REACT МАТЕРИАЛ СО СКИЛЛФЭКТОРИ

**15.2 Компонентный подход**

 Добавить страницу в мои закладки

Поздравляем! У вас за плечами несколько покорённых вершин — *HTML, CSS и JavaScript.* Мы приступаем к изучению фреймворков — палочек-выручалочек в арсенале разработчика, которые значительно упрощают и оптимизируют весь процесс создания сайтов и приложений.

Но прежде чем мы начнём, давайте настроимся на нужный лад. Ведь уже скоро вы будете частью огромного профессионального сообщества, использующего свой особый язык. Чтобы чувствовать себя как рыба в воде и оперировать профессионализмами со знанием дела, рекомендуем ознакомиться с этой [статьёй](https://ravesli.com/slovar-programmista-sleng-kotoryj-dolzhen-znat-kazhdyj-koder/).

https://ravesli.com/slovar-programmista-sleng-kotoryj-dolzhen-znat-kazhdyj-koder/

Ниже мы собрали список слов, часто используемых автором этого раздела. Изучение английского – не та цель, которую мы преследуем в рамках этого курса, но понимание таких слов в контексте значительно упростит восприятие новой темы.

**Деплой** (или ещё «задеплоить», от англ. *deploy*) — это развёртывание (перенос) программного обеспечения (исполняемого кода) на сервер или устройство, где оно будет работать.

**Заимпортить** (от англ. *import*) — импортировать, загружать. Пример: «заимпортить *React*-библиотеку».

**Инициализировать** (от англ. *initialize*) — устанавливать, привести в готовность.

**Коммитить** (или ещё «закоммитить», «сделать коммит», от англ. *commit*) — сохранить/зафиксировать изменения кода в локальном репозитории (хранилище) кода.

**Лоадер** (от англ. *load* — загружать) — анимированная картинка, показывающая пользователю, что идет загрузка.

**Отбранчеваться** (от англ. *branch* — ветка) — отделиться. «От родительского элемента отбранчевались 3 элемента *children*».

**Отконнектиться** (от англ. *disconnect*) — отсоединиться.

**Парсить** (от англ. *parse* — разбивать на составные части) — автоматически обрабатывать с целью получения нужных данных. Пример: «Распарсить код». Также означает «прочитать» код.

**Прилажка** — приложение.

**Пушить** (или ещё «запушить», от англ. *push*) — публикация на сервер с помощью команды *push*, отправить локальные изменения, которые были закоммичены в локальный репозиторий, на удалённый репозиторий (это делает доступным ваш код другим разработчикам).

**Рандом** (или ещё «рандомный», от англ. *random*) — произвольный, случайный.

**Рендерить** (от англ. *render* — исполнять, передавать) — визуализировать, отрисовать компонент на ui. Пример: «*React* рендерит всё дерево *DOM* заново».

**Рутовый** (от англ. *root* — корень) — корневой, находящийся в корне папки. Пример: «Рутовый компонент».

**Скрашиться** (от англ. *crash* — разрушаться, ломаться). Пример: «Приложение скрашилось».

**Стартин кит** (от англ. *starting kit)* — стартовый комплект.

**Транспайлерить**, **транспилить** — преобразовывать синтаксис одного языка в другой для того, чтобы приложение корректно отработало в нужном окружении. Пример: «*JSX* транспилится с помощью программы-транспайлера*Babel* в *JS* код, чтобы наш сайт работал в браузере».

**А**

**Абстрактная функция**(или *«чистая виртуальная функция»*) — это функция, которая не имеет определения. Её переопределяют дочерние классы.

**Ава**(или *«аватарка»*, от англ. *«avatar»*) — графическое представление (картинка/фото/изображение) человека в Интернете/соц.сетях/мессенджерах/играх и т.д.

**Автогад** — система автоматизированного проектирования и черчения AutoCAD.

**Админ**(от англ. *«admin»*) — администратор.

**Айди**(от англ. *«identificator»*) — идентификатор.

**Айпи**(или *«Айпишник»*, *«Айпи-адрес»*, от англ. *«****I****nternet****P****rotocol Address»*) — уникальный адрес компьютера в сети Интернет. Этот адрес каждому компьютеру присваивает провайдер, который предоставляет устройству возможнось выхода в Интернет.

**Айти**(или *«Ай-Ти»*, *«АйТи»*, сокр.*«****IT****»*от англ.*«****I****nformation****T****echnologies»*) — информационные технологии.

**Айти-евангелист**(или *«IT-проповедник/пропагандист»*) — человек, который занимается пропагандой в сфере IT.

**Айтишник**(он же *«айтишнег», «ойтишнег», «ойтишник»*) — специалист в информационных технологиях. Нередко айтишниками называют всех программистов и просто продвинутых пользователей компьютера и Интернета.

**Ака**(или ***«a.k.a»***, от англ. *«****a****lso****k****nown****a****s»*) — если дословно, то «еще известен как». Другими словами, никнейм, псевдоним.

**Аккаунт** (или *«акаунт»*, от англ. *«account»*) — учетная запись пользователя, которая создается после регистрации в какой-либо системе (например, на сайте или в программе). Как правило, для входа в свой аккаунт вам нужно ввести логин и пароль.

**Алгоритм** — последовательность действий для решения определенной задачи.

**Аноним**(или *«анонимус»*, *«анон»*, от англ. *«anonym/anonymous»*) — человек, скрывающий свою личность. Как правило, использует вымышленные или ненастоящие данные о себе.

**Антивирус** (или *«антивирусник»*) — это специализированная программа для обнаружения компьютерных вирусов и их удаления. Также может выполняться профилактика в виде обнаружения и прекращения выполнения нежелательных программ и регулярная проверка файлов компьютера для выявления подозрительных файлов.

**Апгрейд**(от англ *«upgrade»*) — обновление/модернизация аппаратного обеспечения (железа).

**Апдейт**(от англ *«update»*) — обновление/модернизация программного обеспечения (софта).

**Апликуха**(от англ. *«application»*) — прикладное программное обеспечение. Например, компьютерная программа или мобильное приложение.

**Апрув**(от англ. *«approve»*) — подтверждение, согласие, одобрение чего-либо.

**Апрувить**(или *«заапрувить»*) — подтверждать, давать добро на выполнение чего-либо.

**Артефакт** (от англ. *«artefact»*) — это нежелательный визуальный эффект по типу искажения цветов, форм объектов, несовместное движение частей изображения и т.д.

**Архиватор** — программа для создания архивов (например, *7-Zip*, *WinRAR* и др.).

**Ассемблер** (или *«Асм»*, от англ.*«Assembler»*) — это транслятор (переводчик), который переводит код, написанный на [**языке Ассемблера**](https://ravesli.com/uroki-assemblera/), в машинный язык.

**Аська** (или ***ICQ*** от англ. *«I seek you»*) — централизованная служба мгновенного обмена сообщениями в сети Интернет. С 2010 года принадлежит инвестиционному фонду Mail.ru Group. До возникновения социальных сетей была очень популярной, сейчас же ею пользуется сравнительно малое количество пользователей Интернета.

**Аттач** (от англ. *«****attach****ment»*) — прикрепляемый к чему-либо файл.

**Аттачить** — прикреплять (например, файл или фото/скриншот к письму).

**Аутсорс**(или ***«аутсорсинг»***, от англ. *«outsource»*) — передача компанией выполнения определенной работы специалистам вне штата (фрилансерам или работникам других компаний). Например, есть специализированные конторы бухгалтеров, которые обслуживают десятки других компаний *на аутсорсе*, не являясь сотрудниками этих компаний.

**Аутстаффинг**(или ***«аутстаф»***, от англ. *«outstaffing»*) — это «аренда» специалиста, работающего в компании-подрядчике (аутстаффер) для работы на проекте в компании-заказчике. Как правило, заказчики нанимают целые команды на определенные проекты, оплачивая рабочие часы специалистов. Проект закончился = команда переходит на другой проект (может даже от другого заказчика) или специалисты разбрасываются по разным проектам или «сидят на бенче» (*см. определение бенч*). Таким образом, успешно существуют целые компании-гиганты, которые не имеют своих продуктов, а просто «перепродают» часы работы программистов на другие проекты.

**Аффтар** (или *«афтор»*, *«афтар»*) — автор.

[**Agile**](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) (или *«Аджайл»*, от англ. *«agile software development»*) — это семейство гибких методологий разработки, которые используются для создания проектов/продуктов в сфере программного обеспечения. К ним относятся Scrum, экстремальное программирование, DSDM и пр. Отличительной особенностью гибких методологий разработки является наличие итераций (коротких циклов), когда большой проект разбивается на подпроекты с задачами, на выполнение каждого из которых отводится 2-3 недели, с возможностью регулировать вектор движения разработки непосредственно в процессе самой разработки.

[**API**](https://ravesli.com/what-is-an-api/)(или *«Апишка»*, сокр. от англ. *«****A****pplication****P****rogramming****I****nterface»*) — это набор функций, которые позволяют одним приложениям получать доступ к данным и взаимодействовать с внешними (другими) приложениями, программными компонентами, операционными системами или микросервисами. Если упростить, API доставляет ответ пользователя системе и отправляет ответ системы обратно пользователю. Например, когда вы используете приложение на своем смартфоне, оно подключается к Интернету и отправляет данные на сервер —> сервер извлекает эти данные, интерпретирует их, выполняет над ними необходимые действия и отправляет их обратно на ваш смартфон —> приложение на вашем смартфоне интерпретирует полученные от сервера данные и предоставляет вам нужную информацию в читабельном виде. Каждый раз, когда вы используете на своем смартфоне Facebook, отправляете мгновенное сообщение или проверяете погоду, вы используете API.

**ASCII** (сокр. от *«****A****merican****S****tandard****C****ode for****I****nformation****I****nterchange»*) — это американский стандартный код для обмена информацией, который определяет способ представления символов английского языка (+ несколько других) в виде чисел от 0 до 127. Например: код буквы 'а' — 97, код буквы 'b' — 98.

**Б**

**Баг**(от англ. *«bug»*) — ошибка в программе/коде, из-за которой результаты выполнения программы неправильные.

**Багрепорт**(от англ. *«bug report»*) — сообщение или отчет об ошибке с её описанием, причиной возникновения, приоритетом и прочими свойствами.

**Байт** — единица хранения и обработки цифровой информации, равная 8 битам.

**Бан**(от англ. *«ban»*) — лишение или ограничение каких-либо прав пользователя (например: лишение права оставлять комментарии, просматривать определенные страницы и пр.). **Пожизненный бан** — исключение пользователя из системы/игры/сайта/группы и т.д.

**Баннер** — графическое или анимированное изображение, чаще всего, рекламного характера.

**Банщик** — человек-дизайнер, который занимается версткой баннеров.

**Батник**— командный файл с расширением *.bat*. Используется для работы с командной строкой в Windows.

**Баян**— шутка или информация, которая ранее уже была опубликована/известна.

**БД** (сокр. от *«****Б****аза****Д****анных»*) — программный или программно-аппаратный комплекс, разработанный для хранения большого объема различной информации.

**Бенефит**(от англ. *«benefit»*) — непрямое денежное или другое нематериальное вознаграждение, которое чаще всего используется для мотивации сотрудников. Другими словами, бонус.

**Бенч**(от англ. *«bench»*) — режим ожидания. Выражение ***«программист на бенче»*** означает, что программист простаивает вхолостую в ожидании нового проекта/задания и, по факту, ничего не делает, но зарплату получает.

**Бенчмарк**(от англ. *«benchmark»*) — тест производительности. Как правило, используется для определения сравнительных характеристик системы или программы.

**Бета-версия** — тестовая версия какого-либо продукта (например, программы или игры), которая выпускается с целью получить обратную связь от пользователей для усовершенствования и устранения ошибок готового выпускаемого продукта.

**Бета-тестер** — человек, который использует бета-версию продукта и предоставляет свою обратную связь о продукте для разработчиков.

**Библиотека** — это набор скомпилированного кода (например, функций), который был «упакован» для повторного использования в других программах. С помощью библиотек можно расширить возможности программ. Например, если вы пишете игру, то вам придется подключать библиотеки звука или графики (если вы не хотите самостоятельно их писать с нуля).

**Бит** (англ. ***«bit»*** от*«****bi****nary digi****t****«*) — наименьшая единица измерения количества информации/данных. Может содержать в качестве значения 0 или 1.

**Биткоин** (от англ. *«bitcoin»*) — одна из самых известных криптовалют в мире.

**Битый**— нерабочий. Например, *«битые ссылки»* — это ссылки, которые ведут на несуществующие страницы. Пользователь, попадая на подобную страницу, видит *«Ошибка 404»* (Error 404).

**Блок-схема** — графическая нотация для описания алгоритмов. Используется программистами в процессе разработки и анализа логики работы программных компонентов.

**Блокчейн** (от англ. *«blockchain»*) — система учета проводимых транзакций с криптовалютой.

**Болванка**— чистый (незаписанный) компакт-диск.

**Бот** (сокр. от *«ро****бот****«*) — программа, имитирующая действия человека, иногда с зачатками искусственного интеллекта.

**Браузер**(от англ. *«browser»*) — программа, с помощью которой можно просматривать веб-страницы (другими словами, выходить в Интернет).

**Брейншторм**(от англ. *«brainstorm»*) — способ генерации новых идей или решения существующих проблем путем мозгового штурма.

**Буржунет**— зарубежная (нерусскоязычная) часть Интернета. Очень часто под этим термином подразумеваются все англоязычные сайты.

**Бутовый** (от англ. *«boot»*) — загрузочный (например, *«бутовый диск»* — загрузочный диск).

**Буфер данных** — это просто часть памяти, зарезервированная для временного хранения данных, когда они перемещаются из одного места в другое.

**Бэкап**(от англ. *«backup»*) — резервное копирование. Создание резервной копии проекта/сайта/данных, чтобы в случае непредвиденных обстоятельств/сбоя можно было вернуть всю систему в прежнее состояние. Хорошей практикой является регулярное обновление бэкапов и их хранение сразу в нескольких местах (не только на рабочем компьютере).

[**Бэкенд**](https://ravesli.com/kakoj-yazyk-programmirovaniya-uchit-pervym-kak-zarabotat/#server)(от англ. *«back-end»*) — разработка серверной (или «внутренней») части продукта (например, сайта или приложения).

**В**

**Валидный**(от англ. *«valid»*) — корректный, действительный. Тот, который соответствует требованиям/условиям/правилам.

**Варез**(от англ. *«warez»*) — любая интеллектуальная собственность или программное обеспечение, которое свободно распространяется в сети Интернет, нарушая при этом права правообладателя.

**Варезник**— сайт, который распространяет варез (пиратское программное обеспечение, пиратские фильмы/музыку и пр.). Обычно, термин *«варезник»* применяется к сайтам-файлообменникам.

**Вебинар**(от англ. *«****web****based sem****inar****»*) — это семинар, презентация или лекция на определенную тему, которая проходит онлайн (в режиме реального времени) в Интернете. Под этим термином также подразумевают и запись вебинара, который проходил ранее.

**Видос** (или *«видосик»*) — видео.

**Видюха** (или *«видяха»*) — видеокарта.

**Вики** — Википедия (свободная энциклопедия).

**Виндоус**(или *«Винда»*) — операционная система Windows.

**Виндузятник** — саркастическое название пользователя операционной системы Windows.

**Винт**(или *«хард»*) — винчестер. Он же ***«жесткий диск»***, ***«HDD»***.

**Виртуальный базовый класс** — это родительский класс, объект которого является общим для использования всеми дочерними классами.

**Вирус** (от англ. *«virus»*) — программа, заражающая другие программы. Как правило, вирусы используют хакеры для взлома системы и распространения своего влияния на максимальное количество устройств с дальнейшим нанесением вреда (например, повреждение или удаление данных, кража информации и т.п.).

**Ворд** — программное обеспечение MS Word для работы с текстовыми файлами.

**Воркшоп**(от англ. *«workshop»*) — обучающее мероприятие для работников из определенной сферы деятельности (например, программистов, фотографов или дизайнеров).

**Выебщик**— программист или человек, работающий в IT, с завышенным ЧСВ, который всегда выражает свое экспертное мнение, нередко в неподходящих случаях, и не уважает мнения окружающих (очень часто даже не удосуживается их выслушать). Другими словами — «выпендрежник».

**Выкатить** — опубликовать или сделать доступным что-либо. Например, *«выкатить новую часть игры»* — опубликовать новую часть игры.

**Г**

**Гаджет** (от англ. *«gadget»*) — техническое устройства для облегчения выполнения какой-либо задачи (например, телефон является гаджетом).

**Гайд** (от англ. *«guide»*) — инструкция, руководство.

**Галера** — компания или организация, в которой трудится программист. В основном, термин *«галера»* применяется к компаниям, где практикуется плохое отношение начальства к работникам.

**Гейм** (от англ. *«game»*) — игра.

**Геймдев**(от англ. *«****game dev****elopment»*) — разработка/создание игр.

**Геймер**(от англ. *«gamer»*) — игрок.

**Геймпад** (от англ. *«gamepad»*) — игровой пульт. Обратите внимание, это не то же самое, что джойстик.

**Геймплей** (от англ. *«gameplay»*) — игровой процесс, сюжет игры.

**Генерить** (или *«сгенерить»*, от англ. *«generate»*) — сгенерировать, создать, сформировать что-либо.

**Гибернация**(от англ. *«hibernation»*) — режим сна, энергосберегающий режим операционной системы компьютера, при котором все данные пользователя (в т.ч. активных программ) сохраняются на жесткий диск.

**Гиг**(или *«гектар»*) — гигабайт.

**Гик**(от англ. *«geek»*) — человек, который хорошо разбирается в определенной сфере деятельности, являясь фанатиком своего дела. Сейчас под *«гиками»* очень часто подразумевают именно программистов.

**Гиф** (или ***«гифка»***) — анимация, изображение в формате *.gif*.

**Глобальные переменные**— это переменные, которые объявлены вне главного (основного) блока программы. Они создаются при запуске программы и уничтожаются при её завершении. Их можно использовать в любом месте файла, после их объявления.

**Глюк**— непонятный сбой, задержка, торможение процессов в программе или игре.

**Глючит** — медленно работает или работает с ошибками. То же самое, что и *«виснет»*, *«лагает»*.

**Говнокод**— плохой код, который некорректно написан, выполняет не то, что нужно, или код, в котором сложно разобраться и который сложно поддерживать/модифицировать.

**Говнокодер**— программист, пишущий говнокод.

**Гребцы**— работники *галеры* (компании).

**Грызун** — компьютерная мышь.

**Гуглить**— искать информацию в поисковике Google.

**Гуй** (сокр. ***«GUI»*** от англ. *«****G****raphical****U****ser****I****nterface»*) — графический интерфейс пользователя.

**Гуру** — опытный специалист, мастер в своем деле.

**Д**

**Дамп** (от англ. *«dump»*) — файл с полной или частичной копией содержимого памяти компьютера или базы данных в момент создания данного файла. Очень часто дампы используются с целью создания и поддержания в актуальном состоянии системы бэкапа.

[**Дебажить**](https://ravesli.com/urok-26-otladka-programm-stepping-i-breakpoints/)(от англ *«debug»*) — отладка кода, т.е. поиск и исправление ошибок в коде.

**Девайс**(от англ. *«device»*) — техническое устройство (например, ноутбук или смартфон).

**Девелопер**(от англ. *«developer»*) — разработчик, программист.

**Девелопмент**(от англ. *«development»*) — разработка чего-либо (например, программ или игр).

**Дедлайн**(от англ. *«deadline»*) — крайний срок выполнения или сдачи проекта/задания/работы.

**Дезигнер**(от англ. *«designer»*) — дизайнер.

[**Декремент**](https://ravesli.com/urok-40-inkrement-dekrement-pobochnye-effekty/) (от англ. *«decrement»*) — уменьшение значения на заданное значение (как правило, на единицу).

**Делфи** — язык программирования Delphi.

**Демка**(или ***«демо»***, от англ. *«demo»*) — демонстрационная версия чего-либо, как правило, с урезанным функционалом или с пробным периодом использования.

**Деплой**(или *«задеплоить»*, от англ. *«deploy»*) — это развертывание (перенос) программного обеспечения (исполняемого кода) на сервер или устройство, где оно должно функционировать.

[**Деструктор**](https://ravesli.com/urok-120-destruktory/) (от англ. *«destructor»*) — это специальный тип метода класса, который выполняется при удалении объекта класса. В то время как конструкторы предназначены для инициализации класса, деструкторы предназначены для очистки памяти после него.

**Дефолтный**(или *«по дефолту»*,от англ. *«default»*) — тот, что используется по умолчанию, стандартный.

**Джуниор**(или ***«джун»***, от англ. *«****Junior****Developer»*) — начальный уровень специалиста. Человек, имеющий минимальный опыт (как правило, до 2 лет) и навыки в определенной сфере деятельности.

[**Динамическая библиотека**](https://ravesli.com/staticheskie-i-dinamicheskie-biblioteki/) (или *«общая библиотека»*) — состоит из подпрограмм, которые подгружаются в вашу программу во время её выполнения. При компиляции программы, которая использует динамическую библиотеку, эта библиотека не становится частью вашего исполняемого файла, она ​​так и остается отдельным модулем. В Windows динамические библиотеки имеют расширение *.dll* (сокр. от *«****d****ynamic****l****ink****l****ibrary»* — *«библиотека динамической компоновки»*), тогда как в Linux динамические библиотеки имеют расширение *.so* (сокр. от *«****s****hared****o****bject» — «общий объект»*). Одним из преимуществ динамических библиотек является то, что разные программы могут совместно использовать одну копию динамической библиотеки, что значительно экономит используемое пространство и рабочее время программистов. Еще одним преимуществом динамической библиотеки является то, что её можно обновить ​​до более новой версии без необходимости перекомпиляции всех исполняемых файлов, которые её используют.

[**Динамический массив**](https://ravesli.com/urok-86-dinamicheskie-massivy/) (англ. *«dynamic array»*) — это массив, размер которого может изменяться ​​во время выполнения программы.

[**Динамическое выделение памяти**](https://ravesli.com/urok-85-dinamicheskoe-vydelenie-pamyati-operatory-new-i-delete/) — это способ запроса памяти из операционной системы запущенными программами по мере необходимости. Эта память не выделяется из ограниченной памяти стека программы, а выделяется из гораздо большего хранилища, управляемого операционной системой — кучи. На современных компьютерах размер кучи может составлять гигабайты памяти.

**Директория** — папка.

**Дисконнект** (или *«отконнектиться»*, от англ. *«disconnect»*) — разрыв связи или соединения.

**Дистрибутив** — форма распространения программного обеспечения.

**Доки** — документы.

**Домен**(от англ. *«domain»*) — уникальный адрес/имя сайта в Интернете. Например, ravesli.com — это домен, который состоит из доменного имени ravesli и домена первого уровня .com.

**Драйвер** — программа, которая служит для взаимодействия программного обеспечения устройства с аппаратным обеспечением устройства (являясь, по сути, посредником).

**Дрова** — драйвера.

**Дропнуть** (от англ. *«drop»*) — удалить, сбросить, уничтожить.

[**Дружественная функция**](https://ravesli.com/urok-126-druzhestvennye-funktsii-i-klassy/) — это функция, которая имеет доступ к закрытым членам класса, но при этом сама не является членом данного класса. Во всех других отношениях дружественная функция является обычной функцией. Ею может быть как обычная функция, так и метод другого класса.

**Дыра** — лазейка в коде ПО, которую могут использовать хакеры в своих целях.

**DDoS-атака** (от англ. *«****D****istributed****D****enial****o****f****S****ervice attack»*) — хакерская атака, способная частично или полностью вывести из строя серверы атакуемого Интернет-ресурса. Суть атаки заключается в посылании веб-серверу большого количества запросов одновременно с большого количества устройств (так называемой *«ботнет-сети»*).

**DIY**(сокр. от англ. *«****D****o****I****t****Y****ourself»*) — самодельничество, или «сделай сам»; вид деятельности, при котором вы что-то делаете своими руками.

**Ж**

**Железо**— аппаратное обеспечение (комплектующие) устройств (например, компьютеров, ноутбуков или смартфонов).

**З**

[**Заголовочные файлы**](https://ravesli.com/urok-21-zagolovochnye-fajly/) (или *«заголовки»*, англ.*«header files»*) — их целью является удобное хранение набора объявлений объектов для их последующего использования в других программах.

**Задрот**— человек, который слишком много времени уделяет виртуальной реальности (будь то игры, программирование или еще что-либо). Также иногда под этим термином подразумевают фанатиков своего дела (не зависимо от специальности).

**Зазиповать** — сделать архив каких-либо файлов в формате *.zip*.

**Законнектиться**(или *«подконнектиться», «приконнектиться»*, от англ. *«connect»*) — присоединиться к чему-либо.

**Запилить**(или *«залить»*) — загрузить, выставить на всеобщее обозрение. Например, *«запилил видосик в YouTube»* — загрузил видео в YouTube.

**Зарарить** — сделать архив каких-либо файлов в формате *.rar*.

**Зафакапить**— не справиться с выполнением поставленной задачи.

**Зашквар**— позор, гнусное поведение, какой-то постыдный поступок или действие.

**Звуковуха** (или *«звукавуха»*) — звуковая карта в персональном компьютере (сервере).

**И**

**ИБ** — **И**нформационная **Б**езопасность.

**Ибешник** — сотрудник подразделения информационной безопасности.

**Идентификатор** — это имя переменной, функции, класса или другого объекта в языке программирования.

**Иксы** (от англ. ***«Х»***) — так называют в просторечье стандарт X Window System, на котором построена графическая оболочка в [**Linux-дистрибутивах**](https://ravesli.com/obzor-distributivov-linux-kakoj-vybrat/), по аналогии с окнами в ОС Windows.

**Имба** (от англ. *«****imba****lanced»*) — несбалансированный персонаж, предмет или иной элемент компьютерной игры. Данный термин используется в основном в онлайн-играх и применяется к более сильному на фоне остальных персонажу.

**ИМХО**(сокр. *«****IMHO****»*от англ.*«****I****n****M****y****H****umble****O****pinion»*) — по моему скромному мнению.

**Инди-игра**(от англ. *«indie game»*, *«independent video game»*) — компьютерная игра, созданная отдельным разработчиком или небольшим коллективом без финансовой поддержки издателя компьютерных игр. Простыми словами, игра созданная независимым автором.

**Инет** — Интернет.

[**Инкапсуляция**](https://ravesli.com/urok-115-inkapsulyatsiya-gettery-i-settery/) (англ. *«encapsulation»*) — это процесс скрытого хранения деталей реализации объекта.

**Инкремент** (от англ. *«increment»*) — увеличение значения на заданное значение (как правило, на единицу).

**Инсайт**(от англ. *«insight»*) — внезапная догадка, озарение или прозрение. Зачастую приводит к обнаружению скрытой проблемы или внезапному нахождению её решения, или новому видению сложившейся ситуации.

**Инсталлировать**(от англ. *«install»*) — установить. Например, *«инсталлировать Винду»* — установить ОС Windows.

**Интегрированная cреда разработки** (сокр. ***«IDE»*** от англ. *«****I****ntegrated****D****evelopment****E****nvironment»*) — это комплекс программного обеспечения, которое содержит всё необходимое для разработки, компиляции, линкинга и отладки кода.

**Интерпретатор** — это программа, которая напрямую выполняет код без его предыдущей компиляции в исполняемый файл. Интерпретаторы более гибкие, но менее эффективные, так как процесс интерпретации выполняется повторно при каждом запуске программы.

**Инфа** — информация.

**Инфоцыган**— это человек, который продает информационные товары (например, курсы, вебинары, книги и пр.) сомнительного качества за деньги. Как правило, данный «эксперт» не обладает большим опытом в какой-либо сфере, кроме развода доверчивых людей. В маркетинге своих товаров очень часто применяет безосновательные лозунги типа *«Как заработать миллион рублей ничего не делая! Кликай сюда»* или *«Я расскажу 5 способов ежедневного заработка $1000, уделяя работе всего 2 часа в день!»* и т.д. Хотя данный персонаж и работает в сфере инфобизнеса, но не следует называть всех, кто занимается инфобизнесом, инфоцыганами, т.к. данный термин имеет негативный окрас и применятся преимущественно только к «разводилам».

**Исходник**— исходный код или исходный файл(ы).

**ИТ**— **И**нформационные **Т**ехнологии.

**Итератор** (англ. *«iterator»*) — это объект, разработанный специально для перебора элементов контейнера (например, значений массива или символов в строке), обеспечивающий во время перемещения по элементам доступ к каждому из них.

**Итерация** (от англ. *«iteration»*) — повторение какого-либо действия.

**ISP** (сокр. от *«****I****nternet****S****ervice****P****rovider»*) — провайдер.

**К**

**Камень** — процессор.

**Капча** (сокр. ***«CAPTCHA»*** от англ. *«****C****ompletely****A****utomated****P****ublic****T****uring test to tell****C****omputers and****H****umans****A****part»*) — картинка для проверки пользователя, является ли он настоящим человеком или это просто бот (программа). Капча используется в качестве защиты от автоматических регистраций и спам-ботов.

**Кардридер**(или *«картридер»*, от англ. *«card reader»*) — устройство для чтения карт памяти.

**Кастомер**(от англ. *«customer»*) — покупатель, клиент или заказчик.

**Кастомный**(от англ. *«custom»*) — переделанный под индивидуальные требования определенного пользователя, сделанный на заказ.

**Кейс**(от англ. *«case»*) — реальная ситуация (или случай), которая произошла с автором.

**Киберсквоттинг** — покупка или перехват на аукционе перспективных доменных имен для их последующей выгодной перепродажи (зачастую, в разы, если не в десятки раз, больше от первоначальной стоимости).

**Килобаксы**— тысячи долларов. Приставка *«к»* означает*«кило»,*т.е. тысяча. Например, *10к = 10 000*.

**Кирпич**— неработающее или сломанное устройство (например, неработающий телефон).

**Клава**— клавиатура.

[**Класс**](https://ravesli.com/urok-113-klassy-obekty-i-metody-klassov/) — это пользовательский тип данных, который может содержать не только данные (переменные), но и функции, работающие с этими данными (методы).

**Кликать** — нажимать на кнопки компьютерной мыши.

**Коворкинг**(от англ.*«coworking»*) — место для организации труда специалистов одной или сразу нескольких сфер деятельности, фрилансеров.

**Кодер**(или *«кодерок»*, от англ. *«coder»*)— программист.

**Кодить**— программировать.

**Код-ревью** (от англ. *«code review»*) — рецензирование/проверка кода с целью обнаружения и исправления ошибок, которые остались незамеченными при написании кода.

**Командная строка**—разновидность текстового интерфейса между человеком и компьютером, в котором инструкции компьютеру даются в основном путем ввода с клавиатуры текстовых строк.

**Коммитить**(или *«закоммитить», «сделать коммит*», от англ. *«commit»*) — сохранить/зафиксировать изменения кода в репозитории (хранилище) кода.

[**Компиляция**](https://ravesli.com/urok-1-vvedenie-v-programmirovanie/#toc-2) — конвертация кода, написанного на высокоуровневом языке, в исполняемую программу, которую процессор понимает напрямую.

**Компилятор**— это программа, которая читает код и создает автономную (способную работать независимо от другого аппаратного или программного обеспечения) исполняемую программу, которую процессор понимает напрямую. При запуске программы весь код компилируется целиком, а затем создается исполняемый файл, и уже при повторном запуске программы компиляция не выполняется.

**Компуктер**(или *«комп»*, *«компудахтер», «компутер»*) — компьютер.

**Коннект** (от англ. *«connect»*) — установленная связь между объектами.

**Константа** (англ. *«constant»*) — это переменная, значение которой нельзя изменить.

[**Конструктор**](https://ravesli.com/urok-116-konstruktory/) (англ. *«constructor»*) — это особый тип метода класса, который автоматически вызывается при создании объекта этого же класса. Конструкторы обычно используются для инициализации переменных-членов класса значениями, которые предоставлены по умолчанию/пользователем, или для выполнения любых шагов настройки, необходимых для используемого класса (например, открыть определенный файл или базу данных).

**Контент** (от англ. *«content»*) — содержимое или информация.

**Контра** (или ***«КС»***) — компьютерная игра Counter-Strike.

**Конфиг** (от англ. *«****config****uration»*) — конфигурационный файл ПО, в котором записаны настройки данного ПО.

**Конфа** — конференция.

**Копипаст** (от англ. *«copy/paste»*) — скопировать (*copy*) и вставить (*paste*).

**Костыль**— исправление серьезных багов/дыр/ошибок без должного исправления системы в целом, т.е. решение проблемы «на коленке» в кратчайшие сроки в ущерб эффективности и хорошим практикам программирования.

**Кракозябры**(или *«крякозябры»*) — набор непонятных символов, бессмыслица. Возникает в результате неправильной кодировки. Например, иногда кракозябры можно наблюдать на сайтах, которые имеют проблемы с кодировкой, выглядят они следующим образом: РџСЂРёРІРµС‚! РњРµРЅСЏ.

**Кресты**(или *«плюсы»*) — [**язык программирования С++**](https://ravesli.com/uroki-cpp/).

**Кринж**(от англ. *«cringe»*) — чувство неловкости и стыда как реакция на чей-нибудь совершенный (отвратительный) поступок. Впервые термин начал использоваться среди геймеров.

**Крипово** — страшно.

**Кроссплатформенный**— тот, который не зависит от аппаратных платформ и операционных систем (например, может работать как на Windows, так и на Linux).

**Кряк**(или *«таблетка»*, от англ. *«crack»*) — специальная программа или дополнение к ПО для взлома данного ПО. В основном используется для преобразования лицензионной (платной) версии чего-либо (например, программы или игры) в бесплатную.

**Крякнуть** (или *«крэкнуть»*) — взломать ПО (например, программу или игру).

**Куки** (или *«кукисы»*, от англ. *«cookies»*) — небольшие текстовые файлы, создаваемые браузером пользователя, которые используются для определения уникальности пользователя и для отслеживания его действий на сайте.

**Кулер** (от англ. *«cooler»*) — охлаждающее устройство (система охлаждения) для частей персонального компьютера или сервера.

**Кулхацкер** (или *«хацкер»*, от англ. *«cool hacker»*) — человек, считающий себя хакером или опытным программистом, не являющийся таковым на практике.

**Л**

**Лаг**(или *«лаги»*) — задержка в выполнении программы/игры, отсутствие реагирования на действия пользователя. Синоним к термину *«глюк»*.

**Лайфхак**(от англ. *«lifehack»*) — полезный совет, помогающий решить определенную проблему и сэкономить время и/или усилия.

**Ламер**(от англ. *«lamer»*) — неопытный человек, который не разбирается в определенной сфере, но думает, что разбирается. Не путать с *чайником*.

**Лапша** — телефонный кабель старого образца.

**Левел**(от англ. *«level»*) — уровень.

**Легаси-код** (от англ. *«legacy code»*) — устаревший чужой код, написанный очень давно, и с которым сейчас приходится иметь дело.

**Лейбл** (или *«лейба»*, от англ. *«label»*) — этикетка — графический/текстовый знак/наклейка на любом ИТ-продукте, с указанием торговой марки, названия, производителя, даты производства и т.п.

**Либа** (от англ. *«library»*) — библиотека.

**Ликбез**(сокр. от *«****лик****видация****без****грамотности»*) — обучение азам, основам.

**Линк**(от англ. *«link»*) — ссылка/гиперссылка на какой-либо ресурс (например, статью, сайт или страницу сайта).

**Линкануть** — отправить куда-нибудь (кому-нибудь) определенную ссылку.

**Линкинг** (от англ. *«linking»*) — это процесс связывания всех объектных файлов, генерируемых компилятором, в единую исполняемую программу, которую вы затем сможете запустить/выполнить. Это делается с помощью программы, которая называется **линкер** (или ***«компоновщик»***).

[**Линукс**](https://ravesli.com/chto-takoe-linux-ego-struktura-i-preimushhestva/)(или *«Линух»*, от англ. ***«Linux»***) — это семейство Unix-подобных операционных систем, использующих ядро Linux, которое разработал финно-американский программист Линус Торвальдс. Операционные системы, использующие ядро Linux, называются **дистрибутивами Linux**, и они являются такими же операционными системами, как Microsoft Windows, Apple macOS, iOS, но с одной очень важной особенностью, а именно — их исходные коды являются открытыми, так как они распространяются под [**лицензией GNU GPL**](https://ravesli.com/copyright-i-copyleft-litsenziya-gnu-gpl/), которая подразумевает создание бесплатного и открытого программного обеспечения (*open-source software*).

**Линуксоид**— человек, который использует дистрибутив Linux или ярый фанат всего, что касается темы «Linux».

[**Литерал**](https://ravesli.com/urok-36-literaly-magicheskie-chisla/) — это фиксированное значение, которое записывается непосредственно в исходном коде (например, 7 или 3.14159).

**Лицуха**— лицензия.

**Личка** — личные сообщения. Например, *«написать в личку в Инсте»* означает написать в личные сообщения в Instagram.

**Лог-файл**(от англ. *«log file»*) — текстовый файл, в котором записываются в хронологическом порядке все выполненные действия в программе или на сервере, например: время возникновения ошибок/сбоев, действия пользователей и пр.

**Логин** (от англ. *«login»*) — 1) процесс, с помощью которого пользователь получает доступ к чему-либо (например, к сайту или программе) путем идентификации и аутентификации; 2) уникальное имя пользователя для входа в систему.

[**Логический тип данных**](https://ravesli.com/urok-34-logicheskij-tip-dannyh-boolean/) (от англ. *«boolean»*) — это тип данных, переменные которого могут содержать лишь одно из двух возможный значений: true (1) или false (0).

**Лого** — логотип.

**Локалка**— это группа вычислительных устройств (компьютеры, периферия) вместе с активным оборудованием, соединенные между собой кабельной системой, а также беспроводными технологиями, и образующие структуру, покрывающую конкретную, как правило, небольшую территорию (например, одно помещение, здание или несколько зданий/сооружений, расположенных компактно).

[**Локальные переменные**](https://ravesli.com/urok-48-lokalnye-peremennye-oblast-vidimosti-i-vremya-zhizni/) — это переменные, которые определены внутри блока. Они создаются (и инициализируются, если необходимо) в точке определения и уничтожаются при выходе из блока.

**Локейшн**(от англ. *«location»*) — месторасположение.

**ЛОЛ** (или ***«LOL»***, от англ. *«****L****aughing****O****ut****L****oud»*) — англоязычный акроним, интернет-мем. Термин используется в сетевом общении главным образом для выражения смеха в письменной форме.

**Лузер** (от англ. *«loser»*) — неудачник или проигравший в игре.

**Лут** (от англ. *«loot»*) — 1) добыча; 2) различные предметы, ресурсы, оружие или трофеи в компьютерной игре, которые игрок получает с уничтоженных монстров или находит на просторах игрового мира (например, в сундуках).

**Лэптоп**(от англ. *«laptop»*) — ноутбук.

[**Лямбда-выражение**](https://ravesli.com/lyambda-vyrazheniya-anonimnye-funktsii-v-s/) (или просто *«лямбда»*) — позволяет определить анонимную функцию внутри другой функции. Возможность сделать функцию вложенной является очень важным преимуществом, так как позволяет избежать как захламления пространства имен лишними объектами, так и определить функцию как можно ближе к месту её первого использования.

**М**

**Маздай**(от англ. *«must die»*) — жаргонное, ироничное, пренебрежительное название семейства проприетарных операционных систем Windows корпорации Microsoft.

**Майнинг**(от англ. *«mining»*) — добыча какого-либо ресурса/предмета.

**Мамка**(или *«мать», «материнка», «мазерборд»*) — материнская плата.

**Манагер**(от англ. *«manager»*) — менеджер, руководитель.

**Мануал**(от англ.*«manual»*) — руководство, инструкция по использованию чего-либо.

[**Массив**](https://ravesli.com/urok-74-massivy-chast-1/) (англ. *«array»*) — это совокупный тип данных, который позволяет получить доступ ко всем переменным этого типа данных через использование одного идентификатора.

**Мейл** (от англ. *«E-mail»*) — электронное письмо/сообщение.

**Мейнстрим** (от англ. *«mainstream»*) — преобладающее направление в какой-либо области (научной, культурной, технической и др.) для определенного отрезка времени.

**Ментор**(от англ. *«mentor»*) — наставник, учитель.

**Мерч** (от англ. *«****merch****andise»*) — разного рода продукция, товары с определенной символикой или без. Очень часто данный термин используется для обозначения одежды, выпускаемой известными лицами, знаменитостями.

**Мидл**(от англ. *«****Middle****Developer»*) — средний уровень специалиста. Человек, имеющий «средний» опыт (как правило, от 2 и более лет) и навыки в определенной сфере деятельности.

**Митап**(от англ. *«meetup»*) — собрание/встреча специалистов определенной сферы деятельности для обмена опытом, в образовательных целях или просто для общения в неформальной обстановке.

**Митинг** (от англ. *«meeting»*) — собрание (встреча) людей.

**Моб** (от англ. *«mob»*) — 1) любой нестатичный объект в компьютерной игре. Также данным термином называют любых игровых персонажей/монстров, неуправляемых игроком; 2) толпа.

**Модератор** (англ. *«moderator»*) — пользователь, наделенный правами администрирования содержимого на сайте/форуме/ресурсе. Как правило, данный пользователь обладает меньшими полномочиями, чем администратор, но большими, чем обычный пользователь, и занимается модерацией (проверкой) комментариев/постов обычных пользователей.

**Моник** — монитор.

**Мониторить** — наблюдать за чем-либо, отслеживать какие-либо процессы.

**Мост** (или *«бридж»*, от англ. *«bridge»*) — сетевое устройство второго (канального) уровня сетевой модели OSI. Предназначен для объединения сегментов (подсетей) компьютерной сети в единую сеть.

**Мультикаст** (от англ. *«multicast»*) — передача данных сразу группе заинтересованных получателей.

**Мусорка** — корзина (папка) для удаленных файлов и папок.

**Мыло**— адрес электронной почты, E-mail пользователя.

**MVP**(от англ. *«****M****inimal****V****iable****P****roduct»*) — минимально жизнеспособный продукт. Тестовая версия программы или продукта, услуги, сервиса с минимальным набором функций (иногда даже одной), которая несет ценность для конечного потребителя. MVP создают для тестирования гипотез и проверки жизнеспособности задуманного продукта, насколько он будет ценным и востребованным на рынке.

**Н**

**Накрутка** — действия, направленные на увеличение показателей каких-либо счетчиков (например, лайков, репостов или подписчиков).

**Нативный**(от англ. *«native»*)— родной. Часто используют выражение *«нативный код»* — исходный код (не модифицированный позже), написанный разработчиками.

**Нетворкинг**(от англ. *«networking»*) — окружение специалистов определенной сферы деятельности, где, используя связи между этими людьми, можно легко и быстро решить определенные проблемы/вопросы или просто познакомиться с целью будущего сотрудничества.

**Ник** (от англ. *«****nick****name»*) — придуманный псевдоним, прозвище или кличка пользователя.

**Ноулайфер**(от англ. *«no life»*) — человек, который большинство своего времени проводит за компьютером/ноутбуком/смартфоном в виртуальном мире (Интернет/соц.сети/игры). Человек, не имеющий личной жизни, хобби или каких-либо других занятий в реальном мире.

**Ноут**— ноутбук.

**Нуб**(или *«нубчик», «нубас», «нубарь», «нубак»*, от англ *«newbie»*) — новичок или человек у которого ничего не получается, или он не знает, как правильно что-то сделать. Неопытный.

**NPC**(или *«НПСишник»,* от англ *«****N****on-****P****layer****C****haracter»*) — неигровой персонаж в играх, который не находится под контролем игрока, но с которым можно взаимодействовать.

**О**

**Область видимости переменной** — часть программы, в пределах которой можно использовать переменную.

[**Обработка исключений**](https://ravesli.com/urok-182-obrabotka-isklyuchenij/) — это механизм обработки ошибок, возникающих в функции. Если в функции возникает ошибка, с которой она не может справиться, то она может выбросить исключение. Это заставит ЦП перейти к ближайшему блоку кода, который обрабатывает исключения данного типа.

**Оверклокер** — человек, занимающийся всякого рода модернизацией персонального компьютера (часто из-за отсутствия денег).

**Оверклокинг** (от англ. *«overclocking»*) — разгон компьютера — процесс увеличения тактовой частоты (и напряжения) компонента компьютера сверх штатных режимов с целью увеличения скорости его работы.

**Овертайм**(от англ. *«overtime»*) — переработка, сверхурочные часы.

**Околоайтишник** — человек, который не является айтишником/программистом, но работает в сфере IT/программирования. Например, специалист по кадрам (HR), бухгалтер, финансист.

[**ООП**](https://ravesli.com/urok-112-vvedenie-v-oop/) (сокр. от *«****О****бъектно-****О****риентированное****П****рограммирование»*) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию [**наследования**](https://ravesli.com/urok-154-bazovoe-nasledovanie-v-c/).

**Опенсорс**(от англ. *«****open source****software»*) — программное обеспечение с открытым исходным кодом, которое может использовать/дополнять/изменять любой желающий. В большинстве случаев данное ПО является некоммерческим.

**Опенспейс**(от англ. *«open space»*) — офис открытого типа, в котором все (или большинство) сотрудники работают в едином относительно большом помещении без каких-либо перегородок/стен.

**Операнды** — это данные, с которыми работает выражение.

**Оперативка**— оперативная память. Она же ***«RAM»*** — *«****R****andom****A****ccess****M****emory»*, она же ***«ОЗУ»*** — *«****О****перативное****З****апоминающее****У****стройство»*.

**Оптика** — сетевой оптоволоконный кабель.

**ОС**(или *«операционка», «ось»,*сокр. ***«OS»*** от англ. *«****O****perating****S****ystem»*) — **О**перационная **С**истема — комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

**Откатить**— вернуть назад, сделать как было раньше.

**Отладка**— этап в разработке программного обеспечения, на котором выполняется поиск и исправление ошибок.

**Отладчик**(или *«дебаггер»*, от англ. *«debugger»*) — программа, с помощью которой выполняется отладка.

**Офис** — программное обеспечение MS Office или другое подобное ПО, обеспечивающее такой же или похожий функционал.

**Офсайт** — официальный сайт.

**Оффер** (или *«офер»*, *«джоб оффер»*, от англ. *«offer»*) — предложение о работе. Как правило, в нем указываются должность, должностные обязанности (или цели) и зарплата.

**Оффтоп**(или *«оффтопик»*, от англ.*«off topic»*) — любое сообщение не по теме.

**Очепятка**— опечатка, ошибка в слове.

**П**

**Парсить** (от англ. *«parse»*) — собирать (анализировать) определенные данные.

**Паскаль** — язык программирования Pascal.

**Патч**(от англ. *«patch»*) — дополнение или обновление, которое исправляет ошибки/баги/глюки. Применение патча к программе называется *«пропатчиванием»*.

**Паутина** — Интернет.

[**Перегрузка функций**](https://ravesli.com/urok-102-peregruzka-funktsij/) — это возможность определять несколько функций с одним и тем же именем, но с разными параметрами.

**Переполнение** (англ. *«overflow»*) — происходит, когда запрашиваемой памяти нет в наличии (вся память уже занята).

**Песочница** (англ. *«sandbox»*) — специальная область/среда для безопасного выполнения программы. Раздел *«Песочница»* на сайте/форуме обозначает категорию для записей новичков (новоприбывших).

**Пет-проект** (или *«pet-проект»*, от англ. *«pet-project»*) — личный (реже групповой) проект, создаваемый и продвигаемый с целью получения опыта или в качестве хобби. Редко, но бывает, что приносит своим создателям деньги (в отдельных случаях, даже перерастает в полноценный проект или компанию).

**Пингануть** (или *«пропинговать»*, от англ. *«ping»*) — проверить доступность определенного IP-адреса, сетевую связность.

**Пиратка** (от слова *«пиратский»*) — нелицензионное программное обеспечение или операционная система.

**Писюк** (от англ. *«PC»*) — персональный компьютер.

**Плюшки**— бонусы. Очень часто относятся к условиям работы: гибкий график, наличие спортзала, кухни, комнаты отдыха и пр.

**ПО**— **П**рограммное **О**беспечение.

**Побитовые операции**(англ. *«bitwise operations»*) — это манипулирование отдельными битами в пределах переменной. Сюда же входят битовые сдвиги и применение [**побитовых операторов**](https://ravesli.com/urok-45-pobitovye-operatory/).

**Подкаст**(от англ. *«podcast»*) — аудио- или видеоуроки, рассказы на определенную тему, чаще всего образовательного характера. Сейчас очень популярны подкасты-интервью или TED-подкасты.

**Попап** (от англ. *«pop-up»*) — всплывающее окно, зачастую рекламного характера.

**Пост**(от англ. *«post»*) — запись/статья/сообщение в социальной сети или на сайте.

[**Предварительное объявление**](https://ravesli.com/urok-19-predvaritelnoe-obyavlenie-i-prototip-funktsii/) — прототип какой-либо функции, помещенный в начале файла программы, который сообщает компилятору о существовании идентификатора ДО его фактического определения.

**Провайдер** — организация, обеспечивающая доступ в Интернет или к услугам мобильной связи.

**Прогер**— программист.

**Программа** (или *«прога»*, *«программулина»*) — набор инструкций, которые указывают компьютеру, что ему нужно делать.

**Продакшн** (или *«прод»*, от англ. *«production»*) — рабочая версия продукта (например, сайта, программы или приложения).

**Продолжительность жизни** (или *«время жизни»*) **переменной** — часть программы, в пределах которой переменная создается и уничтожается.

**Прокси-сервер** (от англ. *«proxy»*) — промежуточный сервер между пользователем и конечным сайтом/сервером. Прокси-серверы работают в качестве посредника (дополнительного звена), что обеспечивает дополнительную конфиденциальность и затрудняет возможность слежки за пользователем и посещаемыми им сайтами.

**Проприетарный**(от англ. *«proprietary»*) — частный, в составе чьей-либо собственности. Данный термин часто применяется к программному обеспечению, указывая, таким образом, на несвободное программное обеспечение, которое является частной собственностью авторов или правообладателей и не удовлетворяющее критериям свободного ПО. Простыми словами, *«проприетарное программное обеспечение»* = *«несвободное/лицензионное/платное программное обеспечение»*.

[**Пространство имен**](https://ravesli.com/urok-53-prostranstva-imen/) — это область кода, в которой гарантируется уникальность всех идентификаторов. По умолчанию, глобальные переменные и обычные функции определены в глобальном пространстве имен.

**Прототип функции** — состоит из типа возврата функции, её имени и параметров. Тело функции (основная часть) опускается. Прототип функции используется в предварительном объявлении.

**Профайлинг**(от англ. *«profiling»*) или **Профилирование** — сбор характеристик работы и поведения программы. Например, время выполнения отдельных фрагментов кода (обычно подпрограмм), количество верно предсказанных условных переходов и т.д.

**Профан**— человек, который не имеет опыта или не разбирается в определенной теме.

**Профит**(от англ. *«profit»*) — прибыль, выгода, польза.

**Прошивка**— встроенное программное обеспечение, работающее непосредственно на устройстве и управляющее им.

**Прошить** — изменить прошивку.

**Путон** (или *«Питон»*, *«Пайтон»*) — язык программирования Python.

**Пэхапэ** (или *«Пыха»*) — язык программирования PHP.

**Р**

**Разраб**— разработчик.

**Рандом** (или *«рандомный»*, от англ. *«random»*) — произвольный, случайный.

**Рассылка** — автоматизированное отправление письма(ем) сразу большому количеству пользователей. Очень часто рассылку используют в качестве спама, т.е. отправление ненужной (бесполезной) информации или рекламы.

**Ребзя**— ребята.

**Ребутнуть**(или *«ребутить»*, от англ. *«reboot»*) — перезагрузить. Например, *«ребутнуть Винду»* — перезагрузить ОС Windows.

**Регулярки** — [**регулярные выражения**](https://ravesli.com/uroki-po-regexp/).

**Редизайн**(от англ. *«redesign»*)— обновление или модернизация дизайна/интерфейса/внешней оболочки какого-либо продукта (например, сайта или программы). Иногда под этим термином подразумевается и внутреннее обновление продукта.

**Редирект** (от англ. *«redirect»*) — принудительное перенаправление пользователя с одного ресурса (например, сайта) на другой.

[**Рекурсивная функция**](https://ravesli.com/urok-107-rekursiya-chisla-fibonachchi-i-faktorial/) — это функция, которая вызывает сама себя.

**Рекурсия** (от англ. *«recursion»*) — ситуация, когда объект является частью самого себя.

**Релиз**(от англ. *«release»*) — выпуск готовой версии продукта (например, игры, программы или музыкального альбома), представление на всеобщее обозрение. Например, *«релиз игры»*— выпуск игры для старта продаж.

**Релокейт** (от англ. *«relocate»*) — это переезд в другой город или страну, связанный с получением предложения о работе от желаемой компании или в связи с изменением структуры внутри текущей компании.

**Репа** — репозиторий, хранилище данных.

**Респаун** (или сокр. *«респ»*, от англ. *«respawn»*) — перерождение. Данный термин используется в компьютерных играх для описания места, где игрок, моб (*см. определение*) или NPC (*см. определение*) возвращается к жизни, после того как был убит. Впервые данный термин был использован в игре Doom.

**Респект** (от англ. *«respect»*) — проявление уважения к чему-либо или кому-либо.

**Рефакторинг кода** (от англ. *«refactoring»*) — изменение исходного кода программы с целью упрощения и облегчения его понимания и дальнейшей поддержки. С помощью рефакторинга код становится чище, исправляются ошибки и «узкие места» в коде.

**Референс** (или *«рефка»*, *«реф»*, от англ. *«reference»*) — примерное представление того, что хотелось бы увидеть или «получить на выходе». Попытка объяснить собеседнику на чужих примерах свое представление.

**Рисёрч** (или *«ресерч»*, от англ. *«research»*) — исследование, изучение определенной темы. Когда вас просят **выполнить рисёрч**на определенную тему, то подразумевается, что вы должны разобраться с этой темой, изучить её суть, нюансы и детали.

**Рофл** (от англ. *«****R****olling****O****n the****F****loor****L****aughing»*) — кататься по полу от смеха.

**Рофлить** — подшучивать, издеваться над кем-то, ржать с кого-то или чего-то. Очень часто используется в негативном ключе.

**Рояль** — клавиатура.

**Рубиться** (или *«зарубиться»*) — увлеченно играть.

**Рунет**— русскоязычный Интернет, совокупность русскоязычных сайтов.

**Руткит**(от англ. *«rootkit»*) — это программа, которая скрывает от антивирусов собственные вредоносные действия, либо маскирует работу другого вредоносного ПО.

**Рэк** (от англ. *«rack»*) — 19-дюймовая стойка (шкаф) для ИТ-оборудования. 19 дюймов — стандартная ширина ИТ-оборудования для размещения в стойках серверных и коммутационных комнат.

**С**

**Саббатикал** (от англ. *«sabbatical»*) — творческий отпуск или карьерный тайм-аут для сотрудника с целью не допустить его «выгорания» (или ситуация, когда нужно справиться с существующим «выгоранием») и сохранить место на работе. Как правило, это неоплачиваемый отпуск от трех месяцев до года (может быть и больше), когда человек перестает заниматься определенной деятельностью и старается достичь жизненного баланса.

**Сабж**(или *«сабжект»*, от англ. *«subject»*) — тема для дискуссии между пользователями на определенном ресурсе (чаще всего, на форуме). Тема для начала разговора.

**Саппорт**(от англ. *«support»*) — служба поддержки.

**Свитч** (или *«свич»*, от англ. *«switch»*) — сетевой коммутатор — устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети.

**Свитчер**(от англ. *«switcher»*) — человек, который перешел работать в сферу IT или программирования из другой профессии. Например, если человек раньше работал учителем или поваром, а затем стал программистом, то его считают *свитчером*.

**Сейвить** (или *«засейвиться»*, от англ. *«save»*) — сохранить изменения.

**Секьюрный** (от англ.*«secure»*) — надежный, защищенный.

**Семантическая ошибка** — это ошибка, которая возникает, когда код является синтаксически правильным, но делает не то, что задумал программист.

**Сеньор**(или *«сеньор-помидор», «сеньйор», «сениор»*, от англ. *«****Senior****Developer»*) — высокий уровень специалиста. Человек, имеющий большой опыт (как правило, от 5 и более лет) и навыки в определенной сфере деятельности.

**Сервак** (или *«серв»*, *«сервант»*, *«стервер»*) — сервер.

**Сервис пак** (от англ. *«service pack»*) — набор обновлений, исправлений, улучшений компьютерной программы или ОС, поставляемый в виде единого установочного пакета.

**Сетевик** — сотрудник ИТ-подразделения, занимающийся поддержкой рабочего состояния ЛВС: активного оборудования сети и СКС (структурированной кабельной системы).

**Сетевуха** — сетевая карта (плата) в персональном компьютере (сервере).

**Сижка** — язык программирования Си.

**Сиквел** — SQL (язык SQL или сервер SQL).

**Синтаксис** — это правила, которые регулируют построение предложений.

**Синтаксическая ошибка** — это ошибка, которая возникает, когда вы пишете код, который не соответствует правилам грамматики языка программирования. Например, пропущенные точки с запятой, необъявленные переменные, непарные круглые или фигурные скобки и пр.

**Сионист** — программист, пишущий на языке Си.

**Сисадмин**— **сис**темный **админ**истратор — специалист по поддержанию сетевой (и вообще, компьютерной) инфраструктуры в компании.

**Системник**— системный блок.

**Скилл**(или *«скиллы», «скилы»*, от англ. *«skill»*) — навыки, умение хорошо что-то делать.

**Скин** (от англ. *«skin»*) — внешний вид, обложка, оболочка чего-либо, которая может варьироваться по дизайну. **Скины в играх** — это отличающийся от всех остальных внешний вид определенного персонажа или предмета.

**Скипнуть**(от англ. *«skip»*) — пропустить или пролистать. Например, можно *скипнуть* рекламу и сразу перейти к контенту.

**Скрин**(или *«скриншот»*,от англ. *«screenshot»*) — снимок экрана.

**Скрипт**(от англ. *«script»*) — небольшая программа для выполнения определенной задачи.

**Слить**— 1) выложить в открытый доступ «закрытые» файлы (например, слить фото знаменитостей); 2) проиграть в компьютерной игре (например, ***«слить катку»*** — проиграть игровой матч).

**Слоупок**— медленный, тормоз. Название произошло от розового покемона *Slowpoke*.

**Смузи**— любимый напиток выебщиков.

**Смузихлеб**— программист-выебщик.

**Сниппет**(от англ. *«snippet»*) — фрагмент чего-либо (например, кода).

**Сносить** (или *«снести»*) — удалить что-либо (например, программу).

**Собака** (или *«собачка»*) — символ @.

**Собес** — собеседование.

**Софт**(или *«софтина»*, от англ. *«****soft****ware»*) — программное обеспечение, программа.

**Софт скиллы**(от англ. *«soft skills»*) — социальные навыки и качества специалиста, необходимые, в первую очередь, для адаптации и работы в коллективе.

**Спам**(от англ. *«spam»*) — это нежелательные сообщения в любой форме, которые отправляются в большом количестве разным адресатам. Часто это электронные письма/сообщения рекламного и иного характера, отправляющиеся пользователям, которые не нуждаются в данной информации.

**Спамер** — создатель и распространитель спама.

**Спамить**— отправлять спам.

[**Спрайт**](https://ravesli.com/graficheskaya-biblioteka-sfml-sprajty-i-tekstury/#toc-0)(от англ. *«sprite»*) — графический объект в компьютерной графике.

**Спс** (или ***«sps»***) — спасибо.

**Статическая библиотека** — файл, содержащий набор подпрограмм, которые непосредственно компилируются и линкуются с вашей программой. При компиляции программы, которая использует статическую библиотеку, весь функционал статической библиотеки (тот, который использует ваша программа) становится частью вашего исполняемого файла. В Windows статические библиотеки имеют расширение *.lib* (сокр. от *«****lib****rary»*), тогда как в Linux статические библиотеки имеют расширение *.a* (сокр. от *«****a****rchive»*).

[**Статическая переменная**](https://ravesli.com/urok-51-staticheskie-peremennye/) (или *«переменная со статической продолжительностью жизни»*) — это переменная, которая сохраняет свое значение даже после выхода из блока, в котором она определена. То есть она создается (и инициализируется) только один раз, а затем сохраняется на протяжении выполнения всей программы.

**Cтейтмент** (от англ. *«statement»*) — это наименьшая независимая единица в языке программирования, которая является наиболее распространенным типом инструкций в программах. Стейтмент в программировании — это то же самое, что и «предложение» в русском языке. Мы пишем предложения, чтобы выразить какую-то идею. В языке программирования мы пишем стейтменты, чтобы выполнить какое-то задание.

[**Стек**](https://ravesli.com/urok-105-stek-i-kucha/#toc-3) (от англ. *«stack»*) — это структура данных типа **LIFO** (сокр. от *«****L****ast****I****n,****F****irst****O****ut»* — *«Последним пришел, первым ушел»*). Последний элемент, который находится на вершине стека, первым и уйдет из него. По мере того, как элементы помещаются в стек — стек растет, по мере того, как элементы удаляются из стека — стек уменьшается.

[**Стек вызовов**](https://ravesli.com/urok-105-stek-i-kucha/#toc-2) — отслеживает все активные функции (те, которые были вызваны, но еще не завершены) от начала программы и до текущей точки выполнения, и обрабатывает выделение всех параметров функции и локальных переменных.

**Стек технологий** — используется для обозначения списка технологий, с которыми работает разработчик или компания.

**Стрим**(от англ. *«stream»*) — трансляция действий, происходящих на компьютере (или в реальной жизни) в режиме реального времени.

**Стример** — тот, кто делает стрим.

**Стримить** — делать стрим.

[**Структура**](https://ravesli.com/urok-61-struktury/) (от англ. *«structure»*) — это пользовательский тип данных, который позволяет сгруппировать переменные различных типов в единое целое.

[**Scrum**](https://ru.wikipedia.org/wiki/SCRUM)— это Agile-методология управления проектами, которая используется в сфере разработки ПО. Отличительной особенностью является итеративность процесса разработки с наличием т.н. «спринтов» — небольших временных промежутков, за которые команда разработчиков предоставляет конечному пользователю (заказчику) результат своей работы над проектом с возможностью согласования внесения изменений непосредственно в процессе разработки.

**Т**

**Таска** (от англ. *«task»*) — задание.

**Тачпад** (от англ. *«touchpad»*) — сенсорная панель устройства. Это может быть как экран смартфона, так и сенсорная панель ноутбука.

**Текстура** (от англ. *«texture»*) — это особый тип изображения, накладываемый на объекты для создания иллюзии чрезвычайно детализированного объекта.

**Тестировщик** (сокр. ***«QA»*** от англ. *«****Q****uality****A****ssurance»*) — специалист по тестированию программного обеспечения.

**Технарь**— человек, имеющий специальное техническое образование или сотрудник, который работает в бэкенде (*бэкенд-разработчик*).

**Тимлид**(или *«тимлидер»*, от англ. *«****team lead****er»*) — ведущий разработчик, руководитель команды программистов.

[**Тип данных с плавающей точкой**](https://ravesli.com/urok-33-tip-dannyh-s-plavayushhej-tochkoj-floating-point/) (или *«тип данных с плавающей запятой»*, англ. *«floating point»*) — это тип данных, переменные которого могут содержать любые действительные дробные значения, например: 4320.0, -3.33 или 0.01226. Почему точка «плавающая»? Дело в том, что точка/запятая перемещается («плавает») между цифрами, разделяя целую и дробную части значения.

**Точка останова** (англ. *«breakpoint»*) — это специальный маркер, на котором отладчик останавливает процесс выполнения программы.

**Транслитерация** (или ***«транслит»***) — передача текста с одного языка на другой при помощи другого алфавита (иногда с использованием цифр и спецсимволов). Например, слово *«привет»* транслитом можно написать как *«pr1vet»*.

**Трейни**(от англ. *«trainee»*) — стажер. Сотрудник, который не дотягивает до уровня Junior-специалиста.

**Трекать**(от англ. *«track»*) — что-либо отслеживать. Например, *трекать время* = отслеживать рабочее время, *трекать задачу* = отслеживать ход работы над задачей.

**Троллинг** (или *«троллить»*, от англ. *«trolling»*) — писать провокационные посты, комментарии, пытаясь вызвать конфликт или разозлить участников беседы.

**Тролль** (от англ. *«troll»*) — анонимный Интернет-провокатор, который намеренно публикует провокационные статьи и/или сообщения для возникновения конфликта между пользователями или участниками беседы.

**Троян** — вредоносная компьютерная программа, которая используется для заражения системы целевого компьютера, и приводит к вредоносной активности на нем. Трояны могут удалять файлы или уничтожать информацию на жестком диске.

**ТС**(или *«****т****опик****с****тартер»*, от англ. *«topic starter»*) — человек, который начал разговор/дискуссию.

**ТСП** (сокр. ***«TCP»*** от англ. *«****T****ransmission****C****ontrol****P****rotocol»*) — «туда-сюда» протокол. Один из основных протоколов передачи данных в интернете и ЛВС. Предназначен для управления передачей данных.

**Тулза**(от англ. *«tools»*) — вспомогательное программное обеспечение, предназначенное для выполнения специфических задач.

**Туториал**(от англ. *«tutorial»*) — руководство, учебник, уроки по определенной теме.

**Тян**— подруга, девушка.

**У**

**Угарать** (или *«угорать»*) — смеяться (или надсмехаться) над чем-либо или кем-либо. Часто используется в негативном ключе.

**Удалёнка** — удалённая работа или работа из дома — вид дистанционной работы, когда от сотрудника не требуется каждодневное физическое присутствие в офисе.

[**Указатель**](https://ravesli.com/urok-80-ukazateli-vvedenie/) (англ. *«pointer»*) — это переменная, значением которой является адрес ячейки памяти.

[**Умный указатель**](https://ravesli.com/urok-189-umnye-ukazateli-i-semantika-peremeshheniya/) (англ. *«smart pointer»*) — это класс, предназначенный для управления динамически выделенной памятью и обеспечения освобождения (удаления) выделенной памяти при выходе объекта этого класса из области видимости.

**Упс** (сокр. ***«UPS»*** от англ. *«****U****ninterruptible****P****ower****S****upply»*) — источник (устройство) бесперебойного питания электроэнергией.

**Утечка памяти** — происходит, когда ваша программа теряет адрес некоторой динамически выделенной части памяти (например, переменной или массива), прежде чем вернуть её обратно в операционную систему. Когда это происходит, то программа уже не может удалить эту динамически выделенную память, поскольку больше не знает, где выделенная память находится. Операционная система также не может использовать эту память, поскольку считается, что она по-прежнему используется вашей программой.

**Утилита**(или *«утиль»*, *«утилитка»*, от англ. *«utility»*) — вспомогательная программа, предназначенная для выполнения определенной задачи.

**Уши** — наушники.

**Ф**

**Фаервол**(или *«файрвол»*,*«межсетевой экран»*, от англ. *«firewall»*) — это программа, основной задачей которой является защита сегментов сети или отдельных хостов от несанкционированного доступа с использованием уязвимых мест в протоколах сетевой модели OSI или в программном обеспечении, установленном на компьютерах сети. Межсетевые экраны пропускают или запрещают трафик, сравнивая его характеристики с заданными шаблонами. Другое название — **Brandmauer** (или ***«Брандмауэр»***, в переводе с немецкого означает *«стена, защищающая от пожара»*). Это не то же самое, что антивирус.

**Фак** (сокр. ***«FAQ»*** от англ. *«****F****requently****A****sked****Q****uestions»*) — часто задаваемые вопросы.

**Факап**(от англ. *«fuck up»*) — неудача, провал, облом.

**Фейк** (от англ. *«fake»*) — дезинформация, поддельная, лживая информация.

**Фидбек**(от англ. *«feedback»*) — обратная связь.

**Фиксить**(или *«пофиксить»*, от англ. *«fix»*) — исправлять ошибки.

**Фича**(от англ. *«feature»*) — особенность, уникальное свойство. Существует популярное выражение — *«Это не баг, а фича»*.

**Флейм**(или *«флэйминг»*, от англ. *«flame»*) — спор ради спора, когда пользователи начинают общение не по теме разговора/дискуссии и переходят на личностные и взаимные оскорбления.

**Флеха**(или *«флэха», «флешка»*) — запоминающее устройство, подключаемое к компьютеру по USB-интерфейсу.

**Флуд**(от англ. *«flood»*) — сообщения/комментарии не по теме.

**Флудить** — оставлять нетематические сообщения/комментарии.

**Формошлеп**— человек, который занимается формошлепством. Очень часто формошлепами называют тех, кто занимается версткой шаблонов сайтов.

**Формошлепство**— любая низкоквалифицированная работа в программировании для выполнения которой не требуется много ума.

**Фреймворк**(от англ. *«framework»*) — программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов.

**Фриланс**(от англ. *«freelance»*) — ненормированная работа на себя или работа со сменяемыми заказчиками. Как правило, фрилансом можно заниматься дома, в коворкингах или в любом другом месте, где есть Интернет.

**Фрилансер**(от англ. *«freelancer»*) — свободный работник, который работает не по найму и не обязательно привязан к конкретному работодателю в долгосрочной перспективе. Человек, занимающийся фрилансом.

**Фронтенд**(от англ. *«front-end»*) — разработка «внешнего вида» (интерфейса) продукта (например, сайта или программы).

**Фулстек-разработчик** (от англ. *«full stack developer»*) — специалист, у которого есть опыт и навыки разработки одновременно во фронтенде и бэкенде. Он принимает участие как в создании визуальной части создаваемого продукта (фронтенда), так и в реализации серверной (бэкенда).

**Функция** (от англ. *«function»*) — это последовательность стейтментов для выполнения определенного задания.

**Фэйл** (от англ. *«fail»*) — облом, провал, неудача, совершенная ошибка.

**Х**

**Хаб** (от англ. *«hub»*) — класс устройств для объединения компьютеров в сетях Ethernet с применением кабельной инфраструктуры типа витая пара. В настоящее время вытеснены сетевыми коммутаторами (свитчами).

**Хакатон** (от англ. *«****hack****mar****athon****»*) — мероприятие, на которое на несколько дней (или день) собираются специалисты из разных отраслей разработки программного обеспечения (чаще всего программисты, но могут быть и дизайнеры, тестеры) для разработки определенного продукта или решения проблемы.

**Хакнуть** — взломать.

**Хантить** (от англ. *«hunt»*) — на сленге HR-отдела означает охотиться (искать) за нужным человеком для закрытия вакантной позиции.

**Хард скиллы**(от англ.*«hard skills»*) — технические знания и навыки специалиста в определенной сфере деятельности.

**Холивар** (от англ. *«holy war»*) — споры между людьми, которые имеют противоположные взгляды и пытаются навязать их друг другу.

**Хостить** — 1) предоставлять ресурсы для использования; 2) услуга хранения и отгрузки вашего сайта посетителям.

**Хрюша**(сокр. *«****HR****»*от англ. *«****H****uman****R****esources»*) — человек, который занимается работой с персоналом в компании. Очень часто в небольших компаниях *хрюша* еще занимается и поиском кадров, т.е. рекрутингом.

**Ц**

[**Целочисленный тип данных**](https://ravesli.com/urok-31-tselochislennyj-tip-dannyh-integer/) — это тип, переменные которого могут содержать только целые числа (без дробной части, например: -2, -1, 0, 1, 2).

**Цикл** — это конструкция, которая заставляет программу многократно выполнять определенное количество стейтментов до тех пор, пока заданное условие не станет ложным (или наоборот).

**ЦП** — центральный процессор (он же *«проц»* или *«процик»*).

**Ч**

**Чайник** — новичок, неопытный пользователь.

**Чекать** (или *«прочекать»*, от англ. *«check»*) — что-то проверить, изучить.

**Чекиниться**(или *«зачекиниться»*, от англ. *«check-in»*) — отметиться, что вы были в каком-либо месте (например, на мероприятии или конференции).

**Чек-лист**(от англ. *«checklist»*) — контрольный список вещей, дел или задач, которые нужно выполнить или проверить.

**Чит-код** (от англ. *«cheat code»*) — последовательность букв, цифр или нажатий клавиш, которая используется в компьютерных играх для добавления чего-либо полезного, но при этом не документирована.

**Читерство** (или *«читинг»*, от англ. *«cheating»*) — практика получения нечестного преимущества в многопользовательских компьютерных играх с помощью внешних программ и нестандартного аппаратного обеспечения. Впоследствии термин перебрался на настольные и азартные игры.

**Читкодер** (или *«читер»*) — тот, кто использует чит-коды втайне от других играющих для получения преимущества в компьютерной игре.

**ЧСВ** — **Ч**увство **С**обственной **В**ажности.

**Ш**

**Шаблон** — многократно используемый (типовой) кусок кода, позволяющий написать одну версию функции или класса, которая будет работать с разными типами данных. Функция или класс, реализованная через шаблон с фактическим (одним) типом данных, называется **экземпляром**.

[**Шаблоны функций**](https://ravesli.com/urok-173-shablony-funktsij/) — это функции, которые служат образцом для создания других подобных функций. Главная идея — создание функций без указания точного типа(ов) некоторых или всех переменных.

**Шарить**— понимать, разбираться в чем-либо.

**Шарпы** — язык программирования C#.

[**Шейдер**](https://ravesli.com/urok-5-shejdery-v-opengl/) (от англ. *«shader»*) — это небольшая программа, предназначенная для исполнения процессорами видеокарты.

**Шерить** (или *«расшарить»*, от англ. *«share»*) — делиться информацией, открыть доступ к информации.

**Шильдик** (от нем. *«schild»*) — информационная табличка, на которой размещены надписи и обозначения, относящиеся к маркируемому изделию (как правило, это модель и/или основные технические характеристики ИТ-техники).

**Э**

**Эксель** — программа MS Excel для работы с электронными таблицами.

**Эксплойт**(от англ. *«exploit»*) — вредоносная программа, способная выполнять код (хакера) через найденную уязвимость в ПО.

**Эникей**(от англ. *«any key»*) — любая кнопка. Очень часто, при установке или загрузке чего-либо, нас просят *«Press any key»* — отсюда и термин.

**Эникейщик**— неопытный пользователь, человек, который не знает, что ему нужно делать — ищет кнопку «эникей». Как правило, данный термин относится к низкоквалифицированным IT-специалистам, которые занимаются рутинными, несложными заданиями.

**Ю**

**Юзабилити** (от англ. *«usability»*) — уровень удобства использования определенного продукта или интерфейса (например, программы или сайта) для его пользователей.

**Юзать**(от англ. *«use»*) — использовать/пользоваться.

**Юзер**(или *«юзверь»*,от англ. *«user»*) — пользователь.

**Я**

**Яблочник** — пользователь устройств компании Apple или ярый представитель когорты фанатов компании Apple.

**Ява** — язык программирования Java.

**Яваскрипт** — язык программирования JavaScript.

**ЯП** — **Я**зык **П**рограммирования.

<https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate_block_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3>

**Что такое компоненты?**

**Компоненты** являются автономными независимыми частями приложения, которые отвечают за обработку только одной задачи и никак не влияют на работу друг друга.

Концепция компонентного подхода во фронтенде открывает невероятные возможности для повторного использования единожды написанного кода. Например, вы создаёте компонент для одного проекта, а потом используете его во всех остальных, без всякого переписывания или рефакторинга\*.

*\*Рефакторинг* — это контролируемый процесс улучшения кода без написания новой функциональности

**Что входит в понятие веб-компонентов?**

* Пользовательские элементы ([Custom Elements](https://developer.mozilla.org/ru/docs/Web/Web_Components/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D1%85_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2)) — *JavaScript API* для связывания вашего тега с соответствующим ему классом. https://developer.mozilla.org/ru/docs/Web/Web\_Components/Using\_custom\_elements
* Теневой*DOM* ([Shadow DOM](https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM)) — способ инкапсуляции внутреннего содержания тега.
* https://developer.mozilla.org/en-US/docs/Web/Web\_Components/Using\_shadow\_DOM
* *HTML*-шаблоны ([HTML templates](https://developer.mozilla.org/ru/docs/Web/HTML/Element/template)) — специальный тег <template>, предназначенный для хранения шаблонов. https://developer.mozilla.org/ru/docs/Web/HTML/Element/template

Все современные популярные фреймворки базируются на компонентах. В [React](https://proglib.io/p/react-developer-2019/), в [Vue](https://proglib.io/p/5-vue-js-examples/), в [Angular](https://proglib.io/p/learn-angular-2-other-useful/) разработчики занимаются тем, что делят своё приложение на небольшие части и работают с ними как с самостоятельными единицами.

[React](https://proglib.io/p/react-developer-2019/) https://proglib.io/p/react-developer-2019

**При чём здесь ES6?**

С es2015 и далее ведётся разработка стандартов для работы с модульным кодом*JavaScript* в браузере. Благодаря ему у нас появилась возможность разделять проект на самостоятельные модули, инкапсулирующие некоторую функциональность, и подключать их отдельно.

**Инкапсуляция** — это понятие, которое описывает один из принципов ООП (объектно-ориентированного программирования). По сути, инкапсуляция — это свойство, описывающее данные и методы, работающие в классе/функции и закрытые в нём. Они недоступны для изменения и перезаписи снаружи.

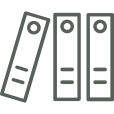
Когда говорят, что «переменная инкапсулирована внутри функции», это значит, что работать с ней мы можем либо внутри этой функции, либо в функции-наследнике.

Замыкания в *JavaScript* — это инкапсуляция.

**Резюмируем**

Компоненты значительно облегчают жизнь разработчикам, обеспечивают реализацию многих важных принципов, например, инкапсуляцию.

Чем сильнее мы будем полагаться на компоненты, разрабатывая архитектуру проекта, тем меньше будет становиться разрыв между дизайнером и разработчиком, тем проще масштабировать разработку, тем ниже порог вхождения новых разработчиков в ваше приложение.

**   Дополнительно**

Для закрепления изученного рекомендуем вам ознакомиться со следующими материалами:

1. Перевод статьи Брэда Фроста (Brad Frost) [«Atomic Web Design»](https://habr.com/ru/post/249223/) о системной разработке. https://habr.com/ru/post/249223/

2.  Библиотека компонентов [Ant-design](https://github.com/SkillfactoryCoding/ant-design). https://github.com/SkillfactoryCoding/PHP-React-ant-design

**15.3 Модули**

 Добавить страницу в мои закладки

В данном разделе вы познакомитесь с **модулем** — ещё одним фундаментальным понятием в *JavaScript*. Это знания, которые необходимы для успешной работы с любым фреймворком, особенно если в разработке используется компонентный подход. Текущий раздел не случайно оторван от раздела *JavaScript* — так вы сможете сразу применить знания на практике.

Каждый компонент принято инкапсулировать и помещать код в отдельный файл (модуль), а затем переиспользовать. Изучение этого материала поможет вам в дальнейшем не отвлекаться на детали вроде поиска способа подключить написанный компонент в другой файл.

**Модуль — привычный способ структурировать код**

<https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate_block_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3>

**Модуль** — шаблон, который базируется на внешней функции. У этой функции могут быть внутренние переменные и другие функции. Эта функция должна возвращать *API* с методами, которые должны замыкаться на внутренних данных.

В стандарте *ES*2015 модули получили первоклассную синтаксическую поддержку — теперь нам не нужно опираться на внешнюю функцию и замыкания.

**Основные особенности:**

1. Модули стали на основе файлов — **один модуль = один файл**. Это говорит о том, что в браузере потребуется загружать не один модуль, а несколько.
2. Модули имеют **статический** *API* — заранее определяются все виды экспорта верхнего уровня на открытом *API* модуля.
3. Каждый модуль — это**синглтон-объект**, который гарантирует, что в однопоточном приложении будет единственный экземпляр некоторого класса, предоставляющий глобальную точку доступа к этому экземпляру.
4. Импорт модуля означает запрос на его загрузку.
5. **Важно:***import* и *export* всегда должны быть на верхнем уровне видимости. Мы не можем вкладывать их в циклы, условные операторы и так далее.
6. **Export модулей**
7. Ключевое слово *export*или помещается перед объявлением того, что вы хотите экспортировать, или функционирует как оператор со списком значений.

|  |  |
| --- | --- |
| **Именованный экспорт**— вы передаете привязку на определённую структуру, и если она изменится, то при импорте у вас будет обновлённое значение, даже если вы не перекомпилировали объект. | **Дефолтный экспорт**— экспорт по умолчанию; содержится только в одном экземпляре, что призвано упростить синтаксис. |
| Например, вы можете экспортировать функцию:  **export** **function** useMe () { … }  **export** **const** links = [...]  **const** names = [...]  **export** { names }  Или изменить имя передаваемого объекта:  **function** useMe () { … }  **const** links = [...]  **const** names = [...]  **export** { useMe, links, names as lastNames } | Создаётся конкретная привязка к определённому объекту, который выбирается как дефолтный при импорте:  **function** useMe () { … }  **export** **default** useMe ;  Всегда будет привязка к функции *useMe*, даже если значение переменной будет переопределено:  **function** useMe () { … }  **export** {useMe as **default**};  Здесь привязка к идентификатору — если useMe изменится, то и используемое импортируемое значение тоже. |
|  |  |

1. **Import модулей**
2. Импортировать можно так же несколькими способами:
3. **import { useName } from ‘module’** — здесь фигурные скобки не являются деструктуризацией или чем-то ещё — таков синтаксис для импорта модулей.
4. Имя импортированного модуля должно совпадать с экспортируемым, но вы можете его переименовать:
5. **import { useName as name } from ‘module’**
6. Обратите внимание, что в случае дефолтного экспорта, импортируем без скобок:
7. **export** **default** useMe ;
8. **import** useMe from 'module';
9. Можно экспортировать всё пространство имён из модуля целиком, например:
10. **import** \* as mainModule from 'module';
11. mainModule.useMe()
12. mainModule.names;

**Для самостоятельного изучения**

1. Для закрепления материала рекомендуем вам изучить тему **циклической зависимости**:

Иногда в разработке возникает ситуация, когда один модуль экспортирует другой и наоборот. Это называется **циклической зависимостью**. Намеренно, конечно, никто так не делает, но, исходя из опыта, такое иногда возможно, и это может решить некоторые проблемы.

До*ES*6, когда модули не были разбиты на файлы, все функции лежали в одном файле и имели одну область видимости. Их объявления поднимались, и становилось возможным без проблем вызывать одну функцию в другой. Сейчас ситуация сложнее и, если такая потребность возникает, необходимо решать циклические зависимости правильно, а именно — следить за корректностью импортов.

**ВАЖНО:**

* Когда начинает загружаться, например, модуль А, который импортирует модуль Б, движок, когда встречает импорт Б, начинает его загружать.
* Когда модуль Б загрузился, движок инициализирует его импорты и подтягивает необходимые данные.
* Модули могут вызывать друг друга.

Поэтому такая возможность существует —*JavaScript* гарантирует загрузку и компиляцию модулей до вызова каждого из них.

2. Дополнительные материалы по модулям:

Статья [ES modules: A cartoon deep-dive](https://hacks.mozilla.org/2018/03/es-modules-a-cartoon-deep-dive/). При необходимости — [её перевод](https://medium.com/web-standards/es-modules-cartoon-dive-71f42c1e851a).

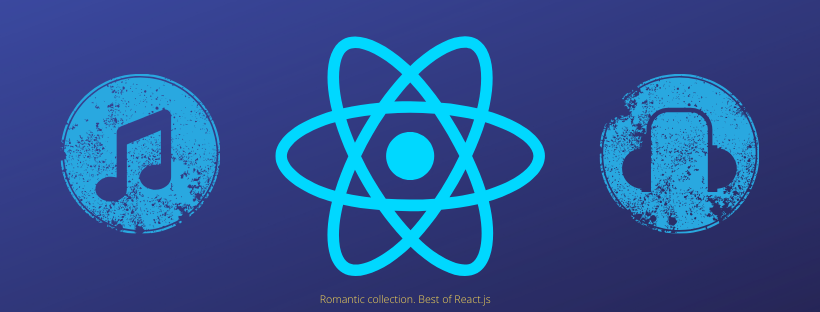
<https://medium.com/web-standards/es-modules-cartoon-dive-71f42c1e851a>

**15.4 Пишем первый компонент React**

<https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate_block_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3>

**Почему React.js?**

Прослушайте [мини-подкаст](https://cms.skillfactory.ru/assets/courseware/v1/d16e412fd5594954c31784a19912db2e/asset-v1:SkillFactory+FR-1+30_SEPT+type@asset+block/%D0%A1%D0%BA%D1%80%D0%B8%D0%BD%D0%BA%D0%B0%D1%81%D1%82_7._%D0%9F%D0%BE%D1%87%D0%B5%D0%BC%D1%83_React.mp3) о преимуществах *React.js*.



 Список преимуществ:

* Формат *jsx*. Он поддерживает специальный удобный синтаксис, который очень похож на *html*. Так же он компилируется, что очень может помочь в разработке.
* Виртуальный *DOM.*
* Код *React* можно рендерить как на клиенте, так и на сервере.
* Понятные механизмы и паттерны самого *React.*
* Развитая экосистема.
* Комьюнити.
* *React*— это *Open Source*.

**Компонент** — строительный материал всех *React*-приложений, некая самостоятельная часть, которая может быть использована как самостоятельная единица или в составе более крупного компонента.

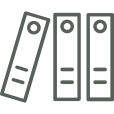
Объединение простых компонентов в сложный называется **композицией**.



Пишем первый компонент:

1. Выбрать *React.*
2. Пишем функциональный компонент.
3. Чтобы подключить и отобразить его, нужно вставить компонент на *html-*страницу. Воспользуемся функцией для работы с *DOM*. Чуть позже мы рассмотрим, что это.

[Здесь](https://jsfiddle.net/boilerplate/react-jsx) вы найдёте альтернативный пример компонента.

**   Дополнительно**

1. [Thinking in React.](https://reactjs.org/docs/thinking-in-react.html)

https://reactjs.org/docs/thinking-in-react.html

1. [Code structure from redux.](https://redux.js.org/faq/code-structure)
2. https://redux.js.org/faq/code-structure
3. [Умные и глупые компоненты](https://habr.com/ru/post/266559/).
4. https://habr.com/ru/post/266559/

## 15.5 Что такое JSX?

## ****https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate\_block\_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3****

## img****Дополнительно****

[JSX — подробности](https://habr.com/ru/post/319270/)

<https://habr.com/ru/post/319270/>

## 15.6 Понятие Virtual DOM

https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate\_block\_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3

## img****Дополнительно****

[React Fiber — продвинутый уровень](https://medium.com/@uncleseneca/%D0%BF%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9-%D0%BE%D0%B1%D0%B7%D0%BE%D1%80-react-fiber-66485d12bd37)

<https://uncleseneca.medium.com/%D0%BF%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9-%D0%BE%D0%B1%D0%B7%D0%BE%D1%80-react-fiber-66485d12bd37>

## 15.7 Быстрый старт React-приложения

## https://lms.skillfactory.ru/courses/course-v1:Skillfactory+FR+2020/courseware/7e39ad2d565349749da6fad7fc8e69e2/e4edcbd805b04d678dc964f516c1b419/2?activate\_block\_id=block-v1%3ASkillfactory%2BFR%2B2020%2Btype%40vertical%2Bblock%40c6b1b9133a7f46a883858acb1fa00bd3

Когда вы хотите подключить *React* на свой собственный сайт, разрабатываемый не онлайн, вы можете использовать несколько вариантов:

**Первый способ**

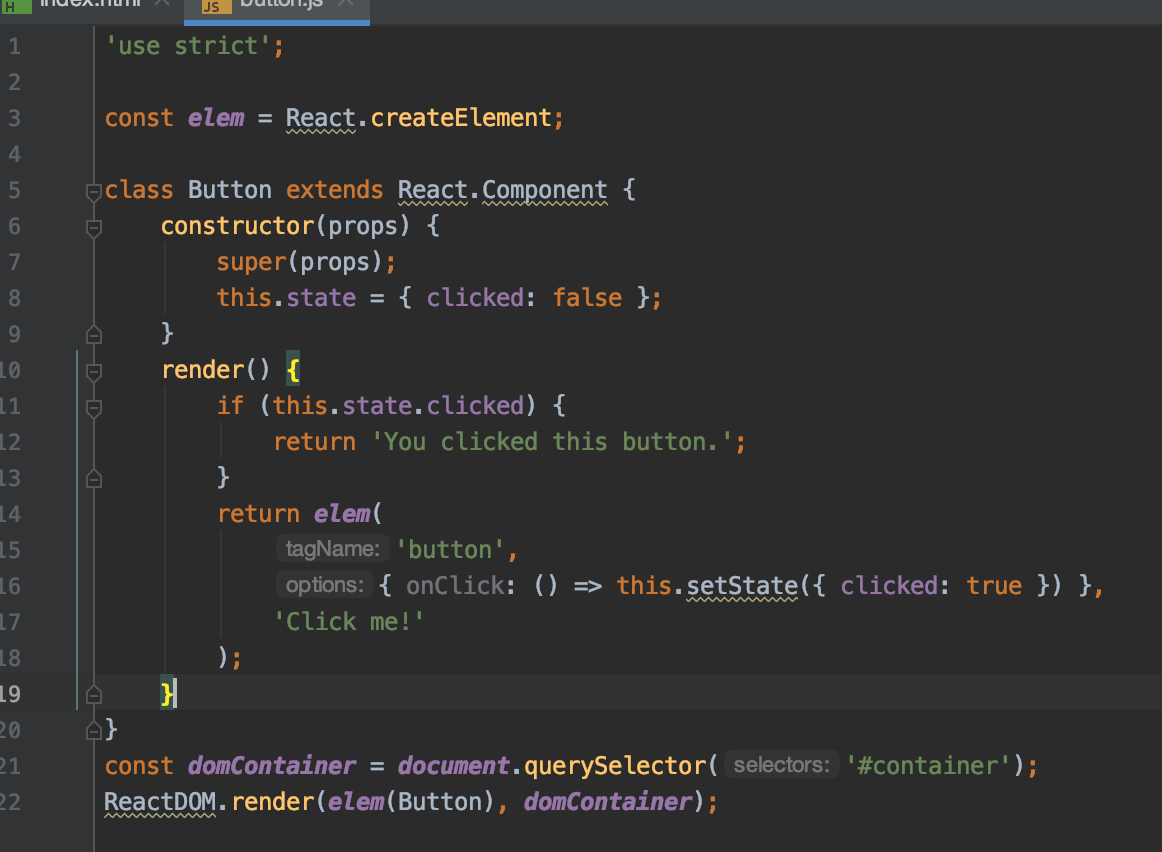
Подключите *React* как библиотеку в ваш *html* через скрипт. Можно писать как в формате*JSX*, так и в привычном *JavaScript.*

1. Определите, куда вы хотите вставить *React-*элемент, создайте некий контейнер в теге *body*, он должен быть пустым.
2. Подключите два скрипта, подгружающих *React:*

<script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>

<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>

3. Запишите скрипт с нашим компонентом:



4. Добавьте скрипт на страницу:



**Второй способ**

Мы с вами будем использовать *npm* пакет **create-react-app.**В нём «из коробки» идёт трансапайлер *babel*, который будет превращать *JSX* в *JS*. Также сборщик проекта *webpack* (он позволяет нам безболезненно использовать все прелести модульного подхода) и многие другие необходимые настройки, в том числе статический сервер *webpack-dev-server.*

1. Перейдите по [ссылке.](https://github.com/facebook/create-react-app)

https://github.com/facebook/create-react-app

2. Установите через *npm i create-react-app.*

3. Запустите приложение *create-react-app app.*

4. Откройте сгенерированное приложение в *IDE*.

** Практика**

1. Настройте окружение через*create-react-app* ([ссылка](https://github.com/facebook/create-react-app) на репозиторий *cra*, где вы можете посмотреть команды и дополнительную информацию).

https://github.com/facebook/create-react-app

1. Изучите *package.json* — какие зависимости установлены, какие скрипты есть.