WORDPRESS od podszewki

POZNAJ NAJPOPULARNIEJSZY SYSTEM CMS!



Co to jest WordPress?

Jak wykorzystać materiały multimedialne we wpisach?

Jak zainstalować WordPress na własnym serwerze?

BUD E. SMITH
MICHAEL MCCALLISTER

Helion

helion.pl

Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

Katalog książek

- Katalog online
- Zamów drukowany katalog

Twój koszyk

- Dodaj do koszyka

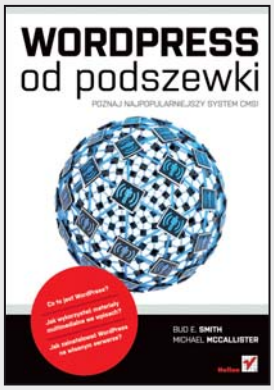
Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

Czytelnia

- Fragmenty książek online

WordPress od podszewki



Autor: Bud E. Smith, Michael McCallister
Tłumaczenie: Julia Szajkowska
ISBN: 978-83-246-3080-6
Tytuł oryginału: WordPress In Depth
Format: 168×237, stron: 480

Poznaj najpopularniejszy system CMS!

- Co to jest WordPress?
- Jak wykorzystać materiały multimedialne we wpisach?
- Jak zainstalować WordPress na własnym serwerze?

WordPress to obecnie najpopularniejszy system zarządzania treścią. Według badań W3Techs jego udział w rynku systemów CMS wynosi ponad 54%! Co wpłynęło na tę popularność? Czym wyróżnia się on na tle konkurencji? Są to niezwykle trudne pytania, ale w tym przypadku można bez wahania postawić na niezwykłą elastyczność, łatwość konfiguracji, estetyczny domyślny wygląd oraz ogrom dodatków, dzięki którym możesz uzupełnić go o nawet najbardziej wymyślną funkcjonalność.

Dzięki tej książce poznasz WordPress od podszewki. Zgłębisz jego najskrytsze tajemnice oraz nauczysz się dostrajać nawet najbardziej zaawansowane funkcje. Na samym początku dowiesz się, jak podejść do tematu tworzenia własnej witryny, opartej o WordPress. Po interesującym wstępie przyjdzie czas na konfigurację. Zobaczysz, jak zadbać o wygląd strony, przygotować własny nagłówek, zorganizować menu oraz dodać pierwszy wpis. W kolejnych rozdziałach zapoznasz się z metodami publikacji materiałów multimedialnych, dodawania grafik do wpisów oraz projektowania własnych motywów. Książka ta sprawdzi się świetnie w rękach każdego użytkownika, pragnącego zaistnieć w sieci!

- Wersje platformy WordPress
- Użytkownicy systemu WordPress
- Przygotowania do otwarcia witryny opartej o WordPress
- Konfiguracja wyglądu – wybór motywu, edycja nagłówka
- Tworzenie wpisów
- Dodawanie odnośników i zarządzanie nimi
- Kategoryzowanie wpisów
- Wykorzystanie własnego kodu HTML
- Dodawanie statycznych stron
- Dodawanie do wpisów materiałów multimedialnych
- Instalacja systemu WordPress na własnym serwerze
- Tworzenie i nabywanie nowych motywów
- Wykorzystanie języka PHP
- Przykładowe witryny, oparte o WordPress

Zacznij prowadzić własną witrynę WWW!

Kontakt

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

SPIS TREŚCI

O autorach 13

Podziękowania 14

Wprowadzenie: Witaj! 15

Część I Początki pisania bloga 21

Rozdział 1. Pierwszy kontakt z systemem WordPress 23

Czym jest WordPress? 23
- Kilka słów o różnych wersjach platformy WordPress 24
- Społeczność użytkowników systemu WordPress 25
- Jakie trzeba ponieść koszty? 26

Kim jest przeciętny użytkownik systemu WordPress? 31
- Godny przeciwnik — system Blogger 31
- Pierwsze studium przypadku — początkujący użytkownik systemu WordPress 33
- Drugie studium przypadku — blog biznesowy 36

Utrzymanie bloga na serwerze 38
- Przenosiny 40
- Porównanie WordPress.com z WordPress.org 40

Inne sposoby zaistnienia w sieci 42

Rozdział 2. Na dobry początek 45

Tworzenie bloga 45
Czy planujesz duży ruch? 47
Gra w „Nazwij domenę" 48
- Rozpocznij z własną nazwą domeny 49
- Dodaj własną domenę później 49
- Pozostań na zawsze wierny domenie WordPress.com 50

Rejestracja 51
- Zakładanie konta na platformie WordPress.com 51
- Zakładanie własnego bloga 53
- Korzystanie ze źródeł 55

Zmienianie ustawień początkowych 55
Ustawienia ogólne 56
Pisanie 58
Czytanie 59
Dyskusja 62
Media 64
Prywatność 66
Inne grupy opcji menu Ustawienia 67
Opcje konfiguracji spoza menu Ustawienia 68
Usuń witrynę 68
Oceny 69
Publikowanie wpisów za pomocą poczty elektronicznej 71

Rozdział 3. Zadbaj o wygląd bloga 75
Ten urok, ten styl 75
O motywach słów kilka 76
Czego należy oczekiwać od motywu? 77
Kolory 78
Liczba kolumn 80
Stała czy zmienna szerokość? 81
Widgety i inne funkcje 82
Czy warto modyfikować kod arkusza stylów CSS wybranego motywu? 84
Spis motywów graficznych 85
Wybieranie motywu 99
Ustawianie własnego nagłówka 101
Zmienianie ustawień dodatku Related Links 103
Jak dodawać widgety? 104
Umieszczanie widgetów na blogu 105
Przegląd dostępnych widgetów 106
Dodawanie nowych widgetów 107
Krótki spis widgetów 109
Jak najlepiej wykorzystać widgety? 113

Część II Prowadzenie bloga 117

Rozdział 4. Dodawanie pierwszego wpisu 119
O wpisach słów kilka 119
Funkcja QuickPress 121
Dodawanie nowego wpisu 123
Pisanie w trybie pełnoekranowym 125
Podstawowe polecenia formatowania 127
Przykład formatowania tekstu 129
Odnośniki jako forma formatowania 131
Dodatkowe metody formatowania tekstu 132
Dodawanie odnośników 134
Przepis na dodawanie odnośników 136
Tworzenie odnośnika 137
Zapisywanie, sprawdzanie pisowni i publikowanie postów 138

Rozdział 5. Przenieś wpisy w inny wymiar 141
Jak jeszcze można rozbudować wpis? 141
Dodawanie wpisów do kategorii 142
Tworzenie kategorii i korzystanie z nich w praktyce 144
Kontrolowanie listy kategorii 145
Przenoszenie wpisów do innej kategorii 149
Nadawanie tagów 151
Tworzenie tagów i ich praktyczne zastosowanie 153
Zarządzanie tagami 154
Zarządzanie tagami powiązanymi z wpisami 157
Zawartość widgetu Odnośniki 158
Dodawanie nowych odnośników do widgetu 159
Dodawanie odnośników 162
Kategorie odnośników 163

Rozdział 6. HTML w widgetach i innych częściach bloga 165
W jaki sposób rozwinąć blog za pomocą kodu HTML? 165
Przybornik zaawansowanego blogera 166
Podstawy HTML dla WordPress 168
Znaczniki kodu HTML 168
Karty Wizualny i HTML 172
Formatowanie znaków 175
Formatowanie list 177
Tworzenie odnośników 180
Znacznik more 182
Edycja kodu HTML w systemie WordPress 182
Z jakich znaczników korzystać? 183
Film z serwisu YouTube — przykład zagnieżdżania kodu HTML 185
Sposoby wykorzystywania widgetu Tekst 188
Możliwe zawartości widgetu Tekst 189
Co można robić z widgetem Tekst 191
Prosty przykład pracy z widgetem Tekst 192

Rozdział 7. Rozbuduj blog o nowe możliwości 195
Popraw swój blog 195
Nowi użytkownicy bloga 197
Podział ról w systemie WordPress 198
Zapraszanie nowych użytkowników i przypisywanie im ról 199
Dodawanie statycznych stron 201
Planowe wprowadzanie stron 202
Projektowanie statycznej strony 207
Tworzenie i edycja statycznych stron 208
Wstawianie ankiet 210

Rozdział 8. Statystyki i przyciąganie czytelników 217
Jak sprawić, by więcej osób zaglądało na Twój blog? 217
Statystyki systemu WordPress 218
Po co to robisz? 219
Statystyki w panelu administracyjnym bloga 221
Strona Statystyki w panelu administracyjnym systemu WordPress 222
Analiza danych z grupy Odsyłający 224

Pouczające wnioski płynące z analizy danych na stronie Odsyłający 225
Odnośniki do witryny 228
Analiza danych z grupy Frazy wyszukiwarek 229
Grupa Top Posts & Pages 232
Kliknięcia 233
Statystyki bloga — tabele z podsumowaniem 235
W jaki sposób dołączyć do grona subskrybentów? 236
Używanie widgetu RSS 238

Część III Dalszy rozwój bloga 241

Rozdział 9. Grafika we wpisach 243
Po co wpisom grafika i inne media? 243
Prawa autorskie i rodzaje plików multimedialnych 245
Umieszczanie grafiki we wpisach 247
Wskazówki dla autorów odnośnie do grafiki 248
Wysyłanie plików na serwer i umieszczanie ich we wpisach 252
Umieszczanie grafiki w bibliotece mediów 253
Edytowanie obrazu dodanego do biblioteki mediów 255
Dodawanie do wpisów grafiki z dysku komputera 256
Umieszczanie we wpisach obrazów z biblioteki mediów 260
Umieszczanie we wpisach obrazów z sieci 261

Rozdział 10. Ulepszenia, audio i wideo 265
Podnoszenie możliwości bloga 265
Krótki opis ulepszeń 266
Własny CSS 267
Nieograniczona liczba prywatnych użytkowników 268
Brak reklam 268
Więcej miejsca na serwerze (5 GB, 15 GB, 25 GB, 50 GB i 100 GB) i możliwość dodawania plików audio 269
Usługa VideoPress 270
Nazwy domen 270
Ogólne różnice 272
Wybieranie dobrej nazwy domeny 273
Poddomeny domeny WordPress.com kontra własne domeny 273
Jak funkcjonują nazwy domen 274
Jak wybrać dobrze nazwę domeny drugiego lub trzeciego poziomu? 275
Wykorzystywanie plików audio na blogu 278
Gdzie umieszczać pliki dźwiękowe? 279
Umieszczanie pliku audio we wpisie 280
Umieszczanie plików wideo we wpisach 283
Gdzie umieszczać pliki wideo? 284
Umieszczanie plików wideo we wpisie 285

Część IV Samodzielne instalowanie WordPress 289

Rozdział 11. Instalowanie i aktualizacja oprogramowania WordPress 291
Niech ktoś zrobi to za Ciebie — preinstalowany WordPress na serwerze 291
Serwer przyjazny blogerowi 292
Po co PHP i MySQL? 294

Umieszczanie plików na serwerze za pomocą protokołu FTP 296
Przygotowania do pięciominutowego instalowania systemu WordPress 299
Instalowanie w pięć minut 302
Aktualizacja systemu WordPress 304
Przenoszenie danych z platformy WordPress.com 307
System WordPress na własnym serwerze 310
Instalowanie serwera WWW i bazy danych w systemie Windows 311
Edytowanie pliku konfiguracyjnego systemu WordPress 314
Krótki przegląd zawartości kokpitu systemu WordPress 317
Dostosowywanie kokpitu do własnych potrzeb 319

Rozdział 12. Przybornik systemu WordPress — motywy 321
Kupowanie motywów 321
Jak szukać darmowych motywów? 324
Wybieranie motywów płatnych 327
Instalowanie i włączanie motywu 332
Kwestie bezpieczeństwa a wybieranie motywu graficznego bloga 333

Rozdział 13. Przybornik systemu WordPress — wtyczki 335
Segmentowa budowa systemu WordPress 335
Czym jest wtyczka? 336
Szukanie wtyczek 337
Wybieranie wtyczek z katalogu WordPress i przeprowadzanie instalowania 338
Szukanie wtyczek poza katalogiem WordPress 340
Szukanie płatnych wtyczek 340
Ręczne instalowanie i włączanie wtyczek 341
Pobieranie nowej wtyczki 341
Testowanie nowej wtyczki 342
Wybieranie i włączanie wtyczki 342
Zarządzanie wtyczkami i przeprowadzanie aktualizacji 343
Niektóre spośród „niezbędnych" wtyczek 344
Podstawowe wtyczki WordPress — Akismet i Hello Dolly 345
Instalowanie wtyczki Akismet 345
Administracja i bezpieczeństwo 346
Statystyki 348
Komentarze 349
Reklama i optymalizacja pod kątem działania wyszukiwarek 350
Odnośniki 351
Multimedia 351
Serwisy społecznościowe 352

Rozdział 14. Tworzenie motywów za pomocą arkuszy stylów CSS 353
Tworzenie arkuszy stylów 353
Wstęp do XHTML 354
Standardy tworzenia strony internetowej 354
XML i XHTML 356
Kod HTML zgodny ze standardem XHTML w WordPress 358
Wstęp do CSS 362
Rola arkusza stylów 362
Czym jest arkusz stylów? 364
Anatomia arkusza CSS 365

Poprawianie motywu poprzez arkusz stylów 368
Gdy gotowy motyw spełnia oczekiwania 369
Poprawianie gotowego motywu WordPress.org 372
Tworzenie arkusza stylów motywu od zera 376

Rozdział 15. Podstawy PHP — motywy i wtyczki 391
Wstęp do PHP 391
Instalowanie PHP na serwerze roboczym 393
Skrypty PHP — podstawy składni 396
PHP i MySQL 402
Szablony motywów graficznych w PHP 404
Szablon strony głównej 405
Szablon panelu bocznego 409
Szablon nagłówka 411
Szablon stopki 412
Analiza kodu istniejącej wtyczki 412
Wysyłanie wtyczek do katalogu WordPress 418
Dlaczego nie warto wysyłać wtyczek do katalogu WordPress? 418
Dodawanie wtyczki do katalogu 419

Dodatki 421

Dodatek A WordPress.com kontra WordPress.org 423
Nazewnictwo 424
Wybierz WordPress.org 426
Zarabianie na blogu 427
Reklamy Automattic na Twojej stronie 428
Pomoc społeczności WordPress 429
Wybieranie dobrego serwera 430
Wybierz WordPress.com 430

Dodatek B Dokumentacja systemu WordPress 433
Przedzieranie się przez gąszcz dokumentów w sieci 433
Kodeks WordPress 434
Kłopoty z dokumentacją WordPress 436
Odkrywanie dokumentacji WordPress za pomocą Google 437

Dodatek C Przykłady blogów WordPress 441
Dlaczego warto odwiedzać słynne blogi? 441
XXL 442
Surfin' Safari 442

Dodatek D Przykłady blogów WordPress.com 445
Równe prawa dla .com-owców 445
Znane cytaty 445
Outland Institute 446
Muzyka na stronach SUFU 447

Dodatek E Mapa witryny WordPress 449
Nawigowanie po serwisach WordPress 449
Nawigowanie po serwisie WordPress.com 449
Strona główna — http://wordpress.com 449
System wsparcia — http://en.support.wordpress.com 450

Fora — http://en.forums.wordpress.com/ 451
Dodatki — http://en.wordpress.com/features 452
Nowości — http://en.blog.wordpress.com 452
O nas — http://en.wordpress.com/about 453
Usługi zaawansowane — http://en.wordpress.com/advanced-services 454
Nawigowanie po serwisie WordPress.org 454
Strona główna — http://wordpress.org 455
Prezentacja — http://wordpress.org/showcase 455
Motywy, wtyczki i miejsce narzekań — http://wordpress.org/extend 457
Świat WordPress — http://wordpress.org/about 460
Co słychać u programistów — http://wordpress.org/development 461
Do przyjaciół po pomoc — http://wordpress.org/support 461
Wybieranie serwera — http://wordpress.org/hosting 462
Pobieranie wersji instalacyjnej — http://wordpress.org/download lub http://pl.wordpress.org/ 462

Dodatek F Przenoszenie zawartości z innych systemów blogowych 465
Bezpośredni import danych na platformę WordPress 466
Przenosiny z platform Blogger i BlogSpot 466
Przenosiny z platformy LiveJournal 467
Przenosiny z platformy Movable Type lub TypePad 467
Przenosiny z platformy Textpattern 468
Transfer pośredni 468
Przenoszenie danych w postaci zawartości kanału RSS 468
Przenoszenie danych z pliku (X)HTML 469

Skorowidz 471

6

HTML W WIDGETACH I INNYCH CZĘŚCIACH BLOGA

W jaki sposób rozwinąć blog za pomocą kodu HTML?

Urok korzystania z systemu WordPress polega w dużej mierze na tym, że możesz osiągnąć wiele bez podawania choćby jednej linii kodu. Miliony wspaniałych blogów powstały i funkcjonują w sieci dlatego, że WordPress i inne tego typu aplikacje zdejmują z autorów obowiązek dbania o techniczną stronę witryny.

Jednak w pewnym momencie możesz odczuć potrzebę odrzucenia piły i hebla na rzecz bardziej zaawansowanych narzędzi. System WordPress został zaprojektowany z najwyższą dbałością o szczegóły w taki sposób, by pozwolić Ci możliwie długo obchodzić się bez skomplikowanej wiedzy informatycznej, a w chwili gdy bardziej złożone rozwiązania okażą się konieczne, podać Ci podstawowy zakres znaczników, który jednak pozwoli osiągnąć wszystkie zamierzone cele.

Pierwszym i najprostszym z zaawansowanych narzędzi oferowanych przez WordPress jest funkcja umożliwiająca stosowanie kodu HTML, podstawowego języka sieci. Skrót HTML pochodzi od angielskiej nazwy *Hypertext Markup Language* (hipertekstowy język znaczników). Znaczniki kodu HTML odpowiadają za wygląd i zachowanie się poszczególnych elementów strony. Przez pierwsze kilka lat funkcjonowania internetu, od jego powstania w 1989 roku, HTML był jedynym narzędziem, jakim dysponowaliśmy.

Język HTML jest również interfejsem dla protokołu HTTP (ang. *Hypertext Transfer Protocol* — hipertekstowy protokół przesyłu danych) odpowiedzialnego za regulację pracy odnośników i serwerów nazw. To właśnie HTML tworzy hiperłącza — odnośniki, które klikasz, by przejść na inne strony.

WordPress.com dopuszcza używanie kodu HTML — do uzyskania tej możliwości potrzebujesz tylko jednego kliknięcia. Wystarczy, że otworzysz kartę *HTML* edytora wpisów.

Niektóre z rutynowych działań każdego autora blogów wymuszą na Tobie użycie fragmentów kodu HTML. Przykładem może być dodawanie do bloga nagrania z serwisu YouTube, który udostępnia odwiedzającym kod HTML odpowiedzialny za wyświetlanie filmu w formacie Flash na stronie WWW. Kod należy wkleić w wybranym miejscu swojej witryny.

Do tego jeden z najczęściej używanych i najbardziej elastycznych widgetów, *Tekst*, pozwala formatować swoją zawartość za pomocą znaczników HTML. Dzięki temu zyskujesz możliwość nadawania wybranej formy fragmentom tekstu i umieszczania na stronie odnośników HTML.

W tym rozdziale pokażemy Ci, jak używać kodu HTML w systemie WordPress. Wiedza ta przyda Ci się niezależnie od tego, czy będziesz próbował wprowadzać proste zmiany w treści wpisu lub zawartości widgetu *Tekst*, czy też pokusisz się o uzyskanie bardziej ambitnych efektów. W tym drugim przypadku będziesz musiał posiłkować się także innymi narzędziami, które opisaliśmy w tej książce.

Przybornik zaawansowanego blogera

Używając bardziej złożonych środków, możesz przenieść swój blog w zupełnie nowy wymiar. Sposób ich wykorzystania zależy w dużej mierze od tego, czy prowadzisz blog na platformie WordPress.com, czy posługujesz się oprogramowaniem WordPress, które z założenia jest bardziej elastyczne i daje Ci więcej możliwości działania, ale też przenosi na Ciebie większość odpowiedzialności za poprawne funkcjonowanie witryny.

Staraliśmy się, jak zawsze, stworzyć szczeble, po których mógłbyś się wspiąć na najwyższy poziom. Zorganizowaliśmy materiał tak, byś opanował najpierw pewien fragment wiedzy, mógł od razu go zastosować, przyswoił następną partię, znów jej użył, i w ten sposób, małymi krokami, osiągnął poziom zaawansowanego blogera.

Kod HTML może pojawić się na każdym blogu z rodziny WordPress, także na tych prowadzonych w systemie WordPress.com, o czym przekonasz się, czytając ten rozdział. Zaprezentujemy Ci zasady pracy z kodem XHTML, nowszą i nieco bardziej restrykcyjną wersją języka HTML.

Ludzie pracujący nad kodem systemu WordPress, użytkownicy oraz wszyscy choć trochę związani z ideą internetu utknęli w pułapce zmiany standardów. Język XHTML miał z czasem zastąpić stary kod HTML, ale ponieważ trwają prace nad nową, piątą wersją HTML, XHTML został zdegradowany do rangi rozwiązania tymczasowego.

O ile nie zaznaczymy tego wyraźnie, zakładaj, że wszystkie prezentowane tu rozwiązania będą sprawdzać się zarówno w czasie pracy w standardzie HTML, jak i XHTML. Jeśli planujesz używanie dużych porcji kodu (X)HTML, powinieneś poznać różnice między jego wersjami. Możesz rozpocząć od lektury artykułu *http://codex.wordpress.org/HTML_to_XHTML*[1] lub zapoznać się z treścią kursu *http://www.kurshtml.boo.pl/html/roznice_miedzy_html_4_a_xhtml_1_0,xhtml.html*.

Kaskadowe arkusze stylów, czyli CSS, to dodatek do języka HTML. Dzięki niemu możesz zarządzać zawartością strony w bardziej zorganizowany sposób. Wszystkie motywy graficzne dla blogów WordPress opierają się na kodzie CSS. Początki tego narzędzia były trudne, gdyż kilka nieco odmiennych wersji ubiegało się o miano standardu, ale obecnie wygląda na to, że o dalszy rozwój tej technologii możemy być spokojni.

W dowolnej chwili możesz przeprowadzić edycję szablonu CSS wybranego motywu i sprawdzić jej efekty, nawet jeśli prowadzisz blog na platformie WordPress.com, ale żeby zapisać zmiany na stałe, musisz uiścić roczną opłatę w wysokości około 15 dolarów. Szczegóły tej operacji opisaliśmy w rozdziale 10. „Ulepszenia, audio i wideo".

Jeśli zaś prowadzisz blog, korzystając z oprogramowania WordPress, możesz używać dowolnych motywów, modyfikować je zgodnie z własnymi potrzebami, a nawet tworzyć nowe — wszystko to dzięki CSS. Tego rodzaju działaniom poświęciliśmy rozdział 11. „Instalowanie i aktualizacja oprogramowania WordPress".

Do zestawu narzędzi dorzuć jeszcze język PHP (*PHP Hypertext Preprocessor* — hipertekstowy preprocesor PHP), który podziała na blog niczym dopalacz. PHP to język skryptowy. To w nim powstają dodatki systemu WordPress, które otwierają nowe możliwości przed blogami prowadzonymi w tym systemie. Dodatki zostały opisane dokładniej na początku rozdziału 13. „Przybornik systemu WordPress — wtyczki". Widgety, kilkadziesiąt dodatków o ograniczonej funkcjonalności, opisaliśmy w rozdziale 3. „Zadbaj o wygląd bloga", tutaj (widget *Tekst*) oraz w rozdziale 8. „Statystyki i przyciąganie czytelników" (dodatek *RSS*).

Oto zawartość przybornika blogera:

- Motywy — dostępne w niewielkiej liczbie dla użytkowników WordPress.com oraz w znacznie bogatszej ofercie dla użytkowników WordPress.org.
- Widgety — kilkadziesiąt miniprogramów dostępnych dla blogów działających na obydwu platformach.
- Kod HTML — można go używać niezależnie od platformy, na której prowadzisz blog.
- Kod CSS — w blogach prowadzonych na platformie WordPress.com pozwala, po wykupieniu odpowiedniego dodatku, swobodnie modyfikować motywy poprzez zmienianie zawartości pliku CSS, a w blogach pracujących na oprogramowaniu WordPress może być wykorzystywany bez ograniczeń.
- Kod PHP — pozwala tworzyć dodatki do blogów działających na oprogramowaniu WordPress.

I co powiesz na takie szczeble zdobywania wiedzy?

[1] Artykuł w języku angielskim — *przyp. tłum.*

Podstawy HTML dla WordPress

Jest wiele doskonałych podręczników do nauki języka HTML, lecz stosowanie go w systemie WordPress nieco odbiega od zwyczajnego tworzenia kodu na potrzeby przeglądarek. WordPress przejmuje za Ciebie wiele tych aspektów pracy z kodem HTML, które zazwyczaj są źródłem problemów, natomiast po pewnym okresie pracy w środowisku WordPress odkryjesz zapewne, że podstawy HTML już opanowałeś.

W tym rozdziale spróbujemy przekazać Ci wiedzę z zakresu podstaw HTML, używając do tego platformy WordPress. Gdy już zrozumiesz zasadę tworzenia kodu HTML i poznasz reguły jego działania w środowisku WordPress, będziesz mógł wykorzystać możliwości, jakie niosą z sobą polecenia HTML i XHTML, żeby rozwijać swoją witrynę.

Będziemy posługiwać się przede wszystkim edytorem wpisów w obydwu jego wersjach — wizualnej i HTML. (Przypominamy, że edytor wpisów służy do tworzenia nowych wpisów i stron — normalnych stron WWW zintegrowanych z blogiem — oraz do edycji opublikowanego i zapisanego materiału).

Style formatowania dostępne w zakładce *Wizualny* to najprostsze polecenia kodu HTML, obsługiwane przez większość przeglądarek i interpretowane niemal niezależnie od systemu operacyjnego.

Strona edytora wpisów zawiera dwie zakładki, dzięki którym możesz posługiwać się albo wizualną wersją edytora, albo oknem, które potrafi interpretować polecenia HTML. Okno edytora wizualnego przedstawia wygląd treści wpisu w taki sposób, w jaki będzie on prezentowany odwiedzającym Twoją witrynę. W typowych edytorach kodu HTML taki tryb pracy określa się często skrótem WYSIWYG, który powstał z angielskiego określenia *What You See Is What You Get* (dostajesz to, co widzisz).

> **Uwaga**
> Opisane tu podstawy języka HTML mają zastosowanie na blogach prowadzonych na obydwu platformach — WordPress.com i oprogramowaniu WordPress.

Znaczniki kodu HTML

Cała prawda na temat podstaw języka HTML kryje się w jego nazwie — hipertekstowy język znaczników.

Określenie **język znaczników** informuje nas, że instrukcje języka są zapisywane za pomocą serii znaków określających formatowanie tekstu lub różnego rodzaju działania. Taka instrukcja w języku HTML nosi nazwę **znacznika**.

Tag przyjmuje postać: `<b>`. Nawiasy ostre informują przeglądarkę internetową, że umieszczony wewnątrz tekst stanowi polecenie języka i w związku z tym nie powinien pojawiać się na ekranie.

Większość znaczników jest dwuczłonowa, to znaczy składa się z części otwierającej i zamykającej:

```
Ostatnie słowo tego zdania zostanie <b>pogrubione</b>.
```

Tak wyglądałby tekst wprowadzony w zakładce *HTML* okna edycji wpisów. W zakładce *Wizualny* zdanie zostałoby wyświetlone następująco:

Ostatnie słowo tego zdania zostanie **pogrubione**.

Oto pierwsza reguła pracy z językiem HTML. Aby pogrubić wybrany fragment tekstu, należy otoczyć go odpowiednimi znacznikami, to jest znacznikiem otwarcia `<b>` i znacznikiem zamknięcia `</b>`. Podobnie uzyskuje się efekt kursywy — wybrany fragment tekstu trzeba objąć znacznikiem otwarcia `<i>` i znacznikiem zamknięcia `</i>`.

> **Uwaga**
> Nie umieszczamy tu informacji na temat kodu HTML wyłącznie dla rozrywki czy na wszelki wypadek. Są one niezbędne do zrozumienia, dlaczego WordPress stosuje je w sposób, w jaki to robi. Bez wprowadzenia do HTML wyjaśnienia byłyby zbyt zawiłe. Pominęliśmy tu całą historię powstawania języka, kilometry tekstu umieszczonego w sieci i na papierze, dokumentującego zmagania pomiędzy konkurującymi standardami. Podajemy Ci suche informacje, które wyjaśnią, jak tworzyć kod HTML w systemie WordPress.

Język HTML powstawał zgodnie z określonymi wcześniej założeniami, ale nie w oderwaniu od pewnych ideałów, co czasami doprowadzało do mniejszych lub większych tarć.

Przykładowo zakładano, że HTML będzie służył do przekazywania **treści**, a nie **formy**. Znaczniki pogrubienia i kursywy zdecydowanie odpowiadają za formę prezentowania danych. Jednocześnie są niezwykle popularne wśród twórców stron, ponieważ każdy z nich sprawdza się na większości urządzeń oferujących dostęp do sieci, a i jego nazwa dokładnie odzwierciedla działanie.

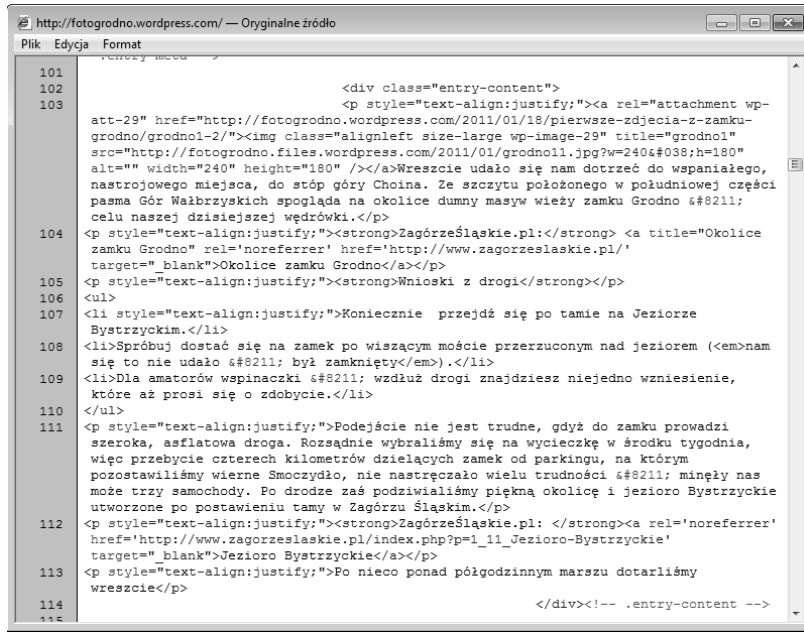
Nazwa formatowania różni się zazwyczaj od naszych wyobrażeń na temat efektów, jakie powinno dawać owo formatowanie. Purysta konwencji HTML wolałby, żeby znaczniki odzwierciedlały intencję kryjącą się za ich zastosowaniem, a nie oddawały samą nazwę formatowania. Dlatego żeby pozostać wiernym ideologii HTML, należałoby stosować znaczniki `<em>` do podkreślania[2] znaczenia tekstu oraz `<strong>`, by je mocniej[3] uwidocznić. Oznacza to, że puryści kodu HTML uzyskują efekt pochylenia czcionki po otoczeniu jej otwierającymi i zamykającymi znacznikami `<em>`, a pogrubienia — otwierającymi i zamykającymi znacznikami `<strong>`, mimo że znaczniki te trudno wpisać w kod i jeszcze trudniej zapamiętać.

[2] Od ang. *emphasize* — podkreślać, kłaść nacisk — *przyp. tłum.*

[3] Od ang. *strong* — mocny, silny — *przyp. tłum.*

Arkusze stylów CSS wydawały się triumfem puryzmu, gdyż wreszcie treść została jawnie oddzielona od formy. W praktyce zaś wygląda to tak, że przez konieczność używania standardu CSS spędzamy znacznie więcej czasu, dopracowując szczegóły formatowania drobiazgów w przypadkach, w których włożony wysiłek znacznie przekracza wagę treści.

Tak właśnie wygląda sprawa ze znacznikami HTML. Rysunek 6.1 przedstawia kod źródłowy strony przykładowego bloga. Aby dokończyć rozważania na temat nazwy języka HTML, zajmijmy się teraz częścią HT tego akronimu — hipertekstem.



Rysunek 6.1. Kod każdej oglądanej strony jest łatwo dostępny

> **Wskazówka**
> Działaniu HTML możesz przyjrzeć się na dowolnej stronie WWW. Niemal każda przeglądarka internetowa oferuje funkcję wyświetlania kodu źródłowego strony. W przeglądarce Internet Explorer 8 polecenie to znajduje się w menu *Strona/Pokaż źródło* (można też użyć skrótu klawiszowego *Ctrl+U*). W innych przeglądarkach należy szukać w poleceniu *Widok* lub podobnych. Polecenia te otwierają okno przeglądarki, w którym pojawia się kod HTML lub CSS aktywnej strony.

To, co odróżnia **hipertekstowy** język znaczników HTML od innych kodów znacznikowych, to wykorzystanie dodatkowych znaczników pozwalających wprowadzać na stronę hiperłącza. Dla przykładu podajemy kod, który umieszczony na dowolnej stronie WWW zaprowadzi odwiedzającego na stronę projektu WordPress.com.

```
Aby dostać się do serwisu WordPress.com, kliknij <a href=
➥"http://www.wordpress.com">tutaj</a>.
```

W przedstawionym przykładzie funkcję hiperłącza pełni słowo **tutaj**. O ile domyślne ustawienia wyświetlania strony nie zostaną zmienione, odnośnik będzie wyróżniony kolorem niebieskim i podkreśleniem. Gdy użytkownik kliknie odnośnik, w oknie przeglądarki pojawi się zawartość strony głównej serwisu WordPress.com.

Nazwa znacznika `<a>` pochodzi od angielskiego słowa **anchor** (kotwica). (Zagadnienie zarządzania kotwicami staje się bardzo istotne w przypadku serwisów o skomplikowanej strukturze).

Nazwa pojawiająca się obok znacznika, `href`, jest skrótem angielskiego określenia **hypertext reference** (odnośnik hipertekstowy). Wyrażenie to nie jest znacznikiem, lecz definicją umieszczaną wewnątrz znacznika (co widać w przykładzie). Określa się je mianem **atrybutu**. Mówiąc o **hipertekście**, mamy na myśli tekst, który może pojawiać się w różnych plikach umieszczonych na różnych serwerach.

Opanowywanie podstaw języka HTML w środowisku WordPress.com jest wygodne, ponieważ sam system zdejmuje z Ciebie odpowiedzialność za dbanie o mniej istotne w danym momencie szczegóły techniczne. Ty opracowujesz jedynie treść i wygląd samego wpisu. To WordPress dba o umieszczenie go w szablonie (pisaliśmy o tym w rozdziale 3.). Również WordPress zapewnia miejsce na serwerach dla Twoich plików.

To rozwiązanie pozwala Ci uniknąć jednej z największych bolączek projektantów stron internetowych. Strony powstają zazwyczaj na komputerach ich twórców i dopiero gdy są gotowe, przenosi się je na serwer WWW.

Transfer danych na inną maszynę zawsze wiąże się ze sporym ryzykiem, choćby dlatego, że obowiązuje na niej inna struktura katalogów. Podczas takiej operacji bardzo łatwo o zerwanie powiązań między stronami witryny oraz innego rodzaju katastrofy.

System WordPress.com sprawuje bardzo szczegółową kontrolę nad tym procesem. Nigdy nie umieszczasz na serwerze tekstu wpisu i jego tytułu bezpośrednio w bazie, lecz wykonujesz to za pośrednictwem formularza edycji. Podobnie przebiega wysyłanie grafiki czy plików audio i wideo — wszystkie one są umieszczane na serwerze po kolei przez utworzone specjalnie w tym celu narzędzie. Dopiero po wysłaniu plików na serwer możesz umieścić je we wpisach swojego bloga. Dzięki takiemu rozwiązaniu nie musisz dbać o zachowanie odpowiednich powiązań odnośników po przeniesieniu plików na inny komputer. (Być może Automattic przenosi Twój blog pomiędzy serwerami, ale odbywa się to zupełnie w tle Twoich działań i nie odczuwasz przykrych skutków takiej operacji).

Z chwilą przeniesienia bloga na oprogramowanie WordPress przejmujesz wszystkie te obowiązki. Być może właściciel serwera zaoferuje odpowiednie usługi, które uchronią Cię przed większością problemów, ale nigdy nie będziesz mógł pozwolić sobie na taką beztroskę jak w czasach, gdy używałeś serwerów WordPress.com. W zamian otrzymujesz znacznie większą swobodę działań.

Obsługa HTML

Pod koniec lat dziewięćdziesiątych XX w. nastąpił gwałtowny rozwój wszelkiego rodzaju działalności gospodarczej związanej z internetem. Amerykanie określili to zjawisko mianem „dot-com-boom", co można by przełożyć jako boom-gospodarczy.com. Przedsiębiorstwa działające w branży sieciowej, które nigdy dotąd nie przyniosły ani grosza zysków, czy nawet takie, które nigdy nie wyrobiły żadnych przychodów, stały się niemal z dnia na dzień warte miliony, a czasem miliardy dolarów. Wartość tych, które przynosiły zyski, osiągnęła jeszcze wyższy poziom. Firma AOL, słynąca z oferowania milionom ludzi dostępu do internetu, wykupiła akcje potentata branży wydawniczej i filmowej, Time Warner, w roku 2000, gdy boom osiągnął szczyt. Po dziesięciu latach, jakie upłynęły od tamtego czasu, spółka, która powstała z ich połączenia, jest wyceniana na więcej bez AOL niż z nią.

W centrum tego zamieszania znajdował się właśnie HTML. Coraz mocniej i coraz częściej naciskano, by język zaczął oferować coraz większe możliwości i otwierać przed stronami WWW coraz lepsze drogi rozwoju. Firmy Microsoft (twórca przeglądarki Internet Explorer) i Netscape (twórcy przeglądarki Navigator) rozpoczęły rywalizację na gruncie wprowadzania nowych znaczników rozpoznawanych przez ich przeglądarki. (Przeglądarka Navigator dała początek popularnemu dziś programowi Firefox).

Podobnie gwałtownego rozwoju doświadczył standard CSS. Początkowo język arkuszy stylów był znacznie mniej uporządkowany i gorzej obsługiwany niż HTML. Rdzeń CSS, jaki dzisiaj znamy, miał dopiero powstać.

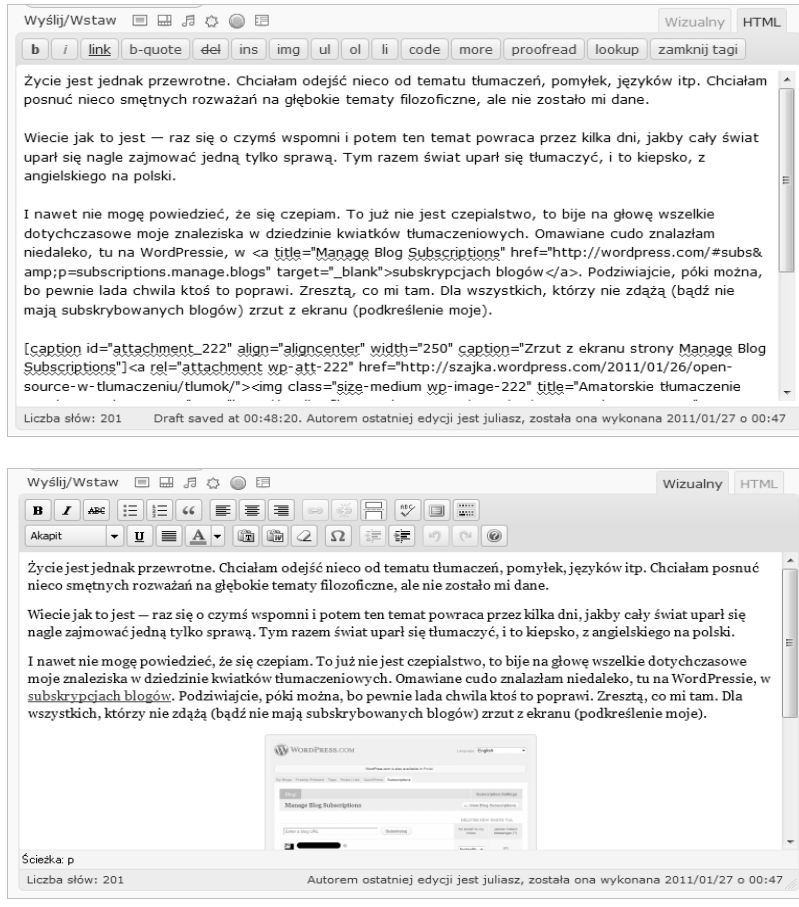
Mimo pewnego spokoju, jaki zapanował w świecie internetu, nadal na porządku dziennym są sytuacje, w których ta czy inna witryna nie działa poprawnie w niektórych przeglądarkach. Problem ten dotyczy szczególnie stron użytkowych, na przykład witryn bankowych, które bardzo często działają poprawnie wyłącznie w najnowszej wersji przeglądarki Internet Explorer w systemie Windows, przez co użytkownicy innych przeglądarek i systemów operacyjnych — m.in. Safari w systemie MAC OS czy Firefox — są w bardzo niewygodnej sytuacji.

Pomimo różnych prób wprowadzenia większych czy mniejszych porządków w standardzie HTML, doświadczeni dostawcy usług internetowych — na przykład twórcy systemu WordPress — polegają wyłącznie na najbardziej podstawowych poleceniach HTML, które są poprawnie interpretowane nie tylko przez komputery, ale także przez wszelkiego rodzaju inne urządzenia — od smartfonów do programów odczytujących zawartość okna osobom niewidomym. Blogi prowadzone w systemie WordPress są dostępne prawie dla każdego.

Karty Wizualny i HTML

Przyjrzyjmy się teraz kartom *Wizualny* i *HTML* edytora wpisów, żeby dowiedzieć się, czym naprawdę różnią się one od siebie. Rysunek 6.2 przedstawia obszar edycji wpisu w obydwu wydaniach.

Przyciski dostępne w zakładce *Wizualny* odpowiadają opcjom oferowanym przez wersję *HTML* edytora. W tabeli 6.1. znajdziesz porównanie funkcji obydwu systemów pracy. Pierwsza część tabeli zawiera przyciski umieszczone w głównym, górnym menu opcji edycji. W drugiej części znajdziesz funkcje menu podrzędnego, dolnego, które pojawia się dopiero po kliknięciu przycisku *Pokaż/ukryj zlew kuchenny*.



Rysunek 6.2. Edytor w zakładkach Wizualny i HTML wykonuje większość tych samych zadań w zupełnie odmienny sposób

Tabela 6.1. Przyciski w zakładkach Wizualny i HTML

Zakładka Wizualny	Zakładka HTML	Działanie
Dodaj obrazek	img	Wstawia grafikę.
B	b	Pogrubienie
I	i	Kursywa
~~ABC~~	~~del~~	Przekreślenie
Punkty	ul	Lista nieuporządkowana (wypunktowanie)
Numery	ol	Lista uporządkowana (numerowanie)
	li	Elementy listy (zarówno wypunktowania, jak i numerowania)
Wyrównanie do lewej		Tekst wyrównany do lewego marginesu, przy prawym marginesie poszarpany
Wyśrodkowanie		Tekst wyśrodkowany

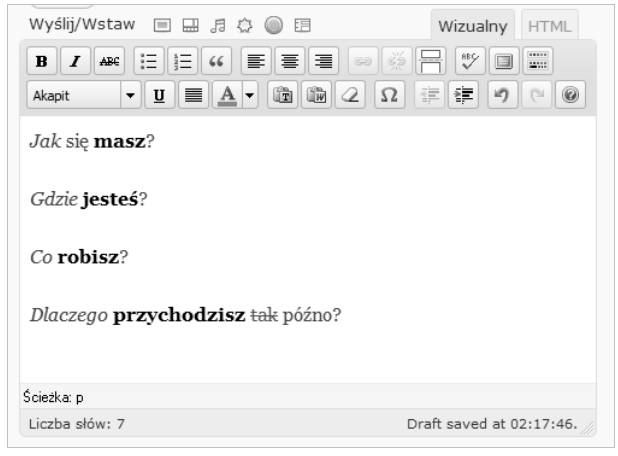
Tabela 6.1. Przyciski w zakładkach Wizualny i HTML — ciąg dalszy

Zakładka Wizualny	Zakładka HTML	Działanie
Wyrównanie do prawej		Tekst wyrównany do prawego marginesu, przy lewym marginesie poszarpany
Odnośnik	link	Hiperłącze
Usuń odnośnik		Usuwa hiperłącze
Wstaw tag „więcej"	more	Wstawia po wskazanej części tekstu odnośnik prowadzący do pozostałego fragmentu wpisu
Proofread writing (sprawdź pisownię)		Sprawdza pisownię tekstu w języku angielskim
Tryb pełnoekranowy		Wyświetla w oknie przeglądarki samo pole edycji tekstu
Pokaż lub ukryj kuchenny zlew		Wyświetla lub ukrywa dodatkowe opcje formatowania
Lista rozwijana *Akapit* (formatowanie stylu *Adres*, *Tekst preformatowany* i nagłówki 1 – 6)	code	Różne opcje formatowania tekstu
Podkreślenie	ul	
Wyjustowany		Tekst wyrównany obustronnie
Kolor tekstu		Zmienia kolor wyświetlanego tekstu (co może prowadzić do mylenia wyróżnionych fragmentów z odnośnikami)
Import tekstu		Wstawia tekst z notatnika
Import zawartości z programu Word.		Wstawia tekst z programu Microsoft Word
Usuń formatowanie		Usuwa wszystkie zmiany wyglądu tekstu nadane formatowaniem
Zmniejsz wcięcie		Usuwa wszelkie wcięcia tekstu
Zwiększ wcięcie		Przesuwa margines bloku tekstu w prawo
Cofnij		Cofa ostatnio wykonaną czynność
Ponów		Cofa cofnięcie wykonanej czynności
Pomoc		Otwiera stronę Codex (nie znajdziesz na niej zbyt szczegółowych wyjaśnień)
	ins	Nadaje formatowanie wstawionego tekstu (służy do aktualizowania wpisów)
	lookup	Sprawdza słowo w słowniku
	zamknij tagi	Wstawia zamknięcia znaczników (warto kontrolować, w jakich miejscach się pojawiają)

Formatowanie znaków

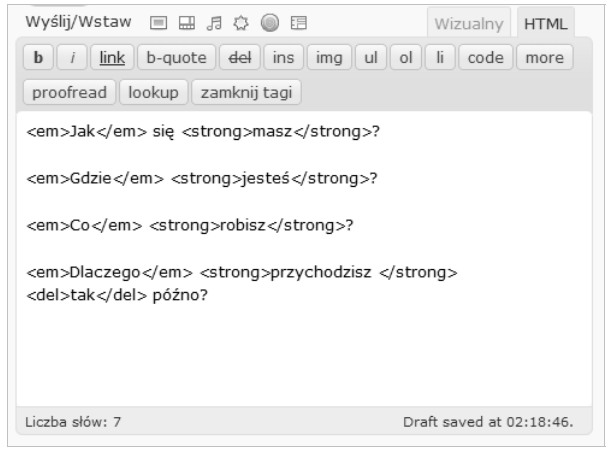
Polecenia formatowania znaków dostępne w systemie WordPress ograniczają się do najlepiej obsługiwanych przez programy znaczników kodu HTML — pogrubienia, kursywy oraz dodanego nieco później przekreślenia.

Rysunek 6.3 przedstawia tekst wpisany w zakładce *Wizualny*, sformatowany za pomocą znaczników pogrubienia, wyróżnienia i przekreślenia. Pierwsze wyrazy każdego z pytań zostały oznaczone *kursywą* — **Jak**, **Gdzie**, **Co** i **Dlaczego**. Wszystkie **czasowniki** zostały **pogrubione**. Słowo **tak** w ostatnim pytaniu zostało ~~przekreślone~~.



Rysunek 6.3. W zakładce Wizualny nie ma wątpliwości co do sposobu formatowania tekstu

Rysunek 6.4 przedstawia ten sam tekst w widoku *HTML*. Tym razem masz okazję przyjrzeć się znacznikom HTML odpowiedzialnym za wyświetlanie tekstu w oknie edytora wizualnego.



Rysunek 6.4. W zakładce HTML wszystko wygląda nieco bardziej skomplikowanie

Pora zastanowić się, co tak naprawdę dzieje się z tekstem. Tabela 6.2. przedstawia porównanie tekstu widocznego w obydwu zakładkach.

Tabela 6.2. Zakładki Wizualny i HTML

Wizualny	HTML
Jak się **masz**?	`<em>Jak</em> się <strong>masz</strong>?`
Gdzie **jesteś**?	`<em>Gdzie</em> <strong>jesteś</strong>?`
Co **robisz**?	`<em>Co</em> <strong>robisz</strong>?`
Dlaczego **przychodzisz** ~~tak~~ późno?	`<em>Dlaczego</em> <strong>przychodzisz</strong> <del>tak</del> późno?`

Wprawdzie na przyciskach zakładek *Wizualny* i *HTML* pojawiają się symbole *B* i *b*, ale po ich użyciu w kodzie pojawia się znacznik `<strong>`, a nie `<b>`. To ustępstwo na rzecz funkcjonalności — gdyby na przycisku pojawił się napis **strong**, nikt nie wiedziałby, do czego służy ten przycisk.

Podobne rozwiązanie zastosowano w przypadku znacznika `<em>`, który odpowiada za nadanie tekstowi formatowania kursywą.

Przekreślenie jest osiągane za pomocą znaczników `<del>` (od angielskiego słowa *delete* — usuwać).

Znaczniki nie są wprawdzie bardzo rozbudowane, ale kod HTML i tak jest znacznie mniej czytelny niż wersja w edytorze wizualnym. Oczywiście z czasem, gdy nabierzesz doświadczenia, nauczysz się ignorować znaczniki i bez trudu odczytasz treść strony. Najwięcej problemów w czasie odczytywania tekstu z kodu strony sprawia znacznik `<strong>`, który nie odbiega długością od przeciętnego słowa (w zasadzie jest słowem). Wygodniej byłoby posługiwać się znacznikiem `<b>`. Z kolei znacznik `<em>`, wprawdzie krótszy, nie kojarzy się z kursywą i nawet jeśli zapamięta się, że pochodzi on od angielskiego słowa oznaczającego „wyróżniać", nadal łatwo pomylić owo wyróżnienie z pogrubieniem.

Oto, jak wyglądałyby te same zadnia opisane częściej używanymi znacznikami. Łatwo stwierdzić, że mniej czysta wersja HTML jest znacznie przystępniejsza.

```
<i>Jak</i> się <b>masz</b>?
<i>Gdzie</i> <b>jesteś</b>?
<i>Co</i> <b>robisz</b>?
<i>Dlaczego</i> <b>przychodzisz</b> <del>tak</del> późno?
```

WordPress bez kłopotów interpretuje popularniejsze znaczniki, na przykład `<b>` i `<i>`, ale musisz wpisać je bezpośrednio w zakładce *HTML* lub skopiować z narzędzia, w którym tworzysz kod strony.

Jeśli sprawdzisz wygląd wpisu w zakładce *Wizualny*, a następnie powrócisz do zakładki *HTML*, odkryjesz, że zamiast wprowadzonych wcześniej znaczników `<b>` i `<i>` pojawiły się znaczniki `<strong>` i `<em>`. Mniej popularne znaczniki, na przykład `<s>`, oznaczające przekreślenie, zostaną w czasie tych działań po prostu usunięte, więc będziesz musiał zauważyć ich brak i wprowadzić je ponownie ręcznie w formie akceptowanej przez WordPress w zakładce *HTML* lub posłużyć się odpowiednim przyciskiem w zakładce *Wizualny*.

Tajemniczy znacznik

Oznaczenie akapitu, czyli znacznik [4], jest jedną z najczęstszych przyczyn frustracji w czasie edycji kodu HTML.

Pierwotnie był on narzędziem formatowania tekstu, a nie określania jego znaczenia. Używano go, by oddzielić jeden akapit od drugiego.

Większość przeglądarek internetowych oddziela akapity od siebie, umieszczając przed kolejnym nieco wolnego miejsca. Projektanci witryn zazwyczaj chcą się go pozbyć lub pragną, by było go znacznie więcej.

Jeżeli nie chcesz oddzielać kolejnych wierszy tekstu odstępem, powinieneś raczej korzystać ze znacznika <br>, wymuszając tym samym złamanie wiersza. Jeśli jednak zależy Ci na powiększeniu odstępów między akapitami, to masz problem. Większość przeglądarek została zaprojektowana w taki sposób, by ignorować wielokrotne wystąpienia znaczników , <br> lub ich kombinacji. Powiększanie odstępów między akapitami za pomocą samego kodu HTML nie jest łatwe.

Z czasem sposób korzystania ze znacznika zmienił się. Projektanci stron zaczęli stosować go w charakterze kontenera tekstu, otaczając treść akapitu znacznikami otwarcia i zamknięcia . To znacznie lepsze podejście do kodowania, ponieważ informuje wszystkich, którzy czytają kod, że „to zamknięty akapit". Umieszczenie samego znacznika jest równoznaczne ze stwierdzeniem „wstaw tu przerwę akapitową".

Jeżeli poza używaniem kodu HTML posiłkujesz się kodem CSS, powinieneś raczej stosować znaczniki sygnalizujące podział zawartości strony na części, których szczegółowa charakterystyka jest zdefiniowana w arkuszu stylów. Takie rozwiązanie znacznie zwiększa pole manewru, lecz jednocześnie zaciemnia kod, ponieważ aby zorientować się, za co odpowiada każdy ze znaczników , trzeba sięgnąć do arkusza stylów. Nie zapominajmy też, że jeden styl może przesłaniać inne — dlatego nazywa się je kaskadowymi — więc niewykluczone, że zanim dotrzesz do sedna zastosowanego rozwiązania, będziesz musiał sprawdzić niejeden plik.

System WordPress ukrywa znaczniki w trybie HTML. Najwyraźniej jego twórcy doszli do wniosku, że i bez nich kod HTML jest wystarczająco skomplikowany!

Teraz już wiesz, jak trudno jest okiełznać kod HTML i pewnie rozumiesz frustrację, jaka od lat narasta u niemal każdego projektanta witryn WWW. Na pewno nieraz natknąłeś się w sieci na pełne goryczy opowieści o zmaganiach z kodem CSS czy HTML i jego wyświetlaniem w różnych przeglądarkach — częstej przyczynie frustracji programistów aplikacji internetowych.

Formatowanie list

Warto przez chwilę zatrzymać się przy temacie stosowania wypunktowania i numerowania w systemie WordPress, ponieważ obydwa te rodzaje formatowania są bardzo wygodne (zarówno podczas pracy nad blogiem, jak i podczas czytania), ale jednocześnie mogą wywoływać wiele negatywnych emocji. Przeglądarki wyświetlają je w wyjątkowo

[4] Od angielskiego słowa *paragraph* — akapit — *przyp. tłum.*

nieprzyjemny sposób — z ogromnym odstępem przed pierwszym elementem listy i bez żadnych odstępów między pozostałymi elementami.

Wirtuozi HTML nie od dziś stosują przeróżne rozwiązania, dzięki którym listy prezentują się znacznie lepiej, natomiast specjaliści od CSS wprowadzają odpowiednie i akceptowalne zmiany w arkuszach, by osiągnąć zamierzony efekt.

Kod HTML oferuje dwa modele list. Każdy z nich jest otoczony inną parą znaczników, ale poszczególne pozycje w obydwu przypadkach ujmuje się w identyczne znaczniki. Dzięki temu zmienianie typów formatowania listy nie nastręcza żadnych trudności. Poniżej kilka słów na temat każdego z nich:

- ***Lista nieuporządkowana*** — w tym przypadku **nieuporządkowana** oznacza „listę, w której kolejność wymieniania elementów nie ma żadnego znaczenia", choć oczywiście muszą one być wyświetlone w jakimś porządku — HTML podaje je zawsze w kolejności, w jakiej je wpiszesz. Listy nieuporządkowane przyjmują zawsze postać wypunktowania. Określanie ich mianem **nieuporządkowanych** zamiast **wypunktowaniem** to ukłon w stronę idei kryjącej się za językiem HTML — kładzeniem nacisku na znaczenie, a nie na szczegóły formatowania.
- ***Lista uporządkowana*** — **uporządkowana**, czyli taka, na której „kolejność elementów ma znaczenie", zatem jest zawsze wyświetlana w postaci listy numerowanej.

Wszystkie pozycje listy nieuporządkowanej obejmuje się zewnętrzną parą znaczników `<ul></ul>`[5], a każdy z elementów listy jest objęty osobną parą znaczników `<li></li>`. Oznaczenie to jest skrótem angielskiego wyrażenia **list item** — element listy.

W przypadku listy uporządkowanej (numerowanej) stosuje się nadrzędną parę znaczników `<ol></ol>`[6], a każdą z pozycji listy obejmuje się osobną parą znaczników `<li></li>`, tak jak w przypadku listy nieuporządkowanej.

Tabela 6.3. przedstawia wygląd list w kartach *Wizualny* i *HTML* edytora wpisów.

Zwróć uwagę na to, jak bardzo podobnie wygląda formatowanie list w obydwu przypadkach. Różnią się jedynie znacznikami otwierającymi i zamykającymi — w pierwszym przypadku jest to para `<ul></ul>`, w drugim `<ol></ol>`. Gdyby zaszła potrzeba edytowania kodu HTML i zmienienia jednego sposobu formatowania na drugi, wystarczy zamienić znaczniki zewnętrzne.

Zauważ też, że w kolumnie *HTML* tabeli 6.3. wszystkie elementy obydwu list są oznaczone dodatkowymi wcięciami. Wcięcia te nie mają nic wspólnego ze sposobem, w jaki kod HTML jest wyświetlany przez przeglądarki. Taki zapis ma ułatwić odczytywanie kodu programiście, niezależnie od tego, czy będzie się sprawdzać polecenia na ekranie, czy na papierze. Niektóre z edytorów HTML wprowadzają dodatkowe ułatwienia w postaci wielopoziomowych wcięć, wyróżniania poszczególnych elementów składni języka kolorami, inną czcionką czy stylem formatowania (pogrubieniem, kursywą itd.), a nie są to jedyne sztuczki mające na celu podniesienie komfortu pracy z kodem.

---

[5] Nazwa znacznika pochodzi od pierwszych liter *unordered list* — lista nieuporządkowana — *przyp. tłum.*

[6] Nazwa znacznika jest skrótem wyrażenia *ordered list* — lista uporządkowana — *przyp. tłum.*

Tabela 6.3. Wygląd list w trybach pracy Wizualny i HTML edytora wpisów

Wizualny	HTML
Oto kilka moich ulubionych cytatów z Mickiewicza:	Oto kilka moich ulubionych cytatów z Mickiewicza:
Zostać na polu samemu i w nocy, / To lubię, rzekłem, to lubię!	Zostać na polu samemu i w nocy, / To lubię, rzekłem, to lubię!
Jakiż to chłopiec piękny i młody? / Jakaż to obok dziewica? / Brzegami sinej Świtezi wody / Idą przy świetle księżyca.	Jakiż to chłopiec piękny i młody? / Jakaż to obok dziewica? / Brzegami sinej Świtezi wody / Idą przy świetle księżyca.
Ja na rok u Belzebuba / Przyjmę za ciebie mieszkanie, / Niech przez ten rok moja luba / Z tobą jak z mężem zostanie.	Ja na rok u Belzebuba / Przyjmę za ciebie mieszkanie, / Niech przez ten rok moja luba / Z tobą jak z mężem zostanie.
Oto kilka moich ulubionych cytatów z Mickiewicza:	Oto kilka moich ulubionych cytatów z Mickiewicza:
Zostać na polu samemu i w nocy, / To lubię, rzekłem, to lubię!	Zostać na polu samemu i w nocy, / To lubię, rzekłem, to lubię!
Jakiż to chłopiec piękny i młody? / Jakaż to obok dziewica? / Brzegami sinej Świtezi wody / Idą przy świetle księżyca.	Jakiż to chłopiec piękny i młody? / Jakaż to obok dziewica? / Brzegami sinej Świtezi wody / Idą przy świetle księżyca.
Ja na rok u Belzebuba / Przyjmę za ciebie mieszkanie, / Niech przez ten rok moja luba / Z tobą jak z mężem zostanie.	Ja na rok u Belzebuba / Przyjmę za ciebie mieszkanie, / Niech przez ten rok moja luba / Z tobą jak z mężem zostanie.

Uwaga
Byłoby miło, gdyby Automattic zdecydowała się na wprowadzenie edytora kodu HTML również w trybie pełnoekranowym, tak wygodnym w czasie pracy w edytorze wizualnym. Dopóki to nie nastąpi, jesteś skazany na szukanie rozwiązań prowizorycznych.

Nawet na tym prostym przykładzie widać, że kod HTML zajmuje sporo miejsca na stronie. W czasie pracy na ekranie komputera sytuacja nie przedstawia się o wiele lepiej. Edycja treści wpisu w widoku kodu HTML jest wyjątkowo niewygodna z powodu braku miejsca w polu tekstowym edytora, który jest dostępny wyłącznie w widoku kokpitu.

Tworzenie odnośników

Tworzenie odnośników to definiowanie hiperłączy wewnątrz kodu HTML. Hiperłącza stanowią serce struktury internetu.

Określanie łączy może być trudne, do tego łatwo popełnić w czasie takiej operacji błąd, czy nawet uszkodzić łącze, które działało już od jakiegoś czasu. Na szczęście budowanie połączeń w systemie WordPress zostało uproszczone tak, by ułatwić Ci pracę, nie pozbawiając jednocześnie możliwości decydowania o pewnych opcjach działania odnośnika. Gdybyś zaś potrzebował jakichś specjalnych rozwiązań, zawsze możesz edytować kod wpisu i dodać kolejne parametry do definicji łącza.

Głównym powodem, dla którego powinieneś starać się zrozumieć strukturę odnośnika, jest to, że właśnie budowa łącza i możliwości jej modyfikacji będą skłaniać Cię najczęściej do analizowania kodu HTML. Od jakości połączeń zależy natomiast komfort pracy i poruszania się po blogach systemów WordPress.com i WordPress.org. Łącza bywają niezbędne w dwóch typowych dla autorów blogów przypadkach:

- gdy na interesującej Cię stronie znajdziesz kod HTML, który pozwoliłby Ci umieścić na blogu na przykład plik wideo lub przycisk pozwalający nawiązać połączenie telefoniczne (Google Voice) czy uruchomić funkcję tłumaczenia (AltaVista);
- gdy chcesz wykorzystać wszystkie możliwości widgetu *Tekst*, do czego wrócimy w dalszej części rozdziału.

Oto trzy rodzaje odnośników, które warto zapamiętać:

- ***Odnośniki do strony głównej, np.:***

  Odwiedź `<a href="http://www.wordpress.com">serwis WordPress</a>`.

  W tym przypadku kod HTML wykonuje naprawdę istotną pracę. Gdy w odnośniku pojawia się nazwa domeny, przeglądarka zaczyna szukać jednej z następujących stron:

  ```
  index.htm
  index.html
  home.htm
  home.html
  ```

  Nazwy te są uprzywilejowane, co oznacza, że nie musisz wpisywać ich jawnie w pasku przeglądarki, by ta zaczęła szukać ich w sieci. Nie musisz wiedzieć, którą z nazw posłużył się autor witryny, ponieważ każda z nich otworzy się po podaniu nazwy domeny.

- ***Odnośniki do dowolnej strony w sieci, np.:***

  Sprawdź listę najnowszych ciekawostek w systemie
  `<a href="http://www.wordpress.com/ciekawostki/najnowsze.htm">WordPress</a>`.

  Taki odnośnik zaprowadzi Cię do zawartości katalogu *ciekawostki*, otwierając stronę *najnowsze.html* (nazwy plików stron internetowych zawsze kończą się rozszerzeniem *.htm* lub *.html*).

- ***Odnośniki przenoszące w obrębie jednej strony, np.:***

  Sprawdź listę najnowszych ciekawostek w systemie
  `<a href="http://www.wordpress.com/ciekawostki/najnowsze.htm#item20">WordPress</a>`.

Polecenie to, jak poprzednie, przeniesie Cię na stronę *najnowsze.htm* umieszczoną w katalogu *ciekawostki*, ale idzie ono jeszcze o krok dalej i otwiera stronę w miejscu oznaczonym specjalnym znacznikiem, który nazywamy **kotwicą** (`anchor`). (Jeśli z jakichś przyczyn kotwica zostanie usunięta z pliku strony, ta otworzy się na samym początku). Oto postać kotwicy:

```
<a name="item20"><h4>Element 20</h4></a>
```

W przedstawionym przykładzie kotwica obejmuje nagłówek, co akurat jest bardzo typowym sposobem wykorzystania tego znacznika. Tego rodzaju kotwica, czyli para znaczników `<a></a>`, nie musi otaczać żadnego tekstu, ponieważ samo jej wystąpienie wystarczy do oznaczenia wybranego punktu pliku. Jednak niektórzy lubią umieszczać wewnątrz tekst na wypadek późniejszych prób przemodelowania zawartości pliku — dobrze jest umieścić wewnątrz kotwicy coś, co wskaże wyraźnie jej przeznaczenie.

Ostrzeżenie

Strony WWW i zdefiniowane w nich kotwice bez przerwy ulegają zmianom. Z kolei wpisy na blogu są poniekąd związane z okresem, w jakim powstały, więc raczej nie wraca się do nich, by nanosić poprawki i upewniać się, że wszystkie odnośniki działają prawidłowo. Jeśli jednak umieściłeś na swojej witrynie statyczne strony lub polecasz ulubione wpisy, powinieneś co jakiś czas sprawdzać, czy wszystkie odnośniki działają poprawnie, i uaktualniać te, które przestały funkcjonować w wyniku zmian na stronach docelowych.

Tego rodzaju odnośniki przydadzą Ci się niejednokrotnie podczas pracy z widgetem *Tekst*, o którym wspomnimy w dalszej części tego rozdziału. Jeżeli chcesz umieszczać odnośniki do konkretnych punktów własnych wpisów lub stron, możesz umieścić w ich kodzie odpowiednie kotwice. Możesz też sprawdzić, czy w kodzie stron, które chciałbyś polecić, nie znajdziesz przygotowanych przez autora kotwic.

Zarządzanie kotwicami

Technicznie rzecz ujmując, zarówno początek odnośnika, jak i miejsce, do którego on prowadzi, nazywają się kotwicami. Kotwica opatrzona nazwą i umieszczona wewnątrz pliku, na przykład `item20`, o którym wspominaliśmy w tym rozdziale, to po prostu pewien szczególny rodzaj odnośnika.

Jednakże większość ludzi, w tym także twórcy kodu HTML — osoby wyjątkowo zważające na swoje słowa — nadal określają początek odnośnika słowami **link** lub **href**. Określenie **anchor** (kotwica) kojarzy się przede wszystkim z odnośnikami prowadzącymi do zawartości tego samego pliku.

Dlatego jeśli kiedyś natkniesz się na określenie „kotwica", pamiętaj, że zapewne mowa o odnośniku prowadzącym do zawartości tej samej strony. No, chyba że zabłąkasz się do portu, wtedy kotwicę rzuć!

Znacznik more

Twórcy systemu WordPress wprowadzili własny znacznik, `more`, który jest doskonałym przykładem sposobu działania kodu HTML oraz metodyki działania pracowników Automattic, za wszelką cenę starających się ułatwić Ci życie.

Na stronie głównej Twojego bloga pojawiają się najbardziej aktualne wpisy. Jeżeli umieścisz we wpisie znacznik `more`, tylko część wpisu będzie widoczna na stronie głównej. Tekst ukryty za tym znacznikiem pojawi się dopiero wtedy, gdy użytkownik kliknie tekst odnośnika, czy to z poziomu strony głównej, czy z dowolnego innego miejsca.

Znacznik `more` wprowadza się do kodu następująco:

```
<!--more-->
```

Z punktu widzenia programów interpretujących kod HTML każdy ciąg znaków wewnątrz znacznika rozpoczynający się od wykrzyknika (`!`) jest komentarzem. Przeglądarki ignorują tego rodzaju znaczniki, dlatego programiści z upodobaniem umieszczają w nich komentarze do kodu. Takie uwagi odgrywają rolę notatek dla kierowników projektów lub ludzi, którzy w późniejszym terminie zajmują się wprowadzaniem na stronę nowych treści i poprawek.

Oczywiście nie zapominajmy, że WordPress nie jest jakąś tam stroną WWW. Tak złożony system może zawierać polecenia nakazujące mu sprawdzać treść komentarzy i pilnować, czy pojawiają się w nich zdefiniowane ciągi znaków, którym przypisano konkretne znaczenie. Znacznik `more` jest właśnie przykładem takiego wyspecjalizowanego komentarza.

Znacznik ten może pojawić się w dowolnym miejscu wpisu, dzięki czemu strona główna Twojego bloga będzie wyglądała dokładnie tak, jak sobie tego zażyczysz.

> **Wskazówka**
> Wielu użytkowników spodziewa się, że po wejściu na stronę główną bloga zastaną tam całą treść najnowszego wpisu. Możesz zatem publikować najnowsze notki bez wstawiania w nich znacznika `more`, natomiast dodawać go do wszystkich poprzednich. Dzięki temu najnowszy post pojawi się na głównej stronie w całości, a kilka innych wpisów (ile — to już zależy od Ciebie) wyświetli się wyłącznie w postaci fragmentów.

Edycja kodu HTML w systemie WordPress

Okno wpisów sprawdza się całkiem nieźle w roli edytora WYSIWYG i jest totalnie chybionym edytorem kodu HTML. Oto dlaczego:

- Edytor WYSIWYG (czyli karta *Wizualny*) pozwala korzystać ze wszystkich funkcji HTML, umożliwia kopiowanie z notatnika i programu Word, oferuje funkcję sprawdzania tekstów w języku angielskim i pozwala pracować w trybie pełnoekranowym.

- Przyciski funkcji w edytorze HTML są mylące i nie ma możliwości rozwinięcia pola edycji do trybu pełnoekranowego. Kod HTML zajmuje bardzo dużo miejsca pola edycji, a ograniczona ilość miejsca w widoku kokpitu uniemożliwia dobrą orientację w tekście. Dodatkowym utrudnieniem jest brak funkcji *Znajdź* i *Zamień*.

Ze względu na ograniczoną funkcjonalność edytora HTML w systemie WordPress, wszelkie próby pracy z bardziej zaawansowanymi postaciami kodu są w nim z góry skazane na porażkę. Jeżeli uznasz, że Twój blog nie obejdzie się bez znacznej ingerencji w kod, powinieneś przeprowadzić wszelkie potrzebne zmiany w innym edytorze, skopiować wyniki pracy i wkleić je w polu edycji kodu WordPress.

Oczywiście w trakcie przełączania się pomiędzy oknami edytorów bardzo łatwo jest pogubić się, utracić wyniki pracy, a przede wszystkim nie da się przy tym uniknąć straty mnóstwa czasu. Na szczęście tego typu operacje zwykle kończą się pomyślnie.

Użytkownicy wszystkich systemów operacyjnych mogą wybierać spośród wielu różnych edytorów kodu HTML — w tej dziedzinie nie obowiązują żadne standardy. Wraz z eksplozją popularności sieci w latach dziewięćdziesiątych XX w. na rynku pojawiło się mnóstwo tego typu aplikacji. Niektóre cieszące się niegdyś powodzeniem programy zniknęły w tym czasie z horyzontu, inne są wprawdzie jeszcze wspierane przez producentów, ale już nierozwijane. Ta sytuacja zmienia się z roku na rok. Doskonałym przykładem takiego traktowania może być popularny niegdyś edytor FrontPage autorstwa Microsoft. Program został **porzucony** — nie można go już kupić, nie pojawiają się żadne aktualizacje. Microsoft wprowadził na jego miejsce znacznie bardziej zaawansowane technicznie programy należące do rodziny Microsoft Expression Studio. Pakiet ten ma konkurować z powszechnie szanowanym, lecz niemniej skomplikowanym w obsłudze programem Adobe Dreamweaver.

Poszukaj w internecie programu odpowiedniego dla swoich potrzeb i właściwego dla systemu, z jakiego korzystasz. Większość edytorów jest dostępna w podstawowej wersji za darmo lub przez okres próbny. Potem, gdy będziesz mieć pewność, że dana aplikacja Ci odpowiada, bez trudu zaktualizujesz ją do pełnej wersji.

Z jakich znaczników korzystać?

Jeśli masz jakiekolwiek rozeznanie w pracy z kodem HTML, zastanawiasz się pewnie teraz, których znaczników będzie Ci wolno używać w czasie tworzenia wpisów w systemie WordPress. Jeśli zaś masz dopiero w planach podszkolenie się nieco w tym temacie, pytanie to zadasz sobie już w najbliższej przyszłości.

Wspominaliśmy już o kilku rodzajach znaczników, które na pewno sprawdzają się we wpisach WordPress (choć system czasami zamienia je na inne znaczniki). Są to: `a`, `b`, `blockquote`, `br`, `del`, `div`, `em`, `li`, `ol`, `p`, `s`, `span`, `strike`, `strong` i `ul`.

Poza tym możesz bez problemu używać kilkunastu innych, które tu wymienimy i opiszemy w kilku słowach — `font` (owszem, możesz zmieniać czcionki w swoim blogu!), `h1`, `h2`, `h3`, `h4`, `h5`, `h6` (nagłówki uszeregowane od największego do najmniejszego), `hr` (pozioma linia

oddzielająca), `img` (wstawianie obrazów rządzi się podobnymi zasadami jak budowanie odnośników) i `table` (potężne, lecz niebezpieczne narzędzie — tabele działają w każdej przeglądarce nieco inaczej).

I, żeby mieć formalności za sobą, oto lista pozostałych rozpoznawanych przez WordPress znaczników (niektóre z nich zależą od wystąpienia wcześniej znacznika `table`): `address`, `abbr`, `acronym`, `big`, `caption`, `cite`, `class`, `code`, `col`, `dd`, `dl`, `dt`, `i`, `ins`, `kbd`, `pre`, `q`, `sub`, `sup`, `tbody`, `td`, `tfoot`, `tr`, `tt` i `var`.

Wspominaliśmy też, że WordPress rozpoznaje znaczniki XHTML. Jak odczuje to osoba pracująca bezpośrednio z kodem HTML?

Niezbyt gwałtownie, ponieważ większość skomplikowanych zmian związanych z wprowadzeniem standardu XHTML pojawia się na poziomie ogólnej budowy pliku strony i jego głównych części (`head`, `body` itp.), a tymi zajmuje się przecież WordPress.

Oto kilka podstawowych zasad obowiązujących podczas pracy z językiem XHTML:

- ***Wszystkie znaczniki HTML zapisuj małymi literami*** — dawniej w ten sposób kod HTML zapisywali tylko zbuntowani nastolatkowie, dziś, w XHTML, taki zapis to standard.
- ***Zagnieżdżaj poprawnie znaczniki*** — jeśli wyróżniasz jakiś fragment kodu pogrubieniem za pomocą znaczników `<b>` i `</b>` i jednocześnie przyjmujesz, że początek tego słowa będzie początkiem odnośnika, pamiętaj, żeby wewnątrz znaczników `<a></a>` znalazły się obydwa znaczniki `<b>`, a nie tylko jeden z nich.
- ***Używaj zawsze pary znaczników*** — chodzi tu w szczególności o posługiwanie się parą znaczników `<p>` i `</p>`, a nie samym `<p>`.
- ***Wartości atrybutów umieszczaj zawsze w znaku cudzysłowu*** — dawniej zastanawiałeś się pewnie wiele razy, czy przeglądarka nie jest zbyt głupia, by odróżnić atrybut od jego argumentu. XHTML przynosi oficjalną odpowiedź — „jest zbyt głupia". Dlatego odnośniki powinny przyjmować postać jak w poniższym przykładzie:

  Więcej szczegółów znajdziesz na stronie `<a href="www.metaverse.wordpress.com">`
  `↪Metaverse</a>`.

Jeżeli będziesz przestrzegać tych prostych zasad, Twój kod zadziała tak samo dobrze w zwykłym środowisku HTML i w środowisku XHTML systemu WordPress.

Oczywiście są też odpowiednie narzędzia programistyczne i źródła informacji przeznaczone dla osób zainteresowanych językami HTML i XHTML. Niektóre z dostępnych w sieci są już nieaktualne, jedne podają zbyt mało szczegółów technicznych, inne zbyt wiele, ale na pewno prędzej czy później znajdziesz odpowiednie dla siebie. Wpisz w wyszukiwarce nazwę interesującego Cię znacznika wraz ze słowami kluczowymi „HTML znaczniki", a zobaczysz, ile ciekawych stron pojawi się w wynikach. Znalezienie takiej, która przypadnie Ci do gustu, to tylko kwestia czasu.

Ze względów bezpieczeństwa WordPress nie dopuszcza używania następujących znaczników: `embed`, `frame`, `iframe`, `form`, `input`, `object` i `textarea`.

Film z serwisu YouTube — przykład zagnieżdżania kodu HTML

Usługi świadczone przez serwis YouTube doskonale nadają się do zilustrowania przykładu procedury umieszczania obcego kodu HTML wewnątrz Twojej strony, a w przypadku bloga WordPress — wewnątrz wpisu.

Łączenie usług oferowanych przez YouTube z wpisami blogów WordPress to doskonały pomysł. Umieszczanie pliku wideo na stronie sprowadza się, w przypadku YouTube, wyłącznie do wstawienia w odpowiednim miejscu fragmentu kodu przygotowanego przez ten serwis. Sam plik wideo pozostaje na serwerach YouTube i nigdy nie zostaje przeniesiony na serwer WordPress.

W chwili gdy osoba odwiedzająca Twój blog kliknie przycisk odtwarzania, zostaje uruchomiona procedura przesyłania strumienia danych pomiędzy serwerem YouTube a komputerem odwiedzającego. Twój blog w ogóle nie bierze bezpośredniego udziału w tym zdarzeniu.

Najprzyjemniejszą częścią tego rozwiązania jest fakt, że nie ponosisz w związku z tym żadnych kosztów. Przed nastaniem ery serwisu YouTube niemal każdy, kto chciał opublikować w sieci film, musiał pokryć niemałe koszty przesyłania strumienia danych — opłaty były dość wysokie, ponieważ przesyłano naprawdę duże ilości danych, ale również z powodu potrzeby wprowadzenia dodatkowych środków kontroli kodu i dzielenia łącza, które zapewniłyby godziwą jakość przesyłu. Jeśli zaś chciałeś skorzystać z usługi kompresji wideo, cena rosła jeszcze bardziej. Firma Google, właściciel serwisu YouTube, pokrywa wszystkie koszty za Ciebie. (I w chwili pisania tej książki ciągle traci na swojej inwestycji, ale to nie nasz problem).

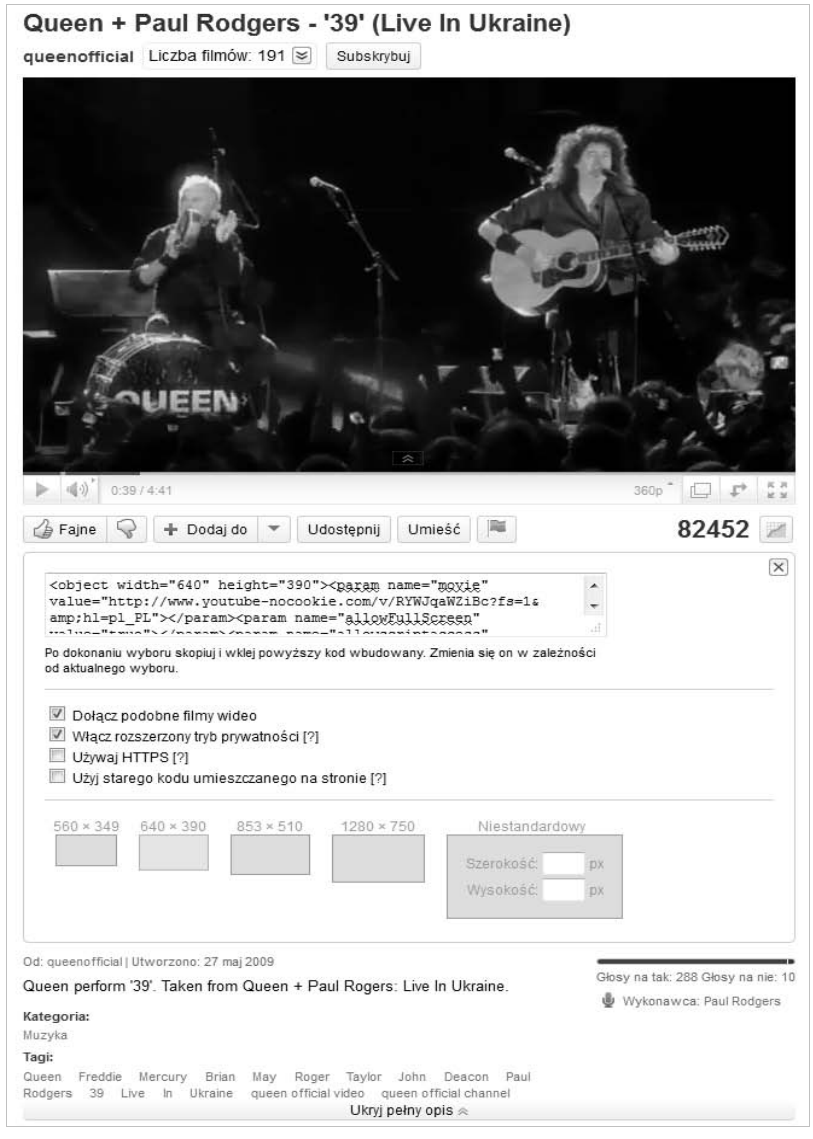
Oczywiście odwiedzający będzie wiedział, że korzysta z darmowej usługi YouTube, ponieważ każdy wyświetlany za pośrednictwem ich serwerów film jest opatrzony solidnych rozmiarów logiem z napisem **YouTube**, lecz nieszkodliwa w sumie reklama jest chyba uczciwą ceną za możliwości, jakie zyskujesz. A te nie są małe — otrzymujesz narzędzie, które pozwala dzielić się z odwiedzającymi wiedzą i rozrywką oraz czerpać inne korzyści z umieszczania na blogu filmów, a dodatkowo zdejmuje z Ciebie przykre obowiązki i nie wymaga ponoszenia kosztów.

Aby wstawić do wpisu film z serwisu YouTube, możesz wprowadzić odpowiedni kod HTML ręcznie lub użyć nowego rozwiązania o nazwie oEmbed (które samo zagnieżdża kod wewnątrz Twojej strony). Funkcja oEmbed wymaga jedynie podania adresu URL filmu, ale warto pamiętać, że ręczne wstawianie kodu HTML daje większą kontrolę nad sposobem umieszczenia filmu na stronie. Dlatego warto zapoznać się z obydwiema metodami.

Oto instrukcje umieszczania we wpisie bloga kodu HTML prowadzącego do filmu znajdującego się na serwerze YouTube. (Metoda ta sprawdzi się także w czasie pracy z innymi partiami kodu, które musiałbyś wkleić do kodu swojego wpisu).

1. Jeśli masz już przygotowaną treść wpisu, w którym chcesz umieścić film, zapisz szkic notatki i podejrzyj jej kształt na blogu, by upewnić się, że przed wstawieniem kodu filmu nie pojawiły się żadne problemy.

2. Otwórz stronę serwisu YouTube (*www.youtube.com*). Instrukcje umieszczania filmu na ich serwerach znajdziesz w rozdziale 9. „Grafika we wpisach".
3. Znajdź film, który chcesz umieścić na blogu, lub otwórz stronę z dodanym właśnie materiałem.
4. Pod oknem filmu znajdziesz kilka przycisków. Ostatni z nich to *Umieść*. Po kliknięciu go otwiera się ramka z kodem, który będziesz musiał umieścić na stronie, i narzędziami ustawiania opcji wyświetlania filmu.
5. Ustaw opcje osadzania filmu. Przykład ustawień zaprezentowano na rysunku 6.5, przedstawiającym materiał z oficjalnego kanału zespołu Queen.



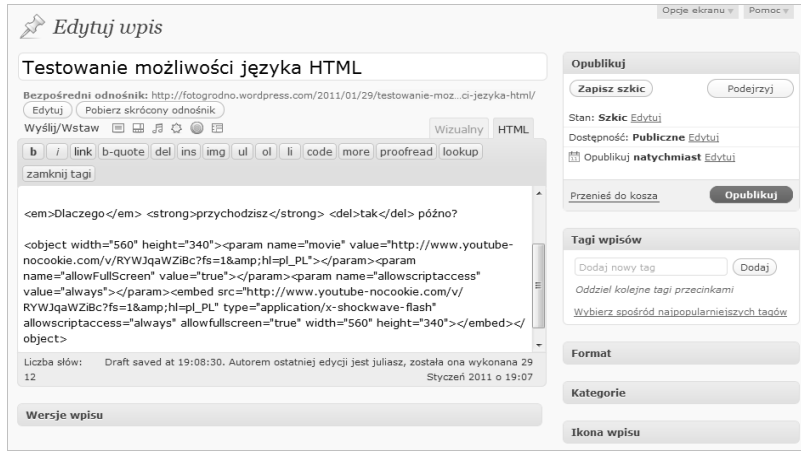
Rysunek 6.5. Trochę dobrej muzyki w czasie lektury nie zaszkodzi

6. Włącz lub wyłącz opcję dołączania podobnych filmów, zmień ustawienia prywatności, jeśli zachodzi taka potrzeba, a także rozważ możliwość skorzystania ze starszej wersji kodu HTML, by umieścić film na swojej stronie.

   Każda zmiana ustawień wpływa na dane umieszczane w osadzanym kodzie. Spróbuj zmienić wielkość okna filmu, który chcesz osadzić, i zobacz, jak wpłynie to na wartości atrybutów `height` i `width` podane w kodzie (widać je na samym początku kodu). (Zauważ też zgodne ze specyfikacją XHTML znaki cudzysłowu, w których podano obydwa parametry).

   Chociaż nie jesteśmy wrogami osadzania filmów na blogu, odradzamy Ci jednak korzystanie z funkcji dołączania podobnych filmów wideo. Natomiast zalecamy używanie opcji *Włącz rozszerzony tryb prywatności* (dzięki czemu pliki cookie zostają utworzone wyłącznie u tych odwiedzających, którzy zaczną odtwarzać film, a nie u tych, którzy jedynie zapoznają się z treścią wpisu). Dobierz jeszcze wymiary okna z filmem tak, by pasowały do układu Twojego bloga — i gotowe.

7. Kliknij w obszarze pola z kodem do osadzenia na stronie i skopiuj jego zawartość.
8. Otwórz stronę edycji wpisu bloga *WordPress* i przełącz widok do zakładki *HTML*.
9. Przewiń kod wpisu, by znaleźć miejsce, w którym chcesz wstawić okno filmu. W przypadku długich wpisów to zajęcie staje się wyjątkowo żmudne. Nie wykluczamy, że będziesz musiał skopiować kod HTML do innego edytora i skorzystać z funkcji wyszukiwania, by odnaleźć odpowiedni fragment. Później znajdź ten sam fragment wpisu na stronie edycji i ustaw tam kursor edytora.
10. Wklej kod HTML ze strony serwisu YouTube, jak pokazano to na rysunku 6.6.



Rysunek 6.6. Praca z kodem HTML nie daje wiele radości — w odróżnieniu od jej wyników

> **Uwaga**
> W rozdziale 9. omówimy inne możliwości wstawiania plików wideo do wpisów WordPress.

11. Podejrzyj kształt wpisu, by upewnić się, że właściwie dobrałeś ustawienia wyświetlania pliku wideo.

12. W razie potrzeby wróć na stronę YouTube i zmień ustawienia, powtarzając kroki 6. – 11. tak długo, aż osiągniesz zadowalające Cię efekty.

Gdy skończysz, wpis będzie wyglądał tak, jak chcesz i zawierał dokładnie to, co chciałeś w nim umieścić.

Ostrzeżenie

Regulamin korzystania z serwisu WordPress zabrania prowadzenia na blogach działalności przynoszącej zyski, a wiele działań mających generować przychód wiąże się z umieszczeniem na stronie WWW odpowiednio przygotowanego fragmentu kodu. Nie daj się skusić!

Wprawdzie niektóre z poczynań na pograniczu łamania regulaminu mogą uchodzić Ci przez jakiś czas na sucho, ale społeczność WordPress nie będzie zadowolona. Prędzej czy później poprosisz kogoś o pomoc, być może Twoja strona trafi do odnośników czyjegoś bloga albo zechcesz dowiedzieć się, w jaki sposób wprowadzić jakieś ulepszenia, albo sam dodasz czyjś blog do swoich odnośników, i wtedy wszystko się wyda. Być może wywoła to nawet reakcję pracowników Automattic, którzy wezwą Cię do zaprzestania tego rodzaju działalności albo po prostu usuną niezgodną z regulaminem treść z Twojego bloga.

Oczywiście użytkowników oprogramowania WordPress pobieranego z witryny WordPress.org nie obowiązują żadne ograniczenia dotyczące zarabiania pieniędzy na prowadzonym blogu. Jeśli chcesz wprowadzić na stronę na przykład reklamę, zacznij prowadzenie bloga na platformie WordPress.org. Jeżeli nadal nie jesteś gotów na taki krok, pozostań na WordPress.com, ucz się, jak prowadzić schludny i ciekawy blog i dopiero gdy będziesz gotów, przejdź na własny serwer. O tym, jak to zrobić, dowiesz się z rozdziału 11.

Sposoby wykorzystywania widgetu Tekst

W rozdziale 3. opisaliśmy szczegółowo widgety, z których możesz korzystać na platformie WordPress.com. Wyjątek uczyniliśmy dla dwóch najważniejszych, wymagających bardziej szczegółowego omówienia. Ten podrozdział jest poświęcony widgetowi *Tekst*. W rozdziale 8. znajdziesz informacje na temat możliwości widgetu *RSS*.

Opis widgetu tekstowego umieszczony na stronie wsparcia technicznego WordPress.com wymienia trzy podstawowe cechy, które zdecydowały o jego niesłabnącej popularności:

- niemal nieograniczone zastosowania,
- elastyczność,
- możliwość umieszczania wielu widgetów tego typu w obrębie jednego bloga.

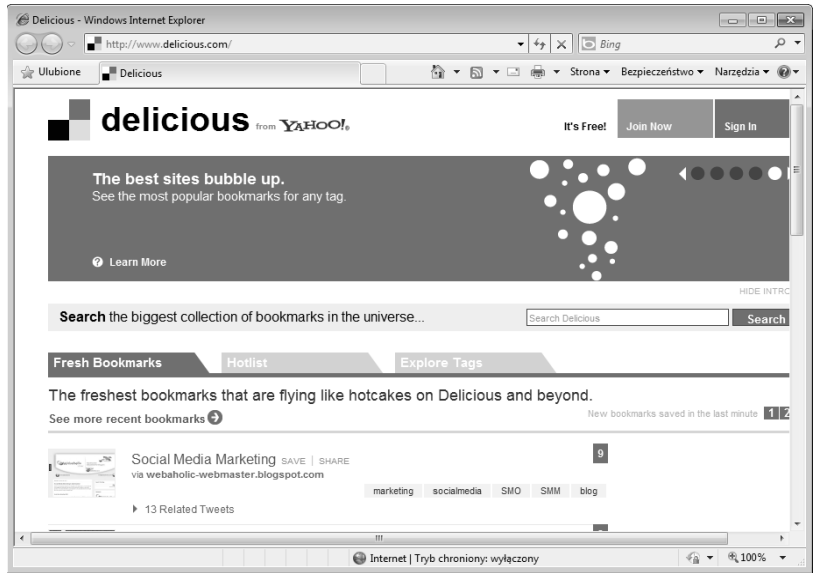
Widget tekstowy jest tak popularny, że z myślą o rozszerzeniu jego możliwości stworzono wiele dodatków — bardziej rozbudowanych odpowiedników widgetów, którym poświęciliśmy rozdział 10., stosowanych na blogach z rodzinyWordPress.org. Przykładem może być widget *WYSIWYG Text*, który pozwala zmieniać kolor tekstu i wprowadzać odnośniki bez potrzeby odwoływania się do kodu.

Możliwe zawartości widgetu Tekst

Jeśli zależy Ci na szybkim wykorzystaniu możliwości widgetu *Tekst*, możesz wstawić w nim kod usług udostępnianych przez niektóre serwisy. Tak wykorzystany widget staje się bramką wyjściową prowadzącą do serwisów, z których pobrałeś kod.

Oto niektóre z naszych ulubionych serwisów tego typu:

- ***Babelfish Yahoo Translation*** — ta usługa umożliwia tłumaczenie zawartości Twojego bloga[7]. Jest ona ciekawym rozwiązaniem dla każdego z odwiedzających, który posługuje się językiem prowadzenia bloga jako drugim. Dzięki temu nie dość, że zrozumie więcej z Twoich notatek, to jeszcze poprawi znajomość drugiego języka.
- ***ClustrMaps*** — jest to jeden z wielu dostępnych w sieci liczników odwiedzin, który jednocześnie pozwala śledzić pochodzenie osób wchodzących na witrynę. To zabawny dodatek, który często przyczynia się do zwiększenia liczby komentarzy na blogu.
- ***Delicious*** — ta witryna, zaprezentowana na rysunku 6.7, to najprawdopodobniej najbardziej licząca się strona o charakterze społecznym, utworzona z myślą o autorach blogów, którzy chcą pisać o blogach oraz śledzić to, co się na nich dzieje. Zyskaj uznanie na tym serwisie, a wprawisz wszystkich odwiedzających w euforię. Sprawdź stronę, a potem dołącz do jej społeczności[8].



Rysunek 6.7. Zaprzyjaźnij się z delicious.com

[7] O ile jest on prowadzony po angielsku, niemiecku, hiszpańsku, francusku, włosku lub portugalsku, a tłumaczenie ma odbywać się na jeden z tych języków. Możesz też dodać do widgetu kod prowadzący do tłumacza Google Translate, umieszczony na stronie *http://en.forums.wordpress.com/topic/translate-your-blog-into-different-languages?replies=64#post-243059*. Zgodnie z zamieszczonymi przez autora wskazówkami należy zamienić w nim nazwę bloga na własną i pierwszy z pary kodów języków w każdym wpisie na kod języka, w jakim prowadzony jest blog — *przyp. tłum.*

[8] Serwis w języku angielskim — *przyp. tłum.*

I jeszcze kilka serwisów, których zawartość może uatrakcyjnić Twój blog:

- ***Bitty Browser*** — dzięki tej aplikacji ludzie mogą przeglądać zawartość internetu, posługując się widgetem! Przedstawiona na rysunku 6.8 przeglądarka Bitty Browser pozwala odwiedzać polecane przez Ciebie witryny bez konieczności opuszczania Twojego bloga. Aplikacja oferuje wiele opcji konfiguracyjnych i zapewnia doskonałą zabawę przez wiele godzin.



Rysunek 6.8. Bitty Browser, choć mała, jest niezwykle potężna

- ***Flickr*** — wrzuć na blog swoje zdjęcia. Jeżeli jesteś użytkownikiem serwisu Flickr albo myślisz o dołączeniu do niego, z pewnością ucieszysz się na wieść, że możesz łatwo dodawać do bloga swoje prace.
- ***Last.fm i Pandora*** — Last.fm pozwoli Ci umieścić na blogu listy przebojów, a Pandora przedstawi odwiedzającym Twoje ulubione piosenki. Czytanie bloga przy muzyce jest ciekawsze!
- ***YouTube*** — możesz też umieścić stałe okno serwisu YouTube na swoim blogu. Odwiedzający będą mieli frajdę, ale pamiętaj, że mogą też porzucić Cię na rzecz YouTube.

Odnośniki do tych i innych podobnych usług znajdziesz na stronie kodeksu WordPress Codex poświęconej widgetowi *Tekst* — *http://codex.wordpress.org/WordPress_Widgets#Using_Text_Widgets*[9].

[9] Strona w języku angielskim — *przyp. tłum.*

Ostrzeżenie
Powtórzymy jeszcze raz ostrzeżenie udzielone Ci w rozdziale 3., ponieważ nabiera ono większego znaczenia w obliczu wszystkich serwisów, z których zawartości możesz skorzystać w widgecie *Tekst*. Jeśli wprowadzisz na blog zbyt wiele widgetów, w tym także kilka różnych wersji widgetu *Tekst*, odwiedzający zapewne zignorują większość z nich. Zmuś się do zachowania umiaru, gdy będziesz zastanawiać się, co wprowadzić na stronę bloga. Sprawdź, jak wyglądają wybrane widgety, a potem zastanów się, które z nich możesz jeszcze usunąć. Pamiętaj, że ich zawartość powinna wiązać się ściśle z treścią bloga. (Możesz też zapytać odwiedzających o opinię w kwestii, które z widgetów uznają za najbardziej potrzebne — przecież blog jest między innymi właśnie po to).

Co można robić z widgetem Tekst

Pokazaliśmy Ci, jakiego rodzaju cudze treści możesz umieszczać w widgecie tekstowym, ale co z danymi, które mógłbyś wprowadzać sam?

Oto kilka pomysłów na dobry początek:

- ***Spis najważniejszych wpisów*** — możesz wymienić w nim wszystkie najpopularniejsze wpisy albo podzielić je tematycznie. Możesz wymienić wpisy najchętniej komentowane. Możesz umieścić w widgecie listę najważniejszych tematów bloga specjalnie z myślą o nowych użytkownikach. Taki widget może stać się doskonałą pomocą dla każdego, kto będzie chciał poznać lepiej Twój serwis.
- ***Informacje o Tobie i interesujących Cię tematach*** — możesz umieścić w widgecie kilka słów wprowadzenia czy powitania, a może po prostu spis odnośników do odpowiednich stron — czy to na blogu, czy poza nim.
- ***Ilustrowany przegląd autorów*** — widget tekstowy może też zawierać informacje o wszystkich autorach wraz z ich zdjęciami.

Więcej informacji na temat sposobów korzystania z widgetu *Tekst* znajdziesz na stronach pomocy technicznej WordPress — *http://support.wordpress.com/widgets/text-widget/*[10].

Widget tekstowy przyjmuje dane zapisane w postaci tekstu, kodu HTML, a nawet nieduże obrazki. Dawniej można było wprowadzać do niego także kod JavaScript, lecz powodowało to zbyt wiele problemów. Dziś widget ten nie przyjmuje żadnych skryptów typu JavaScript czy Flash.

Umieszczanie grafiki w widgecie Tekst
Najprostszym sposobem na umieszczenie obrazka w widgecie jest dodanie wybranej grafiki do biblioteki mediów WordPress, co opisaliśmy w rozdziale 9. Później wystarczy wprowadzić adres URL obrazu do widgetu. Odpowiedni kod HTML będzie miał w tym przypadku postać `<img src"URL">`, gdzie w miejsce skrótu URL wkleisz adres grafiki pobrany z biblioteki mediów.

[10] Strona w języku angielskim — *przyp. tłum.*

Prosty przykład pracy z widgetem Tekst

Dodając widget *Tekst* do bloga, możesz stworzyć wyjątkowe okienko informacyjne. Oto przykład, jak zbudować taki obiekt od podstaw. W naszym przykładzie połączymy w nim tekst, grafikę i prosty kod HTML.

1. Otwórz panel administracyjny bloga i rozwiń menu *Wygląd*, następnie kliknij pozycję *Widgety*. W oknie przeglądarki pojawi się znana Ci z rozdziału 3. strona *Widgety*.
2. Przeciągnij widget *Tekst* z obszaru *Dostępne widgety* (środek strony) w obszar panelu (po prawej). Pole ustawień widgetu rozwinie się samo.
3. Wpisz tytuł widgetu. To ważna opcja — źle dobrany tytuł będzie denerwował wszystkich, którzy odwiedzą Twoją stronę.

   W przykładzie posłużymy się tytułem *Google Translate*. Tworzony właśnie dodatek pozwoli tłumaczyć zawartość strony na inne języki, odwołując się właśnie do usługi Google Translate.
4. Aby wstawić grafikę do widgetu, dodaj ją do biblioteki mediów, postępując zgodnie ze wskazówkami z rozdziału 8. Później skopiuj adres URL dodanego właśnie obrazu i wstaw go do widgetu tekstowego za pomocą znacznika `<img>`.
5. W dowolny sposób dodaj do widgetu tekstowego treść i kod HTML. Pamiętaj, aby zaznaczyć opcję *Automatycznie twórz akapity*, chyba że chcesz wstawiać ręcznie mnóstwo znaczników `<p></p>`.

   Dla przykładowej strony, którą posługujemy się w większości przypadków, i wybranej usługi Google Translate wpis do widgetu składa się z odpowiedniego odnośnika do narzędzia Google Translate, adresu bloga i nazwy języka pojawiającej się tu w charakterze odnośnika. Oto jeden z wpisów w widgecie:

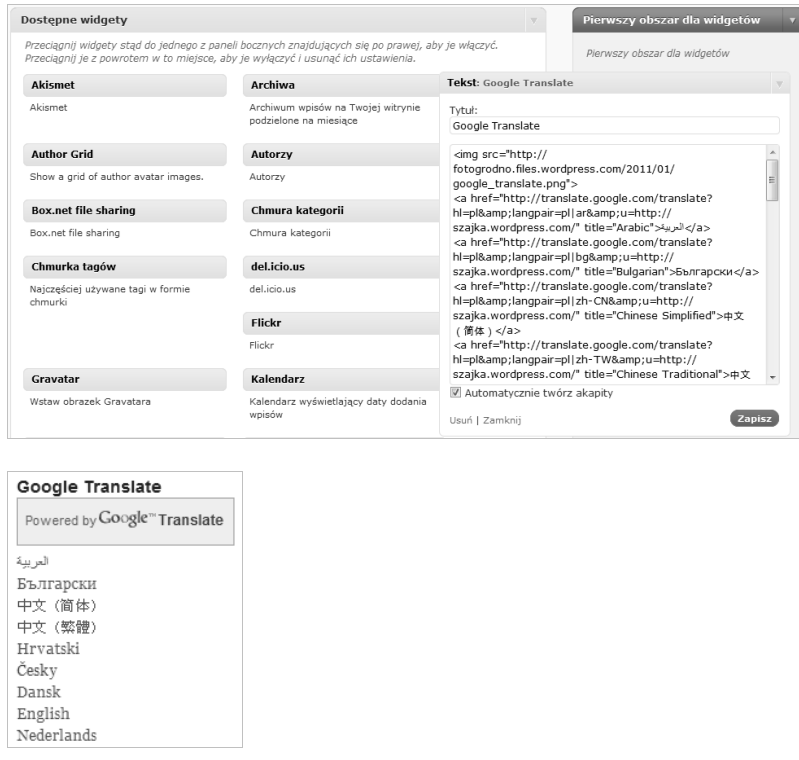
   ```
   <a href="http://translate.google.com/translate?hl=pl&amp;langpair=pl|hr&amp;
   ↪u=http://fotogrodno.wordpress.com" title="Croatian">Hrvatski</a>
   ```
6. Regularnie zachowuj postępy pracy, klikając przycisk *Zapisz*.
7. Gdy skończysz, kliknij ostatni raz przycisk *Zapisz* i podejrzyj efekty, otwierając stronę bloga.

   Przykładowy wpis w widgecie oraz wygląd gotowego dodatku przedstawiliśmy na rysunku 6.9.

Po zakończeniu pracy sprawdź jej wyniki, przeprowadź testy i przekonaj się, co działa bez usterek, a co nie, i jak poprawić wszystkie niedociągnięcia.

Następujące elementy przykładu przypadły nam do gustu:

- wyraźna, rzucająca się w oczy grafika, którą rozpozna każdy powracający na nasz blog;
- dobrze dobrany tytuł — nazwa Google Translate mówi sama za siebie, a kluczowe w tym przypadku słowo „translate" (tłumaczyć) wypada dokładnie nad przyciągającą wzrok grafiką, przez co nie sposób go pominąć;
- wyraźnie widoczna funkcja widgetu.



Rysunek 6.9. Kod wewnątrz widgetu tekstowego rzadko wygląda pięknie… ale efekty bywają zachwycające

Natomiast gdybyśmy mieli wprowadzać w nim poprawki, zapewne zdecydowalibyśmy się na następujące zmiany:

- Popracowalibyśmy nieco nad kodem, by bardziej rozsunąć nagłówek widgetu, grafikę i pierwszy odnośnik. Można by też spróbować przesunąć listę języków w kierunku środka obrazka.
- Zapewne sformatowalibyśmy inaczej listę dostępnych języków (rysunek 6.9 przedstawia tylko fragment widgetu, całość jest znacznie dłuższa). Poszczególne języki można by na przykład oddzielić spacjami i pionową kreską. Wtedy, nie rezygnując z dostępnych opcji tłumaczenia, moglibyśmy uzyskać widget o rozsądnych rozmiarach.
- Dodalibyśmy wpis informujący o nowej funkcji, wyjaśniając, dlaczego tłumaczenie strony jest tak istotne z punktu widzenia poruszanych na niej tematów. Wpis oczywiście powinien pojawić się w najbardziej popularnych językach, by dotrzeć do jak największej liczby odbiorców.

Wskazówka

Zawartość widgetu *Tekst* można także tworzyć inaczej. Jego treść możesz przygotować tak samo jak treść każdego innego wpisu — w zakładkach *Wizualny* i *HTML* edytora wpisów. Przed umieszczeniem kodu w widgecie sprawdź, jak wygląda szkic wpisu, a następnie wytnij zawartość karty *HTML* i wklej ją w polu tekstowym widgetu. Oczywiście nie zapomnij na koniec sprawdzić, jak prezentuje się Twoja praca.

SKOROWIDZ

A

abonament, *Patrz* opłata roczna
adres
  mapowanie, 28
  URL, *Patrz* odnośnik
Akismet, 23, 28, 55, 63, 64, 345
alfabet, 83
Amazon.com, 70
anchor, *Patrz* kotwica
ankieta, 196, 210, 216
  dodawanie, 210, 211, 215
  edycja, 215
AOL, 172
Apple, 67, 221, 222, 226
archiwizator, 66
arkusz stylów, 28, 78, 84, 101, 129, 167, 172, 265, 295, 318, 322, 353, 355, 362, 364, 365, 371, 376
Arrington Michael, 226
AT&T, 221, 226
Automattic, 23, 24, 26, 28, 41, 142, 196, 222, 265, 349, 424
awatar, 64

B

baza danych MySQL, 294, 295, 300, 311, 402, 403
Berners-Lee Tim, 355, 356
biblioteka mediów, 256, 260, 279
Bing, 66
Blip.tv, 351
blog, 34
  administrator, 198, 199
  autor, 198
  biznesowy, 36, 37
  firmowy, 29, 37
  fotoblog, 325, 351
  interfejs, 37
  język, 54, 83
  mikroblog, 325, 352
  nagłówek, 83, 85, 90, 101, 130, 376, 381, 411
  nawigacja, 383, 450
  nazwa, 50, 54
  nowości, 441
  nowy, 54
  osobisty, 36
  otwarty, 24, 47
  pozycjonowanie, 48, 54
  prywatny, 28, 47, 54, 67, 268
  publiczny, 54
  redaktor, 198
  sieć, 29
  stopka, 412
  subskrybent, 198
  szablon, 85, 322
  tytuł, 54
  ustawienia, 46, 55
  usuwanie, 68, 69
  użytkownik, 197, 198, 199, 200, *Patrz też* blog subskrybent
  właściciel, 198
  współtwórca, 198
Blogger, 18, 31
Bloglines, 237
BlogTalkRadio, 279
BuddyPress, 309
Byfield Bruce, 331

C

cena, 23
Creative Commons, 262
CSS, *Patrz* arkusz stylów
czcionka, 360, 363, 366, 368, 376, 378, 385
często zadawane pytania, *Patrz* FAQ
czytnik kanałów, 61

D

deklaracja, 358, 365, 367
właściwość, 365, 366, 367, 368, 382, 383, 386
dokumentacja, 433, 435, 436, 437, 438
DOM, *Patrz* obiektowy model dokumentu
domena nazwa, *Patrz* nazwa domeny
Dropbox, 279
dyskusja, 48, 350
dział pomocy, *Patrz* pomoc

E

edytor
kodu, 359
motywów, 372, 373, 375
szablonów, 330
tekstu, 362
wpisów, 168, 172, 372
edytor grafiki, *Patrz* grafika edytor
Eklund Kevin, 341
ekran OLED, 79
ekstranet, 24
e-mail, 51, 52, 56, 63, 68, 69, 71, 72, 73, 119, 205, 345, 349, 351
enhanced feeds, *Patrz* kanał nowości

F

Facebook, 36, 43, 109, 349
FAQ, 55
FeedDemon, 237
filtr
Akismet, 345, *Patrz* Akismet
funkcji, 99
Firefox, 27, 172, 237, 331, 336, 357
Flash, 285, 351
Flickr, 262, 276
formatowanie, 168, 169
HTML, 122
listy, *Patrz* lista
tekstu, 127, 129, 131, 132, 133, 134
znaków, 175
forum, 451, 461
Free Software Foundation, 331
FTP, *Patrz* protokół FTP
funkcja
bloginfo, 412
get_archives, 411
oEmbed, *Patrz* oEmbed
QuickPress, *Patrz* QuickPress

G

General Public License, *Patrz* GNU General Public License
GNU General Public License, 329, 331, 378, 418
opłaty, 418
Google, 26, 31, 66, 67, 152, 185, 196, 221, 226, 229, 230, 437
Czytnik RSS, 237
ranking, 49
Google AdSense, 26, 29, 32, 35, 328, 350, 428
Google AdWords, 231
Google Analytics, 217, 348
Google News, 245
Google Sites, 279
Google Voice, 221, 222, 226, 230
GPL, *Patrz* GNU General Public License
grafika, 119, 121, 191, 244, 247, 251, 292, 324, *Patrz też* motyw graficzny
dodawanie, 253, 256, 260, 261, 262
edycja, 255
edytor, 247
oblewanie tekstem, 256
rozmiar, 248, 249, 251
zamiana w odnośnik, 258

H

Hannah Doug, 441
hasło, 51, 52
hiperłącze, *Patrz* odnośnik
home.pl, 271

I

IBM, 67
Internet Explorer, 172, 237
intranet, 24
iTunes, 278

J

JavaScript, 215
jądro systemu, 26, 295, 335, 342, 354, 393, 412
język
HTML, 354, 356, *Patrz* kod HTML
HTML 5, 355
JavaScript, 295, 357
PHP, 167, 294, 353, 355, 357, 391, 393, 399, 400, 401
PHP 5, 394
XHTML, 355, 357
XML, 355, 356, 359
znaczników, *Patrz* język XML

K

kanał nowości, 61, 236, 238, 319, 350, 386
kategoria, 32, 36, 61, 78, 120, 122, 140, 141, 142, 157, 236, 371
dodawanie, 144
edycja, 147
lista, 143, 144, 145

modyfikowanie, 147
odnośników, 163
przenoszenie wpisu, 150
układ hierarchiczny, 145
usuwanie, 148
wiekowa, 287
zamiana na tagi, 148
klucz API, 40, 41, 55, 345
kod
CSS, 28, 32, 167
HTML, 126, 129, 165, 166, 167, 168, 169, 177, 182, 184, 185, 191, 357, *Patrz też* język HTML
JavaScript, 191, 348, *Patrz też* język JavaScript
PHP, 167, 295, 324, 332, 396, *Patrz też* język PHP
XHTML, 167, 184, 356, *Patrz też* język XHTML
kolor, 78, 102, 366, 374, 379, 382, 384
kolumna, 80, 100
o stałej szerokości, 81, 85, 90, 93
o zmiennej szerokości, 81, 85, 89
komentarz, 62, 63, 64, 70, 120, 182, 196, 198, 345, 349, 385
ocena, 68, 69, 70, 349
zmiany, 350
konferencja WordCamp, *Patrz* WordCamp
kopia zapasowa, 40, 198, 306, 315
kotwica, 171, 181

L

Lerdorf JavaScript, 294
Link ID, 158
LinkedIn, 43
lista, 178
nieuporządkowana, 178
słów, 64
uporządkowana, 178
Little Mike, 23

M

Mamchenkov Leonid, 404
Mandato Angelo, 340
Matt Mullenweg, 394
Microsoft, 67, 172, 357
Microsoft Paint, 250
mikroformat, 83
motyw graficzny, 24, 25, 41, 42, 46, 55, 70, 76, 77, 83, 85, 99, 130, 147, 167, 267, 295, 318, 321, 322, 323, 324, 326, 328, 330, 341, 348, 354, 366, 404, 457, 461
bezpieczeństwo, 333
dwukolumnowy, 80, 85, 89, 90, 93, 381
instalowanie, 332
poprawianie, 368
rozmiar, 65
szablon, 65, 167, 404, 405
tworzenie, 353, 377
ustawienia, 54, 370, 372
Mozilla Firefox, *Patrz* Firefox
Mullenweg Matt, 23, 340, 346
MySQL, *Patrz* baza danych MySQL

N

Navigator, 172
nazwa
bloga, 50, *Patrz* blog nazwa
domeny, 48, 49, 50, 67, 270, 273, 274, 275, 276, 292
użytkownika, 51
Netscape, 172, 357
notka, *Patrz* wpis

O

OASIS, 355
obiektowy model dokumentu, 355, 357
odnośnik, 55, 62, 63, 109, 130, 131, 135, 170, 171, 224, 245, 256, 259, 262, 275, 281, 351
dodawanie, 134, 137, 162
edycja, 163
kategoria, 163
kolor, 379
lista, 141, 159, 160, 162
powiązany, 103, 104
tworzenie, 180
odsyłacz, *Patrz* odnośnik
oEmbed, 185
OpenID, 67, 349
opinia, *Patrz* komentarz
opłata
Akismet, 29
audio, 244
roczna, 24, 47
Support Network, 29
usunięcie reklam, 265, 268, 269, 429
VideoPress, 265, 270, 284
wideo, 28, 244
większa pamięć, 28, 265, 269
własna nazwa domeny, 28, 48, 270
własny arkusz stylów CSS, 267
WordPress.org, 265, 272
wykorzystania pasma, 261
zmiany w arkuszu stylów, 28, 84
zwiększenie dostępności bloga prywatnego, 28, 265, 268
Organization for the Advancement of Structured Information Standards, *Patrz* OASIS

P

panel boczny, 80, 81, 386, 409, 410
PHP, *Patrz też* język PHP, kod PHP
ping, *Patrz* technologia pingback
pingback, *Patrz* technologia pingback
plik
  audio, 244, 278, 279, 285, 292, 351
    dodawanie, 280, 281, 282
    odtwarzanie, 280, 281, 351
  GIF, 247, 249, 250, 253
  graficzny, *Patrz* grafika
  JPEG, 102, 249, 250, 253
    kompresja, 247
  JPEG 2000, 249
  MP3, 244, 280, 351
  PDF, 244, 253, 259
  PNG, 102, 249, 250, 253
  podcastu, 351
  PPT, 253, 259
  WAV, 244, 280
  wideo, 244, 278, 283, 285, 292, 351
    dodawanie, 285, 286
    odtwarzanie, 285, 351
podcast, 351
podręcznik użytkownika, *Patrz* WordPress User Handbook
PollDaddy, 28, 211, 214
pomoc, 196, 266, 450
post, *Patrz* wpis
powiadomienie, 62, 63
prawa autorskie, 245, 246, 261, 386
Press This, 125
profil użytkownika, 55
programista, 324, 328, 329, 331, 334, *Patrz też* projektant
projektant, 322, 324, 325, 327, 328, 329, 330, 353, *Patrz też* programista
protokół
  FTP, 296
  HTTP, 166
  TrackBack, *Patrz* technologia trackback
prywatność, 47, 48, 66
przechwytywanie z ekranu, 109
przeglądarka, 237 , *Patrz też* Internet Explorer, Firefox, Safari
  silnik, 357
przenosiny, 40
publikacja za pomocą poczty elektronicznej, 71, 72, 73

Q

QuickPress, 121, 122

R

R5 Ltd., 330
Really Simple Syndication, *Patrz* RSS
reguła, 365, 366, 367, 368, 373, 383
reklama, 24, 26, 27, 29, 265, 272, 350, 428
Related Links, *Patrz* odnośnik powiązany
RSS, 61, 218, 232, 236, 319, 350, 352, 386
RSSOwl, 237
RTL Language Support, *Patrz* zapis tekstu od prawej do lewej

S

Safari, 172, 237, 442
ScribeFire, 125
selektor, 365, 367, 368, 374, 376, 383
serwer zewnętrzny, 244, 254, 261, 262, 263, 279, 280, 291, 310, 430
Sharing, 67
sieć
  semantyczna, 356, 408
  wsparcia technicznego, *Patrz* Support Network for WordPress
słowo kluczowe, 142, 208, 224, 339
spam, 28, 63, 64, 345
społeczność
  internetowa, 38
  użytkowników, 25, 38, 429, 454, 457
sprawdzanie pisowni, 122
Stallman Richard, 331
standard OpenID, *Patrz* OpenID
statystyki, 217, 218, 220, 221, 235, 349, 450
  liczba odwiedzin, 35, 37, 41, 222, 223, 348
  najczęściej odwiedzane, 222
  najczęściej szukane, 222
  najpopularniejsze wpisy, 222, 223, 232
  odnośniki, 228
  odsyłający, 224, 225
  wyjścia, 233
Stewart Ian, 334
strona
  dynamiczna, 353
  informacyjna, 204
  kontakt, 204
  mapa, 350
  o autorze, *Patrz* strona informacyjna
  statyczna, 195, 201, 202, 206, 348, 370
    dodawanie, 207, 208
    komentarz, 209, 216
support, *Patrz* pomoc
Support Network for WordPress, 29
system
  CMS, 300
  kategorii wpisów, 32

o budowie segmentowej, 335
ocen, 196
WordPress *Patrz* WordPress.com

T

tagi, 38, 61, 120, 140, 141, 142, 143, 339, 385
definiowanie, 154
dodawanie, 153, 155
nadawanie, 151, 153
nazwa, 156
przeglądarka, 161
usuwanie, 156
zamiana na kategorie, 156
zarządzanie, 154, 157
technologia
pingback, 35, 49, 63, 209, 216
trackback, 35, 49, 63, 216
Technorati, 66
tekst formatowanie, *Patrz* formatowanie tekstu
thumbnail, 326
Time Warner, 172
trackback, *Patrz* technologia trackback
tryb pełnoekranowy, 122, 123, 124, 125, 137, 139, 153
Twitter, 36, 43, 120, 325, 349, 352, 446

U

ulepszenia, 27, 267
usunięcie reklam, 268
własna nazwa domeny, 271
własny arkusz stylów CSS, 267
zwiększenie dostępności bloga prywatnego, 268
uprawnienia, 301, 302, 313, 314, 375
urządzenia przenośne, 80
usługa
hostingowa, 24, 28, 29, 41, 42, 244
No-ads, 28, *Patrz* ulepszenie usunięcie reklam
pingback, *Patrz* technologia pingback
zaawansowana, 454
ustawienia ogólne, *Patrz* blog ustawienia

V

VanFossen Lorelle, 247
VideoPress, 244, 265, 270, 284, 287
Vimeo, 351

W

W3C, *Patrz* World Wide Web Consortium
walidator, 359, 361, 366
WAMP, 311
widget, 24, 25, 41, 42, 75, 82, 100, 104, 109, 113, 143, 167, 202, 318, 336, 361, 371, 386, 387, 388, 409, 457
archiwa, 106, 113, 206
autorzy, 206
dodawanie, 107
domyślnie zainstalowany, 104, 106
kalendarz, 113
kategorie, 113
kolejność, 108
meta, 106, 113
najnowsze komentarze, 113
najpopularniejsze wpisy i strony, 113
najważniejsze komentarze, 206
odnośniki, 113, 135, 158, 206
RSS, 114, 218, 238
strony, 113
szukaj, 113
tagi, 113, 152
tekst, 115, 166, 188, 191, 192, 193, 208, 215, 238, 360
własne menu, 113
witryna
powiązana, 103
usuwanie, *Patrz* blog usuwanie
zaufana, 67
WordCamp, 25, 230, 394
WordPress API, 158
WordPress Codex, 56, 436
WordPress MU, 24, 309
WordPress User Handbook, 435
WordPress VIP Hosting, 29
WordPress.com, 16, 18, 23, 24, 25, 37, 38, 45, 75, 166, 168, 265, 266, 272, 423, 426
aktualizowanie, 304, 306
import danych, 307, 310
instalowanie, 300, 302
kokpit, 317, 318, 319
mapa, 449
plik konfiguracyjny, 314, 317
WordPress.com stats, 41
WordPress.org, 16, 18, 24, 25, 28, 29, 35, 37, 38, 40, 84, 168, 188, 266, 291, 299, 423, 426
World Wide Web Consortium, 353, 355, 357
wpis, 119, 121, 198, 201, 216, 351, 385
długość, 120
dodawanie, 123
edycja, 123
katalogowanie, 141
nadawanie tagów, 151, 158
ocena, 70
publikowanie, 138
rodzaj, 408
typowy, 119
zapisywanie, 138
zmiana kategorii, 150

wtyczka, 42, 105, 107, 217, 311, 318, 324, 335, 336, 337, 361, 418, 457, 461
AddToAny, 351
Ajax Edit Comments, 350
Akismet, 345
All in One SEO Pak, 350
bSuite, 349
Disqus, 349
dodawanie, 338, 341, 342
Easy AdSense, 350
File Monitor, 347
GD Press Tools, 346
GD Star Rating, 349
Google Analitycs for WordPress, 348
Google XML Sitemaps, 350
Hello Dolly, 345, 346
informacje szczegółowe, 339
IntenseDebate, 349
kompatybilność, 340
Lifestream, 352
modyfikowanie, 343
Page Flip Image Gallery, 351
Podcasting, 351
recenzje, 340, 341
Search Meter, 349
SEO No Duplicate, 350
ShareThis, 351
Sociable, 351
Subscribe to Comments, 350
testowanie, 342
Twitter Tools, 352
tworzenie, 353
Update Notifier, 347
Viper's Video Quicktags, 351
Woopra Analytics, 348
WordPress Exploit Scanner, 348
WP Database Backup, 342, 347
WP FollowMe, 352
WP Great Box, 352
WPAudio MP3 Player, 351
WP-FollowMe, 413, 414, 415
wybieranie, 340, 341
wyszukiwarka, 48, 66, 229, *Patrz też* Google, Yahoo!, Bing, AOL, Technorati,

X

XXL, 442

Y

Yahoo!, 146
YouTube, 28, 166, 185, 244, 284, 351

Z

zapis tekstu od prawej do lewej, 83
Zeldman Jeffrey, 355
zlew kuchenny, 132, 172
znacznik, 168, 170, 183, 184
a, 181
b, 176
br, 177
div, 177
em, 169, 176
i, 176
li, 178
more, 182
ol, 178
otwarcia, 169
p, 177, 184
strong, 169, 176
ul, 178
zamknięcia, 169

WordPress to obecnie najpopularniejszy system zarządzania treścią. Według badań W³Techs jego udział w rynku systemów CMS wynosi ponad 54%! Co wpłynęło na tę popularność? Czym wyróżnia się on na tle konkurencji? Są to niezwykle trudne pytania, ale w tym przypadku można bez wahania postawić na niezwykłą elastyczność, łatwość konfiguracji, estetyczny domyślny wygląd oraz ogrom dodatków, dzięki którym możesz uzupełnić go o nawet najbardziej wymyślną funkcjonalność.

Dzięki tej książce poznasz WordPress od podszewki. Zgłębisz jego najskrytsze tajemnice oraz nauczysz się dostrajać nawet najbardziej zaawansowane funkcje. Na samym początku dowiesz się, jak podejść do tematu tworzenia własnej witryny opartej na WordPressie. Po interesującym wstępie przyjdzie czas na konfigurację. Zobaczysz, jak zadbać o wygląd strony, przygotować własny nagłówek, zorganizować menu oraz dodać pierwszy wpis. W kolejnych rozdziałach zapoznasz się z metodami publikacji materiałów multimedialnych, dodawania grafik do wpisów oraz projektowania własnych motywów. Książka ta sprawdzi się świetnie w rękach każdego użytkownika pragnącego zaistnieć w sieci!

- *Wersje platformy WordPress*
- *Użytkownicy systemu WordPress*
- *Przygotowania do otwarcia witryny opartej o WordPressa*
- *Konfiguracja wyglądu — wybór motywu, edycja nagłówka*
- *Tworzenie wpisów*
- *Dodawanie odnośników i zarządzanie nimi*
- *Kategoryzowanie wpisów*
- *Wykorzystanie własnego kodu HTML*
- *Dodawanie statycznych stron*
- *Dodawanie do wpisów materiałów multimedialnych*
- *Instalacja systemu WordPress na własnym serwerze*
- *Tworzenie i nabywanie nowych motywów*
- *Wykorzystanie języka PHP*
- *Przykładowe witryny oparte na WordPressie*

Zacznij prowadzić własną witrynę WWW!

helion.pl
księgarnia internetowa

Helion

Cena 79,00 zł

ISBN 978-83-246-3080-6

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Informatyka w najlepszym wydaniu