

Как стать разработчиком с нуля



Марина Александровна

26 янв 2023

Рассказываем, что нужно знать айтишнику и как стать программистом с нуля ✓ Отучиться на программиста самостоятельно и войти в IT реально!

Как стать программистом и с чего начать, если нет практического опыта, а представление о программировании весьма размытое? Подробно рассказываем, как стать востребованным программистом с нуля и на что обращать внимание в ходе обучения.

1. Кто такой программист
2. Почему вы хотите начать программировать
3. Что нужно знать, чтобы стать программистом
4. Какое направление в программировании выбрать в 2023 году
5. Десктопная разработка



Tproger

всё о программировании



8. План развития iOS разработчика
9. Как стать разработчиком игр

Мы тоже используем куки, потому что без них вообще ничего не работает



Хорошо

13. Как стать фронтенд разработчиком
14. Full Stack разработка
15. Какой язык программирования выбрать в 2023 году
16. Где выучиться на программиста
17. Классификация разработчиков
18. Как стать Junior-разработчиком
19. Где работают программисты
20. Как найти первую работу в IT
21. Что спрашивают у Джунов на собеседовании
22. Что делать, если вам не хватает знаний до вакансии Джуниор-программиста
23. Заключение

1. Кто такой программист

Программист — это специалист по разработке компьютерных программ. Под разработкой понимается продумывание архитектуры программы, её реализация, оптимизация и тестирование. Далее мы разберёмся, как стать настоящим айтишником и что для этого нужно знать.

1.1 Чем занимается программист

Это зависит от специфики выбранной отрасли, но в целом всё сводится к написанию и тестированию кода. Важно понимать свои обязанности и не скатываться в звание **тыжпрограммиста**, которому поручают всё — от установки Windows до починки техники.

1.2 Можно ли стать айтишником с нуля

Мы тоже используем куки, потому что **без них вообще ничего не работает**

задачи, видеоуроки и туториалы.

Ответьте на вопросы и получите 5 курсов
для старта в IT **бесплатно**



Перейти

Реклама ООО «Гикбрейнс» LjN8K6Km4

А вот как стать крутым программистом с нуля и с чего начать — другой вопрос. К примеру, Senior разработчика от джуна отличают годы практики и колоссальный опыт, так что только терпение и труд.

2. Почему вы хотите начать программировать

IT — перспективная отрасль, которая сулит хорошим специалистам немаленькие зарплаты. Но если деньги — единственная причина, подумайте трижды. Вам предстоит работать в офисе, дома, забывать о выходных и возможности расслабиться: все мысли в тасках и багах. А всё, что обязательно нужно знать начинающему программисту, курсы не дадут: будьте готовы доучиваться.

Если же вы питаете неподдельный интерес к процессу, смело ступайте на стезю разработки.

3. Что нужно знать, чтобы стать программистом

Чтобы стать успешным и востребованным программистом, нужно обзавестись рядом hard skills и soft skills. Первые отвечают за технические умения и знания, вторые — за качества, которые помогают эффективно работать и находить общий язык с командой.

3.1 Hard skills

У каждой профессии свой набор хард скиллов. У программиста это общие технические знания (математика, информатика, алгоритмизация) и набор навыков, свойственный конкретному стеку.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

- какие типы данных бывают и чем они отличаются друг от друга;
- понимание систем счисления — двоичной и шестнадцатеричной;
- **булева алгебра**;
- понимание принципов выполнения операций ЭВМ.

3.1.2 Математика

А как же стать настоящим программистом, если плохо с математикой, и **насколько важна математика в разработке**? Требования разнятся в зависимости от сферы: для работы с графикой нужно знать геометрию, а для машинного обучения — теорию вероятностей и статистику. Но базовые знания математики никогда не будут лишними, особенно если вы осваиваете программирование с нуля.

3.1.3 Алгоритмы

Это то, что нужно обязательно знать, чтобы стать программистом. Алгоритмами называют последовательность действий, которая преобразовывает входные данные в некий результат. Наверняка вы слышали об алгоритмах поиска и сортировки — это они и есть. Разобравшись, как устроены алгоритмы, вы сможете создавать свои решения сложных задач и даже комбинировать их.

3.1.4 Принципы ООП

Объекто-ориентированное программирование — это подход в разработке, при котором за основу берутся абстрактные объекты, а не функции или процедуры. Данный подход используется в большинстве современных языков программирования, поэтому крайне важно понимать основные принципы ООП:

- абстракция;
- наследование;
- инкапсуляция;
- полиморфизм

Мы тоже используем куки, потому что **без них вообще ничего не работает**

Софт скиллы, они же гибкие или мягкие навыки, — это совокупность нетехнических неспециализированных навыков, которые тем не менее важны для производительности и профессионального развития. Это то, что также нужно знать программисту для эффективной работы.

3.2.1 Знание английского

Необходимость английского языка в программировании зависит от ваших конечных целей. Если вы видите программирование своим хобби, то можно обойтись базовыми знаниями английского и помощью онлайн-переводчика. Но если вы серьёзно настроены на карьеру разработчика, английский язык необходим.

Держите [бесплатные материалы по английскому для программистов](#).

3.2.2 Умение работать в команде

Программирование — это почти всегда командная работа. Как минимум, чтобы стать программистом, нужно уметь взаимодействовать с РМ, тимлидом и просто камрадом по проекту.

Речь идёт не о социопатии и необходимости быть душой компании. Чтобы стать разработчиком, нужно слушать и слышать, задавать вопросы, [признавать ошибки](#) и исправлять их плечом к плечу, а не сваливать вину на сослуживцев при первой же возможности.

Что делать, если вам не заплатили — советы от фрилансеров для фрилансеров

tproger.ru

Для эффективной командной работы следует понимать и использовать гибкие методологии разработки. Почитайте об [Agile](#), [Scrum](#) и [Kanban](#) — пригодится.

3.2.3 Стрессоустойчивость

Ответственность программиста сравнима с бухгалтерской. Так,

сравнивший не предпринимая код с ошибкой может уволить сотрудника

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Кроме того, вам нужно спокойно принимать пожелания и вносить правки, если требуется, не срываясь на проджект-менеджера и тем более заказчика. Хорошим подспорьем станут книги «[Как разговаривать с кем угодно, когда угодно и где угодно](#)» и «[HBR Guide. Эмоциональный интеллект](#)».

3.2.4 Постоянное саморазвитие

Не думайте, что программирование можно раз освоить и забыть. Отрасль богата переменами, нововведениями — уходом старого и появлением нового. Это и языки, и фреймворки, и сопутствующие технологии, которые всё время меняются и совершенствуются для более эффективной разработки.

Настраивайтесь на постоянные митапы, вебинары и курсы повышения квалификации. А ещё не забывайте о сертификатах, которые нужно получать и подтверждать с разной периодичностью.

3.2.5 Тайм-менеджмент

Контроль времени на решение поставленных задач существенно экономит и само время, и ресурсы, а также увеличивает продуктивность. Тайм-менеджмент для разработчика базируется на тех же концепциях, что и для любой другой профессии.

Что должен знать уважаемый фронтендер в 2024 году

 tproger.ru

При планировании рабочего дня учитывайте дедлайны, используйте тайм-трекер и таблицу Эйзенхауэра:

4. Какое направление в программировании выбрать в 2023 году

Прежде чем думать, как стать айтишником с нуля самостоятельно,

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

В опросе приняло участие 65 000 разработчиков из разных стран.

Но что насчёт России? Согласно данным «Университета 20.35» за 2020 год, в топ-10 востребованных удалённых IT-профессий России вошли:

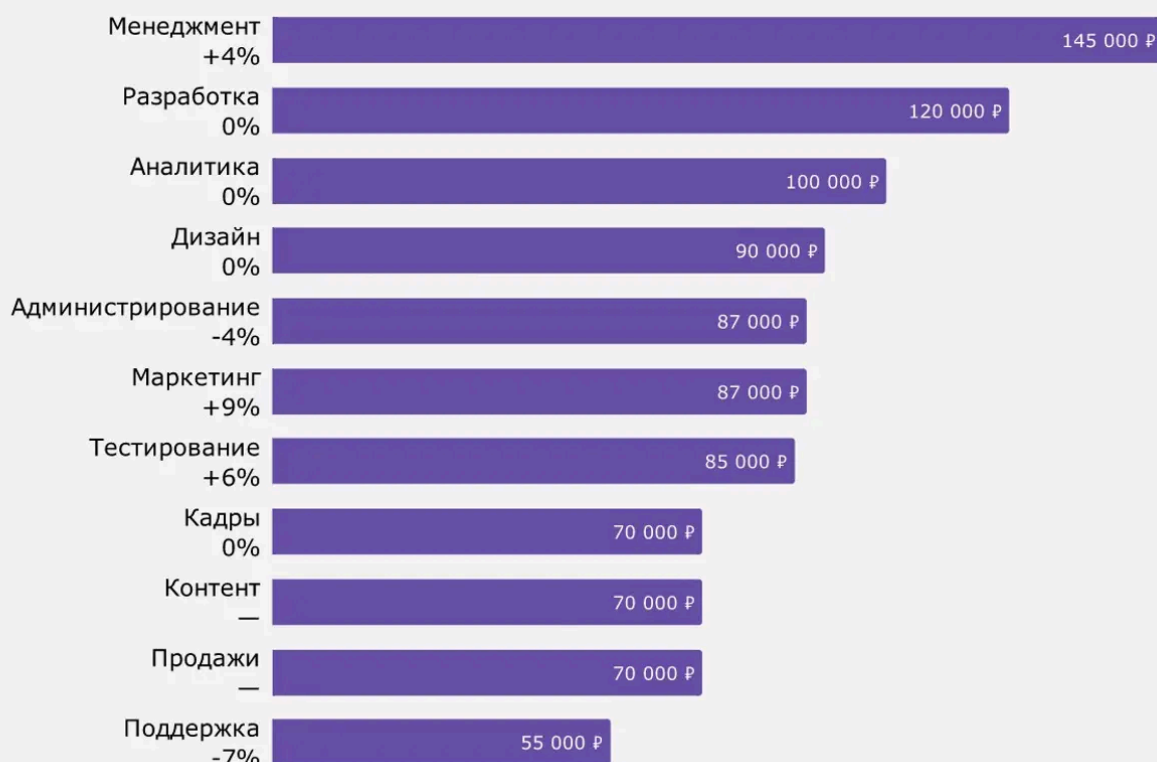
1. Java-разработчик
2. Аналитик данных
3. Разработчик на языке C#
4. PHP-разработчик
5. Специалист службы поддержки
6. Продуктовый аналитик
7. Разработчик 1С
8. Kotlin-разработчик
9. Системный администратор
10. Системный аналитик

А вот зарплаты айтишников по специализациям за второе полугодие 2020:

Мы тоже используем куки, потому что **без них вообще ничего не работает**

Зарплаты айтишников по специализациям

% — разница с предыдущим периодом



Проценты на диаграмме отражают разницу с предыдущим периодом, а прочерк означает, что данных для сравнения недостаточно.

5. Десктопная разработка

Десктопными называют приложения, которые устанавливаются на компьютер и работают под управлением ОС. Как известно, чтобы стать айтишником с нуля, сперва в любом случае нужно практиковаться на консоли и десктопе.

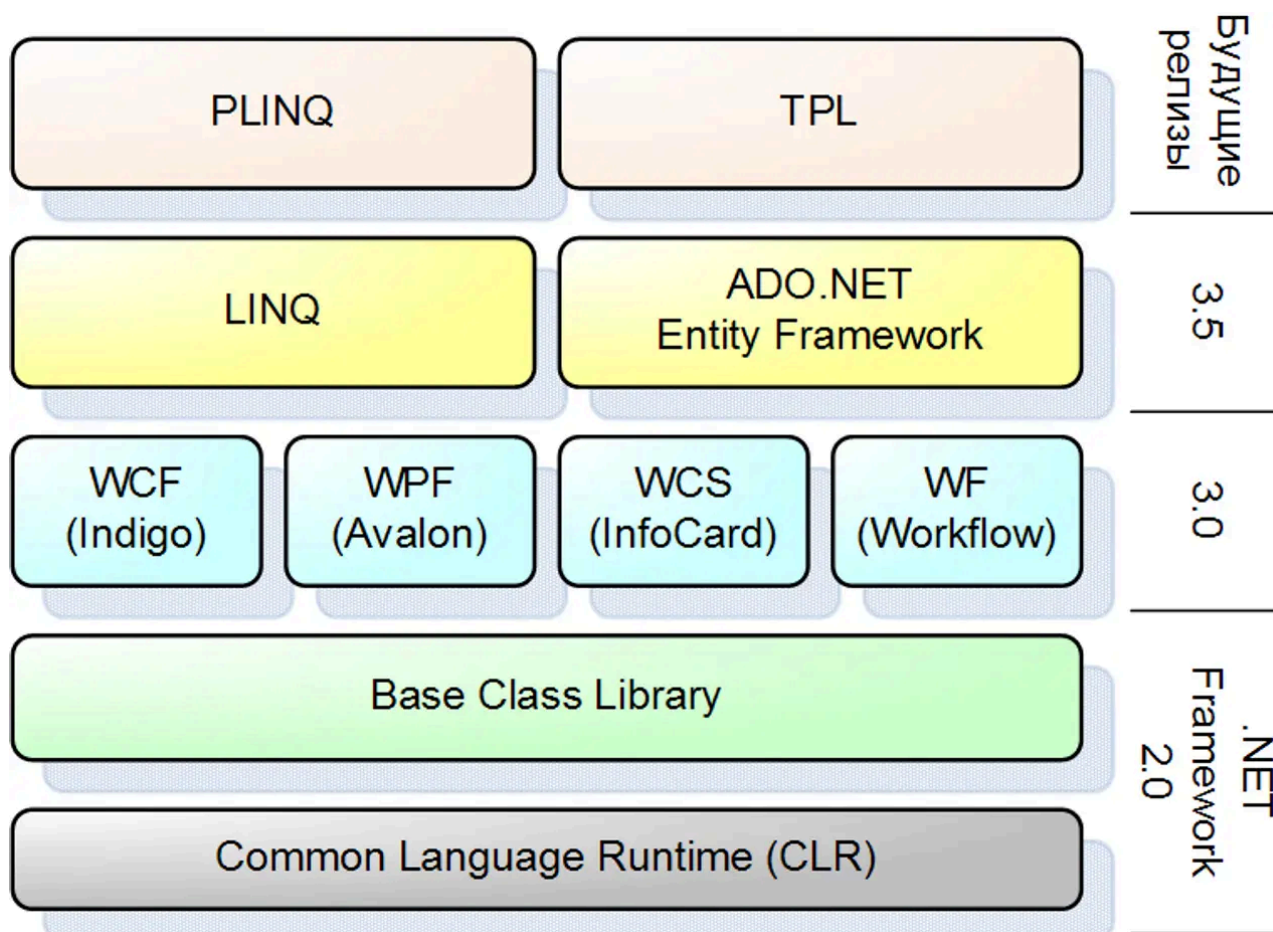
Хотите заниматься этим на серьёзном уровне? Здесь всё просто: если планируете разрабатывать под Windows — учите язык C#, под Linux — C++ или C, под macOS — Swift.

5.1 Программы для Windows

Мы тоже используем куки, потому что **без них вообще ничего не работает**

5.1.1 C# / C++

Для серьёзного высокопроизводительного приложения подойдут «плюсы». Но если жёстких требований к производительности нет, лучше обратиться к GUI-платформе Windows Presentation Foundation в составе .NET Framework. Писать на C# проще, и развивается это язык динамичнее того же C++.



Автор таблицы — [NZeemin](#)

Курс по Go-разработке от Kata Academy

Постоянный доступ, бесплатно, онлайн

tproger.ru

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Чуть больше вы сможете узнать из нашего [гайда по программированию на C++ с нуля](#).

5.2 Для Linux

5.2.1 C

Linux — это семейство Unix-подобных операционных систем, и так сложилось, что ядро Linux написано на языке C. Да, Си считается устаревшим, поэтому многие программисты предпочитают ему Python, Java, Perl, иногда Ruby. Но драйверы, модули ядра и большая часть системного ПО всё же написаны на языке программирования Си, и для Linux-программ он приоритетен.

Если вас интересует, как стать разработчиком на языке C, загляните в [материалы по изучению Си](#).

5.2.2 C++

По сути, это усовершенствованный C. Браузеры, графические окружения, утилиты GNU и прочее программное обеспечение под Linux написаны на C++. А ещё на «плюсах» написан Qt — графический фреймворк, который используется во многих кроссплатформенных программах и является бесплатным для разработки свободного ПО.

5.3 Для macOS

5.3.1 Swift

Swift — родной язык для Apple-приложений, так что он оптимален для разработки под macOS. Выбрать можно и Objective-C, но последний проигрывает в скорости работы.

Достоинства Swift:

- прост в изучении;
- легко читается:

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

А вот ещё [для чего хорош Swift](#).

6. Data Science

Языки программирования Python и R — общепринятый стандарт для Data Science, так как они оба отлично подходят для работы со статистикой.

6.1 Python

Как стать Python программистом с нуля самостоятельно? На самом деле, не так уж сложно: у языка Пайтон простой синтаксис, много библиотек, развёрнутая и понятная документация. Но есть и минусы, такие как отсутствие удобных средств ORM и не самая высокая скорость работы.

Для Data Scientist у Python есть немало библиотек и фреймворков:

- [Pandas](#)
- [PyTorch](#)
- [NumPy](#)
- [TensorFlow](#)
- [Keras](#)
- [scikit-learn](#)
- [Matplotlib](#)

6.2 R

R в свою очередь характеризуется богатой ML-экосистемой, поддержкой векторных вычислений и синтаксисом, заточенным под нужды статиста. Из минусов — меньшее комьюнити и не самый

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

1. Esquisse и Ggplot2 — библиотеки для создания графиков.
2. Dplyr — аналог Pandas для R.
3. Shiny — пакет для интерактивных веб-приложений с визуализацией исследований.

И это не говоря о randomForest, Caret и других библиотеках с методами машинного обучения.

7. Как стать разработчиком на Андроид

В 2019 году компания Google [объявила Kotlin предпочтительным языком программирования для создания Android-приложений](#). Это спровоцировало новую волну споров по выбору языка.

Рост ежемесячной зарплаты от 50 000 рублей до 200 000 рублей за год.
Карьера Java-разработчика с нуля

Постоянный доступ, бесплатно, онлайн

 tproger.ru

Ответ таков: чтобы стать Android-разработчиком, нужны оба языка, но при этом ознакомиться с их отличительными характеристиками не помешает.

7.1 Java

- много библиотек и готовых решений;
- большое комьюнити;
- разные сферы применения, если вдруг надоеет Android-разработка;
- документация Kotlin предполагает, что вы знаете Java.

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

- лаконичный синтаксис;
- актуальная IDE, тесно связанная с языком разработки (Android Studio и Kotlin созданы JetBrains);
- предсказуемое развитие;
- перспективы.

8. План развития iOS разработчика

А как же тогда стать разработчиком iOS? Здесь всё гораздо проще, так как основной язык всего один.

8.1 Swift

Этот язык программирования разработан Apple и многое заимствовал из Objective-C, а также C++.

Преимущества Swift:

- компактный язык программирования;
- производительность приближена к C++;
- тестирование кода в реальном времени со Swift Playgrounds;
- постепенно распространяется на другие платформы.

9. Как стать разработчиком игр

Что нужно знать программисту, чтобы войти в геймдев? Сперва необходимо определиться, как устроены игры. Хороший вариант для начинающих — готовые движки, такие как Unity и Unreal Engine (но последний сложнее и серьезнее).

Мы тоже используем куки, потому что **без них вообще ничего не работает**

9.1 Игры на iOS или macOS

Поговорим о программировании игр на Swift. Для начала необходимо установить «яблочную» IDE Xcode: в ней поддерживается разработка как на iOS, так и на macOS. Среда бесплатная и функциональная. В ней создайте новый проект с шаблоном Single View Application или Game. Тут уж к чему душа лежит:

1. Single View Application предназначен для приложения с одним экраном. Шаблон включает в себя компонент View Controller для настройки вида с конструктором Interface Builder.
2. Это специальный игровой шаблон. В нём предусмотрены фреймворки для двумерной визуализации, 3D-графики без OpenGL и низкоуровневый API, с которым можно писать действительно высокопроизводительные игры.

Дальше следуйте [инструкции по написанию простой игры с игральными костями](#).

9.2 Браузерные игры

Вообще, создать браузерную игру можно и на HTML/CSS/JavaScript, но мы затронем PHP.

9.2.1 PHP

PHP актуален для разработки браузерных игр, хоть и не заточен под них специально. Такие компании, как Funzio, Gameforge, Goodgame Studios и другие продолжают использовать этот язык в своих играх.

Есть исчерпывающий видеокурс по созданию браузерной игры на PHP — от настройки окружения до регистрации и авторизации:

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)



9.2.2 JavaScript

После ухода тяжеловесной технологии Flash из браузеров у JavaScript-разработчиков появилось много возможностей. Сейчас есть сильная альтернатива — элемент Canvas в HTML5 для создания двухмерных изображений через JavaScript-команды.

Чтобы понимать принцип создания игр на JS, проанализируйте код рандомайзера с угадыванием чисел.

Как только освоите азы, попробуйте написать что-то сложнее. В этом видеоуроке как раз используются чистый JavaScript и HTML5 Canvas:

Мы тоже используем куки, потому что **без них вообще ничего не работает**



9.3 Игры на Unity

Это популярная платформа для разработки игр. Очень дружелюбна к новичкам: можно [бесплатно загрузить](#) и обучаться с помощью курсов, словарей и других материалов, которые предоставляет компания и участники сообщества. Платформа поддерживает создание 2D- и 3D-игр.

Игры на Unity пишут с использованием языков C# и JavaScript. Почитайте, [как создать простую 2D-игру на Unity](#).

9.4 Современные большие и требовательные игры

Речь о тяжеловесных многопользовательских играх с сильной нагрузкой на сервер. Такая задача под силу языкам C и C++.

9.4.1 C++

Что нужно знать начинающему программисту о C++? Это самый популярный язык программирования для геймдева: быстрый, мощный, с большим набором полезных инструментов. C++ —

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Не даром ведущие разработчики многопользовательских игр, такие как Blizzard Entertainment, работают с C++. На нём написаны World of Warcraft, Diablo, StarCraft, Counter-Strike и другие популярные игры. В основе Xbox и PlayStation также лежит C++.

9.4.2 C

Да, язык не такой популярный, как «плюсы», но по-прежнему хорошо справляется с большими нагрузками — альтернатив на низком уровне нет. Когда дело доходит до компиляции, сложно представить, какой бы язык мог конкурировать с Си.

10. Как стать Python разработчиком

Чтобы стать питонистом, нужно начинать с азов, таких как принципы ООП и синтаксис языка. А понимание того, как стать программистом Python, придёт с определением сферы деятельности. Основных направления для Пайтон-разработчиков два:

1. Data Science
2. Веб-разработка

Поскольку первое направление мы уже разобрали, сосредоточимся на веб-разработке. Что же что нужно знать Python веб-разработчику? Обратите внимание на фреймворки Django и Flask. Для эффективной работы с бэкендом достаточно одного из них, но в идеале следует понимать оба.

Python можно использовать для создания десктопных приложений, геймдева и мобильной разработки, но наилучших показателей с этим языком можно достичь в перечисленных направлениях.

Разработка на Python с нуля: роадмап программиста

 tproger.ru

Мы тоже используем куки, потому что **без них вообще ничего не работает**

А как можно стать Java разработчиком, если никогда не был программистом? Базис для всех языков плюс-минус одинаков: синтаксис, переменные, условия, циклы, ООП. Для решения специфических задач нужно освоить инструмент сборки и управления проектом Maven. Будет полезен и набор open source сервисов [Netflix Stack](#).

Дальше всё зависит от выбранного направления в Java:

- веб-разработка;
- Android разработка;
- программирование для десктопов;
- Big Data.

Все материалы по этим направлениям, инструменты и путь развития вы найдёте в [дорожной карте Java-программиста](#).

12. Бэкенд разработка — с чего начать?

Как стать бэкенд разработчиком? Нужно понимать, как устроены операционные системы, уметь работать с терминалом, знать стандарты POSIX и концепции построения сети, управлять вводом и выводом (I/O), процессами, а также памятью.

Как освоить бэкенд-разработку в 2022 году: дорожная карта

[🔗](#) tproger.ru

12.1 Обучение PHP

Как вы могли догадаться, стать бэкенд-разработчиком без знания языка невозможно.

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

YouTube, Wikipedia и даже движок WordPress, на котором вы сейчас читаете эту статью.

Другие преимущества PHP:

- простая установка и настройка;
- кроссплатформенность;
- выполнение программы до достижения ошибки в коде;
- большое комьюнити.

Мы подготовили [гайд для тех, кто хочет научиться программировать на PHP](#).

12.2 Обучение Node.js

Как известно, стать бэкенд веб-разработчиком на JavaScript можно с программной платформой Node.js. Она позволяет запускать JavaScript-код без браузера на бекенде, а в пакетной экосистеме npm есть много готовых решений.

Для практики можете [создать простенький блог на Node](#).

12.3 Обучение Java

Мы уже разобрались, как можно стать Java программистом с нуля — поговорим прицельно о роли языка в бэкенде.

Для новичка Java сложнее, чем условный Python или PHP, да и код выйдет более громоздким. Однако это мощный инструмент в создании больших веб-проектов вроде eBay или LinkedIn, которые как раз и написаны на Java.

12.4 Обучение Django

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

фреймворк, который предоставляет достаточно готовых решений для серверной части.

В освоении поможет [туториал с проектами на Django для начинающих](#).

12.5 Основы SQL

Это язык запросов для работы с базами данных. Вы вряд ли вы найдёте по нему гайд в разделах «Как стать программистом», ведь SQL нужно не учить, а понимать.

Держите материалы, которые позволят разобраться в основах SQL:

- [Основные команды SQL, которые должен знать каждый программист](#)
- [27 распространённых вопросов по SQL с собеседований и ответы на них](#)

13. Как стать фронтенд разработчиком

Итак, frontend программирование — с чего начать? Вам нужно освоить HTML, CSS и JavaScript — три кита, на которых стоит фронтенд-разработка.

13.1 Обучение HTML

Разметка — это каркас сайта. Чтобы работать с фронтендом, следует хорошо понимать анатомию HTML-разметки, знать основные атрибуты и теги, обеспечивать доступность и освоить SEO хотя бы на базовом уровне.

Для лучшего понимания рекомендуем ознакомиться с [видеокурсом](#)

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Далее идут каскадные таблицы стилей. В разрезе CSS вам нужно освоить:

- компоновку;
- центрирование и выравнивание;
- видимость элементов.

Это базис. Следующим этапом можно переходить к медиа-запросам, которые позволят учитывать технические параметры разных устройств. В создании адаптивного отзывчивого дизайна помогут CSS Grid и Flexbox.

13.3 Обучение JavaScript

Ну и как стать эффективным фронтенд разработчиком без знания JavaScript? Именно он отвечает за функциональность сайта и возможность с ним взаимодействовать: заполнение и отправка форм, нажатие кнопок, прослушивание событий и многое другое.

Не спешите осваивать фреймворки — сперва чистый JavaScript. Это язык со слабой типизацией, а значит операция вроде $2 + "2"$ вместо ошибки выдаст результат 22. Так что есть с чем разобраться и без фреймворков.

Больше полезных материалов и инструментов для фронтенд-разработчика вы найдёте в [дорожной карте по фронтенду](#).

14. Full Stack разработка

Эдакий гибрид frontend- и backend-разработки. Данное направление набирает популярность. Сейчас сложно найти чистых фронтендеров или бэкендеров: для взаимодействия этим специалистам нужно знать, за что отвечают их коллеги по веб

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

разработчиком, и чем вообще занимаются эти специалисты?

Хороший пример — CRM-разработчик. В его обязанности входят:

1. Работа с базами данных клиента: создание событий, автоматизация процессов и интеграция приложений посредством API.
2. Настройка аналитики в формате репортов/дашбордов.
3. Создание простенького, но понятного интерфейса для работы с готовым продуктом.

Здесь также применимы инструменты вроде Bootstrap или Angular, но без фанатизма — важно создание рабочего функционального продукта.

Как стать full stack разработчиком

tproger.ru

15. Какой язык программирования выбрать в 2023 году

Всё ещё не определились с языком? Для начала взгляните на [рейтинг TIOBE](#):

А теперь давайте пробежимся по самым популярным языкам программирования, затронем их сильные стороны и уточним сферы применения.

15.1 Python

Как уже говорилось ранее, стать Python программистом с нуля самостоятельно несложно, ведь у языка низкий порог вхождения и понятный синтаксис. Пайтон рекомендуется к изучению в вузах в качестве базового языка.

Основные сферы применения Python:

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

- Веб-разработка

15.2 JavaScript

О сферах применения JS говорить не стоит: язык давно стал лидером в веб-разработке, сперва захватив фронтенд, а теперь и отжимая бэкенд за счёт Node.js. Если хотите разрабатывать веб-приложения, учите JavaScript.

15.3 Java

Java — это кроссплатформенный язык программирования, мощный обработчик и внушительная библиотека. У Java много готовых решений и исчерпывающая документация.

Основные сферы применения Java:

- Веб-разработка
- Android разработка
- Desktopная разработка
- Big Data

15.4 C++

Быстрый, универсальный и полезный в качестве фундамента язык. Да, пусть Python в разы проще, но за сложностью C++ скрываются и преимущества. Освоив «плюсы», будет легче понять Java, C# и JavaScript, в основу которых легли принципы C++.

Главные сферы применения C++:

- Геймдев

Мы тоже используем куки, потому что **без них вообще ничего не работает**

- Ускорение существующих приложений

15.5 C#

C# часто сравнивают с Java, и у них действительно много общего. Код на C# удобочитаемый, а синтаксический сахар обеспечивает высокую скорость разработки.

Основные сферы применения C#:

- Геймдев
- AR/VR
- Десктопная разработка

По [этому видеокурсу](#) вы узнаете, как стать C# разработчиком.

15.6 PHP

Это бессмертная классика серверной части веб-приложений. PHP бюджетен, прост в освоении и успел обрести внушительным комьюнити. Пусть современные сайты и пишутся то на Java, то на Node.js, существует огромная база написанных на PHP веб-приложений, которые всегда будут нуждаться в поддержке.

15.6 Kotlin

Здесь только Андроид. Но, как говорилось ранее, чтобы стать Android разработчиком с нуля, стоит начать с изучения Java, а только после переходить на Kotlin.

15.7 Swift

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

компанией Apple.

15.8 На что обратить внимание при выборе

На отрасль, в которой желаете развиваться. Но чтобы понять, как стать программистом и с какими трудностями придётся столкнуться, рекомендуем начать с С или С++, после которых любой другой язык освоить куда проще.

Если же вы точно определились, с чем именно хотите работать, смело выбирайте язык, который для этого подходит. Опирайтесь на рейтинги популярности, просматривайте вакансии по желаемому направлению в вашем городе, сравнивайте уровень востребованности и зарплаты.

16. Где выучиться на программиста

Чтобы ответить на вопрос, где учиться на программиста, необходимо рассмотреть все доступные варианты.

16.1 Самостоятельная учеба

Обилие книг, видеоуроков и сервисов вроде Scratch позволяет освоить базис, не выходя из дома. Но как выстроить план обучения, чтобы стать программистом с нуля самостоятельно? Обратитесь к стандартным программам обучения или роадмапам. Чаще всего основы включают в себя:

1. Знакомство с программированием.
2. Разнообразие IT-направлений.
3. Основы выбранного языка: синтаксис, объекты и классы, функции, списки и словари, условия, циклы.
4. Hello World.
5. Модульное тестирование.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

16.2 ВУЗ

Это более основательный подход, и, вопреки распространённому заблуждению, «корочка» IT-специалиста действительно ценится, особенно в крупных компаниях. Если вы серьёзно настроены на карьеру в IT и думаете, как отучиться на программиста, выбирайте вуз.

16.3 Практические курсы

К курсам можно относиться по-разному, но если у вас нет профильного образования и вы не хотите тратить несколько лет в вузе, курсы — хороший вариант. К тому же, современные курсы покрывают практически все нужды, обеспечивая обучение основам программирования, более продвинутым темам, помогая с составлением резюме и трудоустройством.

16.4 С помощью ментора

Мы уже разобрались, как можно стать айтишником с нуля дома, но не затронули менторство. Всё потому, что оно не всегда подразумевает обучение на дому: это может быть как онлайн-репетитор, так и индивидуальные занятия оффлайн.

Ментор не только объяснит материал, но также обратит ваше внимание на ошибки, которые вы допускаете на практике. Внимание наставника не растекается по группе людей, а сосредоточено именно на вашей работе, что даёт более прицельное обучение программированию.

17. Классификация разработчиков

Мы тоже используем куки, потому что **без них вообще ничего не работает**

17.1 Стажёр

Стажёр — это вчерашний студент, который набивает шишки на практике. Обычно это низкооплачиваемые сотрудники, которых компания нанимает специально, чтобы взрастить нужного специалиста.

Стажировка может быть как оплачиваемой, так и неоплачиваемой: всё зависит от договорённости с нанимателем. Продолжительность стажировки оговаривается индивидуально. Чаще к стажёру приставляют ментора от компании.

17.2 Junior разработчик

Как правило, чтобы стать Джуниор программистом, достаточно освоить синтаксис языка, простые задачи решать самостоятельно, а сложные — заручившись помощью более опытного коллеги. Младший разработчик постоянно учится, совершенствует свои навыки. Его код нуждается в постоянном ревью.

17.3 Middle разработчик

У такого программиста уже есть опыт в разработке, он самостоятельно выполняет сложные задачи, но по-прежнему нуждается в ком-то, кто направит. Чаще стек технологий Мидла ограничен необходимым базисом. Нередко он знает, как выполнить поставленный таск, но не до конца понимает, как его решение интегрируется в готовый проект.

17.4 Senior разработчик

Сеньор, или старший разработчик, уже обладает широким кругозором, знает несколько языков программирования и технологий.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

навыки, но при этом способны генерировать собственные идеи и делиться ими с командой.

Переход между перечисленными уровнями требует не столько времени, сколько усердия со стороны самого разработчика.

[Подробнее об отличиях Junior, Middle и Senior разработчиков.](#)

18. Как стать Junior-разработчиком

Здесь опять всё упирается в выбранную отрасль: у каждой специальности свои требования. Давайте разберёмся, что нужно знать младшему программисту на примере веба и мобильной разработки.

17 вопросов джуну: что должен знать Junior-разработчик

[tproger.ru](#)

18.1 Frontend Junior

Сразу оговоримся, что алгоритмы, Git, тестирование и прочий базис мы затрагивать не будем, так как это актуально для всех направлений.

Итак, что же нужно знать младшему Frontend разработчику в 2023 году?

1. HTML5, понимать разметку, освоить базовые теги, атрибуты и вложенные элементы.
2. CSS3 — позиционирование, компоновка и выравнивание элементов, их видимость, отзывчивый дизайн и подходы mobile-first.
3. Если мы говорим о JavaScript, то сперва следует выучить основы языка, а лишь потом двигаться к фреймворкам.
4. Понимание MongoDB.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

К младшему бэкенд программисту иные требования. Он должен знать:

1. Принцип работы интернета и операционной системы.
2. Язык серверного программирования: PHP, Java, JavaScript (Node.js), Python (Django) или другой.
3. Базы данных — реляционные и NoSQL.
4. API — REST, JSON.

Плюсом будет умение установить и настроить хотя бы один из популярных веб-серверов: Nginx, Apache, Caddy или MS IIS.

18.3 Junior iOS

Чтобы влиться в iOS-разработку, нужно:

1. Знать язык программирования Swift.
2. Иметь практический опыт работы с IDE Xcode.
3. Понимать сущность **MVC**, а также то, как «яблочные» приложения строятся на основе этого паттерна.
4. Уметь запускать приложения на устройствах и распространять их.

18.4 Junior Android

Как уже говорилось, начать следует с выбора языка, но за основу лучше взять именно Java, а при необходимости дополнительно изучить Kotlin. Далее:

1. Практический опыт работы с Android Studio.
2. Понимание Android Architecture: паттерны MVVM, MVI, MVP.
3. Изучение жизненного цикла Activity.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

19. Где работают программисты

Сложно ответить на вопрос, где в целом работают программисты, потому что сегодня профессия требуется чуть ли ни в любой сфере.

19.1 Фриланс

Как стать айтишником на фрилансе? Для этого есть немало площадок: Upwork, [Weblancer](#), [Kwork](#), [FL](#) и другие. Но будьте готовы зарабатывать портфолио и репутацию, а для этого первое время придётся перебиваться дешёвыми заказами.

Большой плюс — знание английского и умение строить вопросы на получение исчерпывающих ответов: вы сами себе РМ, тимлид и разработчик.

19.2 Компании аутсорсеры

Также ласково называемые галерами. А всё потому, что от сотрудников часто требуется многозадачность за зарплату одного специалиста.

Но не всё так плохо. Во-первых, аутсорсинг — это готовые задачи, которые на фрилансе нужно выбивать, доказывая свою профпригодность. Здесь же менеджеры сами договариваются и берут нужный таск. Во-вторых, корпоративная жизнь предполагает бесплатный чай/кофе, абонементы в спортзал, тимбилдинги, корпоративы и вообще комфортную работу. Не всегда, но в крупных компаниях это так.

19.3 Стартапы и небольшие компании

Неплохо для начинающих программистов и опасно для опытных

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

- итоговая зарплата окажется ниже обещанной;
- выплачиваться будет с задержкой;
- компания в любой момент может закрыться, не выдержав конкуренции.

Но это хороший опыт для стажёров, которые всё ещё думают, как стать программистом, младших разработчиков и тех, кто хочет уйти от фриланса.

19.4 Банки и энтерпрайз

Здесь потребуется разработка, направленная на решение проблем бизнеса, а не конечных пользователей. Именно поэтому в банки и бизнес-отрасли часто набирают только Senior-программистов, так как нужно дописывать и поддерживать готовые проекты, нередко с Legasy-кодом. Зарплаты обычно выше средних на рынке, а от самого разработчика требуется терпение и исполнительность.

19.5 Продуктовые компании

Если аутсорсинг сосредоточен на разноплановости, ограниченных сроках и бюджете, то продуктовая компания — на эффективном управлении своими решениями. Во втором случае никто не будет требовать от вас проект на вчера и ведение сложных переписок с кучей недопониманий. К тому же, вы явно будете видеть свой вклад в развитие одного большого проекта.

Как стать разработчиком продуктовой компании? Точно так же, как и аутсорсинговой, но будьте готовы, что требования могут быть выше, ведь компания ищет программиста непосредственно для себя, а не для левого дяди.

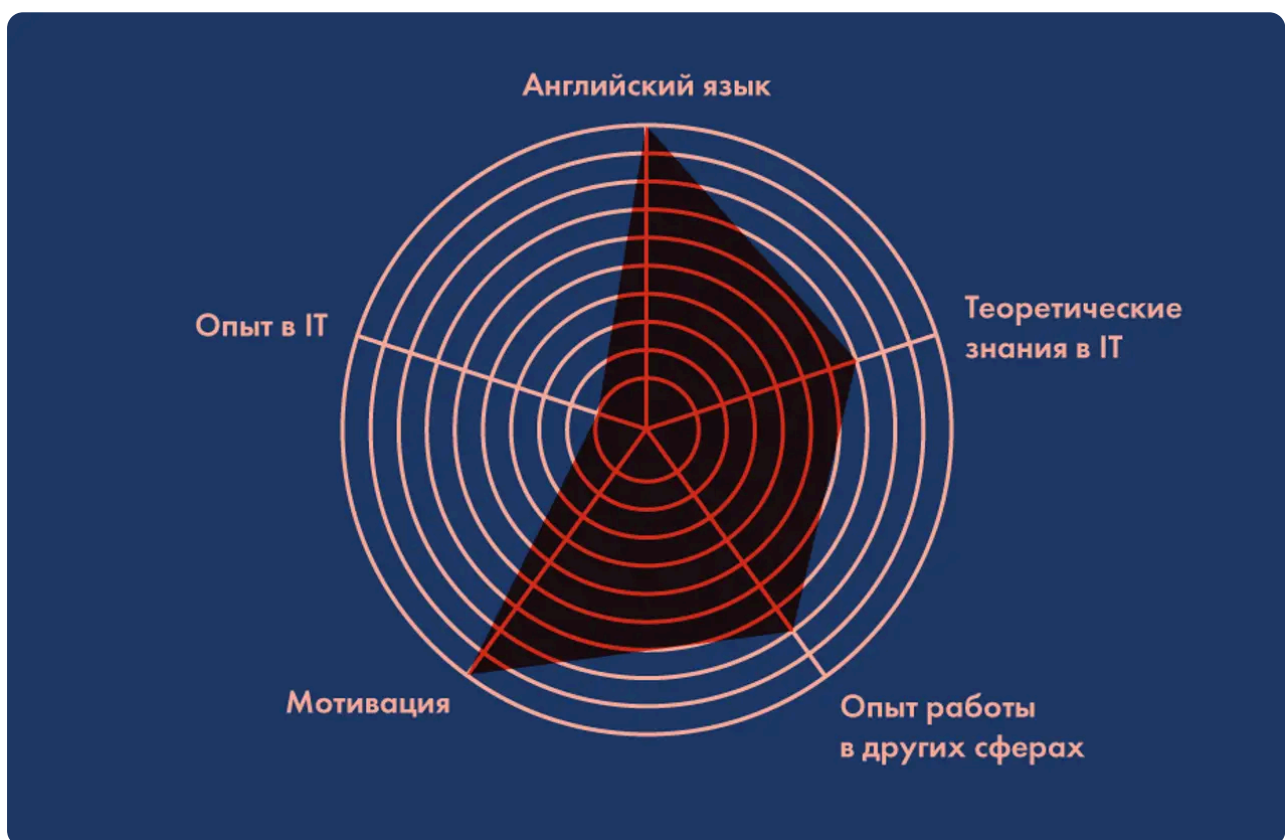
Где ещё можно работать программистом? Это фриланс, open source и

Мы тоже используем куки, потому что **без них вообще ничего не работает**

20. Как найти первую работу в IT

Всё чаще поднимается вопрос поиска и найма недооценённых специалистов — без диплома или соответствующего опыта работы. Исходя из этого, что же всё-таки нужно знать начинающему программисту?

Для начала вы должны показать, что нанять вас выгодно. В этом помогут теоретическая база, опыт в других сферах, знание английского и мотивация. Можно изобразить это так:



Источник — dataart.team

20.1 Где брать проекты для портфолио

Если в поисках ответа на вопрос, где можно выучиться на

Мы тоже используем куки, потому что **без них вообще ничего не работает**

Сложнее будет, если вы самоучка. Но и здесь можно что-то придумать:

- писать небольшие программы по туториалам;
- искать задания на фрилансе и выполнять их самостоятельно;
- принимать участие в разработке open source проектов.

20.2 Как составить резюме, если нет опыта

1. Нормально назовите резюме. В названии должны быть желаемая должность и ФИО кандидата, как здесь: «Junior Java — Пупкин Иван Иванович».
2. Резюме может быть в формате Doc, PDF или сайта-визитки, если вы веб-разработчик.
3. Добавьте адекватное фото и контакты.
4. Укажите места, где учились и повышали квалификацию.
5. Не копируйте бездумно инфу из разделов, что нужно знать, чтобы стать крутым программистом: вы должны доказать эти знания на собеседовании.
6. Если нет опыта работы, указывайте, сколько времени практикуетесь в том или ином языке/технологии.

Последний пункт в резюме программиста может выглядеть примерно так:

- Java — 1 year;
- PHP — 0,5 years;
- JavaScript — 0,5 years.

Мы тоже используем куки, потому что **без них вообще ничего не работает**

по поиску работы в офисе или в ремоут-режиме. Для России и стран

СНГ лучшие сайты с вакансиями для программистов это:

- [HeadHunter](#)
- [Хабр Карьера](#)
- [Вакансии Tproger](#)
- [Вакансии vc.ru](#)
- [Вакансии proglab](#)
- [DOU](#)

20.4 Как общаться с рекрутерами

Как вы понимаете, стать хорошим программистом в обход общения с HR невозможно. На собеседовании не стесняйтесь озвучивать свои желания и вопросы. Если на всё кивать головой, как автомобильная собачка, после трудоустройства можно обнаружить себя в крайне некомфортных условиях работы. Мы подготовили [13 вопросов работодателю, которые стоит задать на собеседовании](#).

Несмотря на это, нужно вести себя в меру сдержанно, демонстрировать серьёзность и говорить о своей мотивации.

21. Что спрашивают у Джунов на собеседовании

Огромный пласт вопросов приходится на теорию, чуть меньше занимает практика. Перед собеседованием изучите информацию о компании — обязательно спросят, почему именно она вас заинтересовала. Также [подготовьтесь к каверзным вопросам рекрутера](#).

Будут и логические задачи. При этом ответ не обязательно должен быть верным — важно озвучивать ход мыслей, показывать, как именно вы пришли к тому или иному решению

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Держите несколько [сложных задач с собеседований](#) с разбором и кодом.

Другие вопросы, которые могут прозвучать на собеседовании:

- Принципы ООП и их понимание.
- Команды Git — как сделать пуш, коммит и т. д.
- Основные SQL-запросы.

Всё остальное будет касаться выбранного языка.

22. Что делать, если вам не хватает знаний до вакансии Джуниор-программиста

Трудоустроиться после первого же интервью почти нереально. Не расстраивайтесь из-за отказов, а используйте их с пользой. Как же стать востребованным айтишником с нуля? Будьте готовы набивать шишки на собеседованиях, запоминать или записывать вопросы, которые ранее не встречались, включать их в программу подготовки к собеседованию.

Держите [учебные материалы и руководства для начинающих айтишников](#).

23. Заключение

Надеемся, этот гайд ответил на вопрос, как стать программистом, дал полезные советы и материалы, а также мотивировал на покорение новых вершин.

Остались вопросы? Задайте их в комментариях.

Мы тоже используем куки, потому что [без них вообще ничего не работает](#)

Для начинающих (+)

Обучение программированию (+)

Работа (+)

Советы (+)

Карьера (+)

Планы обучения (+)

316K открытий 333K показов

♡ 37 💬 22

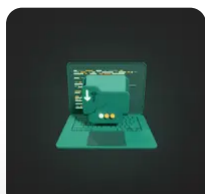


Реклама ... ✕

Канал об IT так, что поймёт даже резиновый утёнок.

Подписаться

Также рекомендуем



Как эффективно оптимизировать большой объем данных во фронтенде

Оптимизация производительности является ключевым аспектом разработки веб-приложений, особенно когда речь идет о работе с большими объемами данных. Именно такая

Мы тоже используем куки, потому что **без них вообще ничего не работает**



Exploring Various Market Instruments: Defining Targets and Potential Features

If you're a complete rookie considering making your first steps on financial markets – this guide is your flashlight in the financial fog.



Как подтянуть английский для собеседований за 2 месяца

Рассказали, как подтянуть английский язык перед собеседованием в IT-компанию. Внутри — план подготовки, упражнения и инструменты.



Практические задачи для изучающих SQL

Рассказываем о SQLTest.online — платформе, которая поможет вам прокачать навык написания SQL запросов на практических примерах.

Мы тоже используем куки, потому что **без них вообще ничего не работает**