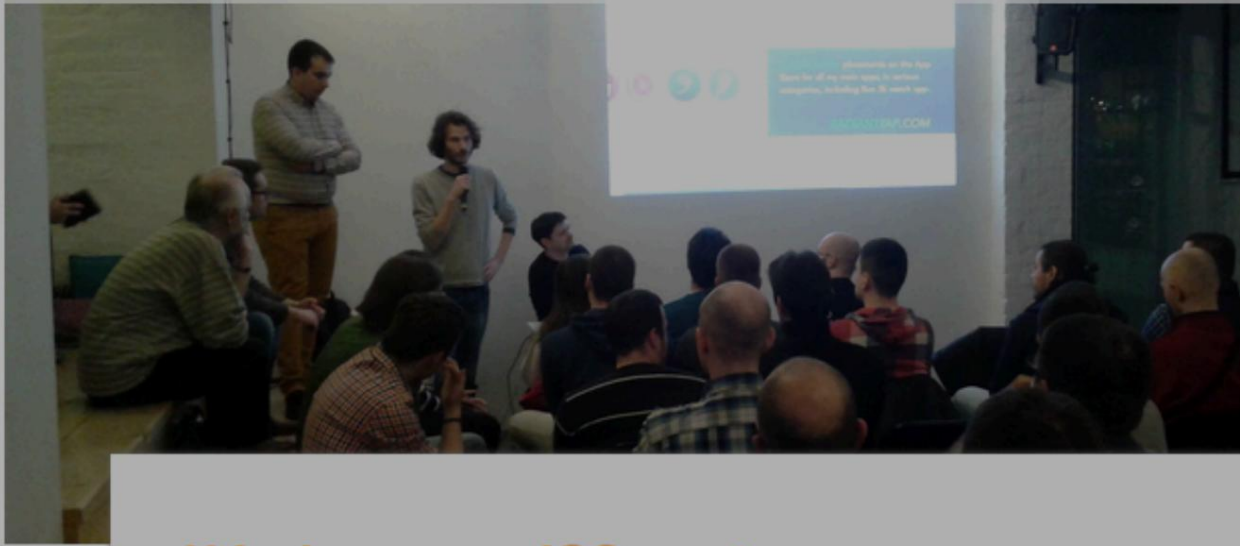
यह सुनिश्चित करने के लिए प्राथमिकता के रूप में 999 का उपयोग करें कि फ़ंक्शन डिफ़ॉल्ट वर्डप्रेस फ़िल्टर के बाद चलता है, अन्यथा यह यहां सेट किए गए को ओवरराइड कर देगा।

अंश के अंत में और पढ़ें लिंक जोड़ रहा हूँ

ऐसा करने के लिए, निम्नलिखित कोड को function.php में डालें:

```
फ़ंक्शन कस्टम_excerpt_more($अधिक) {
    वापसी '<a href="'. get_permalink($post->ID) . "'>और पढ़ें</a>';
} add_filter('excerpt_more', 'custom_excerpt_more');
```

परिणाम इस तरह दिखना चाहिए:



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



अंश के अंत में कुछ बिंदु जोड़ना

हमारे function.php में

```
फ़ंक्शन new_excerpt_more ($ अधिक ) {  
    वापस करना '.....';  
  
} add_filter('excerpt_more', 'new_excerpt_more');
```

हमें यह मिलना चाहिए:



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](#)



अध्याय 18: कस्टम पोस्ट प्रकार

वाक्य - विन्यास

- रजिस्टर_पोस्ट_टाइप (\$पोस्ट_टाइप, \$आर्ग्स);

पैरामीटर

पैरामीटर विवरण	
\$post_type (स्ट्रिंग) (आवश्यक)	
\$args	(सरणी/स्ट्रिंग) (वैकल्पिक)

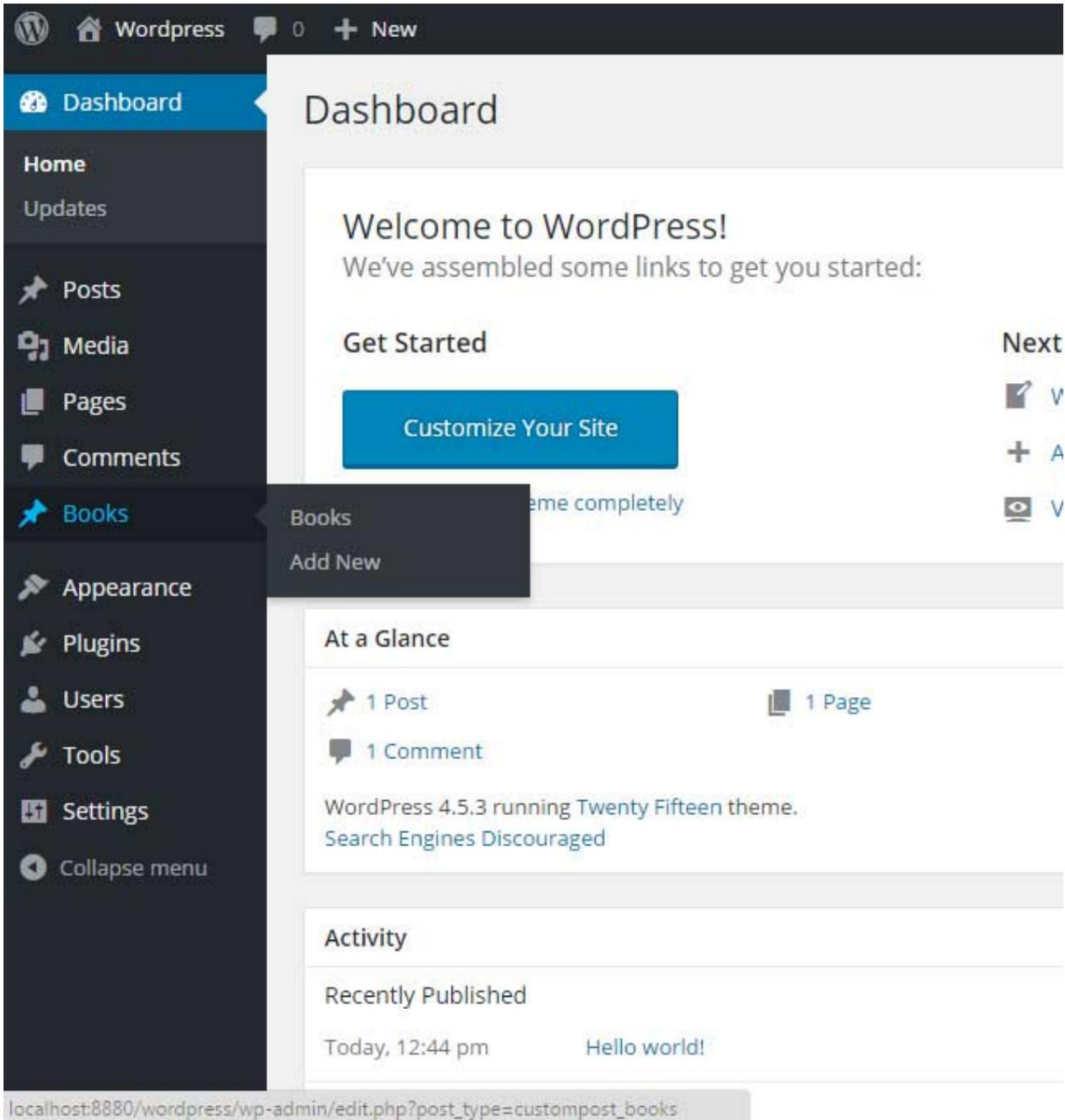
उदाहरण

एक कस्टम पोस्ट प्रकार पंजीकृत करना

मान लें कि आपके पास एक लाइब्रेरी वेबसाइट है, और आप पुस्तकें नामक एक कस्टम पोस्ट प्रकार चाहते हैं। के रूप में पंजीकृत किया जा सकता है

```
फ़ंक्शन create_bookposttype() { $args = array(  
  
    'सार्वजनिक' => सत्य, 'लेबल' =>  
    सरणी ( 'नाम' => __( 'पुस्तकें' ),  
        'एकवचन_नाम' => __( 'पुस्तक' )  
  
    ),  
);  
रजिस्टर_पोस्ट_टाइप ( 'कस्टमपोस्ट_बुक्स', $args );  
}  
  
add_action( 'init', 'create_bookposttype' );
```

और, यह जितना सरल है, अब आपके पास एक कस्टम पोस्ट प्रकार पंजीकृत है।



इस स्निपेट को आपकी थीम function.php फ़ाइल में, या एक प्लगइन संरचना में रखा जा सकता है।

मुख्य क्वेरी में कस्टम पोस्ट प्रकार जोड़ें

कस्टम पोस्ट प्रकार को पंजीकृत करने का मतलब यह नहीं है कि यह स्वचालित रूप से मुख्य क्वेरी में जुड़ जाता है। मुख्य क्वेरी में कस्टम पोस्ट प्रकार जोड़ने के लिए आपको `pre_get_posts` फ़िल्टर का उपयोग करने की आवश्यकता है।

```
// होम पेज पर 'पोस्ट' और 'बुक' कस्टम पोस्ट प्रकार के पोस्ट दिखाएं add_action('pre_get_posts', 'add_my_post_types_to_query' );

फ़ंक्शन add_my_post_types_to_query( $query ) यदि ( is_home() ) && $query->is_main_query() {
    $query->set('post_type', array('post', 'book' ) );
    वापसी $क्वेरी;
}
```

मुख्य RSS फ़ीड में कस्टम पोस्ट प्रकार जोड़ना

कस्टम पोस्ट प्रकार को पंजीकृत करने का मतलब यह नहीं है कि यह मुख्य RSS फ़ीड में जुड़ जाता है

स्वचालित रूप से। आपको मुख्य RSS फ़ीड में कस्टम पोस्ट प्रकार जोड़ने के लिए अनुरोध फ़िल्टर का उपयोग करने की आवश्यकता है।

```
// मुख्य आरएसएस फ़ीड पर 'किताबें' कस्टम पोस्ट प्रकार जोड़ें
फ़ंक्शन add_book_post_types_to_rss($qv) {
    यदि (isset($qv['फ़ीड']) && !isset($qv['post_type']))
        $qv['post_type'] = array('पोस्ट', 'किताबें', );
    वापसी $qv;
}
add_filter('अनुरोध', 'add_book_post_types_to_rss');
```

कस्टम पोस्ट प्रकार पंजीकृत करें

```
अगर ( ! function_exists('products_post_type') )
{
    फ़ंक्शन उत्पाद_पोस्ट_प्रकार() {
        $लेबल = सरणी(
            'नाम' => _x('उत्पाद', 'पोस्ट प्रकार सामान्य नाम', 'text_domain' ),
            'एकवचन_नाम' => _x('उत्पाद', 'पोस्ट प्रकार एकवचन नाम', 'text_domain' ),
            'नाम_एडमिन_बार' => __( 'उत्पाद', 'text_domain' ),
            => _x( 'उत्पाद', 'text_domain' ),
            'अभिलेखागार' => __( 'आइटम पुरालेख', 'text_domain' ),
            'गुण' => __( 'आइटम विशेषताएँ', 'text_domain' ),
            'parent_item_colon' 'all_items' => __( 'मूल उत्पाद:', 'text_domain' ),
            'add_new_item' => __( 'सभी उत्पाद', 'text_domain' ),
            'add_new' 'new_item' => __( 'नया उत्पाद जोड़ें', 'text_domain' ),
            'edit_item' => __( 'नया उत्पाद', 'text_domain' ),
            'update_item' => __( 'नया आइटम', 'टेक्स्ट_डोमेन' ),
            'view_item' => __( 'उत्पाद संपादित करें', 'text_domain' ),
            'view_items' => __( 'अपडेट उत्पाद', 'text_domain' ),
            'search_items' => __( 'उत्पाद देखें', 'text_domain' ),
            'not_found' => __( 'आइटम देखें', 'text_domain' ),
            'not_found_in_trash' => __( 'उत्पाद खोजें', 'text_domain' ),
            'featured_image' => __( 'कोई उत्पाद नहीं मिला', 'text_domain' ),
            'set_featured_image' => __( 'ट्रैश में कोई उत्पाद नहीं मिला', 'text_domain' ),
            'remove_featured_image' => __( 'फीचर्ड इमेज', 'टेक्स्ट_डोमेन' ),
            __( ' चुनिंदा छवि हटाएं', 'text_domain' ), => __( 'विशेष रूप से प्रदर्शित छवि सेट करें', 'text_domain' ),
            'featured_image का उपयोग करें' => __( 'फीचर्ड छवि के रूप में उपयोग करें', 'text_domain' ),
            'आइटम में डालें' => __( 'आइटम में डालें', 'text_domain' ),
            'इस आइटम पर अपलोड किया गया' => __( 'इस आइटम पर अपलोड किया गया', 'text_domain' ),
            'आइटम सूची' => __( 'आइटम सूची', 'टेक्स्ट_डोमेन' ),
            'आइटम सूची नेविगेशन' => __( 'आइटम सूची नेविगेशन', 'टेक्स्ट_डोमेन' ),
            'फ़िल्टर आइटम सूची' => __( 'आइटम सूची फ़िल्टर करें', 'text_domain' ),
        );
        $args = सरणी(
            'लेबल' => __( 'उत्पाद', 'text_domain' ),
            'विवरण' => __( 'उत्पाद जानकारी पृष्ठ', 'text_domain' ),
            'लेबल' => $लेबल,
            'समर्थन' 'टिप्पणियाँ', => सरणी ( 'शीर्षक', 'संपादक', 'अंश', 'लेखक', 'थंबनेल',
            'कस्टम-फ़िल्ड', ),
            'वर्गीकरण' => सरणी ( 'श्रेणी', 'पोस्ट_टैग' ),
            'पदानुक्रमित' => झूठा,
            'सार्वजनिक' => सच,
            'show_ui' => सच,
        );
    }
}
```

```

    'मेनू में दिखाएं' 'मेनू स्थिति'           => सच,
    'मेनू_आइकन' 'एडमिन_बार में
    दिखाएं' 'नेव_मेनू में
    दिखाएं'                               => 'डेशिकॉन-उत्पाद',
                                           => सच,
                                           => सच,
    'निर्यात कर सकते हैं'                 => सच,
    'है_संग्रह'                           => सच,
    'exclude_from_search' => गलत,
    'सार्वजनिक रूप से_क्वेरी करने योग्य' => सच,
    'क्षमता प्रकार' 'बाकी में दिखाएं'    => 'पेज',
                                           => सच,
);
रजिस्टर_पोस्ट_टाइप ('उत्पाद', $args);

}
add_action('init', 'products_post_type', 0);

}

```

ट्वेंटी फिफटीन वर्डप्रेस थीम का उपयोग करके कस्टम पोस्ट प्रकार

आप फ़ंक्शन के लिए किसी भी नाम का उपयोग कर सकते हैं।

```

फ़ंक्शन कस्टम_पोस्टटाइप(){
    रजिस्टर_पोस्ट_प्रकार('cus_post',array(

        'लेबल'=>सरणी(
            'नाम'=>'खाइयम'// आप अपने उपयोगकर्ताओं के लिए मेनू में जो भी नाम दिखाना चाहते हैं उसका उपयोग करें
        ),
        'सार्वजनिक'=>सत्य,// **आवश्यक होना चाहिए
        'समर्थन'=>सरणी('शीर्षक','संपादक','थंबनेल')// सुविधाएँ जो आप अपने पर प्रदान करना चाहते हैं
    पदों
    ));
}
add_action('after_setup_theme','custom_posttype');

```

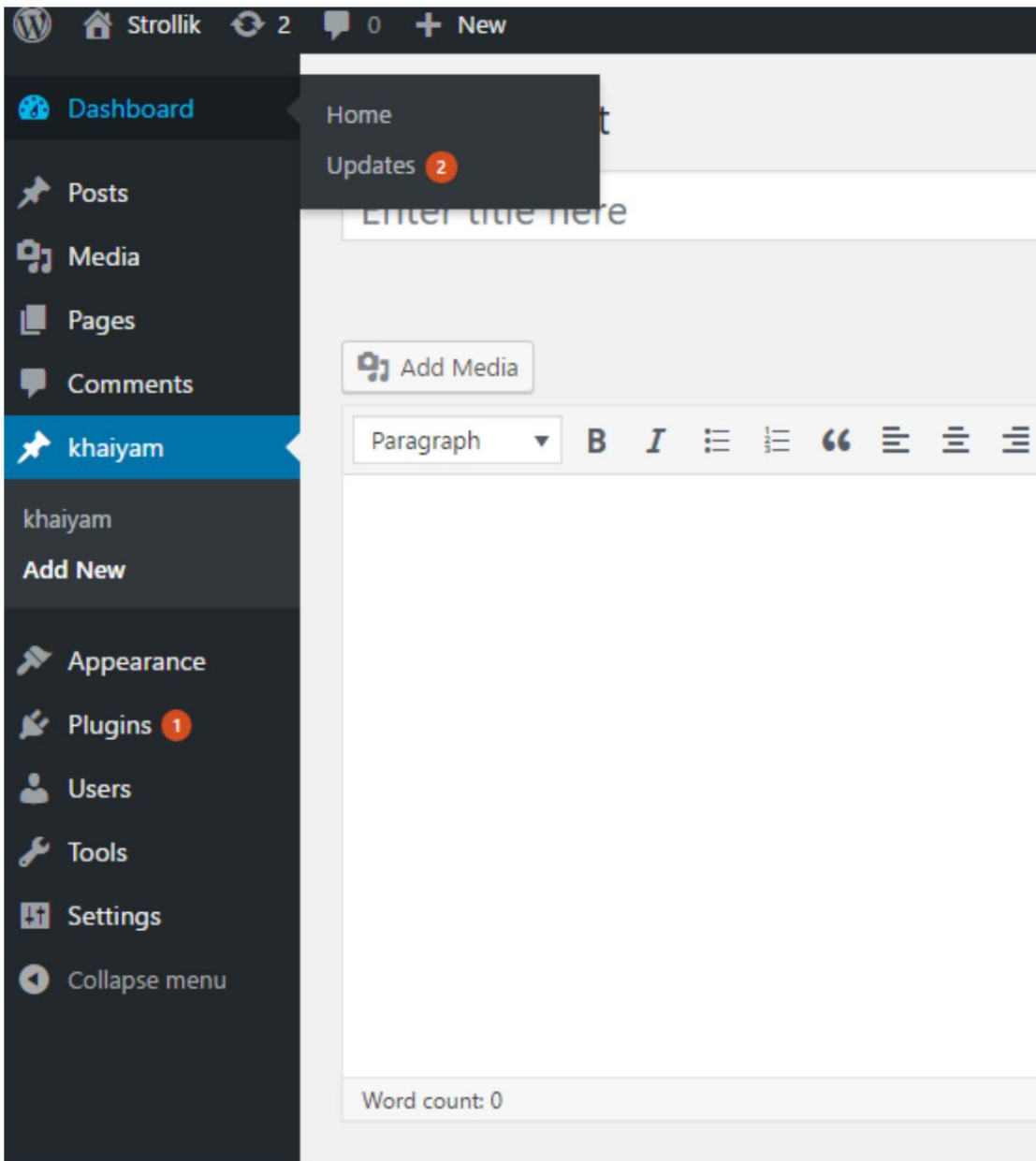
या

```

add_action('init','custom_posttype');

```

आप अपनी इच्छानुसार किसी भी हुक का उपयोग कर सकते हैं लेकिन निश्चित रूप से उनके अलग-अलग अर्थ और उपयोग हैं।



डिफ़ॉल्ट खोज में कस्टम पोस्ट प्रकार

आप डिफ़ॉल्ट वर्डप्रेस खोज पर कस्टम पोस्ट प्रकार के पोस्ट जोड़ सकते हैं, थीम function.php में नीचे दिए गए कोड को जोड़ें

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

}add_action('pre_get_posts','my_search_filter');
```

अध्याय 19: कस्टमाइज़र मूल बातें (पैनल जोड़ें, अनुभाग, सेटिंग, नियंत्रण)

उदाहरण

एक कस्टमाइज़र पैनल जोड़ें

```
<?php
/**
 * पैनल: WPCustomize
 *
 * बुनियादी नियंत्रण के साथ बेसिक कस्टमाइज़र पैनल।
 *
 * @चूँकि *          1.0.0
 * @पैकेज डब्ल्यूपीसी
 */

// यदि सीधे पहुँच हो तो बाहर निकलें।
यदि (! परिभाषित ('ABSPATH')) बाहर निकलें;

}

// फ़ंक्शन को अनुकूलित करें।
यदि (! function_exists('wpc_panel_wpcustomize')) {
    // रजिस्टर कार्रवाई को अनुकूलित करें।
    add_action('customize_register', 'wpc_panel_wpcustomize');
    /**
     * पैनल को अनुकूलित करें।
     *
     * बुनियादी नियंत्रणों के साथ एक पैनल, अनुभाग जोड़ता है।
     *
     * @param ऑब्जेक्ट WP_Customize $wp_customize WP_Customize_Manager वर्ग का उदाहरण।
     * @1.0.0 से
     */
    फ़ंक्शन wpc_panel_wpcustomize( $wp_customize ) {
        // पैनल: बेसिक।
        $wp_customize->add_panel('wpc_panel_wpcustomize', array(
            'प्राथमिकता' 'शीर्षक'          => 10,
            => __('WPCकस्टमाइज़र पैनल शीर्षक', 'WPC' ),
            'विवरण' 'क्षमता'          => __('पैनल विवरण', 'डब्ल्यूपीसी'),
            => 'संपादन_थीम_विकल्प'
        ));
    }
}
```

बुनियादी सेटिंग्स और उनके नियंत्रण के साथ एक कस्टमाइज़र अनुभाग जोड़ें

पैनल में अनुभाग हो सकते हैं, अनुभाग में सेटिंग्स हो सकती हैं, और सेटिंग्स में नियंत्रण हो सकते हैं। सेटिंग्स हैं डेटाबेस में सहेजा गया है, जबकि विशेष सेटिंग्स के नियंत्रणों का उपयोग केवल उन्हें प्रदर्शित करने के लिए किया जाता है उपयोगकर्ता के अनुरूप सेटिंग।

यह कोड ऊपर से पैनल में एक मूल अनुभाग बनाता है। अंदर कुछ बुनियादी सेटिंग्स हैं

को नियंत्रित करता है

जुड़ा हुआ।

```
<?php
/**
 * अनुभाग: बुनियादी
 *
 * बुनियादी नियंत्रण के साथ बुनियादी कस्टमाइज़र अनुभाग।
 *
 * @चूकि *          1.0.0
 *
 * @पैकेज डब्ल्यूपीसी
 */

// यदि सीधे पहुंच हो तो बाहर निकलें।
यदि (! परिभाषित ('ABSPATH') बाहर निकलें; ) {

}

// फ़ंक्शन को अनुकूलित करें।
यदि (! function_exists( 'wpc_customize_panel_basic' ) ) {
    // रजिस्टर कार्रवाई को अनुकूलित करें।
    add_action('customize_register', 'wpc_customize_panel_basic');

    /**
     * पैनल को अनुकूलित करें।
     *
     * बुनियादी नियंत्रणों के साथ एक पैनल, अनुभाग जोड़ता है।
     *
     * @param ऑब्जेक्ट WP_Customize $wp_customize WP_Customize_Manager वर्ग का उदाहरण।
     * @1.0.0 से
     */
    फ़ंक्शन wpc_customize_panel_basic( $wp_customize ) {
        // अनुभाग: मूल।
        $wp_customize->add_section( 'wpc_section_basic', array(
            'प्राथमिकता' 'पैनल'          => 10,
            'शीर्षक'                    => 'wpc_panel_wpcustomize',
            => __('मूल अनुभाग शीर्षक', 'डब्ल्यूपीसी'),
            'विवरण' 'क्षमता'          => __('अनुभाग विवरण', 'डब्ल्यूपीसी'),
            => 'संपादन_थीम_विकल्प'
        ));

        // सेटिंग: टेक्स्ट.
        $wp_customize->add_setting('wpc_text', array(
            'प्रकार'                    => 'थीम_मॉड',
            'गलती करना'              => 'प्लेसहोल्डर',
            'परिवहन' 'क्षमता'        => 'रिफ्रेश', // विकल्प: रिफ्रेश या पोस्टमैसेज।
            'सैनिटाइज़_कॉलबैक'      => 'संपादित करें_थीम_विकल्प',
            => 'esc_attr'
        ));

        // नियंत्रण: पाठ।
        $wp_customize->add_control('wpc_text', array(
            'लेबल'                    => __('पाठ', 'डब्ल्यूपीसी'),
            'विवरण' => __('विवरण', 'डब्ल्यूपीसी'),
            'अनुभाग'                  => 'wpc_section_basic',
            'प्रकार'                    => 'पाठ'
        ));

        // सेटिंग: टेक्स्टएरिया।
        $wp_customize->add_setting( 'wpc_textarea', array(
            'प्रकार'                    => 'थीम_मॉड',
            'गलती करना'              => 'प्लेसहोल्डर टेक्स्टएरिया।',
```

```

'परिवहन' 'क्षमता' => 'रिफ्रेश', // विकल्प: रिफ्रेश या पोस्टमैसेज।
'सैनिटाइज़_कॉलबैक' => 'संपादित करें_थीम_विकल्प',
=> 'exc_textarea'

));

// नियंत्रण: टेक्स्टएरिया।
$wp_customize->add_control('wpc_textarea', array(
    'लेबल' => __('टेक्स्टएरिया', 'डब्ल्यूपीसी'),
    'विवरण' => __( 'विवरण', 'डब्ल्यूपीसी' ),
    'अनुभाग' => 'wpc_section_basic',
    'प्रकार' => 'टेक्स्टएरिया'
));

// सेटिंग: चेकबॉक्स।
$wp_customize->add_setting( 'wpc_checkbox', array(
    'प्रकार' => 'थीम_मॉड',
    'गलती करना' => 'सक्षम करें',
    'परिवहन' 'क्षमता' => 'रिफ्रेश', // विकल्प: रिफ्रेश या पोस्टमैसेज।
    'sanitize_callback' => 'संपादित करें_थीम_विकल्प',
    कस्टमाइज़र-sanitization.php फ़ाइल => 'wpc_sanitize_checkbox' // कस्टम फ़ंक्शन इन
));

// नियंत्रण: चेकबॉक्स।
$wp_customize->add_control('wpc_checkbox', array(
    'लेबल' => __('चेकबॉक्स', 'डब्ल्यूपीसी'),
    'विवरण' => __( 'विवरण', 'डब्ल्यूपीसी' ),
    'अनुभाग' => 'wpc_section_basic',
    'प्रकार' => 'चेकबॉक्स'
));

// सेटिंग: रेडियो।
$wp_customize->add_setting( 'wpc_radio', array(
    'प्रकार' => 'थीम_मॉड',
    'गलती करना' => 'चालू',
    'परिवहन' 'क्षमता' => 'रिफ्रेश', // विकल्प: रिफ्रेश या पोस्टमैसेज।
    'sanitize_callback' => 'संपादित करें_थीम_विकल्प',
    sanitization.php फ़ाइल => 'wpc_sanitize_select'. // कस्टमाइज़र में कस्टम फ़ंक्शन
));

// नियंत्रण: रेडियो।
$wp_customize->add_control('wpc_radio', array(
    'लेबल' => __('रेडियो', 'डब्ल्यूपीसी'),
    'विवरण' => __( 'विवरण', 'डब्ल्यूपीसी' ),
    'अनुभाग' => 'wpc_section_basic',
    'प्रकार' => 'रेडियो',
    'विकल्प' => सरणी(
        'सक्षम करें' => 'सक्षम करें',
        'अक्षम' => 'अक्षम करें'
    )
));

// सेटिंग: चुनें।
$wp_customize->add_setting( 'wpc_select', array(
    'प्रकार' => 'थीम_मॉड',
    'गलती करना' => 'सक्षम करें',
    'परिवहन' 'क्षमता' => 'रिफ्रेश', // विकल्प: रिफ्रेश या पोस्टमैसेज।
    'sanitize_callback' => 'संपादित करें_थीम_विकल्प',
    sanitization.php फ़ाइल => 'wpc_sanitize_select' // कस्टमाइज़र में कस्टम फ़ंक्शन
));

```

```
));  
  
// नियंत्रण: चयन करें. $wp_customize-  
>add_control('wpc_select', array(  
    'लेबल' => __('चयन करें', 'डब्ल्यूपीसी'),  
    'विवरण' => __( 'विवरण', 'डब्ल्यूपीसी' ), 'अनुभाग'  
        => 'wpc_section_basic',  
    'प्रकार' => 'चयन करें',  
    'विकल्प' => सरणी(  
        'सक्षम करें' => 'सक्षम करें',  
        'अक्षम' => 'अक्षम करें'  
    )  
));  
}  
}
```

अध्याय 20: कस्टमाइज़र हैलो वर्ल्ड

पैरामीटर

पैरामीटर विवरण

mytheme आपकी थीम (या चाइल्ड थीम) के लिए एक विशिष्ट पहचानकर्ता। यह आपका थीम स्लग हो सकता है

उदाहरण

हैलो वर्ल्ड उदाहरण

कस्टमाइज़र की मूल अवधारणा यह है कि व्यवस्थापक अपनी साइट पर परिवर्तनों का लाइव पूर्वावलोकन कर सकते हैं, और फिर उन्हें स्थायी रूप से जोड़ सकते हैं।

निम्नलिखित को थीम की function.php फ़ाइल में कॉपी और पेस्ट किया जा सकता है

- माई फ़र्स्ट सेक्शन नामक एक कस्टमाइज़र अनुभाग जोड़ें • हैलो वर्ल्ड कलर नामक एक कस्टमाइज़र सेटिंग जोड़ें जो व्यवस्थापक को एक रंग चुनने की अनुमति देता है .hello-world के लिए एक सीएसएस नियम
- जोड़ें जो चुने गए रंग के अनुरूप होगा और यदि कुछ भी नहीं चुना गया है तो #000000 पर डिफ़ॉल्ट होगा . सेटिंग को <head> के अंत में एक <style> टैग में डाला जाएगा।

```
फ़ंक्शन mytheme_customize_register( $wp_customize )  
{  
  
    $wp_customize->add_section( 'my_first_section_id' , array(  
        'शीर्षक' => __( 'मेरा पहला खंड', 'माइथीम' ),  
        'प्राथमिकता' => 30,  
    ));  
  
    $wp_customize->add_setting( 'hello_world_color' , सरणी(  
        'गलती करना' => '#000000', 'परिवहन'  
    => 'ताज़ा करें',  
    ));  
  
    $wp_customize->add_control(नया WP_Customize_Color_Control( $wp_customize, 'link_color', array(  
  
        'लेबल' => __( 'हैलो वर्ल्ड कलर', 'माइथीम' ), => 'माय_फ़र्स्ट_सेक्शन_आईडी',  
        'अनुभाग'  
        'सेटिंग्स' => 'हैलो_वर्ल्ड_कलर', ));  
    )  
  
} add_action('customize_register', 'mytheme_customize_register' );  
  
फ़ंक्शन mytheme_customize_css() {  
  
    ?>  
    <शैली प्रकार='टेक्स्ट/सीएसएस'>
```

```
.हैलो-वर्ल्ड {रंग: #<?php इको get_theme_mod('हैलो_वर्ल्ड_कलर', '000000'); ?>; } </style> <?php  
}  
add_action('wp_head', 'mytheme_customize_css');
```

अध्याय 21: डिबगिंग

परिचय

https://codex.wordpress.org/Debugging_in_WordPress

PHP कोड को डिबग करना किसी भी प्रोजेक्ट का हिस्सा है, लेकिन वर्डप्रेस विशिष्ट डिबग सिस्टम के साथ आता है जो प्रक्रिया को सरल बनाने के साथ-साथ कोर, प्लगइन्स और थीम में कोड को मानकीकृत करने के लिए डिज़ाइन किया गया है।

टिप्पणियां

वर्डप्रेस में डिबगिंग के लिए प्लगइन्स:

- <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/>

उदाहरण

WP_DEBUG

`WP_DEBUG` एक PHP स्थिरांक (एक स्थायी वैश्विक चर) है जिसका उपयोग पूरे वर्डप्रेस में "डीबग" मोड को ट्रिगर करने के लिए किया जा सकता है। इसे डिफ़ॉल्ट रूप से गलत माना जाता है और आमतौर पर इसे `wp-config.php` में सत्य पर सेट किया जाता है वर्डप्रेस की विकास प्रतियों पर फ़ाइल।

```
परिभाषित करें ('WP_DEBUG', सत्य); परिभाषित करें  
( 'WP_DEBUG', गलत);
```

WP_DEBUG_LOG

`WP_DEBUG_LOG` `WP_DEBUG` का एक साथी है जो सभी त्रुटियों को `/wp-content/` निर्देशिका के अंदर एक `debug.log` लॉग फ़ाइल में सहेजने का कारण बनता है। यह उपयोगी है यदि आप बाद में सभी सूचनाओं की समीक्षा करना चाहते हैं या ऑफ-स्क्रीन उत्पन्न सूचनाओं को देखना चाहते हैं (उदाहरण के लिए AJAX अनुरोध या `wp-cron` रन के दौरान)।

```
//सक्षम  
परिभाषित करें ('WP_DEBUG_LOG', सत्य);  
  
//अक्षम करना  
परिभाषित करें ('WP_DEBUG_LOG', गलत);
```

WP_DEBUG_DISPLAY

`WP_DEBUG_DISPLAY` `WP_DEBUG` का एक अन्य साथी है जो नियंत्रित करता है कि डिबग संदेश हैं या नहीं

पृष्ठों के HTML के अंदर दिखाया गया है या नहीं। डिफॉल्ट 'सत्य' है जो उत्पन्न होने पर त्रुटियों और चेतावनियों को दिखाता है। इसे गलत पर सेट करने से सभी त्रुटियाँ छिप जाएँगी। इसका उपयोग WP_DEBUG_LOG के संयोजन में किया जाना चाहिए ताकि बाद में त्रुटियों की समीक्षा की जा सके। नोट: WP_DEBUG_DISPLAY के लिए कुछ भी करने के लिए, WP_DEBUG सक्षम होना चाहिए (सही)।

```
// परिभाषित
सक्षम करें ('WP_DEBUG_DISPLAY', सत्य);

// परिभाषित अक्षम
करें ('WP_DEBUG_DISPLAY', गलत);
```

SCRIPT_DEBUG

SCRIPT_DEBUG एक संबंधित स्थिरांक है जो वर्डप्रेस को सामान्य रूप से लोड किए गए लघु संस्करणों के बजाय कोर सीएसएस और जावास्क्रिप्ट फ़ाइलों के "डेव" संस्करणों का उपयोग करने के लिए मजबूर करेगा। यह तब उपयोगी होता है जब आप किसी अंतर्निहित .js या .css फ़ाइलों में संशोधन का परीक्षण कर रहे हों। डिफॉल्ट गलत है।

```
//सक्षम
परिभाषित करें ('SCRIPT_DEBUG', सत्य);

//अक्षम करना
परिभाषित करें ('SCRIPT_DEBUG', गलत);
```

सहेजें प्रश्न

SAVEQUERIES परिभाषा डेटाबेस क्वेरीज़ को एक सरणी में सहेजती है और उस सरणी को उन क्वेरीज़ का विश्लेषण करने में सहायता के लिए प्रदर्शित किया जा सकता है। सत्य के रूप में परिभाषित स्थिरांक प्रत्येक क्वेरी को सहेजने का कारण बनता है, उस क्वेरी को निष्पादित होने में कितना समय लगा, और उसे कौन सा फ़ंक्शन कहा जाता है। ध्यान दें: इसका आपकी साइट पर प्रदर्शन पर प्रभाव पड़ेगा, इसलिए जब आप डिबगिंग नहीं कर रहे हों तो इसे बंद करना सुनिश्चित करें।

```
परिभाषित करें ('SAVEQUERIES', सत्य);
```

सारणी को में संग्रहीत किया जाता है

```
वैश्विक $wpdb->प्रश्न;
```

उदाहरण wp-config.php और डिबगिंग के लिए अच्छे अभ्यास

आपकी wp-config.php फ़ाइल में डाला गया निम्न कोड, सभी त्रुटियों, सूचनाओं और चेतावनियों को wp-content निर्देशिका में debug.log नामक फ़ाइल में लॉग करेगा। यह त्रुटियों को भी छिपा देगा ताकि वे पृष्ठ निर्माण में बाधा न डालें।

```
// WP_DEBUG मोड सक्षम करें
परिभाषित करें ('WP_DEBUG', सत्य);

// /wp-content/debug.log फ़ाइल में डिबग लॉगिंग सक्षम करें
परिभाषित करें ('WP_DEBUG_LOG', सत्य);

// त्रुटियों और चेतावनियों का प्रदर्शन अक्षम करें
```

```
परिभाषित करें ('WP_DEBUG_DISPLAY', गलत); @ini_set('display_errors',  
0);
```

```
// कोर जेएस और सीएसएस फ़ाइलों के डेव संस्करणों का उपयोग करें (केवल तभी आवश्यक है जब आप इन कोर फ़ाइलों को संशोधित कर रहे हों) परिभाषित करें ('SCRIPT_DEBUG', सत्य);
```

अच्छा अभ्यास यदि आप डीबग लॉग में कस्टम संदेश जोड़ना चाहते हैं तो अपने प्लगइन या थीम में फ़ोलोइंग कोड जोड़ें।

```
//चेकिंग function_exists है  
यदि ( !function_exists( 'print_to_log' ) ) {  
    //यदि डिबगिंग चालू है तो फ़ंक्शन डिबग.लॉग पर एक संदेश लिखता है। फ़ंक्शन प्रिंट_टू लॉग ($संदेश)  
  
    {  
        यदि (सत्य === WP_DEBUG) {  
            अगर ( is_array( $message ) || is_object( $message ) ) {  
                त्रुटि_लॉग(प्रिंट_आर($संदेश, सत्य));  
            } अन्य {  
                त्रुटि_लॉग ($संदेश);  
            }  
        }  
    }  
}
```

लॉग को एक अलग फ़ाइल में देखें

जब आपके पास अजाक्स कॉल होती है, तो कॉलबैक फ़ंक्शन के अंदर से लॉग प्राप्त करना बेहद मुश्किल होता है।

लेकिन यदि आप डिबगिंग सक्षम करते हैं

```
परिभाषित करें ('WP_DEBUG', सत्य);
```

और फिर उसके बाद जोड़ें

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

आपके रूट फ़ोल्डर में एक error.log.txt फ़ाइल होगी जहां आपके सभी लॉग स्थित हैं। आप उनसे लॉग इन भी कर सकते हैं

```
त्रुटि_लॉग (प्रिंट_आर ('मैं जो जांचना चाहता हूँ वह यहां जाता है', सत्य));
```

आपके कोड के अंदर. इससे आपका जीवन काफी आसान हो जाएगा.

अध्याय 22: लिपियों को कतारबद्ध करना

वाक्य - विन्यास

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

पैरामीटर

पैरामीटर विवरण	
\$संभाल	(स्ट्रिंग) (आवश्यक) स्क्रिप्ट का नाम. अद्वितीय होना चाहिए.
\$src	(स्ट्रिंग) (वैकल्पिक) स्क्रिप्ट का पूरा यूआरएल, या वर्डप्रेस रूट निर्देशिका के सापेक्ष स्क्रिप्ट का पथ। डिफ़ॉल्ट मान: गलत
\$deps	(सरणी) (वैकल्पिक) पंजीकृत स्क्रिप्ट की एक सरणी इस स्क्रिप्ट को संभालती है जिस पर निर्भर करती है। डिफ़ॉल्ट मान: सरणी()
\$ver	(स्ट्रिंग बूल नल) (वैकल्पिक) स्ट्रिंग स्क्रिप्ट संस्करण संख्या निर्दिष्ट करती है, यदि इसमें एक है, जिसे कैश बस्टिंग उद्देश्यों के लिए क्वेरी स्ट्रिंग के रूप में यूआरएल में जोड़ा जाता है। यदि संस्करण गलत पर सेट है, तो वर्तमान स्थापित वर्डप्रेस संस्करण के बराबर एक संस्करण संख्या स्वचालित रूप से जोड़ दी जाती है। यदि शून्य पर सेट किया जाता है, तो कोई संस्करण नहीं जोड़ा जाता है। डिफ़ॉल्ट मान: गलत
\$in_footer	(बूल) (वैकल्पिक) क्या स्क्रिप्ट को <head> के बजाय </body> से पहले पंक्तिबद्ध किया जाए। डिफ़ॉल्ट मान: गलत

उदाहरण

Functions.php में स्क्रिप्टों को कतारबद्ध करना

यदि आप कस्टम.जेएस स्क्रिप्ट जोड़ना चाहते हैं जो आपके थीम के जेएस/ फ़ोल्डर में स्थित है, तो आपको इसे सूचीबद्ध करना होगा। Functions.php में जोड़ें

```
<?php

add_action('after_setup_theme', 'yourtheme_theme_setup' );

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

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

    }
}
```

```

यदि !function_exists('yourtheme_scripts') ) {
    (फ़ंक्शन yourtheme_scripts() {

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

    }
}

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

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

}
}

```

केवल IE के लिए स्क्रिप्टों को कतारबद्ध करें

```

add_action('wp_enqueue_scripts', 'enqueue_my_styles_and_scripts' );

/**
 * स्क्रिप्ट (या शैलियों) को सशर्त रूप से सूचीबद्ध करें।
 *
 * विशेष रूप से IE के लिए स्क्रिप्ट (या स्टाइलशीट) लोड करें। IE10 और इसके बाद के संस्करण मानक मोड में सशर्त टिप्पणियों का समर्थन नहीं करते हैं।
 *
 * @लिंक https://gist.github.com/wpscholar/4947518 * @लिंक https://msdn.microsoft.com/en-us/library/ms537512\(v=vs.85\).aspx */
फ़ंक्शन enqueue_my_styles_and_scripts() {

    // इंटरनेट एक्सप्लोरर HTML5 समर्थन
    wp_enqueue_script('html5shiv',get_template_directory_uri().'/js/html5shiv.js', array(), '3.7.3', false); wp_script_add_data('html5shiv', 'सशर्त', 'lt IE 9');

    // इंटरनेट एक्सप्लोरर 8 मीडिया क्वेरी समर्थन wp_enqueue_script('रिस्पॉंड',
    get_template_directory_uri().'/js/respond.js', ऐरे(), '1.4.2', गलत); wp_script_add_data('प्रतिक्रिया', 'सशर्त', 'lt IE 9');

}

```

विशिष्ट पृष्ठों के लिए सशर्त रूप से लिपियों को सूचीबद्ध करना

आप अपनी वेबसाइट के विशिष्ट पृष्ठों पर स्क्रिप्टों को सूचीबद्ध करने के लिए वर्डप्रेस में सशर्त ऑपरेटर्स का उपयोग कर सकते हैं।

```

फ़ंक्शन लोड_स्क्रिप्ट_फॉर_सिंगल_पोस्ट(){ if(is_single()){ wp_enqueue_script(

    'कुछ',
    get_template_directory_uri().'/js/some.js', सराणी('jquery'), '1.0.0',

```

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

उपरोक्त उदाहरण में, यदि वर्तमान वेबपेज एकल पोस्ट है, तो स्क्रिप्ट को पंक्तिबद्ध किया जाएगा। अन्यथा wp_enqueue_script फ़ंक्शन निष्पादित नहीं किया जाएगा।

अध्याय 23: कतारबद्ध शैलियाँ

वाक्य - विन्यास

1. `wp_enqueue_style($हैंडल, $src, $निर्भरता, $संस्करण, $मीडिया);`

पैरामीटर

पैरामीटर	विवरण
\$संभाल	(स्ट्रिंग) (आवश्यक) स्टाइलशीट के लिए अद्वितीय नाम।
\$src	(स्ट्रिंग) (वैकल्पिक) स्टाइलशीट का यूआरएल जिसका उपयोग लिंक टैग के स्रोत विशेषता के अंदर किया जाएगा।
\$deps	(स्ट्रिंग) (वैकल्पिक) स्टाइलशीट की एक सरणी इस स्टाइलशीट को संभालती है जिस पर निर्भर करती है।
\$ver	(स्ट्रिंग) (वैकल्पिक) स्टाइलशीट के स्टाइलशीट संस्करण को निर्दिष्ट करने वाली स्ट्रिंग।
\$मीडिया	(स्ट्रिंग) (वैकल्पिक) वह मीडिया जिसके लिए यह स्टाइलशीट बनाई गई है। जैसे 'सभी', 'प्रिंट', 'स्क्रीन' आदि

उदाहरण

निर्भरता के रूप में किसी अन्य सीएसएस फ़ाइल के साथ आंतरिक सीएसएस फ़ाइल शामिल करना

```
फ़ंक्शन थीमSlug_enqueue_scripts() {  
    wp_enqueue_style('थीमस्लग-रीसेट', get_template_directory_uri() .'/css/reset.css',  
    '1.0.0');  
    wp_enqueue_style('थीमस्लग-स्टाइल', get_template_directory_uri() .'/style.css',  
    'थीमस्लग-रीसेट', '1.0.0'); }  
  
add_action('wp_enqueue_scripts', 'themeSlug_enqueue_scripts');
```

आंतरिक सीएसएस फ़ाइल सहित

इस मामले में style.css थीम के फ़ोल्डर के मूल में स्थित है

```
फ़ंक्शन थीमSlug_enqueue_scripts() {  
    wp_enqueue_style('themeSlug-style', get_template_directory_uri() .'/style.css', '1.0.0');  
  
} add_action('wp_enqueue_scripts', 'themeSlug_enqueue_scripts');
```

बाहरी सीएसएस फ़ाइल सहित

इस उदाहरण में हम फ्रॉन्ट अड्डुत आइकन फ्रॉन्ट शामिल करना चाहते हैं

```

फ़ंक्शन थीमSlug_enqueue_scripts() {
    wp_enqueue_style('font-awesome', '//cdnjs.cloudflare.com/ajax/libs/font-
अद्भुत/4.6.3/css/font-awesome.css');
}
add_action('wp_enqueue_scripts', 'themeSlug_enqueue_scripts');

```

केवल IE के लिए स्टाइलशीट संकलित करें

```

add_action('wp_enqueue_scripts', 'enqueue_my_styles_and_scripts');

/**
 * शैलियों (या स्क्रिप्ट) को सशर्त रूप से सूचीबद्ध करें।
 *
 * विशेष रूप से IE के लिए स्टाइलशीट (या स्क्रिप्ट) लोड करें। IE10 और इसके बाद के संस्करण मानक मोड में सशर्त टिप्पणियों का समर्थन नहीं करते हैं।
 *
 * @लिंक https://gist.github.com/wpscholar/4947518 * @लिंक https://msdn.microsoft.com/
en-us/library/ms537512(v=vs.85).aspx */
फ़ंक्शन enqueue_my_styles_and_scripts() {
    // इंटरनेट एक्सप्लोरर विशिष्ट स्टाइलशीट। wp_enqueue_style('themenam-ie',
    get_stylesheet_directory_uri() . '/css/ie.css', array(
'द्वैतीफिफ्टीन-स्टाइल' ), '20141010' ); wp_style_add_data('themenam-ie',
'conditional', 'lte IE 9');

    // इंटरनेट एक्सप्लोरर 7 विशिष्ट स्टाइलशीट। wp_enqueue_style('themenam-ie7',
    get_stylesheet_directory_uri() . '/css/ie7.css', array(
'द्वैतीफिफ्टीन-स्टाइल' ), '20141010' ); wp_style_add_data('themenam-
ie7', 'सशर्त', 'lt IE 8');
}

```

आपके प्लगइन क्लास के लिए आंतरिक सीएसएस फ़ाइल शामिल है

```

क्लास My_Plugin() { फ़ंक्शन
    __construct() {
        add_action('wp_enqueue_scripts', array($this, 'init_fe_assets' ));
    }

    सार्वजनिक फ़ंक्शन init_fe_assests() {
        wp_enqueue_style('माय-प्लगइन', प्लगइन_dir_url(__FILE__)
'संपत्ति/सीएसएस/फ्रंटएंड/प्लगइन.सीएसएस', सरणी(), '0.0.1', सत्य);
    }
}

नया My_Plugin();

```

वैकल्पिक स्टाइलशीट जोड़ें

```

<?php wp_enqueue_style('थीम-पांच', get_template_directory_uri() .
'/पथ/से/अतिरिक्त/सीएसएस'); wp_style_add_data('थीम-
पांच', 'alt', true); wp_style_add_data('थीम-पांच', 'शीर्षक', __('थीम-पांच.सीएसएस', 'आपका-थीम-
नाम')); ?>

```


अध्याय 24: फ़ंक्शन: wp_trim_words()

वाक्य - विन्यास

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

पैरामीटर

पैरामीटर विवरण	
\$पाठ	(स्ट्रिंग) (आवश्यक) पाठ जिसे छोटा या काटा जाएगा।
\$num_words	(पूर्णांक) (आवश्यक) उन शब्दों की संख्या जिन तक पाठ प्रतिबंधित होगा।
\$अधिक	(स्ट्रिंग) (वैकल्पिक) यदि \$text को ट्रिम करने की आवश्यकता है तो क्या जोड़ें।

उदाहरण

पोस्ट सामग्री को ट्रिम करना

यह फ़ंक्शन टेक्स्ट को शब्दों की एक निर्दिष्ट संख्या तक छोटा कर देता है और संक्षिप्त टेक्स्ट लौटा देता है।

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

उपरोक्त उदाहरण में हम पोस्ट सामग्री को फ़ंक्शन में पास कर रहे हैं। यह सामग्री की लंबाई को 40 शब्दों तक सीमित कर देगा और बाकी शब्दों को काट देगा।

अध्याय 25: फ़ंक्शन: add_action()

वाक्य - विन्यास

- ऐड_एक्शन (\$टैग, \$फ़ंक्शन_टू_ऐड) • ऐड_एक्शन (\$टैग, \$फ़ंक्शन_टू_ऐड, \$प्राथमिकता) • ऐड_एक्शन (\$टैग, \$फ़ंक्शन_टू_ऐड, \$प्राथमिकता, \$स्वीकृत_आर्ग्स)

पैरामीटर

पैरामीटर	विवरण
\$टैग	(स्ट्रिंग) (आवश्यक) उस क्रिया का नाम जिससे \$function_to_add जुड़ा हुआ है
\$function_to_add	(कॉल करने योग्य) (आवश्यक) वह फ़ंक्शन जिसे \$tag द्वारा इंगित कार्रवाई निष्पादित होने पर कॉल किया जाना चाहिए
\$प्राथमिकता	(int) (वैकल्पिक) डिफ़ॉल्ट मान: 10 उस क्रम को निर्दिष्ट करने के लिए उपयोग किया जाता है जिसमें किसी विशेष कार्रवाई से जुड़े कार्यों को निष्पादित किया जाता है। कम संख्याएँ पहले के निष्पादन से मेल खाती हैं, और समान प्राथमिकता वाले कार्यों को उसी क्रम में निष्पादित किया जाता है जिसमें उन्हें कार्रवाई में जोड़ा गया था।
\$accepted_args	(int) (वैकल्पिक) डिफ़ॉल्ट मान: 1 फ़ंक्शन द्वारा स्वीकार किए जाने वाले तर्कों की संख्या।

टिप्पणियां

Add_action() फ़ंक्शन एक एक्शन हुक बनाता है, जो PHP फ़ंक्शन को एक विशेष एक्शन "टैग" या नाम के साथ जोड़ता है। जब किसी विशिष्ट टैग के साथ do_action() (या do_action_ref_array()) पर कॉल द्वारा कार्रवाई को "ट्रिगर" किया जाता है, तो उस टैग से "जुड़े" सभी फ़ंक्शन निष्पादित किए जाएंगे।

ज्यादातर मामलों में, इस फ़ंक्शन का उपयोग किसी थीम के function.php फ़ाइल या प्लगइन फ़ाइल - या किसी अन्य स्रोत फ़ाइल द्वारा लोड किया जाना चाहिए।

यह फ़ंक्शन प्लगइन एपीआई का एक हिस्सा है

उदाहरण

बेसिक एक्शन हुक

Add_action() का सबसे बुनियादी अनुप्रयोग वर्डप्रेस इंस्टॉलेशन के स्रोत-कोड में एक निश्चित स्थान पर निष्पादित होने के लिए कस्टम कोड जोड़ना है - या तो वर्डप्रेस के कोर द्वारा आपूर्ति की गई क्रियाओं का उपयोग करना, या प्लगइन्स जैसे तृतीय-पक्ष कोड द्वारा बनाए गए कार्यों का उपयोग करना। थीम.

साइट के <head></head> अनुभाग में सामग्री जोड़ने के लिए - एक ऐड <लिंक> मेटा तत्व को इंगित करने के लिए कहें कि साइट की कॉपीराइट जानकारी कहाँ मिल सकती है - add_action() का उपयोग प्रिंट करने वाले फ़ंक्शन को संलग्न करने के लिए किया जा सकता है 'wp_head' क्रिया के लिए उपयुक्त मार्कअप (जो कब "ट्रिगर" होता है

वर्डप्रेस <head> अनुभाग बनाता है):

```
फ़ंक्शन add_copyright_meta_link() {  
    इको( '<link rel='copyright' href=''. get_home_url() . '/copyright'>' );  
}  
  
add_action('wp_head', 'add_copyright_meta_link');
```

एक्शन हुक प्राथमिकता

किसी भी कार्रवाई में कितने भी फ़ंक्शन "जुड़े" हो सकते हैं। कुछ उदाहरणों में किसी हुक किए गए फ़ंक्शन को दूसरों से पहले या बाद में निष्पादित करना महत्वपूर्ण होता है, जहाँ add_action(), \$ प्राथमिकता का तीसरा पैरामीटर चलन में आता है।

यदि \$प्राथमिकता तर्क को छोड़ दिया जाता है, तो फ़ंक्शन 10 की डिफ़ॉल्ट प्राथमिकता के साथ संलग्न किया जाएगा।

जब कार्रवाई "ट्रिगर" होती है, तो "हुक" फ़ंक्शंस को सबसे छोटी \$प्राथमिकता के साथ जोड़े गए फ़ंक्शंस से शुरू करके और सबसे बड़ी \$प्राथमिकता वाले फ़ंक्शंस की ओर प्रगति करते हुए कहा जाएगा। समान प्राथमिकता साझा करने वाले किसी भी हुक किए गए फ़ंक्शन को उसी क्रम में कॉल किया जाएगा जिस क्रम में उन्हें जोड़ा गया था (जिस क्रम में उनके संबंधित add_action() कॉल निष्पादित किए गए थे)।

उदाहरण के लिए, मान लें कि एक तृतीय-पक्ष प्लगइन दैनिक-सौदा पृष्ठ पर आगंतुकों को बाहरी ई-कॉमर्स साइट के संबद्ध लिंक पर अग्रेषित करने के लिए 'template_redirect' क्रिया से जुड़े फ़ंक्शन का उपयोग कर रहा है, लेकिन आप पुनर्निर्देशन चाहेंगे यह केवल लॉग-इन उपयोगकर्ताओं के लिए होगा। लॉग-आउट विज़िटर्स को साइन-इन पृष्ठ पर भेजने के लिए आपको अपने स्वयं के 'टेम्पलेट_रीडायरेक्ट' हुक का उपयोग करने की आवश्यकता होगी। यह निर्धारित करने के बाद कि तृतीय-पक्ष प्लगइन अपने फ़ंक्शन को 10 की डिफ़ॉल्ट \$priority के साथ जोड़ता है, आप यह सुनिश्चित करने के लिए अपने फ़ंक्शन को 9 की प्राथमिकता के साथ जोड़ सकते हैं कि आपका लॉग-इन चेक पहले हो जाए:

```
फ़ंक्शन रीडायरेक्ट_डील_विज़िटर्स_टू_लॉगिन() {  
    यदि (is_page ('दैनिक-सौदा') && !user_is_logged_in()) {  
        wp_redirect(wp_login_url()); बाहर निकलना();  
    }  
}  
  
add_action('template_redirect', 'redirect_deal_visitors_to_login', 9);
```

हुकिंग क्लास और ऑब्जेक्ट मेथड्स टू एक्शन

पीएचपी कक्षाएं कोड संगठन को बेहतर बनाने और नामकरण टकराव को कम करने के लिए शक्तिशाली उपकरण हैं।

किसी न किसी बिंदु पर, किसी क्लास पद्धति के लिए एक्शन हुक कैसे बनाया जाए, यह प्रश्न अनिवार्य रूप से उठता है।

\$function_to_add तर्क को अक्सर फ़ंक्शन के नाम वाली एक स्ट्रिंग के रूप में दिखाया जाता है, हालांकि तर्क का डेटा-प्रकार वास्तव में "कॉल करने योग्य" होता है, जिसे हमारे उद्देश्यों के लिए "किसी फ़ंक्शन या विधि के संदर्भ" के रूप में संक्षेपित किया जा सकता है।

ऐसे कई कॉल करने योग्य प्रारूप हैं जिनका उपयोग कक्षाओं पर विधियों को संदर्भित करने के लिए किया जा सकता है

वस्तुएं. हालाँकि, सभी मामलों में, संदर्भित विधि सार्वजनिक रूप से दृश्यमान होनी चाहिए। कोई विधि तब सार्वजनिक होती है जब उसके पहले सार्वजनिक कीवर्ड लगा हो, या कोई दृश्यता कीवर्ड ही न हो (ऐसी स्थिति में विधि डिफ़ॉल्ट रूप से सार्वजनिक हो जाती है)।

ऑब्जेक्ट मेथड एक्शन हुक

ऑब्जेक्ट विधियों को किसी वर्ग के किसी विशेष उदाहरण पर निष्पादित किया जाता है।

```
कक्षा मेरी_क्लास {
  // कंस्ट्रक्टर
  फ़ंक्शन My_Class() {
    // (तात्कालिकता तर्क)
  }

  // आरंभिकरण फ़ंक्शन
  सार्वजनिक फ़ंक्शन इनिशियलाइज़() {
    // (प्रारंभिकरण तर्क)
  }
}
```

उपरोक्त वर्ग को निम्नानुसार इंस्टेंट करने के बाद,

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

इनिशियलाइज़ () विधि सामान्यतः \$my_class_instance->initialize(); को कॉल करके ऑब्जेक्ट पर लागू की जाएगी। विधि को 'init' वर्डप्रेस एक्शन से जोड़ना उदाहरण के संदर्भ वाले एक सरणी और ऑब्जेक्ट विधि के नाम वाली एक स्ट्रिंग को पास करके किया जाता है:

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

यदि किसी ऑब्जेक्ट विधि में add_action() को कॉल किया जाता है, तो \$this छद्म-चर का भी उपयोग किया जा सकता है:

```
कक्षा मेरी_क्लास {
  // कंस्ट्रक्टर
  फ़ंक्शन My_Class() {
    // (इंस्टेंटिएशन लॉजिक) add_action('init',
    [ $this, 'initialize' ] );
  }

  // आरंभिकरण फ़ंक्शन
  सार्वजनिक फ़ंक्शन इनिशियलाइज़() {
    // (प्रारंभिकरण तर्क)
  }
}
```

क्लास मेथड एक्शन हुक

क्लास विधियों को किसी विशेष उदाहरण के बजाय क्लास पर स्थिर रूप से निष्पादित किया जाता है। देखते हुए

निम्नलिखित वर्ग,

```
कक्षा मेरी_क्लास {
    // इनिशियलाइज़ेशन फ़ंक्शन सार्वजनिक स्टैटिक फ़ंक्शन
    इनिशियलाइज़() {
        // (प्रारंभिकरण तर्क)
    }
}
```

इनिशियलाइज़ () विधि को आम तौर पर :: स्कोप-रिज़ॉल्यूशन ऑपरेटर का उपयोग करके लागू किया जाएगा
My_Class::प्रारंभिक());। किसी स्थिर वर्ग विधि को वर्डप्रेस से जोड़ना कुछ अलग-अलग तरीकों से किया जा सकता है:

- क्लास नाम वाली स्ट्रिंग और विधि नाम वाली स्ट्रिंग से बनी सरणी का उपयोग करना:

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

- :: ऑपरेटर सहित विधि के पूर्ण संदर्भ वाली एक स्ट्रिंग पास करना :

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

- यदि add_action() को एक स्थिर वर्ग विधि के भीतर बुलाया जाता है, तो वर्ग नाम के स्थान पर self कीवर्ड या __CLASS__ जादुई स्थिरांक का उपयोग किया जा सकता है। ध्यान दें कि यह आम तौर पर अनुचित है क्योंकि वर्ग विरासत के मामले में इन वस्तुओं के मूल्य कुछ हद तक प्रति-सहज ज्ञान युक्त हो जाते हैं।

```
कक्षा मेरी_क्लास {
    // सेटअप फ़ंक्शन सार्वजनिक
    स्थैतिक फ़ंक्शन setup_actions() {
        add_action('init', 'self::initialize' );
    }

    // इनिशियलाइज़ेशन फ़ंक्शन सार्वजनिक स्टैटिक फ़ंक्शन
    इनिशियलाइज़() {
        // (प्रारंभिकरण तर्क)
    }
}
```

अध्याय 26: get_bloginfo()

परिचय

वर्तमान साइट के बारे में जानकारी प्राप्त करता है.

वाक्य - विन्यास

• `get_bloginfo ($दिखाएँ , $फ़िल्टर)`

पैरामीटर

पैरामीटर विवरण	
\$दिखाओ	(स्ट्रिंग) पुनः प्राप्त करने के लिए साइट सेटिंग जानकारी।
\$फ़िल्टर	(स्ट्रिंग) फ़िल्टर किए गए मान को लौटाना है या नहीं, इसकी विशिष्टता।

टिप्पणियां

\$दिखाओ

मान	विवरण	उदाहरण
'नाम' (डिफ़ॉल्ट)	घटनास्थल शीर्षक	'मैट मुलेनवेग'
'विवरण'	साइट टैगलाइन	'एक और वर्डप्रेस साइट'
'wpurl'	वर्डप्रेस का यूआरएल स्थापना. के समान साइट_यूआरएल() फ़ंक्शन	'http://example.com' 'http:// localhost/wordpress'
'यूआरएल'	साइट का यूआरएल. के समान होम_यूआरएल () फ़ंक्शन	'http://example.com' 'http:// localhost/wordpress'
'एडमिन_ईमेल'	का ईमेल पता मुख्य प्रशासक	'matt@mullenweg.com'
'चारसेट'	का कैरेक्टर एन्कोडिंग पेज और फ़ीड	'UTF-8'
'संस्करण'	का वर्तमान संस्करण वर्डप्रेस इंस्टालेशन	'4.5'
'html_type'	'टेक्स्ट/एचटीएमएल' का सामग्री-प्रकार मान	

मान	विवरण	उदाहरण
	एचटीएमएल	
'पाठ की दिशा'	पाठ की दिशा साइट द्वारा निर्धारित भाषा	'एल टीआर'
'भाषा'	आईएसओ 639-1 आधारित भाषा कोड	'एन-यूएस'
'स्टाइलशीट_यूआरएल'	की स्टाइलशीट का URL सक्रिय विषय. मूल्य प्राथमिकता: बाल विषय » मूल विषय.	'http://example.com/wp content/themes/ twentysixteen/style.css'
'स्टाइलशीट_निर्देशिका'	का संसाधन स्थान सक्रिय विषय. कीमत प्राथमिकता: बाल विषय » मूल विषय.	'http://example.com/wp content/themes/ twentysixteen'
'टेम्पलेट_यूआरएल'	की यूआरएल निर्देशिका सक्रिय विषय. कीमत प्राथमिकता: मूल विषय » बाल विषय.	'http://example.com/wp content/themes/ twentysixteen'
'template_directory'	'template_url' के समान	
'पिंगबैक_यूआरएल'	पिंगबैक XML-RPC फ़ाइल	'http://example/xmlrpc.php'
'एटम_यूआरएल'	एटम फ़ीड यूआरएल	'http://example/feed/atom/'
'rdf_url'	आरडीएफ/आरएसएस 1.0 फ़ीड यूआरएल	'http://example/feed/rdf/'
'rss_url'	आरएसएस 0.92 फ़ीड यूआरएल	'http://example/feed/rss/'
'rss2_url'	आरएसएस 2.0 फ़ीड यूआरएल	'http://example/feed/'
'comments_atom_url'	टिप्पणियाँ एटम फ़ीड यूआरएल	'http://example/comments/feed/atom/'
'साइट URL'	(अप्रत्याशित) 'यूआरएल' का प्रयोग करें बजाय	
'घर'	(अप्रत्याशित) 'यूआरएल' का प्रयोग करें बजाय	

\$फ़िल्टर

मान	विवरण	उदाहरण
'कच्चा' (डिफ़ॉल्ट)	कोई फ़िल्टर लागू नहीं किया जाएगा	कच्चा डेटा
'दिखाना'	यदि \$show न तो 'url' है तो रिटर्न वैल्यू पर फ़िल्टर लागू किए जाएंगे , 'निर्देशिका' , 'घर'	छाना हुआ आंकड़े

उदाहरण

साइट का शीर्षक प्राप्त करना

```
<?php इको get_bloginfo( 'नाम' ); ?>
```

या

```
<?php इको get_bloginfo(); ?>
```

उत्पादन

```
मेट मुलेनवेग
```

इन नमूना सेटिंग्स के आधार पर

The screenshot shows the WordPress admin 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 options: Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings (highlighted in blue). Under 'Settings', sub-options include General, Writing, Reading, Discussion, Media, and Permalinks. The main content area is titled 'General Settings' and contains the following fields:

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

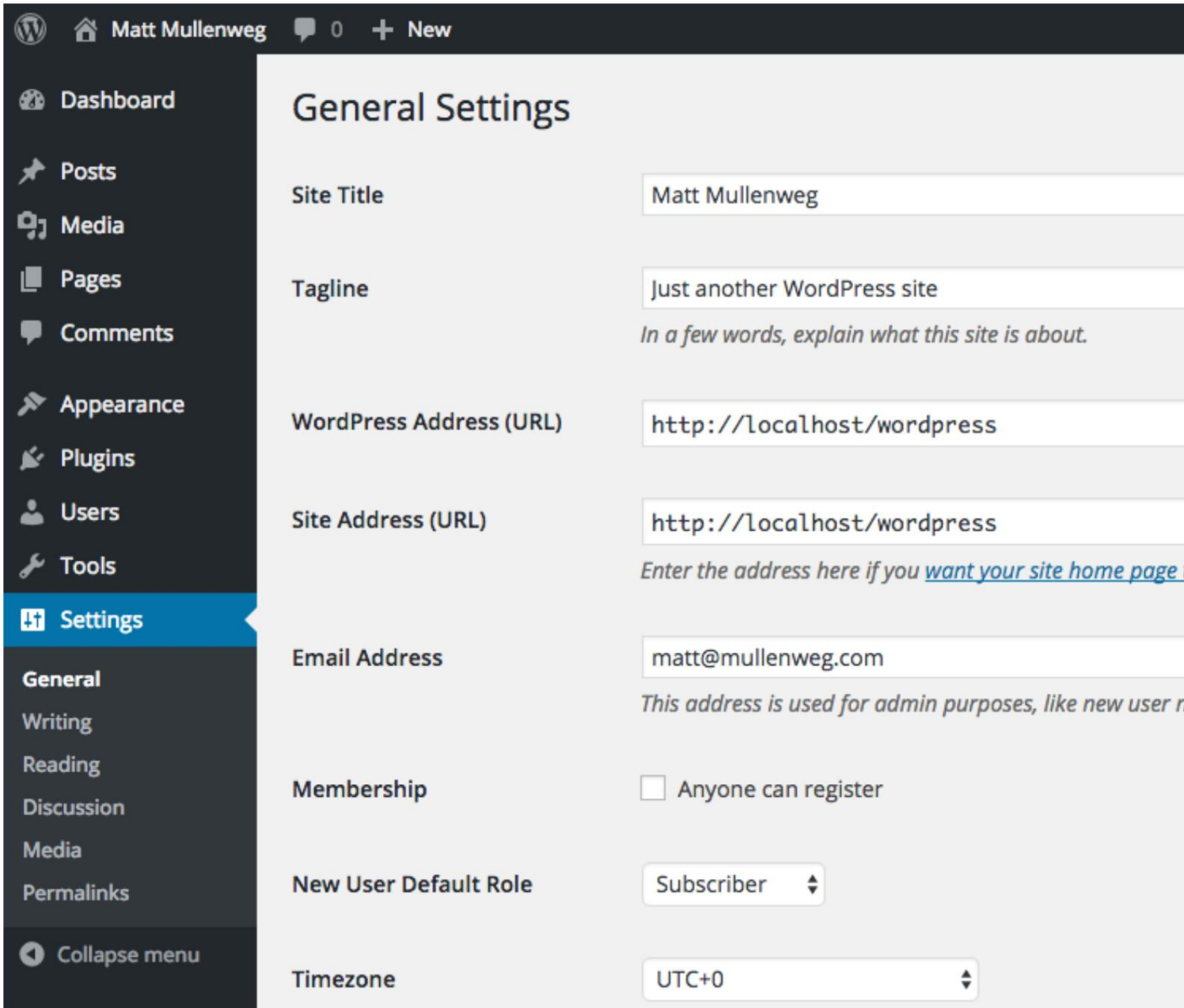
साइट टैगलाइन प्राप्त करना

```
<?php इको get_bloginfo('विवरण'); ?>
```

उत्पादन

एक और वर्डप्रेस साइट

इन नमूना सेटिंग्स के आधार पर



The screenshot shows the WordPress admin dashboard for a user named Matt Mullenweg. The left sidebar contains navigation links for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The Settings menu is expanded to show sub-categories: General, Writing, Reading, Discussion, Media, and Permalinks. The main content area is titled 'General Settings' and contains several configuration options:

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

सक्रिय थीम URL प्राप्त करना

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

उत्पादन

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

वैकल्पिक

आंतरिक रूप से, `get_bloginfo('styleSheet_directory')` `get_styleSheet_directory_uri()` को कॉल करता है, इसलिए हो सकता है कि आप इसके बजाय इसका उपयोग करना चाहें:

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

कई डेवलपर्स इन समर्पित कार्यों का उपयोग करना पसंद करते हैं क्योंकि उनके और `get_bloginfo()` के बीच असंगत नामकरण परंपराएं हैं। उदाहरण के लिए, `get_styleSheet_directory()` चाइल्ड थीम पथ लौटाता है; हालाँकि, जैसा कि हमारा पिछला उदाहरण दिखाता है, `get_bloginfo('styleSheet_directory')` चाइल्ड थीम URL लौटाता है। यदि आप इसके बजाय `get_styleSheet_directory_uri()` का उपयोग करते हैं, तो इस बात पर भ्रम की संभावना कम है कि आप पथ पुनर्प्राप्त कर रहे हैं या URL।

साइट यूआरएल प्राप्त करें

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

या यदि आपको किसी उप पृष्ठ से लिंक करने की आवश्यकता है

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

साइट प्रशासक का ईमेल पता प्राप्त करें

हम साइट व्यवस्थापक का ईमेल पता पुनः प्राप्त करने के लिए `get_bloginfo` फ़ंक्शन का उपयोग कर सकते हैं।

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

अध्याय 27: get_home_path()

परिचय

वर्डप्रेस इंस्टॉलेशन के रूट तक संपूर्ण फ़ाइल सिस्टम पथ प्राप्त करें।

पैरामीटर

पैरामीटर विवरण	
कोई नहीं	यह फ़ंक्शन किसी भी पैरामीटर को स्वीकार नहीं करता है।

टिप्पणियां

के बीच महत्वपूर्ण अंतर

get_home_path()

और ABSPATH

यदि आपके सबफ़ोल्डर में वर्डप्रेस स्थापित है तो कृपया ABSPATH और get_home_path() के बीच अंतर को ध्यान में रखें।

Get_home_path () फ़ंक्शन हमेशा सबफ़ोल्डर के बिना एक पथ लौटाएगा :

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

इस प्रकार यह ABSPATH से भिन्न है, जो भिन्न मान लौटाएगा:

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

ABSPATH को सबसे पहले wp-load.php में परिभाषित किया गया है जो /var/www/htdocs/example/wp/wpload.php पर स्थित होगा, इसलिए यहीं से ABSPATH अपनी परिभाषा लेगा।

get_home_path() जाँचता है कि क्या site_url और Home_url भिन्न हैं, और पथ से सबस्ट्रिंग हटा देता है। अन्यथा यह ABSPATH मान लौटाता है:

```
फ़ंक्शन get_home_path() { $home =
    set_url_scheme( get_option( 'home' ), 'http' ); $siteurl = set_url_scheme( get_option( 'siteurl' ), 'http' );
    यदि ( ! खाली( $home ) && 0 !== strcasecmp( $home, $siteurl )
        ) {
        $wp_path_rel_to_home = str_ireplace( $home, '', $siteurl ); /* $siteurl - $home */ $pos = strrpos( str_replace( '\\', '/',
            $_SERVER['SCRIPT_FILENAME'] ),
            $wp_path_rel_to_home );
        ट्रैलिंगस्लैश( $wp_path_rel_to_home );
        $home_path = पदार्थ( $_SERVER['SCRIPT_FILENAME'], 0, $pos ); $home_path = ट्रैलिंगस्लैश( $home_path );
        अन्यथा { $home_path = ABSPATH;
    }
}
```

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

इसे अपने कोड में उपयोग करना

`get_home_path()` को कॉल करना उस संदर्भ में किया जाना चाहिए जहां `wp-admin/includes/file.php` पहले से ही शामिल किया गया है।

उदाहरण के लिए `admin_init` हुक के भीतर `get_home_path()` का उपयोग करना ठीक है, लेकिन `init` के भीतर इसका उपयोग करना ठीक नहीं है और परिणामस्वरूप PHP में घातक त्रुटि होगी:

```
अपरिभाषित फ़ंक्शन get_home_path() पर कॉल करें
```

यह फ़ाइल केवल व्यवस्थापक (डैशबोर्ड) संदर्भ के भीतर से शामिल की जाती है, यदि आपको इस संदर्भ के बाहर इसकी बिल्कुल आवश्यकता है तो आपको फ़ंक्शन को कॉल करने से पहले फ़ाइल को स्वयं शामिल करना होगा:

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

उदाहरण

प्रयोग

```
$पथ = get_home_path();
```

प्रतिलाभ की मात्रा:

डोरी

वर्डप्रेस इंस्टॉलेशन के रूट तक पूर्ण फ़ाइल सिस्टम पथ, भले ही यह सबफ़ोल्डर में इंस्टॉल हो।

उदाहरण:

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

अध्याय 28: get_option()

परिचय

विकल्प नाम के आधार पर विकल्प मान पुनर्प्राप्त करता है।

वाक्य - विन्यास

- `get_option ($ विकल्प, $ डिफ़ॉल्ट)`

पैरामीटर

पैरामीटर विवरण	
\$विकल्प	(स्ट्रिंग) पुनर्प्राप्त करने के विकल्प का नाम। एसक्यूएल-एस्केड नहीं होने की उम्मीद है।
\$डिफ़ॉल्ट	(मिश्रित) (वैकल्पिक) यदि विकल्प मौजूद नहीं है तो डिफ़ॉल्ट मान लौटाया जाएगा।

टिप्पणियां

\$option के लिए तर्कों की सूची

- 'एडमिन_ईमेल' • 'ब्लॉग नाम' • 'ब्लॉग विवरण'
- 'ब्लॉग_चारसेट' • 'दिनांक_प्रारूप' • 'डिफ़ॉल्ट_श्रेणी' • 'होम'
- 'साइटयूआरएल' • 'टेम्पलेट' •

- 'स्टार्ट_ऑफ_वीक'
- 'अपलोड_पथ' • 'उपयोगकर्ता_रजिस्टर_कर सकते हैं' • 'पोस्ट_प्रति_पेज' • 'पोस्ट_प्रति_आरएसएस'

उदाहरण

ब्लॉग शीर्षक दिखाएँ

कोड

```
<h1><?php इको get_option('ब्लॉगनाम' ); ?></h1>
```

उत्पादन

ब्लॉग का नाम H1 शैली में

कैरेक्टर सेट दिखाएँ

कोड

```
<p><?php इको esc_html(sprintf(__('कैरेक्टर सेट: %s', 'textdomain' ), get_option('blog_charset' )
)); ?></p>
```

उत्पादन

कैरेक्टर सेट: UTF-8

गैर-मौजूदा विकल्पों को संभालना

कोड

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

उत्पादन

असत्य

कोड

```
<?php इको get_option('something_bla_bla_bla', 'अरे नहीं!' ); ?>
```

उत्पादन

ओह तेरी!

अध्याय 29: get_permalink()

परिचय

यह फ़ंक्शन वर्तमान पोस्ट या निर्दिष्ट पोस्ट का पूरा पैरालिंक लौटाता है।

वाक्य - विन्यास

- get_permalink (\$पोस्ट, \$leavename)

पैरामीटर

पैरामीटर विवरण	
\$पोस्ट	(int) (वैकल्पिक) पोस्ट आईडी या पोस्ट ऑब्जेक्ट। डिफ़ॉल्ट वर्तमान पोस्ट की आईडी है।
\$leavename (बूल)	(वैकल्पिक) क्या पोस्ट का नाम या पेज का नाम रखना है।

टिप्पणियां

पैरामीटर \$leavename के लिए, यह डिफ़ॉल्ट रूप से गलत है।

उदाहरण

Get_permalink का सरल उपयोग

कोड

```
इको get_permalink();
```

उत्पादन

वर्तमान पृष्ठ का लिंक, उदाहरण के लिए: <http://website.com/category/name-of-post/>

लिंक पाने के लिए पोस्ट निर्दिष्ट करना

कोड

```
इको get_permalink(44);
```

उत्पादन

पोस्ट का लिंक आईडी:44, उदाहरण के लिए: <http://website.com/category/name-of-post/>

पोस्ट के नाम के बिना लिंक प्राप्त करें

कोड

```
इको get_permalink(44 , असत्य );
```

उत्पादन

उदाहरण के लिए, पोस्ट का लिंक आईडी:44 बिना पोस्ट के नाम के:

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

अध्याय 30: get_template_part()

वाक्य - विन्यास

- `get_template_part($स्लग, $नाम);`
- `get_template_part($slug);`

पैरामीटर

पैरामीटर विवरण	
\$स्लग	(स्ट्रिंग) जेनेरिक टेम्पलेट स्लग नाम
\$नाम	(स्ट्रिंग) विशिष्ट टेम्पलेट नाम

उदाहरण

एक सबफ़ोल्डर से एक टेम्पलेट भाग लोड करें

```
get_template_part('फू/बार', 'पेज');
```

निर्देशिका 'foo' से 'bar-page.php' की आवश्यकता होगी।

एक विशिष्ट फ़ाइल प्राप्त करें

आप इस फ़ंक्शन का उपयोग करके विशिष्ट फ़ाइल प्राप्त कर सकते हैं।

```
get_template_part('टेम्पलेट-पार्ट्स/लेआउट');
```

अपने थीम फ़ोल्डर के मूल में रखे गए टेम्पलेट-पार्ट्स उपनिर्देशिका से लेआउट.php फ़ाइल शामिल करें।

अध्याय 31: get_template_part()

परिचय

इस फ़ंक्शन का उद्देश्य किसी थीम के आंशिक या घटकों को मुख्य थीम टेम्पलेट में आयात करने के तरीके को मानकीकृत करना है। आप एक मानक PHP SSI (सर्वर साइड शामिल) का उपयोग कर सकते हैं, हालाँकि, get_template_part() का उपयोग करने के कुछ लाभ हैं। इस फ़ंक्शन का उपयोग करने से सर्वर पर पूरी तरह से योग्य पथ की पहचान करने की कोशिश करने वाले कम अनुभवी डेवलपर्स द्वारा की जाने वाली त्रुटियों की संभावना कम हो जाती है। साथ ही, जब फ़ाइलें मौजूद नहीं होती हैं तो यह शानदार ढंग से विफल हो जाता है, और एक कस्टम पदानुक्रम फ़ॉलबैक सिस्टम उर्फ "फ़ज़ी टेम्पलेट सर्चिंग" को संभालता है।

वाक्य - विन्यास

- get_template_part(\$slug) •
- get_template_part(\$slug, \$name)

पैरामीटर

पैरामीटर विवरण	
\$स्लग	(स्ट्रिंग) कस्टम टेम्पलेट का स्लग नाम।
\$नाम	(स्ट्रिंग) विशेष टेम्पलेट का नाम. वैकल्पिक

उदाहरण

एक कस्टम टेम्पलेट सहित

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

शामिल

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

डैश-पृथक फ़ाइल नाम के साथ एक कस्टम टेम्पलेट शामिल करना

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

शामिल

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

एक निर्देशिका के अंदर से एक कस्टम टेम्पलेट शामिल करना

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

शामिल

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

एक निर्देशिका के अंदर स्थित डैश-पृथक फ़ाइल नाम के साथ एक कस्टम टेम्पलेट शामिल करना

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

शामिल

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

वैरिएबल को कस्टम टेम्पलेट स्कोप में पास करना

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

इसे foo-bar.php में एक्सेस करें

```
<?php इको $passed_var; ?>
```

अध्याय 32: get_template_part()

वाक्य - विन्यास

- `get_template_part('फ़ाइल-नाम-नो-एक्सटेंशन');`

पैरामीटर

पैरामीटर	विवरण
फ़ाइल-नाम-कोई एक्सटेंशन नहीं	बिना किसी एक्सटेंशन वाले टेम्पलेट भाग का नाम. उदाहरण के लिए 'foo.php' के बजाय 'foo'

उदाहरण

टेम्पलेट भाग लोड हो रहा है

एक निश्चित निर्दिष्ट फ़ाइल से कोड को दूसरी फ़ाइल में खींचता है जहां कॉल किया गया था।

उदाहरण के लिए example.php के अंदर

```
<h1>हेलो वर्ल्ड!</h1>
```

पेज के अंदर.php

```
// हेडर कोड  
get_template_part('example'); // शेष पृष्ठ कोड
```

आउटपुट:

```
// हेडर कोड <h1>हेलो वर्ल्ड!</  
h1> // बाकी पेज कोड
```

अध्याय 33: get_the_category()

परिचय

यह फ़ंक्शन सभी श्रेणियों को वर्तमान पोस्ट या पेज या निर्दिष्ट पोस्ट या पेज की एक सरणी के रूप में लौटाता है।

वाक्य - विन्यास

- श्रेणी प्राप्त करें (\$ आईडी)

पैरामीटर

पैरामीटर विवरण	
\$id	(int) (वैकल्पिक) वर्तमान पोस्ट आईडी पर डिफ़ॉल्ट। पोस्ट आईडी.

टिप्पणियां

कृपया ध्यान दें कि get_the_category() एक सरणी लौटाता है, जिसका अर्थ है कि आप सीधे लौटाए गए मान को प्रतिध्वनित नहीं कर सकते।

यहां प्रत्येक श्रेणी की वस्तुओं की एक सूची दी गई है जिन्हें आप प्रिंट कर सकते हैं:

- टर्म_आईडी
- नाम
- स्लग •
- टर्म_ग्रुप •
- टर्म_टैक्सोनॉमी_आईडी •
- टैक्सोनॉमी •
- विवरण • पैरेंट •
- गिनती

- ऑब्जेक्ट_आईडी
- फ़िल्टर
-
- बिल्ली_आईडी • श्रेणी_गणना
- श्रेणी_विवरण • बिल्ली_नाम •
- श्रेणी_अच्छानाम •
- श्रेणी_अभिभावक

उदाहरण

पोस्ट की सभी श्रेणियों के नाम प्राप्त करें

कोड

```
$श्रेणियाँ = get_the_category();  
foreach($श्रेणियाँ $श्रेणी के रूप में) {  
    प्रतिध्वनि $श्रेणी->नाम। '<br />';  
}
```

उत्पादन

वर्तमान पृष्ठ की श्रेणियों के सभी नाम, प्रत्येक पंक्ति पर एक।

पोस्ट की श्रेणियों की सभी आईडी प्राप्त करें

कोड

```
$श्रेणियाँ = get_the_category();  
foreach($श्रेणियाँ $श्रेणी के रूप में) {  
    प्रतिध्वनि $श्रेणी->term_id। '<br />';  
}
```

उत्पादन

वर्तमान पृष्ठ की श्रेणियों की सभी आईडी, प्रत्येक पंक्ति पर एक।

अध्याय 34: get_the_title()

परिचय

यह फ़ंक्शन वर्तमान पोस्ट या निर्दिष्ट पोस्ट का शीर्षक लौटाता है।

वाक्य - विन्यास

- शीर्षक_प्राप्त करें (\$पोस्ट)

पैरामीटर

पैरामीटर विवरण	
\$पोस्ट	(int) (वैकल्पिक) पोस्ट आईडी या पोस्ट ऑब्जेक्ट। डिफ़ॉल्ट वर्तमान पोस्ट की आईडी है।

टिप्पणियां

यदि आप पोस्ट लूप का उपयोग करके किसी पोस्ट या पेज का शीर्षक प्राप्त करने की योजना बना रहे हैं, तो इसके बजाय the_title() का उपयोग करने का सुझाव दिया जाता है।

उदाहरण

Get_the_title का सरल उपयोग

कोड

```
get_the_title();
```

उत्पादन

वर्तमान पोस्ट या पेज का शीर्षक

किसी निर्दिष्ट पोस्ट आईडी का शीर्षक प्राप्त करें

कोड

```
शीर्षक_प्राप्त करें( 44 );
```

उत्पादन

पोस्ट का शीर्षक आईडी:44.

अध्याय 35: होम_यूआरएल()

वाक्य - विन्यास

- होम_यूआरएल (\$ पथ, \$ योजना);

पैरामीटर

पैरामीटर विवरण	
\$पथ	(स्ट्रिंग, वैकल्पिक) होम यूआरएल के बाद और अधिक सेगमेंट जोड़ने के लिए।
\$योजना	(स्ट्रिंग, वैकल्पिक) होम यूआरएल संदर्भ देने की योजना। 'http', 'https', 'सापेक्ष', 'बाकी', या शून्य स्वीकार करता है।

उदाहरण

होम यूआरएल प्राप्त करना

होम_यूआरएल() वर्तमान साइट होम यूआरएल को पुनः प्राप्त करने के लिए उपयोग किया जाता है।

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

उत्पादन

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

साइट URL

उपयुक्त प्रोटोकॉल (https://) के साथ 'site_url' विकल्प लौटाता है

```
<?php इको साइट_यूआरएल(); ?>
```

उत्पादन

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

अध्याय 36: मैं मार्कडाउन संपादक को एडवांस कस्टम फील्ड के रिपीटर ऐड-ऑन के साथ कैसे एकीकृत कर सकता हूँ।

उदाहरण

मार्कडाउन संपादक जोड़ें

मुझे समाधान मिल गया. कृपया नीचे दिए गए चरणों पर विचार करें।

[wp मार्कडाउन संपादक स्थापित करें लगाना।](#)

फिर "acf-wp-wysiwyg" इंस्टॉल करें पुनरावर्तक फ़ील्ड के लिए. अब आपको कुछ फ़ाइलों में अपडेट करना होगा। इस फ़ाइल को खोलें और लाइन नंबर "180" पर जाएँ या "create_field" फ़ंक्शन ऐड पर जाएँ

```
इको '<स्क्रिप्ट> var Simplemde = new SimpleMDE({element: document.getElementById(".$id."});jQuery("Iquicktags टूलबार").css("display","none"); </स्क्रिप्ट>';
```

अब "एसीएफ-रिपीटर" प्लगइन के तहत "इनपुट.जेएस" फ़ाइल खोलें, लाइन नंबर "142"

बदलना

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

साथ

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

अध्याय 37: init

वाक्य - विन्यास

1. `add_action('init', कॉल करने योग्य $function)`

टिप्पणियां

`init` एक एक्शन हुक है जो वर्डप्रेस लोड होने के बाद लेकिन किसी भी HTTP हेडर भेजे जाने से पहले सक्रिय हो जाता है।

उदाहरण

`$_POST` अनुरोध डेटा संसाधित किया जा रहा है

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

```
फ़ंक्शन प्रोसेस_पोस्ट_डेटा() { if(isset( $_POST ) // यहां $_POST
    डेटा प्रोसेस करें           ) {
    }
}
```

`$_GET` अनुरोध डेटा संसाधित किया जा रहा है

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

```
फ़ंक्शन प्रोसेस_गेट_डेटा() { if(isset( $_GET ) // प्रोसेस $_GET
    डेटा यहां           ) {
    }
}
```

एक कस्टम पोस्ट प्रकार पंजीकृत करना

```
add_action('init', function() {
    रजिस्टर_पोस्ट_प्रकार('इवेंट', सरणी('सार्वजनिक' => सत्य,
        'लेबल' => 'घटनाएं'
    ));
});
```

इवेंट लेबल और स्लग इवेंट के साथ एक नया कस्टम पोस्ट प्रकार पंजीकृत करता है

अध्याय 38: स्थापना और विन्यास

उदाहरण

लैंप पर वर्डप्रेस

मैंने Amazon EC2 पर Ubuntu 14.04 के साथ निम्नलिखित चरणों का परीक्षण किया है

यहां LAMP प्रौद्योगिकी स्टैक स्थापित करने के लिए कमांड लाइन का सारांश दिया गया है (वर्डप्रेस स्थापित करने से पहले):

1: लैंप प्रौद्योगिकी स्टैक स्थापित करें

लिनक्स अपडेट करें

```
#> sudo apt-get update -> अपडेट पैकेज रिपॉजिटरी जानकारी
```

```
#> sudo apt-get अपग्रेड -> पैकेज अपग्रेड इंस्टॉल करें
```

अपाचे2 स्थापित करें

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

php इंस्टॉल करें

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

MySQL इंस्टॉल करें

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

phpmyadmin इंस्टॉल करें

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

यदि कोई चीज़ गलत हो जाती है, तो इंस्टॉल किए गए पैकेज और उसकी कॉन्फिगरेशन फ़ाइल को हटा दें

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

एक बार स्टैक इंस्टॉल हो जाने के बाद वर्डप्रेस इंस्टॉलेशन को पूरा करने के चरण यहां दिए गए हैं:

2: वर्डप्रेस इंस्टॉल करें

1. `wget https://wordpress.org/latest.zip`

2. `sudo mkdir /var/www/wordpress`

3. ज़िप फ़ाइल को `/var/www/wordpress` में ले जाएँ और निकालें।

4. सीडी `/etc/apache2/साइट्स-उपलब्ध/`

5. `sudo cp 000-default.conf WordPress.conf`

Wordpress.conf को संशोधित करें ताकि Apache2 किसी भी अनुरोध या आपके डोमेन को वर्डप्रेस ऐप फ़ोल्डर में अग्रेषित करना जान सके।

7. sudo a2ensite WordPress.conf

8. sudo सेवा apache2 पुनरारंभ करें

9. youdomain.com/phpmyadmin द्वारा अपने ब्राउज़र के माध्यम से PhpMyAdmin पर जाएं। एक नया उपयोगकर्ता "वर्डप्रेस" बनाएं और संबंधित डेटाबेस बनाने के लिए जांच करें। सीडी /var/www/वर्डप्रेस

10. सुडो सीपी wp-config-example.php wp-config.php

11. अपनी MySQL वर्डप्रेस डेटाबेस जानकारी जोड़ने के लिए कॉन्फिगरेशन फ़ाइल को संशोधित करें।

12. sudo chown -R www-data:www-data /var/www/wordpress

13. sudo chmod -R 755 /var/www/wordpress/

14. ब्राउज़र खोलें और अपना डोमेन टाइप करें। आपको वर्डप्रेस इंस्टालेशन पेज देखना चाहिए। निर्देशों का पालन करें और आपका काम हो गया!

MAMP में स्थापना WP

MAMP में वर्डप्रेस इंस्टॉल करना काफी सरल है।

आपको कुछ सरल चरणों का पालन करना होगा:

1 - यहां से एमएमपी डाउनलोड करें, यह मुफ्त है और संभवतः आपको प्रो संस्करण की आवश्यकता नहीं है।

2 - अपने पीसी या मैक पर इंस्टॉल करें।

3 - प्रोग्राम चलाएँ -> आपको एक छोटी सी विंडो खुली हुई दिखाई देगी, वहाँ से आप MAMP सेट कर सकते हैं। फिलहाल सभी पूर्व-निर्धारित मूल्यों को छोड़ दें, पहली स्थापना के लिए आपको अपने जीवन को जटिल बनाने की आवश्यकता नहीं है!

भविष्य में याद रखें कि MySQL डेटाबेस के लिए अपना पासवर्ड और उपयोगकर्ता नाम बदलना अच्छा अभ्यास है। डिफ़ॉल्ट रूप से यह रूट है।

4 - "ओपन वेबस्टार्ट पेज" पर क्लिक करें - यहां आप अपने डेटा की जानकारी जैसे पासवर्ड, एडमिन का नाम और एमएमपी के बारे में भी जानकारी देख सकते हैं।

5 - टूलस -> phpMyAdmin पर क्लिक करें और आपको डेटाबेस पेज पर पुनः निर्देशित किया जाएगा।

6 - एक नया डेटाबेस बनाएं, न्यू पर क्लिक करें और उसे अपना मनचाहा नाम दें, इस नाम की आपको बाद में जरूरत पड़ेगी।

7 - अब "htdocs" नाम का एक फ़ोल्डर ढूँढें, यह आपके पीसी या मैक पर MAMP फ़ोल्डर के अंदर है। यदि आप मैक का उपयोग करते हैं तो आपको फाइंडर में एक एप्लिकेशन की आवश्यकता होगी और "पैकेज सामग्री दिखाएं" का उपयोग करके ऐप खोलें।

अंदर आपको htdocs फ़ोल्डर मिलेगा।

8 - वर्डप्रेस फ़ोल्डर लें और इसे htdocs के अंदर कॉपी करें, आपको सभी फ़ोल्डर के अंदर डालना होगा, ज़िप फ़ाइल नहीं डालनी होगी। फ़ोल्डर का नाम बदलें, आप अपने प्रोजेक्ट नाम से कॉल कर सकते हैं।

9 - अपने ब्राउज़र में एमएमपी की विंडो पर वापस आएं, और "माई साइट" पर क्लिक करें - इससे एक खुल जाएगा

यूआरएल कॉल <http://localhost:8888> (8888 डिफ़ॉल्ट पोर्ट है, आप चाहें तो इसे बदल सकते हैं)। यहां आप वह फोल्डर देख सकते हैं जिसे आपने htdocs फोल्डर के अंदर रखा है, आपने फोल्डर को जो नाम दिया है उस पर क्लिक करें।

10 - अब वर्डप्रेस की सामान्य स्थापना शुरू करें, आपको एडमिन और पासवर्ड का उपयोग करना होगा। डिफ़ॉल्ट रूप से यह रूट और रूट है, और आपके द्वारा पहले बनाए गए डेटाबेस के नाम का उपयोग करें।

11 - इंस्टालेशन चलाएँ और समाप्त करें!

आप अपनी वेबसाइट <http://localhost:8888/namefolder> पर देखेंगे, बेशक आपको MAMP चालू रखना होगा।

अध्याय 39: नेटवर्क अनुरोध करना

HTTP एपीआई

वाक्य - विन्यास

- \$प्रतिक्रिया = wp_remote_get(\$url, \$args);
- \$प्रतिक्रिया = wp_remote_post(\$url, \$args);
- \$प्रतिक्रिया = wp_safe_remote_post(\$url, \$args);

पैरामीटर

पैरामीटर विवरण	
\$यूआरएल	(स्ट्रिंग) (आवश्यक) पुनः प्राप्त करने के लिए साइट यूआरएल।
\$args	(सरणी) (वैकल्पिक) तर्क का अनुरोध करें।

टिप्पणियां

रिटर्न

(WP_Error | array) एक सरणी के रूप में प्रतिक्रिया, या विफलता पर WP_Error।

उदाहरण

एक दूरस्थ JSON संसाधन प्राप्त करें

यह स्निपेट JSON स्वरूपित संसाधन को लेगा, इसे डीकोड करेगा और इसे PHP सरणी प्रारूप में प्रिंट करेगा।

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

अगर ( ! is_wp_error($प्रतिक्रिया) ) {
    $हेडर = wp_remote_retrieve_headers($प्रतिक्रिया);

    यदि ( जारीकर्ता ( $ हेडर [ 'सामग्री-प्रकार' ] ) && 'एप्लिकेशन/जेसन' === $हेडर[ 'सामग्री-प्रकार' ] ) {
        print_r(json_decode(wp_remote_retrieve_body($प्रतिक्रिया)));
    }
}
```

अध्याय 40: मेटा बॉक्स

परिचय

WP-एडमिन सामग्री संपादकों में मेटा बॉक्स का सरल उपयोग

वाक्य - विन्यास

- `_x('टेक्स्ट', 'विवरण', 'टेक्स्टडोमेन')` का उपयोग `__('टेक्स्ट', 'टेक्स्टडोमेन')` के बजाय अनुवाद सेवा के लिए विवरण जोड़ने के लिए किया जाता है जो कि सिर्फ अनुवाद है
- `_ex('टेक्स्ट', 'विवरण', 'टेक्स्टडोमेन')` का उपयोग विवरण के साथ अनुवादित पाठ को प्रतिध्वनित करने के लिए किया जाता है

टिप्पणियां

रेंडर मेटा बॉक्स के अंदर की सामग्री कुछ भी हो सकती है। मानों को सीधे एकीकृत करने के बजाय, आप PHP टेम्पलेट के साथ शामिल का भी उपयोग कर सकते हैं और इसमें डेटा पास करने के लिए `set_query_var` विधि का उपयोग कर सकते हैं। सेव उसी तरह काम करेगा।

उदाहरण

नियमित इनपुट और चयनित इनपुट के साथ एक सरल उदाहरण

```
/**
 * पोस्ट प्रकारों में मेटा बॉक्स जोड़ें।
 *
 * @1.0.0 से
 */
फ़ंक्शन myplugin_add_meta_box() {
    // डिफ़ॉल्ट पोस्ट प्रकार सेट करें / $ प्रकार = सरणी ('पोस्ट',

);

// मेटा बॉक्स को अधिक प्रकारों में जोड़ने के लिए वैकल्पिक फ़िल्टर। उपयोग करने के लिए टिप्पणी रद्द करें. // $ प्रकार = लागू फ़िल्टर ('myplugin_meta_box_types', $
प्रकार);

// पेज पर मेटा बॉक्स जोड़ें add_meta_box(

    'मायप्लगिन-मेटा-बॉक्स', // मेटा बॉक्स आईडी। कुछ भी हो सकता है. _x( 'कस्टम मेटा', 'कस्टम मेटा बॉक्स',
    'मायप्लगिन' ), // मेटा बॉक्स का शीर्षक।
    अनुवाद वैकल्पिक है. बस स्ट्रिंग हो सकता है.
    'myplugin_render_meta_box', // रेंडर मेटा बॉक्स फ़ंक्शन। $types, // वे पोस्ट प्रकार जोड़ें जिनमें मेटा बॉक्स जोड़ना है। 'पक्ष', //
    संपादन के पक्ष में दिखाएँ। 'उच्च' //संपादन के शीर्ष पर दिखाएँ।

);
}

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

```

/**
 * मेटा बॉक्स प्रस्तुत करें.
 *
 * यह मेटा बॉक्स के अंदर बुनियादी इनपुट और चयन के उदाहरण दिखाता है। ये कुछ भी हो सकते हैं.
 *
 * @1.0.0 से
 *
 * @param $post WP_Post पोस्ट संपादित की जा रही है। */ फ़ंक्शन myplugin_render_meta_box ($ पोस्ट)
    {
        // हमारे कस्टम मेटा बॉक्स के लिए वर्तमान पोस्ट मेटा मान प्राप्त करें। $शहर = get_post_meta($पोस्ट->आईडी, 'शहर', सच); // टू सिंगल लौटाने के लिए है
        कीमत।
        $देश = get_post_meta($पोस्ट->आईडी, 'देश', सत्य); // टू सिंगल लौटाने के लिए है
        कीमत।
        // सुरक्षा के लिए WP नॉन्स फ़िल्ड जोड़ें। wp_nonce_field(प्लगइन_बेसनाम(__FILE__),
        'myplugin_meta_nonce');
        ?>
        <पी>
        <label for='city' ><?php _ex('City',
        'Custom Meta Box Template', 'myplugin') ; ?> </लेबल> <इनपुट नाम = "शहर" आईडी = "शहर" मूल्य = "<?php इको $ शहर; ?>" /> </p> <p>
        <लेबल for='देश' ><?php _ex( 'देश', 'कस्टम मेटा
        बॉक्स टेम्पलेट', 'मायप्लगिन' ); ?> </लेबल> <select name='country' id='country'>
        <विकल्प मान = "संयुक्त राज्य अमेरिका" <?php चयनित ($ देश, 'संयुक्त राज्य अमेरिका'); ?>><?php
        _ex('संयुक्त राज्य अमेरिका', 'कस्टम मेटा बॉक्स टेम्पलेट', 'मायप्लगिन') ; ?></विकल्प>
        <विकल्प मान='मेक्सिको' <?php चयनित ($देश, 'मेक्सिको') ; ?>><?php _ex('मेक्सिको',
        'कस्टम मेटा बॉक्स टेम्पलेट', 'मायप्लगिन') ; ?></विकल्प>
        <विकल्प मान='कनाडा' <?php चयनित ($देश, 'कनाडा' ); ?>><?php _ex( 'कनाडा',
        'कस्टम मेटा बॉक्स टेम्पलेट', 'मायप्लगिन') ; ?></option> </select> </p>
        <?php
        }
        /**
        * मेटा बॉक्स डेटा सहेजें।
        *
        * @1.0.0 से
        *
        * @param $post_id int पोस्ट की आईडी सहेजी जा रही है। */ फ़ंक्शन myplugin_save_meta_data( $post_id )
            {
                // सत्यापित करें कि यह ऑटो सेव नहीं है। यदि (परिभाषित('DOING_AUTOSAVE') &&
                DOING_AUTOSAVE )
                    {
                        वापस करना;
                    }
                // सुरक्षा के लिए मेटा बॉक्स को एक बार मान्य करें। अगर (
                ! जारी($_POST['myplugin_meta_nonce'] ) || ! wp_verify_nonce(
                $_POST['myplugin_meta_nonce'], प्लगइन_बेसनाम(__FILE_)
                ) ) {
                    वापस करना;
                }
            }
    }

```

```

// फॉर्म से नए मान प्राप्त करें।
$शहर = $_POST['शहर'];
$देश = $_POST['देश'];

// update_post_meta यदि मौजूद नहीं है तो मान जोड़ देगा या यदि मौजूद है तो इसे अपडेट कर देगा।
अपडेट_पोस्ट_मेटा ($पोस्ट_आईडी, 'शहर', $शहर);
अपडेट_पोस्ट_मेटा ($पोस्ट_आईडी, 'देश', $देश);

/*
*वैकल्पिक विकल्प
*
* केवल update_post_meta का उपयोग करने के बजाय, आप मानों की भी जांच कर सकते हैं
* पोस्ट मेटा वैल्यू पर क्रिएट/अपडेट/डिलीट जारी करें।
*/
// $current_city_value = get_post_meta($post_id, 'शहर', सत्य); //सत्य लौटने के लिए है
एक एकल मान.
//
// // यदि पोस्ट मेटा मौजूद नहीं है तो उसे जोड़ें।
// अगर ($ शहर && === $ शहर ) // add_post_meta ( $
post_id, 'शहर', $ शहर, सच); // सही का मतलब है कि कुंजी अद्वितीय है
पोस्ट। गलत डिफॉल्ट है और एक से अधिक जोड़े जा सकते हैं।
// }
// // यदि पोस्ट मेटा मौजूद है और कोई नया मान है तो उसे संपादित करें।
// अन्यथा ($ शहर और $ शहर! = $ वर्तमान_शहर_मूल्य) // अद्यतन_पोस्ट_मेटा ($ पोस्ट_आईडी, 'शहर', $ शहर); {

// }
// // यदि कोई नया मान नहीं है तो पोस्ट मेटा हटा दें।
// अन्यथा (' //
=== $शहर && $current_city_value ) {
delete_post_meta( $post_id, 'city', $current_city_value ); // $current_city_value
वैकल्पिक है और इसका उपयोग समान कुंजी के साथ अन्य मानों के बीच अंतर करने के लिए किया जाता है।
// }
}

add_action('save_post', 'myplugin_save_meta_data');

```

अध्याय 41: विकल्प एपीआई

परिचय

विकल्प डेटा के टुकड़े हैं जिनका उपयोग वर्डप्रेस विभिन्न प्राथमिकताओं और कॉन्फिगरेशन सेटिंग्स को संग्रहीत करने के लिए करता है। विकल्प एपीआई डेटाबेस में डेटा संग्रहीत करने का एक सरल और मानकीकृत तरीका है। एपीआई विकल्प बनाना, एक्सेस करना, अपडेट करना और हटाना आसान बनाता है।

वाक्य - विन्यास

- `// वर्डप्रेस ऐड_ऑप्शन के भीतर नया विकल्प बनाएं ($ विकल्प, $ मूल्य = , $ बहिष्कृत = , $autoload = 'हाँ');`
- `// डेटाबेस से एक विकल्प हटाता है। डिलीट_ऑप्शन ($ विकल्प);`
- `// सहेजे गए विकल्प को पुनः प्राप्त करें
get_option ($ विकल्प, $ डिफ़ॉल्ट = गलत);`
- `// पहले से जोड़े गए विकल्प का मान अपडेट करें। update_option ($ विकल्प, $ newvalue);`
- `// इन फ़ंक्शंस के *_site_option() संस्करण भी हैं, // वर्डप्रेस मल्टीसाइट में नेटवर्क-वाइड विकल्पों में हेरफेर करने के लिए`
- `// नया नेटवर्क विकल्प बनाएं
add_site_option($option, $value = , $deprecated = , $autoload = 'हाँ');`
- `// एक नेटवर्क विकल्प हटाता है
delete_site_option($option);`
- `// सहेजे गए नेटवर्क विकल्प को पुनः प्राप्त करें
get_site_option ($ विकल्प, $ डिफ़ॉल्ट = गलत);`
- `// पहले से जोड़े गए विकल्प का मान अपडेट करें। update_site_option ($ विकल्प, $ newvalue);`

टिप्पणियां

विकल्प एपीआई MySQL डेटाबेस की विकल्प तालिका में संग्रहीत डेटा के साथ काम करने का एक सरल और मानकीकृत तरीका है। एपीआई विकल्प बनाना, पढ़ना, अपडेट करना और हटाना आसान बनाता है।

उदाहरण

get_option

`get_option` फ़ंक्शन का उपयोग विकल्प नाम के आधार पर विकल्प तालिका से मान पुनर्प्राप्त करने के लिए किया जाता है।

आप वर्डप्रेस साइट व्यवस्थापक का ईमेल पता प्राप्त करने के लिए निम्नलिखित कोड का उपयोग कर सकते हैं।

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

`get_option()` में एक वैकल्पिक दूसरा तर्क है, जो आपको अनुरोधित विकल्प सेट नहीं होने की स्थिति में वापस लौटने के लिए एक डिफ़ॉल्ट मान सेट करने की अनुमति देता है। डिफ़ॉल्ट रूप से, यह तर्क ग़लत है।

टेक्स्ट स्ट्रिंग को पुनः प्राप्त करने के लिए, और यदि टेक्स्ट विकल्प तालिका में सेट नहीं है तो बॉयलरप्लेट स्ट्रिंग का उपयोग करें, आप यह कर सकते हैं:

```
<?php get_option('my_text', "मेरे पास कुछ भी लिखा नहीं है। अभी तक।"); ?>
```

add_option

`add_option` फ़ंक्शन इन्स का उपयोग विकल्प तालिका में नई पंक्ति सम्मिलित करने के लिए किया जाता है।

यह विकल्प तालिका में विकल्प नाम `some_option_name` और मान `some_option_value` के साथ एक नई पंक्ति सम्मिलित करेगा

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

हटाएँ_विकल्प

विकल्प तालिका से किसी विकल्प को हटाने के लिए `delete_option` फ़ंक्शन का उपयोग किया जाता है।

यह विकल्प तालिका से `my_custom_option` को हटा देगा।

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

अद्यतन_विकल्प

`update_option` फ़ंक्शन का उपयोग विकल्प तालिका में पहले से मौजूद मान को अद्यतन करने के लिए किया जाता है। यदि विकल्प मौजूद नहीं है, तो विकल्प को विकल्प मान के साथ जोड़ा जाएगा।

यह डिफ़ॉल्ट टिप्पणी स्थिति को 'बंद' पर सेट कर देगा:

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

अध्याय 42: प्लगइन विकास

वाक्य - विन्यास

- `add_action`(स्ट्रिंग \$टैग, कॉल करने योग्य \$function_to_add, int \$प्राथमिकता = 10, int \$accepted_args = 1)
- `add_filter`(स्ट्रिंग \$टैग, कॉल करने योग्य \$function_to_add, int \$प्राथमिकता = 10, int \$accepted_args = 1)

पैरामीटर

पैरामीटर	विवरण
\$टैग	(स्ट्रिंग) (आवश्यक) \$function_to_add कॉलबैक को हुक करने के लिए फ़िल्टर का नाम।
\$function_to_add (कॉल करने योग्य)	(आवश्यक) फ़िल्टर लागू होने पर चलाया जाने वाला कॉलबैक।
\$प्राथमिकता	(int) (वैकल्पिक) उस क्रम को निर्दिष्ट करने के लिए उपयोग किया जाता है जिसमें किसी विशेष कार्रवाई से जुड़े कार्यों को निष्पादित किया जाता है। कम संख्याएँ पहले के निष्पादन से मेल खाती हैं, और समान प्राथमिकता वाले कार्यों को उसी क्रम में निष्पादित किया जाता है जिसमें उन्हें कार्रवाई में जोड़ा गया था। डिफ़ॉल्ट मान: 10
\$accepted_args	(int) (वैकल्पिक) फ़ंक्शन द्वारा स्वीकार किए जाने वाले तर्कों की संख्या। डिफ़ॉल्ट मान: 1

टिप्पणियां

जिस तरह से प्लगइन हुक काम करता है वह यह है कि वर्डप्रेस चलने के दौरान विभिन्न समय पर, वर्डप्रेस यह देखने के लिए जांच करता है कि क्या किसी प्लगइन में उस समय चलने के लिए पंजीकृत फ़ंक्शन हैं, और यदि हां, तो फ़ंक्शन चलाए जाते हैं। ये फ़ंक्शन वर्डप्रेस के डिफ़ॉल्ट व्यवहार को संशोधित करते हैं।

हुक दो प्रकार के होते हैं:

फ़िल्टर आपको वर्डप्रेस के निष्पादन के दौरान डेटा के एक टुकड़े के मूल्य को बदलने की क्षमता देते हैं। फ़िल्टर के लिए कॉलबैक फ़ंक्शंस को एक वेरिएबल के माध्यम से पारित किया जाएगा, संशोधित किया जाएगा, और फिर वापस लौटाया जाएगा। वे अलग-अलग तरीके से काम करने के लिए हैं, और उन्हें कभी भी वैश्विक चर या फ़ंक्शन के बाहर किसी अन्य चीज़ को प्रभावित नहीं करना चाहिए।

इसके विपरीत, क्रियाएँ आपको वर्डप्रेस के संचालन के तरीके को जोड़ने या बदलने की अनुमति देती हैं। आपका कॉलबैक फ़ंक्शन वर्डप्रेस के निष्पादन में एक विशिष्ट बिंदु पर चलेगा, और कुछ प्रकार के कार्य कर सकता है, जैसे उपयोगकर्ता के लिए आउटपुट को प्रतिध्वनित करना या डेटाबेस में कुछ डालना।

[फ़िल्टर संदर्भ](#)

[क्रिया संदर्भ](#)

पुस्तिका

प्लगइन एपीआई

फ़िल्टर बनाम क्रियाएँ

उदाहरण

फ़िल्टर

```
add_filter('comment_text','before_comment');
add_filter('comment_text','after_comment'); फ़ंक्शन
before_comment($comment_text){
    'टिप्पणी से पहले इनपुट' लौटाएँ।$comment_text;
}
फ़ंक्शन After_comment($comment_text){
    $comment_text लौटाएँ.'टिप्पणी के बाद इनपुट';
}
```

कार्य

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

फ़ंक्शन हुक_जावास्क्रिप्ट() {
    $output='<script> चेतावनी('पेज लोड हो रहा है...'); </script>"; प्रतिध्वनि $आउटपुट;
}
}
```

प्लगइन विकास उदाहरण: पसंदीदा गीत विजेट

```
<?php
फ़ंक्शन wpshout_register_widgets() {
    रजिस्टर_विजेट('पसंदीदा_सॉन्ग_विजेट');
}

add_action('widgets_init', 'wpshout_register_widgets');

वर्ग पसंदीदा_सॉन्ग_विजेट WP_Widget का विस्तार करता है {

फ़ंक्शन पसंदीदा_सॉन्ग_विजेट() {
    // पैरेंट ऑब्जेक्ट को इंस्टेंट करें पैरेंट::__कंस्ट्रक्ट(

        'पसंदीदा_सॉन्ग_विजेट', // बेस आईडी
        __('पसंदीदा गाना', 'टेक्स्ट_डोमेन'), // नाम सरणी ('विवरण' => __('बजाने योग्य पसंदीदा गाने के लिए
        विजेट', 'टेक्स्ट_डोमेन'),
    ) // तर्क
    );
}

फ़ंक्शन विजेट ($args, $instance ) echo $args['before_widget']; इको {
    '<h3>पसंदीदा गीत सूचियाँ:</h3>'; प्रतिध्वनि
    $उदाहरण['songinfo']; इको '<a href="' . $instance['link'] . "'>इसे डाउनलोड
    करें</a><br>';
```

```

    प्रतिध्वनि $उदाहरण['विवरण']; प्रतिध्वनि $args['after_widget'];
}

फ़ंक्शन अपडेट ($new_abc,$old_abc) { $instance = $old_abc;

    // फ़िल्ड्स
    $instance['link'] = Strip_tags($new_abc['link']); $instance['songinfo'] =
    Strip_tags($new_abc['songinfo']);
        $instance['description'] = Strip_tags($new_abc['description']); वापसी $उदाहरण;
}

// विजेट फॉर्म निर्माण फ़ंक्शन फॉर्म ($ उदाहरण)
{ $ लिंक = ''; $songinfo = ''; $विवरण = '';

    // मान जाँचें यदि ($instance)
    { $link =
        esc_attr($instance['link']); $songinfo =
        esc_textarea($instance['songinfo']); $विवरण = esc_textarea($instance['description']);
    } ?>

    <पी>
        <label for='<?php echo $this->get_field_id('link'); ?>'><?php _e('Link', 'wp_widget_plugin'); ?></लेबल> <इनपुट क्लास = "वाइडफ़ैट"
आईडी = "<?php इको $ यह-> get_field_id ('लिंक'); ?>" नाम
        = "<?php
प्रतिध्वनि $यह->get_field_name('link'); ?>" type='text' value='<?php echo $link; ?>' />
    </p>

    <पी>
        <label for='<?php echo $this->get_field_id('songinfo'); ?>'><?php _e('Song Info:', 'wp_widget_plugin'); ?></लेबल> <इनपुट क्लास='वाइडफ़ैट' आईडी='<?
php इको $this->get_field_id('songinfo'); ?>' name='<?
        php
प्रतिध्वनि $यह->get_field_name('songinfo'); ?>" type='text' value='<?php echo $songinfo; ?>' />
    </p>

    <पी>
        <label for='<?php echo $this->get_field_id('description'); ?>'><?php _e('Description:', 'wp_widget_plugin'); ?></लेबल>

        <textarea class='वाइडफ़ैट' id='<?php इको $this->get_field_id('description'); ?>' name='<?php echo $this->get_field_name('description'); ?>'
टाइप ``पाठ`` मान='<?php प्रतिध्वनि $विवरण; ?>'></textarea> </p>

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

<?php }
}

```

अध्याय 43: पोस्ट प्रारूप

टिप्पणियां

यदि थीम उनके लिए समर्थन सक्षम करती है, तो निम्नलिखित पोस्ट प्रारूप उपयोगकर्ताओं के लिए चुनने के लिए उपलब्ध हैं।

ध्यान दें कि हालांकि वास्तविक पोस्ट सामग्री प्रविष्टि नहीं बदलेगी, थीम चुने गए प्रारूप के आधार पर पोस्ट को अलग ढंग से प्रदर्शित करने के लिए इस उपयोगकर्ता की पसंद का उपयोग कर सकती है। उदाहरण के लिए, कोई थीम "स्थिति" पोस्ट के लिए शीर्षक का प्रदर्शन बंद कर सकती है। चीजें कैसे प्रदर्शित की जाती हैं यह पूरी तरह से विषय पर निर्भर है, लेकिन यहां कुछ सामान्य दिशानिर्देश दिए गए हैं।

- एक तरफ - आमतौर पर बिना किसी शीर्षक के स्टाइल किया गया। फेसबुक नोट अपडेट के समान।
- गैलरी - छवियों की एक गैलरी। पोस्ट में संभवतः एक गैलरी शोर्टकोड होगा और छवि संलग्नक होंगे।
- लिंक - किसी अन्य साइट का लिंक। थीम पोस्ट सामग्री में पहले टैग को उस पोस्ट के बाहरी लिंक के रूप में उपयोग करना चाह सकती हैं। एक वैकल्पिक दृष्टिकोण यह हो सकता है कि यदि पोस्ट में केवल एक यूआरएल है, तो वह यूआरएल होगा और शीर्षक (पोस्ट_टाइटल) इसके लिए एंकर से जुड़ा नाम होगा। छवि - एक एकल छवि। पोस्ट में पहला टैग छवि माना जा सकता है। वैकल्पिक रूप से, यदि पोस्ट में केवल एक यूआरएल है,
- तो वह छवि यूआरएल होगा और पोस्ट का शीर्षक (पोस्ट_टाइटल) छवि के लिए शीर्षक विशेषता होगी। उद्धरण - एक उद्धरण। संभवतः इसमें उद्धरण सामग्री रखने वाला एक ब्लॉककोट होगा।
- वैकल्पिक रूप से, उद्धरण केवल सामग्री हो सकता है, स्रोत/लेखक शीर्षक हो सकता है।
- स्टेटस - ट्विटर स्टेटस अपडेट के समान एक संक्षिप्त स्टेटस अपडेट। वीडियो - एक एकल वीडियो या वीडियो प्लेलिस्ट। पोस्ट सामग्री में पहला टैग या ऑब्जेक्ट/एम्बेड को वीडियो माना जा सकता है। वैकल्पिक रूप से, यदि पोस्ट में केवल एक यूआरएल है, तो वह वीडियो यूआरएल होगा। यदि ब्लॉग पर वीडियो समर्थन सक्षम है (जैसे प्लगइन के माध्यम से) तो पोस्ट में अनुलग्नक के रूप में वीडियो भी शामिल हो सकता है। ऑडियो - एक ऑडियो फ़ाइल या प्लेलिस्ट। पॉडकास्टिंग के लिए इस्तेमाल किया जा सकता है।
- चैट - एक चैट प्रतिलेख

उदाहरण

थीम में पोस्ट प्रकार जोड़ा जा रहा है

post_type 'पेज' में पोस्ट-फ़ॉर्मेट जोड़ें

```
add_post_type_support('पेज', 'पोस्ट-प्रारूप');
```

अगला उदाहरण कस्टम पोस्ट प्रकार 'my_custom_post_type' को पंजीकृत करता है, और पोस्ट प्रारूप जोड़ता है।

कस्टम पोस्ट प्रकार 'my_custom_post_type' पंजीकृत करें

```

add_action('init', 'create_my_post_type'); फ़ंक्शन create_my_post_type() {

    रजिस्टर_पोस्ट_टाइप('my_custom_post_type',
        सारणी(
            'लेबल' => सरणी ('नाम' => __( 'उत्पाद' ) ), 'सार्वजनिक' => सत्य
        )
    );
}

```

पोस्ट_टाइप 'my_custom_post_type' में पोस्ट-फ़ॉर्मेट जोड़ें

```

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

```

या फ़ंक्शन रजिस्टर_पोस्ट_टाइप () में, 'सपोर्ट' पैरामीटर ऐसे में 'पोस्ट-फ़ॉर्मेट' जोड़ें। अगला उदाहरण उपरोक्त उदाहरण के बराबर है।

कस्टम पोस्ट प्रकार 'my_custom_post_type' को 'समर्थन' पैरामीटर के साथ पंजीकृत करें

```

add_action('init', 'create_my_post_type'); फ़ंक्शन create_my_post_type() {

    रजिस्टर_पोस्ट_टाइप('my_custom_post_type',
        सारणी(
            'लेबल' => सरणी ('नाम' => __( 'उत्पाद' ) ), 'सार्वजनिक' => सत्य, 'समर्थन' => सरणी ('शीर्षक', 'संपादक',
            'पोस्ट-प्रारूप')
        )
    );
}

```

पोस्ट के लिए थीम समर्थन जोड़ें

फ़ंक्शन कॉल

```

add_theme_support('पोस्ट-प्रारूप')

```

अध्याय 44: पोस्ट पूछना

वाक्य - विन्यास

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

पैरामीटर

पैरामीटर विवरण	
<code>\$args</code>	(सरणी) किसी क्वेरी के लिए आवश्यक तर्कों की एक श्रृंखला - आपकी आवश्यकताओं के अनुरूप बनाई जा सकती है, उदाहरण के लिए केवल एक श्रेणी से पोस्ट क्वेरी करना, कस्टम पोस्ट प्रकार से या यहां तक कि कुछ वर्गीकरण से क्वेरी करना

टिप्पणियां

क्वेरी तर्क असंख्य हैं। `WP_Query()` कोडेक्स पृष्ठ में पैरामीटरों की एक सूची है। उनमें से कुछ हैं

- [लेखक पैरामीटर](#)
- [श्रेणी पैरामीटर](#) • [टैग पैरामीटर्स](#) • [टैक्सोनॉमी पैरामीटर्स](#) • [खोज पैरामीटर](#) • [पोस्ट और पेज पैरामीटर्स](#) • [पासवर्ड पैरामीटर्स](#)
- [पैरामीटर टाइप करें](#) • [स्थिति पैरामीटर](#) • [पेजिनेशन पैरामीटर्स](#)
- [ऑर्डर और ऑर्डरबाय पैरामीटर्स](#) • [दिनांक पैरामीटर्स](#) • [कस्टम फ़िल्ड पैरामीटर्स](#) • [अनुमति पैरामीटर](#) • [माइम प्रकार पैरामीटर्स](#) • [कैशिंग पैरामीटर्स](#) • [वापसी फ़िल्ड पैरामीटर](#)

सबसे महत्वपूर्ण बात जो ध्यान में रखनी चाहिए वह है:

कभी भी `query_posts()` का उपयोग न करें

`query_posts()` मुख्य क्वेरी को ओवरराइड करता है, और आपकी बाकी थीम में समस्याएँ पैदा कर सकता है। कोई

आपको मुख्य क्वेरी (या उस मामले के लिए कोई भी क्वेरी) को संशोधित करने के लिए `pre_get_posts` का उपयोग करना होगा फ़िल्टर. यह आपको क्वेरी चलने से पहले उसे संशोधित करने की अनुमति देगा।

साथ ही जब आप पोस्ट के लिए क्वेरी कर रहे हों, तो आपको हमेशा इसे `wp_reset_postdata()` का उपयोग करके रीसेट करना चाहिए। यह मुख्य क्वेरी लूप के वैश्विक `$post` वैरिएबल को पुनर्स्थापित करेगा, और आपको बाद में कोई समस्या नहीं होगी (जैसे श्रेणियों को बाहर रखा जाना, क्योंकि आपके द्वितीयक लूप में आपने उन्हें बाहर कर दिया है और क्वेरी को रीसेट करना भूल गए हैं)।

उदाहरण

`WP_Query()` ऑब्जेक्ट का उपयोग करना

`WP_Query` ऑब्जेक्ट का एक अलग उदाहरण बनाना आसान है:

```
$query_args = सरणी(
    'पोस्ट_टाइप' => 'पोस्ट', 'पोस्ट_पर_पेज' =>
    10
);

$my_query = नया WP_Query($query_args);

अगर( $my_query->have_posts() ):
    जबकि( $my_query->have_posts() ): $my_query->the_post();
    //मेरा कस्टम क्वेरी लूप एंडव्हाइल; अगर अंत;

wp_reset_postdata();
```

ध्यान दें कि आपको अपने विनिर्देश के अनुसार क्वेरी तर्क सरणी बनाने की आवश्यकता है। अधिक विवरण के लिए, `WP_Query` कोडेक्स पृष्ठ देखें।

`Get_posts()` का उपयोग करना

`get_posts()` `WP_Query` ऑब्जेक्ट के एक अलग उदाहरण के लिए एक रैपर है। लौटाया गया मान पोस्ट ऑब्जेक्ट की एक सरणी है।

```
वैश्विक $पोस्ट;

$args = सरणी ( 'नंबरपोस्ट'
    => 5, 'ऑफ़सेट' => 1, 'श्रेणी' => 1
);

$myposts = get_posts($args);

foreach($myposts $post के रूप में) :
    सेटअप_पोस्टडेटा ($पोस्ट); ?> <h2><a href='<?
    php the_permalink(); ?>'><?php the_title(); ?></a></h2> <?php Endforeach; wp_reset_postdata(); ?>
```


अध्याय 45: यहां से ऑटो लाइन ब्रेक हटाएं सामग्री और अंश

परिचय

उन साइटों के लिए जो संपादक या अंशों में हाथ से HTML पर निर्भर हैं, जिन्हें आप स्वयं कोड करना चाहते हैं, ऑटो लाइन ब्रेक एक परेशानी हो सकती है। आप इन फ़िल्टर को हटाकर उन्हें अक्षम कर सकते हैं।

टिप्पणियां

इन्हें सीधे सम्मिलित फ़ाइल में निष्पादित किया जाना चाहिए। चाहे यह function.php में हो या किसी अन्य फ़ाइल में, इन्हें एक हुक में लपेटा नहीं जा सकता। वे init या अब तक मुझे मिले किसी भी अन्य पर काम नहीं करेंगे।

उन्हें केवल उस टेम्पलेट के लिए निष्पादित करने के लिए सीधे पेज.php जैसे टेम्पलेट में भी शामिल किया जा सकता है।

ध्यान दें: इसे किसी वितरित थीम या प्लगइन में शामिल न करें (जब तक कि यह डिफ़ॉल्ट रूप से अक्षम न हो, जैसे कि इसमें शामिल फ़ाइल को शामिल न करें जब तक कि उपयोगकर्ता निर्दिष्ट न करे)।

किसी ऐसी साइट पर इसे शामिल करना बुरा अभ्यास है जिसे आप नियंत्रित नहीं करते हैं क्योंकि यह किसी भी अन्य थीम या प्लगइन के आउटपुट को तोड़ सकता है और तोड़ देगा।

उदाहरण

फ़िल्टर हटाएँ

```
// सामग्री से ऑटो-पैराग्राफ और ऑटो-लाइन-ब्रेक हटाएँ रिमूव_फिल्टर ('द_कंटेंट', 'डब्ल्यूपीटॉप');  
  
// अंश निकालें फ़िल्टर ('the_excerpt', 'wpauto') से ऑटो-पैराग्राफ और ऑटो-लाइन-ब्रेक हटाएँ;
```

फ़िल्टर हटाने का कार्य

```
/**  
 * सामग्री और अंशों से स्वचालित लाइन ब्रेक हटा दें।  
 *  
 * @1.0.0 से  
 */  
फ़ंक्शन रिमूव_कंटेंट_ऑटो_लाइन_ब्रेक्स() {  
    // सामग्री से ऑटो-पैराग्राफ और ऑटो-लाइन-ब्रेक हटाएँ रिमूव_फिल्टर ('द_कंटेंट', 'डब्ल्यूपीटॉप');  
  
    // अंश निकालें फ़िल्टर ('the_excerpt', 'wpauto') से ऑटो-पैराग्राफ और ऑटो-लाइन-ब्रेक हटाएँ;  
  
}
```

```
// फ़ंक्शन निष्पादित करें  
delete_content_auto_line_breaks();
```

अध्याय 46: वर्डप्रेस और स्टाइलशीट से संस्करण हटाएं

परिचय

दूसरों के लिए आपकी वेबसाइट को हैक करना अधिक कठिन बनाने के लिए आप अपनी साइट, अपने सीएसएस और जेएस से वर्डप्रेस संस्करण संख्या हटा सकते हैं। उस संख्या के बिना यह देखना संभव नहीं है कि क्या आप पुराने संस्करणों से बग का फायदा उठाने के लिए वर्तमान संस्करण नहीं चला रहे हैं।

इसके अतिरिक्त यह आपकी साइट की लोडिंग गति में सुधार कर सकता है, क्योंकि यूआरएल में क्वेरी स्ट्रिंग के बिना सीएसएस और जेएस फ़ाइलों को कैश किया जा सकता है।

वाक्य - विन्यास

- `add_filter($टैग, $function_to_add, $प्राथमिकता, $accepted_args)`

पैरामीटर

पैरामीटर	विवरण
\$टैग	(स्ट्रिंग आवश्यक) फ़िल्टर का नाम जहां \$function_to_add जुड़ा हुआ है
\$function_to_add	(कॉल करने योग्य आवश्यक) फ़ंक्शन का नाम जो फ़िल्टर लागू होने पर चलता है
\$प्राथमिकता	(int वैकल्पिक) एक क्रिया में अन्य कार्यों के बीच \$function_to_add का स्थान (डिफ़ॉल्ट = 10)
\$accepted_args	(int वैकल्पिक) पैरामीटर की संख्या जो \$ function_to_add स्वीकार करता है (डिफ़ॉल्ट = 1)

टिप्पणियां

साइट की गति और सुरक्षा में सुधार करने का इरादा है।

उदाहरण

सीएसएस/जेएस से संस्करण संख्याएं हटाएं

बस इस फ़ंक्शन को अपने `function.php` में जोड़ें। यह सभी कतारबद्ध जेएस और सीएसएस फाइलों से संस्करण को हटा देगा।

```
फ़ंक्शन रिमूव_सीएसएसजेएस_वर( $src ) यदि( strpos( $src, '?ver=' ) ) {  
    $src = delete_query_arg( 'ver', $src ); रिटर्न $src; }  
  
}
```

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

वर्डप्रेस से संस्करण संख्याएँ हटाएँ

यदि आप इसे अपने function.php में जोड़ते हैं तो यह RSS फ़ीड और हेडर से वर्डप्रेस संस्करण संख्या को हटा देता है।

```
फ़ंक्शन रिमूव_वर्डप्रेस_वर() {  
    वापस करना '';  
  
} add_filter( 'the_generator', 'remove_wordpress_ver', 999 );
```

अध्याय 47: रेस्ट एपीआई

परिचय

वर्डप्रेस REST एपीआई वर्डप्रेस डेटा प्रकारों के लिए एपीआई एंडपॉइंट प्रदान करता है जो डेवलपर्स को JSON (जावास्क्रिप्ट ऑब्जेक्ट नोटेशन) ऑब्जेक्ट भेजकर और प्राप्त करके दूरस्थ रूप से साइटों के साथ बातचीत करने की अनुमति देता है।

जब आप एपीआई को सामग्री भेजते हैं या अनुरोध करते हैं, तो प्रतिक्रिया JSON में लौटा दी जाएगी।

यह डेवलपर्स को क्लाइंट-साइड जावास्क्रिप्ट या बाहरी अनुप्रयोगों से, यहां तक कि PHP से परे भाषाओं में लिखे गए वर्डप्रेस सामग्री को बनाने, पढ़ने और अपडेट करने में सक्षम बनाता है।

टिप्पणियां

इस वर्डप्रेस REST API के सरल उदाहरण को आपके लिए काम में लाने के लिए, आपको यह अधिक विस्तार से सीखना होगा कि यह कैसे काम करता है। आधिकारिक दस्तावेज़ इसके बारे में सीखने की अनुशंसा करता है:

1. रूट/एंडपॉइंट - जो "एंडपॉइंट" के रूप में जाने जाने वाले मार्गों के लिए अलग-अलग HTTP तरीकों की मैपिंग है - आप इसे रजिस्टर_रेस्ट_रूट () फ़ंक्शन का उपयोग करके करते हैं, और यहां आप रूट और एंडपॉइंट के बारे में अधिक जानकारी पा सकते हैं।
2. अनुरोध - वर्डप्रेस REST API WP_REST_Request क्लास को परिभाषित करता है जिसका उपयोग वर्तमान अनुरोध के लिए जानकारी संग्रहीत और पुनर्प्राप्त करने के लिए किया जाता है। जब भी आप किसी पंजीकृत रूट के लिए HTTP अनुरोध करते हैं तो WP_REST_Request ऑब्जेक्ट स्वचालित रूप से आपके लिए जनरेट हो जाते हैं। अनुरोध में निर्दिष्ट डेटा यह निर्धारित करेगा कि आपको एपीआई से क्या प्रतिक्रिया मिलेगी। यहां WP_REST_Request क्लास के बारे में अधिक जान सकते हैं।
3. प्रतिक्रियाएँ - वह डेटा है जो आपको एपीआई से वापस मिलता है। WP_REST_Response एंडपॉइंट्स द्वारा लौटाए गए प्रतिक्रिया डेटा के साथ बातचीत करने का एक तरीका प्रदान करता है। अपनी समापन बिंदु परिभाषा में आप अपनी बातचीत को पूरा करने के लिए कॉलबैक (प्रतिक्रिया) फ़ंक्शन को नाम देते हैं। यहां WP_REST_Response क्लास के बारे में अधिक जान सकते हैं।
4. स्कीमा - प्रत्येक समापन बिंदु को थोड़ी अलग डेटा संरचनाओं की आवश्यकता होती है और प्रदान करती है, और उन संरचनाओं को एपीआई स्कीमा में परिभाषित किया गया है। यदि आप रखरखाव योग्य, खोजने योग्य और आसानी से विस्तार योग्य समापन बिंदु चाहते हैं तो स्कीमा का उपयोग करने की अनुशंसा की जाती है। यहां आप स्कीमा के बारे में अधिक जान सकते हैं।
5. नियंत्रक वर्ग - वे सभी तत्वों को एक ही स्थान पर एक साथ लाते हैं। नियंत्रक वर्ग के साथ आप मार्गों और समापन बिंदुओं के पंजीकरण का प्रबंधन कर सकते हैं, अनुरोधों को संभाल सकते हैं, स्कीमा का उपयोग कर सकते हैं और एपीआई प्रतिक्रियाएं उत्पन्न कर सकते हैं। आप पहले ही दो नियंत्रक वर्गों के बारे में जान चुके हैं: WP_REST_Request और WP_REST_Response। यहां आप नियंत्रक कक्षाओं के बारे में अधिक जान सकते हैं

नोट: इनमें से कुछ जानकारी आधिकारिक Wordpress REST API हैंडबुक से ली गई है

उदाहरण

पूर्ण कामकाजी उदाहरण

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

    // अपना नेमस्पेस घोषित करें $namespace =
    'myrest/v1';

    // मार्ग पंजीकृत करें
    // इस मार्ग से मेल खाने वाला उदाहरण URL: // http://yourdomain/
    wp-json/myrest/v1/guides/tag=europe/price=29 रजिस्टर_रेस्ट_रूट($namespace,

        // रेगुलर एक्सप्रेसन का उपयोग करके हम प्रारंभ में इनपुट '/guides/tag=(?P<tag>[a-zA-Z0-9-]+)/price=(?P<price>[0-9]+ को
        मान्य कर सकते हैं)', // हमारे पास एक रूट एरे के लिए कई एंडपॉइंट हो सकते हैं ( एरे( 'मेथड्स' => 'जीईटी', 'कॉलबैक' =>
        'my_get_guides_handler'

    )

    ),
    // हम अपना स्कीमा कॉलबैक पंजीकृत कर सकते हैं // 'schema' =>
    'my_get_guide_schema',

    );

    // आप इसी तरह यहां दूसरा रूट भी रजिस्टर कर सकते हैं
}

// एंडपॉइंट फ़ंक्शन के लिए कॉलबैक हैंडलर
my_get_guides_handler(WP_REST_Request $request) {

    // पैरामीटर प्राप्त करें: $tag = $request-
    >get_param('tag'); $कीमत = $अनुरोध->get_param('कीमत');

    // पैरामीटर के साथ कुछ करें // उदाहरण के लिए: डीबी से मिलान गाइड को
    एक सरणी में प्राप्त करें - $ परिणाम // ...

    // प्रतिक्रिया तैयार करें $message = "हमने पाया
    है " $message .= "(एक टैग की खोज:          . गिनती($परिणाम) "मार्गदर्शक"; . $टैग। ", मूल्य टैग के साथ:
    "          . $कीमत। ")";

    // प्रतिक्रिया स्वचालित रूप से JSON प्रारूप में परिवर्तित हो जाती है, नया WP_REST_Response (सरणी ('परिणाम' => $ परिणाम,
    'संदेश' => $ संदेश, 'स्थिति' => 'ठीक' लौटाता है)

    ),
    200 );

}
```

अध्याय 48: वर्डप्रेस लोकल को इसके साथ चलाएं एक्सएएमपीपी

परिचय

XAMPP से आप अपने स्थानीय पीसी पर एक वेब सर्वर स्थापित कर सकते हैं। इसमें एक Apache वेब सर्वर, डेटाबेस MariaDB (MySQL) है और यह पर्ल और PHP के साथ काम करता है। इंस्टॉलेशन के बाद आप अपने स्थानीय पीसी पर वर्डप्रेस जैसे कंटेंट मैनेजमेंट सिस्टम को चलाने और डिबग करने में सक्षम हैं। आप इसे विंडोज़, लिनक्स, मैक ओएस एक्स और सोलारिस के साथ उपयोग कर सकते हैं।

उदाहरण

1. एक्सएएमपीपी स्थापित करना

- वर्तमान संस्करण के लिए इंस्टॉलर डाउनलोड करें XAMPP 2 का.

यदि आप सुनिश्चित नहीं हैं कि आप क्या बदल रहे हैं तो इंस्टॉलेशन विज़ार्ड से गुजरें और कुछ भी न बदलें। 3.

इंस्टालेशन के बाद आपको XAMPP कंट्रोल पैनल दिखाई देगा। दोनों को बुनियादी कॉन्फ़िगरेशन के साथ चलाने के लिए बस Apache और MySQL के स्टार्ट बटन पर क्लिक करें। 4.

यदि आपका कंप्यूटर किसी स्थानीय नेटवर्क से जुड़ा है, और आप अन्य कंप्यूटरों से भी वेब सर्वर तक पहुंच चाहते हैं, तो सुनिश्चित करें कि आपका फ़ायरवॉल इसकी अनुमति देता है।

2. XAMPP इंस्टाल करने के बाद एक डेटाबेस सेटअप करें

अपने कंप्यूटर पर वर्डप्रेस चलाने के लिए आपको पहले एक डेटाबेस कॉन्फ़िगर करना होगा। सुनिश्चित करें कि Apache और MySQL चल रहे हैं (XAMPP इंस्टाल करना देखें)। चरण 3)

- अपनी पसंद का वेब ब्राउज़र प्रारंभ करें और पते में "लोकलहोस्ट" दर्ज करें।
- अपनी पसंदीदा भाषा चुनें और "टूल्स" -> phpMyAdmin श्रेणी में चयन करें।
- टैब डेटाबेस चुनें.
- "डेटाबेस बनाएं" के नीचे अपने डेटाबेस का नाम दर्ज करें (आपको बाद में इस नाम की आवश्यकता होगी) और ड्रॉपडाउन "कोलेशन" में utf8_general_ci चुनें।
- बाईं ओर अपने बनाए गए डेटाबेस पर क्लिक करें और उपयोगकर्ता टैब चुनें।
- उपयोगकर्ता जोड़ें पर क्लिक करके एक नया उपयोगकर्ता जोड़ें। 7.
अपना उपयोगकर्ता नाम दर्ज करें, "होस्ट" ड्रॉपडाउन में स्थानीय चुनें (यदि आप नेटवर्क में अन्य कंप्यूटरों से डेटाबेस तक पहुंच चाहते हैं तो कोई भी होस्ट चुनें) और अपना पासवर्ड दर्ज करें (आपको बाद में भी इसकी आवश्यकता होगी)।
- जांचें कि क्या "उपयोगकर्ता के लिए डेटाबेस" के नीचे वाइल्डकार्ड नाम पर सभी विशेषाधिकार प्रदान करना चेक किया गया है।
- जाओ दबाएँ.

3. डेटाबेस सेटअप करने के बाद वर्डप्रेस इंस्टाल करना

XAMPP इंस्टॉल करने और MySQL डेटाबेस सेटअप करने के बाद आप वर्डप्रेस इंस्टॉल कर सकते हैं।

1. डाउनलोड करें वर्डप्रेस का नवीनतम संस्करण।

है फ़ाइल को अनज़िप करें और फ़ोल्डर को अपने वेबसर्वर की रूट डायरेक्टरी में डालें (यदि आपने XAMPP के सेटअप के दौरान 2. सुझाए गए पथ का उपयोग किया तो यह "c:\xampp\htdocs" है)।

3. दो बार मौजूद सभी फाइलों को बदला जा सकता है।

4. अपने वेबब्राउज़र के पता फ़ील्ड में "localhost/wordpress/wp-admin/install.php" दर्ज करें।

5. अपनी भाषा चुनने के बाद कंटिन्यू पर क्लिक करें और फिर लेट्स गो पर क्लिक करें। 6.

सभी डिफ़ॉल्ट टेक्स्ट फ़ील्ड को अपने चुने हुए कॉन्फ़िगरेशन से बदलें (देखें "डेटाबेस सेटअप करें")।

"डेटाबेस होस्ट" और "टेबल प्रीफ़िक्स" को न बदलें, सिवाय इसके कि आप अपने स्थानीय वर्डप्रेस इंस्टॉलेशन को ऑनलाइन माइग्रेट करना चाहते हैं। फिर ध्यान रखें कि सुरक्षा महत्वपूर्ण है. आपको नई वर्डप्रेस इंस्टालेशन में सेट प्रीफ़िक्स में अधिक जानकारी मिलेगी ।

7. सबमिट दबाएं और फिर इंस्टॉल चलाएं।

8. अब आपको अपनी साइट का नाम, एक यूजरनेम, पासवर्ड और एक वैध ई-मेल दर्ज करना होगा।

9. सेटअप पूरा करने के लिए वर्डप्रेस इंस्टॉल करें पर क्लिक करें और अपनी नई स्थानीय वर्डप्रेस साइट पर लॉग इन करें।

अध्याय 49: अपनी स्थापना सुरक्षित करें

टिप्पणियां

वर्डप्रेस वेबसाइटें अक्सर हैक होती रहती हैं। यह विषय उन तकनीकों और प्रथाओं के लिए है जो आपके वर्डप्रेस इंस्टॉलेशन की सुरक्षा को बेस इंस्टॉल में हासिल की गई सुरक्षा से अधिक बढ़ा देती हैं।

इस विषय के अलावा, वर्डप्रेस इंस्टॉलेशन को सुरक्षित करने के बारे में पढ़ने के लिए एक और अच्छी जगह हार्डनिंग वर्डप्रेस कोडेक्स पेज है।

उदाहरण

फ़ाइल संपादक अक्षम करें

वर्डप्रेस के साथ आने वाला फ़ाइल संपादक एक सुरक्षा जोखिम है। यदि कोई हमलावर आपकी वर्डप्रेस वेबसाइट पर व्यवस्थापक पहुंच प्राप्त कर लेता है तो वे आसानी से थीम और प्लगइन फ़ाइलों में दुर्भावनापूर्ण कोड डालने में सक्षम होंगे। यह उन ग्राहकों के लिए भी जोखिम है जो नहीं जानते कि वे क्या कर रहे हैं। एक बार फ़ाइल संपादक में गलत स्थान पर रखा गया कोलन किसी साइट को तोड़ सकता है और उसे ब्राउज़र से अप्राप्य बना सकता है।

अपनी वर्डप्रेस wp-config.php फ़ाइल में, कोड की निम्न पंक्ति जोड़कर फ़ाइल संपादक को अक्षम करें।

```
परिभाषित करें ('DISALLOW_FILE_EDIT', सत्य);
```

आपके थीम के function.php फ़ाइल में भी जोड़े जाने पर उस पंक्ति का वांछित प्रभाव होगा लेकिन इसे wp-config.php में जोड़ना बेहतर है।

यदि आप वर्डप्रेस सीएलआई का उपयोग कर रहे हैं वर्डप्रेस स्थापित करने के लिए आप फ़ाइल संपादन अक्षम के साथ एक wp-config.php फ़ाइल बनाने के लिए निम्न कमांड का उपयोग कर सकते हैं।

```
/* पहले से वैरिएबल घोषित करें या */ wp कोर कॉन्फ़िगरेशन में स्ट्रिंग्स को प्रतिस्थापित करें --dbname='$MYSQL_DBNAME' --dbuser='$MYSQL_USERNAME' --dbpass='$MYSQL_PASS' -- dbprefix='$WP_DBPREFIX' --locale=en_AU --extra-php <<PHP परिभाषित ('DISALLOW_FILE_EDIT', सत्य);
```

पीएचपी

यदि आप स्क्रिप्ट के साथ वर्डप्रेस इंस्टॉल करते हैं तो यह विधि उपयोगी है।

WP-config.php ले जाएँ

वर्डप्रेस इंस्टाल की सबसे संवेदनशील जानकारी wp-config.php फ़ाइल में संग्रहीत होती है। यदि किसी हैकर को इस फ़ाइल तक पहुंच मिल जाती है तो आपकी वेबसाइट पर उनका पूरा नियंत्रण हो जाता है।

डिफ़ॉल्ट रूप से wp-config.php वर्डप्रेस इंस्टॉल फ़ोल्डर में संग्रहीत होता है। इस फ़ाइल को चुराना कठिन बनाने के लिए आप इसे वेब एक्सेसिबल फ़ोल्डर से बाहर ले जा सकते हैं। यदि आप इसे केवल एक फ़ोल्डर ऊपर ले जाते हैं, तो वर्डप्रेस स्वचालित रूप से इसे ढूँढ लेगा। यदि आप wp-config.php को किसी भिन्न स्थान पर ले जाते हैं, तो वर्डप्रेस इंस्टॉलेशन फ़ोल्डर में wp-config.php नामक एक खाली फ़ाइल बनाएंगे। फिर निम्नलिखित जोड़ें:

```
परिभाषित('ABSPATH', dirname(__FILE__) . '/'); // '../wp-config.php'  
इंस्टॉलेशन फ़ोल्डर के ऊपर दो फ़ोल्डर्स का स्थान परिभाषित करता है।  
//आवश्यकतानुसार wp-config.php फ़ाइल के वास्तविक स्थान के साथ प्रतिस्थापित करें। require_once(ABSPATH . '../wp-config.php');
```

आपको उस फ़ोल्डर में php को निष्पादन योग्य बनाने की आवश्यकता हो सकती है जिसमें आप wp-config.php रखते हैं। आपको यथासंभव कम से कम फ़ोल्डरों में php को निष्पादन योग्य बनाना चाहिए। एक अच्छा सिस्टम वर्डप्रेस इंस्टाल को /path/to/wordpress/install/ में और कॉन्फ़िग को /path/to/wordpress/config/ में रखता है। आप यह सुनिश्चित करेंगे कि कॉन्फ़िगरेशन फ़ोल्डर वेब पहुंच योग्य नहीं है और कोई अन्य संवेदनशील जानकारी फ़ोल्डर पदानुक्रम में /path/to/ या उच्चतर में नहीं रखी जाएगी। उस स्थिति में आप अपने php.ini में निम्नलिखित के समान एक पंक्ति लिखेंगे :

```
open_आधारितार = "/पथ/से/वर्डप्रेस/इंस्टॉल/;/पथ/से/वर्डप्रेस/कॉन्फ़िगरेशन"
```

यह तकनीक विवादास्पद है और कुछ लोगों को नहीं लगता कि इससे सुरक्षा बढ़ती है। विषय पर व्यापक चर्चा इस वर्डप्रेस स्टैक एक्सचेंज प्रश्न पर पढ़ी जा सकती है।

वर्डप्रेस तालिकाओं के लिए एक कस्टम उपसर्ग सेट करें

जब आप अपने सर्वर पर वर्डप्रेस इंस्टॉल करते हैं, तो इंस्टॉलेशन स्क्रिप्ट सभी वर्डप्रेस MySQL टेबल नामों के सामने एक उपसर्ग लगाएगी। यह उपसर्ग डिफ़ॉल्ट रूप से 'wp_' पर सेट है। उदाहरण के लिए वर्डप्रेस पोस्ट तालिका को wp_posts कहा जाएगा। तालिका उपसर्ग को बदलकर आप अस्पष्टता से कुछ सुरक्षा बना सकते हैं। इस तरह जब कोई हैकर SQL इंजेक्शन हमलों का प्रयास करता है, तो उन्हें केवल 'wp_' का उपयोग करने के बजाय आपकी तालिका के उपसर्ग का अनुमान लगाना होगा। आप इस उपसर्ग को अपनी पसंद के अनुसार सेट कर सकते हैं।

नए वर्डप्रेस इंस्टालेशन में प्रीफ़िक्स सेट करें

यदि प्रसिद्ध 5 मिनट की स्थापना का उपयोग कर रहे हैं तो स्थापना के दौरान फ़ील्ड में उपसर्ग बदलें।



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="username"/>	Your database username.
Password	<input type="text" value="password"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

यदि वर्डप्रेस सीएलआई के माध्यम से इंस्टॉल किया जा रहा है तो निम्न कमांड का उपयोग करें:

```
// ऊपर अन्य वैरिएबल सेट करें, या अपनी स्ट्रिंग्स को इसमें बदलें।
WP_DBPREFIX=foo wp
कोर कॉन्फिगरेशन --dbname='$MYSQL_DBNAME' --dbuser='$MYSQL_USERNAME' --dbpass='$MYSQL_PASS' --dbprefix='$WP_DBPREFIX' --locale=en_AU
```

मौजूदा इंस्टालेशन में उपसर्ग बदलें

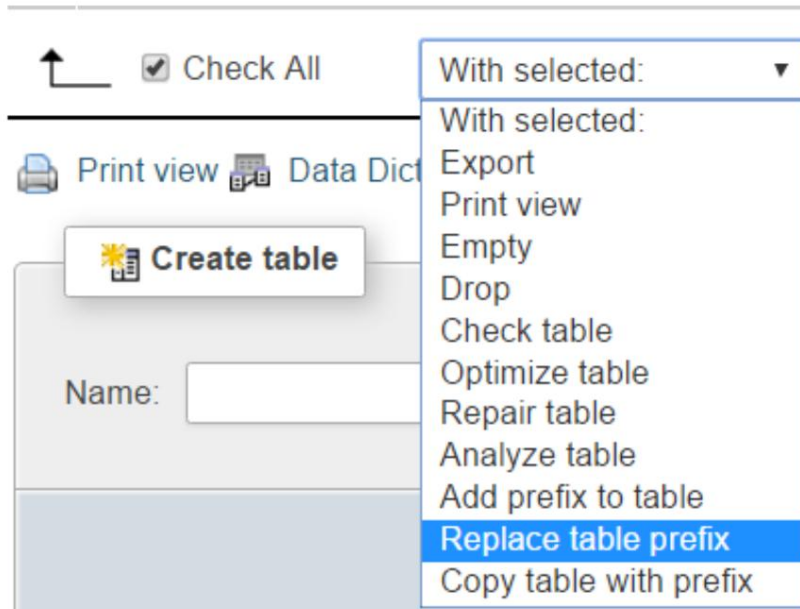
उपसर्ग बदलना थोड़ा अधिक कठिन है. सबसे पहले config.php फ़ाइल को संपादित करने के लिए FileZilla जैसे FTP प्रोग्राम का उपयोग करें। प्रविष्टि बदलें \$table_prefix = 'wp_'; से \$table_prefix = 'foo_'; अपने इच्छित उपसर्ग के लिए 'foo' प्रतिस्थापित करें।

आगे हमें डेटाबेस को संपादित करने की आवश्यकता होगी। यदि आपके पास phpMyAdmin तक पहुंच है, तो लॉगिन करें और निम्नलिखित कार्य करें:

- वर्डप्रेस डेटाबेस का चयन करें

<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

- सभी तालिकाओं का चयन करें और ड्रॉपडाउन में तालिका उपसर्ग बदलें का चयन करें।



- "From" में 'wp_' टाइप करें। इस उदाहरण में "To" में अपना उपसर्ग, 'foo_' टाइप करें और "सबमिट" दबाएँ।

Replace table prefix:

From

To

- तालिकाएँ अब इस तरह दिखनी चाहिए:

<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

यदि आप phpMyAdmin का उपयोग नहीं कर सकते हैं तो निम्न MySQL कमांड का उपयोग करें:

तालिका 'wp_comments' का नाम बदलकर 'foo_comments' कर दें




































आपको अन्य तालिका नामों के लिए 'टिप्पणियाँ' प्रतिस्थापित करते हुए, प्रत्येक तालिका के लिए उस आदेश को चलाने की आवश्यकता होगी।

आगे हमें कुछ तालिकाओं में कुछ प्रविष्टियाँ बदलने की आवश्यकता है। इस क्वेरी को 'foo_options' तालिका पर चलाएँ




चुनें * foo_options से जहाँ विकल्प_नाम '%user_roles%' जैसा है

'wp_user_roles' के विकल्प_नाम वाली एक प्रविष्टि दिखाई देनी चाहिए। उस प्रविष्टि में 'option_name' प्रविष्टि को wp_user_roles से foo_user_roles में बदलें।

फिर 'foo_usermeta' तालिका खोलें और सामने 'wp_' वाली प्रत्येक प्रविष्टि ढूँढें।

wp_...	wp_...	wp_...	wp_...	wp_...
 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

और इसे 'foo_' में बदलें। आपको कितनी प्रविष्टियाँ बदलनी होंगी यह इस बात पर निर्भर करेगा कि आपके पास कितने उपयोगकर्ता हैं।

<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

किसी मौजूदा इंस्टॉलेशन में उपसर्ग को बदलने के लिए आपको बस इतना ही चाहिए

अध्याय 50: वर्डप्रेस में सुरक्षा - बचना

वाक्य - विन्यास

- esc_html(स्ट्रिंग \$टेक्स्ट)
- esc_url(स्ट्रिंग \$url, एरे \$प्रोटोकॉल, स्ट्रिंग \$_context) • esc_js(स्ट्रिंग \$टेक्स्ट) • wp_json_encode(मिश्रित \$डेटा, int \$options, int \$depth = 512) • esc_attr(स्ट्रिंग \$पाठ) • esc_textarea(स्ट्रिंग \$पाठ)

टिप्पणियां

विकास करते समय सुरक्षा को हमेशा ध्यान में रखना चाहिए। सुरक्षा के बिना कोई ऐप विभिन्न हमलों जैसे SQL इंजेक्शन, XSS, CSRF, RFI आदि के लिए खुला रहता है जिससे गंभीर समस्याएं हो सकती हैं।

अविश्वसनीय डेटा कई स्रोतों (उपयोगकर्ताओं, तृतीय पक्ष साइटों, आपका अपना डेटाबेस!, ...) से आता है और इन सभी को इनपुट और आउटपुट दोनों पर मान्य करने की आवश्यकता होती है। (स्रोत: वर्डप्रेस कोडेक्स)

उपयोग और उद्देश्य के आधार पर डेटा को मान्य, स्वच्छ या एस्केप किया जाना चाहिए।

सत्यापित करने का अर्थ यह सुनिश्चित करना है कि आपने उपयोगकर्ता से जो डेटा अनुरोध किया है वह उनके द्वारा सबमिट किए गए डेटा से मेल खाता है। (स्रोत: वर्डप्रेस कोडेक्स)

उपयोगकर्ता डेटा स्वीकार करने के दृष्टिकोण में सेनिटाइजेशन थोड़ा अधिक उदार है। स्वीकार्य इनपुट की सीमा होने पर हम इन विधियों का उपयोग करने के लिए वापस आ सकते हैं। (स्रोत: वर्डप्रेस कोडेक्स)

बचने का मतलब है कि आपके पास पहले से मौजूद डेटा लेना और अंतिम उपयोगकर्ता के लिए इसे प्रस्तुत करने से पहले इसे सुरक्षित करने में मदद करना। (स्रोत: वर्डप्रेस कोडेक्स)

उदाहरण

HTML कोड में डेटा से बचें

जब भी हम HTML कोड के अंदर डेटा आउटपुट कर रहे हों तो esc_html का उपयोग किया जाना चाहिए।

```
<h4><?php इको esc_html($शीर्षक); ?></h4>
```

एक यूआरएल से बचें

```
<a href='<?php echo esc_url(home_url( '/' ) ); ?>'>होम</a>
```

```
<img src='<?php echo esc_url( $user_picture_url ); ?>' />
```

जेएस कोड में डेटा से बचें

esc_js() का उपयोग टैग विशेषता के अंदर इनलाइन JS के लिए किया जाना है।

<script> टैग के अंदर डेटा के लिए wp_json_encode() का उपयोग करें।

```
<इनपुट प्रकार = "टेक्स्ट" ऑनफोकस = "यदि (यह.वैल्यू == '<?php इको esc_js ($ फ़िल्ड्स ['इनपुट_टेक्स्ट']); ?>'
) {यह.मूल्य = '';} " name='नाम'>
```

wp_json_encode() कुछ विवेक जांच के साथ एक वेरिएबल को JSON में एनकोड करता है।

ध्यान दें कि wp_json_encode() में स्ट्रिंग-डीलिमिटिंग उद्धरण स्वचालित रूप से शामिल हैं।

```
<?php
$book = array( "शीर्षक"
    => "जावास्क्रिप्ट: निश्चित गाइड",
    "लेखक" => "स्टैक ओवरफ्लो",
);
?>

<स्क्रिप्ट प्रकार = "टेक्स्ट/जावास्क्रिप्ट"> var पुस्तक = <?php इको
wp_json_encode($book) ?>; /* वर पुस्तक = { "शीर्षक": "वर्डप्रेस में सुरक्षा",

    "लेखक" => "स्टैक ओवरफ्लो",
}; */ </
स्क्रिप्ट>
```

या

```
<स्क्रिप्ट प्रकार='टेक्स्ट/जावास्क्रिप्ट'>
var शीर्षक = <?php इको wp_json_encode( $title ); ?>; var सामग्री = <?php इको
wp_json_encode($content); ?>; var comment_count = <?php echo wp_json_encode( $comment_count ); ?
>; </स्क्रिप्ट>
```

भागने के गुण

```
<इनपुट प्रकार = "पाठ" मान = "<?php इको esc_attr($_POST['उपयोगकर्ता नाम']); ?>" />
```

टेक्स्टएरिया में डेटा से बचें

```
<textarea><?php echo esc_textarea( $text ); ?></textarea>
```

अध्याय 51: वर्डप्रेस में सुरक्षा - सफ़ाई

वाक्य - विन्यास

- `sanitize_text_field(स्ट्रिंग $str)` •
- `sanitize_title(स्ट्रिंग $title, string $fallback_title, string $context)` •
- `sanitize_email(string $email)` •
- `sanitize_html_class(string $class, string $fallback)` •
- `sanitize_file_name(string $name)` •
- `sanitize_user(स्ट्रिंग $ उपयोगकर्ता नाम, बूलियन $ सख्त)`

टिप्पणियां

विकास करते समय सुरक्षा को हमेशा ध्यान में रखना चाहिए। सुरक्षा के बिना कोई ऐप विभिन्न हमलों जैसे SQL इंजेक्शन, XSS, CSRF, RFI आदि के लिए खुला रहता है जिससे गंभीर समस्याएं हो सकती हैं।

अविश्वसनीय डेटा कई स्रोतों (उपयोगकर्ताओं, तृतीय पक्ष साइटों, आपका अपना डेटाबेस!, ...) से आता है और इन सभी को इनपुट और आउटपुट दोनों पर मान्य करने की आवश्यकता होती है। (स्रोत: वर्डप्रेस कोडेक्स)

उपयोग और उद्देश्य के आधार पर डेटा को मान्य, स्वच्छ या एस्केप किया जाना चाहिए।

सत्यापित करने का अर्थ यह सुनिश्चित करना है कि आपने उपयोगकर्ता से जो डेटा अनुरोध किया है वह उनके द्वारा सबमिट किए गए डेटा से मेल खाता है। (स्रोत: वर्डप्रेस कोडेक्स)

उपयोगकर्ता डेटा स्वीकार करने के दृष्टिकोण में सेनिटाइजेशन थोड़ा अधिक उदार है। स्वीकार्य इनपुट की सीमा होने पर हम इन विधियों का उपयोग करने के लिए वापस आ सकते हैं। (स्रोत: वर्डप्रेस कोडेक्स)

बचने का मतलब है कि आपके पास पहले से मौजूद डेटा लेना और अंतिम उपयोगकर्ता के लिए इसे प्रस्तुत करने से पहले इसे सुरक्षित करने में मदद करना। (स्रोत: वर्डप्रेस कोडेक्स)

उदाहरण

टेक्स्ट फ़ील्ड को स्वच्छ करें

```
$शीर्षक = sanitize_text_field($_POST['शीर्षक']);
```

शीर्षक को स्वच्छ करें

लौटाए गए मान का उद्देश्य यूआरएल में उपयोग के लिए उपयुक्त होना है, न कि मानव-पठनीय शीर्षक के रूप में। इसके बजाय `sanitize_text_field` का उपयोग करें।

```
$new_url = sanitize_title($title);
```

ईमेल को सेनिटाइज़ करें

```
$sanitize_email = sanitize_email('admin@example.com!');
```

एचटीएमएल क्लास को सेनिटाइज़ करें

```
$post_class = sanitize_html_class( $post->post_title ); प्रतिध्वनि '<div class="' . $post_class . "'>;
```

फ़ाइल नाम को स्वच्छ करें

```
$incfile = sanitize_file_name($_REQUEST["फ़ाइल"]); include($incfile . ".php");
```

फ़ाइल नाम को साफ़ किए बिना कोई हमलावर आसानी से http://attacker_site/malicious_page पास कर सकता है इनपुट के रूप में और अपने सर्वर में जो भी कोड हो उसे निष्पादित करें।

उपयोगकर्ता नाम को स्वच्छ करें

```
$उपयोगकर्ता = sanitize_user ("हमलावर उपयोगकर्ता नाम<स्क्रिप्ट>कंसोल.लॉग(दस्तावेज़.कुकी)</स्क्रिप्ट>");
```

सेनिटाइज़ के बाद `$user` मान "हमलावर उपयोगकर्ता नाम" है

अध्याय 52: शॉर्टकोड

उदाहरण

शॉर्टकोड पंजीकृत करना

शॉर्टकोड कोड का एक छोटा टुकड़ा है जिसे वर्डप्रेस संपादक में जोड़ा जा सकता है और पेज प्रकाशित या पूर्वावलोकन होने के बाद कुछ अलग आउटपुट देगा।

अक्सर, थीम `function.php` फ़ाइल में शॉर्टकोड जोड़े जाते हैं, लेकिन यह एक अच्छा अभ्यास नहीं है क्योंकि थीम बदलने के बाद भी शॉर्टकोड के काम करते रहने की उम्मीद की जाती है। इसके बजाय, एक प्लगइन लिखें इस कार्यक्षमता को जोड़ने के लिए।

शॉर्टकोड पंजीकृत करने की संरचना है:

```
फ़ंक्शन new_shortcode($atts, $content = null){
    // यदि शॉर्टकोड में पैरामीटर की आवश्यकता है // पैरामीटर को कुछ एक्सट्रेक्ट के लिए डिफ़ॉल्ट पर सेट
    किया जा सकता है (shortcode_atts( array( 'param_one' => 'h1' ), $atts )); $shortcode =
    '<'. $param_one '>'. $content. '</'. $param_one. '>';
    $शॉर्टकोड लौटाएँ;

} // यह वह है जो वर्डप्रेस add_shortcode('demo-shortcode','new_shortcode') के साथ शॉर्टकोड को पंजीकृत
करता है;
```

वर्डप्रेस संपादक के अंदर, आप टाइप कर सकते हैं:

```
[डेमो-शॉर्टकोड पैराम_वन="एच2"]डेमो[/डेमो-शॉर्टकोड] // यदि इसका डिफ़ॉल्ट मान है तो आपको संपादक में पैराम_ऑन
डालने की आवश्यकता नहीं है। // इसे संपादक में रखने से डिफ़ॉल्ट ओवरराइड हो जाएगा
```

एक बार पृष्ठ प्रकाशित हो जाने पर, यह बदल जाएगा

```
<h2>डेमो</h2>
```

वर्डप्रेस बैकएंड में शॉर्टकोड का उपयोग करना

```
[फुटैग फू = "1 का मान" विशेषता-2 = "2 का मान"]
```

वर्डप्रेस एडमिन में हम वर्गाकार कोष्ठकों के अंदर शॉर्टकोड नाम लिखकर पूर्व परिभाषित शॉर्टकोड का उपयोग करते हैं और वैकल्पिक रूप से इसमें स्पेस द्वारा अलग करते हुए विशेषताएँ जोड़ते हैं।

नए शॉर्टकोड जोड़ना

```
फ़ंक्शन footag_func( $atts ) {
    वापसी "foo = {$atts['foo']}";
}
```

```
}  
add_shortcode('footag', 'footag_func');
```

प्लगइन्स में हम add_shortcode फ़ंक्शन का उपयोग करके शॉर्टकोड जोड़ सकते हैं।

शॉर्टकोड का उपयोग किसी भी वर्डप्रेस पेज या पोस्ट में केवल वर्गाकार कोष्ठक में संलग्न करके किया जा सकता है।

[फ़ुटैग]

PHP कोड के अंदर शॉर्टकोड का उपयोग करना (थीम और प्लगइन्स)

```
<?php echo do_shortcode("[footag foo='Hi! मैं एक foo आउटपुट हूँ']"); ?>
```

PHP का उपयोग करके शॉर्टकोड प्रिंट करने के लिए do_shortcode फ़ंक्शन का उपयोग करें और लौटाए गए मान को प्रतिध्वनित करें।

विजेट्स में शॉर्टकोड का उपयोग करना

```
add_filter('widget_text', 'shortcode_unautop'); add_filter('widget_text', 'do_shortcode' );यहां  
कोड दर्ज करें
```

विजेट में शॉर्टकोड सक्षम करने के लिए इसे प्लगइन या function.php फ़ाइल में जोड़ें। कोड पहले वर्डप्रेस टर्निंग लाइन को पैराग्राफ टैग में विभाजित होने से रोकता है और फिर शॉर्टकोड को विजेट्स के लिए पार्स करने देता है।

दो पंक्तियों का क्रम महत्वपूर्ण है।

अध्याय 53: विशेषता के साथ शॉर्टकोड

वाक्य - विन्यास

- `add_shortcode('your_short_code', 'your_function_name');`

पैरामीटर

पैरामीटर विवरण और उपयोग	
\$टैग	(स्ट्रिंग) (आवश्यक) पोस्ट सामग्री में खोजा जाने वाला शॉर्टकोड टैग डिफ़ॉल्ट: कोई नहीं
\$func	(कॉल करने योग्य) (आवश्यक) शॉर्टकोड मिलने पर चलने वाला हुक डिफ़ॉल्ट: कोई नहीं

टिप्पणियां

महत्वपूर्ण - अपनी विशेषताओं के लिए कैमलकेस या अपर-केस का उपयोग न करें

आप यहां [विशेषता के](#) साथ एक शॉर्टकोड तैयार कर सकते हैं

उदाहरण

शॉर्टकोड के उदाहरण

वर्डप्रेस शॉर्टकोड 2.5 में पेश किए गए थे

यहां शॉर्टकोड का उदाहरण दिया गया है

```
[बटन]
```

थीम में सीधे शॉर्टकोड का उपयोग करने के लिए आपको `do_shortcode()` का उपयोग करना होगा

```
<?php echo do_shortcode('[बटन]'); ?>
```

बटन को अनुकूलित करने के लिए, हम बस कुछ इस तरह जोड़ सकते हैं:

```
[बटन प्रकार = "ट्विटर"]
```

या इसे और भी बेहतर बनाने के लिए, हम एक संलग्न शॉर्टकोड का उपयोग कर सकते हैं:

```
[बटन प्रकार='ट्विटर']मुझे ट्विटर पर फ़ॉलो करें!/[बटन]
```

एक स्व-समापन शॉर्टकोड बनाना

सबसे सरल शॉर्टकोड स्वयं-समापन वाला है। हम अपने ट्विटर खाते के लिए एक सरल लिंक बनाने जा रहे हैं, और फिर इसे एक ब्लॉग पोस्ट में जोड़ देंगे। सारा कोड function.php में जाता है, जो /wp content/themes/your-theme/ में स्थित होता है। यदि आपके पास कोई नहीं है, तो बस इसे बनाएं और इसमें कोड डालें।

```
<?php
फ़ंक्शन बटन_शॉर्टकोड() { return '<a href='http://
twitter.com/rupomkhondaker' class='twitter-button'>मुझे ट्विटर पर फ़ॉलो करें!</a>';
}
add_shortcode('बटन', 'बटन_शॉर्टकोड');
?>
```

उपयोग: [बटन]

मापदंडों के साथ एक स्व-समापन शॉर्टकोड बनाना

मापदंडों के साथ एक स्व-समापन शॉर्टकोड बनाना

```
<?php
फ़ंक्शन बटन_शॉर्टकोड ($ प्रकार) {
    निकालें(shortcode_atts(
        सरणी
        ('प्रकार' => 'मूल्य'), $प्रकार));

    // जांचें कि उपयोगकर्ता ने किस प्रकार का स्विच ($ प्रकार) दर्ज किया है {

        मामला 'ट्विटर':
            वापसी '<a href='http://twitter.com/rupomkhondaker' class='twitter-button'>Folw
में ट्विटर पर!</a>'; तोड़ना;

        मामला 'आरएसएस':
            return '<a href='http://example.com/rss' class='rss-button'>की सदस्यता ले
फ़ीड!</a>'
            तोड़ना;
    }

} add_shortcode('बटन', 'बटन_शॉर्टकोड');
?>
```

अब आप अपने शॉर्टकोड में टाइप को परिभाषित करके चुन सकते हैं कि कौन सा बटन प्रदर्शित करना है।

```
[बटन प्रकार='ट्विटर'] [बटन प्रकार='आरएसएस']
```

एक संलग्न शॉर्टकोड बनाना

शॉर्टकोड संलग्न करना

संलग्न शॉर्टकोड आपको बीबीसीओडी की तरह, अपने शॉर्टकोड के भीतर सामग्री एम्बेड करने की अनुमति देता है, यदि आपने कभी इसका उपयोग किया है।

```
<?php
फ़ंक्शन बटन_शॉर्टकोड( $attr, $content = null ) रिटर्न '<a href='http://twitter.com/filipstefansson'
क्लास='twitter-button'>'। $सामग्री. '</a>';

}
add_shortcode('बटन', 'बटन_शॉर्टकोड');
?>
```

इस शॉर्टकोड का उपयोग करने के लिए, आप उस टेक्स्ट को एम्बेड करें जिसे आप इस तरह उपयोग करना चाहते हैं:

```
[बटन]मुझे ट्विटर पर फ़ॉलो करें![/बटन]
```

इस बटन को और भी बेहतर बनाने के लिए, हम पैरामीटर जोड़ सकते हैं जैसे हमने पिछले उदाहरण में किया था।

```
<?php
फ़ंक्शन बटन_शॉर्टकोड ( $atts, $content = null )
निकालें(shortcode_atts(संरणी(
'खाता' => 'खाता',
'शैली' => 'शैली' ), $atts ));

वापसी '<a href='http://twitter.com/' . esc_attr($account) . esc_attr($style) .. $content. '</a>';
''वर्ग='ट्विटर-बटन'
">"

}
add_shortcode('बटन', 'बटन_शॉर्टकोड');
?>
```

उपयोग:

```
[बटन अकाउंट='रुपोमखोंडाकर' स्टाइल='सिंपल']ट्विटर पर मुझे फ़ॉलो करें![/बटन]
```

विजेट्स में शॉर्टकोड

डिफ़ॉल्ट रूप से, वर्डप्रेस साइडबार विजेट के भीतर शॉर्टकोड का समर्थन नहीं करता है। यह केवल पोस्ट, पेज या कस्टम पोस्ट प्रकार की सामग्री के भीतर शॉर्टकोड का विस्तार करता है। साइडबार विजेट में शॉर्टकोड समर्थन जोड़ने के लिए, आप एक प्लगइन इंस्टॉल कर सकते हैं, या नीचे दिए गए कोड का उपयोग कर सकते हैं:

```
add_filter('widget_text', 'shortcode_unautop'); add_filter('widget_text', 'do_shortcode');
```

इन पंक्तियों को इसी क्रम में जोड़ा जाना जरूरी है। पहली पंक्ति वर्डप्रेस को लाइन ब्रेक को पैराग्राफ टैग में बदलने से रोकती है, क्योंकि यह शॉर्टकोड को काम करने से रोकती है। दूसरी पंक्ति वह है जो शॉर्टकोड को काम में लाती है।

अध्याय 54: लघुकोड

उदाहरण

शॉर्टकोड परिचय

शॉर्टकोड तब उपयोगी होते हैं जब आप इनलाइन में अधिक जटिल तत्व जोड़ने में सक्षम होना चाहते हैं सामान्य सामग्री संपादक.

इसके सरलतम रूप में एक शॉर्टकोड इस तरह दिखेगा:

```
फ़ंक्शन my_shortcode(){
    वापसी "यह एक शॉर्टकोड है";
}
add_shortcode('my_shortcode', 'my_shortcode');
```

यह "यह एक शॉर्टकोड है" टेक्स्ट को आउटपुट करेगा और आप इसे [my_shortcode] लिखकर उपयोग करेंगे कंटेंट एडिटर।

बटन शॉर्टकोड

यहां एक साधारण बटन शॉर्ट कोड का उदाहरण दिया गया है:

```
<?php
फ़ंक्शन my_button_shortcode( $atts ) {

    // इनपुट विशेषताओं को पार्स करें और इसके लिए डिफ़ॉल्ट मान निर्दिष्ट करें
    // विशेषताएँ जो शॉर्टकोड पर निर्दिष्ट नहीं हैं
    $a = शॉर्टकोड_एट्स(सरणी(
        'आईडी' => '',
        'यूआरएल' => '#',
        'वर्ग' => '',
        'पाठ' => ''
    ), $atts );

    // बेहतर पहुंच के लिए एंकर टैग खोलें और रोल=बटन जोड़ें
    $btn_html = '<a भूमिका="बटन"';

    // href(लिंक) विशेषता जोड़ें
    $btn_html .= ' href="' . $a['url'] . '""';

    // यदि निर्दिष्ट हो तो आउटपुट में आईडी विशेषता जोड़ें
    अगर ( !खाली ($a['id']) ) {
        $btn_html .= ' id="' . $a['id'] . '""';
    }

    $btn_classes = 'बटन';

    // आउटपुट में क्लास विशेषता जोड़ें
    $btn_html .= ' वर्ग="बटन' . ( !empty($a['class']) ? $a['class'] : '' ) . '""';

    // ओपनिंग एंकर टैग बंद करें
```

```
$btn_html .= '>'.  
    // बटन टेक्स्ट जोड़ें $a['text']।  
  
    // क्लोजिंग एंकर टैग जोड़ें '</a>'।"\n";  
  
$btn_html लौटाएँ;  
}  
add_shortcode('बटन', 'my_button_shortcode');
```

इस शोर्टकोड का उपयोग संपादक में [बटन url = "/ my-other-page" id = "my-other-page-button" class = "dark" text = "Click me!"] टाइप करके किया जा सकता है और आउटपुट देगा निम्नलिखित HTML:

```
<a भूमिका = "बटन" href = "/my-other-page" id = "my-other-page-button" class = "बटन डार्क">मुझे क्लिक करें!</a>
```

अध्याय 55: साइडबार

वाक्य - विन्यास

- रजिस्टर_साइडबार(\$args) •
get_sidebar(स्ट्रिंग \$नाम = शून्य)

पैरामीटर

पैरामीटर विवरण	
\$args	(स्ट्रिंग ऐरे) (वैकल्पिक) के आधार पर साइडबार बनाता है नाम और आईडी values
\$नाम	*(स्ट्रिंग) (वैकल्पिक) विशेष साइडबार का नाम। डिफ़ॉल्ट मान: शून्य

टिप्पणियां

तर्क विकल्प हैं:

- नाम - साइडबार नाम (डिफ़ॉल्ट: स्थानीयकृत 'साइडबार' और संख्यात्मक आईडी)।
- आईडी - साइडबार आईडी - सभी लोअरकेस में होनी चाहिए, बिना किसी रिक्त स्थान के (डिफ़ॉल्ट: एक संख्यात्मक स्वचालित वृद्धिशील आईडी)। यदि आप आईडी तर्क मान सेट नहीं करते हैं, तो आपको संस्करण 4.2 से शुरू होकर, डिबग मोड में E_USER_NOTICE संदेश मिलेंगे। विवरण - साइडबार क्या/कहां है इसका टेक्स्ट विवरण। विजेट प्रबंधन स्क्रीन पर
- दिखाया गया। (2.9 से) (डिफ़ॉल्ट: खाली) वर्ग - उपस्थिति में साइडबार को असाइन करने के लिए सीएसएस वर्ग -> विजेट व्यवस्थापक पृष्ठ। यह क्लास केवल वर्डप्रेस विजेट एडमिन पेज के स्रोत में दिखाई देगी। इसे
- आपकी वेबसाइट के फ्रंट एंड में शामिल नहीं किया जाएगा. नोट: मान साइडबार को वर्ग मान से जोड़ा जाएगा। उदाहरण के लिए, ताल के एक वर्ग के परिणामस्वरूप साइडबार-ताल का वर्ग मान आएगा। (डिफ़ॉल्ट: खाली). before_widget - प्रत्येक विजेट के पहले रखने के लिए HTML (डिफ़ॉल्ट: <li id='%1\$s' class='widget %2\$s'>) नोट: वेरिएबल प्रतिस्थापन के लिए sprintf का उपयोग करता है • After_widget - प्रत्येक विजेट के बाद रखने के लिए HTML (डिफ़ॉल्ट) : \n) • before_title - HTML को प्रत्येक शीर्षक से पहले रखें (डिफ़ॉल्ट: <h2 class='widgettitle'>)।
-
- After_title - HTML को प्रत्येक शीर्षक के बाद रखें (डिफ़ॉल्ट: </h2>\n)।

उदाहरण

साइडबार पंजीकृत करना

अपने function.php में आप इस कोड के साथ नए साइडबार पंजीकृत कर सकते हैं

```
/**
```

* साइडबार पंजीकृत करता है

*

* @param सरणी डिफॉल्ट या निर्दिष्ट सरणी मानों के साथ सरणी * @1.0.0 से */

```
यदि ( function_exists( 'रजिस्टर_साइडबार' ) रजिस्टर_साइडबार( सारणी (
    ) {
    'नाम' => esc_html__( 'प्राथमिक साइडबार', 'माइथीम' ), => 'प्राथमिक-विजेट-क्षेत्र', 'विवरण' =>
    'पहचान' esc_html__( 'प्राथमिक विजेट क्षेत्र', 'माइथीम' ),
    '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>',

    ) );

    रजिस्टर_साइडबार ( सारणी (
    'नाम' => esc_html__( 'माध्यमिक साइडबार', 'माइथीम' ), => 'माध्यमिक-विजेट-क्षेत्र', 'विवरण' =>
    'पहचान' esc_html__( 'माध्यमिक विजेट क्षेत्र', 'माइथीम' ),
    '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>',

    ) );
}
```

आप जितने चाहें उतने साइडबार जोड़ सकते हैं।

साइडबार प्राप्त करें

आप इसे विभिन्न टेम्पलेट्स पर कॉल करने के लिए थीम में अपनी स्वयं की साइडबार फ़ाइल भी बना सकते हैं। मौजूदा थीम के साइडबार.php को कॉपी और पेस्ट करें और नाम बदलें (यानी साइडबार-बुक.php)

टेम्पलेट में आप `get_sidebar('book')` का उपयोग करके इस साइडबार को कॉल कर सकते हैं। इसका उपयोग करके आप अलग-अलग पेज पर अलग-अलग साइडबार को कॉल कर सकते हैं।

अध्याय 56: साइट स्थानांतरण

वाक्य - विन्यास

- OLD_SITE - माइग्रेट की जा रही पुरानी साइट (उदाहरण: <http://localhost/example>) • NEW_SITE - नई साइट जिस पर माइग्रेट करना है (उदाहरण: <https://example.com>)

उदाहरण

नए यूआरएल के साथ तालिकाओं को अपडेट किया जा रहा है

```
अद्यतन wp_options सेट option_value = प्रतिस्थापित करें (option_value, 'OLD_SITE', 'NEW_SITE') जहां option_name = 'home' या option_name = 'siteurl';
```

```
अद्यतन wp_posts सेट गाइड = प्रतिस्थापित करें (गाइड, 'OLD_SITE', 'NEW_SITE');
```

```
अद्यतन wp_posts SET post_content = प्रतिस्थापित करें (post_content, 'OLD_SITE', 'NEW_SITE');
```

अध्याय 57: वर्गीकरण

वाक्य - विन्यास

- रजिस्टर_टैक्सोनॉमी (\$वर्गीकरण, \$ऑब्जेक्ट_प्रकार, \$आर्ग्स);

पैरामीटर

पैरामीटर	विवरण
\$वर्गीकरण	(स्ट्रिंग) (आवश्यक) वर्गीकरण का नाम। नाम में केवल शामिल होना चाहिए लोअरकेस अक्षर और अंडरस्कोर अक्षर, और 32 से अधिक नहीं होना चाहिए वर्ण लंबे (डेटाबेस संरचना प्रतिबंध)।
\$ऑब्जेक्ट_प्रकार	(सरणी/स्ट्रिंग) (आवश्यक) वर्गीकरण वस्तु के लिए वस्तु प्रकार का नाम। ऑब्जेक्ट-प्रकार अंतर्निहित पोस्ट प्रकार या कोई भी कस्टम पोस्ट प्रकार हो सकता है दर्ज कराई।
\$args	(सरणी/स्ट्रिंग) (वैकल्पिक) तर्कों की एक सारणी।

उदाहरण

शैलियों के लिए वर्गीकरण दर्ज करने का उदाहरण

```
<?php
// init क्रिया से जुड़े और सक्रिय होने पर create_book_taxonomies को कॉल करें
add_action('init', 'create_book_taxonomies', 0);

// पोस्ट प्रकार "पुस्तक" के लिए वर्गीकरण शैली बनाएं
फ़ंक्शन create_book_taxonomies() {
    // नई वर्गीकरण जोड़ें, इसे श्रेणीबद्ध बनाएं (श्रेणियों की तरह)
    $लेबल = सरणी(
        'नाम' => _x('शैलियाँ', 'वर्गीकरण सामान्य नाम'),
        'एकवचन_नाम' => _x('शैली', 'वर्गीकरण एकवचन नाम'),
        'खोज_आइटम' => __('शैलियाँ खोजें'),
        'सभी वस्तुएं' => __('सभी शैलियाँ'),
        'parent_item' => __('मूल शैली'),
        'parent_item_colon' => __('जनक शैली: '),
        'संपादित_आइटम' => __('शैली संपादित करें'),
        'अद्यतन_आइटम' => __('अपडेट शैली'),
        'नया_आइटम_जोड़ें' => __('नई शैली जोड़ें'),
        'नया_आइटम_नाम' => __('नई शैली का नाम'),
        'मेनू_नाम' => __('शैली'),
    );

    $args = सरणी(
        'पदानुक्रमित' => सच,
        'लेबल' => $लेबल,
        'show_ui' => सच,
    );
}
```

```

        'show_admin_column' => सत्य,
        'query_var' 'फिर से लिखें' => सच,
        => सरणी('स्लग' => 'शैली'),
    );

    रजिस्टर_टैक्सोनॉमी ('शैली', सरणी ('पुस्तक'), $args);

}
?>

```

आप थीम के function.php टेम्पलेट फ़ाइल में कस्टम टैक्सोनॉमी को परिभाषित कर सकते हैं :

पेज में श्रेणी जोड़ें

आप नीचे दिए गए कोड का उपयोग करके उसी कस्टम निर्मित वर्गीकरण को पोस्ट प्रकार पृष्ठ में भी जोड़ सकते हैं।

```

फ़ंक्शन add_taxonomies_to_pages() { रजिस्टर_टैक्सोनॉमी_फॉर_ऑब्जेक्ट_टाइप
    ('शैली', 'पेज');
}
add_action( 'init', 'add_taxonomies_to_pages' );

```

उपरोक्त कोड को अपनी थीम की function.php फ़ाइल में जोड़ें। उसी तरह आप पोस्ट टाइप पेज में कस्टम या डिफॉल्ट पोस्ट_टैग जोड़ सकते हैं।

कस्टम टैक्सोनॉमी क्वेरी का उपयोग करके पेज प्राप्त करने के लिए उसी फ़ाइल में नीचे दिया गया कोड जोड़ना होगा।

```

अगर ( ! is_admin() ) {
    add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

फ़ंक्शन श्रेणी_और_टैग_आर्काइव्स ( $wp_query ) $my_post_array = array('पेज'); यदि {
    ( $wp_query->प्राप्त करें( 'श्रेणी_नाम' ) || $wp_query->प्राप्त
    करें( 'बिल्ली' ) $wp_query->सेट( 'post_type', $my_post_array );
}

```

पेजों में श्रेणियाँ और टैग जोड़ें और उन्हें क्लास के रूप में डालें

```

// पेजों पर टैग और श्रेणियाँ जोड़ें

फ़ंक्शन add_taxonomies_to_pages() {
    रजिस्टर_टैक्सोनॉमी_फॉर_ऑब्जेक्ट_टाइप ('पोस्ट_टैग', 'पेज'); रजिस्टर_टैक्सोनॉमी_फॉर_ऑब्जेक्ट_टाइप ('श्रेणी', 'पेज'); }

add_action( 'init', 'add_taxonomies_to_pages' );
यदि ! is_admin() {
    ( add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

फ़ंक्शन श्रेणी_और_टैग_आर्काइव्स ( $wp_query ) $my_post_array = array('पोस्ट', 'पेज'); {

यदि ( $wp_query->प्राप्त करें( 'श्रेणी_नाम' ) || $wp_query->प्राप्त करें( 'बिल्ली' ) $wp_query->सेट( 'post_type', $my_post_array ); )

```

```

यदि ($wp_query->प्राप्त करें('टैग') $wp_query-
>सेट('post_type', $my_post_array);
}

// टैग और श्रेणियों को <body> में वर्ग के रूप में जोड़ें

फ़ंक्शन add_categories_and_tags( $classes = if( is_page() ) $categories = '' ) {
    get_the_category();
    {
        foreach( $श्रेणी के रूप में $श्रेणी ) $वर्ग[] = 'श्रेणी-'. $श्रेणी->स्लग;
        {
            }
        $टैग = get_the_tags();
        foreach( $tags को $tag के रूप में ) $classes[] {
            = 'tag-'. $tag->slug;
        }
    }
}
$कक्षाएँ लौटाएँ;
}
add_filter('body_class', 'add_categories_and_tags');

```

पेजों में श्रेणियाँ और टैग जोड़ें और क्लास के रूप में डालें

आप इस कोड को अपनी कस्टम function.php फ़ाइल में जोड़ सकते हैं:

```

// पेजों पर टैग और श्रेणियाँ जोड़ें

फ़ंक्शन add_taxonomy_to_pages() {
    रजिस्टर_टैक्सोनॉमी_फॉर_ऑब्जेक्ट_टाइप ('पोस्ट_टैग', 'पेज');
    रजिस्टर_टैक्सोनॉमी_फॉर_ऑब्जेक्ट_टाइप ('श्रेणी', 'पेज');
}
add_action( 'init', 'add_taxonomy_to_pages' );

यदि ( ! is_admin() ) {
    add_action( 'pre_get_posts', 'category_and_tag_archives' );
}

फ़ंक्शन श्रेणी_और_टैग_आर्काइव्स ( $wp_query ) $my_post_array = array('पोस्ट', 'पेज'); {

    यदि ( $wp_query->प्राप्त करें( 'श्रेणी_नाम' ) || $wp_query->प्राप्त करें( 'बिल्ली' ) $wp_query->सेट( 'post_type', $my_post_array );

    यदि ( $wp_query->प्राप्त करें('टैग') $wp_query-
    >सेट('post_type', $my_post_array);
}

// टैग और श्रेणियों को <body> में वर्ग के रूप में जोड़ें

फ़ंक्शन add_categories_and_tags( $classes = if( is_page() ) $categories = '' ) {
    get_the_category();
    {
        foreach( $श्रेणी के रूप में $श्रेणी ) $वर्ग[] = 'श्रेणी-'. $श्रेणी->स्लग;
        {
            }
        $टैग = get_the_tags();
        foreach( $tags को $tag के रूप में ) $classes[] {
            = 'tag-'. $tag->slug;
        }
    }
}

```

```
    }  
  }  
  $कक्षाएँ लौटाएँ;  
}  
add_filter('body_class', 'add_categories_and_tags');
```

बढ़िया काम करता है, वर्डप्रेस 4.8 में परीक्षण किया गया

अध्याय 58: टेम्पलेट पदानुक्रम

टिप्पणियां

वर्डप्रेस में डिबगिंग के लिए प्लगइन्स:

- <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/>

उदाहरण

परिचय

जब आप वर्डप्रेस थीम बना रहे हों तो सीखने वाली सबसे महत्वपूर्ण चीजों में से एक थीम के लिए वर्डप्रेस टेम्पलेट पदानुक्रम है। टेम्पलेट पदानुक्रम परिभाषित करता है कि प्रत्येक अनुरोध के लिए कौन सी टेम्पलेट फ़ाइल लोड की जाएगी और किस क्रम में। यदि पदानुक्रम में पहला टेम्पलेट मौजूद नहीं है, तो वर्डप्रेस अगले टेम्पलेट को लोड करने का प्रयास करेगा और इसी तरह जब तक आप `Index.php` पर नहीं पहुँच जाते।

टेम्पलेट पदानुक्रम का विस्तार से वर्णन करने के लिए निश्चित रूप से पूर्ण संरचना के साथ एक छवि का उपयोग करना सबसे अच्छा तरीका है:

Archive Page



Singular Page

Single Post Page

Static Page



Site Front Page

front-page.php



Category-books.php का उपयोग केवल स्लग बुक वाली श्रेणी के लिए किया जाएगा। एक अन्य उदाहरण पेज-\$id.php है जो केवल एक विशिष्ट आईडी वाले पेज को लक्षित करता है, उदाहरण के लिए पेज-41.php केवल आईडी 41 वाले पेज को लक्षित करेगा।

विशिष्ट प्रकार या पोस्ट को लक्षित करने वाले टेम्पलेट के बाद हमें सामान्य प्रकार के टेम्पलेट मिलते हैं, जैसे सभी संग्रह पृष्ठों के लिए Archive.php या सभी पृष्ठों के लिए page.php। लेकिन याद रखें, उनका उपयोग केवल तभी किया जाएगा जब वर्तमान पृष्ठ पदानुक्रम में उच्चतर किसी भी टेम्पलेट से मेल नहीं खाता है।

अंत में, यदि वर्डप्रेस को टेम्पलेट डायरेक्टरी में कोई मेल खाने वाला टेम्पलेट नहीं मिल पाता है, तो अंतिम फॉलबैक हमेशा Index.php होता है जो कि वर्डप्रेस थीम में एकमात्र आवश्यक टेम्पलेट फ़ाइल है।

डिबगिंग

पदानुक्रम को डीबग करते समय खो जाना आसान है। आप PHP के अंतर्निहित कमांड का उपयोग कर सकते हैं

डिबग_बैकट्रेस.

अगले स्निपेट को किसी भी टेम्पलेट के अंदर रखें जिसे आप डीबग करना चाहते हैं और जनरेट किए गए पेज को देखें:

```
<!--  
<?php print_r(debug_backtrace()) ?>  
-->
```

अध्याय 59: टेम्पलेट_शामिल

पैरामीटर

पैरामीटर	स्पष्टीकरण
\$टेम्पलेट	फ़िल्टर में एक पैरामीटर पास करता है, \$template पोस्ट प्रकार के लिए उपयुक्त फ़ाइल का वर्तमान पथ है जैसा कि सक्रिय चाइल्ड थीम या पैरेंट थीम में पाया जाता है (यदि कोई चाइल्ड थीम मौजूद नहीं है या चाइल्ड थीम में कम रैंक वाले टेम्पलेट हैं। अधिक विवरण के लिए वर्डप्रेस टेम्पलेट पदानुक्रम देखें)।

टिप्पणियां

संशोधित न करने पर भी आपको \$template वापस करना होगा। यदि यह आपको भ्रमित करता है, तो ऐसे उदाहरण देखें जहां कोड में `apply_filter()` का उपयोग किया गया है

आपको बाद में उपयोग के लिए यहां वेरिएबल सेट नहीं करना चाहिए, इसके लिए बेहतर हुक हैं।

इस फ़िल्टर के लिए एक उपयोगी प्रोग्राम प्रवाह है:

1. जांचें कि \$टेम्पलेट में हमारे कस्टम पोस्ट प्रकार का नाम शामिल है -->टेम्पलेट पदानुक्रम!! 2. यदि नहीं, तो उपयुक्त फ़ाइलों के लिए हमारे प्लगइन को खोजें -> फ़ाइलों के लिए फ़ोल्डरों में खोज करने के बजाय विशिष्ट फ़ाइलों को इंगित करना बेहतर है। अधिक कुशल। लेकिन पूरी तरह से डेवलपर पर निर्भर है।
3. टेम्पलेट वापस करें.

उदाहरण

सरल उदाहरण

यह फ़िल्टर बहुत उपयोगी है. डेवलपर्स के लिए आम समस्याओं में से एक यह है कि उनके द्वारा विकसित प्लगइन्स में टेम्पलेट्स को कैसे शामिल किया जाए।

वर्डप्रेस द्वारा wp पदानुक्रम का उपयोग करके सक्रिय चाइल्ड/पैरेंट थीम में उपयुक्त टेम्पलेट का पता लगाने के तुरंत बाद फ़िल्टर लागू किया जाता है।

यह परिभाषित करने में सावधानी बरतें कि आप टेम्पलेट पथ को कब संशोधित करना चाहते हैं। नीचे दिए गए उदाहरण में, कोड यह देखने के लिए जाँच करता है कि क्या वर्तमान पृष्ठ हमारे कस्टम पोस्ट प्रकार सीपीटी का एकल दृश्य है।

आरंभ करने के लिए पर्याप्त सरल उदाहरण!

```
add_filter('template_include', 'custom_function');
```

```
फ़ंक्शन कस्टम_फ़ंक्शन ($ टेम्पलेट) {
```

```
//एकल पोस्ट टेम्पलेट बदलें...
```

```
अगर( is_singular('cpt') ){ $template= 'path/  
to/another/template_file';  
}
```

```
वापसी $ टेम्पलेट;
```

```
}
```

अधिक विज्ञापन उदाहरण

```
add_filter('template_include', 'custom_function');
```

```
फ़ंक्शन कस्टम_फ़ंक्शन ($ टेम्पलेट) {
```

```
/*  
* यह उदाहरण थोड़ा अधिक उन्नत है.  
* यह यह देखने के लिए जाँच करेगा कि क्या $template में पथ में हमारा पोस्ट-प्रकार शामिल है।  
* यदि ऐसा होता है, तो थीम में एक उच्च स्तरीय टेम्पलेट होता है, उदाहरण के लिए सिंगल-सीपीटी.php यदि नहीं, तो हम फ़ाइल के लिए प्लगइन पैरेंट फ़ोल्डर में देखते हैं।  
* उदाहरण के लिए सिंगल-cpt.php  
*/
```

```
//यह देखने के लिए जांचें कि पोस्ट प्रकार फ़ाइल नाम में है (उच्च प्राथमिकता फ़ाइल) //यदि है तो टेम्पलेट लौटाएं!
```

```
वैश्विक $पोस्ट;
```

```
if( strpos($template, 'single-'. $post->post_type.'.php' ) !== false && strpos($template, 'archive-'. $post->post_type.'.php' ) !== गलत ){वापसी $टेम्पलेट;
```

```
}
```

```
$plugin_path = 'var/etc/wp-content/plugins/plugin'; //यहां अपना तर्क शामिल करें...
```

```
if( is_singular($post->post_type) && file_exists($plugin_path.'/single-'. $post->post_type.'.php' ) ){  
    $template= $plugin_path.'/single-'. $post->post_type.'.php';  
} अन्यथा ( is_archive($post->post_type) && file_exists($plugin_path.'/archive-'. $post->post_type.'.php' ) ) $template= $plugin_path.'/archive-'. $post->post_type.'.php';  
    {
```

```
}
```

```
वापसी $ टेम्पलेट;
```

```
}
```

अध्याय 60: \$wpdb ऑब्जेक्ट

टिप्पणियां

\$wpdb ऑब्जेक्ट को संदर्भित करने के दो तरीके हैं। ऑब्जेक्ट के वैश्विक उदाहरण पर कार्य करने के लिए सबसे पहले PHP कीवर्ड ग्लोबल का उपयोग करना है।

```
वैश्विक $wpdb; प्रतिध्वनि
$wpdb->उपसर्ग;
// डेटाबेस के लिए उपसर्ग आउटपुट करता है
```

\$wpdb ऑब्जेक्ट का उपयोग करने का दूसरा तरीका PHP के \$GLOBALS सुपर ग्लोबल वेरिएबल को संदर्भित करना है।

```
प्रतिध्वनि $ग्लोबल्स['wpdb']->उपसर्ग;
// यह डेटाबेस के लिए उपसर्ग भी आउटपुट करेगा
```

दूसरे तरीके को हतोत्साहित किया जाता है क्योंकि इसे सर्वोत्तम अभ्यास नहीं माना जा सकता है।

उदाहरण

एक वेरिएबल का चयन करना

सबसे बुनियादी रूप में, SQL क्वेरी में ऑब्जेक्ट की get_var विधि को कॉल करके तालिका से एकल चर का चयन करना संभव है।

```
वैश्विक $wpdb; $user =
$wpdb->get_var( "$wpdb->उपयोगकर्ताओं से आईडी चुनें जहां user_email='foo@bar.com'");
```

यह ध्यान रखना बहुत महत्वपूर्ण है कि हमलों से बचाने के लिए प्रश्नों में उपयोग किए गए किसी भी अविश्वसनीय मान से बचना चाहिए। यह ऑब्जेक्ट की तैयार विधि का उपयोग करके किया जा सकता है।

```
वैश्विक $wpdb; $ईमेल =
$_POST['ईमेल']; $user = $wpdb-
>get_var( $wpdb->तैयार करें( "$wpdb-
>उपयोगकर्ताओं से आईडी चुनें जहां user_email=%s", $email )
);
यदि( !is_null( $उपयोगकर्ता )){
    प्रतिध्वनि $उपयोगकर्ता; }
अन्यथा { इको
    'उपयोगकर्ता नहीं मिला';
}
```

एकाधिक पंक्तियों का चयन करना

आप डेटाबेस से एकाधिक पंक्तियाँ प्राप्त करने के लिए get_results का उपयोग कर सकते हैं।

```
वैश्विक $wpdb;
```

```
$userTable = $wpdb->prefix."users";
```

```
$selectUser = $wpdb->get_results("SELECT * FROM $userTable");
```

यह सभी उपयोगकर्ताओं की सूची को सरणी में दिखाएगा।

अध्याय 61: एडमिन बार (उर्फ "टूलबार")

टिप्पणियां

वर्डप्रेस एडमिन टूलबार संस्करण 3.1 में जोड़ा गया था और इसमें सामान्य के लिंक शामिल हैं

प्रशासनिक कार्यों के साथ-साथ उपयोगकर्ता की प्रोफ़ाइल और अन्य वर्डप्रेस जानकारी के लिंक।

हालाँकि, कई साइट मालिक सभी लॉग-इन उपयोगकर्ताओं को डिफ़ॉल्ट रूप से टूलबार दिखाना नापसंद करते हैं और/या इसमें अपने स्वयं के विकल्प जोड़ना चाहते हैं।

उदाहरण

एडमिनिस्ट्रेटर को छोड़कर बाकी सभी से एडमिन टूलबार को हटाना

व्यवस्थापक उपयोगकर्ता स्तर को छोड़कर सभी से इसे हटाने के लिए निम्नलिखित कोड को function.php में जोड़ें:

```
add_action('after_setup_theme', 'no_admin_bar');

function no_admin_bar() {
    अगर (!current_user_can('एडमिनिस्ट्रेटर') && !is_admin()) {
        show_admin_bar(गलत);
    }
}
```

फ़िल्टर का उपयोग करके एडमिन टूलबार को हटाना

एडमिन बार को छिपाने का दूसरा तरीका जोड़ना है

```
यदि ( !current_user_can( 'manage_options' ) ) {
    add_filter('show_admin_bar', '_return_false' , 1000 );
}
```

जिन उपयोगकर्ताओं के पास सेटिंग्स पृष्ठ तक पहुंचने का विशेषाधिकार नहीं है, वे एडमिन बार नहीं देख पाएंगे।

एडमिन बार से वर्डप्रेस लोगो कैसे हटाएं

वर्डप्रेस एडमिन बार या टूलबार से आइटम हटाने के लिए डेवलपर्स admin_bar_menu क्रिया का उपयोग कर सकते हैं।

```
add_action('admin_bar_menu', 'remove_wp_logo_from_admin_bar', 999); फ़ंक्शन रिमूव_wp_logo_from_admin_bar( $wp_admin_bar )
$wp_admin_bar->remove_node('wp-logo');
}
```

उपरोक्त कोड एडमिन बार से वर्डप्रेस लोगो को हटा देता है। आपको बस कोड पेस्ट करना है

आपके function.php फ़ाइल के अंदर।

रिमूव_नोड विधि में पारित पैरामीटर उस नोड की आईडी है जिसे आप हटाना चाहते हैं। आईडी को टूलबार के साथ वर्डप्रेस पेज के HTML स्रोत कोड में पाया जा सकता है। उदाहरण के लिए, टूलबार में बाईं ओर वर्डप्रेस लोगो के लिए ली तत्व की आईडी "wp-admin-bar-wp-logo" है:

```
<li id='wp-admin-bar-wp-logo' class='menuitem'> ... </li>
```

नोड की आईडी प्राप्त करने के लिए ली की आईडी से "wp-admin-bar-" हटाएं। इस उदाहरण से नोड आईडी "wp-लोगो" है।

आप अपने एडमिन बार पर विभिन्न आइटमों या नोड्स की नोड आईडी का पता लगाने के लिए ब्राउज़र इंस्पेक्टर टूल का उपयोग कर सकते हैं।

एडमिन लॉगिन पेज पर अपना कस्टम लोगो और कस्टम लिंक जोड़ें

आप डिफ़ॉल्ट वर्डप्रेस लोगो को बदलने के लिए अपना स्वयं का लोगो और लिंक जोड़ने के लिए नीचे दिए गए हुक जोड़ सकते हैं।

कस्टम लोगो जोड़ने के लिए

```
फ़ंक्शन कस्टम_लॉगिन_लोगो() { इको '<स्टाइल टाइप='टेक्स्ट/सीएसएस'>  
एच1 ए { बैकग्राउंड-इमेज: यूआरएल('.get_bloginfo('टेम्पलेट_डायरेक्टरी').'/  
इमेजेज/कस्टम-लोगो.पीएनजी) !महत्वपूर्ण; पृष्ठभूमि-आकार : 100% !महत्वपूर्ण; चौड़ाई: 300px !महत्वपूर्ण; ऊँचाई : 100px !महत्वपूर्ण;} </style>; } add_action('login_head', 'custom_login_logo');
```

कस्टम लोगो लिंक जोड़ने के लिए

```
add_filter('login_headerurl', 'custom_loginlogo_url'); फ़ंक्शन कस्टम_लॉगिनलोगो_यूआरएल($यूआरएल) {  
  
    घर लौटें यूआरएल();  
}
```

अध्याय 62: लूप (मुख्य वर्डप्रेस लूप)

उदाहरण

मूल वर्डप्रेस लूप संरचना

हर बार जब वर्डप्रेस पेज लोड करता है, तो यह मुख्य लूप चलाएगा।

लूप उस पृष्ठ से संबंधित सभी तत्वों को पुनरावृत्त करने का तरीका है जिस पर आप वर्तमान में हैं।

मुख्य लूप वैश्विक WP_Query ऑब्जेक्ट पर काम करेगा। क्वेरी में एक वैश्विक विधि है `_posts()`, जो हमें सभी परिणामों के माध्यम से लूप करने की अनुमति देती है। अंत में लूप के अंदर आप `the_post()` विधि (एक वैश्विक फ़ंक्शन के रूप में भी) को कॉल कर सकते हैं, जो लूप के अंदर वर्तमान पोस्ट पर वैश्विक पोस्ट ऑब्जेक्ट सेट करता है, और पोस्टडेटा को वर्तमान पोस्ट पर सेट करता है। इसके लिए धन्यवाद आप सीधे लूप के अंदर `the_title`, `the_content`, `the_author` (टेम्पलेट टैग) जैसे फ़ंक्शन को कॉल कर सकते हैं।

उदाहरण के लिए यदि आप पोस्ट सूचियों पर हैं, तो मुख्य लूप में सभी पोस्ट के साथ एक क्वेरी ऑब्जेक्ट होगा।

यदि आप एकल पोस्ट (या पेज) पर हैं, तो इसमें उस एकल पोस्ट (पेज) के साथ एक क्वेरी शामिल होगी जो आप वर्तमान में हैं पर।

```
अगर (हैव_पोस्ट्स()) जबकि (हैव_पोस्ट्स()) :
    पोस्ट();
    var_dump($पोस्ट); अंत तक; अगर
अंत;
```

वैकल्पिक लूप सिंटैक्स

आप इस तरह घुंघराले ब्रैकेट वाले लूप का भी उपयोग कर सकते हैं:

```
अगर (हैव_पोस्ट्स()) {जबकि (हैव_पोस्ट्स())
    {
        पोस्ट();
        var_dump($पोस्ट);
    }
}
```

लूप में कोई आइटम नहीं संभालना

यदि आप ऐसे परिदृश्य को संभालना चाहते हैं तो बस एक `if/else` कथन जोड़ें।

```
यदि (have_posts()) : जबकि (have_posts()) :
    पोस्ट();
    var_dump($पोस्ट);
```

अंत तक; अन्य :

__('इस क्वेरी का कोई परिणाम नहीं है');

अगर अंत;

अध्याय 63: the_title()

परिचय

यह फ़ंक्शन वर्तमान पोस्ट का शीर्षक लौटाता है।

वाक्य - विन्यास

- शीर्षक (\$पहले, \$बाद, \$गूज);

पैरामीटर

पैरामीटर विवरण	
\$पहले	(स्ट्रिंग) (वैकल्पिक) शीर्षक से पहले रखने के लिए पाठ।
\$बाद	(स्ट्रिंग) (वैकल्पिक) शीर्षक के बाद रखा जाने वाला टेक्स्ट।
\$गूज	(बूलियन) (वैकल्पिक) शीर्षक प्रदर्शित करें या इसे PHP में उपयोग के लिए वापस कर दें

टिप्पणियां

- पैरामीटर \$echo के लिए, शीर्षक प्रदर्शित करने के लिए true का उपयोग करें और PHP में उपयोग के लिए इसे वापस करने के लिए false का उपयोग करें
- कृपया ध्यान दें कि the_title का उपयोग केवल लूप्स में किया जा सकता है, यदि आप इसे लूप्स के बाहर उपयोग करना चाहते हैं, तो कृपया get_the_title का उपयोग करें

उदाहरण

शीर्षक_का सरल उपयोग

कोड

```
शीर्षक( );
```

उत्पादन

वर्तमान पोस्ट या पेज का शीर्षक

शीर्षक को पहले और बाद में कोड के साथ प्रिंट करना

कोड

```
the_title( '<h1>', '</h1>' );
```

उत्पादन

h1 टैग में वर्तमान पोस्ट या पेज का शीर्षक

अध्याय 64: विषय-वस्तु

परिचय

वर्डप्रेस थीम आपकी वेबसाइट का फ्रंट-एंड हैं। जब लोग साइट पर जाते हैं तो उन्हें यही दिखाई देता है। चुनने के लिए हज़ारों थीम हैं, सशुल्क और निःशुल्क संस्करण। आप केवल कुछ आवश्यक फ़ाइलों के साथ अपनी स्वयं की कस्टम थीम भी बना सकते हैं।

उदाहरण

वर्डप्रेस थीम्स

थीम कैसे चुनें

प्रत्येक वर्डप्रेस इंस्टाल एक पूर्व-स्थापित थीम के साथ आता है। आप डैशबोर्ड से अपनी थीम प्रबंधित करते हैं। थीम इंस्टॉल करने, पूर्वावलोकन करने, हटाने, सक्रिय करने और अपडेट करने के लिए उपस्थिति > थीम पर जाएं। वर्तमान थीम इस मेनू के ऊपरी बाएँ कोने में पाई जाती है।

थीम छवि पर होवर करने से 'थीम विवरण' बटन का पता चलता है। यह बटन थीम के बारे में जानकारी प्रदान करता है, जैसे संस्करण और विवरण। वर्तमान थीम छवि पर क्लिक करने से आपको शीर्षक जैसी विशिष्ट थीम सेटिंग्स को अनुकूलित करने की सुविधा मिलती है।

उपलब्ध अद्यतन

यदि इंस्टॉल किए गए थीम के लिए अपडेट उपलब्ध हैं, तो आपको एक संदेश मिलेगा जो आपको सूचित करेगा कि एक नया संस्करण उपलब्ध है। आपको अभी नए संस्करण का विवरण या अद्यतन देखने में सक्षम होना चाहिए।

- संस्करण विवरण देखें

संस्करण विवरण लिंक पर क्लिक करने से आप वर्डप्रेस थीम निर्देशिका के एक पृष्ठ पर पहुंच जाएंगे। यहां आपको अपग्रेड संस्करण के विवरण मिलेंगे।

- अभी अद्यतन करें

अपडेट नाउ लिंक पर क्लिक करने से थीम अपग्रेड इंस्टॉल हो जाएगी। थीम को व्यवस्थापन > डैशबोर्ड > अपडेट स्क्रीन से भी अपग्रेड किया जा सकता है।

आपके वर्तमान थीम के अलावा, थीम प्रबंधित करें स्क्रीन अन्य थीम भी दिखाती है जो स्थापित हैं लेकिन वर्तमान में निष्क्रिय हैं। प्रत्येक थीम को एक छोटे स्क्रीनशॉट द्वारा दर्शाया गया है। इन छवियों पर होवर करने से 'थीम विवरण', 'सक्रिय करें' और 'लाइव पूर्वावलोकन' बटन दिखाई देते हैं। आप इस पेज से निष्क्रिय थीम को अपग्रेड या हटा भी सकेंगे। इस स्क्रीन पर प्रत्येक पृष्ठ एक समय में 15 थीम स्क्रीनशॉट तक प्रदर्शित करेगा।

- सक्रिय

इस लिंक पर क्लिक करने से यह वर्तमान थीम बन जाता है।

- सजीव पूर्वावलोकन

इस लिंक पर क्लिक करने पर इस विशिष्ट थीम संस्करण के साथ ब्लॉग का पूर्वावलोकन प्रदर्शित होता है।

- मिटाना

इस लिंक पर क्लिक करने से यह थीम पूरी तरह से हट जाती है, इसमें सभी थीम फ़ाइलें और फ़ोल्डर शामिल हैं। जिस भी चीज़ का बैकअप नहीं लिया गया वह हमेशा के लिए खो जाएगी।

- उपलब्ध अद्यतन

ऊपर अद्यतन उपलब्ध अनुभाग देखें।

थीम्स स्थापित करें

थीम इंस्टॉल करने के कई तरीके नीचे सूचीबद्ध हैं:

- स्वचालित थीम इंस्टालर

इसका उपयोग वर्डप्रेस थीम डायरेक्टरी से थीम इंस्टॉल करने के लिए किया जा सकता है। अपीयरेंस थीम्स स्क्रीन ढूँढने के लिए एडमिनिस्ट्रेशन > अपीयरेंस > थीम्स पर जाएं। नया जोड़ें बटन पर क्लिक करें। यहां से आपको उपयोग करने के लिए थीम मिलेंगी जिनमें कोई बदलाव नहीं है। इस स्क्रीन के शीर्ष पर एक नई थीम खोजने के लिए तीन उपलब्ध तरीकों के साथ खोज सुविधा है; फ़िल्टर, कीवर्ड और विशेषता खोज।

- अपलोड विधि का उपयोग करना

अपलोड विधि एक ज़िप फ़ाइल के माध्यम से एक थीम स्थापित करती है। वर्डप्रेस थीम डायरेक्टरी में सभी थीम्स को इस तरह से इंस्टॉल किया जा सकता है। ज़िप फ़ाइल डाउनलोड करने के बाद, प्रशासन > उपस्थिति > थीम्स पर जाएं, और नया जोड़ें बटन पर क्लिक करें। इसके बाद, थीम अपलोड करें लिंक पर क्लिक करें।

ज़िप फ़ाइल ब्राउज़ करें और अभी इंस्टॉल करें पर क्लिक करें। इसे वर्तमान थीम बनाने के लिए सक्रिय लिंक पर क्लिक करें।

- एफ़टीपी पद्धति का उपयोग करना

एफ़टीपी विधि के साथ थीम स्थापित करने के लिए आपको पहले थीम फ़ाइलों को अपने स्थानीय कंप्यूटर पर डाउनलोड करना होगा। ज़िप फ़ाइल की सामग्री निकालें, फ़ाइल संरचना को संरक्षित करें, और उन्हें एक नए फ़ोल्डर में जोड़ें। यदि थीम लेखक के निर्देश हैं तो उनका पालन करना सुनिश्चित करें।

अपनी साइट के वेब सर्वर तक पहुंचने के लिए एफ़टीपी क्लाइंट का उपयोग करें। अपलोड की गई थीम फ़ाइलों को वर्डप्रेस द्वारा प्रदान की गई अपनी wp-content/themes निर्देशिका में जोड़ें। यदि आवश्यक हो, तो अपने नए थीम को wp-content/themes निर्देशिका के अंदर रखने के लिए एक फ़ोल्डर बनाएं। इसका एक उदाहरण होगा, यदि आपके थीम का नाम टेस्ट है तो इसे wp-content/themes/test में रहना चाहिए।

प्रशासन > उपस्थिति > थीम्स पर जाएं, और सक्रिय लिंक पर क्लिक करें थीम को अपने वर्तमान थीम के रूप में चुनें।

- cPanel के साथ इंस्टाल करना

cPanel नियंत्रण पैनल ज़िप या GZ फ़ाइलों के साथ थीम्स स्थापित करने के लिए एक और विधि प्रदान करते हैं। सीपीनल मैनेजर में, यदि वर्डप्रेस स्थापित है, तो अपने थीम्स फ़ोल्डर पर जाएं। पथ 'public_html/wp-content/themes' के समान दिखेगा। अपलोड फ़ाइल पर क्लिक करें और ज़िप फ़ाइल अपलोड करें।

CPanel में ज़िप फ़ाइल का चयन करें और उस फ़ाइल को अनकंप्रेस करने के लिए दाईं ओर पैनल में फ़ाइल सामग्री निकालें पर क्लिक करें।

प्रशासन > उपस्थिति > थीम्स पर जाएं, और सक्रिय लिंक पर क्लिक करें थीम को अपने वर्तमान थीम के रूप में चुनें।

ऊपर सूचीबद्ध सभी जानकारी वर्डप्रेस कोडेक्स के अनुसार है। संक्षिप्तता के लिए इसे छोटा कर दिया गया है। मूल स्रोत सामग्री यहां पाई जा सकती है। या अधिक जानकारी के लिए Codex.wordpress.org पर जाएं।

एक कस्टम थीम बनाना

ये निर्देश एक बहुत ही बुनियादी, न्यूनतम मानकों के अनुरूप वर्डप्रेस थीम बनाते हैं।

पहला कदम आपके वर्डप्रेस थीम डायरेक्टरी के अंदर एक नया थीम फ़ोल्डर बनाना है। सही पथ होगा: > wp-content > थीम > एक वैध थीम बनाने के लिए, वर्डप्रेस थीम के लिए कम से कम इन दो फ़ाइलों की आवश्यकता होती है:

- Index.php
- style.css

आपकी स्टाइलशीट में एक टिप्पणी होनी चाहिए जो वर्डप्रेस को सचेत करे कि यहां एक थीम मौजूद है।

```
/*  
थीम का नाम: <थीम का नाम>  
लेखक: <लेखक का नाम>  
विवरण: <विवरण यहाँ जाता है>  
संस्करण: <थीम संस्करण #>  
टैग: <टैग टू आईडी थीम>  
*/
```

अब आपकी थीम बन गई है। वर्डप्रेस डैशबोर्ड पर जाएं, और इसे सक्रिय करने के लिए Appearance > Themes पर क्लिक करें।

अध्याय 65: वर्डप्रेस को मैनुअल रूप से अपडेट करें

उदाहरण

एफ़टीपी के माध्यम से

वर्डप्रेस का वांछित संस्करण www.wordpress.org से अपने स्थानीय कंप्यूटर पर डाउनलोड करें 1. और फ़ाइल को अनज़िप करें।

- अपने वर्तमान संस्करण का बैकअप भी रखें... शायद।
2. अपने पसंदीदा FTP क्लाइंट के साथ अपनी वेबसाइट से कनेक्ट करें (FileZilla लोकप्रिय और आसान है, लेकिन कोई भी FTP क्लाइंट ठीक रहेगा)
 - इसके लिए निर्देश वर्डप्रेस के दायरे से बाहर हैं, लेकिन भविष्य में प्रस्तावित एफ़टीपी विषय में पाए जा सकते हैं।
 3. "wp-admin" और "wp-includes" शीर्षक वाले फ़ोल्डर (और उनकी सामग्री) को अपने सर्वर पर उनकी मेल खाती निर्देशिकाओं में अपलोड करें। वर्तमान फ़ोल्डरों को अधिलेखित करना सुनिश्चित करें。
 - आप "wp-content" फ़ोल्डर को अपलोड करना छोड़ सकते हैं जब तक कि आप इसमें शामिल थीमों में से किसी एक का उपयोग करना नहीं चुनते। यदि आप अपने चुने हुए संस्करण में शामिल डिफ़ॉल्ट थीम को अपडेट/अपलोड करना चाहते हैं तो आपको यह फ़ोल्डर भी अपलोड करना चाहिए।
 4. होम फ़ोल्डर में अलग-अलग फ़ाइलें अपलोड करें (index.php, wp-*.php, आदि)।
 - आप "license.txt" और "readme.html" शीर्षक वाली फ़ाइलों को छोड़ सकते हैं क्योंकि उन्हें कार्य करने की आवश्यकता नहीं है और उन्हें सुरक्षा शोषण के लिए आपके WP संस्करण को निर्धारित करने के तरीकों के रूप में उपयोग किया जा सकता है।
 5. किसी भी आवश्यक डेटाबेस अपडेट को करने के लिए अपनी वेबसाइट पर जाएँ और लॉग इन करें।
 - सभी WP अपडेट में DB परिवर्तन नहीं होते हैं, लेकिन कुछ में होते हैं।

ध्यान दें: यह विधि अनाथ फ़ाइलें बनाएगी जो समय के साथ जमा हो सकती हैं/होंगी और सुरक्षा जोखिम पेश कर सकती हैं। पूरा होने के बाद फ़ाइल की तुलना करना सुनिश्चित करें और पिछले WP संस्करणों से अपने सर्वर से पुरानी फ़ाइलों को हटा दें जो अब उपयोग में नहीं हैं।

अध्याय 66: वर्डप्रेस प्लगइन निर्माण

परिचय

वर्डप्रेस प्लगइन्स को आपके वेबसाइट एप्लिकेशन के सर्वर लॉजिक और/या एडमिन भागों पर ध्यान केंद्रित करना चाहिए। अच्छे प्लगइन्स अच्छे ऐप्स की तरह होते हैं, वे एक काम बहुत अच्छे से करते हैं। उनका उद्देश्य सीएमएस के कुछ हिस्सों को मॉड्यूलर तरीके से बढ़ाना और स्वचालित करना है, क्योंकि आप उन्हें सक्रिय और निष्क्रिय कर सकते हैं। अच्छे प्लगइन्स वर्डप्रेस कोर एक्शन, फिल्टर और मौजूदा जावास्क्रिप्ट और सीएसएस फ्रेमवर्क का उपयोग करते हैं।

उदाहरण

प्लगइन फ़ोल्डर और फ़ाइलों का न्यूनतम सेटअप

प्लगइन बनाने का पहला चरण फ़ोल्डर और फ़ाइल बनाना है जिससे प्लगइन लोड होगा।

प्लगइन्स `/wp-content/plugins/` में स्थित हैं।

वर्डप्रेस मानक एक फ़ोल्डर और फ़ाइल नाम बनाना है जो एक दूसरे को इस प्रकार प्रतिबिंबित करता है:

```
/wp-content/plugins/myplugin/ /wp-content/  
plugins/myplugin/myplugin.php
```

अपनी प्लगइन फ़ाइल बनाने के बाद आपको प्लगइन हेडर के साथ अपना प्लगइन शुरू करना होगा। यह वर्डप्रेस को आपकी प्लगइन फ़ाइल को स्कैन करने और प्लगइन के बारे में मेटा डेटा संग्रहीत करने की अनुमति देता है, और उपयोगकर्ताओं को इसका उपयोग करने और यह निर्धारित करने की अनुमति देता है कि वे आपके प्लगइन को सक्रिय या निष्क्रिय चाहते हैं। इस टेम्पलेट को आपके द्वारा बनाई गई मुख्य प्लगइन फ़ाइल के शीर्ष पर कॉपी करें, और आवश्यकतानुसार इसे संशोधित करें:

```
<?php  
/**  
 * प्लगइन नाम: प्लगइन-नाम  
 * प्लगइन यूआरआई: HTTP-लिंक-टू-वेबसाइट-प्लगइन-पेज-या-रेपो  
 * विवरण: संक्षिप्त विवरण - इसे छोटा रखें  
 * लेखक: वर्डप्रेस-डॉट-ओआरजी-उपयोगकर्ता नाम  
 * संस्करण: 0.0.1  
 * लेखक यूआरआई: HTTP-लिंक-टू-मेनेजर  
 * लाइसेंस: जीपीएनयू जनरल पब्लिक लाइसेंस v2 या बाद का  
 * लाइसेंस यूआरआई: http://www.gnu.org/licenses/gpl-2.0.html * टेक्स्ट डोमेन: शॉर्ट_प्रीफिक्स  
  
 */  
  
// कस्टम PHP वर्डप्रेस प्लगइन काम शुरू करें
```

ध्यान दें कि वर्डप्रेस प्लगइन्स को आमतौर पर जीपीएल के रूप में लाइसेंस प्राप्त होना चाहिए। हालाँकि इस विषय के भाग के रूप में लाइसेंसिंग पर चर्चा नहीं की जानी चाहिए।

इस बिंदु पर, आप पहले से ही वर्डप्रेस एडमिन क्षेत्र में अपना नया प्लगइन देख पाएंगे। एक मानक सेटअप में आप इस क्षेत्र को `/wp-admin/plugins.php` पर पाएंगे। आगे बढ़ें और अपना सक्रिय करें

प्लगइन, और आप अपना प्लगइन बनाने के अगले चरण में जाने के लिए तैयार हैं!

हमारे उदाहरण को कार्रवाई योग्य किसी चीज़ पर समाप्त करने के लिए, अब आप अपनी प्लगइन फ़ाइल के नीचे निम्नलिखित जोड़ सकते हैं:

```
डाई ('मेरा कस्टम प्लगइन लोड हो गया है:'। __फ़ाइल__);
```

इस परिवर्तन के बाद अपनी साइट को रीफ़्रेश करने पर सभी पेजों पर यह टेक्स्ट प्रिंट हो जाएगा। प्रोडक्शन (लाइव) साइटों पर ऐसा कभी न करें, और जारी रखने से पहले इसे वापस लेना हमेशा याद रखें।

अध्याय 67: वर्डप्रेस थीम और चाइल्ड थीम विकास

परिचय

वर्डप्रेस सरल सूचना वेबसाइट बनाने के साथ-साथ अधिक परिष्कृत वेबसाइट और यहां तक कि छोटी वेबशॉप बनाने के लिए भी व्यापक रूप से उपयोग किया जाने वाला सीएमएस है।

वर्डप्रेस थीम का उपयोग करता है। इन थीम का उपयोग वर्डप्रेस वेबसाइट के लेआउट और सामग्री कार्यक्षमता को बनाने के लिए किया जाता है। थीम पूरे इंटरनेट पर पाई जा सकती हैं।

प्रत्येक थीम की अपनी अनूठी कार्यक्षमता और लेआउट होता है लेकिन कभी-कभी किसी वेबसाइट के लिए सही थीम ढूंढना कठिन होता है। सौभाग्य से हम अपनी स्वयं की थीम बनाने में भी सक्षम हैं।

उदाहरण

अपनी खुद की थीम विकसित करना

एक वर्डप्रेस थीम में दो प्रकार की फ़ाइलें होती हैं। प्रत्येक थीम की मूल फ़ाइलें और वे फ़ाइलें जो थीम के लेआउट और कार्यक्षमता को परिभाषित करती हैं। इस दूसरे समूह को मैं थीम विशिष्ट फ़ाइलों को कॉल करने जा रहा हूँ।

मूल थीम फ़ाइलें मूल थीम फ़ाइलें वे

फ़ाइलें हैं जिनका उपयोग किसी थीम को सेटअप और पंजीकृत करने के लिए किया जाता है। नीचे दी गई सूची में मैं शीघ्र ही प्रत्येक फ़ाइल और उसके उपयोग का वर्णन करूंगा। बाद में मैं सबसे बुनियादी उदाहरण फ़ाइलें जोड़ूंगा जो आपकी खुद की वर्डप्रेस थीम सेट करने के लिए आवश्यक हैं।

- फ़ंक्शंस.php: फ़ंक्शंस.php फ़ाइल का उपयोग थीम के सभी फ़ंक्शंस, साइडबार, स्क्रिप्ट और शामिल को पंजीकृत करने के लिए किया जाता है। उदाहरण के लिए, इस फ़ाइल में आप CSS फ़ाइलें, JS फ़ाइलें आदि शामिल कर सकते हैं।
- शीर्ष लेख और पाद लेख: शीर्ष लेख और पाद लेख फ़ाइलें (header.php और footer.php) वे फ़ाइलें हैं जिनका उपयोग शीर्ष लेख और पाद लेख पर कॉल करने के लिए किया जाता है। उदाहरण के लिए हेडर और फूटर फ़ाइल में वर्डप्रेस बैक-एंड सिस्टम का लिंक होता है। Index.php: Index.php फ़ाइल वह फ़ाइल है जो डिफ़ॉल्ट-पेज टेम्पलेट बनाती है। इस
- फ़ाइल में आप इस डिफ़ॉल्ट-टेम्पलेट लेआउट के टुकड़े देख सकते हैं, संपादित कर सकते हैं और हटा सकते हैं। सिंगल.php: सिंगल.php फ़ाइल वह फ़ाइल है जो सिंगल पोस्ट टेम्पलेट पेज बनाती है। पेजों के लिए डिफ़ॉल्ट-टेम्पलेट की तरह, लेकिन अब एकल
- पोस्ट पेजों के लिए। प्रारूप.php प्रारूप.php फ़ाइल वह फ़ाइल है जो किसी पृष्ठ से सामग्री-पाठ टेम्पलेट बनाती है। तो यदि आपके पास एक होम पेज होगा और आप एक टेक्स्ट जोड़कर इसे बैक-एंड से संपादित करेंगे।

यह फ़ाइल इस पाठ का मानक मार्कअप बनाती है। 404.php 404.php फ़ाइल

- 404 टेम्पलेट बनाती है। इस फ़ाइल में इस पृष्ठ का मूल लेआउट शामिल है।
- Archive.php Archive.php फ़ाइल संग्रह पृष्ठ का लेआउट बनाती है। style.css मूल स्टाइलशीट फ़ाइल।

तो इस सूची में आप अपनी स्वयं की वर्डप्रेस थीम के सेट अप के लिए सभी आवश्यक फ़ाइलें देख सकते हैं।

आइए अब कुछ फ़ाइलों पर एक नज़र डालें जिन्हें आप चाहें तो बना सकते हैं लेकिन वर्डप्रेस थीम के लिए आवश्यक फ़ाइलें नहीं हैं। ये फ़ाइलें अधिकतर टेम्पलेट फ़ाइलें और अन्य कार्यात्मक एक्सटेंशन हैं।

कस्टम पेज टेम्पलेट पेज-<आपका

अपना नाम>.php: एक वर्डप्रेस थीम में आप कई पेज टेम्पलेट बनाने में सक्षम हैं। नई पेज टेम्पलेट फ़ाइलें बनाकर। एक मानक पृष्ठ टेम्पलेट फ़ाइल में निम्नलिखित नाम विशेषताएँ होती हैं। टेम्पलेट का पेज नाम और .php यदि उदाहरण के लिए आप अपने ब्लॉग पेज के लिए एक नया पेज टेम्पलेट बनाना चाहते हैं तो आप इसे पेज-ब्लॉग.php कह सकते हैं। वर्डप्रेस स्वचालित रूप से फ़ाइल को पढ़ता है और फ़ाइल को टेम्पलेट चुनें मेनू में जोड़ता है। सुनिश्चित करें कि आपने कम से कम `get_header()` और `get_footer()` फ़ंक्शन शामिल किए हैं। यह भी सुनिश्चित करें कि आप निम्नलिखित उदाहरण जोड़कर फ़ाइल के शीर्ष पर एक टिप्पणी में अपने टेम्पलेट का नाम दें।

```
<?php /
*
* टेम्पलेट का नाम: मुखपृष्ठ टेम्पलेट */ get_header();

?>
```

कस्टम सिंगल पोस्ट पेज टेम्पलेट सिंगल-<आपका अपना

नाम>.php: वर्डप्रेस थीम में ऊपर वर्णित पेज टेम्पलेट की तरह आप भी अपना खुद का सिंगल पोस्ट पेज टेम्पलेट बनाने में सक्षम हैं। पेज टेम्पलेट की तरह ही फ़ाइल में तीन भाग होते हैं, यह घोषित करने के लिए कि यह एक एकल पोस्ट पेज है <टेम्पलेट का आपका नाम> और फ़ाइल एक्सटेंशन .php। पेज टेम्पलेट की तरह ही न्यूनतम आवश्यकताएं यह सुनिश्चित करने के लिए कि वर्डप्रेस नए टेम्पलेट को पढ़ता है, फ़ंक्शन `get_header()` और `get_footer()` जोड़ रहा है। और निश्चित रूप से नीचे दिए गए उदाहरण की तरह अपना टेम्पलेट नाम भी जोड़ रहे हैं

```
<?php

/*
* टेम्पलेट का नाम: पोस्ट पोर्टफोलियो
* टेम्पलेट पोस्ट प्रकार: पोस्ट, पेज */

?>
```

हम टेम्पलेट पोस्ट प्रकार भी इंगित करते हैं: जिसका अर्थ है कि यह किस प्रकार का टेम्पलेट है, इस मामले में पोस्ट और पेज।

कस्टम पोस्ट टेक्स्ट टेम्पलेट प्रारूप -<आपका

अपना नाम>.php: वर्डप्रेस थीम में आप पोस्ट आउटपुट टेम्पलेट भी बनाने में सक्षम हैं। ये प्रारूप टेम्पलेट किसी पोस्ट का लेआउट और सामग्री हैं। उदाहरण के लिए, यदि कुछ मामलों में आप चाहते हैं कि पोस्ट केवल पोस्ट की सामग्री या शीर्षक दिखाए, तो आप इस प्रकार के समायोजन बनाने के लिए इन टेम्पलेट्स का उपयोग कर सकते हैं। चूंकि इस प्रकार के टेम्पलेट केवल पोस्ट बैक-एंड सामग्री को फॉर्मेट कर रहे हैं जो किसी उपयोगकर्ता द्वारा बनाई गई थी, इसलिए हमें `get_header()` और `get_footer()` को शामिल करने की आवश्यकता नहीं है क्योंकि ये पहले से ही पेज टेम्पलेट में परिभाषित हैं। निम्नलिखित बुनियादी उदाहरण का उपयोग करके सुनिश्चित करें कि आपका टेम्पलेट किसी पोस्ट को पहचानने में सक्षम है।

```
<div>
```

<article id='post-<?php the_ID(); ?>' <?php post_class(); ?>> </लेख> </div>

तो अब जब हम बुनियादी फ़ाइलों और कई टेम्पलेट विशिष्ट फ़ाइलों में से कुछ के बारे में कुछ जानते हैं, तो साइडबार और विजेट्स के बारे में बात करना शुरू करने का समय आ गया है। भविष्य में इसे एक बहुत ही वर्डप्रेस थीम बनाने पर चरण दर चरण ट्यूटोरियल की शुरुआत के साथ जोड़ा जाएगा।

अध्याय 68: wp_get_current_user()

वाक्य - विन्यास

- wp_get_current_user()

उदाहरण

वर्तमान उपयोगकर्ता प्राप्त करना

पूर्वनिर्धारित फ़ंक्शन का उपयोग करके वर्डप्रेस में वर्तमान उपयोगकर्ता की सभी जानकारी प्राप्त करना

wp_get_current_user();

```
<?php
```

```
$current_user = wp_get_current_user();
```

```
इको "उपयोगकर्ता नाम:"।$current_user->user_login; इको "उपयोगकर्ता नाम:"।
$current_user->ID; इको "उपयोगकर्ता नाम:"।$current_user-
>user_pass; इको "उपयोगकर्ता नाम:"।$current_user->user_nicename; इको
"उपयोगकर्ता नाम:"।$current_user->user_email; इको "उपयोगकर्ता नाम:"।$current_user-
>user_url; इको "उपयोगकर्ता नाम:"।$current_user->user_registered; इको
"उपयोगकर्ता नाम:"।$current_user->user_activation_key; इको "उपयोगकर्ता
नाम:"।$current_user->user_status; इको "उपयोगकर्ता नाम:"।$current_user->display_name;
```

```
?>
```

wp_get_current_user() से उपयोगकर्ता जानकारी प्राप्त करने के लिए फ़ोरैच लूप का उपयोग करें

```
$उपयोगकर्ता = wp_get_current_user();
```

```
foreach($user->data as $key=>$user_data){
```

```
    यदि($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));
        }
```

```
        इको $ नाइस_की। ' : ' . $user_data . '<br />';
```

```
    }
}
```

अध्याय 69: wp_get_current_user()

उदाहरण

वर्तमान लॉगइन उपयोगकर्ता जानकारी प्राप्त करें

वर्तमान में लॉग इन उपयोगकर्ता से संबंधित जानकारी पुनर्प्राप्त करता है, और इसे वैश्विक में रखता है परिवर्तनीय `$current_user`

यह फ़ंक्शन किसी भी पैरामीटर को स्वीकार नहीं करता है.

उपयोग:

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

उदाहरण:

```
<?php
$current_user = wp_get_current_user();

इको 'उपयोगकर्ता नाम:'           . $current_user->user_login . "\n";
इको 'उपयोगकर्ता ईमेल:'           . $current_user->user_email . "\n";
प्रतिध्वनि 'उपयोगकर्ता स्तर:'     . $current_user->user_level . "\n";
इको 'उपयोगकर्ता का पहला नाम:'   . $current_user->user_firstname . "\n";
इको 'उपयोगकर्ता का अंतिम नाम:'  . $current_user->user_lastname . "\n";
इको 'उपयोगकर्ता प्रदर्शन नाम:'  $current_user->display_name . "\n";
इको 'उपयोगकर्ता आईडी:'        $current_user->आईडी . "\n";

?>
```

यह निर्धारित करने के लिए कि कोई विज़िटर लॉग इन है या नहीं, आप पहले `is_user_logged_in()` का उपयोग कर सकते हैं, फिर प्राप्त करें यदि विज़िटर लॉग इन है तो वर्तमान उपयोगकर्ता जानकारी:

```
<?php
यदि (is_user_logged_in()) $current_user = {
    wp_get_current_user();

    प्रतिध्वनि 'स्वागत है,' } अन्यथा { . $current_user->user_login . '!';

    प्रतिध्वनि 'आपका स्वागत है, आगतुक!';
}
?>
```

अध्याय 70: WP_Query() लूप

परिचय

पोस्ट, पेज और कस्टम पोस्ट प्रकारों के लिए क्वेरी करने के लिए WP_Query। आपको विशिष्ट पोस्ट और पेज या कस्टम पोस्ट प्रकारों की सूची मिलेगी। WP_Query आपको अपने मानदंडों के अनुसार डेटाबेस से पोस्ट खींचने की अनुमति देता है।

उदाहरण

नवीनतम 10 पोस्ट पुनः प्राप्त करें

```
$args =  
    array('post_type'=>'post',  
        'posts_per_page' =>'10'  
    );  
$latest_posts_query = नया WP_Query( $args ); if($latest_posts_query-  
>have_posts()) : जबकि ( $latest_posts_query-> has_posts() ) :  
$latest_posts_query->the_post();  
//पोस्ट विवरण यहीं से प्राप्त करें; अगर अंत;
```

अध्याय 71: WP-CLI

परिचय

WP-CLI वर्डप्रेस इंस्टॉलेशन को प्रबंधित करने के लिए कमांड-लाइन टूल का एक सेट है। आप वेब ब्राउज़र का उपयोग किए बिना प्लगइन अपडेट कर सकते हैं, मल्टीसाइट इंस्टॉल कॉन्फ़िगर कर सकते हैं और बहुत कुछ कर सकते हैं।

उदाहरण

थीम प्रबंधित करें

विषयों की एक सूची प्राप्त करें.

```
$ wp थीम सूची
```

Wordpress.org से नवीनतम संस्करण स्थापित करें और सक्रिय करें

```
$ wp थीम इंस्टॉल करें द्वाँटीसिक्सटीन -- सक्रिय करें
```

स्थानीय ज़िप फ़ाइल से इंस्टॉल करें

```
$ wp थीम इंस्टॉल करें ../my-theme.zip
```

रिमोट ज़िप फ़ाइल से इंस्टॉल करें

```
$ wp थीम इंस्टॉल करें http://s3.amazonaws.com/bucketname/my थीम.zip?  
AWSAccessKeyId=123&Expires=456&Signature=abcdef
```

स्थापित थीम का विवरण प्राप्त करें

```
$ wp थीम को छब्बीस मिलते हैं --फ़िल्ड=नाम, शीर्षक, संस्करण
```

थीम की स्थिति प्राप्त करें

```
$ wp थीम स्टेटस द्वाँटीसिक्सटीन
```

प्लगइन्स प्रबंधित करें

प्लगइन्स की एक सूची प्राप्त करें

```
$ wp प्लगइन सूची
```

साइट पर सक्रिय प्लगइन्स की सूची बनाएं।

```
$ wp प्लगइन सूची --status=सक्रिय --format=json
```

नेटवर्क में प्रत्येक साइट पर प्लगइन्स की सूची बनाएं।

```
$ wp साइट सूची --फ़िल्ड=यूआरएल | xargs -I % wp प्लगइन सूची --url=%
```

प्लगइन सक्रिय करें

```
$ wp प्लगइन हैलो-डॉली को सक्रिय करता है
```

प्लगइन निष्क्रिय करें

```
$ wp प्लगइन हैलो-डॉली को निष्क्रिय करें
```

प्लगइन हटाएँ

```
$ wp प्लगइन हैलो-डॉली हटाएं
```

Wordpress.org से नवीनतम संस्करण स्थापित करें और सक्रिय करें

```
$ wp प्लगइन इंस्टॉल करें bbpress --activate
```

WP-CLI को स्वयं प्रबंधित करें

वर्तमान में स्थापित संस्करण प्रदर्शित करें।

```
$ wp सीएलआई संस्करण
```

WP-CLI के अपडेट की जाँच करें।

```
$ wp cli चेक-अपडेट
```

WP-CLI को नवीनतम स्थिर रिलीज़ में अपडेट करें।

```
$ wp सीएलआई अद्यतन
```

सभी उपलब्ध उपनामों की सूची बनाएं.

```
$ wp सीएलआई उपनाम
```

WP-CLI पर्यावरण के बारे में विभिन्न विवरण प्रिंट करें।

```
$ wp सीएलआई जानकारी
```

स्थापित आदेशों की सूची को JSON के रूप में डंप करें।

```
$ wp सीएलआई सीएमडी-डंप
```

वर्डप्रेस इंस्टाल को डाउनलोड, इंस्टॉल, अपडेट और प्रबंधित करें।

वर्डप्रेस कोर डाउनलोड करें

```
$ wp कोर डाउनलोड --locale=nl_NL
```

वर्डप्रेस इंस्टॉल करें

```
$ wp कोर इंस्टाल --url=example.com --शीर्षक=उदाहरण --admin_user=पर्यवेक्षक -- admin_password=strongpassword --  
admin_email=info@example.com
```

वर्डप्रेस संस्करण प्रदर्शित करें

```
$ wp कोर संस्करण
```

एकल-साइट इंस्टाल को वर्डप्रेस मल्टीसाइट इंस्टाल में बदलें।

```
$ wp कोर मल्टीसाइट-कन्वर्ट
```

स्क्रीच से वर्डप्रेस मल्टीसाइट इंस्टॉल करें।

```
$ wp कोर मल्टीसाइट-इंस्टॉल
```

उपयोगकर्ताओं को प्रबंधित करें

उपयोगकर्ता आईडी सूचीबद्ध करें

```
$ wp उपयोक्ता सूची--फ़िल्ड=आईडी
```

एक नया उपयोगकर्ता बनाएं.

```
$ wp उपयोगकर्ता बॉब बनाएं bob@example.com --role=author
```

किसी मौजूदा उपयोगकर्ता को अपडेट करें.

```
$ wp उपयोगकर्ता अद्यतन 123 --display_name=मैरी --user_pass=marypass
```

उपयोगकर्ता 123 को हटाएं और पोस्ट को उपयोगकर्ता 567 पर पुनः असाइन करें

```
$ wp उपयोक्ता हटाएँ 123--पुनर्असाइन=567
```

WP-config.php में संग्रहीत क्रेडेंशियल्स का उपयोग करके बुनियादी डेटाबेस संचालन करें

एक नया डेटाबेस बनाएं.

```
$ wp db बनाएं
```

किसी मौजूदा डेटाबेस को छोड़ें.

```
$ wp db ड्रॉप --हाँ
```

वर्तमान डेटाबेस को रीसेट करें.

```
$ wp db रीसेट--हाँ
```

किसी फ़ाइल में संग्रहीत SQL क्वेरी निष्पादित करें।

```
$ wp db क्वेरी <debug.sql
```

अध्याय 72: WP-क्रोन

उदाहरण

wp_schedule_event() उदाहरण

```
// रजिस्टर एक्टिवेशन हुक रजिस्टर_एक्टिवेशन_हुक ( __FILE__,
'example_activation' );

// सक्रियण हुक फ़ंक्शन के लिए फ़ंक्शन example_activation() {

    // जांचें कि क्या शेड्यूल किया गया हुक मौजूद है यदि ( !
    wp_next_scheduled( 'my_event' ) ) { // एक हुक शेड्यूल करता है // समय() - किसी इवेंट के
    चलने का पहला समय (UNIX टाइमस्टैम्प
    प्रारूप) // 'प्रति घंटा' - पुनरावृत्ति ( 'प्रति घंटा', 'दैनिक दो बार', 'दैनिक' ) // 'my_event' - निष्पादित करने के लिए एक एक्शन हुक का नाम। wp_schedule_event(समय(), 'प्रति
    घंटा', 'my_event');

    }

}

add_action('my_event', 'do_this_hourly');

// आपके प्रति घंटा ईवेंट फ़ंक्शन का कोड do_this_hourly() { // अपना कोड
यहां डालें

}

// रजिस्टर निष्क्रियकरण हुक रजिस्टर_डीएक्टिवेशन_हुक( __FILE__,
'example_deactivation');

// हुक फ़ंक्शन को निष्क्रिय करने के लिए फ़ंक्शन example_deactivation() {

    // शेड्यूल किए गए हुक को साफ़ करें
    wp_clear_scheduled_hook('my_event');

}

}
```

महत्वपूर्ण: वर्डप्रेस क्रॉन तभी चलता है जब आपकी वेबसाइट का कोई पेज हिट होता है। इसलिए, कम ट्रैफिक वाली वेबसाइट के लिए आपको पेजों को हिट करने के लिए अपनी होस्टिंग पर क्रॉन सेटअप करना होगा।

wp_schedule_event() में कस्टम पुनरावृत्ति अंतराल

```
// यह फ़ंक्शन $schedules फ़ंक्शन में कस्टम अंतराल (5 मिनट) जोड़ता है five_mिनट_interval( $schedules ) $schedules['पांच_मिनट'] = array(
{

    'अंतराल' => 60 * 5, 'प्रदर्शन' => '5 मिनट';

});

// शेड्यूल वापस करें;
}

// एक कस्टम अंतराल फ़िल्टर जोड़ें add_filter( 'cron_schedules', '
five_mिनट_interval' );
```

```
// एक हुक शेड्यूल करता है  
wp_schedule_event(time(), 'पंच_मिनट', 'my_event');
```

क्रेडिट

एस। नहीं	अध्याय	योगदानकर्ताओं
1	शुरू करना WordPress के	4444, ए रजा, एंड्रयू, दुश्मनी, अनुपम, क्रिस फ्लेचर, सिप्रियन, समुदाय, फ्लोरिडा, जेम्स जोन्स, जोनास सीज़, लियो एफ, मार्क सेंट रेमंड, मयंक गुप्ता, मिलाप, नस, पांडा, रैप-2-एच, सेठ सी., शुभम, ट्रेवर क्लार्क, वज्रसार
2	क्रियाएँ और फ़िल्टर	डेविड, इहोर वीरोटनोव, मृणाल हक, खुद बनने की कोशिश कर रहा हूँ
3	शोर्टकोड जोड़ें	पूर्वविक7373, रेमनशेफ
4	संपर्क जोड़ें/निकालें उपयोगकर्ताओं के लिए जानकारी उपयोगकर्ता_संपर्क विधियाँ फ़िल्टर हुक	पेटार पोपोविक, रेमनशेफ
5	add_action()	हाबिल मेलक्विट्स कैलेजो, वकास बुखारी
6	add_editor_style()	गेब्रियल ची होंग ली
7	add_menu_page()	ब्रासोफिलो, गेब्रियल ची होंग ली
8	add_submenu_page()	गेब्रियल ची होंग ली, सैद्धांतिक
9	add_theme_support()	गेब्रियल ची होंग ली
10	व्यवस्थापक डैशबोर्ड विजेट	सैद्धांतिक
11	अजाक्स	एंडी, दिग्विजयाद, गौरव श्रीवास्तव, ग्रेटब्लेक, निसर्ग पटेल, रुस्तान मुरारोव, swab
12	मुख्य लूप को बदलना (pre_get_posts फ़िल्टर)	डेविड अर्बनस्की, पेटार पोपोविक
13	बाल थीम मूल बातें	आंद्रेई, रज़वान ओनोफ़ेई, व्लाद ओलारू
14	एक पोस्ट बनाएं प्रोग्राम के रूप में	रेमनशेफ, रोएल मैग्दालेनो, रिकेश
15	के लिए टेम्पलेट बनाएं कस्टम पोस्ट प्रकार	अशोक जी, एग्नारो, जो डूले, मनोरोन्हा
16	एक रिवाज बनाना खाका	पेटार पोपोविक

17	कस्टम विशेषज्ञ साथ में अंश_लंबाई और अंश_अधिक	इंकिस्टा, पेटार पोपोविक, रेमनशेफ
18	कस्टम पोस्ट प्रकार	कैओ फेलिप परेरा, डैन डिवाइन, जेडी, जानव, जेगुप, कुशल शाह, उमर खय्याम, रानुका, सैद्धांतिक
19	कस्टमाइज़र मूल बातें (वैनल, अनुभाग जोड़ें, सेटिंग, नियंत्रण)	4444, अहमद अवैस, रेमनशेफ
20	कस्टमाइज़र नमस्ते दुनिया	डेन ग्रीन-लीप्सिगर
21	डिबगिंग	बारबोक, डिंगो_डी, jgroup
22	लिपियों को कतारबद्ध करना	डिंगो_डी, हर्षल लिमये, जेडी, ambacon40, मॉन्ट्रियलिस्ट, पेल्लमरेड, पेटार पोपोविक
23	कतारबद्ध शैलियाँ	डिंगो_डी, हर्षल लिमये, लक्ष्मण, मनोहरोन्हा, मॉन्ट्रियलिस्ट, पेटार पोपोविक, रेमनशेफ, रुस्लान मुरारोव, आभासीअंतिम
24	समारोह : wp_trim_words()	हर्षल लिमये
25	फ़ंक्शन: add_action() बॉस्को, रेमनशेफ	
26	get_bloginfo()	हाबिल मेलक्विएड्स कैलेजो, हर्षल लिमये, हेकैमरून, केनबी , नेट बीयर्स, रेमनशेफ, टॉम जे नोबेल, आभासीअंतिम, वेस मोबर्ली
27	get_home_path()	इहोर वोरोटनोव
28	get_option()	गेब्रियल ची होंग ली
29	get_permalink()	गेब्रियल ची होंग ली
30	get_template_part()	डेन डिवाइन, कुशल शाह
31	get_the_category()	गेब्रियल ची होंग ली
32	get_the_title()	गेब्रियल ची होंग ली
33	होम_यूआरएल()	डिंगो_डी, कुशल शाह, मैथ्यू, श्रीमान डेवलपर
34	मैं कैसे एकीकृत कर सकता हूँ मार्कडाउन संपादक के साथ अग्रिम कस्टम फ़ील्ड का पुनरावर्तक जोड़ें पर।	फैटबिट

35	इनिट	हाबिल मेलक्विएड्स कैलेजो, बारबोक
36	स्थापना और विन्यास	केन्योन, मार्को रोमानो, पिंग.चेन, एसएल बार्थ, स्टिग-जेएस, सैद्धांतिक, युआन लुंग लुओ
37	नेटवर्क बनाना HTTP के साथ अनुरोध एपीआई	जॉर्डन, मज़ांगदा, सबसे दुर्लभ
38	मेटा बॉक्स	ऑस्टिन विंस्टनले
39	विकल्प एपीआई	हर्षल लिमये, थपथपाना जे, रेमेनशेफ
40	प्लगइन विकास	कोण.ए.ए, पिंग.चेन
41	पोस्ट प्रारूप	शशांक अग्रवाल
42	पूछताछ पोस्ट	डिंगो_डी
43	ऑटो लाइन हटाएँ सामग्री से विराम और अंश	ऑस्टिन विंस्टनले
44	से संस्करण हटाएँ वर्डप्रेस और स्टाइलशीट	जय.जीवनी, मनोहरोन्हा, सैद्धांतिक
45	रेस्ट एपीआई	पिकार्ड
46	वर्डप्रेस लोकल चलाएँ एक्सएएमपीपी के साथ	पियरे.विरेन्स, सैद्धांतिक
47	अपना सुरक्षित करें इंस्टालेशन	जेम्स जोन्स
48	वर्डप्रेस में सुरक्षा - भागना	लक्ष्मण, the4kman
49	वर्डप्रेस में सुरक्षा - स्वच्छता	लक्ष्मण
50	शॉर्टकोड	ऐड विक्स, एडम जेनशाफ्ट, ब्रासोफिलो, जॉन स्लेगर्स, काइलर, शशांक अग्रवाल
51	के साथ शॉर्टकोड गुण	दिविजयाद, फ़ायरफ़ॉग, रेमेनशेफ
52	शॉर्टकोड	पेलमरेड
53	साइडबार	डिंगो_डी, कुशल शाह, सैद्धांतिक

54	साइट माइग्रेशन	ऑस्टिन विंस्टनले
55	वर्गीकरण	एडिफ़ैटज़, कुशल शाह, पूर्वविक7373
56	टेम्पलेट पदानुक्रम	जेग्रुप, मरज़ब, पेल्मरेड, सैद्धांतिक
57	टेम्पलेट_शामिल	हाबिल मेलक्विएड्स कैलेजी, डेविड, रेमनशेफ
58	\$wpdb ऑब्जेक्ट	कुशल शाह, एमकॉन, sweb
59	एडमिन बार (उर्फ "द टूलबार")	डिंगो_डी, हर्षल लिमये, जेसीएल1178, कुशल शाह
60	लूप (मुख्य वर्डप्रेस लूप)	anik4e, डेविड अर्बनस्की
61	the_title()	गेब्रियल ची होंग ली
62	थीम्स	जेईएफ़
63	वर्डप्रेस अपडेट करें मैनुअल	नाइटहॉक
64	वर्डप्रेस प्लगइन निर्माण	सेठ सी.
65	वर्डप्रेस थीम और बच्चों के विषय विकास	मौल का तूफ़ान
66	wp_get_current_user()	एबेल मेलक्विएड्स कैलेजी, बेनोटी, मुहम्मद फारुख फ़ैज़ी
67	WP_Query() लूप	ब्रजेश
68	WP-सीएलआई	jgroup
69	WP-क्रॉन	sweb