

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

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

505 praktycznych skryptów dla webmastera

Autor: Marcin Lis
ISBN: 83-7361-749-3
Format: B5, stron: 944



Od dziś Twoje strony WWW będą bardziej funkcjonalne i atrakcyjne

Dynamiczny rozwój internetu zmienił również oblicze stron WWW. Skończyła się epoka statycznych stron WWW. Dziś niemal na każdej stronie WWW, zarówno korporacyjnej, jak i prywatnej, wykorzystywane są technologie skryptowe, dzięki którym można znacznie wzbogacić możliwości standardowego języka HTML. Technologie skryptowe nie tylko pomagają w zwiększeniu atrakcyjności wizualnej strony, ale także w jej zabezpieczeniu, zarządzaniu nią i analizie ruchu odwiedzających. Połączenie skryptów z bazami danych pozwala na stworzenie systemów dynamicznego generowania treści, wymianę informacji z osobami oglądającymi stronę i tworzenie statystyk.

Książka „505 praktycznych skryptów dla webmastera” to zbiór skryptów, napisanych w różnych językach skryptowych, dzięki którym Twoja strona zyska na atrakcyjności i funkcjonalności. Skrypty przedstawione w książce to zarówno efekty graficzne i tekstowe, wzbogacające wygląd strony, jak i złożone mechanizmy, które służą do tworzenia interakcji z użytkownikiem, generowania treści i odnośników na podstawie bazy danych oraz analizowania odwiedzin na witrynie. Książka zawiera przykłady rozwiązań, wykorzystujące wszystkie najpopularniejsze technologie skryptowe – DHTML, JavaScript, VBScript, Java, PHP, ASP oraz Perl.

- Dynamiczny HTML
- Wyświetlanie daty i czasu
- Interakcja z oknem przeglądarki
- Modyfikowanie wyglądu łączy
- Zarządzanie banerami w witrynie
- Obsługa logowania i autoryzacji użytkowników
- Pobieranie i wysyłanie plików z poziomu strony WWW
- Rozpoznawanie adresu IP oraz typu i wersji przeglądarki
- Statystyki i liczniki odwiedzin
- Współpraca z bazą danych

Jeśli chcesz, aby Twoje strony WWW wyróżniały się spośród innych, zastosuj w nich skrypty. Wszystkie niezbędne informacje na ten temat znajdziesz w książce.



Spis treści

Wstęp	15
Rozdział 1. Dynamiczny HTML.....	21
Skrypty różne	21
Skrypt 1. Blokada prawego przycisku myszy.....	21
Skrypt 2. Okno powitalne.....	23
Skrypt 3. Okno pożegnalne	24
Skrypt 4. Rozpoznanie typu przeglądarki.....	24
Skrypt 5. Ładowanie różnych stron w zależności od typu przeglądarki.....	26
Skrypt 6. Ładowanie różnych stron w zależności od systemu operacyjnego	27
Skrypt 7. Zmiana wyglądu kursora myszy	28
Skrypt 8. Sprawdzenie poprawności danych w formularzu.....	30
Skrypt 9. Kalkulator	33
Skrypt 10. Automatyczne przenoszenie kursora między polami formularza.....	38
Skrypt 11. Zablokowanie możliwości wpisywania określonych znaków w formularzu.....	39
Skrypt 12. Pole tekstowe dopasowujące swój rozmiar do wprowadzanego tekstu....	40
Skrypt 13. Zmiana obrazka po najechnaniu myszą (dynamiczne przyciski)	42
Skrypt 14. Weryfikacja adresu e-mail	43
Skrypt 15. Ukrycie adresu e-mail (antyspam)	45
Skrypt 16. Podświetlenie komórki tabeli.....	46
Odnośniki	48
Skrypt 17. Zmiana kolorów odnośników	48
Skrypt 18. Akapit tekstowy udający odnośnik	49
Skrypt 19. Odnośniki w polu Combo	49
Skrypt 20. Odnośniki w polu Combo z automatyczną zmianą strony	51
Skrypt 21. Odnośniki na liście zwykłej.....	52
Skrypt 22. Odnośniki w polach wyboru typu radio	53
Skrypt 23. Odnośniki na przyciskach	54
Skrypt 24. Odnośniki z dodatkowym opisem w polu tekstowym.....	55
Skrypt 25. Odnośniki z dodatkowym wyróżnieniem.....	57
Okna.....	59
Skrypt 26. Automatyczne załadowanie innej strony [JavaScript]	59
Skrypt 27. Automatyczne załadowanie innej strony [HTML].....	60
Skrypt 28. Automatyczne wczytanie innej strony z regulowanym opóźnieniem	61
Skrypt 29. Uniemożliwienie wczytania strony do ramki.....	62
Skrypt 30. Dodanie strony do listy Ulubione	62
Skrypt 31. Ustawianie strony jako startowej.....	64

Skrypt 32. Otworzenie dokumentu w nowym oknie	65
Skrypt 33. Strona zależna od rozdzielczości ekranu.....	68
Skrypt 34. Modyfikacja wyglądu suwaków okna.....	69
Skrypt 35. Dynamiczna zmiana kolorów suwaków okna.....	70
Skrypt 36. Modyfikacja wyglądu suwaków w ramce.....	72
Skrypt 37. Potrząśnięcie ekranem	74
Skrypt 38. Zamknięcie okna przeglądarki.....	74
Data i czas	75
Skrypt 39. Wyświetlenie aktualnej daty	75
Skrypt 40. Zegar w polu tekstowym.....	77
Skrypt 41. Zegar na pasku stanu	79
Skrypt 42. Zegar w pasku tytułu	79
Skrypt 43. Zegar na warstwie.....	80
Skrypt 44. Zegar na przycisku.....	82
Skrypt 45. Określenie dnia tygodnia	83
Skrypt 46. Wyświetlenie daty, czasu oraz dnia tygodnia	84
Skrypt 47. Ile dni do...?	86
Skrypt 48. Data aktualizacji strony	88
Skrypt 49. Kalendarz.....	89
Skrypt 50. Strona zależna od pory dnia.....	92
Banery	93
Skrypt 51. Losowo generowany baner	93
Skrypt 52. Banery wyświetlane w określonej kolejności	94
Skrypt 53. Zmieniające się losowo banery	96
Skrypt 54. Banery zmieniające się w określonej kolejności.....	98
Skrypt 55. Baner zależny od pory dnia	99
Skrypt 56. Baner zależny od dnia tygodnia.....	100
Skrypt 57. Baner przenoszący na losową stronę	101
Przesuwanie tekstu	102
Skrypt 58. Tekst pływający w polu tekstowym w lewo.....	102
Skrypt 59. Tekst pływający w polu tekstowym w prawo	103
Skrypt 60. Tekst pływający w lewo uwzględniający rozmiar pola tekstowego.....	104
Skrypt 61. Symulacja wprowadzania znaków z klawiatury	105
Skrypt 62. Tekst odbijający się od lewej i prawej krawędzi pola tekstowego.....	106
Skrypt 63. Zamiana tekstu przez losowe wstawianie znaków.....	108
Skrypt 64. Zamiana tekstów poprzez najazd od prawej strony.....	110
Skrypt 65. Zamiana tekstów poprzez najazd od lewej strony.....	111
Skrypt 66. Tekst pływający po pasku stanu	113
Skrypt 67. Tekst pływający po pasku tytułu.....	114
Skrypt 68. Pływający zegar.....	114
Skrypt 69. Zegar pływający w wierszu statusu	116
Użytkownicy i hasła	117
Skrypt 70. Kod dostępu do strony	117
Skrypt 71. Kod dostępu do strony ze zliczaniem błędnych prób.....	119
Skrypt 72. Logowanie użytkowników.....	120
Skrypt 73. Logowanie użytkowników ze zliczaniem błędnych prób	122
Skrypt 74. Automatyczne logowanie użytkowników	123
Skrypt 75. Nazwa strony jako hasło.....	126
Skrypt 76. Zapamiętanie danych użytkownika.....	127
Skrypt 77. Zliczanie liczby odwiedzin	129
Skrypt 78. Ograniczenie liczby odwiedzin.....	131
Warstwy i animacje	132
Skrypt 79. Tło pływające w pionie.....	132
Skrypt 80. Tło pływające w poziomie.....	133
Skrypt 81. Tło z cyklicznie zmieniających się obrazów.....	134

Skrypt 82. Pulsująca warstwa (ściemnianie i rozjaśnianie).....	135
Skrypt 83. Pływająca warstwa.....	136
Skrypt 84. Zegar pływający na warstwie HTML	137
Skrypt 85. Tekst płynnie zmieniający kolor	139
Skrypt 86. Skalowanie obrazu.....	142
Skrypt 87. Padający śnieg (prosty efekt).....	144
Skrypt 88. Padający śnieg (efekt zaawansowany).....	145
Skrypt 89. Spadająca piłka	148
Rozdział 2. Aplety	151
Graficzne napisy.....	151
Skrypt 90. Napis pływający w lewo	151
Skrypt 91. Napis pływający w prawo.....	154
Skrypt 92. Napis pływający w poziomie odbijający się od brzegów apletu	156
Skrypt 93. Teksty wpływające raz z jednej, raz z drugiej strony apletu	158
Skrypt 94. Teksty nadjeżdżające od prawej strony.....	162
Skrypt 95. Różnokolorowe teksty nadjeżdżające od prawej strony.....	165
Skrypt 96. Tekst płynący w górę.....	168
Skrypt 97. Tekst płynący w dół.....	172
Skrypt 98. Pływający tekst płynnie zmieniający kolory	174
Skrypt 99. Pulsowanie tekstu	177
Skrypt 100. Wymiana tekstów przez ściemnianie i rozjaśnienie.....	179
Skrypt 101. Kolory płynące przez tekst.....	183
Skrypt 102. Kolory płynące przez przesuwany się tekst.....	187
Skrypt 103. Symulacja pisania na klawiaturze	189
Skrypt 104. Symulacja pisania na klawiaturze z losowym opóźnieniem	192
Skrypt 105. Symulacja pisania na klawiaturze z wymianą tekstów.....	194
Skrypt 106. Prosty napis poruszający się po sinusoidzie.....	198
Skrypt 107. Litery tańczące po sinusoidzie	201
Skrypt 108. Litery płynące po sinusoidzie	204
Skrypt 109. Napisy pojawiające się w losowych punktach	208
Skrypt 110. Napisy pojawiające się w losowych punktach II	210
Skrypt 111. Napis odbijający się od brzegów apletu.....	214
Banery	217
Skrypt 112. Zmieniające się losowo banery	217
Skrypt 113. Banery zmieniające się w określonej kolejności.....	219
Skrypt 114. Baner nadjeżdżający z prawej strony.....	221
Skrypt 115. Baner nadjeżdżający z lewej strony	225
Skrypt 116. Baner nadjeżdżający z góry	228
Skrypt 117. Baner nadjeżdżający z dołu	231
Skrypt 118. Baner zależny od dnia miesiąca.....	234
Skrypt 119. Baner zależny od dnia tygodnia.....	235
Skrypt 120. Baner zależny od pory dnia	237
Skrypt 121. Baner przenoszący na wybraną witrynę.....	238
Hasła dostępowe.....	242
Skrypt 122. Hasło dostępu	242
Skrypt 123. Szyfrowane hasło dostępu	244
Skrypt 124. Aplet szyfrujący hasła.....	247
Skrypt 125. Wiele hasel dostępu	248
Skrypt 126. Wiele szyfrowanych hasel dostępu.....	250
Skrypt 127. Szybkie szyfrowanie wielu hasel	252
Skrypt 128. Nazwa pliku jako hasło dostępu	254
Skrypt 129. Zabezpieczenie hasłem ze zliczaniem błędnych prób.....	256
Skrypt 130. Oddzielna nazwa i hasło dla każdego użytkownika I.....	258

Skrypt 131. Oddzielna nazwa i hasło dla każdego użytkownika II	261
Skrypt 132. Szyfrowana nazwa i hasło użytkownika	264
Skrypt 133. Nazwa i hasło ze zliczaniem błędnych prób	267
Skrypt 134. Indywidualna strona dla każdego użytkownika	269
Odnośniki	271
Skrypt 135. Odnośniki na etykietach.....	271
Skrypt 136. Dynamicznie wyróżniane odnośniki na etykietach.....	275
Skrypt 137. Odnośniki z opisami	278
Skrypt 138. Odnośniki na etykietach ze zmieniającymi się opisami	281
Skrypt 139. Odnośniki na etykietach z opisami w wierszu statusu	285
Skrypt 140. Odnośniki na przyciskach.....	289
Skrypt 141. Odnośniki na przyciskach dynamicznie zmieniających kolor.....	292
Skrypt 142. Przyciski dynamicznie zmieniające kolor tła.....	295
Skrypt 143. Odnośniki otwierane w nowym oknie.....	298
Skrypt 144. Menu.....	300
Skrypt 145. Wielopoziomowe menu	302
Skrypt 146. Menu kontekstowe.....	304
Skrypt 147. Wielopoziomowe menu kontekstowe	306
Skrypt 148. Obrazy jako odnośniki	308
Skrypt 149. Obrazy zmieniające się po najechnaniu myszą.....	311
Data i czas	314
Skrypt 150. Wyświetlenie aktualnej daty	314
Skrypt 151. Data płynąca po ekranie w poziomie	317
Skrypt 152. Data płynąca po ekranie w pionie.....	319
Skrypt 153. Data pływająca w obszarze apletu	322
Skrypt 154. Zegar cyfrowy.....	325
Skrypt 155. Zegar cyfrowy płynący w poziomie.....	327
Skrypt 156. Zegar cyfrowy płynący w pionie	330
Skrypt 157. Zegar cyfrowy pływający w obszarze apletu	333
Skrypt 158. Zegar analogowy	335
Skrypt 159. Zegar analogowy pływający w obszarze apletu.....	339
Skrypt 160. Ile pozostało dni do...?	342
Skrypt 161. Zegar odmierający czas do wybranego zdarzenia	345
Skrypt 162. Zegar odmierający czas do wybranego zdarzenia II.....	349
Skrypt 163. Aplet obliczający liczbę dni między dwiema datami	352
Efekty graficzne	354
Skrypt 164. Generowanie kolorowego tła z płynnymi przejściami tonalnymi	354
Skrypt 165. Animacja wielokolorowego tła	356
Skrypt 166. Pływające wielokolorowe tło.....	358
Skrypt 167. Animacja pokłatkowa	360
Skrypt 168. Padający śnieg	362
Skrypt 169. Obraz odbijający się od brzegów apletu	366
Skrypt 170. Obraz płynący po sinusoidzie.....	368
Skrypt 171. Przycisk samoczynnie zmieniający kolor tła	370
Skrypt 172. Przycisk samoczynnie zmieniający kolor tekstu.....	373
Skrypt 173. Przycisk samoczynnie zmieniający kolor tekstu i tła.....	375
Skrypt 174. Etykieta samoczynnie zmieniająca kolor tekstu.....	377
Skrypt 175. Falowanie obrazu w pionie	379
Skrypt 176. Falowanie obrazu w pionie z możliwością definiowania kierunku ruchu.....	384
Skrypt 177. Obraz falujący w poziomie	387
Skrypt 178. Obraz falujący w poziomie z możliwością definiowania kierunku ruchu.....	390

Skrypt 179. Gwiazdy płynące po ekranie.....	394
Skrypt 180. Gwiazdy warstwowe.....	396
Skrypt 181. Gwiazdy parametryczne	399
Rozdział 3. VBScript.....	403
Rozmaitości.....	403
Skrypt 182. Okno ostrzegawcze.....	403
Skrypt 183. Okno pożegnalne	404
Skrypt 184. Wykrycie typu systemu operacyjnego	405
Skrypt 185. Wybór wyglądu kursora myszy z listy.....	406
Skrypt 186. Przetwarzanie formularzy	407
Skrypt 187. Kalkulator.....	411
Skrypt 188. Automatyczne ustawianie kursora w formularzu.....	415
Skrypt 189. Uniemożliwienie wprowadzania liter w polu tekstowym	416
Skrypt 190. Automatyczna zmiana rozmiaru pola tekstowego.....	417
Skrypt 191. Dynamiczne obrazy	419
Skrypt 192. Sprawdzenie poprawności adresu e-mail	420
Skrypt 193. Ukrycie adresu e-mail przed robotami skanującymi witryny	421
Odnośniki	422
Skrypt 194. Odnośniki na liście rozwijalnej.....	422
Skrypt 195. Odnośniki na liście rozwijalnej z automatyczną zmianą strony	423
Skrypt 196. Odnośniki na liście zwykłej.....	424
Skrypt 197. Odnośniki umieszczone w polach wyboru typu radio.....	426
Skrypt 198. Odnośniki z opisem w polu tekstowym	427
Skrypt 199. Odnośniki z opisem na pasku stanu	429
Skrypt 200. Dodatkowe wyróżnienie odnośników	430
Skrypt 201. Odnośniki na przyciskach z dodatkowym wyróżnieniem	432
Okna.....	434
Skrypt 202. Przekierowanie użytkownika na inną witrynę [VBScript].....	434
Skrypt 203. Przekierowanie z opóźnieniem	434
Skrypt 204. Uniemożliwienie wczytania strony do ramki.....	435
Skrypt 205. Dodanie wpisu do zakładki Ulubione	436
Skrypt 206. Różne wersje witryny w zależności od rozdzielczości ekranu.....	437
Skrypt 207. Otworzenie nowego okna przeglądarki.....	438
Skrypt 208. Dynamiczna zmiana kolorów suwaków okna.....	439
Skrypt 209. Potrząśnięcie ekranem	441
Data i czas	442
Skrypt 210. Określenie bieżącej daty	442
Skrypt 211. Wyświetlenie daty systemowej.....	443
Skrypt 212. Zegar cyfrowy w polu tekstowym	444
Skrypt 213. Zegar cyfrowy w wierszu statusu	446
Skrypt 214. Zegar cyfrowy umieszczony na pasku tytułu.....	447
Skrypt 215. Aktualny czas wyświetlany na przycisku	448
Skrypt 216. Zegar cyfrowy na niezależnej warstwie HTML.....	449
Skrypt 217. Jaki dziś dzień tygodnia?	450
Skrypt 218. Jednoczesne określenie daty, czasu i dnia tygodnia.....	451
Skrypt 219. Obliczenie liczby dni do wybranej daty.....	454
Skrypt 220. Wyświetlenie daty aktualizacji strony	455
Skrypt 221. Uzależnienie wyświetlanej strony od pory dnia.....	456
Skrypt 222. Strona zależna od dnia tygodnia	457
Skrypt 223. Dynamiczny kalendarz	458
Banery	461
Skrypt 224. Losowanie numeru banera.....	461
Skrypt 225. Określenie kolejności wyświetlania banerów	462
Skrypt 226. Zmieniający się losowo baner.....	465

Skrypt 227. Banery zmieniające się w określonej kolejności.....	466
Skrypt 228. Wyświetlanie różnych banerów w zależności od pory dnia.....	468
Skrypt 229. Wyświetlanie różnych banerów w zależności od dnia tygodnia.....	469
Skrypt 230. Baner przenoszący na losową stronę.....	470
Pływające napisy.....	471
Skrypt 231. Przewijanie napisu w lewo.....	471
Skrypt 232. Przewijanie napisu w prawo.....	472
Skrypt 233. Przewijanie napisu z uwzględnieniem rozmiaru pola tekstowego.....	473
Skrypt 234. Wyświetlanie napisu z opóźnieniem przy każdym znaku.....	475
Skrypt 235. Pływający napis odbijający się od brzegów pola tekstowego.....	476
Skrypt 236. Losowa zamiana znaków w napisie.....	477
Skrypt 237. Napisy wjeżdżające do pola tekstowego z prawej strony.....	479
Skrypt 238. Napisy wjeżdżające do pola tekstowego z lewej strony.....	481
Skrypt 239. Przewijanie napisu na pasku statusu.....	483
Skrypt 240. Przewijanie tekstu na pasku tytułu.....	484
Skrypt 241. Zegar pływający w polu tekstowym.....	485
Skrypt 242. Zegar pływający na pasku stanu.....	487
Użytkownicy i hasła.....	488
Skrypt 243. Strona dostępna po podaniu hasła.....	488
Skrypt 244. Hasło dostępu do strony ze zliczaniem błędnych prób.....	490
Skrypt 245. Logowanie użytkowników.....	491
Skrypt 246. Logowanie użytkowników ze zliczaniem błędnych prób.....	493
Skrypt 247. Automatyczne logowanie użytkowników.....	494
Skrypt 248. Nietypowe hasło.....	497
Skrypt 249. Rozpoznawanie użytkownika.....	498
Skrypt 250. Zliczanie liczby odwiedzin.....	501
Skrypt 251. Ograniczenie liczby odwiedzin.....	503
Efekty graficzne.....	505
Skrypt 252. Pływające tło.....	505
Skrypt 253. Tło z cyklicznie zmieniających się obrazów.....	506
Skrypt 254. Pulsujące tło.....	507
Skrypt 255. Ściemnianie koloru warstwy.....	508
Skrypt 256. Rozjaśnianie koloru warstwy.....	509
Skrypt 257. Pulsująca warstwa.....	510
Skrypt 258. Warstwa pływająca w oknie przeglądarki.....	511
Skrypt 259. Zegar pływający w oknie przeglądarki.....	513
Skrypt 260. Płynne powiększanie obrazu.....	515
Skrypt 261. Pulsowanie obrazu.....	516
Skrypt 262. Symulacja odbijającej się piłki.....	517
Rozdział 4. PHP.....	521
Globalna sieć.....	521
Skrypt 263. Pobranie pliku z serwera ftp.....	521
Skrypt 264. Wysłanie pliku do serwera ftp.....	522
Skrypt 265. Wysłanie pliku przez formularz (upload pliku).....	524
Skrypt 266. Wysłanie pliku ze strony WWW do serwera ftp.....	526
Skrypt 267. Przekierowanie przy użyciu znacznika META.....	527
Skrypt 268. Przekierowanie na inną witrynę przy użyciu nagłówka HTTP.....	527
Skrypt 269. Przekierowanie na losową witrynę.....	528
Skrypt 270. Inna strona dla użytkowników o znanych IP.....	529
Skrypt 271. Walidacja adresu e-mail.....	529
Skrypt 272. Pobranie pliku (download pliku).....	530
Skrypt 273. Pobieranie plików z listy I.....	531
Skrypt 274. Pobieranie plików z listy II.....	534

Skrypt 275. Generowanie listy plików do pobrania	537
Skrypt 276. Wysłanie ze strony WWW listu elektronicznego	537
Skrypt 277. Wysłanie listu z walidacją adresów	540
Skrypt 278. Wysłanie listu z ograniczeniem rozmiaru	541
Skrypt 279. Odczytanie adresu IP komputera	543
Skrypt 280. Lista plików z opisami	544
Skrypt 281. Lista plików z identyfikatorami i opisami	546
Skrypt 282. Sortowana lista plików.....	548
Rozmaitości.....	551
Skrypt 283. Strona zależna od adresu IP	551
Skrypt 284. Rozpoznanie typu przeglądarki.....	552
Skrypt 285. Strona zależna od typu przeglądarki	553
Skrypt 286. Strona zależna od systemu operacyjnego.....	555
Skrypt 287. Ochrona adresu e-mail przed spamem	556
Skrypt 288. Ochrona adresu e-mail z wykorzystaniem pliku graficznego	557
Skrypt 289. Lista odwiedzin strony.....	559
Skrypt 290. Ocenzowanie tekstu.....	561
Skrypt 291. Ocenzowanie tekstu z wykorzystaniem zewnętrznego słownika.....	562
Skrypt 292. Lista odnośników.....	562
Skrypt 293. Lista odnośników z opisami.....	564
Skrypt 294. Sortowana lista odnośników	565
Użytkownicy i hasła	566
Skrypt 295. Hasło dostępu do strony.....	566
Skrypt 296. Wiele haseł dostępu	568
Skrypt 297. Wiele haseł dostępu II	569
Skrypt 298. Strona zależna od kodu dostępu.....	569
Skrypt 299. Logowanie użytkowników.....	571
Skrypt 300. Logowanie użytkowników z kodowaniem danych	574
Skrypt 301. Zarządzanie hasłami	575
Skrypt 302. Logowanie użytkowników z danymi w kodzie skryptu	579
Skrypt 303. Generowanie losowego hasła.....	580
Skrypt 304. Zablokowanie wybranych adresów IP	580
Skrypt 305. Zablokowanie wybranych adresów IP z danymi w pliku.....	581
Skrypt 306. Dopuszczenie jedynie wybranych adresów IP	582
Skrypt 307. Dopuszczenie jedynie wybranych adresów IP II	583
Skrypt 308. Automatyczne logowanie.....	584
Skrypt 309. Strona zależna od nazwy użytkownika	587
Skrypt 310. Powiązanie nazwy użytkownika z adresem IP.....	589
Skrypt 311. Zliczanie liczby odwiedzin każdego użytkownika.....	590
Skrypt 312. Ograniczenie liczby odwiedzin ze względu na adres IP	592
Data i czas	594
Skrypt 313. Wyświetlenie bieżącej daty i czasu.....	594
Skrypt 314. Wyświetlenie daty ostatniej modyfikacji strony	596
Skrypt 315. Wyświetlenie dnia tygodnia.....	597
Skrypt 316. Strona zależna od pory dnia.....	598
Skrypt 317. Strona zależna od dnia tygodnia	599
Skrypt 318. Odliczanie dni.....	600
Skrypt 319. Różnica między datami.....	601
Skrypt 320. Czas generowania strony	603
Skrypt 321. Kalendarz.....	604
Banery	607
Skrypt 322. Losowy baner	607
Skrypt 323. Losowy baner II (dane w tablicy)	608
Skrypt 324. Baner losowany z plików z wybranego katalogu.....	608

Skrypt 325. Zapamiętanie kolejności wyświetlania banerów.....	609
Skrypt 326. Losowe banery bez powtórzeń.....	610
Skrypt 327. Baner zależny od pory dnia	612
Skrypt 328. Baner zależny od dnia tygodnia.....	612
Skrypt 329. Baner zależny od adresu IP.....	613
Liczniki, księgi gości itp.	614
Skrypt 330. Prosty licznik tekstowy.....	614
Skrypt 331. Licznik graficzny	615
Skrypt 332. Licznik filtrujący adresy IP.....	616
Skrypt 333. Licznik uwzględniający datę początkową.....	617
Skrypt 334. Licznik uwzględniający tylko jedno odwołanie z jednego adresu IP...	618
Skrypt 335. Księga gości.....	619
Skrypt 336. Księga gości z nawigacją.....	623
Skrypt 337. Księga gości z filtrowaniem znaczników HTML.....	627
Skrypt 338. Ankieta	630
Skrypt 339. Porada dnia	635
Skrypt 340. Porada dnia z hasłami w pliku	635
Bazy danych.....	636
Skrypt 341. Licznik.....	636
Skrypt 342. Ankieta	638
Skrypt 343. Liczba osób obecnych na stronie	640
Skrypt 344. Liczba osób przeglądających stronę z różnych adresów IP	643
Skrypt 345. Księga gości.....	644
Skrypt 346. Logowanie użytkowników.....	648
Skrypt 347. Logowanie z filtrowaniem danych i kodowaniem haseł	650
Skrypt 348. Zarządzanie hasłami	652
Skrypt 349. Zapamiętanie danych użytkownika.....	654
Skrypt 350. Statystyka odwiedzin	660
Skrypt 351. Liczba osób obecnych na stronie II	662
Skrypt 352. Pobieranie plików	664
Skrypt 353. Ranking plików.....	667
Skrypt 354. Wprowadzenie plików do bazy danych	671
Skrypt 355. Porada dnia	672
Skrypt 356. Lista odnośników.....	674
Skrypt 357. Lista odnośników z opisami.....	675
Skrypt 358. Ograniczenie liczby odwiedzin dla tych samych adresów IP.....	677
Skrypt 359. Ograniczenie liczby odwiedzin dla wybranych użytkowników	679
Skrypt 360. Zliczanie liczby odwołań z każdego adresu IP	682
Skrypt 361. Statystyka przeglądarek	684
Skrypt 362. Statystyka przeglądarek na podstawie statystyki strony	686
Skrypt 363. Statystyka systemów na podstawie statystyki strony	689
Grafika.....	691
Skrypt 364. Galeria plików graficznych.....	691
Skrypt 365. Galeria z podpisami	693
Skrypt 366. Galeria z miniaturami obrazów.....	696
Skrypt 367. Przeskalowanie obrazu	697
Skrypt 368. Obracanie obrazu	699
Skrypt 369. Przeskalowanie serii obrazów.....	700
Rozdział 5. ASP — Active Server Pages.....	703
Internet.....	703
Skrypt 370. Odczytanie adresu IP komputera osoby odwiedzającej stronę.....	703
Skrypt 371. Zablockowanie wybranych adresów IP	704
Skrypt 372. Zablockowanie wybranych adresów IP II (dane w pliku)	705

Skrypt 373. Zablokowanie wybranych adresów IP z symulacją braku dokumentu.....	706
Skrypt 374. Dopuszczenie jedynie wybranych adresów IP	708
Skrypt 375. Dopuszczenie jedynie wybranych adresów IP II (dane w pliku tekstowym).....	709
Skrypt 376. Przekierowanie na inną witrynę	710
Skrypt 377. Przekierowanie z opóźnieniem	711
Skrypt 378. Rozpoznanie typu przeglądarki.....	712
Skrypt 379. Różne wersje strony zależne od typu przeglądarki	713
Skrypt 380. Rozpoznanie systemu operacyjnego klienta	714
Skrypt 381. Różne wersje strony zależne od typu systemu operacyjnego.....	715
Skrypt 382. Inna wersja strony dla wybranych adresów IP.....	715
Skrypt 383. Różne wersje strony w zależności od adresu IP użytkownika	716
Skrypt 384. Wysłanie danych z formularza na adres e-mail.....	717
Skrypt 385. Wysłanie listu elektronicznego	720
Skrypt 386. Wysłanie wiadomości e-mail z ograniczeniem rozmiaru listu.....	723
Skrypt 387. Wysłanie listu elektronicznego z wykorzystaniem obiektu CDONTS.....	724
Skrypt 388. Sprawdzenie poprawności adresu e-mail	726
Skrypt 389. Proste wysłanie pliku do przeglądarki	727
Skrypt 390. Wysłanie pliku do przeglądarki z wykorzystaniem ADODB	729
Skrypt 391. System pobierania plików.....	730
Skrypt 392. System pobierania plików z filtrowaniem danych	733
Skrypt 393. Pobieranie plików z danymi w pliku tekstowym	735
Skrypt 394. Pobieranie plików z opisami.....	738
Skrypt 395. Pobieranie plików z identyfikatorami	741
Skrypt 396. Pobieranie plików z identyfikatorami i opisami.....	745
Skrypt 397. Pobieranie plików z tabelarycznym wyświetlaniem danych.....	747
Skrypt 398. Sortowana lista plików do pobrania.....	751
Systemy logowania (z danymi w systemie plików).....	755
Skrypt 399. Strona zabezpieczona kodem dostępu (z przekierowaniem).....	755
Skrypt 400. Strona zabezpieczona kodem dostępu (bez przekierowania)	756
Skrypt 401. Strona akceptująca wiele kodów dostępu	757
Skrypt 402. Różne wersje strony w zależności od kodu dostępowego.....	758
Skrypt 403. System logowania użytkowników	759
Skrypt 404. System logowania użytkowników z danymi zapisanymi w pliku tekstowym	761
Skrypt 405. System logowania zapamiętujący stronę startową	762
Skrypt 406. Tworzenie losowego hasła.....	764
Skrypt 407. System logowania uwzględniający adres IP użytkownika.....	765
Skrypt 408. System logowania blokujący wybrane adresy IP.....	767
Skrypt 409. System logowania dopuszczający jedynie wybrane adresy IP.....	769
Skrypt 410. Zliczanie liczby odwiedzin pojedynczego użytkownika.....	771
Skrypt 411. Ograniczenie liczby logowań.....	774
Data i czas	776
Skrypt 412. Wyświetlanie aktualnej daty	776
Skrypt 413. Wyświetlenie daty uwzględniające ustawienia systemowe serwera	778
Skrypt 414. Wyświetlenie nazwy aktualnego dnia tygodnia.....	778
Skrypt 415. Pełne określenie daty	779
Skrypt 416. Różne wersje strony I (w zależności od pory dnia).....	780
Skrypt 417. Różne wersje strony II (w zależności od dnia tygodnia).....	782
Skrypt 418. Dynamicznie generowany kalendarz	782
Skrypt 419. Obliczenie czasu generowania strony	786
Skrypt 420. Odliczanie.....	787
Skrypt 421. Określenie daty aktualizacji strony	789

Liczniki, statystyki itp.	791
Skrypt 422. Licznik odwiedzin	791
Skrypt 423. Licznik graficzny	792
Skrypt 424. Licznik odrzucający połączenia z wybranych adresów IP I.....	793
Skrypt 425. Licznik odrzucający połączenia z wybranych adresów IP II	795
Skrypt 426. Licznik zliczający odwołania jedynie z wybranych adresów IP I.....	797
Skrypt 427. Licznik zliczający odwołania jedynie z wybranych adresów IP II.....	798
Skrypt 428. Licznik uwzględniający tylko różne adresy IP	800
Skrypt 429. Lista odwiedzin strony.....	802
Skrypt 430. Lista odwiedzin strony z ograniczeniem liczby wyświetlanych wpisów	804
Skrypt 431. Lista odwiedzin strony uwzględniająca jedynie wybrane adresy IP	805
Skrypt 432. Lista odwiedzin strony odrzucająca odwołania z wybranych adresów IP	807
Skrypt 433. Lista odwiedzin uwzględniająca tylko różne adresy IP	809
Skrypt 434. Statystyka przeglądarek	812
Skrypt 435. Statystyka systemów operacyjnych	814
Skrypt 436. Księga gości.....	816
Skrypt 437. Księga gości odrzucająca wybrane adresy IP	820
Skrypt 438. Księga gości filtrująca znaczniki HTML	822
Skrypt 439. Porada dnia	823
Bazy danych.....	824
Skrypt 440. Licznik.....	824
Skrypt 441. Licznik umożliwiający ustawienie daty początkowej	826
Skrypt 442. Licznik odrzucający połączenia z wybranych adresów IP	827
Skrypt 443. Licznik zliczający odwołania jedynie z wybranych adresów IP	829
Skrypt 444. Licznik uwzględniający tylko różne adresy IP	831
Skrypt 445. Zliczanie liczby odwołań z każdego adresu IP	832
Skrypt 446. Statystyka strony.....	834
Skrypt 447. Statystyka przeglądarek	836
Skrypt 448. Statystyka przeglądarek na podstawie statystyki strony	839
Skrypt 449. Statystyka systemów operacyjnych	841
Skrypt 450. Statystyka systemów na podstawie statystyki strony	843
Skrypt 451. Księga gości.....	845
Skrypt 452. Księga gości z ograniczeniem wpisów z pojedynczego adresu IP	849
Skrypt 453. Liczba osób obecnych na stronie	851
Skrypt 454. Zablokowanie wybranych adresów IP	852
Skrypt 455. Dopuszczenie jedynie wybranych adresów IP	854
Skrypt 456. Wysyłanie e-maili z ograniczeniem czasowym	855
Skrypt 457. Ankieta	858
Skrypt 458. Ankieta ograniczająca głosowanie z pojedynczego adresu IP	861
Skrypt 459. Ankieta czasowo ograniczająca oddawanie głosów z jednego adresu IP	864
Rozdział 6. Perl	867
Internet	867
Skrypt 460. Adres IP.....	867
Skrypt 461. Typ przeglądarki	868
Skrypt 462. Typ systemu operacyjnego	869
Skrypt 463. Przekierowanie z opóźnieniem	870
Skrypt 464. Przekierowanie ze względu na adres IP	871
Skrypt 465. Przekierowanie ze względu na typ przeglądarki	871
Skrypt 466. Blokada użytkowników I	872
Skrypt 467. Blokada użytkowników II.....	873

Skrypt 468. Ograniczenie dostępu dopuszczające jedynie wybrane adresy IP	874
Skrypt 469. Wysyłanie poczty.....	875
Skrypt 470. Walidacja adresu e-mail.....	878
Skrypt 471. Wysyłanie poczty z walidacją adresów.....	879
Skrypt 472. Wysyłanie poczty z ograniczeniem długości listu	880
Skrypt 473. Wysyłanie pliku.....	882
Skrypt 474. Pobieranie plików I.....	883
Skrypt 475. Pobieranie plików II	885
Skrypt 476. Pobranie pliku z serwera ftp	887
Skrypt 477. Wysłanie pliku do serwera.....	889
Data i czas	891
Skrypt 478. Wyświetlenie aktualnej daty	891
Skrypt 479. Wyświetlenie aktualnego dnia tygodnia	892
Skrypt 480. Czas generowania strony	893
Skrypt 481. Obliczenie liczby dni pozostałych do wybranej daty	894
Skrypt 482. Obliczanie różnicy dni pomiędzy dwoma datami	896
Skrypt 483. Strona zależna od pory dnia.....	898
Skrypt 484. Strona zależna od dnia tygodnia	899
Liczniki	899
Skrypt 485. Licznik tekstowy.....	899
Skrypt 486. Licznik graficzny	900
Skrypt 487. Licznik odrzucający niektóre odwołania I.....	901
Skrypt 488. Licznik odrzucający niektóre odwołania II.....	903
Skrypt 489. Licznik zliczający tylko jedno odwołanie z jednego adresu IP.....	904
Skrypt 490. Licznik przechowujący datę początkową.....	906
Banery	907
Skrypt 491. Losowy baner	907
Skrypt 492. Losowy baner II.....	907
Skrypt 493. Baner losowany z wybranego katalogu.....	908
Skrypt 494. Baner zależny od pory dnia	909
Skrypt 495. Baner zależny od dnia tygodnia.....	910
Skrypt 496. Wyświetlanie banera w zależności od adresu IP.....	911
Autoryzacje	912
Skrypt 497. Pojedyncze hasło dostępu do strony	912
Skrypt 498. Wiele haseł dostępu do strony I.....	913
Skrypt 499. Wiele haseł dostępu do strony II.....	914
Skrypt 500. Różne witryny w zależności od hasła dostępu.....	915
Skrypt 501. Logowanie użytkowników I	916
Skrypt 502. Logowanie użytkowników II	917
Skrypt 503. Logowanie uwzględniające adres IP użytkownika	919
Skrypt 504. Różne witryny w zależności od nazwy użytkownika.....	920
Skrypt 505. Logowanie blokujące niektóre adresy IP	921
Skorowidz.....	923

Rozdział 2.

Aplety

Graficzne napisy

Skrypt 90. Napis pływający w lewo

Wykorzystując język programowania Java, możemy napisać aplet, który będzie realizował efekt płynnego przewijania tekstu w poziomie. Tekst ten będzie mógł mieć dowolną wielkość, kolor czy krój czcionki. Takie zadanie realizuje klasa `FlowString` przedstawiona na listingu 90B. Aplet powinien zostać osadzony w kodzie HTML za pomocą znacznika `<APLET>` przedstawionego na listingu 90A. W tym znaczniku powinny zostać uwzględnione następujące parametry:

- ◆ `text` — napis, który będzie przewijany, wartość domyślna: Brak parametru `text`!
- ◆ `fontsize` — wielkość czcionki, wartość domyślna: 36.
- ◆ `speedstep` — liczba pikseli, o którą ma zostać przesunięty tekst w każdym ruchu, wartość domyślna: 1.

Listing 90A. *Kod HTML z zagnieżdżonym znacznikiem `<APPLET>`*

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<APPLET
    code      = "FlowString.class"
    text      = "Przykładowy tekst..."
    fontsize  = "30"
    speedstep = "1"
    width     = "640"
    height    = "100">
</APPLET>
</BODY>
</HTML>
```

Listing 90B. *Poziomy napis przesuwany w lewą stronę*

```
/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis          http://marcinlis.com */

import java.awt.*;
import java.applet.*;

public
class FlowString extends Applet implements Runnable
{
    int x, y, speedStep;
    int appWidth, appHeight;
    int strWidth, strHeight;
    String napis;
    Font fontTimesRoman;
    Image img;
    Graphics gDC, mDC;
    boolean stopped = false;
    public void init()
    {
        if((napis = getParameter("text")) == null)
            napis = "Brak parametru text!";

        String strFontSize;
        int fontSize;
        if((strFontSize = getParameter("fontsize")) == null)
            strFontSize = "36";

        try{
            fontSize = Integer.parseInt(strFontSize);
        }
        catch(Exception e){
            fontSize = 36;
        }

        String strSpeedStep;
        if((strSpeedStep = getParameter("speedstep")) == null)
            strSpeedStep = "1";

        try{
            speedStep = Math.abs(Integer.parseInt(strSpeedStep));
        }
        catch(Exception e){
            speedStep = 1;
        }

        appWidth = getWidth();
        appHeight = getHeight();

        fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
        img = createImage(appWidth, appHeight);
        mDC = img.getGraphics();
        gDC = getGraphics();

        gDC.setFont(fontTimesRoman);
        FontMetrics fm = gDC.getFontMetrics();
        strHeight = fm.getHeight();
    }
}
```

```
        strWidth = fm.stringWidth(napis);

        x = appWidth;
        y = (int)((appHeight + strHeight) / 2);
    }
    public void start()
    {
        Thread thread = new Thread (this);
        thread.start();
    }
    public void stop()
    {
        stopped = true;
    }
    public void run()
    {
        while (!stopped){
            if (x < -strWidth){
                x = appWidth;
            }
            try{
                Thread.sleep(1);
            }
            catch (InterruptedException e){
            }
            x -= speedStep;
            mDC.setColor(Color.WHITE);
            mDC.fillRect(0, 0, appWidth, appHeight);
            mDC.setColor (Color.blue);
            mDC.setFont (fontTimesRoman);
            mDC.drawString (napis, x, y);
            gDC.drawImage (img, 0, 0, this);
        }
    }
}
```

W przypadku, gdyby zabrakło któregoś z parametrów, w aplecie zostanie przyjęta odpowiadająca temu parametrowi wartość domyślna. Oprócz tego znacznik <APLET> powinien zawierać standardowe argumenty:

- ♦ code — nazwa klasy zawierającej kod apletu,
- ♦ width — szerokość apletu,
- ♦ height — wysokość apletu.

Wykonywanie kodu rozpoczyna się od metody `init`, która odczytuje wszystkie argumenty i przypisuje je odpowiadającym im zmiennym. W razie konieczności są wykorzystywane wymienione wyżej wartości domyślne. Oprócz tego w metodzie `init` są wykonywane wstępne obliczenia oraz tworzone niezbędne obiekty. Za pomocą metod obiektu klasy `FontMetrics` jest pobierana wysokość (metoda `getHeight`) i szerokość (metoda `stringWidth`) napisu, jest także obliczane jego położenie w pionie, tak aby znalazł się on po środku apletu — w tym celu jest wykorzystywany wzór:

$$\text{pozycja w pionie} = (\text{wysokość apletu} + \text{wysokość napisu}) / 2$$

Jest także tworzony nowy font, który będzie wykorzystywany do rysowania napisu. Korzystamy z czcionki z rodziny TimesRoman, która powinna być dostępna w większości systemów. Można się również pokusić o przekazywanie nazwy czcionki w postaci parametru apletu (tak jak np. jej rozmiar). W kodzie z listingu 90B pozostawiamy jednak tę wartość zapisaną na stałe w kodzie. W metodzie `init` jest również tworzony, za pomocą metody `createImage`, obraz buforowy, który jest przypisywany zmiennej o nazwie `img`. Wszystkie operacje rysowania będą wykonywane na tym obrazie, a dopiero po ich całkowitym zakończeniu obraz ten będzie wyświetlany na powierzchni apletu.

Kod wątku realizującego procedurę animacji napisu jest zawarty w metodzie `run`. Znajduje się tam pętla `while`, która działa non stop, aż do zakończenia pracy apletu, co jest sygnalizowane zmianą stanu zmiennej `stopped` z `false` na `true`. W każdym przebiegu pętli współrzędna `x` napisu jest zmniejszana o wartość zapisaną w zmiennej `speedStep`, kontrolującej szybkość i płynność ruchu. Następnie jest wykonywane wyczyszczenie obszaru rysowania (`mDC.clearRect(0, 0, appWidth, appHeight);`), ustawienie koloru i czcionki, narysowanie napisu (`mDC.drawString(napis, x, y);`) oraz przeniesienie obrazu na obszar apletu (`gDC.drawImage(img, 0, 0, this);`). W każdym przebiegu jest również sprawdzane, czy współrzędna `x` napisu nie jest mniejsza od wartości `-strWidth`, co by oznaczało, że napis w całości znalazł się za lewym brzegiem obszaru apletu. W takiej sytuacji zmieniamy współrzędną `x`, tak aby napis został przesunięty całkowicie za prawy brzeg apletu, co jest równoznaczne z ponownym rozpoczęciem przesuwania napisu od prawej strony do lewej.

Skrypt 91. Napis pływający w prawo

Opierając się na kodzie apletu z przykładu 90., który wykonywał animację napisu pływającego w lewą stronę, możemy utworzyć aplet, w którym animacja będzie się odbywała w przeciwnym kierunku. Zmiany nie będą duże. W procedurze `init` musimy zmienić początkową pozycję napisu, kontrolowaną przez zmienną `x`. W przypadku przykładu z listingu 90B była to pozycja za prawym brzegiem apletu (czyli współrzędna o wartości `appWidth + strWidth`), tym razem początkowa pozycja musi się znaleźć za lewym brzegiem apletu, czyli będzie to po prostu `-strWidth`. Kolejne zmiany muszą nastąpić w metodzie `run`. Obecnie pozycja napisu, czyli współrzędna `x`, musi być w każdym ruchu zwiększana o wartość zapisaną w zmiennej `speedStep`, czyli działanie będzie miało postać: `x += speedStep;`. W inny sposób trzeba także kontrolować wyjście napisu poza obszar apletu. Napis zniknie bowiem całkowicie z ekranu z prawej strony, kiedy zostanie spełniony warunek `x > appWidth + strWidth`. Po wprowadzeniu wszystkich niezbędnych zmian otrzymamy kod widoczny na listingu 91.

Listing 91. Poziomy napis przesuwany w prawą stronę

```
/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis http://marcinlis.com */

import java.awt.*;
import java.applet.*;

public
class FlowString extends Applet implements Runnable
```



```
{
    int x, y, speedStep;
    int appWidth, appHeight;
    int strWidth, strHeight;
    String napis;
    Font fontTimesRoman;
    Image img;
    Graphics gDC, mDC;
    boolean stopped = false;
    public void init()
    {
        if((napis = getParameter("text")) == null)
            napis = "Brak parametru text!";

        String strFontSize;
        int fontSize;
        if((strFontSize = getParameter("fontsize")) == null)
            strFontSize = "36";

        try{
            fontSize = Integer.parseInt(strFontSize);
        }
        catch(Exception e){
            fontSize = 36;
        }

        String strSpeedStep;
        if((strSpeedStep = getParameter("speedstep")) == null)
            strSpeedStep = "1";

        try{
            speedStep = Math.abs(Integer.parseInt(strSpeedStep));
        }
        catch(Exception e){
            speedStep = 1;
        }

        appWidth = getWidth();
        appHeight = getHeight();

        fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
        img = createImage(appWidth, appHeight);
        mDC = img.getGraphics();
        gDC = getGraphics();

        gDC.setFont(fontTimesRoman);
        FontMetrics fm = gDC.getFontMetrics();
        strHeight = fm.getHeight();
        strWidth = fm.stringWidth(napis);

        x = -strWidth;
        y = (int)((appHeight + strHeight) / 2);
    }
    public void start()
    {
        Thread thread = new Thread (this);
        thread.start();
    }
}
```

```

    }
    public void stop()
    {
        stopped = true;
    }
    public void run()
    {
        while (!stopped){
            if (x > appWidth){
                x = -strWidth;
            }
            try{
                Thread.sleep(1);
            }
            catch (InterruptedException e){
            }
            x += speedStep;
            mDC.setColor(Color.white);
            mDC.fillRect(0, 0, appWidth, appHeight);
            mDC.setColor (Color.blue);
            mDC.setFont (fontTimesRoman);
            mDC.drawString (napis, x, y);
            gDC.drawImage (img, 0, 0, this);
        }
    }
}

```

Skrypt 92. Napis pływający w poziomie odbijający się od brzegów apletu

Aplet z przykładu 90. pozwalał na płynną animację przewijania napisu w lewą stronę, natomiast aplet z przykładu 91. — na przewijanie napisu w prawą stronę. Jeśli przedstawione w tych przykładach techniki połączymy ze sobą, możemy uzyskać efekt, w którym napis będzie przesuwał się raz jedną, raz w drugą stronę, odbijając się od brzegów obszaru apletu. Takie zadanie realizuje kod zaprezentowany na listingu 92.

Listing 92. *Napis pływający w poziomie odbijający się od brzegów ekranu*

```

/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis          http://marcinlis.com */

import java.awt.*;
import java.applet.*;

public
class FlowString extends Applet implements Runnable
{
    int x, y, speedStep;
    int appWidth, appHeight;
    int strWidth, strHeight;
    String napis;
    Font fontTimesRoman;
    Image img;
}

```

```
Graphics gDC, mDC;
boolean stopped = false;
public void init()
{
    if((napis = getParameter("text")) == null)
        napis = "Brak parametru text!";

    String strFontSize;
    int fontSize;
    if((strFontSize = getParameter("fontsize")) == null)
        strFontSize = "36";

    try{
        fontSize = Integer.parseInt(strFontSize);
    }
    catch(Exception e){
        fontSize = 36;
    }

    String strSpeedStep;
    if((strSpeedStep = getParameter("speedstep")) == null)
        strSpeedStep = "1";

    try{
        speedStep = Integer.parseInt(strSpeedStep);
    }
    catch(Exception e){
        speedStep = 1;
    }

    appWidth = getWidth();
    appHeight = getHeight();

    fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
    img = createImage(appWidth, appHeight);
    mDC = img.getGraphics();
    gDC = getGraphics();

    gDC.setFont(fontTimesRoman);
    FontMetrics fm = gDC.getFontMetrics();
    strHeight = fm.getHeight();
    strWidth = fm.stringWidth(napis);

    x = 0;
    y = (int)((appHeight + strHeight) / 2);
}
public void start()
{
    Thread thread = new Thread (this);
    thread.start();
}
public void stop()
{
    stopped = true;
}
public void run()
{
```

```

while (!stopped){
    if ((x > appWidth - strWidth) || (x < 0)){
        speedStep = -speedStep;
    }
    try{
        Thread.sleep(1);
    }
    catch (InterruptedException e){
    }
    x += speedStep;
    mDC.setColor(Color.white);
    mDC.fillRect(0, 0, appWidth, appHeight);
    mDC.setColor (Color.blue);
    mDC.setFont (fontTimesRoman);
    mDC.drawString (napis, x, y);
    gDC.drawImage (img, 0, 0, this);
}
}
}

```

W stosunku do poprzednich przykładów nie musimy wprowadzać wielu zmian. Procedura `init` pozostanie w swej dotychczasowej postaci, jedyną zmianą będzie początkowa wartość przypisywana zmiennej `x` kontrolującej położenie napisu. Tym razem będzie to zero, czyli po uruchomieniu apletu napis będzie wyświetlony przy jego lewym brzegu. W metodzie `run` trzeba nieco zmienić sposób przesuwania napisu — otóż w każdym przebiegu do zmiennej `x` będzie dodawana wartość zmiennej `speedStep`. Kierunek ruchu będzie kontrolowany przez znak wartości zapisanej w zmiennej `speedStep`, tzn. kiedy wartość będzie dodatnia, napis będzie przesuwany się w prawo, natomiast kiedy wartość będzie ujemna, napis będzie przesuwany się w lewo. Zmiana znaku, a tym samym kierunku ruchu, będzie odbywała się po każdorazowym osiągnięciu prawego lub lewego brzegu apletu, czyli kiedy będzie spełniony jeden z warunków: `x > appWidth - strWidth` lub `x < 0`.

Skrypt 93. Teksty wpływające raz z jednej, raz z drugiej strony apletu

Skrypty 90. – 92. pokazywały różne efekty pływającego tekstu, w każdym przypadku do dyspozycji był jednak tylko jeden napis. Jeśli tekstów miałoby być więcej, co jest często spotykanym efektem, trzeba by zastosować nieco inną realizację apletu. Przykład widoczny na listingu 93B pozwala na zastosowanie dowolnej liczby tekstów (nie mniej niż dwóch), przekazywanych jako parametry apletu, które pływają naprzemiennie, raz w lewą, raz w prawą stronę.

Niezbędne jest w tym przypadku zastosowanie dodatkowych argumentów znacznika `<APPLET>`, jego przykładowy kod jest widoczny na listingu 93A. Znaczenie argumentów `code`, `width`, `height`, `fontSize`, `speedStep` zostało wyjaśnione w opisie apletu z przykładu 90. Argumentem dodatkowym jest `textsNo`, który definiuje liczbę tekstów, oraz argumenty w postaci `textN`, gdzie `N` to liczba określająca kolejny numer tekstu. Wartości argumentów są odczytywane w metodzie `init`, a odczytane teksty są zapisywane w tablicy `textsArr`.

Listing 93A. *Kod HTML ze znacznikiem <APPLET> z dodatkowymi parametrami*

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<APPLET
    code      = "FlowString.class"
    text1     = "Przykładowy tekst..."
    text2     = "Jeszcze jeden tekst..."
    text3     = "Kolejny przykład tekstu..."
    textsNo   = "3"
    fontsize  = "30"
    speedstep = "1"
    width     = "640"
    height    = "100">
</APPLET>
</BODY>
</HTML>
```

Listing 93B. *Teksty wypływające naprzemiennie z lewej i prawej strony apletu*

```
/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis          http://marcinlis.com */

import java.awt.*;
import java.applet.*;
import java.util.*;

public
class FlowString extends Applet implements Runnable
{
    int mX, y, speedStep, currText = 0;
    int appWidth, appHeight;
    int strWidth, strHeight, strMaxDescent;
    int direction = 0;
    String[] textsArr;
    String sText, mText;
    Font fontTimesRoman;
    Image img;
    Graphics gDC, mDC;
    boolean stopped = false;
    public void init()
    {
        String strTextsNo;
        int textsNo;

        if((strTextsNo = getParameter("textsNo")) == null)
            strTextsNo = "2";

        try{
            textsNo = Integer.parseInt(strTextsNo);
        }
        catch(Exception e){
            textsNo = 2;
        }
    }
}
```

```
textsArr = new String[textsNo];

String str;

for(int i = 1; i <= textsNo; i++){
    if((str = getParameter("text" + i)) == null)
        str = "Brak parametru text" + i + "!";
    textsArr[i - 1] = str;
}

String strFontSize;
int fontSize;
if((strFontSize = getParameter("fontsize")) == null)
    strFontSize = "36";

try{
    fontSize = Integer.parseInt(strFontSize);
}
catch(Exception e){
    fontSize = 36;
}

String strSpeedStep;
if((strSpeedStep = getParameter("speedstep")) == null)
    strSpeedStep = "1";

try{
    speedStep = Math.abs(Integer.parseInt(strSpeedStep));
}
catch(Exception e){
    speedStep = 1;
}

mText = textsArr[0];

appWidth = getWidth();
appHeight = getHeight();

fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
img = createImage(appWidth, appHeight);
mDC = img.getGraphics();
gDC = getGraphics();

gDC.setFont(fontTimesRoman);
resetText();
}
public void start()
{
    Thread thread = new Thread (this);
    thread.start();
}
public void stop()
{
    stopped = true;
}
public void resetText()
{
```

```
FontMetrics fm = gDC.getFontMetrics();
strHeight = fm.getHeight();
strWidth = fm.stringWidth(mText);
strMaxDescent = fm.getMaxDescent();
if (direction == 0){
    mX = appWidth;
}
else if (direction == 1){
    mX = -strWidth;
}
y = (int)((appHeight + strHeight) / 2);
}
public void run()
{
    while (!stopped){
        if(direction == 0){
            if (mX < -strWidth){
                if(++currText >= textsArr.length)
                    currText = 0;
                mText = textsArr[currText];
                direction = 1;
                resetText();
            }
            mX -= speedStep;
        }
        else if (direction == 1){
            if (mX > appWidth){
                if(++currText >= textsArr.length)
                    currText = 0;
                mText = textsArr[currText];
                direction = 0;
                resetText();
            }
            mX += speedStep;
        }

        try{
            Thread.sleep(1);
        }
        catch (InterruptedException e){
        }
        mDC.setColor(Color.white);
        mDC.fillRect(0, 0, appWidth, appHeight);
        mDC.setColor (Color.blue);
        mDC.setFont (fontTimesRoman);
        mDC.drawString (mText, mX, y);
        gDC.drawImage (img, 0, 0, this);
    }
}
}
```

Ponieważ każdy tekst może mieć inną długość, przy każdej jego zmianie będą musiały być modyfikowane również wartości zmiennych określających parametry animacji. Dlatego też obliczenia te zostały przeniesione do nowej metody o nazwie `resetText`. Aktualny kierunek ruchu jest kontrolowany przez zmienną `direction`, wartość 0 oznacza

ruch w lewą stronę, natomiast wartość 1 — ruch w prawą stronę. Aktualnie poruszający się po ekranie tekst jest zawarty w zmiennej `mText`, natomiast współrzędna `x` tego tekstu w zmiennej o nazwie `mX`.

Kod wątku wykonującego animację jest zawarty, podobnie jak we wcześniejszych przykładach, w metodzie `run`. Część rysująca tekst na ekranie ma budowę bardzo podobną do wykorzystywanej w poprzednich przykładach, choć są wykorzystywane inne zmienne, a pozostała część kodu jest również zbudowana inaczej. Przede wszystkim są wykonywane różne fragmenty w zależności od kierunku ruchu, czyli od stanu zmiennej `direction`. Jeżeli wartość tej zmiennej jest równa 0, a zatem ruch odbywa się w lewo, w każdym przebiegu pętli `while` wartość zmiennej `mX` jest zmniejszana o wartość zmiennej `speedStep`. Kiedy natomiast wartość zmiennej `direction` jest równa 1, czyli ruch odbywa się w prawo, w każdym przebiegu pętli do zmiennej `mX` jest dodawana wartość zmiennej `speedStep`.

Jeżeli ruch odbywa się w lewo i zostanie wykryte, że tekst znalazł się poza lewą krawędzią ekranu (czyli prawdziwy jest warunek `mX < -strWidth`), następuje wymiana tekstu. Zmiennej `mX` jest przypisywany kolejny tekst z tablicy `textsArr`. Numer aktualnego tekstu jest kontrolowany przez zmienną `currText`. Jeżeli ruch odbywa się w prawo, wymiana tekstu nastąpi dopiero wtedy, kiedy całkowicie zniknie on za prawą krawędzią apletu, czyli kiedy warunek `mX > appWidth + strWidth` będzie prawdziwy.

Skrypt 94. Teksty najeżdżające od prawej strony

Skrypt przedstawiony na listingu 94. prezentuje jeszcze jeden efekt pływających napisów, tym razem teksty najeżdżają od prawej strony ekranu, poruszając się w lewo. Kiedy tekst osiągnie lewy brzeg apletu, zatrzymuje się, a wędrówkę rozpoczyna kolejny napis. Nowy napis najeżdża na stary, przesłaniając coraz większą jego część (rysunek 2.1), aż w końcu sam dotrze do lewego brzegu apletu, wtedy proces powtarza się z kolejnym tekstem. Znacznik `<APPLET>` musi oczywiście posiadać wszystkie parametry określające teksty i ich liczbę, tak jak zostało to pokazane w przykładzie 93. (listing 93A).

Rysunek 2.1.

Przykładowy efekt działania apletu 94



Listing 94. Teksty najeżdżające z prawej strony

```

/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis                               http://marcinlis.com */

import java.awt.*;
import java.applet.*;

public
class FlowString extends Applet implements Runnable

```



```
{
    int sX, mX, y, speedStep, currText = 0;
    int appWidth, appHeight;
    int strWidth, strHeight, strMaxDescent;
    String[] textsArr;
    String sText, mText;
    Font fontTimesRoman;
    Image img;

    Graphics gDC, mDC;
    boolean stopped = false;
    public void init()
    {
        setBackground(Color.white);
        String strTextsNo;
        int textsNo;

        if((strTextsNo = getParameter("textsNo")) == null)
            strTextsNo = "2";

        try{
            textsNo = Integer.parseInt(strTextsNo);
        }
        catch(Exception e){
            textsNo = 2;
        }

        textsArr = new String[textsNo];

        String str;

        for(int i = 1; i <= textsNo; i++){
            if((str = getParameter("text" + i)) == null)
                str = "Brak parametru text" + i + "!";
            textsArr[i - 1] = str;
        }

        String strFontSize;
        int fontSize;
        if((strFontSize = getParameter("fontsize")) == null)
            strFontSize = "36";

        try{
            fontSize = Integer.parseInt(strFontSize);
        }
        catch(Exception e){
            fontSize = 36;
        }

        String strSpeedStep;
        if((strSpeedStep = getParameter("speedstep")) == null)
            strSpeedStep = "1";

        try{
            speedStep = Math.abs(Integer.parseInt(strSpeedStep));
        }
        catch(Exception e){
```

```
        speedStep = 1;
    }

    sText = "";
    mText = textsArr[0];

    appWidth = getWidth();
    appHeight = getHeight();

    fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
    img = createImage(appWidth, appHeight);
    mDC = img.getGraphics();
    gDC = getGraphics();

    gDC.setFont(fontTimesRoman);
    FontMetrics fm = gDC.getFontMetrics();
    strHeight = fm.getHeight();
    strWidth = fm.stringWidth(mText);
    strMaxDescent = fm.getMaxDescent();

    sX = 0;
    mX = appWidth;
    y = (int)((appHeight + strHeight) / 2);
}
public void start()
{
    Thread thread = new Thread (this);
    thread.start();
}
public void stop()
{
    stopped = true;
}
public void run()
{
    while (!stopped){
        if (mX <= 0){
            sText = mText;
            if(++currText >= textsArr.length)
                currText = 0;
            mText = textsArr[currText];
            FontMetrics fm = gDC.getFontMetrics();
            strHeight = fm.getHeight();
            strWidth = fm.stringWidth(mText);
            strMaxDescent = fm.getMaxDescent();

            sX = 0;
            mX = appWidth;
            y = (int)((appHeight + strHeight) / 2);
        }
        try{
            Thread.sleep(1);
        }
        catch (InterruptedException e){
        }
        mX -= speedStep;
    }
}
```

```
mDC.clearRect(0, 0, appWidth, appHeight);
mDC.setColor (Color.blue);
mDC.setFont (fontTimesRoman);
mDC.drawString (sText, sX, y);
mDC.clearRect(mX, y - strHeight + strMaxDescent, appWidth - mX, strHeight);
mDC.drawString (mText, mX, y);
gDC.drawImage (img, 0, 0, this);
    }
}
}
```

Technika animacji jest podobna, jak w poprzednio wykorzystywanych przykładach. Aktualna pozycja poruszającego się tekstu jest kontrolowana przez zmienną `mX`, która w każdym przebiegu pętli `while` jest zmniejszana o wartość zapisaną w zmiennej `speedStep`. Należy jednak pamiętać, że na ekranie znajdują się jednocześnie dwa napisy: jeden statyczny umieszczony u lewego brzegu apletu i drugi poruszający się w lewo. Oba muszą być rysowane w każdym przebiegu pętli. Tekst pierwszy, statyczny, jest zapisany w zmiennej `sText`, natomiast tekst drugi, poruszający się, w zmiennej `mText`. Kiedy tekst poruszający się osiągnie lewy brzeg ekranu, a tym samym całkowicie zasłoni tekst statyczny, następuje zamiana, tzn. tekst, który się poruszał (zapisany w zmiennej `mText`), staje się tekstem statycznym (zapisanym w zmiennej `sText`), a ruch zaczyna kolejny napis z tablicy `textsArr`.

Podczas rysowania napisów należy dodatkowo zwrócić uwagę na dwie rzeczy. Pierwsza to kolejność wyprowadzania napisów, oczywiście najpierw trzeba narysować napis statyczny, a dopiero potem napis poruszający się. Druga ważna sprawa to konieczność wyczyszczenia obszaru (prostokąta), w którym będzie narysowany napis poruszający się. Jeśli tego nie zrobimy, oba teksty po prostu nałożą się na siebie, tworząc nieczytelny wzór. Oczywiście trzeba odpowiednio obliczyć współrzędne tego prostokąta. Lewy górny róg będzie znajdował się we współrzędnych: $x = mX$, $y = y - strHeight + strMaxDescent$, natomiast prawy dolny róg — we współrzędnych: $x = appWidth - mX$, $y = strHeight$.

Skrypt 95. Różnokolorowe teksty nadjeżdżające od prawej strony

Aplet 94. pokazywał sposób realizacji efektu napisów nadjeżdżających od prawej strony i zatrzymujących się przy lewej krawędzi obszaru apletu. Efekt ten można urozmaicić, zmieniając kolor każdego napisu. Zmiany takie mogą być losowe bądź też kolory mogą być z góry ustalone. Kod zaprezentowany na listingu 95. zakłada losowy dobór kolorów, oddzielny dla każdej animacji, tzn. każdy napis będzie zachowywał swój kolor tylko tak długo, jak długo będzie widoczny na ekranie.

Listing 95. Różnokolorowe teksty płynące w lewo

```
/* Aplet pochodzi z książki "505 skryptów dla webmastera" */
/* autor: Marcin Lis http://marcinlis.com */

import java.awt.*;
import java.applet.*;
```

```
import java.util.*;

public
class FlowString extends Applet implements Runnable
{
    int sX, mX, y, speedStep, currText = 0;
    int appwidth, appHeight;
    int strWidth, strHeight, strMaxDescent;
    String[] textsArr;
    String sText, mText;
    Font fontTimesRoman;
    Color sColor, mColor;
    Random rnd;
    Image img;
    Graphics gDC, mDC;
    boolean stopped = false;
    public void init()
    {
        String strTextsNo;
        int textsNo;

        if((strTextsNo = getParameter("textsNo")) == null)
            strTextsNo = "2";

        try{
            textsNo = Integer.parseInt(strTextsNo);
        }
        catch(Exception e){
            textsNo = 2;
        }

        textsArr = new String[textsNo];

        String str;

        for(int i = 1; i <= textsNo; i++){
            if((str = getParameter("text" + i)) == null)
                str = "Brak parametru text" + i + "!";
            textsArr[i - 1] = str;
        }

        String strFontSize;
        int fontSize;
        if((strFontSize = getParameter("fontsize")) == null)
            strFontSize = "36";

        try{
            fontSize = Integer.parseInt(strFontSize);
        }
        catch(Exception e){
            fontSize = 36;
        }

        String strSpeedStep;
        if((strSpeedStep = getParameter("speedstep")) == null)
            strSpeedStep = "1";
    }
}
```

```
try{
    speedStep = Math.abs(Integer.parseInt(strSpeedStep));
}
catch(Exception e){
    speedStep = 1;
}

sText = "";
mText = textsArr[0];

appWidth = getWidth();
appHeight = getHeight();

setBackground(Color.white);

fontTimesRoman = new Font ("TimesRoman", Font.BOLD, fontSize);
img = createImage(appWidth, appHeight);
mDC = img.getGraphics();
gDC = getGraphics();

gDC.setFont(fontTimesRoman);
FontMetrics fm = gDC.getFontMetrics();
strHeight = fm.getHeight();
strWidth = fm.stringWidth(mText);
strMaxDescent = fm.getMaxDescent();

sX = 0;
mX = appWidth;
y = (int)((appHeight + strHeight) / 2);

rnd = new Random();
mColor = new Color(rnd.nextInt(16777215));
sColor = new Color(rnd.nextInt(16777215));
}
public void start()
{
    Thread thread = new Thread (this);
    thread.start();
}
public void stop()
{
    stopped = true;
}
public void run()
{
    while (!stopped){
        if (mX <= 0){
            sText = mText;
            sColor = mColor;
            mColor = new Color(rnd.nextInt(16777215));
            if(++currText >= textsArr.length)
                currText = 0;
            mText = textsArr[currText];
            FontMetrics fm = gDC.getFontMetrics();
            strHeight = fm.getHeight();
            strWidth = fm.stringWidth(mText);
            strMaxDescent = fm.getMaxDescent();
        }
    }
}
```

```
        sX = 0;
        mX = appWidth;
        y = (int)((appHeight + strHeight) / 2);
    }
    try{
        Thread.sleep(1);
    }
    catch (InterruptedException e){
    }
    mX -= speedStep;

    mDC.clearRect(0, 0, appWidth, appHeight);
    mDC.setColor (sColor);
    mDC.setFont (fontTimesRoman);
    mDC.drawString (sText, sX, y);
    mDC.setColor (mColor);
    mDC.clearRect(mX, y - strHeight + strMaxDescent, appWidth - mX, strHeight);
    mDC.drawString (mText, mX, y);
    gDC.drawImage (img, 0, 0, this);
    }
}
}
```

Kolor tekstu statycznego jest przechowywany w zmiennej `sColor`, natomiast kolor tekstu poruszającego się — w zmiennej `mColor`. Losowanie koloru odbywa się za pomocą obiektu klasy `Random` i jego metody `nextInt`, zwracającej losową liczbę typu `int` z zadanego przedziału. Pierwsze losowanie odbywa się w metodzie `init`, natomiast kolejne losowania odbywają się w pętli `while` realizującej animację za każdym razem, kiedy poruszający się tekst dotrze do lewego brzegu ekranu.