

Przedmowa (11)

Wprowadzenie (13)

Podziękowania (15)

O autorze (17)

Rozdział 1. Przyzwyczajanie się do JavaScriptu (19)

- Sposób 1. Ustal, której wersji JavaScriptu używasz (19)
- Sposób 2. Liczby zmiennoprzecinkowe w JavaScriptcie (24)
- Sposób 3. Uważaj na niejawną konwersję typu (27)
- Sposób 4. Stosuj typy proste zamiast nakładek obiektowych (32)
- Sposób 5. Unikaj stosowania operatora == dla wartości o różnych typach (34)
- Sposób 6. Ograniczenia mechanizmu automatycznego dodawania średników (37)
- Sposób 7. Traktuj łańcuchy znaków jak sekwencje 16-bitowych jednostek kodowych (43)

Rozdział 2. Zasięg zmiennych (47)

- Sposób 8. Minimalizuj liczbę obiektów globalnych (47)
- Sposób 9. Zawsze deklaruuj zmienne lokalne (50)
- Sposób 10. Unikaj słowa kluczowego with (51)
- Sposób 11. Poznaj domknięcia (54)
- Sposób 12. Niejawne przenoszenie deklaracji zmiennych na początek bloku (czyli hoisting) (57)
- Sposób 13. Stosuj wyrażenia IIFE do tworzenia zasięgu lokalnego (59)
- Sposób 14. Uważaj na nieprzenośne określanie zasięgu nazwanych wyrażeń funkcyjnych (62)
- Sposób 15. Uważaj na nieprzenośne określanie zasięgu lokalnych deklaracji funkcji w bloku (65)
- Sposób 16. Unikaj tworzenia zmiennych lokalnych za pomocą funkcji eval (67)
- Sposób 17. Przedkładaj pośrednie wywołania eval nad bezpośrednie wywołania tej funkcji (68)

Rozdział 3. Korzystanie z funkcji (71)

- Sposób 18. Różnice między wywołaniami funkcji, metod i konstruktorów (71)
- Sposób 19. Funkcje wyższego poziomu (74)
- Sposób 20. Stosuj instrukcję call do wywoływania metod dla niestandardowego odbiorcy (77)
- Sposób 21. Stosuj instrukcję apply do wywoływania funkcji o różnej liczbie argumentów (79)
- Sposób 22. Stosuj słowo kluczowe arguments do tworzenia funkcji wariadycznych (81)
- Sposób 23. Nigdy nie modyfikuj obiektu arguments (82)
- Sposób 24. Używaj zmiennych do zapisywania referencji do obiektu arguments (84)
- Sposób 25. Używaj instrukcji bind do pobierania metod o stałym odbiorcy (85)
- Sposób 26. Używaj metody bind do wiązania funkcji z podzbiorem argumentów (technika currying) (87)
- Sposób 27. Wybieraj domknięcia zamiast łańcuchów znaków do hermetyzowania kodu (88)
- Sposób 28. Unikaj stosowania metody toString funkcji (90)

- Sposób 29. Unikaj niestandardowych właściwości przeznaczonych do inspekcji stosu (92)

Rozdział 4. Obiekty i prototypy (95)

- Sposób 30. Różnice między instrukcjami prototype, getPrototypeOf i __proto__ (95)
- Sposób 31. Stosuj instrukcję Object.getPrototypeOf zamiast __proto__ (99)
- Sposób 32. Nigdy nie modyfikuj właściwości __proto__ (100)
- Sposób 33. Uniezależnianie konstruktorów od instrukcji new (101)
- Sposób 34. Umieszczaj metody w prototypach (103)
- Sposób 35. Stosuj domknięcia do przechowywania prywatnych danych (105)
- Sposób 36. Stan egzemplarzy przechowuj tylko w nich samych (107)
- Sposób 37. Zwracaj uwagę na niejawne wiązanie obiektu this (109)
- Sposób 38. Wywoływanie konstruktorów klasy bazowej w konstruktorach klas pochodnych (111)
- Sposób 39. Nigdy nie wykorzystuj ponownie nazw właściwości z klasy bazowej (115)
- Sposób 40. Unikaj dziedziczenia po klasach standardowych (117)
- Sposób 41. Traktuj prototypy jak szczegół implementacji (119)
- Sposób 42. Unikaj nieprzemyślanego stosowania techniki monkey patching (120)

Rozdział 5. Tablice i słowniki (123)

- Sposób 43. Budowanie prostych słowników na podstawie egzemplarzy typu Object (123)
- Sposób 44. Stosuj prototypy null, aby uniknąć zaśmiecania przez prototypy (126)
- Sposób 45. Używaj metody hasOwnProperty do zabezpieczania się przed zaśmiecaniem przez prototypy (128)
- Sposób 46. Stosuj tablice zamiast słowników przy tworzeniu kolekcji uporządkowanych (132)
- Sposób 47. Nigdy nie dodawaj enumerowanych właściwości do prototypu Object.prototype (134)
- Sposób 48. Unikaj modyfikowania obiektu w trakcie enumeracji (136)
- Sposób 49. Stosuj pętlę for zamiast pętli for...in przy przechodzeniu po tablicy (140)
- Sposób 50. Zamiast pętli stosuj metody do obsługi iteracji (142)
- Sposób 51. Wykorzystaj uniwersalne metody klasy Array w obiektach podobnych do tablic (146)
- Sposób 52. Przedkładaj literały tablicowe nad konstruktor klasy Array (148)

Rozdział 6. Projekty bibliotek i interfejsów API (151)

- Sposób 53. Przestrzegaj spójnych konwencji (151)
- Sposób 54. Traktuj wartość undefined jak brak wartości (153)
- Sposób 55. Stosuj obiekty z opcjami do przekazywania argumentów za pomocą słów kluczowych (157)
- Sposób 56. Unikaj niepotrzebnego przechowywania stanu (161)
- Sposób 57. Określaj typy na podstawie struktury, aby tworzyć elastyczne interfejsy (164)
- Sposób 58. Różnice między tablicami a obiektami podobnymi do tablic (167)
- Sposób 59. Unikaj nadmiernej koercji (171)
- Sposób 60. Obsługa łańcuchów metod (174)

Rozdział 7. Współbieżność (179)

- Sposób 61. Nie blokuj kolejki zdarzeń operacjami wejścia-wyjścia (180)
- Sposób 62. Stosuj zagnieżdżone lub nazwane wywołania zwrotne do tworzenia sekwencji asynchronicznych wywołań (183)
- Sposób 63. Pamiętaj o ignorowanych błędach (187)
- Sposób 64. Stosuj rekurencję do tworzenia asynchronicznych pętli (190)
- Sposób 65. Nie blokuj kolejki zdarzeń obliczeniami (193)
- Sposób 66. Wykorzystaj licznik do wykonywania współbieżnych operacji (197)
- Sposób 67. Nigdy nie uruchamiaj synchronicznie asynchronicznych wywołań zwrotnych (201)
- Sposób 68. Stosuj obietnice, aby zwiększyć przejrzystość asynchronicznego kodu (203)

Skorowidz (207)