

Earl Miles, Lynette Miles,
Emma Jane Hogbin, Karen Stevenson

KOMPONENTY TWORZĄCE SYSTEMU DRUPAL

Szybkie budowanie witryn internetowych za pomocą modułów CCK, Views i Panels



Opanuj możliwości najbardziej funkcjonalnych modułów dla systemu **Drupal** —
twórz atrakcyjne strony internetowe, w pełni dopasowane do Twoich potrzeb!

Jak szybko zdefiniować dane witryny za pomocą modułu Content Construction Kit?
Jak tworzyć panele nadpisujące domyślne układy stron
i stosować dla tych układów różne motywy?

Jak zbudować wielofunkcyjne formaty wyświetlania zapytań modułu Views?



» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

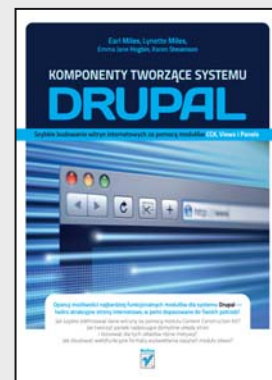
» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Komponenty tworzące systemu Drupal. Szybkie budowanie witryn internetowych za pomocą modułów CCK, Views i Panels

Autorzy: Earl Miles, Lynette Miles,
Emma Jane Hogbin, Karen Stevenson
Tłumaczenie: Piotr Pilch
ISBN: 978-83-246-3548-1

Tytuł oryginału: [Drupal's Building Blocks: Quickly Building Web Sites with CCK, Views, and Panels](#)
Format: 170×230, stron: 392



Opanuj możliwości najbardziej funkcjonalnych modułów dla systemu Drupal – twórz atrakcyjne strony internetowe, w pełni dopasowane do Twoich potrzeb!

- Jak szybko zdefiniować dane witryny za pomocą modułu Content Construction Kit?
- Jak tworzyć panele nadpisujące domyślne układy stron i stosować dla tych układów różne motywy?
- Jak zbudować wielofunkcyjne formaty wyświetlania zapytań modułu Views?

Chcesz szybko stworzyć wydajny sklep internetowy, elegancką witrynę firmową, sprawny intranet korporacyjny bądź funkcjonalny portal społecznościowy, ale już przeraża Cię ilość związanej z tym pracy? Zapewne dlatego zainteresowałeś się Drupalem? I słusznie. Jakie inne narzędzie open source umożliwia programistom tak błyskawiczną oraz bezproblemową realizację nawet bardzo ambitnych i zaawansowanych projektów IT? Dzięki zaangażowaniu utalentowanej i aktywnej społeczności Drupala masz przecież możliwość korzystania z całego bogactwa tysięcy praktycznych modułów, które rozszerzają możliwości rdzenia tego systemu i pozwalają na niemal nieograniczone wzbogacanie projektu o kolejne przydatne funkcjonalności. Choć trudno w to uwierzyć, już kilka kliknięć wystarczy, aby tworzyć witryny internetowe o elastycznych, zróżnicowanych możliwościach.

Niezależnie od tego, czy jesteś programistą, projektantem, administratorem, architektem czy konsultantem – oto doskonały przewodnik po najbardziej wartościowych modułach i technikach włączania ich do swojego projektu. Znajdziesz tu słynną "wielką trójkę", czyli moduły Content Construction Kit (CCK), Views i Panels, których potencjał pozwala na niesłychanie szybkie tworzenie idealnie skrojonych stron. Niemożliwe? Przekonaj się, że tak! Współautor tej niezwyklej książki, Earl Miles, od wielu lat jest aktywnym członkiem społeczności związanej z Drupalem, dzięki czemu w dużym stopniu wpłynął na kierunek, w jakim podąża ten system. Earl jest także kluczowym uczestnikiem projektu budowy modułu CCK oraz autorem modułów Views i Panels. Trudno zatem wyobrazić sobie lepszą osobę, która mogłaby wprowadzić Cię w tajniki ich skutecznego wykorzystania! Autorzy zamieścili tu wszystkie informacje, których nie znajdziesz nigdzie indziej, a także obszerny przykładowy kod „recepturowy”, stworzony w celu łatwej adaptacji i ponownego wykorzystania.

Błyskawicznie twórz elastyczne, wielofunkcyjne witryny internetowe!

Spis treści

Słowo wstępne	13
Przedmowa	17
O autorach	23
Część I Moduły Content Construction Kit	25
Rozdział 1 Wprowadzenie do modułów CCK i segmentów	27
System segmentów	27
Dlaczego segmenty odgrywają istotną rolę?	28
Domyślne typy zawartości	29
Elementy segmentu	30
Dlaczego pola są dodawane do segmentów?	32
W poszukiwaniu Graala: jak powstał pakiet CCK?	33
Rozpoczęcie pracy z modułami CCK	36
Tworzenie nowego typu zawartości	37
Ćwiczenie 1.1.	39
Podsumowanie	42
Rozdział 2 Zagadnienia związane z polami	43
Co należy wiedzieć przed utworzeniem pól i typów zawartości?	43
Strony typów zawartości	44
Sekcja Identification (identyfikacja)	44
Sekcja Submission form settings (ustawienia formularzy wysyłania)	46
Ustawienia sekcji Workflow Settings (ustawienia przepływu zadań)	47
Sekcja Comment settings (ustawienia komentarzy)	48
Karty Fields (pola), Export (eksport) i Import (import)	50
Tworzenie nowych pól typu zawartości	50
Dodawanie pól	51
Typy danych	53
Ćwiczenie 2.1.	55
Kontrolki	57

Składanie elementów w całość	57
Ćwiczenie 2.2.	58
Zmiana sposobu wyświetlania pola	61
Współużytkowanie pól	64
Ćwiczenie 2.3.	65
Podsumowanie	66
Rozdział 3 Pola — dodatkowe informacje	67
Wybór typów pól	67
Zastosowanie tekstu	68
Zastosowanie typów liczbowych	68
Zastosowanie typu Node Reference	69
Ćwiczenie 3.1.	70
Zastosowanie typu User Reference	71
Ograniczanie danych za pomocą kontroltek	72
Ćwiczenie 3.2.	76
Moduł Date	77
Ćwiczenie 3.3.	79
Pola obliczeniowe	80
Ćwiczenie 3.4.	81
Pola modułów Link i Email	84
Wizualne środki przekazu	85
FileField	85
ImageField	86
Moduły ImageAPI, ImageCache i ImageCache UI	87
Podsumowanie	87
Rozdział 4 Motywy i moduły CCK	89
Motywy — podstawy	89
Ćwiczenie 4.1.	90
Stosowanie motywów powiązanych z pakietem CCK	92
Elementy formatujące	92
Szablony pól	92
Ćwiczenie 4.2.	95
Szablony segmentów	96
Ćwiczenie 4.3.	97
Wykluczanie pól	99
Node Reference	100
Moduły pomocnicze	101
Theme Developer	101
Ćwiczenie 4.4.	102
Contemplate	103
Podsumowanie	104

Rozdział 5 Interfejs API modułów CCK105

Zastosowanie interfejsu API modułów CCK	105
Model pól	106
Tworzenie nowych typów pól, typów kontrolek i elementów formatujących	107
Tworzenie niestandardowych typów pól, typów kontrolek lub elementów formatujących	108
Ćwiczenie 5.1.	109
Moduły typów pól	109
Ćwiczenie 5.2.	110
Moduły typów kontrolek	115
Moduły elementów formatujących	117
Tworzenie instancji pól za pomocą modułu Content Copy	120
Tworzenie instancji pól za pomocą interfejsu API funkcji CRUD	121
Tworzenie danych dla pól modułów CCK	124
Różne funkcje pomocnicze	127
Podsumowanie	128

Część II Moduł Views129**Rozdział 6 Relacyjne bazy danych131**

System Drupal, język SQL i pojawienie się modułu Views	131
Wymowa skrótu SQL	131
„Urwisko uczenia” systemu Drupal	132
Podstawy relacyjnych baz danych	133
Wiersze i pola	133
Klucze	134
Filtrowanie i sortowanie	136
Filtrowanie	136
Sortowanie	136
Złączenia	137
Od języka SQL, poprzez moduł Views, do języka zrozumiałego przez człowieka	138
Podsumowanie	138

Rozdział 7 Tworzenie widoków139

Interfejs użytkownika modułu Views	139
Strona List (lista)	140
Strona Add (dodaj)	143
Ćwiczenie 7.1.	144
Strona Import (importuj)	157
Strona Tools (narzędzia)	158
Ćwiczenie 7.2.	161

Publiczne prezentowanie widoków:	
tworzenie formatów wyświetlania	164
Ćwiczenie 7.3.	164
Ćwiczenie 7.4.	165
Bloki	166
Ćwiczenie 7.5.	166
Ćwiczenie 7.6.	168
Załączniki	169
Kanał	169
Podsumowanie	170
Rozdział 8 Argumenty, filtry ujawniane i zależności	171
Argumenty	171
Argumenty w roli filtrów	172
Konfigurowanie argumentu	173
Ćwiczenie 8.1.	175
Użycie argumentów jako części widoku	177
Ćwiczenie 8.2.	177
Ćwiczenie 8.3.	180
Filtry ujawniane	181
Ćwiczenie 8.4.	182
Zależności	183
Ćwiczenie 8.5.	184
Ćwiczenie 8.6.	186
Podsumowanie	187
Rozdział 9 Stosowanie motywu dla widoków	189
Przegląd	189
Klasy modułu Views	190
Ćwiczenie 9.1.	191
Pliki szablonów	192
Szablony formatów wyświetlania	196
Style widoków	197
Szablony wierszy	198
Inne szablony	200
Praca z szablonami	202
Ponowne skanowanie plików szablonów	203
Ćwiczenie 9.2.	204
Debugowanie	205
Wyświetlanie domyślnych komunikatów dla pustych pól	206
Ćwiczenie 9.3.	206
Grupowanie w szablonie	208
Ćwiczenie 9.4.	208
Ćwiczenie 9.5.	211
Podsumowanie	213

Rozdział 10 Optymalizacja zapytań215

Zrównoważenie czasu projektowania z czasem obciążenia procesora	215
Możliwości oferowane przez moduł Views	216
Gdy moduł Views nie wystarcza	216
Określanie wydajności zapytania	216
Ćwiczenie 10.1.	217
Osadzanie zapytań	217
Instrukcja EXPLAIN	218
Porównanie indeksowania z buforowaniem	220
Eksperymentowanie z witryną	221
Podsumowanie	222

Rozdział 11 Interfejs API modułu Views223

Architektura danych	223
Programowanie obiektowe	224
Tabele bazowe i zależności	229
Obiekty wykorzystywane w widoku	230
Interfejs API modułu Views	239
Cykl istnienia widoku	240
Cykl wykonywania widoku	240
Wykonywanie formatu wyświetlania modułu Views	243
Haki związane z wykonywaniem	245
Hak związany z danymi i schematem bazy danych	246
Tworzenie zależności między tabelami	246
Deklarowanie tabel w haku hook_views_data()	247
Deklarowanie pól tabel	250
Porównanie procedur obsługi z dodatkami	251
Procedury obsługi	251
Dodatki	253
Podsumowanie	255

Część III Moduł Panels257**Rozdział 12 Moduł Panels — wprowadzenie259**

Wprowadzenie do modułu Panels	259
Krótka historia modułu Panels	259
Przekazywanie i pobieranie: różnice w działaniu modułu Panels	260
Układ typu wskaź i kliknij	262
Kontekst	263
Architektura umożliwiająca podłączanie	264
Moduły	265
Pakiet modułów Panels	265
Chaos Tool Suite	266
Podsumowanie	267

Rozdział 13 Tworzenie paneli	269
Pierwszy panel	269
Karta Dashboard (panel kontrolny) na stronie Panels (panele) ...	270
Strony paneli	271
Ćwiczenie 13.1.	276
Segmenty paneli	279
Minipanele	282
Dodawanie zawartości do paneli	284
Dodawanie sekcji zawartości	284
Ćwiczenie 13.2.	287
Buforowanie	289
Konfigurowanie istniejących sekcji zawartości	289
Reguły dostępu	291
Ćwiczenie 13.3.	291
Nadpisywanie podstawowych stron	293
Widok segmentu	294
Ćwiczenie 13.4.	296
Taksonomia	298
Ćwiczenie 13.5.	299
Widok użytkownika	300
Nadpisywanie podstawowych stron edycji segmentu	301
Podsumowanie	302
Rozdział 14 Kontekst, zależności i argumenty w module Panels	303
Konteksty	303
Konteksty na stronach paneli	304
Ćwiczenie 14.1.	305
Taxonomy	308
Ćwiczenie 14.2.	309
Argumenty w module Panels	311
Ćwiczenie 14.3.	312
Okno modalne Add content (dodaj zawartość) dla widoków	313
Formaty wyświetlania sekcji widoku	315
Ćwiczenie 14.4.	316
Ćwiczenie 14.5.	318
Zależności	319
Użycie zależności	319
Typy zawartości User Reference (odwołanie do użytkownika) i Node Reference (odwołanie do segmentu)	320
Podsumowanie	321

Rozdział 15 Moduł Panels — stosowanie motywów	323
Układ	323
Elastyczny układ	323
Zmiana układów	327
Stylizer	328
Praca ze stylami	330
Ćwiczenie 15.1.	330
Ćwiczenie 15.2.	331
Arkusze stylów w interfejsie użytkownika modułu Panels	332
Ćwiczenie 15.3.	333
Arkusze stylów CSS w kodzie źródłowym	334
Identyfikowanie konkretnej sekcji	335
Inne zmiany stylistyczne	336
Podsumowanie	337
Rozdział 16 Wdrażanie witryny	339
Konfigurowanie środowiska wdrażania	339
Porównanie zawartości i struktury	339
Kontrola źródeł	340
Przenoszenie do środowiska produkcyjnego	341
Oddzielenie wersji projektowej od produkcyjnej	341
Testowanie zmian	342
Dokumentowanie wyników pracy	343
Eksportowanie struktur	343
Eksportowanie modułów CCK	343
Ćwiczenie 16.1.	344
Eksportowanie widoków	345
Eksportowanie paneli	346
Moduły pomocnicze	347
Deploy	347
Features	348
Drush	348
Podsumowanie	348
Dodatki	349
Dodatek A Inne przydatne moduły	351
Rozszerzanie zakresu zastosowania modułów	351
Moduły ogólne	352
Pathauto	352
Views	352
Views_or	352
Nodequeue	353
Flag	353

Views Slideshow	354
Views Bonus Pack	354
Views Attach	354
Views Import	355
ApacheSolr Views	355
SimpleViews	355
Views Bulk Operations	356
Views Datasource	356
Sheetnode	356
CCK	357
Calendar	357
Panels	357
Advanced Profile Kit	357
Total Control Admin Dashboard	358
Dodatek B Zgłaszanie problemów	359
Wysyłanie kompletnego raportu	359
Czytaj dokumentację	361
Sprawdzaj inne źródła	361
Różnica między błędem i żądaniem wsparcia	362
Trzymaj się tematu	363
Cykl istnienia błędu	363
Bądź cierpliwy	364
Pamiętaj, że prosisz kogoś innego o poświęcenie swojego czasu	365
Zaangażuj się czynnie	365
Dodatek C Procedury obsługi i dodatki interfejsu API modułu Views	367
Procedury obsługi modułu Views	367
Procedury obsługi pól	367
Procedury obsługi sortowania	368
Procedury obsługi filtrów	368
Procedury obsługi dla argumentów	368
Procedury obsługi zależności	369
Dodatki modułu Views	369
Dodatki formatów wyświetlania	369
Dodatki stylów	370
Dodatki wierszy	370
Klasy modułu Views	370
Skorowidz	381

Motywy i moduły CCK

Po utworzeniu treści pora nadać jej profesjonalny wygląd i zwiększyć jej czytelność. Pakiet CCK znakomicie się sprawdza, umożliwiając dodawanie przeróżnej dostosowanej treści. Jednakże nie pozwala on na wyświetlanie danych w przejrzysty sposób, który zapewniłby użytkownikom możliwość wygodnego czytania. W tym rozdziale przyjrzymy się systemowi motywów, a także sposobowi jego współpracy z pakietem CCK. Zajmiemy się również modulem *Contemplate*, który może być pomocny. System szablonów i motywów w wersji 6. Drupala prezentuje spore osiągnięcia mające na celu ułatwienie stosowania motywów dla witryny. Na końcu omówimy dodatkowe moduły pomocnicze, które jeszcze bardziej upraszczają obsługę motywów w systemie Drupal.

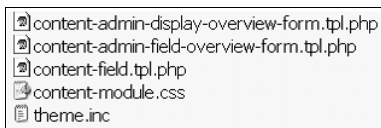
Motywy — podstawy

Motywy są złożone z dwóch podstawowych części: funkcji i szablonu motywów. Drupal i jego moduły przekazują informacje warstwie motywów, tak by możliwe było ich wyświetlenie. Jeśli pracujesz z plikami poza obrębem plików szablonów motywu, postępujesz niewłaściwie. Pliki szablonów mają rozszerzenie *.tpl.php*. Plik szablonu może znajdować się w kilku miejscach instalacji. Choć podstawowe pliki są umieszczone w katalogu motywów, każdy moduł może też używać katalogu z motywami, który zawiera bardziej specyficzne informacje dla każdej części modułu. Dla przypomnienia, *Garland* oraz jego motyw podrzędny *Minelli* są używane na potrzeby instalacji i aktualizacji. Modyfikowanie ich nie jest wskazane.

Przeglądając się różnym katalogom, zauważysz kilka plików o tej samej nazwie. Są to pliki domyślne, których system Drupal użyje, jeśli sam nie wprowadzisz żadnych zmian. Drupal korzysta z motywu, takiego jak *Garland*, aby dla treści zastosować styl. Wiele modułów, takich jak CCK, *Views* i inne, zapewnia dodatki w celu rozszerzenia możliwości systemu Drupal związanych z motywami. Na rysunku 4.1 pokazano kilka plików szablonów używanych przez domyślny motyw systemu Drupal, czyli *Garlanda*. Na rysunku 4.2 zaprezentowano katalogów motywów pakietu CCK, który zawiera pliki nadpisujące style motywu podstawowego.



Rysunek 4.1. Katalog /themes/garland



Rysunek 4.2. Katalog /cck/theme

Aby zmodyfikować jeden z motywów podstawowych, należy wykonać kilka kroków w celu zachowania motywu oraz zapewnienia sobie spokoju ducha.

Ćwiczenie 4.1.

Zachowywanie motywu podstawowego

Wprawdzie formalnie rzecz biorąc, zachowywanie motywu podstawowego nie stanowi elementu stosowania motywów dla pakietu CCK, warto wiedzieć, że jest to najbezpieczniejsza metoda modyfikowania motywu podstawowego. Jeśli zaczniesz wprowadzać zmiany w motywie administracyjnym, pojawia się ryzyko przypadkowego ukrycia treści, spowodowania, że będzie ona zbyt mała lub zbyt duża, aby ją odczytać, lub wykonania dowolnej liczby nieciekawych rzeczy, które są trudne do monitorowania w arkuszu stylów.

1. Przejdź do katalogu /sites/all/themes. Katalog /sites/all/themes powinien już istnieć. W przeciwnym razie utwórz go.
2. Utwórz nowy katalog dla nowych plików motywów. Te pliki nadpiszą informacje w motywie podstawowym. Ponieważ w moim przypadku motyw podstawowy to *Garland*, w celu uproszczenia procedury utworzyłem katalog o nazwie *garland_extend*. A zatem kompletna ścieżka ma postać: */sites/all/themes/garland_extend*.

3. Utwórz plik *.info* dla rozszerzenia motywu. Ten krok jest istotny! Bez niego motyw nie uzyska informacji pozwalających mu znaleźć motyw podstawowy. Nazwa pliku *.info* musi być zgodna z nazwą katalogu motywu. Na potrzeby tego ćwiczenia użyj nazwy *garland_extend.info*.
4. Dodaj do pliku *.info* informacje, które umożliwią motywowi znalezienie jego motywu podstawowego, a ponadto pozwolą witrynie skorzystać z tego motywu. Oto przykład:

```
name = garland_extend
description = Motyw pozbawiony tabeli, umożliwiający zmianę kolorów, wielokolumnowy ze stałą szerokością.
version = VERSION
core = 6.x
base theme = garland
stylesheets[all][] = garland_extend.css
```

5. Utwórz plik CSS, który zostanie użyty przez motyw na potrzeby modyfikacji. W poprzednim kroku widać, że mój plik będzie miał nazwę *garland_extend.css*. Jeśli chcesz teraz utworzyć plik, nie ma przeciwwskazań. Będzie to pusty plik do momentu zdecydowania o wprowadzeniu zmian.

Arkusze stylów w motywach

Dobłą praktyką jest dodawanie nazwy arkusza stylów do pliku *.info*, nawet jeśli sam plik CSS jest pusty. Jeżeli nie podasz nazwy arkusza stylów, domyślnie motyw użyje pliku *style.css*.

Pojawiający się w tym przypadku potencjalny problem polega na tym, że wiele motywów korzysta z pliku *style.css* jako standardowego arkusza stylów. Utworzony przez użytkownika nowy pusty arkusz stylów nadpisze ten standardowy. Takie rozwiązanie jest świetne, jeśli zamierzasz całkowicie usunąć stary arkusz. Jednakże spowoduje problem, gdy nowy arkusz nie będzie zawierał żadnego kodu CSS.

Nowy folder motywu zawiera tylko dwa pliki: *garland_extend.css* i *garland_extend.info*.

6. Możesz teraz włączyć „nowy” motyw w witrynie. Zmiany dokonane w tym motywie będą obowiązywały. Ponadto będziesz miał możliwość użycia oryginalnego motywu *Garland*, gdy nastąpi konieczność zaktualizowania go!
 7. Przejdź do pozycji *Administer/Site building/Themes* (administruj/budowanie witryny/motywy). Zaznacz pole wyboru widoczne obok nowego motywu (*garland_extend*) i wybierz przycisk opcji znajdujący się obok tego motywu, aby stał się on domyślny. Spowoduje to włączenie motywu.
-

Stosowanie motywów powiązanych z pakietem CCK

Dane generowane przez moduły CCK można podzielić na trzy poziomy tworzenia motywu: elementy formatujące, szablony pól i szablony segmentów. Każdy z tych poziomów można ponownie podzielić na części, które mają wpływ na określone cechy wyświetlanej strony lub stron. Pamiętaj o tym, że mechanizm motywów renderuje informacje dotyczące formatowania w kolejności od najbardziej do najmniej szczegółowego szablonu. Szczegółowy motyw nadpisuje bardziej ogólne szablony. Motyw dla jednego konkretnego pola (np. nazwa użytkownika) ma pierwszeństwo względem informacji zawartych w ogólnym motywie stosowanym dla wszystkich pól. Pliki nadpisujące pliki domyślnego motywu są nazywane **sugestiami**. Sugestie to bardziej szczegółowe pliki przeznaczone dla motywów.

Elementy formatujące

W rozdziale 2. przyjrzelśmy się elementom formatującym pola w określonych typach zawartości. Dokonując szybkiego przeglądu kluczowych kwestii, należy wspomnieć, że elementy formatujące umieszczone w interfejsie użytkownika wykonują proste operacje rozmieszczania pól. W celu uzyskania dostępu do tych elementów skorzystaj z następującej ścieżki nawigacyjnej: *Home/Administer/Content management/(wybierz typ zawartości)/Display fields* (strona główna/administruj/zarządzanie treścią/(wybierz typ zawartości)/wyświetl pola).

Elementy formatujące używają domyślnej warstwy motywów systemu Drupal, która umożliwia zastosowanie dla pól pakietu CCK wszystkich nakładek, szablonów i innych funkcji motywów systemu Drupal. Wiele z dostarczanych elementów formatujących korzysta z funkcji motywów. Ogólnie element formatujący służy do umieszczenia danych wybranego pola w odpowiednim kodzie HTML w celu wygenerowania danych wyjściowych.

Szablony pól

Jednym z aspektów stosowania motywów dla pakietu CCK (i modułu *Views*!) oferujących największe możliwości jest opcja tworzenia i używania motywu dla każdego pojedynczego pola w przypadku każdej wyświetlanej strony. Prawdopodobnie stwarza to znacznie większe możliwości kontroli, niż kiedykolwiek będą wymagane. Choć szablony pól nie są ściśle powiązane z pakietem CCK (lub modułem *Views*), poznanie ich umożliwia znacznie szybsze określenie miejsca, w którym konieczne będzie podzielenie motywu.

Szablony pól i elementy formatujące w pewnym stopniu wykonują podobne funkcje, ponieważ obsługują ten sam typ danych. Jednakże szablony pól koncentrują się w o wiele większym stopniu na kwestii „Jak to pole wygląda?” niż na kwestii „Gdzie znajduje się etykieta i czy w ogóle powinna zostać wyświetlona?”. Element formatujący pozwala umieścić etykietę Cena na stronie. Szablon i kod CSS umożliwiają sformatowanie rzeczywistej ceny 19,99 zł pogrubioną czcionką i kolorem zielonym.

Wspomnieliśmy już, że pliki szablonów mają rozszerzenie *.tpl.php*, a ponadto zaprezentowaliśmy kilka przykładów nazw szablonów. Typy pól pakietu CCK pozwalają na użycie wyjątkowo szczegółowych nazw na potrzeby stosowania motywu dla poszczególnych pól. Struktura nazw jest zgodna z konwencją, która ułatwia zidentyfikowanie przeznaczenia pola.

Ostrzeżenie

Nigdy nie powinieneś modyfikować plików domyślnego motywu znajdujących się w katalogu dowolnego modułu (np. */sites/all/modules/cck/theme*). Wywoła to rozproszenie danych motywu w obrębie instalacji i może spowodować spore zamieszanie, gdy później konieczne będzie wprowadzenie zmian.

Zawsze kopiuj plik domyślny do katalogu przetwarzanego motywu (*/sites/all/modules/nazwa_modułu*) i tam modyfikuj ten plik.

Pliki szablonów i nazwy plików

Plik ogólnego szablonu dla pola nosi nazwę *content-field.tpl.php*. Jeśli nie dysponujesz bardziej szczegółowymi szablonami motywów, ten plik będzie używany dla układu zawartości pól. Moduł *Devel* może być pomocny w uzyskaniu bardziej szczegółowych plików motywów. Domyślny plik *content-field.tpl.php* można znaleźć w katalogu */cck/theme*.

Następnym krokiem na drodze do zwiększenia unikalności motywu jest zmiana pola dla wszystkich typów zawartości. Zauważ, że jeśli pole nie jest współużytkowane w wielu typach zawartości, ta operacja w rzeczywistości nie będzie miała dużego wpływu. Z kolei gdy pola są współużytkowane w kilku typach zawartości, jest to dobry sposób na zmianę układu pola w obrębie całej witryny. Niestety zastosowanie tej metody często wywołuje zdziwienie z powodu tego, dlaczego pole typu zawartości A zostało zmienione po utworzeniu pliku szablonu dla pola typu zawartości B. Konwencja nazewnictwa plików z rozszerzeniem *.tpl* przedstawia się następująco: *content-field-<NAZWA_POLA>-<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php*. W przypadku typu zawartości *Joe's Shirts* (koszulki Janka) plik szablonu dla pola *T-shirt size* (rozmiar koszulki) ma nazwę *content-field-field_size.tpl.php*.

Kolejna operacja zwiększania szczegółowości pozwala określić widok dla wszystkich pól dla konkretnego typu segmentu. Ten krok zapobiega przypadkowemu nadpisaniu współużytkowanych typów pól przez osoby stosujące motywy, lecz powoduje też, że widok nie jest udostępniany. Jeśli współużytkowane pole dla jednego typu zawartości jest pokazywane w określony sposób, w przypadku drugiego typu zawartości może być prezentowane w inny. W przykładzie plik o formacie nazwy *content-field-<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php* przyjmuje nazwę *content-field-t_shirts.tpl.php* dla typu zawartości *Joe's Shirts* (koszulki Janka).

Na najniższym poziomie możesz utworzyć dane motywu dla konkretnego pola jednego typu zawartości. W przypadku tego typu pliku format nazwy ma postać *content-field-<NAZWA_POLA>.tpl.php*.

Podsumowując, pakiet CCK zawsze dodaje słowo *field* na początku każdej nazwy pola, mimo że w nazwie plików szablonów już występuje słowo *field*. W przypadku stosowania motywów dla pakietu CCK podczas wyszukiwania tych plików obowiązuje następująca kolejność pierwszeństwa:

```
content-field-<NAZWA_POLA>-<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php
content-field-<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php
content-field-<NAZWA_POLA>.tpl.php
content-field.tpl.php
```

Jeśli żaden z tych plików nie jest obecny w folderze motywu użytkownika (lub w folderze podstawowego motywu), pakiet CCK użyje szablonu swojego domyślnego motywu.

Zmienne dostępne w szablonie pól

Zmienne dostępne w szablonie pól nie powinny być zaskoczeniem dla nikogo, kto zaznał się ze strukturą segmentu systemu Drupal lub wykonał jakąkolwiek czynność związaną ze stosowaniem motywu. Poniższa lista pochodzi bezpośrednio z pliku *content-field.tpl.php*. Poszerzono ją o kilka opisów.

- `$node` — obiekt całego segmentu.
- `$field` — tablica pól, która zawiera wszystkie pola dostępne w segmencie.
- `$items` — tablica zawierająca wartości wszystkich elementów tablicy pól. Każda wartość w tej tablicy jest w pełni sformatowana i przefiltrowana.
- `$teaser` — określa, czy dane pole zawartości jest wyświetlane w skrócie artykułu.
- `$page` — określa, czy dane pole zawartości jest wyświetlane na stronie całego artykułu. Pole może znajdować się w skrócie artykułu lub na stronie całego artykułu, a także w obu miejscach lub w żadnym z nich.
- `$field_name` — nazwa określonego pojedynczego pola.
- `$field_type` — typ danych, które zawiera pole.
- `$field_name_css` — nazwa, której kod CSS użyje podczas odwoływania się do danego pola. Wartości zmiennych `$field_name_css` i `$field_type_css` są takie same jak wartości zmiennych `$field_name` i `$field_type` z tym wyjątkiem, że w przypadku wartości zmiennej `$field_name_css` pola znaki podkreślenia (`_`) są zastępowane kreskami (`-`), gdy strona jest generowana w celu wyświetlenia. Oznacza to, że dla pola `field_shirt_size` w pakiecie CCK klasa CSS będzie miała nazwę `field-shirt-size`.
- `$field_type_css` — typ danych, którego kod CSS użyje podczas odwoływania się do pola. W odniesieniu do znaków podkreślenia i kresek ta zmienna zachowuje się tak samo jak zmienna `$field_name_css`.
- `$label` — etykieta nadana elementowi, taka jak *T-shirt size* (rozmiar koszulki).

- `$label_display` — miejsce wyświetlania etykiety określone przez element formatujący. Etykieta może być pokazana w wierszu, ponad nim lub zostać ukryta.
- `$field_empty` — wartość TRUE lub FALSE. Dla zmiennej jest ustawiana wartość TRUE, jeśli pole nie zawiera danych do wyświetlenia.

Ćwiczenie 4.2.

Zmiana części motywu: kolor pola

Sklep Joe's Clothes (ubrania Janka) zamierza sprawić, by dostępne rozmiary dla każdego typu koszulki stały się doskonale widoczne dla klientów. Firma chciałaby też, aby rozmiary były podawane w kolorze czerwonym.

1. Określ poprawny plik szablonu, który zostanie użyty. Na razie przyjmijmy, że jeśli Janek rozbuduje sklep, może oczekiwać zapewnienia identycznego wyglądu wszystkim polom rozmiaru. W tym przypadku najlepszym kandydatem do spełnienia tych wymagań jest plik `content-field-<NAZWA_POLA>.tpl.php`.
2. Skopiuj plik `content-field.tpl.php` do katalogu `/sites/all/modules/theme/garland_extend` i zmień jego nazwę na `content-field-field_shirt_size.tpl.php`.
3. Musisz teraz zmodyfikować plik CSS, aby określić nowy kolor dla rozmiaru. Używany plik arkusza stylów nosi nazwę `garland_extend.css`. Zostanie on wywołany przez motyw. Otwórz plik CSS i wprowadź następujące informacje dotyczące stylu:

```
div.field-field-shirt-size div.field-items {
  color: #FF0033;
}
```

4. Zapisz plik.

Po ponownym załadowaniu strony dowolnej koszulki, która udostępniała informacje o rozmiarach, będą one wyróżnione na czerwono. Możesz wymusić operację pełnego odświeżania, ponieważ wiele przeglądarek będzie buforować arkusze stylów.

Być może zastanawiasz się, skąd uzyskać nazwę elementu `div` w celu dokonania zmiany koloru. Plik szablonu dla pola zawiera wszystkie wymagane informacje. Oto fragment pliku `content-field-field_shirt_size.tpl.php`:

```
<div class="field field-type-<?php print $field_type_css ?> field-<?php
print $field_name_css ?>">
  <?php if ($label_display == 'above') : ?>
    <div class="field-label"><?php print t($label) ?>&nbsp;  </div>
  <?php endif;?>
<div class="field-items">
```

Ostatni wiersz jest najbardziej interesujący. Zawiera on łańcuch `field-items` oznaczający wartości wewnątrz samego pola. Ponieważ konieczne jest wyizolowanie zmiany koloru tekstu tak, aby używała tylko tych wartości, wymagana jest dokładność. Selektor `div.field-field-shirt-size` izoluje styl tylko dla pola rozmiaru koszulki, a selektor `div.field-items` dodatkowo poszerza to ograniczenie w celu dokonania zmiany wyłącznie wartości zamiast zarówno wartości, jak i etykiety. Jeśli zamierzasz zmienić etykietę, możesz użyć następującego selektora CSS, aby w polu `shirt-size` określić tylko etykietę:

```
div.field-field-shirt-size div.field-label
```

Szablony segmentów

Szablony segmentów pełnią rolę dużych opakowań dla wszystkich elementów treści. Szablony są integralną częścią systemu motywów rdzenia. Szablon segmentów zawiera dane i informacje o rozmieszczeniu, a ponadto jest jednym z łatwiejszych do utworzenia elementów motywu w sytuacji, gdy nie wymagasz zastosowania wielu kolorów dla tekstu lub masz specjalne oczekiwania względem obrazów i innych danych.

Większość motywów będzie uwzględniać domyślny szablon segmentów o nazwie *node.tpl.php*. System motywów rdzenia systemu Drupal umożliwia zmianę tego stanu rzeczy w celu zawężenia zasięgu motywu. Domyślny szablon *node.tpl.php* ma wpływ na wygląd wszystkich segmentów. Jeśli oczekujesz czegoś trochę innego dla wybranego typu zawartości, możesz po prostu utworzyć nowy szablon przez skopiowanie pliku domyślnego i zmienić nazwę pliku do postaci *node-<TYP_ZAWARTOŚCI>.tpl.php*. Aby zmodyfikować motyw dla wszystkich segmentów artykułów, należałoby użyć pliku szablonu segmentów o nazwie *node-story.tpl.php*. Po utworzeniu tego szablonu możesz zmienić wygląd wszystkich artykułów bez wpływu na resztę systemu!

Pakiet CCK zapewnia kilka zmiennych dla szablonu segmentów, które obsługują większość scenariuszy stosowania motywu dla segmentu.

```
<NAZWA_POLA>_rendered
```

Ta zmienna zapewnia w pełni renderowane dane wyjściowe wybranego pola uzupełnione o kod HTML. Są to ostateczne dane wyjściowe szablonu pola. Renderowane dane wyjściowe uwzględniają wszystkie wartości pola, a także każde ustawienie, które zostało utworzone przez elementy formatujące.

```
<NAZWA_GRUPY>_rendered
```

Podobnie do zmiennej `<NAZWA_POLA>_rendered` ta zmienna generuje w pełni renderowane dane wyjściowe. W tym przypadku dotyczy to całej grupy pól (jeśli jest używana). W przykładowym typie zawartości *Batch brewed* (partia warzonego piwa) użyto grupy pól *Alcohol level* (poziom alkoholu). Ta zmienna może być użyta w szablonie segmentów w celu dokonania jednoczesnej zmiany wszystkich pól w grupie z wykorzystaniem zmiennej `alcohol_level_rendered`. W zasadzie zmienna `<NAZWA_GRUPY>_rendered` jest podzbiorem zmiennej `<NAZWA_POLA>_rendered`.

`$NAZWA_POLA`

Ta zmienna zawiera dwie grupy danych. Jedna z grup uwzględnia tylko nieprzetworzone dane bez żadnych informacji elementu formatującego zapewnianych wcześniej opisany zmiennym. Dane te są udostępniane w tym samym formacie tablicy, który jest używany wewnętrznie przez pakiet CCK. Każdorazowo gdy ta tablica jest stosowana, zawartość może być inna w zależności od typu pola, dla którego jest używany motyw.

Zmienna `$NAZWA_POLA` zawiera drugi zestaw danych przechowywanych w elemencie `view`. Ten element w rzeczywistości nie jest powiązany z modulem *Views*, dlatego nie należy mylnie sugerować się nazwą. Element zawiera te same informacje, które zostałyby znalezione w zmiennej `<$NAZWA_POLA>_rendered`. Dzięki temu dysponujesz czystymi danymi dostępnymi do wyświetlenia, a także nieprzetworzonymi danymi nadającymi się do modyfikowania.

Uwaga

Nieprzetworzone dane zapewniane przez zmienną `$NAZWA_POLA` nie są okrajane na potrzeby wyświetlania! Takie dane mogą być przydatne dla twórcy motywów lub programisty, lecz nie powinny być używane bezpośrednio jako wyświetlane dane wyjściowe.

Na potrzeby prezentowania danych użyj wartości zawartych w elemencie `view` lub skorzystaj z funkcji `content_format()`, aby zapewnić, że dane przekazywane do wyświetlenia będą czyste.

Ćwiczenie 4.3.

Zastosowanie motywu dla grupy pól

Czasem istnieje grupa pól, którą chcesz uwidocznić w typie zawartości lub wręcz przeciwnie. Pakiet CCK oferuje nazwy grup jako część danych wyjściowych wyświetlanych ostatecznie w przeglądarce. W przykładzie w typie zawartości *Batch brewed* (partia warzonego piwa) użyto grupy pól *Alcohol level* (poziom alkoholu).

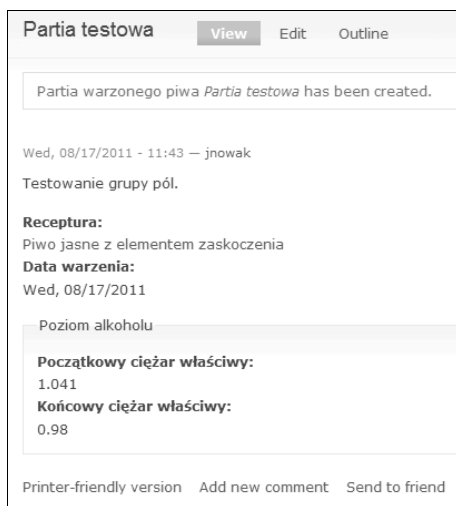
1. Określ nazwę grupy, która została przekazana jako nazwa klasy. Dowolny segment obejmujący tę zawartość udostępni nazwę grupy w miejscu, w którym z łatwością możesz ją znaleźć.
2. Wyświetl kod źródłowy strony (rysunek 4.3).
3. W kodzie źródłowym poszukaj łańcucha `fieldgroup`. Możesz znaleźć wiersz podobny do następującego:

```
<fieldset class="fieldgroup group-alcohol-level">
```

Ta klasa umożliwia zastosowanie motywu dla wszystkich grup pól lub tylko dla grupy `group-alcohol-level`.

4. Otwórz plik `garland_extend.css` i dodaj następujący kod:

```
fieldset.group-alcohol-level {  
  font-size: x-small;  
}
```



Rysunek 4.3. Stan przed dokonaniem zmiany w kodzie CSS

5. Zapisz plik CSS i odśwież stronę w przeglądarce (rysunek 4.4).



Rysunek 4.4. Stan po zmodyfikowaniu grupy pól

Użycie grupy pól jako klasy umożliwia zmianę tylko jednego zestawu pól zamiast wpływania na cały segment. Pozwala to również na jednoczesną zmianę wszystkich tych pól zamiast stosowania motywu osobno dla każdego pola.

Dodawanie plików do motywów

System Drupal buforuje listę plików dostępnych dla danego motywu. Jeśli utworzysz nowy plik szablonu, będziesz musiał ponownie zbudować rejestr motywów, aby mechanizm motywów znalazł nowe pliki.

W celu ponownego zbudowania rejestru motywów wykonaj jedną z następujących czynności:

- ♦ Przejdź do pozycji *Administer/ Site configuration/ Performance* (administruj/ konfiguracja witryny/wydajność). Przewiń do dołu strony i kliknij przycisk o nazwie *Clear cached data* (wyczyść buforowane dane).
- ♦ Użyj odnośnika *Clear cache* (wyczyść bufor) modułu *Devel*.

Wykluczanie pól

Gdy dowolny element treści jest w całości wyświetlany w witrynie opartej na systemie Drupal, dostarczany jest przede wszystkim przez zmienną `$content`. Ta zmienna zawiera cały renderowany kod HTML, który jest prezentowany w głównej części strony. W większości przypadków nie stanowi to problemu. Ponieważ motyw będzie często stosowany dla witryny w spójny sposób, wskazane jest wyświetlanie wszystkich dodatków i kontrolki w ramach jednej zmiennej. W zmiennej mogą być uwzględnione takie elementy jak na przykład dołączone obrazy i kontrolki do głosowania.

Sporadycznie takie rozwiązanie nie jest pożądane. Na przykład możesz oczekiwać, że niektóre elementy treści zostaną wyświetlone niezależnie do reszty. Aby zapobiec dołączeniu pól do zmiennej `$content`, zaznacz pole *Exclude* (wyklucz) na karcie *Display fields* (wyświetl pola) typu zawartości. Przejdź do pozycji *Administer/ Content management/ Content types/ (wybierz typ zawartości)/ Display fields* (administruj/ zarządzanie treścią/ typy zawartości/ (wybierz typ zawartości)/ wyświetl pola). Na tej karcie możesz wykluczyć elementy ze skrótu artykułu lub całego segmentu. Zauważ, że istnieje różnica między opcjami *Hidden* (ukryte) i *Exclude* (wyklucz). Opcja *Hidden* (ukryte) uwzględni dane, lecz nie wyświetli ich na stronie. Dla porównania opcja *Exclude* (wyklucz) całkowicie zapobiega dodawaniu danych do zmiennej. Ponownie przyjrzyj się rysunkowi 2.11, aby zlokalizować przycisk *Exclude* (wyklucz). Jeśli na karcie *Display fields* (wyświetl pola) wybierzesz pole, menu rozwijane opcji *Teaser* (skrót artykułu) i *Full node* (cały segment) umożliwi wybranie opcji *Hidden* (ukryte) w celu ukrycia pola.

Na tym etapie możesz dostosować plik szablonu, używając zmiennych `<NAZWA_GRUPY>_rendered` lub `<NAZWA_POLA>_rendered` w instrukcji `print` języka PHP w celu umieszczenia ich w dowolnym żądanym miejscu strony.

Bądź świadomy tego, że zastosowanie pól wyboru ustawienia *Exclude* (wyklucz) spowoduje wykluczenie danych wybranego pola ze wszystkich motywów. Jednakże każdy motyw może zawierać inne informacje w zmiennej `$content`, dlatego rzeczywiście wyświetlane pola mogą być różne dla poszczególnych motywów.

Node Reference

Z typem Node Reference (odwołanie do segmentu) związanych jest kilka specjalnych zmiennych i innych elementów, które mogą spowodować jeszcze więcej zmian w motywie. Pola typu Node Reference używają wszystkich wcześniej omówionych funkcji motywów, ale też uwzględniają kilka specjalnych szablonów, które mogą być zastosowane, gdy pola tego typu służą do wyświetlenia segmentów w przypadku korzystania z elementów formatujących skrót artykułu lub cały segment. Te szablony sprawdzają typ przywoływanego segmentu i stosują dla niego odpowiedni motyw. Sugestie zawarte w następującym pliku mają pierwszeństwo względem innych:

```
node-nodereference-<NAZWA_POLA_ODWOŁUJĄCEGO>-  
<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php
```

Jest to najbardziej szczegółowy szablon sugestii. Szablon szuka pola odwołującego się, a także typu przywoływanego segmentu. W poprzednich ćwiczeniach stworzyliśmy pole typu Node Reference dla typu zawartości *Batch brewed* (partia warzonego piwa). Celem tego pola było odwołanie się do receptury użytej do przygotowania konkretnej partii piwa. W zasadzie nie było wymagane, aby użytkownik określił w tym polu receptury, może ono odwołać się do dowolnego innego pola. Jeśli pole ma zostać użyte w celu odwołania do receptury, powinieneś skorzystać z pliku szablonu o następującej nazwie:

```
node-nodereference-field_recipe_used-beer_recipe.tpl.php
```

Jeżeli to pole odwołujące się zostało zastosowane do odwołania do artykułu, w nazwie pliku szablonu zamiast łańcucha *-beer_recipe.tpl.php* wystąpi łańcuch *-story.tpl.php*.

```
node-nodereference-<NAZWA_TYPU_ZAWARTOŚCI>.tpl.php
```

Wykorzystanie tego szablonu spowodowałoby zastosowanie motywu dla wszystkich pól typu Node Reference, w przypadku których przywoływane pole jest w określonym typie zawartości. Jeśli szablonowi nadano na przykład nazwę *node-nodereference-t_shirts.tpl.php*, może być użyty do zastosowania motywu dla dowolnego segmentu *t_shirts*, gdy odwołuje się do niego pole typu Node Reference.

```
node-nodereference-<NAZWA_POLA_ODWOŁUJĄCEGO>.tpl.php
```

Gdziekolwiek zostanie użyte pole odwołujące się, spowoduje, że szablon ten zostanie wykorzystany do zastosowania motywu dla przywoływanego segmentu.

```
node-nodereference.tpl.php
```

Ten szablon stosuje motyw dla dowolnego segmentu, który jest przywoływany przez pole typu Node Reference.

Używając przedstawionych szablonów, możesz utworzyć układ kolekcji jesiennej dla typu zawartości *Joe's Shirts* (koszulki Janka) i zastosować pole typu Node Reference w celu odwołania się do każdego segmentu koszulki. Użytkownicy przeszukujący cały katalog mogą następnie z łatwością stwierdzić, które produkty są nowe w kolekcji jesiennej,

jeśli dla segmentów przywoływanych zostanie zastosowany motyw różniący się trochę od standardowego.

Pakiet CCK oferuje jeszcze dwie zmienne w celu użycia ich w plikach szablonów w przypadku korzystania z odwołań do segmentów.

`$referring_field`

Ta zmienna wybiera pole, które odwołuje się do bieżącego segmentu. Potencjalnie zmienna może posłużyć do śledzenia motywu aż do jego źródła.

`$referring_node`

Ta zmienna przechowuje segment, który zawiera pola odwołujące się bieżącego segmentu.

Moduły pomocnicze

Podczas pracy z motywami systemu Drupal nie ma potrzeby określania wszystkiego we własnym zakresie. Dostępne są różne moduły pomocnicze, które znacznie ułatwiają stosowanie motywów systemu Drupal. W tym podrozdziale opisaliśmy kilka naszych ulubionych modułów.

Theme Developer

Theme Developer (http://drupal.org/project/devel_themer) nie jest modulem służącym wyłącznie do tworzenia motywów. Jednakże jest jednym z najbardziej wartościowych narzędzi udostępnionych w celu ułatwienia stwierdzenia, które szablony wpływają na dowolny wybrany element treści. Aby zainstalować ten moduł, konieczny będzie również moduł *Devel* (<http://drupal.org/project/devel>), który także zapewnia wiele przydatnych funkcji.

Po zainstalowaniu i włączeniu modułu *Theme Developer* udostępnia pole wyboru w dolnym lewym narożniku okna przeglądarki o nazwie *Themer info* (informacje modułu *Themer*). Po włączeniu opcja powoduje wyłączenie możliwości śledzenia jakichkolwiek odnośników na konkretnej stronie. Jeśli klikniesz w danej sekcji strony, pojawi się okno wyszczególniające dokładnie, jakie pliki *.tpl* wpływają na ten element treści strony. Udostępniana jest pełna ścieżka ważności szablonów uporządkowanych w kolejności od najbardziej do najmniej szczegółowych. Lista obejmuje wszystkie kandydujące pliki szablonu, które mogą mieć wpływ na treść, nawet jeśli te pliki nie istnieją w systemie. Skorzystaj z tych informacji, gdy będziesz musiał zdecydować, czy powinieneś zmienić ogólny motyw, bądź w sytuacji, kiedy chcesz nadpisać motyw na tak szczegółowym poziomie jak konkretne pole jednego typu zawartości. Dysponując taką wiedzą, znacznie łatwiej będziesz mógł określić, jaki plik szablonu powinien zostać zmieniony.

Dzięki temu, że lista plików szablonu umożliwia klikanie, o wiele łatwiejsze staje się otwarcie pliku szablonu. Zamiast przeszukiwać cały system w celu znalezienia pliku możesz go szybko zlokalizować na przejrzystej liście. Ta kwestia jest szczególnie istotna, gdy pliki szablonu mają identyczną nazwę. Na przykład istnieje domyślny plik *content-field.tpl.php*,

lecz pakiet CCK uwzględnia również szablon o tej samej nazwie, który będzie domyślnie używany. Pole wyboru *Themer info* (informacje modułu Themer) jednoznacznie identyfikuje plik, który jest używany w celu zastosowania motywu dla każdego elementu treści.

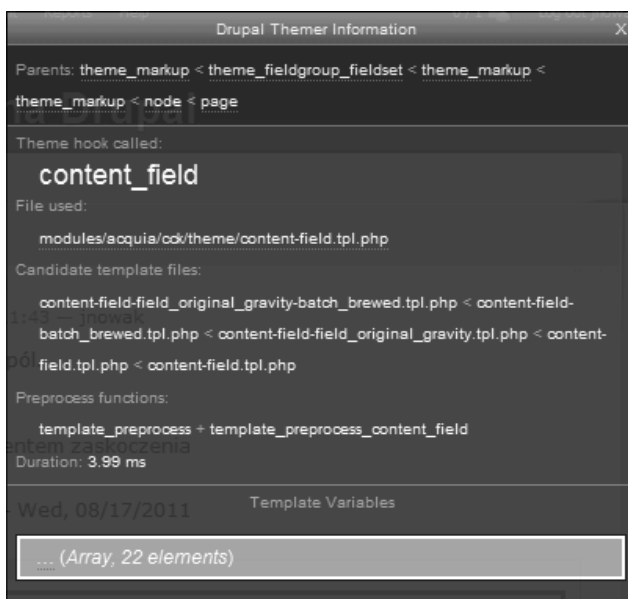
Ćwiczenie 4.4.

Wyświetlanie informacji o motywie za pomocą modułu Themer

W tym ćwiczeniu pokażemy, jak pomocne może się okazać nawet niewielkie wsparcie ze strony modułu *Themer*, którego zastosowanie ułatwia stwierdzenie, jakie mogą być pliki szablonu.

1. Zainstaluj moduły *Themer* i *Devel*.
2. Przejdź do pozycji *Administer/Site building/Modules* (administruj/budowanie witryny/moduły) i włącz moduły *Devel* i *Theme Developer*.
3. Otwórz stronę główną witryny.
4. Kliknij pole wyboru *Themer info* (informacje modułu Themer) w dolnym lewym narożniku okna.
5. Kliknij dowolny komponent na stronie.

Jeśli we wcześniejszym ćwiczeniu utworzyłeś pole *T-shirt size* (rozmiar koszulki), kliknij sekcję *T-shirt size* (rozmiar koszulki). Powinno się pojawić okno podobne do pokazanego na rysunku 4.5.



Rysunek 4.5. Nakładka informacyjna modułu Theme Developer

Pole wyboru *Themer info* (informacje modułu Themer) powoduje też wyszczególnienie elementów nadrzędnych elementu, dla którego zastosowano motyw. Sekcja elementów nadrzędnych również umożliwia klikanie. Uaktywnienie sekcji powoduje wyróżnienie grupy zawartości, na którą sekcja ma wpływ. Tego rodzaju informacja zwrotna ma kluczowe znaczenie, gdy próbujesz zidentyfikować problemy dotyczące nadpisywania występującego między komponentami, dla których zastosowano motyw.

Oto ostrzeżenie: moduł *Themer* powinien być włączony tylko w witrynach testowych lub wtedy, gdy jest on wyraźnie niezbędny do zadziałania motywu. Jeśli włączono ten moduł, podczas wykonywania określonych zadań w widokach może być zauważalne nieoczekiwane zachowanie. Może to powodować niejasności, gdy wykonujesz czynności niemające związku ze stosowaniem motywu! Podobnie do wielu znakomitych narzędzi rozwiązujących problemy ten moduł powinien być włączony, gdy aktywnie z niego korzystasz, natomiast wyłączony, jeśli go nie używasz.

W celu utworzenia bardziej szczegółowego pliku możesz użyć plików wyszczególnionych przez moduł *Devel*. Okazuje się, że gdy poznasz konwencję nazewnictwa, z łatwością będziesz mógł określić poprawny plik, aby zmienić konkretne pole. Jeżeli przeglądasz listę priorytetów modułu *Devel*, ujrzysz dwa pliki *content-field.tpl.php*. W systemie istnieją dwa takie pliki: jeden to ogólny plik systemu Drupal, a drugi to domyślny plik pakietu CCK. Ponieważ pola tego pakietu używają jego szablonu, w porównaniu z podstawową instalacją systemu Drupal uzyskujesz już o jeden poziom większą szczegółowość.

Contemplate

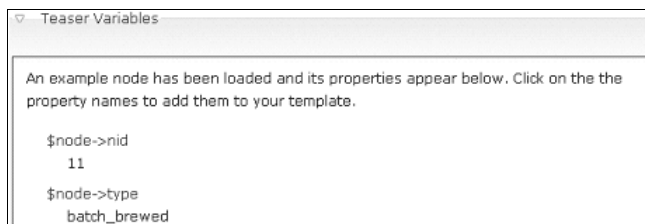
Contemplate, znany również jako *Content Templates* (<http://drupal.org/project/contemplate>), to moduł, który stanowi centrum bardziej filozoficznej debaty trwającej w społeczności związanej z systemem Drupal. Wraz z jego rozwojem poczyniono większe starania na drodze do oddzielenia treści od warstwy prezentacji. Wiele osób jest przekonanych, że ta dychotomia to dobry pomysł, ponieważ pozwala programistom i twórcom motywów na pracę absolutnie bez konieczności orientowania się, czym się zajmują inni. Motyw jest niezależną treścią, która nie musi być modyfikowana, gdy w informacjach dotyczących całej witryny są wprowadzane mniej istotne aktualizacje. Taka elastyczność zmniejsza koszty ogólne i związane z konserwacją, co jest wskazane.

Moduł *Contemplate* reprezentuje coś pośredniego między motywem i treścią. Ten moduł zapewnia widoczność zawartości motywu przez umożliwienie tworzenia danych szablonu za pośrednictwem stron pakietu CCK, co jest zaletą. Oznacza to jednak również, że nie korzystasz z „poprawnej” metody stosowania zwykłego pliku szablonu, co przez niektórych jest uznawane za coś złego. Dla nowego użytkownika widoczność może być kluczem do zrozumienia. Moduł *Contemplate* oferuje informacje dotyczące stosowania motywów tylko dla skrótów artykułów, pola *Body* (cała treść) i kanałów RSS. Stanowi to absolutne minimum elementów, dla których możesz zamierzać użyć motywu.

Ten moduł dodaje kilka nowych odnośników do interfejsu administratora: *Administer/Content management/Content templates* (administruj/zarządzanie treścią/szablony zawartości) lub *Administer/Content management/Content types/Templates* (administruj/zarządzanie treścią/typy zawartości/szablony). Każda z tych ścieżek skieruje na tę samą stronę, która umożliwia utworzenie nowego szablonu.

Szablon udostępnia trzy grupy pól — po jednej dla skrótów artykułów, pola *Body* i kanałów RSS. Każda grupa zapewnia pole wyboru, za pomocą którego możesz określić, czy dana część szablonu będzie używana. Jeśli typ zawartości nie uwzględnia skrótu artykułu lub kanału RSS, nie ma sensu tworzyć szablonów dla tych sekcji.

Najważniejszym aspektem związanym z tymi sekcjami jest lista zmiennych. Lista prezentuje wszystko, co jest dostępne dla funkcji stosowania motywów z poziomu samego segmentu. Na rysunku 4.6 przedstawiono przykład takiej listy w wersji skróconej.



Rysunek 4.6. Częściowa lista pozycji dostępnych dla skrótu artykułu

Inną ważną funkcją realizowaną przez moduł *Contemplate* jest identyfikowanie pól, które mogą nie być bezpieczne i są potencjalnie podatne na atak. Te pola są oznaczane dwoma gwiazdkami (**). System Drupal zapewnia dwie funkcje (`check_plain()` i `check_markup()`) do oczyszczania danych, aby użytkownicy nieumyślnie lub celowo nie spowodowali problemów, korzystając z danych, których nie oczyszczono. Te funkcje można znaleźć w preprocesorze motywu. Jeśli używasz modułu *Contemplate*, automatycznie doda on funkcję `check_plain()` do typów uznanych za takie, które powinny z niej korzystać.

Podsumowanie

W rozdziale poznałeś system motywów Drupala, a także wiele rodzajów interakcji między nim i pakietem CCK. Dzięki użyciu zestawu plików szablonu, modułów pomocniczych i opcji wyświetlania interfejsu użytkownika rdzenia powinno być teraz możliwe określanie przeznaczenia, dostosowywanie i wyświetlanie dowolnego pola zawartości prezentowanej w witrynie.

Skorowidz

A

abstrakcja, 224
alias \$table_\$field, 237
API (Application Programming Interface), 34
argumenty, 172, 179
argumenty w module Panels, 311
arkusze stylów, 91, 213, 332, 334
 panels.css, 334
 panels_admin.css, 334
 panels_dashboard.css, 334
 panels_dnd.css, 334
 panels_page.css, 334
automatyczne uzupełnianie, 107

B

Bazaar, 340
blok, 166
buforowanie, 289
buforowanie anonimowych stron, 220
buforowanie bloków, 221

C

CMS (Content Management System), 132
Content types (typy zawartości), 38
CVS, 340
cykl istnienia widoku, 240
cykl wykonywania widoku, 240

D

debugowanie, 205
definicja pola, 133
deklarowanie pól tabel, 250

deklarowanie tabel, 247
destruktor, 228
dodatek
 domyślne ustawienia argumentu, 231, 253
 dostęp, 231, 253
 format wyświetlania, 231, 253
 sprawdzania poprawności argumentu,
 231, 253
 styl, 231, 253
 styl wiersza, 231, 253
dodawanie
 argumentu, 175
 kontekstu, 305
 kontroli dostępu, 291
 sekcji zawartości, 284
 terminu taksonomii do sekcji, 309
 widoku do sekcji, 312
 zawartości, 287
 zawartości do paneli, 284
dokumentacja, 343
domyślny format wyświetlania, 195
Drupal, 36
dziedziczenie, 227

E

edytowanie
 kontekstu minipanelu, 305
 kontekstu stron paneli, 304
 minipanelu, 283
 segmentu panelu, 282
eksportowanie
 modułów CCK, 343
 paneli, 346
 pól modułów CCK, 344

- eksportowanie
 - struktur, 343
 - widoków, 345
- elementy
 - formatujące, 106, 107
 - funkcjonalne modułu Panels, 262
 - nadrzędne, 319
 - segmentu, 31
- enkapsulacja, 224

F

- filtr
 - Basic, 75
 - Displays, 142
 - Full HTML, 75
 - Node: Published, 156
 - Storage, 142
 - Tag, 142
 - Type, 142
- filtrowanie, 136
- filtry ujawniane, 154, 181, 182
- flaga aggregate, 237
- flaga boolowska distinct, 237
- flaga count, 237
- format wyświetlania bloku, 169
- format wyświetlania domyślny, 148
- funkcja
 - check_markup(), 104
 - check_plain(), 104
 - content_clear_type_cache(), 127
 - content_field(), 113
 - content_field_instance_create(), 121
 - content_field_instance_read(), 123
 - content_format(), 127
 - content_write_record(), 127
 - CRUD, 121
 - db_rewrite_sql(), 243
 - drupal_execute(), 125
 - drupal_render(), 117
 - drupal_set_message(), 205
 - drupal_write_record(), 127
 - dsm(), 205
 - handler->construct(), 251
 - handler->init(), 251

- handler->pre_query(), 251
- handler->pre_render(), 251
- handler->query(), 251
- handler->render(), 252
- handler->set_definition(), 251
- node_content_build_modes(), 128
- node_load(), 125
- node_save(), 125
- search_content_build_modes(), 128
- strtotime, 78
- theme_views_view_field, 202
- views_get_view(), 240
- funkcje pomocnicze, 127

G

- Git, 340
- grupowanie pól, 208
- grupowanie w szablonie, 208

H

- hak, 239
 - hook_block(), 243
 - hook_content_generate(), 114
 - hook_content_is_empty(), 109
 - hook_content_notify(), 108
 - hook_disable(), 108
 - hook_elements(), 115
 - hook_enable(), 108
 - hook_field(), 109, 113
 - hook_field_info(), 109
 - hook_field_settings(), 109
 - hook_form_alter(), 108
 - hook_install(), 108
 - hook_uninstall(), 108
 - hook_views_admin_links_alter(), 246
 - hook_views_data(), 246, 247
 - hook_views_data_alter(), 248
 - hook_views_handlers(), 233, 252
 - hook_views_plugins(), 233
 - hook_views_pre_build(), 241, 245
 - hook_views_pre_execute(), 243, 245
 - hook_views_pre_render(), 243, 246
 - hook_views_pre_view(), 245

- hook_views_query_alter (), 245
- hook_views_query_substitutions(), 250
- hook_widget(), 115
- hook_widget_info(), 115
- hook_widget_settings(), 115
- haki, 105, 223
- haki kontrolek, 108
- haki pól modułów CCK, 108

I

- identyfikacja typu zawartości, 45
- identyfikator pola, 195
- identyfikator segmentu, 30
- identyfikator systemowy, 50
- indeks, 134
- indeksowanie, 221
- instancja pola, 65
- instrukcja
 - DESCRIBE, 218
 - EXPLAIN, 218, 220
 - JOIN, 64, 138, 220, 235
 - ORDER BY, 138
 - SELECT, 138, 219
 - WHERE, 136, 138
- interfejs API, 105, 123, 223
- interfejs API modułu Views, 239
- interfejs API REST, 232
- interfejs użytkownika modułu Panels, 269, 273
- interfejs użytkownika modułu Views, 139, 218

J

- język obiektowy, 226

K

- kanal RSS, 169
- kanały użytkownika z argumentami, 177
- karta
 - Basic, 158
 - Bulk export, 160, 346
 - Contexts, 304
 - Convert, 161
 - Dashboard, 270
 - Display fields, 51, 99, 107

- Edit, 282
- Export, 50
- Fields, 50
- Import, 50
- Manage fields, 51, 61
- Node, 305
- Panel content, 282
- Panel layout, 282
- RSS, 64
- Search, 64
- Settings, 271
- View, 282
- katalog
 - /cck/theme, 90
 - /sites/all/themes, 90
 - /themes/garland, 90
 - /views/theme, 200
- klasa
 - attachment-after, 191
 - attachment-before, 191
 - feed-icon, 191
 - parent, 252
 - view, 190
 - view-content, 191
 - view-empty, 191
 - view-filters, 191
 - view-footer, 191
 - view-header, 191
 - views_object, 233
- klasy modułu Views, 190
- klauszula
 - GROUP BY, 237
 - HAVING, 237
 - ORDER BY, 219
 - WHERE, 219
- klonowanie widoku, 165
- klucz do tablicy, 252
- klucz obcy, 135
- klucz podstawowy, 134
 - nid, 134
 - tid, 134
 - uid, 134
 - vid, 134
- kolejność argumentów, 180
- konfigurowanie argumentu, 173

- konfigurowanie istniejących sekcji
 - zawartości, 289
- konfigurowanie środowiska wdrażania, 339
- konstruktor, 228
- kontekst, 263, 313
 - node, 304
 - node_add_form, 304, 308
 - node_edit_form, 304
 - string, 304
 - term, 304
 - terms, 304
 - user, 304
 - vocabulary, 304
- konteksty, 298, 303
- kontrola źródeł, 340
- kontrolka, 57, 72, 106
 - Extensible File, 85
 - File Upload, 85
 - Image, 85
 - Optionwidgets, 106
 - Textfield, 106
- kontrolki opcji, 107
- konwencja nazewnictwa używana
 - przez moduł Views, 232

Ł

- łącza segmentu, 30

M

- menedżer układu, 324
- menu
 - Canvas, 325
 - Column, 325
 - Region, 326
 - Row, 325
- menu administracyjne, 269
- metoda
 - _content_field_invoke(), 108
 - construct, 234
 - construct(), 241
 - db_query(), 238
 - destroy(), 234, 243
 - ensure_table(), 236

- init(), 241
- pre_execute(), 241
- pre_query(), 242
- query(), 242
- set_arguments(), 241
- view
 - execute_display(), 243
 - preview(), 244
 - get_title(), 244
 - get_breadcrumb(), 244
 - override_path, 244
- minipanele, 282
- moduł
 - Advanced Help, 140
 - Book, 62
 - Bulk Export, 266
 - CCK, 33, 36, 78
 - CCK Blocks, 41
 - CCK Formatters, 41
 - Chaos Tools, 266
 - Computed Field, 80
 - Contemplate, 89, 103
 - Content, 40, 114
 - Content Construction Kit, 320
 - Content Copy, 40, 120, 123
 - Content Permissions, 40
 - CTools, 304
 - Custom content panes, 266
 - Custom rulesets, 266
 - Date, 41, 54, 77, 78
 - Deploy, 347
 - Devel, 93, 102, 159
 - Devel Generate, 114
 - Drush, 348
 - Email, 84
 - Features, 348
 - Fieldgroup, 40, 62
 - FileField, 85
 - Fivestar, 28, 183
 - Flexinode, 34
 - Image, 85
 - ImageAPI, 87
 - ImageCache, 87
 - ImageCache UI, 87
 - ImageField, 86

- Inline, 28
- Invisimail, 85
- Link, 84
- Mini panels, 265
- Number, 41
- Page manager, 266, 267, 291
- Panel nodes, 265
- Panels, 2, 36, 259–266, 303, 323, 332
- Panels In-Place Editor, 266
- Pathauto, 298
- Plus1, 183
- Print, 62
- Range, 41
- Search, 62
- Stylizer, 266
- Taxonomy, 308
- Theme Developer, 101, 102
- Themer, 102
- Upload, 28, 230
- Views, 35, 36, 57, 129–139, 157, 164, 189, 230
- Views content panes, 266
- moduły
 - elementów formatujących, 117
 - formatów wyświetlania, 265
 - pomocnicze, 265
 - typów kontrolek, 115
 - typów pól, 109
- motyw
 - Garland, 89
 - Minelli, 89
- motyw podstawowy, 90
- motywy, 89, 336
- motywy modułu Views, 189
- MySQL, 131

N

- nadpisanie, 156
- nadpisanie adresu URL dla widoku, 316
- nadpisanie taksonomii, 299
- nadpisywanie stron, 293
- nadpisywanie stron edycji segmentu, 301
- narzędzie Convert, 161
- narzędzie Stylizer, 328

O

- obiekt
 - \$join, 236
 - dodatki, 230
 - łączenia, 230
 - procedury obsługi, 230
 - widoku, 230
 - zapytania, 230, 234
- obiekty równorzędne, 224
- obszar tekstowy, 107
- odnośnik Access, 292
- odnośnik Go to list, 271
- odwołanie do segmentu, 107
- odwołanie do użytkownika, 107
- okna konfiguracyjne, 283
- okno
 - Add content, 284, 286, 295, 313
 - Activity, 286
 - Existing node, 286
 - Menus, 286
 - Miscellaneous, 286
 - New custom content, 286
 - Page elements, 286
 - Views, 286
 - Widgets, 286
 - Content, 283
 - Context, 283
 - Layout, 283
 - Panel content, 284
 - Display settings, 284
 - Panel Regions, 285
 - Title Type, 285
 - Update/Save, 285
 - Settings, 283
- okno edycji strony panelu, 278
- okno edycji zawartości, 284
- okno modalne, 313
- opcja
 - Add variant, 274
 - Clone, 273
 - Create new revision, 48
 - Delete, 273
 - Disable, 273
 - Exclude, 99

opcja

- Export, 273
- Hidden, 99
- Import variant, 274
- New custom content, 302
- Node, 199
- nofollow, 84
- Num posts, 314, 315
- Offset, 314
- Promoted to front page, 48
- Published, 48
- Sticky at top of lists, 48
- Summary, 274
- Title, 174
- Use pager, 313

opcja pola, 58

optymalizacja zapytań, 216

osadzanie widoku, 231

osadzanie zapytań, 217

P

pager systemu Drupal, 314

pakiet CCK, 33, 41

pakiet Content Construction Kit, 35

pakiet CTools (Chaos Tool Suite), 37, 266, 293

plik

- .info, 91
- .module, 252
- classname.inc, 253
- content-field.tpl.php, 93
- garland-extend.css, 191
- panels.css, 334
- text.module, 120
- theme.inc, 203
- views-view.tpl.php, 195, 203

plik szablonu, 89, 93, 99, 192

podgląd dynamiczny, 157

podłączanie mechanizmu buforowania, 264

pola obliczeniowe, 80

pola wyboru, 107

pole, 43

- Allowed values list, 60
- Argumenty, 172
- Authoring information, 281

Body, 57, 61, 75, 206

Body field label, 46

Book outline, 281

Book Page, 65

Comment settings, 281

Computed, 80

Computed Code, 81

CSS ID, 280

Customize Default Value, 78

Data length, 81

Description, 45

Exclude, 99

File attachments, 281

Label, 51

Menu settings, 280

New field, 53

New group, 53

Node Reference, 63

Nodereference, 41

Not NULL, 81

Number, 106

Publishing options, 281

Revision information, 281

Teaser, 279

Text, 41

Text processing, 54

Time zone handling, 79

Title, 61, 279

Title field label, 46

Type, 45, 51

URL path settings, 281

Userreference, 41

Wildcard title, 174

pole tekstowe CSS code, 332

pole typu, 75

Date, 77

Datestamp, 77

Datetime, 77

Decimal, 68

Float, 69

Integer, 69

Node Reference, 69, 100

Text, 68

User Reference, 71

pole wyboru

- Add Views signature to all SQL queries, 158
- Allow multiple arguments to work together, 175
- Disable Views data caching, 158
- Do not display items with no value in summary, 175
- Enable views performance statistics via the Devel module, 216
- Force single, 182
- Limit list to selected items, 182
- Optional, 181
- Reduce duplicates, 175
- Remember, 182
- Send arguments, 316
- Show other queries run during render during live preview, 159
- Show query above live preview, 159
- Themer info, 102
- Unlock operator, 181

polimorfizm, 227

PostgreSQL, 131

procedura obsługi

- views_handler_filter_in_operator, 252

procedury obsługi, 251

programowanie obiektowe, 224

przesłanianie, 227

przycisk Rescan template files, 193, 203

przycisk Expose, 181

R

rdzeń systemu Drupal, 223

region, 336

reguły dostępu, 291

rekord, 133

relacja, 135

renderowanie strony rdzenia systemu

- Drupal, 261

rozszerzanie obiektów, 223

S

segment, 27, 333

segmenty paneli, 279

sekcja, 335

- Actions to take if argument not present, 174

Add, 51

Additional Display Settings, 79

Allowed Block content, 272

Arguments, 155

Basic settings, 149, 185

Comment settings, 48, 49

Configure, 51

Customize date parts, 78

Describing tables to Views, 249

Fields, 155

Filters, 156

Granularity, 78

Identification, 44

New content behavior, 272

Relationships, 154

Remove, 51

Sort Criteria, 155

Submission form settings, 46

To date, 78

View Settings, 149

Workflow Settings, 47

selektor CSS, 333

skrót artykułu, 31

słownik, 308

sortowanie, 136

sprawdzenie zawartości szablonu, 204

SQL, 131

standard ANSI SQL, 238

stos XAMPP, 36

strona

- Add, 143

Add Content Type, 44

Basic, 62

Blocks, 187

Display fields, 62

Edit, 158

Getting Started, 140

Import, 157

List, 140, 141, 273

Manage fields, 61

Mini panels, 271

Modules, 265

Panel nodes, 271

strona

Panel pages, 271, 293

RSS, 62

Tools, 158

strony nadpisanie taksonomii, 298

strony paneli, 271

strumień wiadomości, 31

styl widoku, 197

Subversion, 340

sugestia, 92

szablon

views-exposed-form.tpl.php, 200

views-more.tpl.php, 201

views-ui-*.tpl.php, 200

views-view.tpl.php, 196

views-view-field.tpl.php, 202

views-view-fields.tpl.php, 199

views-view-grid.tpl.php, 197

views-view-list.tpl.php, 197

views-view-row-comment.tpl.php, 202

views-view-row-node.tpl.php, 202

views-view-row-rss.tpl.php, 200

views-view-rss.tpl.php, 197

views-view-summary.tpl.php, 202

views-view-summary-

-unformatted.tpl.php, 201

views-view-table.tpl.php, 198

views-view-unformatted.tpl.php, 198

szablon domyślny, 194

szablon domyślny node.tpl.php, 96

szablon szczegółowy, 203

szablony, 192

modułu Views, 195

pól, 92

segmentów, 96

wierszy, 198

Ś

środowisko produkcyjne, 341

środowisko projektowe, 342

T

tabela, 133

bar, 248

baz, 249

foo, 248

node, 32

node_revisions, 32

users_roles, 247

tabela bazowa, 222, 229

tablica

\$definition, 233

\$field, 121

\$info, 121

\$options, 241

taksonomia, 298

technologia AJAX, 151

termin, 308

testowanie zmian, 342

tworzenie

danych dla pól modułów, 124

elementów formatujących, 107

formatów wyświetlania, 164

formatu wyświetlania, 186

instancji pól, 120

minipanelu, 283

nowego stylu, 330

nowego typu zawartości, 37

nowego widoku, 147

nowych pól typu zawartości, 50

nowych typów pól, 107

pola, 59

pola typu Node Reference, 70

segmentu panelu, 279

strony nadpisanie segmentu, 296

strony panelu, 276

typów kontrolek, 107

widoku, 161

typ danych, 53

Decimal, 53

Float, 53

Integer, 53

Node Reference, 53

Text, 53

User Reference, 53

typ formatu wyświetlania, 335

Attachment, 148

Block, 148

Feed, 148

Page, 148

typ kontrolki, 76, 107

typ pola, 43, 67

typ widoków, 229

Access log, 146

Comment, 146

File, 146

Node, 146

Node revision, 146

Term, 146

User, 146

typ zawartości, 321, 344

Blog, 38

Blog Entry, 29

Book Page, 29

Comment settings, 44

Forum, 30

Identification, 44

Node Reference, 320

Page, 29

Poll, 30

Receptura, 38

Story, 29, 38

Submission form settings, 44

User Reference, 320

U

układ, 323

układ Flexible, 323, 324

kontrolka Canvas, 325

kontrolka Column, 325

kontrolka Region, 325

kontrolka Row, 325

uprawnienie User

role, 302

ustawienia

globalne, 56

globalne pola, 73

instancji pola, 43

modułu Panels, 271

paneli, 272

pola dla typu segmentu, 73

sekcji w module Views, 315

ścieżki, 176

typu zawartości, 55

ustawienie

Access, 152, 275

Admin, 154

Allow settings, 316

Allowed values list, 56

Argument input, 315

Arguments, 316

Basic, 274

Basic settings, 153, 193

Caching, 152, 290

Content, 275

CONTENT_CALLBACK_CUSTOM, 115

CONTENT_CALLBACK_DEFAULT, 115

CONTENT_CALLBACK_NONE, 115

CONTENT_HANDLE_CORE, 115, 118

CONTENT_HANDLE_MODULE, 115, 118

Contexts, 275

CSS properties, 290

Default display, 79

Disable this pane, 290

Distinct, 152

Empty text, 153

Footer, 152

General, 275

Header, 152

Items per page, 152

Layout, 275

Link to view, 315

Maximum, 76

Menu, 153, 275

More link, 152

No cache, 289

Number of values, 56, 78

Path, 153

Preview, 275

Required, 56, 78

Row style, 170, 199

Search display field, 62

Selection rules, 275

ustawienie

- Settings, 290
- Style, 197, 290
- Summary, 275
- Theme, 153
 - Information, 193, 199
- type, 121
- Use AJAX, 151
- Use panel path, 315
- Visibility rules, 290
- widget_type, 121

użycie

- argumentu w sekcji z widokiem, 316
- formatu wyświetlania, 318
- suwaka, 326
- zależności, 184, 319

W

wartość

- database columns, 111
- delete, 112
- delete revision, 112
- form, 111, 116
- insert, 112
- load, 112
- prepare translation, 112
- presave, 112
- sanitize, 112
- save, 111, 116
- update, 112
- validate, 111, 112, 116
- views data, 111

widok

- Default, 142
- Normal, 142
- Overridden, 142

widok domyślny, 144

widok segmentu, 294

widok słownikowy, 177

widok użytkownika, 300

wiersz, 133

współużytkowanie pola, 64

wydajność zapytania, 216

wykluczanie pól, 99

wywołanie #process, 115

Z

zadania lokalne, 31

zależność, 183, 319

jawna, 230, 247

niejawna, 229, 246

Upload:Attached files, 230

załącznik, 169

zapisywanie widoków, 176

zapytania modułu Views, 216

zapytania SQL, 138

zastosowanie motywu dla grupy pól, 97

zastosowanie stylów, 331

złączenia, 137

złączenie lewe, 137

złączenie modułu Views, 229

złączenie wewnętrzne, 137

zmiana układu, 327

zmienna

\$<NAZWA_GRUPY>_rendered, 96

\$<NAZWA_POLA>_rendered, 96

\$admin_links, 196

\$admin_links_raw, 196

\$alias, 238

\$button, 201

\$class, 198

\$content, 99

\$css_name, 196

\$element, 113

\$empty, 196

\$error_field, 113

\$exposed, 196

\$feed_icon, 196

\$field, 94, 110, 127, 237

\$field_empty, 95

\$field_name, 94

\$field_name_css, 94

\$field_type, 94

\$field_type_css, 94

\$field->class, 199

\$field->content, 199

\$field->handler, 199

\$field->inline, 199

\$field->inline_html, 199

\$field->raw, 199

\$field->separator, 200

\$fields, 198, 199
\$footer, 196
\$header, 196, 198
\$item, 127
\$items, 94
\$label, 94
\$label_display, 95
\$more, 196
\$NAZWA_POLA, 97
\$node, 94, 127
\$op, 110
\$order, 238
\$page, 94
\$pager, 196
\$referring_field, 101
\$referring_node, 101
\$row, 200
\$row_classes, 198
\$rows, 196, 198
\$teaser, 94
\$this, 225
\$title, 198
\$view, 199
\$view->args, 242
\$widget->label, 201
\$widget->operator, 201
\$widget->widget, 201
\$widgets, 201
views-exposed-widget, 201
views-operator, 201
views-widget, 201
znacznik span, 211

Chcesz szybko stworzyć wydajny sklep internetowy, elegancką witrynę firmową, sprawny intranet korporacyjny bądź funkcjonalny portal społecznościowy, ale już przeraża Cię ilość związanej z tym pracy? Zapewne dlatego zainteresowałeś się Drupaliem? I słusznie. Jakże inne narzędzie open source umożliwiła programistom tak błyskawiczną oraz bezproblemową realizację nawet bardzo ambitnych i zaawansowanych projektów IT? Dzięki zaangażowaniu utalentowanej i aktywnej społeczności Drupala masz przecież możliwość korzystania z tysięcy praktycznych modułów, które rozszerzają możliwości rdzenia tego systemu i pozwalają na niemal nieograniczone wzbogacanie projektu o kolejne przydatne funkcjonalności. Choć trudno w to uwierzyć, już kilka kliknięć wystarczy, aby stworzyć witrynę internetową o elastycznych, zróżnicowanych możliwościach.

Niezależnie od tego, czy jesteś programistą, projektantem, administratorem, architektem czy konsultantem — oto doskonały przewodnik po najbardziej wartościowych modułach i technikach włączania ich do swojego projektu. Znajdziesz tu słynną „wielką trójkę”, czyli moduły Content Construction Kit (CKK), Views i Panels, których potencjał pozwala na niesłychanie szybkie tworzenie idealnie skrojonych stron. Niemożliwe? Przekonaś się, że tak! Współautor tej niezwykłej książki, Earl Miles, od wielu lat jest aktywnym członkiem społeczności związanej z Drupalem, dzięki czemu w dużym stopniu wpłynął na kierunek, w jakim podąża ten system. Earl jest także kluczowym uczestnikiem projektu budowy modułu CKK oraz autorem modułów Views i Panels. Trudno zatem wybrać sobie lepszą osobę, która mogłaby wprowadzić Cię w tajniki ich skutecznego wykorzystania! Autorzy zamieścili tu wszystkie informacje, których nie znajdziesz nigdzie indziej, a także obszerny przykładowy kod „recepturowy”, stworzony w celu łatwej adaptacji i ponownego wykorzystania.

Ta książka pomoże Ci:

- użyć modułu Content Construction Kit (CKK) w celu dostosowania formularzy wprowadzania danych zgodnie z własnymi wymaganiami
- zwiększyć elastyczność witryny za pomocą systemu segmentów Drupala
- dodać opcjonalne typy pól z uwzględnieniem pól opartych na kodzie PHP
- dostosować motyw w celu wyświetlania danych w sposób bardziej atrakcyjny i czytelny
- zbudować wielofunkcyjne formaty wyświetlania zapytań modułu Views
- dokonać integracji z bazami danych SQL
- dostosować moduł Views za pomocą zależności, argumentów i filtrów
- zoptymalizować wydajność zapytań
- dodać style niestandardowe za pomocą modułu Panels
- utrzymać pełną kontrolę nad witrynami podczas wdrażania ich w środowisku produkcyjnym

Błyskawicznie twórz elastyczne, wielofunkcyjne witryny internetowe!

Earl Miles jest stałym członkiem i założycielem Drupal Association. Stworzył moduły Views i Panels, które przyczyniły się do dobrego rozwoju systemu Drupal. Earl nadal służy pomocą, a także publikuje wskazówki i szkolenia dla społeczności związanej z tym systemem. Obecnie pracuje też jako programista w firmie K1, która bierze udział w rozwoju i wykorzystaniu oprogramowania open source. Błąd Earla jest dostępny pod adresem: <http://www.andreymiles.com>.

Lynette Miles od piętnastu lat jest związana zawodowo z obsługą techniczną oprogramowania. Aktywnie współpracuje z zespołem tworzącym dokumentację systemu Drupal, a także pomaga w zarządzaniu kolejkami wsparcia modułów Views i Panels.

helion.pl
Internetowa księgarnia

W katalogu: 7459



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki nabywanej czytać:
• <http://helion.pl/kuchnia>
Zamów informacje o nowościach:
• <http://helion.pl/news>

Helion SA
ul. Kościelna 1c, 44-100 Gliwice
tel.: 32 230 10 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po **WIECEJ**



KOD KORZYŚĆ

ISBN 978-83-246-3548-1



9 788324 635481

Cena: 59,00 zł

Informatyka w najlepszym wydaniu