

Spis treści

O autorach	15
Wstęp	17
Wprowadzenie	19
Rozdział 1. Wprowadzenie do wtyczek	23
Co to jest wtyczka?	23
W jaki sposób wtyczki współdziałają z platformą WordPress?	24
Kiedy wtyczki są wczytywane?	25
Dostępne wtyczki	25
Oficjalny katalog wtyczek	26
Przykłady popularnych wtyczek	26
Popularne tagi wtyczek	27
Zalety wtyczek	27
Brak konieczności modyfikacji jądra platformy	27
Nie trzeba wywierać otwartych drzwi	28
Oddzielenie wtyczek i motywów	28
Łatwe uaktualnienia	29
Łatwiejsze dzielenie się wtyczkami i ich ponowne używanie	29
Wtyczki są oddzielone od siebie	29
Społeczność tworząca wtyczki	30
Instalacja wtyczek i zarządzanie nimi	30
Instalacja wtyczki	30
Zarządzanie wtyczkami	31
Edycja wtyczek	31
Katalog wtyczek	32
Typy wtyczek	32
Testowanie funkcji wtyczek	33
Podsumowanie	34
Rozdział 2. Podstawy wtyczek	35
Utworzenie pliku wtyczki	35
Nadanie nazwy wtyczce	35
Używanie katalogu	36
Stosowanie rozsądnych praktyk	36
Stosowanie prefiksu w każdej sytuacji	36
Organizacja pliku	37
Struktura katalogów	37

Wymagania dotyczące nagłówka	38
Utworzenie nagłówka	38
Licencja wtyczki	39
Określanie ścieżek dostępu	39
Ścieżki dostępu wtyczki	40
Lokalne ścieżki dostępu	40
Adresy URL	41
Aktywacja i dezaktywacja funkcji	42
Funkcja aktywacji wtyczki	42
Utworzenie ustawień domyślnych podczas aktywacji	43
Funkcja dezaktywacji wtyczki	43
Dezaktywacja to nie dezinstalacja wtyczki	44
Metody dezinstalacji	44
Dlaczego dezinstalacja wtyczki jest konieczna?	44
Plik uninstall.php	44
Zaczep uninstall	45
Standardy tworzenia kodu	46
Twórz dokumentację kodu	46
Nazwy zmiennych, funkcji i plików	47
Apostrof i cudzysłów	47
Wcięcia	48
Styl stosowania nawiasów	48
Używanie spacji	49
Skrócone znaczniki PHP	49
Polecenia SQL	49
Lista rzeczy do sprawdzenia podczas prac nad wtyczkami	49
Podsumowanie	50

Rozdział 3. Zaczepy	51
Akcje	52
Czym jest akcja?	53
Funkcje zaczepu akcji	54
Najczęściej używane zaczepy akcji	58
Filtry	61
Funkcje zaczepu filtru	63
Funkcje szybko zwracające wartość	67
Najczęściej używane zaczepy filtru	68
Używanie zaczepów z poziomu klasy	71
Tworzenie własnych zaczepów	72
Zalety utworzenia własnego zaczepu	73
Przykład utworzenia własnego zaczepu akcji	73
Przykład własnego zaczepu filtru	73
W jaki sposób wyszukiwać zaczepy?	75
Wyszukiwanie zaczepów w kodzie tworzącym jądro WordPress	75
Zaczepy zmienne	75
Listy zaczepów	76
Podsumowanie	76

Rozdział 4. Integracja z platformą WordPress	77
Dodawanie menu i podmenu	77
Utworzenie menu najwyższego poziomu	77
Dodawanie podmenu	78
Dodawanie elementu menu do już istniejącego menu	80
Tworzenie widgetów	82
Utworzenie widgetu	82
Widget zaawansowany	87
Tworzenie widgetów kokpitu	92
Utworzenie widgetu kokpitu wraz z opcjami	93
Pola użytkowników	96
Dodawanie własnego pola użytkownika	97
Zapis danych pola użytkownika	97
Zaawansowane pole użytkownika	101
Zachowanie spójności	106
Korzystanie z interfejsu użytkownika platformy WordPress	106
Podsumowanie	113
Rozdział 5. Internacjonalizacja	115
Internacjonalizacja i tłumaczenie na inne języki	115
Dlaczego warto przeprowadzać internacjonalizację?	116
Zrozumienie zagadnienia internacjonalizacji w profesjonalnej pracy	116
Przygotowanie wtyczki do tłumaczenia na inne języki	117
Wyświetlanie i zwracanie ciągów tekstowych	118
Używanie miejsc zarezerwowanych	125
Internacjonalizacja kodu JavaScript	127
Tworzenie plików tłumaczenia	130
Pliki MO i PO	130
Narzędzia służące do tłumaczenia	130
W jaki sposób utworzyć plik POT?	131
Gdzie przechowywać pliki tłumaczeń?	131
Podsumowanie	132
Rozdział 6. Bezpieczeństwo wtyczki	133
Zabezpieczenie wtyczki	133
Czym jest zapewnienie bezpieczeństwa wtyczce?	134
Czym nie jest zapewnienie bezpieczeństwa wtyczce?	134
Uprawnienia użytkownika	134
W jaki sposób używać funkcji <code>current_user_can()</code> ?	134
Nie sprawdzaj zbyt wcześnie	135
Unikalne identyfikatory	136
Uprawnienia kontra zamiary	136
Czym jest unikalny identyfikator?	137
Jak tworzyć i weryfikować unikalne identyfikatory?	137
Unikalne identyfikatory w skryptach Ajax	142
Weryfikacja i oczyszczenie danych	143
Potrzeba weryfikacji i oczyszczania danych	143
Dobra praktyka: identyfikacja potencjalnie niebezpiecznych danych	144
Weryfikacja czy oczyszczanie danych wejściowych?	146
Przykłady weryfikacji i oczyszczania danych	147

Formatowanie poleceń SQL	163
Obiekt \$wpdb	163
Dlaczego metody obiektu wpdb są lepsze?	163
Metody typu „wszystko w jednym”	164
Najczęściej stosowane metody	166
Ochrona zapytań przed atakami typu SQL Injection	170
Różne metody i właściwości obiektu wpdb	172
Dobre nawyki bezpieczeństwa	172
Podsumowanie	174

Rozdział 7. Ustawienia wtyczki 175

API Options	175
Zapisywanie opcji	175
Zapisywanie tablicy opcji	176
Pobieranie opcji	177
Wczytywanie tablicy opcji	178
Usuwanie opcji	178
Parametr autoload	179
API Settings	180
Zalety API Settings	181
Funkcje API Settings	181
Zebranie całości: pełna strona zarządzania wtyczką	185
Usprawnienie reakcji funkcji i weryfikacja błędów	187
Dodawanie pól na istniejącej stronie	188
API Transients	192
Zapisywanie opcji, która ma utracić ważność	192
Pobieranie opcji, która ma utracić ważność	192
Usunięcie opcji, która utraciła ważność	192
Praktyczny przykład użycia krótkotrwałych danych	193
Szczegółowe informacje techniczne	193
Idea krótkotrwałych danych	193
Zapisywanie ustawień poszczególnych użytkowników	194
Tworzenie wtyczki	194
Metadane użytkownika	194
Uaktualnianie metadanych użytkownika	195
Pobieranie metadanych użytkownika	196
Usunięcie metadanych użytkownika	196
Pobieranie identyfikatora użytkownika	197
Dodawanie pól na stronie profilu	197
Wtyczka BOJ Admin Lang	199
Ustawienia dla poszczególnych użytkowników — najlepsze praktyki	200
Przechowywanie danych we własnych tabelach	201
Typy danych	201
Standardowe tabele WordPress	202
Tworzenie własnej tabeli	202
Uaktualnienie struktury własnej tabeli	203
Uzyskanie dostępu do własnej tabeli	206
Podsumowanie	207

Rozdział 8. Użytkownicy	209
Praca z użytkownikami	210
Funkcje użytkownika	210
Tworzenie, uaktualnianie i usuwanie użytkowników	214
Dane użytkownika	218
Metadane użytkownika	223
Role i możliwości	229
Czym są role i możliwości?	230
Role domyślne	230
Własne role	231
Ograniczanie dostępu	231
Sprawdzanie uprawnień użytkownika	232
Czy użytkownik jest administratorem?	236
Nadanie własnych uprawnień	237
Dostosowanie ról do własnych potrzeb	238
Tworzenie roli	238
Usunięcie roli	239
Dodanie możliwości do roli	241
Usuwanie możliwości z roli	241
Wtyczka obsługująca własne role i możliwości	242
Podsumowanie	245
Rozdział 9. API HTTP	247
Szybki kurs wykonywania żądań HTTP	247
Czym jest żądanie HTTP?	247
Jak wykonywać żądania HTTP w PHP?	250
Funkcje obsługi HTTP oferowane przez WordPress	251
Funkcje rodziny wp_remote_*	252
Konfiguracja zaawansowana i wskazówki	257
Ćwiczenie praktyczne: odczyt formatu JSON z zewnętrznego API	263
Pobieranie i odczytywanie danych JSON	263
Funkcjonująca wtyczka	264
Ćwiczenie praktyczne: wysyłanie danych do zdalnego API	267
Formatowanie parametrów dla żądań POST	267
Gotowa wtyczka	268
Ćwiczenie praktyczne: odczyt dowolnej treści	269
Utworzenie własnego repozytorium wtyczki	270
Jak działa proces uaktualnienia wtyczki na platformie WordPress?	270
Wykonywanie żądań do alternatywnego API z poziomu wtyczki	272
Utworzenie alternatywnego API	274
Kilka ostrzeżeń dotyczących własnych API	276
Przypadek specjalny: pobieranie zdalnych wiadomości RSS	276
Podsumowanie	277
Rozdział 10. API Shortcode	279
Tworzenie skrótu	279
Czym jest skrót?	279
Rejestracja własnego skrótu	280
Wskazówki dotyczące skrótów	284
Pomysł o prostocie	284
Pamiętaj o dynamiczności	287

Wewnętrzny sposób działania	288
Kod BBCode we wtyczce obsługującej komentarze	290
Ograniczenia skrótów podczas obsługi struktur zagnieżdżonych	292
Integracja z usługą Google Mapy	293
Uzyskanie dostępu do API Google Geocoding	293
Przechowywanie wyników	295
Uzyskanie dostępu do API Google Maps	296
Więcej pomysłów dotyczących skrótów	300
Wyświetlanie treści jedynie dla zalogowanych użytkowników	301
Wyświetlenie treści ograniczonej czasowo	302
Zaciemnienie adresu e-mail	302
Podsumowanie	303

Rozdział 11. Rozbudowa wpisów bloga: metadane, własne typy wpisów bloga i taksonomie305

Tworzenie własnych typów wpisów bloga	306
Możliwe typy wpisów bloga	306
Rejestracja typu wpisu bloga	306
Ustawianie etykiet we własnym typie wpisu bloga	311
Wykorzystanie własnych możliwości	313
Dołączanie istniejących taksonomii	314
Używanie własnych typów wpisów bloga	315
Utworzenie pętli własnego typu wpisu bloga	315
Pobieranie treści własnego typu wpisu bloga	317
Sprawdzenie istnienia typu wpisu bloga	318
Metadane wpisu bloga	319
Dodawanie metadanych wpisu bloga	320
Pobieranie metadanych	321
Uaktualnienie metadanych wpisu bloga	321
Usuwanie metadanych	322
Tworzenie własnych taksonomii	323
Zrozumienie taksonomii	323
Rejestracja własnej taksonomii	324
Przypisanie taksonomii do typu wpisu bloga	329
Używanie własnych taksonomii	329
Pobieranie taksonomii	329
Używanie taksonomii wraz z wpisami bloga	330
Tagi warunkowe taksonomii	332
Wtyczka typu wpisu bloga oraz taksonomii	334
Podsumowanie	336

Rozdział 12. Technologie JavaScript i Ajax na platformie WordPress337

Krótkie wprowadzenie do jQuery	337
Zalety wynikające z używania jQuery	337
Krótki kurs jQuery	338
Technologia Ajax	341
Czym jest Ajax?	341
Najlepsze praktyki dotyczące technologii Ajax	344
Dodawanie kodu JavaScript do WordPress	345
Prawidłowy sposób dołączania skryptów	345
Gdzie umieszczać skrypty?	351

Dodawanie skryptów jedynie wtedy, gdy są potrzebne	353
Skrypty dynamiczne na platformie WordPress	357
Technologia Ajax na platformie WordPress	360
Technologia Ajax na platformie WordPress: reguły	360
Kompletny przykład: natychmiastowe odnośniki „Czytaj dalej”	362
Kolejny przykład: usunięcie komentarza ze strony	369
Usuwanie błędów podczas używania technologii Ajax	373
Podsumowanie	374
Rozdział 13. Cron	375
Czym jest cron?	375
W jaki sposób działa demon cron?	375
Tworzenie harmonogramu zadań cron	376
Utworzenie powtarzającego się zadania harmonogramu	376
Utworzenie jednorazowego zadania harmonogramu	378
Usunięcie zadania z harmonogramu	380
Zdefiniowanie własnych odstępów czasu	381
Wyświetlenie zadań harmonogramu cron	381
Prawdziwy cron	385
Przykłady praktyczne	385
Usuwanie co tydzień wcześniejszych wersji wpisu bloga	385
Wtyczka automatycznie wysyłająca wiadomość e-mail	390
Wtyczka usuwająca komentarze	393
Podsumowanie	399
Rozdział 14. API Rewrite	401
Dlaczego czasem trzeba zmieniać adresy URL?	401
Zasady dotyczące odnośników bezpośrednich	402
Moduł mod_rewrite serwera Apache	402
Zmiany adresów URL na platformie WordPress	403
W jaki sposób WordPress obsługuje zapytania?	404
Ogólny opis procesu wykonania zapytania	404
Obiekt rewrite	405
Obiekt query	405
Co można zrobić przy użyciu wtyczek?	406
Przykłady praktyczne	406
Zmiana adresu URL w celu utworzenia listy sklepów	407
Tworzenie nowej struktury odnośników bezpośrednich oraz integracja ze stronami, które nie powstały w WordPress	412
Wyświetlanie produktów sklepu	414
Dodawanie punktu końcowego i zmiana formatu danych wyjściowych	415
Dodanie własnego kanału wiadomości informującego o ostatnio dodanych obrazach	419
Podsumowanie	421
Rozdział 15. Sieć Multisite	423
Różnice	424
Standardowa konfiguracja WordPress kontra sieć Multisite	424
Zrozumienie terminologii sieci Multisite	424
Zalety sieci Multisite	425
Włączenie sieci Multisite na platformie WordPress	425

Funkcje sieci Multisite	427
Potęga identyfikatora bloga	427
Najczęściej używane funkcje	427
Przełączenie i przywracanie witryn internetowych	429
Przykłady skrótów uzyskania dostępu do treści sieci	432
Przykład widgetu z treścią sieciową	436
Utworzenie nowej witryny	443
Opcje witryny sieci Multisite	448
Użytkownicy w sieci	448
Rola Superadministratora w sieci Multisite	452
Sprawdzenie właściciela witryny	453
Dane statystyczne dotyczące sieci	454
Schemat bazy danych sieci Multisite	455
Tabele stosowane w sieci Multisite	455
Tabele przeznaczone dla konkretnych witryn	455
Podsumowanie	456
Rozdział 16. Usuwanie błędów i optymalizacja	457
Zapewnienie (lub nie) obsługi starszych wersji	457
Aktualizacja oprogramowania zgodnie z cyklem rozwojowym	
WordPress	458
Funkcje uznane za przestarzałe	459
Obsługa zbędnych instalacji	460
Usuwanie błędów	460
Włączenie trybu usuwania błędów	461
Wyświetlanie komunikatów związanych z usuwaniem błędów	461
Poprawianie błędów wskazywanych przez komunikaty	462
Rejestrowanie błędów	466
Włączenie rejestrowania błędów	466
Położenie pliku dziennika błędów	467
Plik dziennika błędów	467
Buforowanie	467
Zapisywanie, wczytywanie i usuwanie buforowanych danych	468
Buforowanie danych we wtyczce	469
Podsumowanie	471
Rozdział 17. Działania marketingowe	473
Wybór licencji dla wtyczki	474
Różne opcje	474
Dlaczego licencja ma znaczenie?	475
Zarabianie pieniędzy pomimo stosowania licencji GPL	476
Udostępnienie wtyczki na witrynie WordPress.org	477
Utworzenie konta	478
Zgłoszenie wtyczki do oficjalnego repozytorium	479
Konfiguracja SVN	479
Utworzenie pliku readme.txt	480
Rozślawienie wtyczki	483
Nadawanie nazwy wtyczce	483
Zbudowanie witryny internetowej	485
Utworzenie strony dla wtyczki	487
Ogłoszenie wydania wtyczki	487

Pomoc techniczna dla użytkowników wtyczki	488
Zbieranie informacji od użytkowników	488
Wyjście z piwnicy	490
Inne metody promocji	490
Podsumowanie	491
Rozdział 18. Narzędzia programisty	493
Jądro platformy jako punkt odniesienia	493
Dokumentacja osadzona na platformie	493
Wyszukiwanie funkcji	495
Najważniejsze pliki tworzące jądro platformy	495
Codex	497
Przeszukiwanie witryny Codex	497
Opis funkcji	498
Narzędzia oferowane przez inne witryny internetowe	498
PHPXref	498
Baza danych zaczepów platformy WordPress	500
Zasoby oferowane przez społeczność	500
Fora pomocy technicznej	500
Listy dyskusyjne	501
Czat WordPress	501
Informacje dotyczące prac rozwojowych nad WordPress	502
Zgłaszanie pomysłów dla WordPress	502
Obsługiwane przez społeczność witryny z nowościami	502
Wydarzenia lokalne	503
Narzędzia	503
Przeglądarka internetowa	504
Edytor tekstu	504
Obsługa plików za pomocą FTP, SFTP i SSH	505
phpMyAdmin	505
Podsumowanie	506
Skorowidz	507