

Spis treści (podsumowanie)

Wprowadzenie	xxiii
1 Szybki skok na głębokie wody JavaScriptu. Czas się zamoczyć	1
2 Pisanie prawdziwego kodu. Idziemy dalej	45
3 Przedstawienie funkcji. Stawiamy na funkcjonalność	79
4 Porządkowanie naszych danych. Tablice	125
5 Zrozumieć obiekty. Wycieczka do Obiektowa	173
6 Interakcja ze stronami WWW. Poznajemy DOM	229
7 Typy, równość, konwersje i cały ten jazz. Poważne typy	263
8 Łączenie wszystkiego w całość. Tworzenie aplikacji	317
9 Obsługa zdarzeń. Programowanie asynchroniczne	383
10 Funkcje anonimowe i funkcje wyższego rzędu. Wyzwolone funkcje	429
11 Nowoczesna składnia, zasięg leksykalny i domknięcia. Poważne funkcje	475
12 Zaawansowane sposoby konstruowania obiektów. Tworzenie obiektów	525
A Pozostałości. Dziesięć najważniejszych zagadnień (których nie opisaliśmy)	573
Skorowidz	589

Spis treści (ten właściwy)

Wprowadzenie

Twój mózg myśli o JavaScriptcie. Czytając tę książkę, Ty starasz się czegoś nauczyć, natomiast Twój mózg wyświadcza Ci przysługę, starając się, by cała ta wiedza *nie została zapamiętana*. Twój mózg myśli sobie: „Lepiej zostawić miejsce na istotne rzeczy, takie jak to, których dzikich zwierząt należy unikać oraz czy jeżdżenie na snowboardzie nago jest kiepskim pomysłem, czy nie”. Jak więc możesz *przekonać* swój mózg, że Twoje życie zależy od poznania JavaScriptu?

Dla kogo jest ta książka?	xxiv
Wiemy, co myślisz	xxv
Metapoznanie	xxvii
Zmusz swój mózg do posłuszeństwa	xxix
Przeczytaj to	xxx
Zespół recenzentów technicznych	xxxiii
Podziękowania	xxxiv

szybki skok na głębokie wody javascriptu

Czas się zamoczyć

1

JavaScript to dane Ci supermoce. To prawdziwy język programowania internetu, który pozwala **dodawać** do stron WWW **zachowania**. Zapomnij o suchych, nudnych i statycznych stronach — korzystając z języka JavaScript, będziesz w stanie porozumiewać się ze swoimi użytkownikami, pobierać z internetu dane do wyświetlania na swoich stronach, rysować grafikę bezpośrednio na nich i robić wiele innych świetnych rzeczy. Co więcej, znając JavaScript, będziesz także mógł zapewniać swoim użytkownikom **całkowicie nowe** możliwości.

Jako programista posługujący się językiem JavaScript znajdziesz się w dobrym towarzystwie — jest on nie tylko jednym z **najbardziej popularnych** języków programowania, lecz także językiem **obsługiwanym** przez wszystkie nowoczesne przeglądarki i stosowanym w wielu środowiskach, które nie są powiązane z przeglądarkami. Więcej o tym napiszemy dalej w tej książce, a na razie bierzmy się do roboty!



Sposób działania języka JavaScript	2
Jak należy pisać kod JavaScript?	3
Jak umieszczać kod JavaScript na stronie?	4
Dzieńko, JavaScript przebył długą drogę...	6
Jak tworzyć instrukcje?	10
Zmienne i wartości	11
Stałe, inny rodzaj zmiennych	12
Odsuń się od tej klawiatury!	14
Wyraż się	17
Wykonywanie operacji więcej niż jeden raz	19
Jak działa pętla while?	20
Podejmowanie decyzji w języku JavaScript	24
A kiedy trzeba podejmować WIELE decyzji...	25
Wyciągnij rękę i nawiąż kontakt z użytkownikami	27
Poznajemy bliżej funkcję console.log	29
Otwieranie konsoli	30
Piszemy poważną aplikację JavaScript	31
Jak mogę dodać kod do strony? (Niech policzę wszystkie sposoby)	34
Będziemy musieli was rozdzielić	35

pisanie prawdziwego kodu

Idziemy dalej

2

Znasz już zmienne, typy, wyrażenia..., możesz więc pójść o krok dalej.

Chodzi o to, że już trochę poznałeś język JavaScript. Wiesz na tyle dużo, by móc napisać jakiś **prawdziwy kod**. Kod, który robi coś interesującego, którego ktoś chciałby używać. Wciąż brakuje Ci jednak **praktycznego doświadczenia** w pisaniu kodu. Mamy zamiar temu zaradzić właśnie tu i teraz! A w jaki sposób? Skacząc na główkę na głęboką wodę i pisząc prostą grę — w całości w języku JavaScript. Cel jest bardzo ambitny, jednak będziemy go realizować krok po kroku. Chodź, zaczynamy! Jeśli zechcesz przy okazji uruchomić kolejny start-up, nie będziemy Ci przeszkadzać — kod jest Twój.

Napiszmy grę w okręty	46
Zacznijmy od projektu wysokiego poziomu	47
Analiza pseudokodu	49
Zanim przejdziemy dalej, nie zapomnij o kodzie HTML!	51
Pisanie kodu prostej wersji gry w okręty	52
A teraz zajmijmy się logiką gry	53
Krok 1. Przygotowanie pętli i pobranie danych	54
Jak działa funkcja prompt?	55
Krok 2. Sprawdzanie komórki wskazanej przez użytkownika	56
Dodanie kodu wykrywającego trafienia	59
Krok 3. Hej, zatopiłeś mój okręt!	59
Krok 4. Prezentacja informacji o zakończonej grze	60
Chwilka na zapewnienie jakości	63
Czy możemy pogadać o rozwlekłości Twojego kodu?	67
Kończymy prostą wersję gry w okręty	68
Przepis na generowanie liczb losowych	69
Gratulujemy pierwszego prawdziwego programu w języku JavaScript... i kilka słów o wielokrotnym używaniu kodu	73



przedstawienie funkcji

3

Stawiamy na funkcjonalność

Przygotuj się na użycie pierwszej ze swoich supermocy. Zdobyłeś już nieco umiejętności programistycznych; teraz nadszedł czas, aby rozwinąć je jeszcze bardziej przy użyciu **funkcji**. Funkcje zapewniają możliwość pisania kodu, który można stosować we wszelkich możliwych okolicznościach, kodu **używanego wielokrotnie**, którym można znacznie łatwiej **zarządzać** i, wreszcie, który można **wyodrębnić**, nadać mu łatwą do zapamiętania nazwę, a potem zapomnieć o całej jego złożoności i zająć się innymi ważnymi problemami. Przekonasz się, że funkcje to nie tylko droga, która zmieni Cię z autora skryptów w programistę. Są one kluczowym czynnikiem określającym styl programowania w języku JavaScript. W tym rozdziale zaczniemy od podstaw: poznasz mechanikę funkcji i tajniki ich działania, a dalej w tej książce będziesz stopniowo powiększać swoją wiedzę i umiejętności ich stosowania. Zacznij więc budować solidne podstawy znajomości JavaScriptu i *zrób to już teraz*.

Co z tym kodem było nie tak?	81
Swoją drogą, czy wspominaliśmy już o FUNKCJACH?	83
No dobrze, ale jak to właściwie działa?	84
Co można przekazywać do funkcji?	89
JavaScript przekazuje przez wartość	92
Zakręcone funkcje	94
Funkcje mogą także coś zwracać	95
Śledzenie wykonania funkcji z instrukcją return	96
Zmienne globalne i lokalne	99
Poznanie zasięgu zmiennych lokalnych i globalnych	101
To jeszcze nie koniec historii	102
Nie zapominaj o deklarowaniu zmiennych	104
Krótkie życie zmiennych	105



porządkowanie naszych danych

Tablice

4

JavaScript to nie tylko liczby, łańcuchy znaków i wartości logiczne.

Dotychczas pisałeś jedynie kod JavaScript, w którym były używane wartości **typów prostych** – proste łańcuchy znaków, liczby i wartości logiczne, takie jak "Fido", 23 oraz true. Korzystając z takich wartości, można zrobić naprawdę dużo, jednak w którymś momencie będziesz musiał zacząć posługiwać się znacznie **większą ilością danych**. Przykładowo mogą to być wszystkie produkty umieszczone w koszyku zakupowym albo utwory na liście odtwarzania, albo gwiazdorzbiory i współrzędne poszczególnych gwiazd, albo cały katalog produktów. Do tego potrzebujesz jednak czegoś bardziej... *sexy*. W języku JavaScript preferowanym typem danych dla takich uporządkowanych zbiorów informacji jest **tablica**, a w tym rozdziale dokładnie przeanalizujemy, jak umieszczać dane w tablicach, przekazywać tablice oraz jak na nich operować. Dalej w książce omówimy także kilka innych sposobów **strukturyzacji danych**, jednak zaczniemy od tablic.



Czy możesz pomóc firmie BańkoCorp?	126
Jak reprezentować wiele wartości w JavaScriptcie?	126
Jak działają tablice?	128
A w ogóle jak duża jest tablica?	130
Korpo-zdanie-budowator	132
W międzyczasie w firmie BańkoCorp...	135
Jak pobrać wszystkie elementy tablicy?	138
Chwila, istnieje lepszy sposób iteracji po tablicy	140
Znowu nadszedł czas... Czy możemy porozmawiać o rozwekłości Twojego kodu?	147
Poprawienie pętli for przy użyciu operatora postinkrementacji	148
Tworzenie pustej tablicy (i dodawanie do niej danych)	152
Test ostatecznego raportu	156
Krótką inspekcją kodu...	158
Piszemy funkcję printAndGetHighScore	159
Refaktoryzacja kodu z użyciem funkcji printAndGetHighScore	160
Zastosowanie zmian...	162

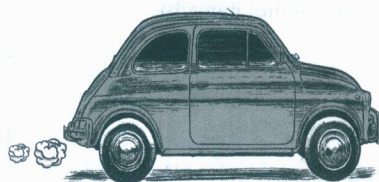
5

zrozumieć obiekty

Wycieczka do Obiektowa

Do tej pory w tworzonym kodzie używałeś jedynie danych typów podstawowych oraz tablic. Dodatkowo podchodziłeś do programowania w sposób proceduralny — korzystałeś z prostych instrukcji, instrukcji warunkowych oraz pętli, ewentualnie umieszczałeś je w funkcjach — to właściwie nie jest **programowanie obiektowe**. Prawdę powiedziawszy, to w ogóle nie jest programowanie obiektowe! Tu i tam, nawet o tym nie wiedząc, użyłeś kilku obiektów, jednak na razie jeszcze nie napisałeś żadnego własnego obiektu. Najwyższy czas, żeby zostawić to stare i nudne proceduralne miasteczko i zacząć tworzyć własne **obiekty**. W tym rozdziale dowiesz się, dlaczego stosowanie obiektów sprawi, że Twoje życie stanie się znacznie lepsze — przynajmniej pod względem **programistycznym** (niestety, w jednej książce nie możemy poprawić Twojej znajomości mody i jednocześnie nauczyć Cię programowania w języku JavaScript). I jeszcze jedno ostrzeżenie: kiedy już poznasz obiekty, nigdy nie będziesz chciał ich porzucić. Wyślij nam pocztówkę, kiedy już dojedziesz do krainy obiektów.

Czy ktoś powiedział „obiekty”?	174
Myśląc o właściwościach...	175
W jaki sposób tworzy się obiekty?	177
Czym w ogóle jest „programowanie obiektowe”?	180
Jak działają właściwości?	181
W jaki sposób zmienna przechowuje obiekt?	
Dociekliwe umysły chciałyby to wiedzieć...	186
Porównanie danych typów podstawowych i obiektów	187
Jeszcze inne operacje z wykorzystaniem obiektów...	188
Wstępna selekcja	189
Porozmawiajmy nieco więcej o przekazywaniu obiektów do funkcji	192
Klasyko-auto-inator	195
Zachowuj się! Jak dodawać zachowania do obiektów?	198
Poprawianie metody drive	199
Bierzemy fiata na jazdę próbną	201
Dlaczego metoda drive nic nie wie o właściwości started?	202
Jak działa this?	204
Skrótowny zapis metod	210
Jak zachowanie wpływa na stan?	211
A teraz niech stan wpływa na zachowanie	212
Gratulujemy utworzenia pierwszych obiektów!	214
Wiesz co? Obiekty są wszędzie dookoła Ciebie!	215



Komunikat ze strony localhost

Brum wrrrr!

OK

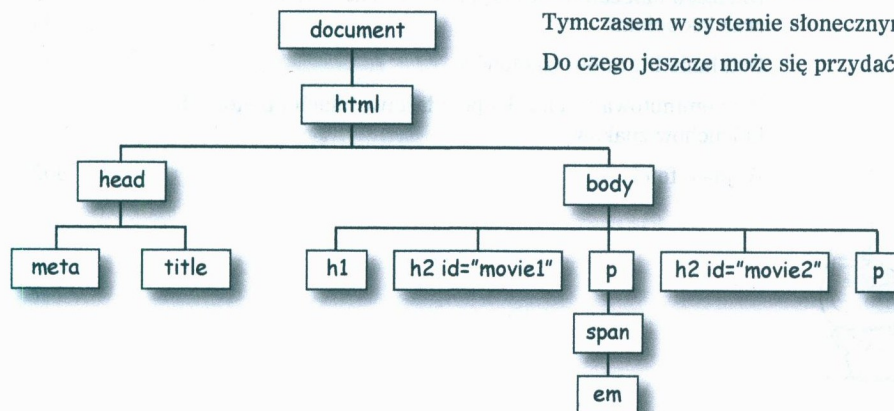
interakcja ze stronami WWW

6

Poznajemy DOM

Przebyłeś już długą drogę, poznając JavaScript. Powoli z żółtodzioba zmieniałeś się w twórcę prostych skryptów, a potem w końcu w **programistę**. Jednak wciąż czegoś Ci brakuje. Abyś mógł naprawdę wykorzystać całą swoją znajomość języka JavaScript, musisz się dowiedzieć, jak prowadzić interakcję ze stronami WWW, w których umieszczasz swoje skrypty. Tylko to pozwoli Ci tworzyć strony, które są **dynamiczne**, które reagują, odpowiadają i aktualizują swoją zawartość już po jej wczytaniu przez przeglądarkę. W jaki sposób można prowadzić interakcję ze stroną WWW? Służy do tego **DOM**, nazywany także **modelem obiektów dokumentu**. W tym rozdziale opiszemy go szczegółowo i wyjaśnimy, jak można z niego korzystać i jak go używać wraz z językiem JavaScript, by nauczyć nasze strony wykonywania wielu nowych sztuczek.

W poprzednim rozdziale miałeś wykonać niewielką misję — misję złamania kodu	230
Co robi ten kod?	231
Jak naprawdę wygląda interakcja JavaScriptu ze stroną WWW?	233
Jak wypiec swój własny DOM?	234
Pierwszy smak modelu DOM	235
Pobieranie elementu przy użyciu metody getElementById	240
Co pobieramy z modelu DOM?	241
Dostęp do kodu HTML w elemencie	242
Co się dzieje, kiedy zmieniamy DOM?	244
Nawet nie myśl o uruchamianiu mojego kodu, zanim strona nie zostanie w całości wczytana	249
Ty mówisz: „procedura obsługi zdarzeń”, ja mówię: „wywołanie zwrotne”	250
Czemu mielibyśmy na tym poprzestać? Pójdźmy jeszcze dalej...	254
Jak ustawiać atrybuty przy użyciu metody setAttribute?	255
Więcej zabawy z atrybutami!	256
Tymczasem w systemie słonecznym...	257
Do czego jeszcze może się przydać DOM?	258



typy, równość, konwersje i cały ten jazz

7

Poważne typy

Nadszedł czas, by poważnie przyrzeć się typom. Jedną ze wspaniałych cech JavaScriptu jest to, że można w nim zrobić całkiem dużo bez jego szczegółowej znajomości. Aby jednak **perfekcyjnie opanować język**, dostać awans i zacząć robić to, co naprawdę chcesz robić w życiu, musisz się zaprzyjaźnić z **typami**. Czy pamiętasz, co już dawno, w pierwszym rozdziale tej książki, powiedzieliśmy na temat JavaScriptu? Że nie może się poszczycić rozpuszczoną, popularną, akademicką definicją? No cóż... To prawda, jednak brak życia studenckiego nie zatrzymał ani Steve'a Jobsa, ani Billa Gatesa, nie zatrzymał także języka JavaScript. Oznacza to jednak, że JavaScript nie ma... hm... doskonale przemyślanego systemu typów i znajdziemy w nim sporo **dziwactw**. Nie obawiaj się jednak, w tym rozdziale dokładnie wszystko wyjaśnimy, dzięki czemu już niebawem nauczysz się unikać tych zawstydzających problemów z typami.

Gdzieś tam jest ukryta prawda...	264
Uważaj, możesz się natknąć na undefined, kiedy będziesz się tego najmniej spodziewać...	266
Jak używać null?	269
Stosowanie wartości NaN	271
Sprawy stają się jeszcze dziwniejsze	271
Musimy coś wyznaczyć	273
Zrozumienie operatora równości (pseudonim: ==)	274
Jak wykonywana jest konwersja operandów operatora równości?	275
Jak ściśle podejść do zagadnienia równości?	278
Jeszcze więcej konwersji typów...	284
Jak określić, czy dwa obiekty są równe?	287
Gdzieś tam leży prawda...	289
JavaScript uwzględnia fałsz	290
Sekretne życie łańcuchów znaków	292
Dlaczego łańcuch może wyglądać jak dana typu prostego oraz jak obiekt?	293
Działanie literałów szablonów	296
Pięciominutowa wycieczka po właściwościach i metodach łańcuchów znaków	297
Wojna o fotel	302



łączenie wszystkiego w całość

Tworzenie aplikacji

8

Włóż to do swojego przybornika z narzędziami, czyli do skrzynki, w której umieszczasz wszystkie swoje nowe umiejętności programistyczne, wiedzę dotyczącą modelu DOM, a nawet HTML-a i CSS-a. W tym rozdziale wykorzystasz wszystkie te umiejętności, by utworzyć pierwszą prawdziwą **aplikację internetową**. Wystarczy już **prymitywnych namiastek gier**, w których na jednowymiarowej planszy pływa jeden okręt. W tym rozdziale wykonasz **pełne doświadczenie**: dużą, atrakcyjną planszę do gry, wiele okrętów oraz obsługę wprowadzania danych przez użytkownika. Struktura strony, na której będzie prowadzona gra, powstanie w języku HTML, jej wygląd określony zostanie przy użyciu stylów CSS, a zachowanie samej gry będzie napisane w języku JavaScript. Przygotuj się: to będzie jazda na całym, rozdział, w którym pełnym gazem będziesz zmierzać w kierunku pisania naprawdę poważnego kodu.



Tym razem napiszemy PRAWDZIwą grę w okręty	318
Krok wstecz... do HTML-a i CSS-a	319
Tworzenie strony HTML: postać ogólna	320
Dodawanie stylów	324
Stosowanie klas hit i miss	327
Jak zaprojektować grę?	329
Implementacja widoku	331
Model	336
Będziesz potrzebować większych okrętów... i większej planszy	337
W jaki sposób będziemy reprezentować okręty?	338
Składamy wszystko w jedną całość...	344
Czy możemy ponownie porozmawiać o Twoim przydługim kodzie?	345
Prezentacja zatopienia...	347
Implementacja kontrolera	349
Przetwarzanie pola wskazanego przez użytkownika	350
Pobieranie współrzędnych	359
Jak rozmieszczać okręty?	365
Unikanie kolizji!	369
Gratulujemy, czas na własny start-up!	372

obsługa zdarzeń

Programowanie asynchroniczne

9

Po przeczytaniu tego rozdziału zdasz sobie sprawę z tego, że to nie są przelewki i nie jesteśmy już w Kansas. Do tej pory pisałeś kod, który zazwyczaj wykonywany był od samego początku do końca — oczywiście Twój kod mógł być nieco bardziej skomplikowany i zawierać kilka funkcji, obiektów i metod, jednak w jakimś momencie ten kod po prostu był wykonany wiersz po wierszu. Cóż, jest nam bardzo przykro, że mówimy Ci to tak późno, jednak **typowy kod JavaScript nie jest pisany w taki sposób**. Kod pisany w tym języku **reaguje na zdarzenia**. Jakiego rodzaju zdarzenia? Może to być kliknięcie strony przez użytkownika, przesłanie danych z serwera, upływ czasu w przeglądarce, jakaś zmiana wprowadzona w modelu DOM oraz wiele innych. W rzeczywistości w niewidoczny dla nas sposób w przeglądarce **cały czas** zachodzą jakieś zdarzenia. W tym rozdziale jeszcze raz przemyślisz swoje podejście do sposobu pisania kodu JavaScript i dowiesz się, dlaczego trzeba pisać kod reagujący na zdarzenia oraz jak należy to robić.

Czym są zdarzenia?	385
Czym jest procedura obsługi zdarzeń?	386
Jak napisać pierwszą procedurę obsługi zdarzeń?	387
Poznajemy zdarzenia, pisząc grę	390
Implementacja gry	391
Dodajmy więcej obrazków	396
Jak użyć tej samej funkcji do obsługi wszystkich obrazków?	398
Jak działa obiekt zdarzenia?	401
Zaprzęgamy obiekt zdarzenia do pracy	403
Zdarzenia i kolejki	406
Jak działa funkcja setTimeout?	410
Kończenie gry	414



10

funkcje anonimowe i funkcje wyższego rzędu

Wyzwolone funkcje

Poznaj funkcje, a potem baw się na całego. Każda sztuka, rzemiosło czy też dyscyplina sportowa ma swoją kluczową cechę, która odróżnia graczy przeciętnych od wirtuozów. W przypadku języka JavaScript jest to prawdziwe i dokładne zrozumienie funkcji. W języku JavaScript funkcje mają kluczowe znaczenie, a wiele technik służących do **projektowania i organizacji** kodu bazuje na zaawansowanej znajomości funkcji i umiejętności korzystania z nich. Droga prowadząca do poznania funkcji na takim poziomie jest interesująca i niejednokrotnie wymagająca dla mózgu, zatem dobrze się do niej przygotuj. W tym rozdziale będziesz się czuł jak dziecko oprowadzane przez Willy'ego Wonkę po fabryce czekolady — będziesz w nim kontynuował poznawanie funkcji w języku JavaScript, a przy okazji zobaczysz dziwaczne, wariackie i cudowne rzeczy.

Tajemnicze dwa oblicza słowa kluczowego function	430
W jaki sposób funkcje są jednocześnie wartościami?	431
Skoro funkcje są wartościami, możemy je zapisywać w zmiennych	434
Czy wspominaliśmy już, że w JavaScriptcie funkcje mają status pierwszej klasy?	437
Rzut oka na inną stronę funkcji...	438
Jak używać funkcji anonimowych?	439
Musimy ponownie pomówić o rozwlekłości Twojego kodu	441
Używając funkcji strzałkowych, możemy skracać kod	443
Tworzenie funkcji strzałkowych	445
Cola sieciowicka	448
Wyjaśnienie działania metody sort tablic	450
Łączymy wszystko w całość	451
W międzyczasie w firmie Cola sieciowicka	452
Prezentacja funkcji wyższego rzędu	455
Filtrowanie z użyciem funkcji wyższego rzędu	456
Nie zapominaj o funkcjach anonimowych i strzałkowych	457
Użycie reduce do obliczenia całkowitej liczby sprzedanych skrzynek	460
Tworzenie łańcuchów wywołań metod map, filter i reduce	462
Iteracje z użyciem forEach	465



11

nowoczesna składnia, zasięg leksykalny i domknięcia

Poważne funkcje

W poprzednim rozdziale rozłożyłeś funkcje na czynniki pierwsze, ale wciąż musisz się o nich jeszcze sporo dowiedzieć. W tym rozdziale pójdziemy jeszcze dalej — to będzie prawdziwa jazda na całego. Pokażemy Ci, jak bawić się ze składnią, dzięki zastosowaniu zaawansowanych technik obsługi argumentów, parametrów i przypisań. Następnie przyjrzymy się zasięgom i niektórym z najdrobniejszych szczegółów związanych z zarządzaniem nimi w języku JavaScript. Ta podróż przez niuanse zasięgów doprowadzi nas do serca domknięć (ang. *closures*) — koncepcji często owianej tajemnicą, lecz jednocześnie kluczowej dla opanowania języka JavaScript. Po przeczytaniu tego rozdziału będziesz potrafił pisać bardziej ekspresyjny kod JavaScript, niż myślałeś, że to możliwe.

Składnia funkcji na poważnie	476
Rozpraszanie argumentów	479
Jest coś, czego jeszcze Ci nie powiedzieliśmy o funkcjach...	481
Deklaracje funkcji są „wynoszone”	482
Zajęliśmy się deklaracjami funkcji, teraz możemy zająć się resztą	483
Musimy porozmawiać o zasięgu	487
Przenoszenie funkcji poza zasięg globalny	488
Przypomnienie o zasięgu leksykalnym	491
Inne spojrzenie na nasze funkcje zewnętrzne i wewnętrzne	492
Stosowanie zasięgu do hermetyzacji	494
Dwie ważne reguły zasięgów w JavaScriptcie	496
Odkrywanie tajemnicy	500
Jak tworzyć domknięcia?	500
Zastosowanie domknięć w celu zaimplementowania magicznego licznika	505
Próbne odliczanie magicznego licznika	506
Implementacja licznika z użyciem domknięcia	507
Jak działa funkcja makeTimer?	511
Implementacja funkcji onlyOnceMaker	516



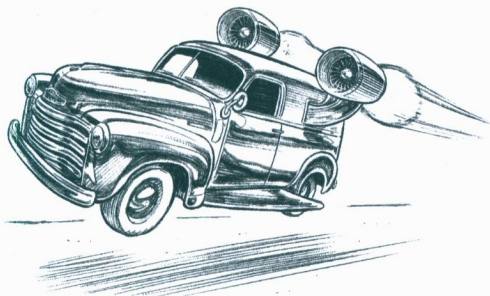
12

zaawansowane sposoby konstruowania obiektów

Tworzenie obiektów

Dotychczas wszystkie obiekty tworzyłeś własnoręcznie. Opracowując każdy z nich, korzystałeś z **literału obiektowego**, w którym podawałeś wszystkie właściwości i metody. Na niewielką skalę takie rozwiązanie będzie się sprawdzać, jednak podczas tworzenia poważnego kodu będziesz potrzebować czegoś lepszego. Właśnie w tym miejscu do akcji wkraczają **klasy**. Dzięki klasom tworzenie obiektów jest znacznie łatwiejsze, a wszystkie budowane obiekty mogą być zgodne z jednym **wzorcem** — oznacza to, że klasy zapewniają, że wszystkie obiekty będą miały te same właściwości i metody. Kiedy korzystasz z klas, kod obiektów może być znacznie bardziej **zwięzły** i mniej podatny na występowanie błędów, zwłaszcza gdy tworzysz bardzo dużo obiektów. Bierzmy się zatem do pracy...

Tworzenie obiektów przy użyciu literałów obiektowych	526
Stosowanie konwencji podczas tworzenia obiektów	527
Przedstawienie klas	529
Jak zdefiniować klasę?	530
Tworzenie obiektu przy użyciu klasy	531
Sposób działania klas	533
Dodajmy kilka metod	536
Czas na produkcję!	540
Podstawowa klasa Car	542
Implementacja klasy Taxi z użyciem extends	545
Dodawanie nowych metod do klasy Taxi	546
Implementacja klasy RocketCar	549
Stosowanie literałów obiektowych do upraszczania konstruktora	553
Przeróbka konstruktora klasy Car	556
Właściwości dostępne	557
Stosowanie metod pobierających	558
Czymże jest metoda pobierająca bez metody ustawiającej?	559
Właściwości i metody statyczne	564
Zliczanie produkowanych aut	566



pozostałości

A

Dziesięć najważniejszych zagadnień (których nie opisaliśmy)

Opisaliśmy naprawdę sporo zagadnień i już niemal udało Ci się skończyć tę książkę. Będziemy za Tobą tęsknili, jednak zanim pozwolimy Ci odejść, musimy jeszcze coś powiedzieć, bo nie czulibyśmy się w porządku, wypuszczając Cię w świat bez tych informacji. Nie ma możliwości, byśmy w tym stosunkowo niewielkim rozdziale zdążyli zmieścić wszystko, co ewentualnie mogłoby się przydać. Prawdę mówiąc, wcześniej *zamiściliśmy* w tym rozdziale wszystko to, co przydałoby się, żebyś wiedział o programowaniu w języku JavaScript, lecz musieliśmy zmniejszyć rozmiar czcionki do 0,00004 punktu. Wszystko się zmieściło, ale nikt nie był w stanie tego przeczytać. Dlatego większość tego tekstu odrzuciliśmy, pozostawiając tu jedynie dziesięć najważniejszych zagadnień.

1. Moduły	574
2. JSON	576
3. Obietnice	577
4. Przypisanie destrukuryzujące	578
5. Symbol i BigInt	579
6. Map i Set	580
7. Więcej operacji na modelu DOM	583
8. Obiekt window	584
9. JavaScript po stronie serwera	585
10. Rekurencja	586

Skorowidz	589
-----------	-----

