

Основы программирования

# Выбор языка программирования

Актуальные профессии. Как выбрать язык программирования?

## Оглавление

[Обучение в GeekBrains](#)

[Где искать описания основных профессий](#)

[Выбор направления](#)

[Что более востребовано](#)

[Какой язык выбрать](#)

[Профессия «веб-разработчик»](#)

[Frontend-часть сайта](#)

[Что должен знать верстальщик](#)

[Принцип работы сайта в интернете](#)

[Backend-часть сайта](#)

[На чем пишут backend](#)

[Full-stack разработчик](#)

[Что должен знать full-stack разработчик](#)

[Выбор языка программирования](#)

[Алгоритм выбора языка программирования](#)

[Профессия «мобильный разработчик»](#)

[Универсальные языки программирования](#)

[Язык C++](#)

[Java vs C# vs Python](#)

[Практическое задание](#)

# Обучение в GeekBrains

Часть курсов GeekBrains объединяется в блоки с рабочим названием «профессия». Такой блок обеспечивает студента необходимым объемом знаний для работы по специальности.

Первая задача, которую нужно решить, — выбрать курс.

Курсы GeekBrains учитывают особенности восприятия студентами каждого предмета — например, синтаксиса конкретного языка программирования. Выпускники курсов, выполнявшие обязательные и дополнительные задания, нарабатывают и закрепляют навыки чтения кода, понимают логику работы приложения или части кода, быстро пишут код, осваивают логику разработки и методы решения рутинных задач.

## Где искать описания основных профессий

Все пакеты с направлениями по профессиям размещены на сайте: <https://geekbrains.ru/courses>.

## Выбор направления

На сайте GeekBrains используется термин «профессия». Например, веб-разработчик. Для каждой профессии разработан пакет учебных материалов, который состоит из одного или нескольких курсов.

Получить консультацию по интересующему вас направлению можно на [сайте](#). Нажмите на изображение заинтересовавшей вас профессии. В открывшемся окне следует оставить свои контактные данные — наш менеджер с вами свяжется и ответит на интересующие вас вопросы. Заполнение заявки ни к чему не обязывает.

Выбор направления является ключевым моментом в обучении. Распыляться на несколько профессий не стоит. Следует выбрать одну и хорошо ее освоить. Если считаете, что начав изучать сразу несколько, вы охватите большее количество вакансий, то это неверный подход. Легче и правильнее освоить один язык до хорошего уровня, после чего перейти к следующему.

## Что более востребовано

Мало знать язык программирования, следует научиться разрабатывать на нем — веб-сайты, приложения и другое. Язык — всего лишь инструмент.

Поэтому в первую очередь определяемся с направлением. Чтобы его выбрать, необходимо знать, что на рынке труда наиболее востребовано.

На рисунке слева направо представлены актуальные направления в порядке убывания востребованности. Анализ рынка вакансий в этой области (статистика hh.ru) показывает следующий порядок: веб-разработка (около 50 % вакансий), разработка мобильных приложений под iOS и Android. Остальная часть рынка — вакансии, в которых необходим навык создания десктопных и серверных приложений.

Для каждого направления существует множество языков, на которых можно написать frontend- и backend-части приложения.

Front end — та часть приложения, которую видит пользователь в браузере или в окне приложения.

Back end — часть приложения, которую обрабатывает сервер.

Каждый язык программирования применяется для решения задач в своей области. Например, Ruby — для веб, Java — для приложений под Android.

## Какой язык выбрать

Работодатель выбирает специалистов под популярные языки. Топ-10:

1. JavaScript.
2. Java.
3. PHP.
4. Python.
5. C#.
6. C++.
7. Ruby.
8. Swift.
9. Objective-C.
10. C.

Можно выучить непопулярный язык, но будет сложно найти работу.

## Профессия «веб-разработчик»

Около половины всех вакансий открываются в области веб-разработки. К ней же относятся языки: JavaScript, PHP, Ruby, Python (на  $\frac{1}{3}$  часть), C# ( $\frac{1}{2}$ ), Java ( $\frac{1}{3}$ ).

Профессия «Веб-разработчик» полностью описана [здесь](#).

### Frontend-часть сайта



Сначала внешний вид сайта дизайнер отрисовывает в графическом редакторе (например, в Photoshop). Он формирует внешний вид страниц, создает картинку по требованиям заказчика, то есть выполняет дизайн сайта.

Дизайнер должен обладать следующими личностными качествами:

- быть творческим человеком;
- уметь заботиться о людях (чтобы создавать удобный продукт);
- иметь инженерный склад ума и уметь конструировать.

Навыки веб-дизайнера:

- знать Photoshop;
- уметь сделать дизайн удобным, красивым и продающим;
- уметь рисовать на бумаге не обязательно!

У GeekBrains есть профессия «[веб-дизайнер](#)».

Когда отрисован макет (внешний вид сайта, всех его элементов и страниц), в работу вступает [верстальщик](#).

Он заставляет картинку, нарисованную в графическом редакторе, отображаться в браузере. Рабочие инструменты верстальщика — HTML, CSS и JavaScript.

На [HTML](#) реализовывают структуру сайта, задают расположение его основных блоков.

[CSS](#) отвечает за внешний вид элементов: блоков, графическое оформление, анимацию.

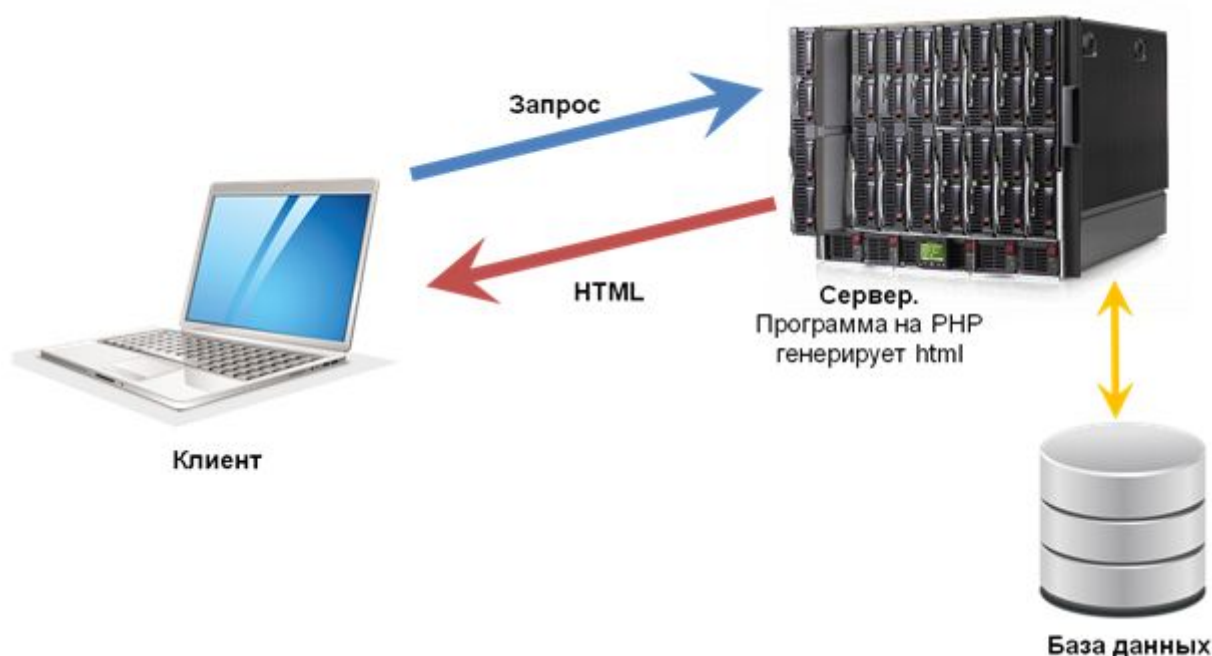
[JavaScript](#) реализует работу и поведение интерактивных элементов страницы (плавающих кнопок, всплывающих блоков и подобного). JavaScript работает на стороне клиента (то есть в вашем браузере).

## Что должен знать верстальщик

Если вы заинтересовались версткой, записывайтесь на профессию «[верстальщик](#)».

Должен знать	Плюсы	Минусы
<ul style="list-style-type: none"><li>• HTML</li><li>• CSS</li><li>• JavaScript</li></ul>	<ul style="list-style-type: none"><li>• Низкий порог вхождения</li><li>• Быстрота освоения</li><li>• Создаешь визуальную часть</li><li>• Фриланс</li></ul>	<ul style="list-style-type: none"><li>• Меньше денег (в среднем 56 тысяч рублей)</li><li>• Очень большая конкуренция</li></ul>

## Принцип работы сайта в интернете



С клиента (ноутбука или ПК) запрос отправляется на сервер, где располагается ваш веб-сайт. Сервер обрабатывает запрос и возвращает на ваш компьютер HTML-код, который и формирует страницу сайта в браузере.

Если использовать только HTML & CSS, то для однотипных страниц потребуется писать каждый раз новую HTML-часть. Чтобы этого избежать, применяются серверные языки программирования, например PHP, которые на стороне сервера формируют HTML-код страницы под запрос пользователя. После чего отправляют сформированный запрос клиенту. В случае с интернет-магазином для каждого товара будет сформирована своя страница с HTML-кодом непосредственно для описания этого товара. Данные о каждом товаре обычно хранятся в базе данных, откуда они запрашиваются командами серверного языка программирования.

Рассмотрим примеры HTML и PHP-кода.

HTML-код:

```
<div id="tooplate_footer">
  Copyright 2016 <a href="http://geekbrains.ru">GeekBrains"</a>
</div>
```

HTML-код + код PHP:

```
<div id="tooplate_footer">
  Copyright <?php echo date("Y"); ?> <a
href="http://geekbrains.ru">GeekBrains"</a>
</div>
```

В первой строке кода на чистом HTML написан вывод строки **Copyright 2016** и название портала **GeekBrains**. В данном случае код жестко прописан и без вмешательства человека изменяться не будет.

Во второй строке данного примера — строка php-кода, которая начинается с **<?php echo date.....>**, автоматически формирует и выводит текущий год. Намного удобнее, чем делать это вручную.

## Backend-часть сайта

Backend'ом называется та часть сайта или приложения, которая выполняется на стороне сервера и не видна пользователю.

После того как сайт сверстан, реализуется его серверная часть на backend. Здесь работают языки программирования, которые формируют код на стороне сервера: PHP, Ruby, Python, C#, Java.



Серверные языки обрабатывают такие запросы от пользователей, как отправка письма с сайта на почту, оставление заявки и подобные.

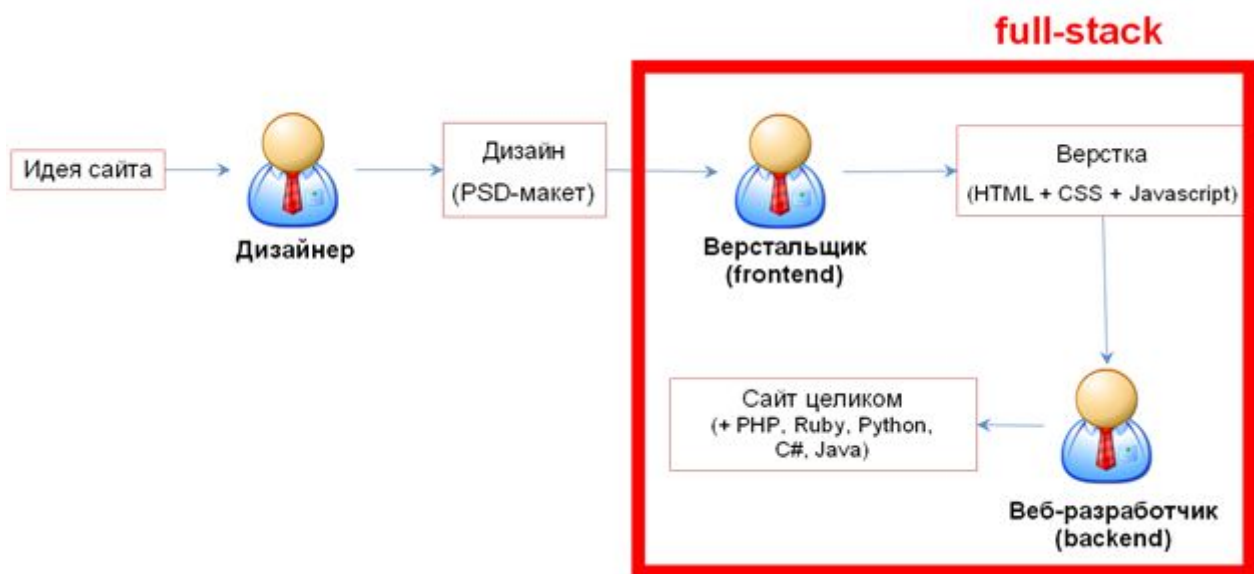
## На чем пишут backend

1. PHP.
2. Python.

3. C#.
4. Java.
5. Ruby.
6. Node.JS.

## Full-stack разработчик

Специалисты этого уровня умеют писать и клиентскую, и серверную часть приложения. Соответственно, они владеют набором инструментов верстальщика (HTML, CSS, JavaScript) и знают как минимум один серверный язык программирования.



После того как реализована и клиентская, и серверная часть, продукт передается сначала тестировщику, затем SEO-специалисту.

Тестировщик ищет ошибки в проекте, а задача SEO-специалиста — повысить позицию сайта в выдаче поисковых систем (например, Яндекс или Google).

### Что должен знать full-stack разработчик

Должен знать	Плюсы	Минусы
<ul style="list-style-type: none"> <li>• Back-end язык программирования + фреймворк</li> <li>• HTML</li> <li>• CSS</li> <li>• JavaScript + фреймворк</li> </ul>	<ul style="list-style-type: none"> <li>• Небольшая конкуренция</li> <li>• Большие деньги</li> </ul>	<ul style="list-style-type: none"> <li>• Нужно много учиться</li> </ul>





















Фреймворк — это набор библиотек и инструментов. В нем вы не пишете код от точки до точки, а выбираете уже имеющийся в библиотеке инструмент или функциональность. Фреймворк уже

включает в себя реализацию функций. Речь о том, чтобы не описывать функцию, а взять готовую — одна строка кода вместо пятидесяти.

Если вы умеете пользоваться фреймворком, разработка идет значительно быстрее. По ссылке вы найдете полное описание нашего курса по php-фреймворку [Yii](#).

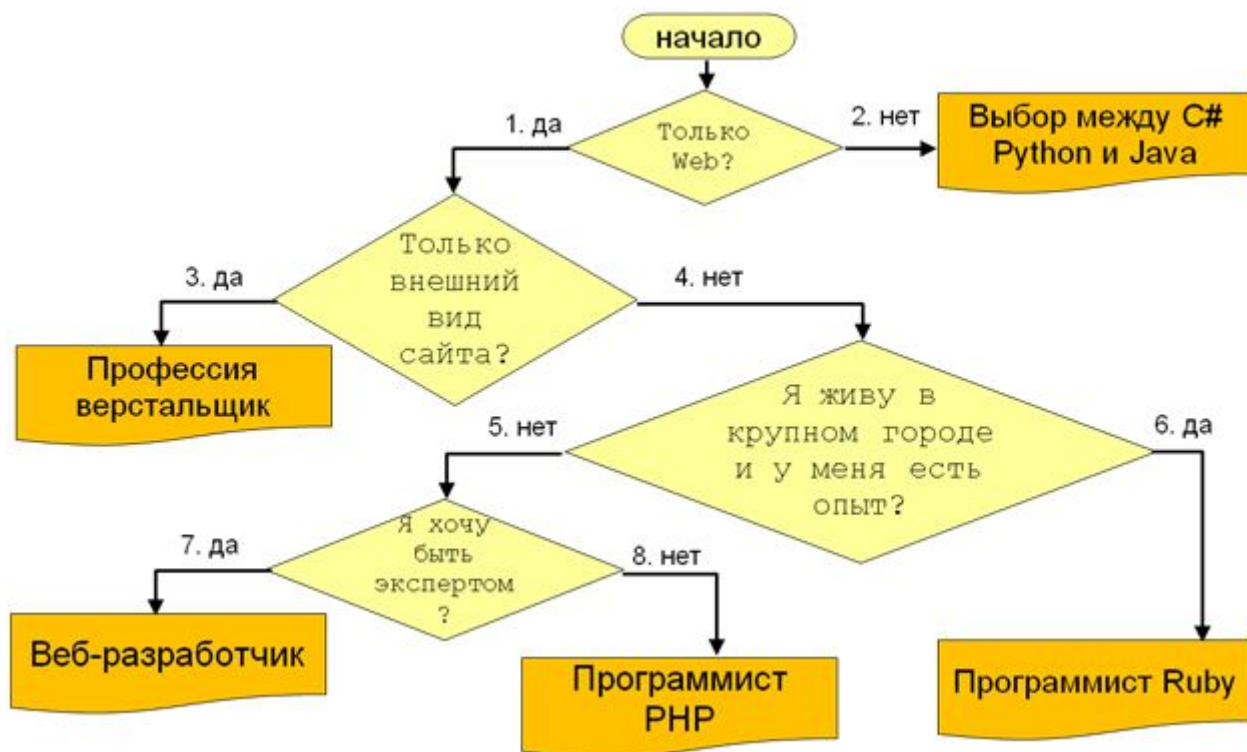
## Выбор языка программирования

Если хотите заниматься исключительно веб-разработкой, из серверных языков вам подойдут PHP и Ruby — они специализированы для веба. Остальные универсальны, подходят для реализации любых задач.

	PHP	Ruby (on Rails)	C# ASP.NET	Python Django	Java (JSP, Spring)
Только web?	ДА	ДА	НЕТ	НЕТ	НЕТ
Сложность					
Количество вакансий					
Конкуренция					
Скорость разработки					
	Фриланс	Стартапы	Корпорации	Стартапы	Корпорации



## Алгоритм выбора языка программирования



Пройдите по этому алгоритму, отвечая на вопросы в каждом блоке, и сможете понять, в каком направлении стоит выбирать профессию.

Если вы заинтересовались версткой, записывайтесь на профессию «[верстальщик](#)».

Хотите стать программистом PHP — [вам сюда](#).

Профессия «программист Ruby» описана [здесь](#).

Полное описание профессии «веб-разработчик» — [по этой ссылке](#)

По следующим ссылкам вы найдете профессии «[программист Python](#)», «[программист Java](#)» и «[программист C#](#)».

## Профессия «мобильный разработчик»

Эти специалисты разрабатывают продукты под мобильные платформы, такие как Android и iOS.

Для разработки приложений под Android необходимо знать Java, после чего нарабатывается опыт программирования под эту платформу. В GeekBrains обучение по этому направлению проводится [в профессии «программист Android»](#). Под управлением этой операционной системы работает большинство гаджетов в мире.

Разработка приложений для платформы iOS ведется на Objective-C и Swift.

Objective-C — традиционный и устоявшийся язык для программирования под iOS. Вакансий, где требуется только Swift, намного меньше по сравнению с предложениями, для которых нужно знать Objective-C. Если вы в дополнение к нему владеете Swift, это плюс.

Для обучения программированию под iOS в GeekBrains есть направление [«Программист iOS»](#).

Разработка под Windows Phone ведется на C#, но вакансий в этой области сейчас мало.

# Универсальные языки программирования

## Язык C++

C++ — третий по популярности язык программирования. Неоспоримый плюс — его кроссплатформенность: написанные на нем программы подходят для любой из платформ. Основная ниша для C++ — высокопроизводительные и низкоуровневые проекты. C++ подойдет как второй язык для изучения. C++ невозможно осваивать без понимания алгоритмов и знания архитектуры компьютера («железа»). В этом языке намного больше тонкостей, чем в других, нужно хорошо разбираться в математике. C++ сложнее, чем остальные языки, но он позволяет писать гораздо более мощные и быстрые программы. Например, 3D-игры, приложения, работающие с графикой, то есть те, от которых требуется большая скорость выполнения.

[Здесь описание профессии «программист C++».](#)

## Java vs C# vs Python

Часто возникает вопрос, какой из этих языков лучше.

Они похожи и имеют тождественную функциональность, поэтому не важно, какой из них вы выберете для изучения. При хорошем знании одного переход на другой не составит труда. На всех этих языках можно писать как сайты, так и приложений для десктопов и серверов.

Небольшие отличия:

Java	C#	Python
<ul style="list-style-type: none"><li>• Больше вакансий</li><li>• Приложения Android</li><li>• Сайты — реже</li><li>• Чаще, чем остальные — для тестирования</li></ul>	<ul style="list-style-type: none"><li>• Только под Windows</li><li>• Проще учить после C/C++</li><li>• Игры — чаще, чем на остальных языках</li></ul>	<ul style="list-style-type: none"><li>• Подходит для изучения в качестве первого языка</li></ul>

Если у вас есть опыт написания программ на каком-то из этих языков, правильнее будет продолжать углубляться в его изучение, а не переключаться на другой язык.

При полном отсутствии опыта в программировании лучше сделать выбор в пользу Python. Языки C# и Java намного легче изучать при наличии хотя бы небольшого опыта в программировании. Вы

получите его, если пройдете до конца данный курс или бесплатный видеокурс «[Основы программирования](#)».

Какое бы направление вы ни выбрали, главным остается чтение тематической литературы и практика.

## Практическое задание

1. Выбрать язык программирования, который бы вам хотелось изучить. Написать, почему остановились именно на нем и что можно будет разрабатывать с помощью этого языка.