

## ОТПРАВНАЯ ТОЧКА ДЛЯ БУДУЩЕГО ПРОГРАММИСТА

Полагаю, что всё это началось ещё в 2002 году. Я думал о преподавании программирования и о том, каким превосходным языком для обучения программированию стал бы Ruby. В том смысле, что мы все были в восхищении от Ruby, потому что это мощный, элегантный и действительно просто интересный язык. И мне подумалось, что было бы здорово начать знакомство с программированием именно с него.

К сожалению, в то время было немного документации по Ruby, предназначенной для новичков. Некоторые из нашего сообщества говорили о том, что очень пригодился бы учебник наподобие "Ruby для новичка", или в более широком смысле — для обучения программированию вообще. Чем больше я думал обо всём этом, тем больше накапливалось того, что мне нужно было высказать (и это меня несколько удивило). Наконец, кто-то сказал: "Крис, почему бы тебе тогда просто не написать учебник вместо того, чтобы всё время говорить о нём?" Так я и сделал.

Но получалось не слишком хорошо. У меня была масса идей, которые были хороши *в теории*, но в реальной задаче написать классный учебник для не-программистов таился гораздо больший вызов, чем я мог осознать. (Я имею в виду, что он казался хорошим для меня, но я-то уже знал, как программировать.)

Спасло меня вот что: я устроил так, что людям было действительно легко связаться со мной, и я всегда старался помочь людям, когда они на чём-нибудь застревали. Когда я видел, что многие застревают на одном и том же месте, я переписывал его. Работы было много, но учебник потихоньку становился всё лучше и лучше.

Спустя пару лет, он был уже весьма хорош. :-) В самом деле, настолько хорош, что я был готов сказать, что он закончен, и двигаться дальше к чему-то ещё. И как раз примерно тогда появилась возможность превратить этот учебник в книгу. Поскольку он был в основном почти готов, я посчитал, что трудностей не возникнет. Я только подчищу в нём некоторые места, добавлю несколько упражнений, ну, возможно, ещё несколько упражнений, и ещё несколько глав, и дам его на просмотр ещё 50-ти рецензентам...

Это заняло у меня ещё один год, но теперь я думаю, что он действительно *в самом деле* хорош, в основном благодаря сотням храбрецов, что помогли мне написать его.

То, что размещено на этом [сайте](#), — это исходный вариант учебника, почти совсем не изменённый с 2004 года. Чтобы получить самую свежую и улучшенную версию, вам, возможно, захочется обратиться вот к [этой книге](#).

## СООБРАЖЕНИЯ ДЛЯ ПРЕПОДАВАТЕЛЕЙ

Было несколько руководящих принципов, которых я старался придерживаться. Я думаю, они делают процесс обучения гораздо более лёгким — ведь учиться программировать и так довольно тяжело. Если вы преподаёте или наставляете кого-то на путь благородного хакерства, то эти идеи могут помочь и вам.

Во-первых, я старался как можно больше разделять понятия так, чтобы учащемуся требовалось изучать каждый раз только одно понятие. Это было трудно сначала, но уж *слишком* легко после того, как я приобрёл некоторый опыт. Некоторые вещи должны изучаться прежде других, но я был поражён, насколько мало в действительности имеется таких иерархических зависимостей. В конечном счёте, я просто выбирал порядок изложения и старался скомпоновать материал таким образом, чтобы каждый новый раздел основывался на предыдущих.

Другой принцип, о котором я всё время помнил, — учить только одному способу делать что-либо. В этом очевидное преимущество учебника для тех, кто никогда не программировал ранее. С одной стороны, один способ делать что-то легче выучить, чем два. Хотя, возможно, более важная выгода от этого состоит в том, что, чем меньшему числу приёмов вы учите начинающего программиста, тем более изобретательным и сообразительным ему придётся быть в своих программах. Поскольку основная деятельность в программировании связана с решением задач, критически важным становится поощрять это насколько возможно на каждом этапе разработки.

Я постарался переводить понятия программирования на те понятия, которые уже есть у начинающего программиста, с целью представить идеи таким образом, чтобы нагрузка ложилась более на его интуицию, нежели на учебник. Объектно-ориентированное программирование подходит для этого весьма хорошо. Мне можно было достаточно рано начать ссылаться в учебнике на "объекты" и различные "типы объектов", невинно роняя такого рода фразы в подходящие моменты. Я не говорил что-либо подобное "всё в Ruby является объектами," или "числа и строки — это разновидности объектов", поскольку эти утверждения в действительности ничего не значат для начинающего программиста. Вместо этого я предпочёл говорить о строках (а не о "строковых объектах"), но иногда я упоминал "объекты", имея в виду просто "вот эти вещи в этих программах". А то, что все эти *вещи* в Ruby *фактически являются* объектами, позволило этим уловкам с моей стороны хорошо сработать.

Хотя я желал избежать ненужного объектно-ориентированного жаргона, я хотел быть уверенным, что если им в самом деле нужно было узнать какой-то термин, они выучат правильное слово. (Я не хотел, чтобы им пришлось учить его дважды, верно?) Вот почему я применял слово "строки", а не "текст". Методы тоже нужно было как-нибудь назвать, и я называл их "методы".

Что касается упражнений, думаю, что я приготовил несколько удачных, но их никогда не бывает слишком много. Если честно, могу поспорить, что я половину времени провёл, просто пытаюсь подобрать забавные и интересные упражнения. Скучные упражнения напрочь убивают всякое желание программировать, в то время как от безупречно подходящих упражнений появляется профессиональный зуд, перед которым начинающий программист вряд ли сможет устоять. Короче говоря, невозможно потратить слишком много времени на подготовку хороших упражнений.

## ОБ ОРИГИНАЛЬНОМ УЧЕБНИКЕ

Страницы этого учебника (и даже эта самая страница) сгенерированы [большой программой](#) — конечно, написанной на Ruby. :-) И поэтому, в оригинальном учебнике есть несколько тонкостей. Например, все примеры программного кода в

действительности исполняются каждый раз, когда вы просматриваете страницу, и показанный результат — это результат, сгенерированный при выполнении. Думаю, это лучший, самый лёгкий и, конечно, самый крутой способ удостовериться, что весь представленный мной код отработывает *в точности* так, как я утверждаю. Вам не нужно беспокоиться о том, что я мог ошибочно скопировать выдачу какого-нибудь из примеров или забыл протестировать какую-то часть кода: он весь проверяется каждый раз, когда вы его просматриваете. Так, в разделе по генераторам случайных чисел, если вы перезагрузите страницу, то увидите, что числа каждый раз будут изменяться... *Прекрасно.* (Я применил похожий трюк для примеров кода, когда писал книгу, но очевидно, это наиболее заметно именно в этом учебнике.)



## Благодарности

И наконец, мне бы хотелось поблагодарить всех в списке рассылки [ruby-talk](#) за их мысли и ободрение; всех моих замечательных рецензентов за их помощь сделать эту книгу гораздо лучше, чем я сделал бы это один; в особенности — мою дорогую жену за то, что она была моим главным рецензентом / тестером / подопытным кроликом / музой; Мацумото — за создание этого сказочного языка программирования; издательство Pragmatic Programmers — за то, что сообщили мне об этом и, конечно, за публикацию моей книги!

Если вы заметили какие-то ошибки или опечатки, или у вас есть замечания, предложения или хорошие упражнения, которые можно добавить, пожалуйста, [сообщите мне](#).

[А если у вас появятся замечания или уточнения по поводу перевода на русский язык, можете написать [переводчику](#). — *Прим. перев.*]