



# Резюме

Резюме — это целый проект, который может хорошо вам послужить, если вы над ним как следует поработаете. Грамотно составленное резюме позволит привлечь внимание даже самого взыскательного работодателя. Особенно это актуально, когда есть только учебный опыт. В таком случае привлечь внимание можете только вы лично и базовый багаж знаний — результат обучения. И даже если релевантный вакансии опыт уже есть, плохое резюме может просто не понравиться представителям компании, и с вами не свяжутся. Поэтому всё самое релевантное о себе и о полученных знаниях нужно уложить в резюме, да ещё кратко и понятно. Этому и научит вас наша методичка.

1. Введение
2. Как писать резюме: общие моменты
3. Структура: что и где писать
4. Финальные штрихи
5. Чек-лист по написанию резюме

## 1. Введение

### Где взять шаблон и сделать резюме

Можно использовать готовые шаблоны [hh.ru](http://hh.ru), [moikrug.ru](http://moikrug.ru) или [LinkedIn.com](https://www.linkedin.com) (осторожно, у *LinkedIn* проблемы с экспортом резюме на кириллице, составленных с русским интерфейсом). Можно сделать вручную красивый документ по этой методичке самому или взять [шаблоны Google Docs](#). Или любой другой шаблон файла резюме из интернета.

Но истинный путь самурая, а особенно фронтендера — это самодельный сайт-резюме с прямой ссылкой на него. Он послужит реальной демонстрацией ваших навыков. Высший пилотаж — сделать на нем маленькую потайную (или явную) анимацию, которая порадует своим наличием любого нанимателя. Лучше, конечно, авторскую, а не прикрутить готовую.

### Что нужно знать о позиции Младший фронтенд-разработчик

Задачи как минимум первое время будут довольно простыми и одинаковыми по сути своей, например, вёрстка однотипных страниц. На них вы набьёте руку, прибавите в скорости и сможете браться за что-нибудь посложнее. А потом ещё сложнее, и ещё. На это уйдет какое-то время. Возможно, понадобится даже сменить нанимателя, чтобы работать над более сложными вещами. Рутинные задачи и относительно низкая оплата труда нормальны для старта в любой сфере. Но готовность учиться и проактивность помогут это изменить и заниматься всё более крутыми задачами за всё более большую зарплату. Или, наоборот, вам могут выдать задачи, которые покажутся или действительно будут слишком сложными. И придется долго и мучительно в них разбираться, привлекая старших коллег. Но это нормально — зато вы быстро обростёте опытом, и, вероятно, бонусом стрессоустойчивостью. :)

### Виды компаний

Есть несколько основных видов компаний: стартапы, аутсорс большой и маленький, продукты большие и маленькие, и не-IT-компании с IT-отделом. У всех достаточно своих плюсов и минусов, и полезно понимать, к чему готовиться. Большие компании реже ищут джунов и начинающему разработчику попасть туда сложнее, но расскажем немного и о них.

- Стартапы.** Это новые, совсем маленькие компании, часто до 10 или даже до 5 человек. Бывают и больше по размерам, но духом, смыслом и уровнем процессов остаются такими же. Смысл существования такой компании — получение следующего раунда финансирования, запуск продукта или выпуск следующей версии. И нет никакого точного прогноза, получится ли это, и будут ли у компании деньги через месяц, полгода или год. Маленькой командой стартап старается выдать на рынок нечто сложное, что не всегда физически по силам такому количеству сотрудников. Поэтому стартапы ищут людей, которым понравится идея — и которые готовы будут вкладывать в неё силы, нервы и время. Если предлагаемая стартапом идея (то, что он в итоге выпустит или уже даже выпустил на рынок) вас не вдохновляет, вы вряд ли подружитесь. К плюсам стартапов относятся: семейная ламповая атмосфера; отсутствие бюрократии — откуда ей быть у трёх человек; обычно новый модный стек (набор применяемых технологий); команда энтузиастов, как правило работающих 24x7. Минусы тоже растут из самой сути стартапов. Семейная атмосфера может быть слишком семейной — и конфликт с техдиром при код-ревью закончится увольнением. Но когда все договоренности на словах, а деньги отправляют по номеру карты, уволить могут одним днём и без выходного пособия, — это другая сторона полного отсутствия бюрократии. Или может быть просто полный хаос в рабочих процессах, приоритетах и постановке задач. В конце концов, команда энтузиастов будет ожидать энтузиазма и от вас, — готовьтесь к переработкам. Но если проект действительно крутой, пока финансирование есть (и/или у вас есть финансовая подушка на случай неудачи), а вы любите приключения — можно и попробовать.
- Небольшие продуктовые компании.** Это выросшие стартапы. Общая обстановка становится стабильнее: компания кормит сама себя, а не ждёт финансирования. А ещё копится легаси во всех его видах: древний бесхозный код, технический долг, недокументированные части продукта. Все то, что оставила после себя спешная антибюрократическая разработка стартапа. Легаси осложняет разработку нового функционала и требует поддержки. Когда продукт разрастается, ранее написанные «костыли» самым непредсказуемым образом ломаются, и регулярно приходится всё чинить. Отдел разработки растёт, зарплата в компании — белеет. Бюрократии ещё не много, но она уже есть. Рабочие процессы и подходы ещё настраиваются и могут быть специфичны или даже вызывать отторжение. Таким компаниям нравятся те, кто думает про бизнес и обеспечивающих его существование пользователей. Как сделать лучше? Почему так, а не эдак? Что это нам принесет? Как обеспечить баланс между пожеланиями пользователей, задачами бизнеса и техническими нуждами продукта? Если вам подходит такая постоянная оглядка на уже сделанное и постоянное заглядывание в будущее, если сердце греет мысль о фидбеке от пользователей (спойлер: как правило, он не очень позитивный) — вам в продуктовую разработку.
- Большие продуктовые компании.** Уже обросли бюрократией, зато точно полностью обелились и всё, что касается персонала, делают максимально по закону. Вместе с компанией растут и требования: нужно уметь ещё больше, быть ещё лучше, знать ещё глубже. Вместо одного продукта большие компании делают или несколько, или эдакие ответвления основного: не просто Google, а Google Картинки и Google Видео. При этом в одной команде просто Google'а могут быть интересные задачи и/или крутая команда — а в другой наоборот. И так же для других ответвлений продукта. В этом специфика любой крупной компании. Не важно, аутсорс или продуктовой: пока не попадёшь на конкретный проект и в конкретную команду — не знаешь до конца, как там работается. Различаются процессы, размеры команд, их состав, значение и важность тех или иных ролей в команде, масштаб задач. Различаться может даже оформление — в разные юридические лица, с разными условиями (просто оклад или оклад и бонусы).
- Студии.** Это небольшие компании, разрабатывающие на заказ. К ним приходит клиент и заказывает сайт, даёт все вводные, макеты и все ТЗ, какие у него есть. А студия по ТЗ этот самый сайт и делает — и тратит бюджеты на отступления от ТЗ, как правило, у заказчика нет ни времени, ни желания. После приёмки работ исходный код передается заказчику, а команда берётся за следующий проект. Поэтому в студии улучшайзингом заниматься так много, как в продуктовой разработке, просто не получится. Зато проекты постоянно сменяются, опыт можно получить очень разнообразный. Но часто и поверхностный, потому что нет времени погрузиться в тот или иной стек, не говоря о тонкостях проекта. Сами по себе развиваются софт-скиллы от общения с условными индийскими субподрядчиками или упёртым заказчиком. И даже есть шанс не поддерживать легаси, а постоянно делать что-то новое и с нуля. В отличие от продуктовой разработки, заказчик тут не внутренний, а внешний — и установленные им сроки не сдвигаются, потому что кто-то заболел. Из-за этого чаще, чем в продукте, случаются переработки и авралы. Часто даже они оплачиваемые, но всё ещё не всем подходящие.

- **Интеграторы.** Когда продуктовой компании не хватает рук на все задачи по своему продукту, она отдаёт их на аутсорс. Когда их не хватает большой продуктовой компании, она отдаёт их интегратору — большому аутсорсу. Интеграторы имеют длительные контракты с одними и теми же заказчиками, а значит в конкретный проект будет время углубиться. Иногда это новые проекты и задачи, на самостоятельный запуск которых не хватило рук у заказчика прямо сейчас. Иногда — поддержка именно легаси в работоспособном состоянии и работа третьей линии поддержки. Команды, люди и проекты постоянно тасуются, из-за переходящих архитекторов и лидов плавают туда-сюда уровень технической квалификации. Зато проекты в крупнейших интеграторах уже международные, можно поработать на проекте с именем (правда, закрытым NDA для резюме и обсуждений), с командами из разных стран. И есть всё та же специфика больших компаний, описанная в пункте выше.

## Формы занятости

С работой в офисе для любой из перечисленных выше компаний всё понятно: приходишь, работаешь. Но есть отдельные формы занятости со своими особенностями, которые стоит осветить.

- **Удалённая работа.** Может встретиться в любой компании. Требует высокого уровня самоорганизации: нужно делать задачи в срок и рабочее время. И быть на связи, а не чесать пузо коту или смотреть ютуб (начальник же не видит!). Кому-то такой формат не подходит из-за одиночества, кому-то приходится оборудовать домашний офис или арендовать место в коворкинге. При удалённой работе тяжелее с коммуникациями: вы не можете просто подойти или подсесть к коллеге и за 3 секунды решить вопрос, на котором застряли. И очень важно, как строит процесс работы с удалёнщиками сама компания, особенно если основная команда находится в офисе. В общем, удалённую работу сложно готовить, но иногда получается круто.
- **Фриланс.** Это временные проекты, которые, как правило, выполняются удалённо. Начинающие фрилансеры вынуждены «соревноваться» за каждый заказ на биржах фриланса с более опытными или вообще ботами, которые автоматически откликаются на заказы по ключевому слову. Часто попадают на не очень адекватные заказчики, чаще всего это люди не из IT и даже не рядом, которым нужно сделать какой-нибудь сайт. Желательно сделать его позавчера и с функционалом, который невозможно реализовать. Фрилансеров часто обманывают или пытаются обмануть с оплатой. Набравшись опыта на мелких заказах, можно повстречать заказчика-мечту и годами поддерживать его проекты, но обычно до встречи с ним проходит от года работы на фрилансе с нестабильным заработком.
- **Аутстафф.** Это как бы следующий уровень аутсорса: вы не просто делаете проект стороннего заказчика, вы даже сидите у него в офисе. В большинстве случаев аутстафф используют на своих проектах банки. Плюс: один и тот же проект (это могут быть сменяющиеся его подчасти, но глобально один и тот же) в работе, потом при желании и наличии вакансий в штате самого банка можно перейти к ним. Минус: как правило, штатные сотрудники всё равно не считают вас равносильной частью команды, а проект может оказаться очень скучным. И ещё один важный минус: вы не определяете, над чем работать, даже на уровне того, над продуктами какого банка трудиться. То есть в один день контракт поддержки банка А может закончиться, и уже на следующий день вы, всё бросив, обнаружите себя в офисе банка Б.

## 2. Как писать резюме: общие моменты

### Учитывайте опыт, бэкграунд и место жительства

#### Нерелевантный опыт

Если у вас есть нерелевантный IT-отрасли опыт работы более пары лет, его обязательно нужно указать, но кратко. Так работодатель поймёт, что вы не просто два года ничего не делали (как если бы вы скрыли этот опыт из резюме), а просто трудились в другой сфере. Для позиций, специфика которых пересекается с имеющимся у вас опытом, его можно раскрывать подробнее. Например, компания автоматизирует маркетинговые штуки, а вы в прошлом маркетолог, — вы знаете предметную область, это полезно.

Если нерелевантного опыта много, например, вы 15 лет работали в продажах — тоже указывайте. Из такого опыта можно попробовать «выжать» в Summary (о нём позже) что-то общепольное: например, вы просто гений переговоров. Или вы работали инженером на заводе — отлично, вы уже готовый технарь,

просто из другой области. Вы знаете, как работает огромный станок и умеете его отладить — научитесь отлаживать и небольшой рабочий проект.

## Место жительства

Если вы живёте в маленьком городе и не планируете или пока не можете переехать, предложений на рынке вакансий для вас будет меньше, чем в миллионниках. Но всё ещё есть удалённая работа, которая, хотя и не всем подходит, может послужить хорошим стартом в плане обучения и дать финансовое подспорье для переезда.

## Пишите под вакансию

Писать резюме под конкретную вакансию, особенно если вы джун, — отличная идея. Это привлечёт внимание и повысит шансы на какой-либо фидбек, даже если вас не готовы будут пригласить на встречу. Самый верный путь при оформлении такого резюме — отталкиваться от конкретных требований позиции. Например, компания делает сайт новостей о киберспорте и ищет человека с хорошим опытом вёрстки. А вы как раз любите игры, и у вас симпатичное портфолио верстальщика — пройдите по каждому пункту требований и напишите, что вы по нему умеете. Честно пишите, если ещё недостаточно хороши в чём-то из нужного — это даже внушит доверие. Часто в вакансиях требования несколько завышены: всегда хочется найти человека, который умеет всё и сразу, но часто подходит и кандидат с не таким обширным опытом. Приложите ссылки, обязательно укажите свою любовь к сфере и желание приложить руку к развитию российского киберспорта или популяризации самой дисциплины.

Но будьте внимательны, когда делаете резюме под вакансию, а то может получиться неловко. Например, вы нашли компанию, проект которой много лет назад начинался как сервис для автоматизации рекрутмента. Сейчас он уже разросся до автоматизации и остальных эйчар-процессов. А вы, прочитав по диагонали, возьмёте и напишете, что автоматизация рекрутмента — это ваша мечта. И попросту спалите собственную невнимательность к вопросу. Нужно делать или хорошо, то есть разбираться как следует, прежде чем писать — или не писать вообще того, в чём вы не уверены на 100%.

## Придерживайтесь общего формата

### На чём делать акцент

При составлении резюме нет жёстких правил, но есть некие базовые вещи, которые нужно и важно указать. Расставлять акценты на тех или иных моментах нужно исходя из вашего чувства прекрасного и того, что именно вам кажется ценным для конкретного работодателя. Акцентировать внимание на 15 лет опыта в продажах не стоит, а вот на учебном опыте в такой ситуации — обязательно. Укажите самые крутые курсы, которые вы прошли, самые важные книги, которые действительно помогли вам вырасти как разработчику. И будьте готовы про эти книги при встрече поговорить. Скорее всего они очень известные, и интервьюер может полюбопытствовать, что именно вам в них понравилось. Указывайте ссылки на учебные проекты, пишите, на каких IT-мероприятиях вы были. Посещение условного BeerJS добавляет плюсы в карму по умолчанию — за один такой митап можно узнать много нового, не говоря о знакомстве с кучей интересных ребят из индустрии.

### Внешний вид

Форматирование — это важно. Аккуратно оформленное резюме радует глаз и улучшает карму. Лучше всего высылать резюме в формате PDF — и оформление останется красивым, и откроется в любой системе. Инфографика — хороший вариант, но главное не переборщить. Плюс, её не переварят системы трекинга кандидатов по циклу подбора — рекрутер или тимлид будут долго мысленно ругаться, руками копируя ваши данные и переписывая информацию.

### Формулировки

Фразы в описании опыта нужно начинать со слов вида «Умею, Владею, Сделал, Реализовал, Улучшил». Формулировки должны быть короткими и чёткими. Без сложноподчинённых предложений на полстраницы. Как в книге «Пиши @ Сокращай» Максима Ильяхова. Если не читали, есть два простых метода проверить оптимальную длину предложения:

- 1) Прочитать вслух, если дыхание сбивается до конца предложения — нужно сокращать или делить на два;
- 2) Посчитать слова. Если их больше 13, к концу предложения читатель забывает начало. Решение такое же: сокращать или разделять.

## Определяйте длину резюме и дизайн

### Опыт

Если есть только учебный опыт и/или всего пара лет рабочего, резюме должно занимать 1 страницу. У вас ещё не так много полезного и релевантного опыта, чтобы он занимал столько места. Максимум — 2 страницы: если вы расстарались описать учебные проекты и выбрали относительно крупный шрифт. Если опыт есть не только учебный — резюме должно занимать 2-3 страницы, не больше. Резюме на 4, 6 или 10 страниц уже попросту тяжело читать, а время нанимающих менеджеров ограничено.

Нерелевантный опыт пишется максимум в 2-3 предложения. Перечислять обязанности из него на два листа категорически нельзя.

### Шрифт

Используйте относительно стандартные шрифты. Не надо брать шрифт, превращающий букву «б» в цифру «6», не надо брать жирные курсивные. Arial, Times New Roman и Helvetica, и их близкие родственники ещё никого не подводили. Выделять жирным можно только подзаголовки резюме, никакого выделения ключевых слов делать не нужно.

### Выравнивание и направляющие

Выравнивайте текст по левому краю, выравнивание по ширине читается хорошо только в дипломных работах. Направляющие можно оставить стандартными для Word'a или Google Docs, можно оставить всего по сантиметру от краёв страницы. Если контент резюме при этом удачно смотрится и попадает в строки, не повисая одним словом из конца предложения на следующей строке — вы великолепны. Если так не получилось, вернитесь к стандартному форматированию, оно всё простит.

## 3. Структура: что и где писать

### Заголовок или название

Здесь нужно написать должность, которую вы собственно ищете. Ориентируясь на свои предпочтения, воспользуйтесь нашим конструктором:

### Общий момент

На русском языке разработчик фронтенда назовёт резюме «Фронтенд-разработчик», а на английском «Frontend Developer» — слово developer нужно писать с большой буквы. При этом по-русски в названии «Разработчик бэкенда» слово «бэкенд» будет писаться именно с маленькой буквы.

### Первое слово

Младший или Junior. Его использование опционально: из вашего скромного или только учебного опыта уже будет понятно, что вы только начинающий разработчик. Но его может быть полезно использовать, если есть нерелевантный опыт.

### Второе слово

Основа вашего стека или направление. Если берём технологический стек, словом может быть JavaScript (можно написать и JS, но лучше полностью), или PHP, или React, или можно написать JavaScript (React). Варианты, если берём направление: web-/веб- (именно как приставки с дефисом к третьему слову), фронтенд/Frontend, бэкенд/Backend. Верстальщикам нужно пропустить этот и следующий пункты и назваться просто верстальщиками или младшими верстальщиками.

### Третье слово

Разработчик/Developer.

В итоге собираются следующие примеры:

- Junior Frontend Developer
- Младший фронтенд-разработчик
- Junior фронтенд-разработчик
- Младший веб-разработчик

## Аккуратнее со словом «стажёр»

Важно: добавлять в заголовок словом «стажёр» нужно тогда и только тогда, когда вы ищете и согласны именно на стажировку. То есть работу на несколько месяцев на совсем лёгких задачах, которые могут быть и не профильными, например, в задачах появится организационная помощь руководителю. После стажировки вас могут взять в штат (то есть на постоянную работу), а могут и не взять — это на усмотрение работодателя, и принимать решение он будет в самом конце стажировки.

## Имя Фамилия

Полностью, на том же языке, на котором пишется резюме. Не Васечка Пупкин, а Василий Пупкин. Псевдоним лучше не писать, даже если очень хочется: чтобы озадаченный тимлид не был вынужден писать вам письмо, начинающееся со слов «Ракетный комсомолец, добрый день!»

## Контакты

### Город

Если планируете переезд или, тем более, сидите на чемоданах и полностью к нему готовы, — указывайте. Во втором случае с ожидаемыми сроками переезда и наличием или отсутствием возможности приехать на очное интервью в город переезда.

### Телефон

Опционально — если готовы, что вам будут звонить. Можно просто указать никнейм в мессенджере. Если указываете телефон, напишите, в какие часы вам удобно по нему общаться.

### email-адрес

Опять же лучше, если это будет не почта васечкапупкин96@мэйл.ру или нагибатор777@бк.ру, а хотя бы василийпупкин@yandex.ru. Часть компаний имеют плохо объяснимое предубеждение к айтишникам-владельцам почт на доменах мэйла, имейте в виду.

## Ссылки

Ссылки на профили в социальных сетях: LinkedIn, Github, Facebook. Если соцсеть совсем пустая, лучше не указывать. Github, конечно, должен быть приведён в порядок, а к каждому проекту написано хорошее ридми.

Как оформлять любые ссылки:

- Перед ссылкой нужно краткое описание, что по ней доступно.
- Ссылка указывается полностью, а не ставится на текст. В идеале ссылка уже достаточно короткая (примерно половина строки) и не требует сокращения через специальные сервисы. Так видно, куда она собственно ведёт: на Github, сайт-портфолио, сайт компании, в которой вы работали или куда-то ещё.

## Фотография

Фотография в резюме размещается также в начале, в одном из верхних углов.

«Нет!» фотографиям на фоне ковра, с пляжа, с другими людьми и в странных одеяниях. «Да!» — портретным фото при хорошем освещении в нейтральной одежде или удачно сидящем костюме. Фото из паспорта — крайний вариант, лучше не надо. Но фотографию лучше иметь — она даёт понимание, что за всеми этими буквами, опытом и стэком стоит вон тот улыбающийся из угла резюме кандидат.

## Цель

В резюме можно указывать цель ваших поисков, эдакий ответ на вопрос «кем вы хотите стать, когда вырастете». Это опциональный пункт, но если есть, что интересного здесь написать — пишите.

Разбор примеров целей со сравнением:

- Пример 1: «Найти стабильную работу в динамически развивающейся международной компании, чтобы достойно применить свои умения и навыки на благо коллектива». Звучит хорошо для крупной корпорации, но так себе для маленькой. Маленькая (да и большая, не совсем ещё обросшая корпоративщиной) компания любит точность и конкретику, прямую пользу, которую вы

можете принести. «Умения и навыки» — это не конкретика, да и «благо коллектива», формально говоря, никого не волнует — волнует благо компании, благо продукта, благо его пользователей.

- Пример 2: «Начать карьеру junior фронтенд-разработчика в компании X и через 5 лет дорасти до тимлида». Амбициозно, сразу понятны большие планы и желание в компании задержаться на несколько лет. Понравится почти любой компании. Но есть слабое место — дорасти до тимлида можно и быстрее, или может компания не растит тимлидов внутри, а берёт снаружи. Или вообще люди дольше двух лет в ней не работают, и тогда будет звучать круто, но странно. Лучше использовать чуть более гибкую формулировку.

## Summary (Представление себя)

Этот пункт похож на цель, но он отвечает на вопрос «кем я вырос сейчас».

В Summary должно быть до пяти ключевых тезисов, которые в выгодном свете расскажут о вас и вашем опыте. Здесь можно упомянуть тот самый не-IT-опыт, который релевантен сфере компании/конкретному проекту, например: «Знаю, как сделать вашу CRM лучше, потому что два года страдал с 1С». Если же не получается интересно подать ключевые моменты вашей карьеры, лучше пропустите этот раздел.

Пример с разбором:

*«Опыт разработки программного обеспечения более 2 лет, последние два года как web-разработчик на Javascript/HTML/CSS/Java. Также имею опыт программирования на C/C++/C#, PHP, AS3, SQL и работы в графических пакетах, таких как Photoshop, CorelDraw и 3DStudio MAX. Имею хорошие аналитические способности. Легко обучаюсь. Хорошо работаю в коллективе».*

В таком саммари есть 2 потенциальных слабых места: 1) скорее всего, опыт, перечисленный после слова «также», был получен ДО перехода в веб-разработку на новом стэке, но это не до конца понятно. В чём проблема? Если весь перечисленный стек использовался разом, опыта именно с JS и вёрсткой в итоге получится не так много, как написано. 2) Аналитические способности, обучаемость и работа в коллективе — с одной стороны, подходит для формата саммари, но с другой — это всё ещё неинформативно. «Имею опыт работы в команде» или «умею/люблю работать в команде» здесь имело бы больший вес.

## Опыт работы

Его принято писать в обратном порядке: от последнего места к первому.

### Как описать свой опыт 6-7 месяцев в Практикум

Зависит от того, есть ли уже релевантный опыт работы под вакансию. Если есть — пишите его выше, как последний опыт, а проекты практикума включайте в раздел с IT-курсами. Или можно указать их прямо в последнем опыте, мол, параллельно с работой прошел курсы, вот ссылки на мои проекты. Так эта информация сразу окажется под рукой у нанимателя. Если релевантного опыта нет — выше всего как последний опыт располагайте учебный. Как компанию можно указать собственно Яндекс.Практикум, «срок работы» — это срок вашего обучения в таком опыте. Как проекты, конечно, нужно перечислить сами учебные проекты с кратким описанием и ссылками. Не забудьте указать, что ваши проекты проходили ревью у менторов: это полезный опыт и значит, что вы не просто сами учились, но и кто-то ещё ваш код читал.

### Структура описания опыта

- Период работы. В месяцах и годах, конкретная дата начала/окончания проекта никому не нужна
- Компания. Можно просто указать название, можно дать ссылку, можно вставить ссылку на текст. Лучше использовать не официальное название ООО «Рога и копыта», а то, которое написано у компании на главной — оно явно понятнее и более узнаваемо.
- Должность. Не нужно переписывать должность из трудовой книжки: пишите по факту. Если брать должность из трудовой, все будут инженерами. А вы, например, Middle JavaScript (React) Developer.
- Проекты. Краткое описание проекта/продукта, над которым вы работали — радость для сердца нанимающего менеджера. Позволяет примерно прикинуть масштаб задач, увидеть близкую предметную область или даже попросту направление — например, больше в задачах было вёрстки или JS.
- Обязанности. «Написание кода» или «разработка» — не подходит. Опишите размер команды, вашу роль в ней, дополнительные постоянные задачи: общение с заказчиком, код-ревью коллег. Сделали/

спроектировали/реализовали что-то крутое — укажите, это ваше достижение. Не используйте слово «делаю», «работаю над» — старайтесь писать о том, что уже сделано, закончено, что прошло все ревью и пошло на продакшен.

- Технологии, используемые на вашем проекте: учебном или рабочем. Можно указывать базовые: языки, основные фреймворки, БД. Можно взять шире и писать вообще всё, с чем сталкивались, и что используют коллеги по проекту — но только если вы действительно это трогали и хотя бы имеете представление. Лучше ограничиться промежуточным вариантом: например, добавить к базовому стэку инструменты DevOps, которыми вы пользовались в работе. Можно выделять технологии в отдельный список, а можно просто перечислять по ходу описания проекта и задач.

## Навыки

Можно писать отдельно или оформить инфографикой («знаю JS на 8 из 10»), но за такую самооценку тоже нужно быть готовым держать ответ. А можно не писать — все предыдущие пункты уже всё расскажут.

### Какие навыки обязательно ожидают от начинающего фронтенд-разработчика:

- HTML + CSS. Верстать любому новичку нужно будет минимум 40% времени, обычно в самом начале работы это число близко к 100%.
- Знание нативного JavaScript'a.
- Знание алгоритмов. О них можно посмотреть лекции Яндекса: <https://yandexdataschool.ru/education/courses/algorithms>.

### Что еще желательно знать и уметь и когда:

- Знание или опыт самостоятельного изучения любого фреймворка — большой плюс для начинающего. Браться за изучение правильнее всего после прохождения курса по нативному JS. React сейчас самый популярный, за ним идет Angular. Vue используют реже, как правило, в стартапах и небольших компаниях.
- Знание какой-либо CMS — только если хотите и готовы работать с CMS и сама вакансия его требует.
- CSS-фреймворки — только если обязательно для конкретной вакансии, в среднем их не нужно знать хорошо.
- Полезно понимать, что такое UI и UX и знать разницу между этими понятиями. Ещё полезно знать, что делают и зачем нужны дизайнеры и аналитики.
- Знание, как работает backend-часть, тоже идёт вам в плюс. Даже если вакансия этого не требует совсем — просто как показатель вашего здорового интереса к сфере и желания знать, как всё работает внутри.
- В крупных компаниях как правило приветствуется знание Node.js и интерес к фулстэк-разработке.

## Образование

- Название учебного заведения, факультет, специальность, полученная степень, годы обучения или просто год окончания.
- Указать основные IT-курсы, приложить ссылки на сертификаты, если они есть.

## Дополнительная информация

Это может быть что-то полезное для работодателя, например, информация о том, что вы готовы выйти на работу только после окончания текущего проекта через месяц. Или что угодно на ваш вкус: «люблю котиков», «конвертирую кофе в чистый код», «стреляю из лука почти так же хорошо, как верстаю». Резюме — это серьёзно, но в нём должен просматриваться человек, так оно становится интереснее. Но есть и другие способы показать себя как человека: даже ваш авторский стиль письма.

## Хобби

Если вы решите написать здесь про хобби, пишите нескучно. Потому что можно написать «читаю книги», а можно «упарываюсь по малоизвестной фантастике двадцатого века». И первый вариант, действительно, не про вашу индивидуальность. Зато второй — отличный повод свободно поболтать на собеседовании.



## Качества

Здесь можно указать личные качества. Но не нужно перечислять 20 слов вида «коммуникабельный, исполнительный, обучаемый...» — потому что их перечисляют все, за ними нет ничего реального и их никак особо нельзя проверить. Если очень хочется сделать на чём-то упор, выберите 3-5 пунктов (меньше — лучше) и кратко проиллюстрируйте. Например, вы точно быстро учитесь, потому что закончили школу и вуз экстерном. Или вы коммуникабельный, потому что умудрились построить прочные дружеские отношения с десятком бывших коллег. Можно вообще обойтись без стандартных слов и писать сразу реальный опыт: «умею работать по 80 часов в неделю, когда горит проект. Потом беру отпуск, но проект сдаём в срок».

## Языки

Здесь же можно указать знание языков, если они требуются в вакансии. В остальных случаях писать уровень английского и русского не нужно, если вы из русскоговорящей страны и уже способны прочитать документацию на английском.

# 4. Финальные штрихи

## Вычитать резюме

Вернитесь к готовому тексту через несколько часов или на следующий день. Это позволит увидеть пропущенные ошибки и поправить части, которые казались хорошо написанными только в конкретный момент времени. Вставьте текст в Google Docs, чтобы он подсветил опечатки. Дайте почитать резюме другу, он может заметить что-то, что вы упустили: неполноту информации или достижение, о котором вы просто забыли упомянуть. Пропустите текст через Главред: <https://glvrd.ru/>, он подсветит проблемные места. Только править по рекомендациям Главреда нужно без фанатизма: высокий балл оценки значит только то, что в тексте нет «лишних» по мнению сервиса слов.

Честность — лучшая политика. Не указывайте в резюме ложную информацию. Враньё и приукрашивания обязательно всплывут, не на собеседовании — так уже в работе, и вы сами пожалеете, что решили приврать в резюме.

# 5. Чек-лист по написанию резюме

## 1. Изучить вакансию

- Читайте описание вакансии, особое внимание уделяя стэку и требованиям — сделайте резюме максимально релевантным, убрав лишнее
- Читайте о компании, проектах, условиях. Для своего понимания и более предметного диалога на собеседовании

## 2. Наполнить контентом по структуре

1. (опционально) Пишите заголовок
2. Вверху должны быть фотография, имя с фамилией и контакты
3. (опционально) Цель поисков и саммари
4. Релевантный рабочий опыт. Или учебный, если нет рабочего. Описывайте в обратном порядке (от последнего — к раннему), нерелевантный вакансии старый опыт сокращайте
5. (опционально) Перечисление навыков отдельно, если не вписали в опыт. Обязательно указывая базовые (вёрстка, JS), остальное добавляйте под конкретные требования вакансии
6. Образование, здесь же любимые IT-курсы и Яндекс.Практикум, если релевантный опыт есть
7. Дополнительная информация: хобби и любые нерабочие моменты, которые важно указать. Знание языков

## 3. Оформлять аккуратно

- 1 страница для начинающего, 2-3 для опытного кандидата
- Шрифт читаемый, нет ярких вставок, нет **игр с форматированием** и шрифтами

- Фразы короткие, понятные и читаются на одном дыхании. Не более двух запятых на одно предложение.

#### **4. Проверять**

- Первым делом — просто отложите резюме в сторонку, чтобы вернуться к нему после паузы
- Вычитав, дайте прочесть другу
- Проверьте, что все ссылки работают, а по ним лежит аккуратно оформленный контент
- Уберите лишнее и нерелевантное для вакансии
- Подумайте ещё раз, не забыто ли что-нибудь важное. Добавьте, если забыли, и перечитайте ещё раз.