

КАЙЛ
СИМПСОН

ПОЗНАКОМЬТЕСЬ, JAVASCRIPT

2-Е ИЗДАНИЕ

{
ВЫ ПОКА
ЕЩЕ
НЕ ЗНАЕТЕ
JS
}



You Don't Know JS Yet: Get Started

Get to know JS

Kyle Simpson

КАЙЛ
СИМПСОН

ПОЗНАКОМЬТЕСЬ, JAVASCRIPT

2-Е МЕЖДУНАРОДНОЕ ИЗДАНИЕ

{
ВЫ ПОКА
ЕЩЕ
НЕ ЗНАЕТЕ
JS
}



Санкт-Петербург • Москва • Минск

2022

ББК 32.988.02-018
УДК 004.738.5
С37

Симпсон Кайл

С37 {Вы пока еще не знаете JS} Познакомьтесь, JavaScript.
2-е изд. — СПб.: Питер, 2022. — 192 с.: ил. — (Серия
«Библиотека программиста»).

ISBN 978-5-4461-1875-5

Вы пока еще не знаете JS. И Кайл Симпсон признается, что тоже его не знает (по крайней мере полностью)... И никто не знает. Но все мы можем начать работать над тем, чтобы узнать его лучше. Сколько бы времени вы ни провели за изучением языка, всегда можно найти что-то еще, что стоит изучить и понять на другом уровне. Учтите, что, хотя книга и называется «Познакомьтесь, JavaScript», она не для новичков. У нее другая задача: дать обзор тем, в которых необходимо разобраться на начальном этапе изучения JS. Даже если вы уже написали достаточно кода JS, эту книгу не стоит пропускать, возможно, в ваших знаниях есть пробелы, которые необходимо заполнить перед углубленным изучением сложных тем. Пора начать изучение JS!

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г.
№ 436-ФЗ.)

ББК 32.988.02-018
УДК 004.738.5

Права на издание получены по соглашению с Kyle Simpson. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

ISBN 979-8602477429 англ.
ISBN 978-5-4461-1875-5

© Kyle Simpson
© Перевод на русский язык
ООО Издательство «Питер», 2022
© Издание на русском языке, оформление
ООО Издательство «Питер», 2022
© Серия «Библиотека программиста», 2022

Оглавление

Благодарности	9
Предисловие	11
Вступление	14
Части языка	15
Название?	17
Миссия	19
Путь	20
От издательства	22
Глава 1. Что такое JavaScript?	23
О книге	24
Откуда взялось название?	26
Спецификация языка	28
Веб (JS)	32
Не только (веб) JS... ..	34
Не всегда JS	36
Многоликий язык	39
Прямая и обратная совместимость	41
Транспилиция	44

Полифилы.	47
Что такое интерпретация?	50
WASM (Web Assembly)	57
Строго говоря.	61
После определения.	65
Глава 2. Обзор возможностей JS.	67
Каждый файл является программой.	69
Значения	71
Массивы и объекты	76
Определение типа значения.	78
Объявление и использование переменных	79
Функции	84
Сравнения	87
Равно... или типа того.	88
Сравнения с преобразованием типа.	92
Организация кода JS	96
Классы	97
Наследование классов	99
Модули	103
Классические модули	103
Модули ES.	107
Кроличья нора становится глубже.	111
Глава 3. JS: копаем вглубь	113
Итерации	115
Потребление итераторов.	117
Итерируемые значения	119
Замыкания	123
Ключевое слово this.	128

Прототипы	133
Связывание объектов	134
Снова о this.	138
А теперь — «почему?»	140
Глава 4. Общая картина	143
Столп 1: области видимости и замыкания	145
Столп 2: прототипы	147
Столп 3: типы и преобразования	149
По ветру	151
По порядку	155
Приложение А. Дальнейшее изучение	159
Значения и ссылки	160
Многоликие функции	163
Условное сравнение с преобразованием типа	169
Прототипические классы	171
Приложение Б. Практика, практика, практика!	175
Сравнения	176
Замыкания	177
Прототипы	178
Предлагаемые решения	181

Благодарности

Прежде всего спасибо моей жене и детям. Их постоянная поддержка позволила мне продолжать работу. Также хочу поблагодарить 500 бэкеров первого издания «Вы не знаете JS» (YDKJS) на Kickstarter, а также сотни тысяч людей, которые купили и прочли эти книги. Без вашей финансовой поддержки второе издание не состоялось бы. Также спасибо интервьюеру из одной соцсети с птичьим названием, который сказал, что я «недостаточно знаю JS», чем помог мне выбрать название для серии книг.

Своей карьерой я в значительной мере обязан Марку Грабански (Marc Grabanski) и FrontendMasters. Много лет назад Марк оказал мне доверие и помог сделать первые шаги в области преподавания. Если бы не он, я не начал бы писать книги! Frontend Masters является главным спонсором «Вы все еще не знаете JS» (2-е издание). Спасибо вам, Frontend Masters (и Марк!).

Наконец, мой редактор Саймон Сен-Лоран (Simon St. Laurent) помог мне определиться с первоначальным замыслом серии YDKJS и стал редактором моей первой книги. Поддержка и советы Саймона оказали на меня серьезное влияние, и именно благодаря им я в значительной мере сформировался как автор. Прошло много лет с тех пор, как за выпивкой в Driskill родился замысел YDKJS. Спасибо тебе, Саймон, за все эти годы, что ты указывал мне путь и улучшал эти книги!

Предисловие

Когда я впервые увидел твит с рекламой сбора денег на оригинальную серию книг «Вы не знаете JS», я подумал: кем бы ни был этот Кайл Симпсон, пускай идет лесом. Конечно же, я знаю JavaScript! В то время я работал на JavaScript уже много лет с самыми авторитетными представителями отрасли и поэтому считал, что на подобные обобщения могу смотреть свысока.

После того как кампания завершилась, я заметил, что многие люди с большим энтузиазмом относятся к этой серии. И тогда я решил все же попробовать — просто чтобы показать всем, что я знаю JavaScript. Но когда я погрузился в материал и стал внимательно изучать текст, то испытал глубокое чувство удивления, любопытства и даже некоторого раздражения. У Кайла настоящий дар говорить что-то такое, что противоречит моим представлениям о мире, и заставлять меня думать об этом, пока я не пойму, что сказанное было правдой (хотя я ему в этом никогда не признаюсь).

В общем, выяснилось, что JavaScript я не знал. Я не знал, почему были приняты те или иные паттерны; я не знаю, почему в некоторых ситуациях JavaScript ведет себя именно так, а не иначе; я не знал многие нюансы языка, которые, как мне казалось, были мне известны. Я в принципе не догадывался, что не знал многих аспектов, и это снижало мою квалификацию как разработчика.

Именно этим и примечательна книга. Она написана не только для тех, кто хочет освоить новый язык (хотя и для них тоже); она написана для ремесленников от программирования, которые хотят мастерски овладеть своими инструментами, понимать все тонкости своего ремесла и выбирать подходящие средства для решения задач.

В Кайле и его работе мне прежде всего нравится то, что он остается восхитительно непоколебимым перед влиянием окружающего его мира программирования. Это вовсе не означает, что он не в курсе происходящего в сообществе; скорее он не отклоняется от своего поиска лучших ответов на правильно поставленные вопросы. Часто он начинает противоречить новейшим «лучшим практикам», но в действительности это именно то, что нужно: точка зрения, которая вне времени и просто выявляет истину. Этим-то так хороша эта серия. Первое издание «Вы не знаете JS» остается актуальным даже годы спустя! Немногие

книги выдержали проверку временем с учетом меняющегося ландшафта JavaScript.

Несколько слов о первой книге, «Познакомьтесь, JavaScript». Возможно, у вас возникнет соблазн пропустить ее, потому что вы думаете, что «первые шаги» уже давно сделаны. И все же вам стоит ее прочитать! Вы не поверите, сколько глубины, нюансов и странностей кроется в основных структурных элементах JavaScript. Очень важно, чтобы вы освоили всю эту подоплеку, прежде чем переключаться на конструкции языка. Именно такой фундамент понадобится вам для того, чтобы действительно хорошо знать JavaScript.

Итак, окажите себе услугу, основательно прочитайте эту книгу и высвободите содержащиеся в ней знания. Этот надежный фундамент послужит вам лучше, чем любой фреймворк или библиотека; они приходят и уходят, но все мы еще будем писать код JavaScript в ближайшие десятилетия. Сохраняйте объективность и ставьте под сомнение свои стереотипы.

Потому что, как я сам убедился, скорее всего, вы не знаете JavaScript (пока).

*Брайан Холт (Brian Holt),
старший разработчик Visual Studio Code
и Node.js для Azure Microsoft*

Вступление

Вашему вниманию предлагается 2-е издание снискавшей популярность серии книг «Вы не знаете JS»: «Вы пока еще не знаете JS» (YDKJSY).

Если вы уже читали предыдущее издание, то заметите, что в этом появился обновленный подход к изложению с подробными описаниями того, что изменилось в JS за последние 5 лет.

Я надеюсь и верю, что вы все еще сохраняете то же стремление изучить JS и разобраться в том, как он устроен.

Если вы читаете эти книги впервые, я рад, что они попались вам на глаза. Подготовьтесь к увлекательному путешествию по закоулкам JavaScript.

Если вы недавно занимаетесь программированием или JS, то учтите, что эти книги не задумывались как «деликатный вводный курс по JavaScript». Временами материал становится сложным и требующим се-

рьезных усилий для понимания, и многие темы рассматриваются намного глубже, чем в книгах для новичков. Книга может пригодиться всем читателям независимо от уровня подготовки, но я писал ее с прицелом на то, что вы уже знакомы с JS, а ваш практический опыт работы с этим языком составляет хотя бы полгода, если не больше.

Части языка

В этих книгах я намеренно отошел от традиционного подхода, в котором рассматриваются *хорошие части* языка. Нет, это не означает, что мы будем рассматривать только *плохие части* — скорее рассматриваться будут **все части**.

Возможно, вы слышали (или сами считаете), что JS — глубоко ущербный язык, плохо спроектированный и непоследовательно реализованный. Многие считают, что это худший из популярных языков, что никто не пишет код JS добровольно, а только из-за того, что он занял свое место в сети. Это смехотворные, нездоровые и высокомерные утверждения.

Миллионы разработчиков ежедневно пишут код JavaScript, и многие из них уважают и ценят этот язык.

Как и у любого великого языка, у него есть как выдающиеся достоинства, так и недостатки. Даже сам создатель JavaScript Брендан Эйх сожалеет по пово-

ду некоторых частей и называет их ошибками. Но он заблуждается: они вовсе не были ошибками. В наши дни JS стал тем, чем он стал — самым распространенным, а следовательно, самым влиятельным языком программирования, — именно из-за *всех этих частей*.

Не ведитесь на утверждения, будто вам следует изучить и использовать только небольшой набор хороших частей, а от всего плохого нужно держаться подальше. Не ведитесь на шарлатанство «X — это новый Y», будто с появлением в языке некоторой новой возможности все предшествующее использование старой функциональности мгновенно устаревает и отмирает. Не слушайте, когда кто-то вам говорит, что ваш код «не современен», потому что в нем еще не используется функция стадии 0, предложенная лишь несколько недель назад!

Все части JS полезны. Некоторые части полезнее других. Некоторые требуют действовать более внимательно и осознанно.

На мой взгляд, абсурдно даже пытаться стать по-настоящему эффективным разработчиком JavaScript, используя только узкий срез возможностей языка. Можно ли представить рабочего с полным ящиком инструментов, который пользуется только молотком, а отвертку и рулетку презирует, считая их недостойными? Это просто глупо.

Я утверждаю, что изучать нужно все части JavaScript и пользоваться ими там, где они уместны! И я даже наберусь смелости предложить: выбросьте все книги, в которых говорится обратное.

Название?

Какой же смысл заложен в название серии?

Я не пытаюсь обидеть вас, ставя под сомнение ваш уровень знания или понимания JavaScript. Я не предполагаю, что вы не можете или не сможете изучить JavaScript. Я не хвастаюсь некими секретными тайными знаниями, которыми обладаю только я и еще несколько избранных.

Серьезно, все это реальные реакции на название оригинальной серии, которые появились еще до того, как книги увидели свет. И они совершенно необоснованны.

Главный смысл названия «Вы пока еще не знаете JS» — подчеркнуть, что большинство разработчиков JS не тратит время на то, чтобы по-настоящему понять, как работает написанный ими код. Они знают, что код *работает* — он выдает желаемый результат. Но они либо не понимают, *как* он работает, либо, что еще хуже, руководствуются неточной ментальной моделью, которая дает сбой при ближайшем рассмотрении.

Я предлагаю вам отложить все свои допущения по поводу JS, взглянуть на язык свежим взглядом и подойти к нему с заново пробужденной любознательностью. Спрашивайте себя «почему?» каждый раз, когда пишете строчку. Почему она работает именно так, а не иначе? Почему один способ лучше или уместнее пяти-шести других возможных решений? Почему все «лидеры мнений» предлагают делать X в вашем коде, но выясняется, что вариант Y оказывается лучше?

Я добавил в название «пока» не только потому, что это второе издание, но и из-за того, что в конечном итоге я хочу, чтобы книги вселяли в вас надежду, а не наоборот.

Не думаю, что JS вообще возможно знать полностью. Это не достижение, которое необходимо получить, а цель, к которой нужно стремиться. Не думайте, что вы все узнаете о JS и на этом все закончится; нет, вы просто продолжаете учиться, все чаще практикуясь в написании кода. И чем глубже вы погружаетесь, тем чаще возвращаетесь к тому, что изучали ранее, и переосмысливаете его с позиций более опытного разработчика.

Рекомендую сформировать особую систему взглядов на JavaScript (и на разработку в целом): вы никогда не освоите его полностью, но можете (и должны) работать над тем, чтобы приблизиться к этой цели.

Этот путь растянется на всю вашу карьеру разработчика и даже дальше.

Вы всегда можете знать JS лучше, чем сейчас. Надеюсь, именно эту мысль передают книги серии YDKJSY.

Миссия

На самом деле не нужно обосновывать, почему разработчики должны относиться к JS серьезно, — думаю, язык уже доказал, что заслуживает статуса первого-классного среди языков программирования.

Важно обосновать другое, более глобальное утверждение, и эти книги пытаются справиться с этой задачей.

Я обучал более 5000 разработчиков из групп и компаний по всему миру более чем в 25 странах на шести континентах. Мне часто приходилось видеть, что главным фактором считается только результат программы, а не то, как программа написана или как/почему она работает.

Мой опыт не только как разработчика, но и как преподавателя говорит мне: вы всегда можете повысить эффективность своего труда, если четко будете понимать, как работает ваш код (а не просто добиваться того, чтобы он выдавал желаемый результат).

Иначе говоря, «код достаточно хорош, чтобы работать» — не то же самое, что «код достаточно хорош» (и не должно быть тем же самым).

Всем разработчикам постоянно приходится мучиться с каким-нибудь блоком кода, который по неизвестной причине работает неправильно. Но слишком часто разработчики JS обвиняют язык, вместо того чтобы винить себя за нехватку понимания. Эти книги служат вопросом и ответом: почему произошло именно *это* и как нужно действовать, чтобы произошло *вот это*.

Моя миссия — дать возможность каждому разработчику JS полностью контролировать написанный им код, понять его и программировать сознательно и ясно.

Путь

Некоторые из вас начали читать эту книгу с целью изучить все шесть книг от начала и до конца.

Давайте немного скорректируем этот план. Последовательное чтение книг серии не входило в мои намерения. Материал в них освоить не так-то просто, потому что JavaScript — язык мощный, замысловатый и порой достаточно сложный. Никому не удастся *загрузить* всю эту информацию в мозг за один проход, вы неизбежно забудете почти все прочитанное. Лучше даже не пытаться.

Мой совет: не торопитесь. Возьмите одну главу, прочитайте ее полностью от начала до конца, потом вернитесь и перечитайте раздел за разделом. Разберите код и идеи в каждом разделе. Если вы столкнетесь с чем-то сложным, лучше провести несколько дней за усвоением, повторным чтением и тренировками, а потом продолжить изучение.

На каждую главу можно выделить неделю или две, на каждую книгу — месяц или два, на всю серию — год и более, и даже в этом случае вы еще не выжмете из YDKJSY все возможное.

Не читайте эти книги взахлеб; будьте терпеливы. Чередуйте чтение с практикой: применяйте знания в рабочих задачах или собственных проектах. Оспаривайте мои идеи, возражайте, а самое главное — не соглашайтесь со мной! Организуйте учебную группу или клуб. Проводите мини-семинары в своем офисе. Пишите посты о прочитанном. Обсудите эти темы на локальных встречах JS.

Моя цель не навязать вам свое мнение. Скорее я хочу выработать у вас собственное мнение и умение его отстаивать. Вы не сможете достичь *этой цели* скоростным чтением. На это уйдет немало времени. Вы будете двигаться вперед шаг за шагом, пока изучаете, размышляете и возвращаетесь к прочитанному. Эти книги были задуманы как путеводители по JavaScript от вашего текущего местонахождения в знаниях о язы-

ке до точки более глубокого понимания. А теперь самая интересная часть: чем глубже вы понимаете JS, тем больше вопросов у вас появится и тем больше придется изучать!

Я очень рад, что вы отправляетесь в путешествие, и для меня большая честь, что вы сочли мои книги достойными своего внимания и решили довериться им. Пришло время начать *изучение* JS!

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.