

Wstęp (11)

1. Wprowadzenie (15)

- Wzorce (15)
- JavaScript - podstawowe cechy (16)
 - o Zorientowany obiektowo (16)
 - o Brak klas (17)
 - o Prototypy (18)
 - o Środowisko (18)
- ECMAScript 5 (18)
 - o Narzędzie JSLint (19)
 - o Konsola (20)

2. Podstawy (21)

- Tworzenie kodu łatwego w konserwacji (21)
- Minimalizacja liczby zmiennych globalnych (22)
 - o Problem ze zmiennymi globalnymi (22)
 - o Efekty uboczne pominięcia var (24)
 - o Dostęp do obiektu globalnego (25)
 - o Wzorzec pojedynczego var (25)
 - o Przenoszenie deklaracji - problem rozrzuconych deklaracji var (26)
- Pętle for (27)
- Pętle for-in (29)
- Modyfikacja wbudowanych prototypów (31)
- Wzorzec konstrukcji switch (31)
- Unikanie niejawnego rzutowania (32)
- Unikanie eval() (32)
- Konwertowanie liczb funkcją parseInt() (34)
- Konwencje dotyczące kodu (34)
 - o Wcięcia (35)
 - o Nawiasy klamrowe (35)
 - o Położenie nawiasu otwierającego (36)
 - o Białe spacje (37)
- Konwencje nazewnictwa (38)
 - o Konstruktory pisane od wielkiej litery (38)
 - o Oddzielanie wyrazów (39)
 - o Inne wzorce nazewnictwa (39)
- Pisanie komentarzy (40)
- Pisanie dokumentacji interfejsów programistycznych (41)
 - o Przykład dokumentacji YUIDoc (42)
- Pisanie w sposób ułatwiający czytanie (44)
- Ocenianie kodu przez innych członków zespołu (45)
- Minifikowanie kodu tylko w systemie produkcyjnym (46)
- Uruchamianie narzędzia JSLint (47)
- Podsumowanie (47)

3. Literały i konstruktory (49)

- Literał obiektu (49)

- o Składnia literału obiektowego (50)
 - o Obiekty z konstruktora (51)
 - o Pułapka konstruktora Object (51)
- Własne funkcje konstruuujące (52)
 - o Wartość zwracana przez konstruktor (53)
- Wzorce wymuszania użycia new (54)
 - o Konwencja nazewnictwa (54)
 - o Użycie that (54)
 - o Samowywołujący się konstruktor (55)
- Literał tablicy (56)
 - o Składnia literału tablicy (56)
 - o Pułapka konstruktora Array (56)
 - o Sprawdzanie, czy obiekt jest tablicą (57)
- JSON (58)
 - o Korzystanie z formatu JSON (58)
- Literał wyrażenia regularnego (59)
 - o Składnia literałowego wyrażenia regularnego (60)
- Otoczki typów prostych (61)
- Obiekty błędów (62)
- Podsumowanie (63)

4. Funkcje (65)

- Informacje ogólne (65)
 - o Stosowana terminologia (66)
 - o Deklaracje kontra wyrażenia - nazwy i przenoszenie na początek (67)
 - o Właściwość name funkcji (68)
 - o Przenoszenie deklaracji funkcji (68)
- Wzorzec wywołania zwrotnego (70)
 - o Przykład wywołania zwrotnego (70)
 - o Wywołania zwrotne a zakres zmiennych (72)
 - o Funkcje obsługi zdarzeń asynchronicznych (73)
 - o Funkcje czasowe (73)
 - o Wywołania zwrotne w bibliotekach (74)
- Zwracanie funkcji (74)
- Samodefiniujące się funkcje (75)
- Funkcje natychmiastowe (76)
 - o Parametry funkcji natychmiastowych (77)
 - o Wartości zwracane przez funkcje natychmiastowe (77)
 - o Zalety i zastosowanie (79)
- Natychmiastowa inicjalizacja obiektu (79)
- Usuwanie warunkowych wersji kodu (80)
- Właściwości funkcji - wzorzec zapamiętywania (82)
- Obiekty konfiguracyjne (83)
- Rozwijanie funkcji (84)
 - o Aplikacja funkcji (84)
 - o Aplikacja częściowa (85)
 - o Rozwijanie funkcji (87)
 - o Kiedy używać aplikacji częściowej (89)
- Podsumowanie (89)

5. Wzorce tworzenia obiektów (91)

- Wzorec przestrzeni nazw (91)
 - Funkcja przestrzeni nazw ogólnego stosowania (92)
- Deklarowanie zależności (94)
- Metody i właściwości prywatne (95)
 - Składowe prywatne (96)
 - Metody uprzywilejowane (96)
 - Problemy z prywatnością (96)
 - Literały obiektów a prywatność (98)
 - Prototypy a prywatność (98)
 - Udostępnianie funkcji prywatnych jako metod publicznych (99)
- Wzorec modułu (100)
 - Odkrywczy wzorec modułu (102)
 - Moduły, które tworzą konstruktory (102)
 - Import zmiennych globalnych do modułu (103)
- Wzorec piaskownicy (103)
 - Globalny konstruktor (104)
 - Dodawanie modułów (105)
 - Implementacja konstruktora (106)
- Składowe statyczne (107)
 - Publiczne składowe statyczne (107)
 - Prywatne składowe statyczne (109)
- Stałe obiektów (110)
- Wzorec łańcucha wywołań (112)
 - Wady i zalety wzorca łańcucha wywołań (112)
- Metoda method() (113)
- Podsumowanie (114)

6. Wzorce wielokrotnego użycia kodu (115)

- Klasyczne i nowoczesne wzorce dziedziczenia (115)
- Oczekiwane wyniki w przypadku stosowania wzorca klasycznego (116)
- Pierwszy wzorec klasyczny - wzorec domyślny (117)
 - Podążanie wzdłuż łańcucha prototypów (117)
 - Wady wzorca numer jeden (119)
- Drugi wzorec klasyczny - pożyczanie konstruktora (119)
 - Łańcuch prototypów (120)
 - Dziedziczenie wielobazowe przy użyciu pożyczania konstruktorów (121)
 - Zalety i wady wzorca pożyczania konstruktora (122)
- Trzeci wzorec klasyczny - pożyczanie i ustawianie prototypu (122)
- Czwarty wzorec klasyczny - współdzielenie prototypu (123)
- Piąty wzorec klasyczny - konstruktor tymczasowy (124)
 - Zapamiętywanie klasy nadrzędnej (125)
 - Czyszczenie referencji na konstruktor (125)
- Podejście klasowe (126)
- Dziedziczenie prototypowe (129)
 - Dyskusja (129)
 - Dodatki do standardu ECMAScript 5 (130)
- Dziedziczenie przez kopiowanie właściwości (131)

- Wzorzec wmieszania (132)
- Pożyczanie metod (133)
 - o Przykład - pożyczanie metody od obiektu Array (134)
 - o Pożyczenie i przypisanie (134)
 - o Metoda `Function.prototype.bind()` (135)
- Podsumowanie (136)

7. Wzorce projektowe (137)

- Singleton (137)
 - o Użycie słowa kluczowego `new` (138)
 - o Instancja we właściwości statycznej (139)
 - o Instancja w domknięciu (139)
- Fabryka (141)
 - o Wbudowane fabryki obiektów (143)
- Iterator (143)
- Dekorator (145)
 - o Sposób użycia (145)
 - o Implementacja (146)
 - o Implementacja wykorzystująca listę (148)
- Strategia (149)
 - o Przykład walidacji danych (150)
- Fasada (152)
- Pośrednik (153)
 - o Przykład (153)
 - o Pośrednik jako pamięć podręczna (159)
- Mediator (160)
 - o Przykład mediatora (160)
- Obserwator (163)
 - o Pierwszy przykład - subskrypcja magazynu (163)
 - o Drugi przykład - gra w naciskanie klawiszy (166)
- Podsumowanie (169)

8. DOM i wzorce dotyczące przeglądarek (171)

- Podział zadań (171)
- Skrypty wykorzystujące DOM (172)
 - o Dostęp do DOM (173)
 - o Modyfikacja DOM (174)
- Zdarzenia (175)
 - o Obsługa zdarzeń (175)
 - o Delegacja zdarzeń (177)
- Długo działające skrypty (178)
 - o Funkcja `setTimeout()` (178)
 - o Skrypty obliczeniowe (179)
- Komunikacja z serwerem (179)
 - o Obiekt `XMLHttpRequest` (180)
 - o JSONP (181)
 - o Ramki i wywołania jako obrazy (184)
- Serwowanie kodu JavaScript klientom (184)

- o Łączenie skryptów (184)
 - o Minifikacja i kompresja (185)
 - o Nagłówek Expires (185)
 - o Wykorzystanie CDN (186)
- Strategie wczytywania skryptów (186)
 - o Lokalizacja elementu <script> (187)
 - o Wysyłanie pliku HTML fragmentami (188)
 - o Dynamiczne elementy <script> zapewniające nieblokujące pobieranie (189)
 - o Wczytywanie leniwe (190)
 - o Wczytywanie na żądanie (191)
 - o Wstępne wczytywanie kodu JavaScript (192)
- Podsumowanie (194)

Skorowidz (195)