Оглавление

Введение

С самых первых дней существования Всемирной паутины язык JavaScript стал фундаментальной технологией для управления интерактивностью контента, потребляемого пользователями. Хотя история JavaScript начиналась с мерцающих следов от указателя мыши и раздражающих всплывающих подсказок, через два десятилетия технология и возможности JavaScript выросли на несколько порядков, и лишь немногие сомневаются в его важности как ядра самой распространенной программной платформы в мире: веб-технологий.

Но как язык JavaScript постоянно подвергался серьезной критике — отчасти из-за своего происхождения, еще больше из-за своей философии проектирования. Даже само его название наводит на мысли, как однажды выразился Брендан Эйх (Brendan Eich), о «недоразвитом младшем брате», который стоит рядом со своим старшим и умным братом Java. Тем не менее такое название возникло исключительно по соображениям политики и маркетинга. Между этими двумя языками существуют колоссальные различия. У JavaScript с Java общего не больше, чем у луна-парка с Луной.

Так как JavaScript заимствует концепции и синтаксические идиомы из нескольких языков, включая процедурные корни в стиле C и менее очевидные функциональные корни в стиле Scheme/Lisp, он в высшей степени доступен для широкого спектра разработчиков — даже обладающих минимальным опытом программирования. Программа «Hello World» на JavaScript настолько проста, что

язык располагает к себе и кажется удобным с самых первых минут знакомства.

Пожалуй, JavaScript — один из самых простых языков для изучения и начала работы, но из-за его странностей хорошее знание этого языка встречается намного реже, чем во многих других языках. Если для написания полноценной программы на C или C++ требуется достаточно глубокое знание языка, полномасштабное коммерческое программирование на JavaScript порой (и достаточно часто) едва затрагивает то, на что способен этот язык.

Хитроумные концепции, глубоко интегрированные в язык, проявляются в простых на первый взгляд аспектах, например, передаче функций в форме обратных вызовов. У разработчика JavaScript появляется соблазн просто использовать язык «как есть» и не беспокоиться о том, что происходит «внутри».

Это одновременно простой и удобный язык, находящий повсеместное применение, и сложный, многогранный набор языковых механик, истинный смысл которых без тщательного изучения останется непонятным даже для самого опытного разработчика JavaScript.

В этом заключается парадокс JavaScript; ахиллесова пята языка; проблема, которой мы сейчас займемся. Так как JavaScript можно использовать без полноценного понимания, очень часто понимание языка так и не приходит к разработчику.

Задача

Если каждый раз, сталкиваясь с каким-то сюрпризом или неприятностью в JavaScript, вы заносите их в «черный список» (как многие привыкли делать), вскоре от всей мощи JavaScript у вас останется пустая скорлупа. Хотя это подмножество принято на-

зывать «Хорошими Частями», я призываю вас, дорогой читатель, рассматривать его как «Простые Части», «Безопасные Части» и даже «Неполные Части».

Серия «Вы не знаете JS» идет в прямо противоположном направлении: изучить и глубоко понять весь язык JavaScript, и особенно «Сложные Части».

Здесь мы прямо говорим о существующей среди разработчиков JS тенденции изучать «ровно столько, сколько нужно» для работы, не заставляя себя разбираться в том, что именно происходит и почему язык работает именно так. Более того, мы воздерживаемся от распространенной тактики отступить, когда двигаться дальше становится слишком трудно.

Я не привык останавливаться в тот момент, когда что-то просто работает, а я толком сам не знаю почему, — и вам не советую. Приглашаю вас на путешествие по этим неровным, непростым дорогам; здесь вы узнаете, что собой представляет язык JavaScript и что он может сделать. С этими знаниями ни один метод, ни один фреймворк, ни одно модное сокращение или термин недели не будут за пределами вашего понимания.

В каждой из книг серии мы возьмем одну из конкретных базовых частей языка, которые часто понимаются неправильно или недостаточно глубоко, и рассмотрим ее очень глубоко и подробно. После чтения у вас должна сформироваться твердая уверенность в том, что вы понимаете не только теорию, но и практические аспекты «того, что нужно знать для работы».

Скорее всего, то, что вы сейчас знаете о JavaScript, вы узнавали по частям от других людей, которые тоже недостаточно хорошо разбирались в теме. Такой JavaScript — не более чем тень настоящего языка. На самом деле вы пока JavaScript не знаете, но будете знать, если как следует ознакомитесь с этой серией книг. Читайте дальше, друзья мои. JavaScript ждет вас.

Благодарности

Я должен поблагодарить многих людей, без которых эта книга и вся серия не появились бы на свет.

Прежде всего, я должен поблагодарить свою жену Кристен Симпсон (Christen Simpson) и своих детей Итана и Эмили — они мирились с тем, что папа подолгу сидит за компьютером. Даже когда я не пишу книги, мое увлечение JavaScript приковывает меня к экрану намного чаще, чем следовало бы. Впрочем, именно благодаря времени, позаимствованному у моей семьи, эти книги так глубоко и полно объясняют JavaScript вам, читателю. Я обязан моей семье всем.

Хочу поблагодарить своих редакторов из O'Reilly, а именно Симона Сен-Лорана (Simon St.Laurent) и Брайана Макдональда (Brian MacDonald), а также весь остальной коллектив издательства и специалистов по маркетингу. Работать с ними было невероятно интересно, и они особенно доброжелательно отнеслись к эксперименту с написанием, редактированием и печатью этой книги как проекта «с открытым кодом».

Спасибо всем, кто внес свой вклад в работу над этой серией книг, кто поделился своими предложениями и улучшениями: Шелли Пауэрс (Shelley Powers), Тим Ферро (Tim Ferro), Эван Борден (Evan Borden), Форрест Л. Норвелл (Forrest L Norvell), Дженнифер Дэвис (Jennifer Davis), Джесс Харлин (Jesse Harlin) и многие другие. Спасибо Шейну Хадсону (Shane Hudson) за отличное введение.

Спасибо бесчисленным участникам сообщества, включая представителей комитета TC39. Они делились с нами своими знаниями, терпеливо и подробно отвечали на мои бесчисленные вопросы: Джон-Дэвид Далтон (John-David Dalton), Юрий «kangax» Зайцев (Juriy Zaytsev), Матиас Байненс (Mathias Bynens), Рик

Уолдрон (Rick Waldron), Аксель Раушмайер (Axel Rauschmayer), Николас Закас (Nicholas Zakas), Энгус Кролл (Angus Croll), Джордан Харбанд (Jordan Harband), Дэйв Херман (Dave Herman), Брендан Эйх (Brendan Eich), Аллен Вирфс-Брок (Allen Wirfs-Brock), Брэдли Мек (Bradley Meck), Доменик Деникола (Domenic Denicola), Дэвид Уолш (David Walsh), Тим Дисней (Tim Disney), Крис Ковал (Kris Kowal), Петер ван дер Зее (Peter van der Zee), Андреа Джиаммарчи (Andrea Giammarchi), Кит Кэмбридж (Kit Cambridge)... и так много других, что я не затронул даже вершину айсберга.

Так как серия книг «Вы не знаете JS» родилась на Kickstarter, я также хочу поблагодарить всех моих (почти) 500 щедрых бэкеров, без которых эта серия не могла состояться. Вот они: Ян Шпила (Jan Szpila), nokiko, Мурали Кришнамурти (Murali Krishnamoorthy), Райан Джой (Ryan Joy), Крейг Пэтчетт (Craig Patchett), pdqtrader, Дэйл Фуками (Dale Fukami), Рэй Хэтфилд (ray hatfield), Родриго Перес (R0drigo Perez) [Mx], Дэн Петитт (Dan Petitt), Джек Фрэнклин (Jack Franklin), Эндрю Берри (Andrew Berry), Брайан Гринстед (Brian Grinstead), Роб Сазерленд (Rob Sutherland), Сержи Месегер (Sergi Meseguer), Филип Гурли (Phillip Gourley), Марк Уотсон (Mark Watson), Джефф Карут (Jeff Carouth), Альфредо Сумаран (Alfredo Sumaran), Мартин Сакс (Martin Sachse), Марсио Барриос (Marcio Barrios), Дэн (Dan), AimelyneM, Мэтт Салливан (Matt Sullivan), Делнатт Пьер-Антуан (Delnatte Pierre-Antoine), Джейк Смит (Jake Smith), Юджин Тудорансеа (Eugen Tudorancea), Айрис (Iris), Дэвид Трин (David Trinh), simonstl, Рэй Дэли (Ray Daly), Урос Грубер (Uros Gruber), Джастин Майерс (Justin Myers), Шай Зонис (Shai Zonis), Mom & Dad, Дэвин Кларк (Devin Clark), Дэннис Палмер (Dennis Palmer), Брайан Панахи Джонсон (Brian Panahi Johnson), Джош Маршалл (Josh Marshall), Маршалл (Marshall), Дэннис Керр (Dennis Kerr), Мэтт Стил (Matt Steele), Эрик Слагтер (Erik Slagter), Sacah, Джастин Рэйнбоу (Justin Rainbow), Кристиан Нилссон (Christian

Nilsson), Делапуи (Delapouite), Д. Перейра (D. Pereira), Николас Хойзи (Nicolas Hoizey), Джордж В. Рейли (George V. Reilly), Дэн Ривс (Dan Reeves), Бруно Латернер (Bruno Laturner), Чед Дженнингс (Chad Jennings), Шейн Кинг (Shane King), Джереми Ли Кохик (Jeremiah Lee Cohick), od3n, Стэн Ямейн (Stan Yamane), Марко Вучинич (Marko Vucinic), Jim B, Стивен Коллинз (Stephen Collins), Эгир Торстейнссон (Ægir Þorsteinsson), Эрик Педерсон (Eric Pederson), Овейн (Owain), Нейтан Смит (Nathan Smith), Jeanetteurphy, Александр (Alexandre) ELISÉ, Крис Петерсон (Chris Peterson), Рик Уотсон (Rik Watson), Люк Мэтьюз (Luke Matthews), Джастин Лоуэри (Justin Lowery), Мортен Нильсен (Morten Nielsen), Вернон Кеснер (Vernon Kesner), Четан Шеной (Chetan Shenoy), Пол Трегоинг (Paul Tregoing), Марк Грабански (Marc Grabanski), Дион Альмейр (Dion Almaer), Эндрю Салливан (Andrew Sullivan), Кейт Элзасс (Keith Elsass), Том Берк (Tom Burke), Брайан Эшенфелтер (Brian Ashenfelter), Дэвид Стюарт (David Stuart), Карл Сведберг (Karl Swedberg), Грэм (Graeme), Брэндон Хейс (Brandon Hays), Джон Кристофер (John Christopher), Gior, manoj reddy, Чед Смит (Chad Smith), Джаред Харбор (Jared Harbour), Минору Тода (Minoru TODA), Крис Уигли (Chris Wigley), Дэниел Ми (Daniel Mee), Майк (Mike), Handyface, Алекс Яраус (Alex Jahraus), Карл Фурроу (Carl Furrow), Роб Фулкрод (Rob Foulkrod), Макс Шишкин (Max Shishkin), Ли Пенни мл. (Leigh Penny Jr.), Роберт Фергусон (Robert Ferguson), Майк ван Хенселаар (Mike van Hoenselaar), Хасс Шугаард (Hasse Schougaard), Раджан Венкатагуру (rajan venkataguru), Джефф Адамс (Jeff Adams), Трей Роббинс (Trae Robbins), Рольф Лангенхузен (Rolf Langenhuijzen), Хорхе Антунес (Jorge Antunes), Алекс Колосков (Alex Koloskov), Хью Гриниш (Hugh Greenish), Тим Джонс (Tim Jones), Хосе Очоа (Jose Ochoa), Майкл Бреннан-Уайт (Michael Brennan-White), Нага Хариш Муввa (Naga Harish Muvva), Баркоци Давид (Barkóczi Dávid), Китт Ходсден (Kitt Hodsden), Пол Макгроу (Paul McGraw), Саша Голдхофер (Sascha Goldhofer), Эндрю Меткаф (Andrew Metcalf), Маркус Крог

(Markus Krogh), Майкл Мэтьюз (Michael Mathews), Мэтт Джаред (Matt Jared), Juanfran, Джорджи Киршнер (Georgie Kirschner), Кенни Ли (Kenny Lee), Тед Чжан (Ted Zhang), Амит Пахва (Amit Pahwa), Инбал Синаи (Inbal Sinai), Дэн Рейн (Dan Raine), Шабсе Лакс (Schabse Laks), Майкл Терворт (Michael Tervoort), Александр Абро (Alexandre Abreu), Алан Джозеф Уильямс (Alan Joseph Williams), NicolasD, Синди Вонг (Cindy Wong), Рег Брайтуэйт (Reg Braithwaite), LocalPCGuy, Джон Фрискис (Jon Friskics), Крис Мерриман (Chris Merriman), Джон Пена (John Pena), Джейкоб Кац (Jacob Katz), Сью Локвуд (Sue Lockwood), Магнус Йоханссон (Magnus Johansson), Джереми Крэпси (Jeremy Crapsey), Гжегож Павловски (Grzegorz Pawłowski), Нико Нуззачи (nico nuzzaci), Кристин Уилкс (Christine Wilks), Ханс Бергрен (Hans Bergren), Чарльз Монтгомери (charles montgomery), Ариэль Фогел (Ariel דב-בבל Fogel), Иван Колев (Ivan Kolev), Дэниел Кампос (Daniel Campos), Хью Вуд (Hugh Wood), Кристиан Брэдфорд (Christian Bradford), Фредерик Харпер (Frédéric Harper), Ионут Дан Попа (Ionuț Dan Popa), Джефф Тримбл (Jeff Trimble), Руперт Вуд (Rupert Wood), Трей Каррико (Trey Carrico), Панчо Лопес (Pancho Lopez), Джоэл Куйтен (Joël kuijten), Том А. Марра (Tom A Marra), Джефф Джуисс (Jeff Jewiss), Джейкоб Риос (Jacob Rios), Паоло Ди Стефано (Paolo Di Stefano), Соледад Пенадес (Soledad Penades), Крис Гербер (Chris Gerber), Андрей Долганов (Andrey Dolganov), Уилл Мур III (Wil Moore III), Томас Мартино (Thomas Martineau), Карим (Kareem), Бен Type (Ben Thouret), Уди Нир (Udi Nir), Морган Лаупис (Morgan Laupies), Джори Карсон-Берсон (jory carson-burson), Натан Л. Смит (Nathan L. Smith), Эрик Дэймон Уолтерс (Eric Damon Walters), Дерри Лозано-Хойленд (Derry Lozano-Hoyland), Джеоффри Уайзман (Geoffrey Wiseman), mkeehner, KatieK, Скотт Макфарлейн (Scott MacFarlane), Брайан Лашомб (Brian LaShomb), Адриен Мас (Adrien Mas), Кристофер Росс (christopher ross), Иэн Литтман (Ian Littman), Дэн Аткинсон (Dan Atkinson), Эллиот Джоуб (Elliot Jobe), Ник Дозье (Nick Dozier), Питер Вули (Peter Wooley), Джон Гувер (John Hoover),

dan, Мартин Э. Джексон (Martin A. Jackson), Гектор Фернандо Хуртадо (Héctor Fernando Hurtado), Энди Эннаморато (andy ennamorato), Пол Селтман (Paul Seltmann), Мелисса Гор (Melissa Gore), Дэйв Поллард (Dave Pollard), Джек Смит (Jack Smith), Филип Да Силва (Philip Da Silva), Гай Израэли (Guy Israeli), @megalithic, Дэмиан Кроуфорд (Damian Crawford), Феликс Глише (Felix Gliesche), Эйприл Картер Грант (April Carter Grant), Хайди (Heidi), Джим Тирни (jim tierney), Андреа Джиаммарчи (Andrea Giammarchi), Нико Виньола (Nico Vignola), Дон Джонс (Don Jones), Крис Хартьес (Chris Hartjes), Алекс Хоуз (Alex Howes), Джон Гиббон (john gibbon), Дэвид Дж. Грум (David J. Groom), BBox, Ю Дилис Сун (Yu Dilys Sun), Нэйт Стейнер (Nate Steiner), Брэндон Сатром (Brandon Satrom), Брайан Уайант (Brian Wyant), Уэсли Хейлз (Wesley Hales), Иэн Паунси (Ian Pouncey), Тимоти Кевин Оксли (Timothy Kevin Oxley), Джордж Терезакис (George Terezakis), Санджай Радж (sanjay raj), Джордан Харбанд (Jordan Harband), Марко Маклайон (Marko McLion), Вольфганг Кауфман (Wolfgang Kaufmann), Паскаль Пеккерт (Pascal Peuckert), Дэйв Нагент (Dave Nugent), Маркус Либелт (Markus Liebelt), Уэллинг Гусман (Welling Guzman), Ник Кули (Nick Cooley), Дэниел Мескита (Daniel Mesquita), Роберт Сайварт (Robert Syvarth), Крис Койе (Chris Coyier), Реми Бах (Rémy Bach), Адам Дугал (Adam Dougal), Алистер Даггин (Alistair Duggin), Дэвид Лойдолт (David Loidolt), Эд Ричер (Ed Richer), Брайан Шено (Brian Chenault), GoldFire Studios, Карлес Андре (Carles Andrés), Карлос Кабо (Carlos Cabo), Юя Сайто (Yuya Saito), Роберто Рикардо (roberto ricardo), Барнетт Клейн (Barnett Klane), Майк Мур (Mike Moore), Кевин Маркс (Kevin Marx), Джастин Лав (Justin Love), Джо Тейлор (Joe Taylor), Пол Дижу (Paul Dijou), Майкл Колер (Michael Kohler), Роб Кэсси (Rob Cassie), Майк Тирни (Mike Tierney), Коди Лерой Линдли (Cody Leroy Lindley), tofuji, Шимон Шварц (Shimon Schwartz), Рэймонд (Raymond), Люк Де Бруве (Luc De Brouwer), Дэвид Хейс (David Hayes), Рис Бретт-Боуэн (Rhys Brett-Bowen), Дмитрий (Dmitry), Азиз Хури (Aziz Khoury), Дин (Dean), Скотт

Толински (Scott Tolinski Level Up), Клеман Буари (Clement Boirie), Джордже Лукич (Djordje Lukic), Антон Котенко (Anton Kotenko), Рафаэль Коррал (Rafael Corral), Филипп Гурвиц (Philip Hurwitz), Джонатан Пиджин (Jonathan Pidgeon), Джейсон Кэмпбелл (Jason Campbell), Джозеф С. (Joseph C.), SwiftOne, Иэн Хонер (Jan Hohner), Дерик Бэйли (Derick Bailey), getify, Дэниел Кузино (Daniel Cousineau), Крис Чарлтон (Chris Charlton), Эрик Тернер (Eric Turner), Дэвид Тернер (David Turner), Джоэл Галеран (Joël Galeran), Dharma Vagabond, Адам (adam), Дирк ван Берген (Dirk van Bergen), dave ♥♫★ furf, Ведран Закани (Vedran Zakanj), Райан Макаллен (Ryan McAllen), Натали Пэтрис Такер (Natalie Patrice Tucker), Эрик Дж. Бивона (Eric J. Bivona), Адам Спунер (Adam Spooner), Аарон Кавано (Aaron Cavano), Келли Пэкер (Kelly Packer), Эрик Джней (Eric J), Мартин Дреновак (Martin Drenovac), Эмилис (Emilis), Майкл Пеликан (Michael Pelikan), Скотт Ф. Уолтер (Scott F. Walter), Джош Фримен (Josh Freeman), Брэндон Хадженс (Brandon Hudgeons), Виджай Ченнупати (vijay chennupati), Билл Гленнон (Bill Glennon), Робин Р. (Robin R.), Трой Форстер (Troy Forster), otaku_coder, Брэд (Brad), Скотт (Scott), Фредерик Острандер (Frederick Ostrander), Адам Брилл (Adam Brill), Себ Флиппенс (Seb Flippence), Майкл Андерсон (Michael Anderson), Джейкоб (Jacob), Адам Рэндлетт (Adam Randlett), Standard, Джошуа Клэнтон (Joshua Clanton), Себастиан Куба (Sebastian Kouba), Крис Дек (Chris Deck), SwordFire, Ханнес Папенберг (Hannes Papenberg), Ричард Вобер (Richard Woeber), hnzz, Роб Кроутер (Rob Crowther), Джедидайя Бродбент (Jedidiah Broadbent), Сергей Чернышев (Sergey Chernyshev), Джей-Эр Хамон (Jay-Ar Jamon), Бен Комби (Ben Combee), Лучиано Боначела (luciano bonachela), Марк Томлинсон (Mark Tomlinson), Кит Кэмбридж (Kit Cambridge), Майкл Мелгарес (Michael Melgares), Джейкоб Адамс (Jacob Adams), Адриан Бруинхаут (Adrian Bruinhout), Бев Вибер (Bev Wieber), Скотт Пулео (Scott Puleo), Томас Херцог (Thomas Herzog), Эйприл Леоне (April Leone), Дэниел Мизилински (Daniel Mizieliński), Кеес ван Гинкел

(Kees van Ginkel), Джон Абрамс (Jon Abrams), Эрвин Хайзер (Erwin Heiser), Ави Лавиад (Avi Laviad), Дэвид Ньюэлл (David Newell), Жан-Франсуа Тюрко (Jean-Francois Turcot), Нико Робертс (Niko Roberts), Эрик Дана (Erik Dana), Чарльз Нилл (Charles Neill), Аарон Холмс (Aaron Holmes), Гжегож Циолковски (Grzegorz Ziółkowski), Нейтан Янгман (Nathan Youngman), Тимоти (Timothy), Джейкоб Мазер (Jacob Mather), Майкл Аллен (Michael Allan), Мохит Сет (Mohit Seth), Райан Эвинг (Ryan Ewing), Бенджамин Ван Тризе (Benjamin Van Treese), Марсело Сантос (Marcelo Santos), Денис Вольф (Denis Wolf), Фил Киз (Phil Keys), Крис Юнг (Chris Yung), Тимо Тийхоф (Timo Tijhof), Мартин Леквалл (Martin Lekvall), Agendine, Грег Уитворт (Greg Whitworth), Хелен Хамфри (Helen Humphrey), Дугал Кэмпбелл (Dougal Campbell), Йоханнес Харт (Johannes Harth), Бруно Гирин (Bruno Girin), Брайан Хоу (Brian Hough), Даррен Ньютон (Darren Newton), Крейг Макфит (Craig McPheat), Оливье Тилл (Olivier Tille), Деннис Ротиг (Dennis Roethig), Матиас Байненс (Mathias Bynens), Брендан Стромбергер (Brendan Stromberger), sundeep, Джон Мейер (John Meyer), Рон Мэйл (Ron Male), Джон Ф. Кростон III (John F Croston III), gigante, Карл Бергенхэм (Carl Bergenhem), Б. Дж. Мэй (B.J. May), Ребека Тайлер (Rebekah Tyler), Тед Фоксберри (Ted Foxberry), Джордан Риз (Jordan Reese), Терри Сьютор (Terry Suitor), afeliz, Том Кифер (Tom Kiefer), Даррах Даффи (Darragh Duffy), Кевин Вандербекен (Kevin Vanderbeken), Энди Пирсон (Andy Pearson), Саймон Макдональд (Simon Mac Donald), Абид Дин (Abid Din), Крис Джоэл (Chris Joel), Томас Тониссен (Tomas Theunissen), Дэвид Дик (David Dick), Пол Грок (Paul Grock), Брэндон Вуд (Brandon Wood), Джон Вайс (John Weis), dgrebb, Ник Дженкинс (Nick Jenkins), Чак Лейн (Chuck Lane), Джонни Мегахан (Johnny Megahan), marzsman, Тату Тамминен (Tatu Tamminen), Джеоффри Кнаут (Geoffrey Knauth), Александр Тармолов (Alexander Tarmolov), Джереми Таймс (Jeremy Tymes), Чед Олд (Chad Auld), Шон Пармели (Sean Parmelee), Роб Стенке (Rob Staenke), Дэн Бендер (Dan Bender), Янник Дерва (Yannick

Derwa), Джошуа Джонс (Joshua Jones), Герт Плейзир (Geert Plaisier), Том Лезотт (Tom LeZotte), Кристен Симпсон (Christen Simpson), Стефан Брувик (Stefan Bruvik), Джастин Фальконе (Justin Falcone), Карлос Сантана (Carlos Santana), Майкл Вайс (Michael Weiss), Пабло Виллослада (Pablo Villoslada), Петер де-Хаан (Peter deHaan), Димитрис Илиопулос (Dimitris Iliopoulos), seyDoggy, Адам Джорденс (Adam Jordens), Ной Кантровиц (Noah Kantrowitz), Эмол М. (Amol M), Мэтью Уиннард (Matthew Winnard), Дирк Гинейдер (Dirk Ginader), Финэм Буи (Phinam Bui), Дэвид Рэпсон (David Rapson), Эндрю Бакстер (Andrew Baxter), Флориан Бугел (Florian Bougel), Майкл Джордж (Michael George), Альбан Эскалье (Alban Escalier), Дэниел Селлерс (Daniel Sellers), Саша Рудан (Sasha Rudan), Джон Грин (John Green), Роберт Ковальски (Robert Kowalski), Дэвид И. Тезейра (David I. Teixeira), @ditma, Чарльз Карпентер (Charles Carpenter), Джастин Йост (Justin Yost), Сэм С. (Sam S), Денис Чиккале (Denis Ciccale), Кевин Шорс (Kevin Sheurs), Янник Круассан (Yannick Croissant), По Фрейсес (Pau Fracés), Стивен Макгоуэн (Stephen McGowan), Шон Сирси (Shawn Searcy), Крис Раппел (Chris Ruppel), Кевин Лэмпинг (Kevin Lamping), Джессика Кэмпбелл (Jessica Campbell), Кристофер Шмитт (Christopher Schmitt), Sablons, Джонатан Рейсдорф (Jonathan Reisdorf), Банни Гек (Bunni Gek), Тедди Хафф (Teddy Huff), Майкл Маллани (Michael Mullany), Майкл Фюрстенберг (Michael Fürstenberg), Карл Хендерсон (Carl Henderson), Рик Йостинг (Rick Yoesting), Скотт Николс (Scott Nichols), Эрнан Сьюдад (Hernán Ciudad), Эндрю Майер (Andrew Maier), Майк Стапп (Mike Stapp), Джесси Шоул (Jesse Shawl), Серхио Лопес (Sérgio Lopes), jsulak, Шон Прайс (Shawn Price), Джоэл Клермон (Joel Clermont), Крис Ридманн (Chris Ridmann), Шон Тимм (Sean Timm), Джейсон Финч (Jason Finch), Эйден Монтгомери (Aiden Montgomery), Элайджа Мэнор (Elijah Manor), Дерек Гатрайт (Derek Gathright), Джесси Харлин (Jesse Harlin), Диллон Карри (Dillon Curry), Кортни Майерс (Courtney Myers), Диего Каденас (Diego Cadenas), Арн де Бри (Arne de Bree), Жоао Пауло Дуба (João Paulo Dubas), Джеймс

Тейлор (James Taylor), Филипп Креутли (Philipp Kraeutli), Михай Паун (Mihai Păun), Сэм Гарегозлу (Sam Gharegozlou), joshjs, Мэтт Мерчисон (Matt Murchison), Эрик Уиндэм (Eric Windham), Тимо Берманн (Timo Behrmann), Эндрю Холл (Andrew Hall), Джошуа Прайс (joshua price) и Теофил Виллар (Théophile Villard).

Эта книга писалась на условиях открытого доступа к информации, включая редактирование и печать. Мы многим обязаны репозиторию Github, благодаря которому перед сообществом открылись такие возможности!

Спасибо всем бесчисленным людям, которых я не назвал, но которые заслуживают благодарности. Пусть эта книжная серия «принадлежит» всем нам, и пусть она внесет свой вклад в популяризацию и понимание языка JavaScript к пользе всех нынешних и будущих участников сообщества.

О книге

JavaScript — замечательный язык. Его легко изучать частично и намного сложнее изучать полностью (или хотя бы в достаточной мере). Когда разработчики сталкиваются с трудностями, они обычно винят в этом язык вместо своего ограниченного понимания. В этих книгах я постараюсь исправить такое положение дел и помогу оценить по достоинству язык, который вы можете (и должны) знать достаточно глубоко.

Многие примеры, приведенные в книге, рассчитаны на современные (и обращенные в будущее) среды JavaScript, такие как ES6. При запуске в более старых версиях движка (предшествующих ES6) поведение некоторых примеров может отличаться от описанного в тексте.

Типографские соглашения

В этой книге приняты следующие типографские соглашения:

Курсив

Используется для обозначения новых терминов.

`Моноширинный шрифт`

Применяется для оформления листингов программ и программных элементов внутри обычного текста, таких как имена переменных и функций, базы данных, типы данных, переменные окружения, инструкции и ключевые слова.

`Моноширинный курсив`

Обозначает текст, который должен замещаться фактическими значениями, вводимыми пользователем или определяемыми из контекста.

Так выделяются советы и предложения.

Так обозначаются советы, предложения и примечания общего характера.

Так обозначаются предупреждения и предостережения.

Использование программного кода примеров

Вспомогательные материалы (примеры кода, упражнения и т. д.) доступны для загрузки по адресу http://bit.ly/1c8HEWF и http://bit.ly/ydkjs-this-code.

Эта книга призвана оказать вам помощь в решении ваших задач. В общем случае все примеры кода из этой книги вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам необходимо получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию вам необходимо будет получить разрешение издательства.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

ЧАСТЬ 1

Область видимости и замыкания

Предисловие

В детстве мне нравилось разбирать и собирать всякие устройства: старые мобильные телефоны, стереосистемы и вообще все, до чего я мог добраться. Я был еще слишком мал, чтобы нормально пользоваться гаджетами, но каждый раз, когда что-то ломалось, я сразу же спрашивал у родителей, можно ли мне разобраться, как оно работает.

Помню, однажды я рассматривал печатную плату старого радиоприемника. На ней была какая-то странная длинная трубка, обмотанная медным проводом. Я не мог понять ее назначение и тут же перешел в режим исследования. Что она делает? Зачем она в приемнике? Она не похожа на другие части платы, почему? А для чего ее обмотали медным проводом? Что будет, если снять провод? Теперь я знаю, что это была рамочная антенна, которая строится наматыванием медного провода на ферритовый сердечник и которая часто используется в транзисторных приемниках.

А вам интересно пытаться искать ответы на все «почему»? Большинству детей это интересно. Пожалуй, мне это больше всего нравится в детях — желание узнавать новое.

К сожалению, теперь я считаюсь профессионалом и провожу будни за созданием новых вещей. Когда я был молод, мне нравилась представлять, что когда-нибудь я буду создавать те вещи, которые разбираю сегодня. Конечно, большинство того, что я создаю сегодня, представляет собой код JavaScript, а не ферритовые сердечники... но не так уж далеко код от них ушел! Но хотя мне когда-то нравилась идея создания, сейчас мне часто не хватает

желания разбирать и разбираться. Конечно, я часто выясняю лучший способ решения задачи или исправления ошибок, но редко трачу время на то, чтобы ставить под сомнение свои инструменты.

Именно по этой причине мне так интересна серия книг «Вы не знаете JS». Это *правильно*. Я не знаю JS. Я использую JavaScript в своей повседневной работе, занимаюсь этим уже много лет, но действительно ли понимаю этот язык? Нет. Конечно, я понимаю многие его стороны, часто читаю спецификации и рассылки, но нет — я не понимаю его в той мере, как того желал бы мой внутренний двойник в шестилетнем возрасте.

Книга, которую вы держите в руках, — прекрасное начало серии. Она ориентирована на таких людей, как я (и будем надеяться, и вы). Она не учит вас JavaScript с нуля. Она показывает, насколько мало вы знаете о внутреннем устройстве языка. Кроме того, книга вышла в идеальный момент: стандарт ES6 постепенно приживается, а в разных браузерах успешно идет работа над его реализацией. Если вы еще не взялись за изучение новых возможностей (таких, как `let` и `const`), это издание станет отличным введением в тему.

Итак, я надеюсь, что книга вам понравится. Но я еще сильнее надеюсь, что подход Кайла, его привычка критически осмысливать каждый крошечный фрагмент языка, закрепится у вас в мозгу и в общем рабочем процессе. Не используйте антенну просто так, разберитесь в том, как и почему она работает.

Шейн Хадсон (Shane Hudson)
www.shanehudson.net

1 Что такое область видимости?

Одна из самых фундаментальных парадигм почти всех языков программирования — возможность хранения значений в переменных и последующего чтения или изменения этих значений. Возможность сохранения и чтения значений из переменных — то, что образует *состояние* программы.

Без этой концепции программа сможет выполнять некоторые операции, но они будут в высшей степени ограниченными и не особенно интересными.

Однако включение переменных в программу поднимает самый интересный вопрос, которым мы сейчас займемся: где размещаются эти переменные? Другими словами, где они хранятся? И самое важное, как ваша программа находит их, когда в них возникнет надобность?

Эти вопросы показывают, почему так необходим четко определенный набор правил для хранения переменных в определенном месте и их нахождения в будущем. Этот набор правил называется *областью видимости* (scope).

Но где и как задаются правила области видимости?

Немного теории компиляторов

Это может быть очевидно, а может быть и удивительно, в зависимости от вашего опыта работы с разными языками, но несмотря на тот факт, что JavaScript относится к общей категории «динамических» или «интерпретируемых» языков, на самом деле это компилируемый язык. Код не компилируется заранее, как во многих традиционных компилируемых языках, а результаты компиляции не портируются между разными распределенными системами. Тем не менее движок JavaScript выполняет многие те же действия, что и любой традиционный компилятор (хотя и на более сложном уровне).

В традиционном процессе компиляции блок исходного кода — ваша программа — *перед* выполнением обычно проходит через три фазы обработки, которые приближенно объединяются термином «компиляция»:

- *Лексический анализ/Разбиение на токены (Tokenizing/Lexing)* — разбиение последовательности символов на осмысленные (с точки зрения языка) фрагменты, называемые *токенами*. Для примера возьмем программу `var a = 2;`. Скорее всего, эта программа будет разбита на следующие токены: `var`, `a`, `=`, `2` и `;`. Пропуски могут сохраняться в виде токенов, а могут и не сохраняться в зависимости от того, имеет это смысл или нет.

Разница между *tokenizing* и *lexing* — вопрос достаточно тонкий и теоретический. Важно то, будет ли происходить идентификация токеном с состоянием или без. Проще говоря, если при вызове токенизатора активизируются правила разбора с состоянием, определяющие, должен ли данный токен считаться отдельным токеном или частью другого токена, это будет называться lexing.

- *Разбор (parsing)* — преобразование потока (массива) токенов в дерево вложенных элементов, которые в совокупности пред-