

Junior	Middle	Senior
Опыт работы		
	От 1 года	От 3 лет
Зарплата		
20000 - 30000	30000 - 60000	60000 - 80000
Работа в Linux		
Умеет пользоваться консолью, знает основные команды (ls, cd, cat). Уверенно ориентируется в графическом интерфейсе.	Может самостоятельно поставить и настроить любой необходимый софт.	Имеет представление о работе операционной системы: системные вызовы, модули ядра, планировщик задач, приоритеты процессов.
Администрирование Linux		
Знает, где искать логи и что с ними делать. Что такое сигналы, какие бывают, что делают. Понятие Load Average.	Может поставить и настроить любой пакет. Может редактировать правила iptables. Знает понятия потоков, их перенаправления, модификация с awk/sed/grep. Понятие сокетов, портов. Что такое cron и синтаксис crontab'a. Знает протоколы передачи почты и их различия (POP3, SMTP, IMAP)	Умеет писать юниты в systemd, знает, какие они бывают. Умеет писать скрипты в init.d.
Работа с системой контроля версий		
Знает назначение и основные возможности VCS. Умеет работать с git'ом (commit, push, pull, merge, branch). Умеет разрешать конфликты при мерже. Уметь логически разделять коммиты между собой. Может писать к коммитам внятные сообщения. Понимает концепт "feature branch". Может построить свою работу так, чтобы вести разработку на отдельных ветках.	Умеет в работу с историей (log, blame, bisect, rebase). Знает, как перезаписать историю и случаи, когда это нужно. Имеет представление о других существующих VCS и их особенностях. Может провести ревью junior'a. Может провести ревью самому себе.	Знает основные модели ветвления (gitflow, gitlab flow, github flow), плюсы и минусы. Может провести ревью middle. Имеет представление о внутренностях git'a.
Верстка		
Знает все основные теги и их семантический смысл.	Может объяснить про наследование стилей и специфичность	Умеет верстать по БЭМ.

<p>Знает все основные стили и их назначение.</p> <p>Знает основы блочной и табличной верстки, когда и какую нужно применять.</p> <p>Умеет анализировать макет, представляя в уме способ реализации.</p>	<p>селекторов.</p> <p>Понимает принципы адаптивной верстки.</p> <p>Верстка писем.</p> <p>Может заверстать анимации, тени, повороты, градиенты и другие фишки CSS 3.</p> <p>Знает основы микроразметки и основные схемы данных, встречающихся в вебе.</p> <p>Знает об основных псевдоклассах и моментах их наступления.</p> <p>Знает все возможные селекторы (по значению атрибута и прочие).</p> <p>Знает и использовал препроцессоры: SCSS, LESS.</p>	<p>Знает основные принципы работы движка браузера. Что такое reflow. Как браузер парсит стили, как повысить производительность селекторов.</p> <p>Знает принципы оптимизации верстки - минификация стилей, скриптов, изображений, спрайты, почему спрайты не нужны.</p>
<p>PHP</p>		
<p>Знает синтаксис языка и основные функции стандартной библиотеки.</p> <p>Может рассказать про основные типы и правила их приведения.</p> <p>Знает и использует стандарты оформления кода (PSR-1, PSR-2).</p> <p>В чем разница между передачей параметров по ссылке и по значению.</p> <p>Что такое суперглобальные массивы и что в них находится.</p>	<p>Отличия self от static.</p> <p>Понятие статических методов класса.</p> <p>Что такое генераторы и область их применения.</p> <p>Что такое магические методы, как они работают и когда вызываются.</p> <p>Различия модификаторов доступа членов класса (private, protected, public).</p> <p>Интерфейсы стандартной библиотеки (Countable, ArrayAccess и прочие).</p> <p>Псевдотипы: callable, iterable.</p> <p>Что такое кэширование байткода.</p> <p>Что такое автозагрузка классов. Стандарты автозагрузки (PSR-0, PSR-4).</p> <p>Как работать с composer'ом - установка, обновление и создание пакетов, где их искать.</p> <p>Настройка параметров в php.ini.</p> <p>В чем польза тайпхинтов.</p>	<p>Знает Reflection API, как и когда его использовать.</p> <p>Может спроектировать архитектуру приложения с нуля.</p> <p>Общее представление об устройстве интерпретатора PHP.</p>
<p>Javascript</p>		
<p>Знает синтаксис языка и основы стандартной библиотеки.</p>	<p>Знает все возможности ES 2017.</p>	<p>Знает, что такое вебсокеты, когда и как их использовать.</p>

<p>Работал с JQuery и имеет представления о возможностях, которые оно предоставляет.</p> <p>В состоянии поставить и настроить плагин на JQuery или просто плагин.</p> <p>Имеет примерное представление о устройстве DOM.</p> <p>Что такое события, откуда они появляются, какие бывают, как добавить свой обработчик.</p>	<p>Может в подробностях рассказать про this.</p> <p>Работал с Backbone - модели, представления, коллекции. Какие методы имеются в каждом из них, какие события, как они связаны между собой.</p> <p>Знает синтаксис EJS шаблонов и работал с ними.</p> <p>Понимает принципы работы системы сборки (gulp, webpack), может вносить мелкие модификации.</p> <p>Может рассказать о модульности в JS: зачем нужна, что такое AMD, что такое CommonJS, импорты в ES 2017.</p>	
<p>Английский язык</p>		
<p>В состоянии читать техническую документацию.</p> <p>В состоянии читать статьи и заметки в блогах.</p> <p>Может письменно формулировать короткие предложения - коммиты, комментарии.</p>	<p>Может вести письменный диалог на технические темы.</p> <p>В состоянии воспринимать на слух вебинары, презентации и речь носителей языка.</p>	<p>Может устно вести разговор на технические темы.</p>
<p>Техподдержка пользователей</p>		
	<p>Владеет грамматикой и орфографией на приемлимом уровне.</p> <p>Навыки вежливости в переписке.</p> <p>Умеет идентифицировать проблему по описанию "Все сломалось".</p> <p>Стойкость к критике и негативу от пользователей.</p>	<p>Четкая и внятная речь.</p> <p>Навыки вежливости при разговоре по телефону.</p> <p>Может оказывать устную поддержку.</p>
<p>Прием требований от заказчика</p>		
	<p>Умеет анализировать требования на предмет внутренней непротиворечивости.</p> <p>Может предлагать решения несложных проблем.</p> <p>Может составить техническую спецификацию задачи.</p> <p>Может проконсультировать заказчика на несложные темы.</p> <p>Имеет общие представления о возможностях и ограничениях используемого стека технологий.</p> <p>Умеет разделять задачи на подзадачи.</p> <p>Владеет культурой переписки.</p>	<p>Умеет анализировать реальные неназванные потребности заказчика.</p> <p>Имеет представление о потенциальных вопросах, которые возникнут у заказчика в будущем и заранее знает их решение.</p>

Работа с Bitrix'ом		
<p>Имеет базовые представления об устройстве системы - сайты, шаблоны сайтов, компоненты, параметры компонентов, шаблоны компонентов.</p> <p>Может интегрировать готовую верстку как шаблон битрикса.</p> <p>Умеет создавать и редактировать страницы и разделы.</p> <p>Может поставить и настроить стандартные компоненты.</p> <p>Имеет представление об инфоблоках.</p> <p>Может редактировать шаблоны компонентов.</p> <p>Может редактировать шаблоны сайтов.</p> <p>Отличия result_modifier и component_epilog.</p> <p>Уровень "Базовый" по классификации битрикса.</p>	<p>Может разрабатывать собственные (простые и комплексные) компоненты с нуля.</p> <p>Знает, как работает кэширование в битриксе.</p> <p>Знает о различиях старого и нового ядра.</p> <p>Знает, как работает почтовая система.</p> <p>Знает, что такое события и как с ними работать.</p> <p>Знает, как работает ЧПУ.</p> <p>Умеет в ORM.</p> <p>Умеет в Api битрикса.</p> <p>Уровень "Профессионал" по классификации битрикса.</p>	<p>Может разработать собственный модуль с нуля.</p> <p>Знает, как устроены внутренности стандартных модулей.</p> <p>Принципы работы композитного сайта. Адаптация сайта под композит.</p> <p>Уровень "Эксперт" по классификации битрикса.</p>
Разработка API и внешние интеграции		
<p>Знает понятие API.</p> <p>В состоянии использовать любое API при наличии документации.</p>	<p>Знаком с наиболее распространенными форматами передачи - JSON/XML.</p> <p>Знаком с наиболее распространенными схемами аутентификации - Password, Token, OAuth.</p> <p>Знаком с наиболее распространенными протоколами взаимодействия - REST, RPC, SOAP, JSONP.</p>	
Безопасность в вебе		
<p>Знаком с основным правилом - никогда не доверять данным, пришедшим от пользователей.</p>	<p>Знает об основных уязвимостях - XSS, SQLi, CSRF, RCE, повышение привилегий, отравление кэша и прочие.</p> <p>Знает способы противодействия всему вышеперечисленному.</p> <p>Умеет находить неочевидные логические ошибки, приводящие к уязвимостям.</p>	<p>Имеет представления о том, как работает шифрование в общем и SSL в частности.</p> <p>Имеет представления о методах противодействия DDOS атакам.</p>
Инфраструктура проектов		
<p>Знает, зачем нужен веб сервер, сервер для кэша, сервер базы данных. Как они взаимодействуют между собой.</p>	<p>Знает, что такое очередь сообщений, брокер сообщений, какую задачу он решает.</p> <p>Знаком со всеми компонентами инфраструктуры, используемыми в рамках проектов, с которыми он работает.</p>	<p>Может разработать архитектуру проекта с нуля на основе требований.</p> <p>Умеет в прогнозы по требуемым изменениям в инфраструктуре.</p> <p>Знаком с концептами шардирования и репликации баз данных.</p>

		<p>Знает, как работает сетевой стек. Чем различаются 1G/10G карты.</p> <p>Имеет представление о принципах работы виртуальных машин.</p> <p>Умеет в автоматизацию развертывания инфраструктуры. В частности, Ansible.</p> <p>Имеет представление об альтернативных инструментах деплоя.</p> <p>Читал и разбирался в кейсах построения высоконагруженных систем - Twitter, Instagram, прочие.</p>
Проектная документация		
<p>Умеет читать.</p>	<p>Может связно и разборчиво написать статью в документацию на заданную тему.</p> <p>Может объяснить простыми словами сложные вещи.</p> <p>Имеет навыки работы с наиболее распространенными форматами доков - Markdown, RST, HTML.</p>	<p>Умеет планировать структуру разделов документации и список необходимых в них статей.</p> <p>Умеет в делегирование работы по документации.</p>
SQL и базы данных		
<p>Знает, что такое БД.</p> <p>Может писать несложные CRUD запросы.</p> <p>Может писать несложные запросы на изменение структуры таблиц.</p>	<p>Знает, зачем нужны внешние ключи, как их создавать, удалять, использовать. Какие бывают "ON DELETE".</p> <p>Способен в уме представить, как отработают триггеры внешних ключей при связи нескольких таблиц (более двух).</p> <p>Знает, какие еще бывают constraint'ы.</p> <p>Умеет в сложные запросы.</p> <p>Умеет в группировки.</p> <p>Знает отличия WHERE от HAVING.</p> <p>Знает принципы анализа и оптимизации запросов. Что такое индексы и как ими пользоваться.</p> <p>Знает основные функции по работе с строками, числами, датами.</p> <p>Знает про кэширование запросов на стороне БД.</p>	<p>Концепт Materialized View.</p> <p>Нормальные формы.</p> <p>Имеет представление о том, как работают индексы.</p> <p>Может рассказать про двухфазный коммит.</p> <p>Может рассказать про уровни изоляции.</p> <p>Знаком с реляционной алгеброй.</p> <p>Что такое денормализация данных и когда она нужна.</p> <p>Умеет в проектирование архитектуры БД.</p> <p>Знает разницу между суррогатным ключом, естественным ключом, составным ключом. Может объяснить, когда и какой применять.</p> <p>Имеет представление о том, как БД оптимизирует запросы перед выполнением.</p>
Основы веба		
<p>Знает, что такое ajax.</p> <p>Знает, что такое cookie.</p>	<p>Знает, зачем нужен robots.txt и как его редактировать.</p> <p>Знает, зачем нужен sitemap, какие ссылки в него включать, какую</p>	<p>Может рассказать про схему работы HTTP.</p> <p>Может рассказать про схему работы HTTPS.</p>

<p>Знает, что такое сессия.</p>	<p>информацию можно в нем передавать.</p> <p>Знает, что такое RSS.</p> <p>Знаком с основными DNS записями и их назначением: A, AAAA, MX, CNAME.</p> <p>Знает о существовании и семантике разных HTTP методов (PUT, DELETE, POST, etc).</p> <p>Знает назначение часто используемых хедеров: Cache-Control, Expires, User-Agent, Referer.</p> <p>Основы SEO.</p> <p>Знает структуру запросов и ответов, способ их кодирования в текст.</p>	<p>Знает принципы работы DNS.</p> <p>Знает принципы работы TCP/IP.</p> <p>Может рассказать про модель OSI.</p>
<p>Computer Science</p>		
<p>Знаком с основными структурами данных: Stack, Tree, Hashmap, Queue, Set, List и другие.</p> <p>Знает, как используются регулярные выражения и как их писать.</p> <p>Что такое кодировки символов и чем они отличаются.</p>	<p>Может рассказать про O-нотацию.</p> <p>Знаком с концептами параллельного программирования - мутексы, семафоры, взаимодействия между процессами.</p> <p>Знаком с концептами функционального программирования - функции высшего порядка, неизменяемые структуры, чистые функции.</p>	<p>Знает принципы работы процессора - прерывания, регистры, уровни кэшей, прочее.</p> <p>Знает принципы работы HDD.</p> <p>Знает принципы работы SSD.</p>
<p>Тестирование</p>		
<p>В состоянии тестировать интерфейс с позиции пользователя.</p> <p>Может мысленно представить и проверить все возможные варианты взаимодействия пользователя и тестируемой функции.</p> <p>Может описать сценарий воспроизведения для обнаруженной ошибки.</p>	<p>Умеет в автоматическое тестирование по методу черного ящика (functional, integration).</p> <p>Знаком с концептом подмены зависимостей.</p> <p>Знаком с основными возможностями используемых для тестирования инструментов.</p>	<p>Умеет в автоматическое тестирование по методу белого ящика (unit).</p>
<p>Работа в команде</p>		
<p>Может выполнять поставленные задачи по технической формулировке.</p>	<p>Может выполнять поставленные задачи по функциональной формулировке.</p> <p>Может составить техническую формулировку на основе функциональной.</p> <p>Может самостоятельно давать оценку по времени и сложности при наличии функционального или технического описания.</p> <p>При несовпадении оценки с реальностью, может провести анализ для выявления причин.</p>	<p>Может выполнять поставленные задачи при полном отсутствии формулировок.</p> <p>Может проанализировать требование заказчика "Хочу, чтоб было круто" и составить на его основе функциональную формулировку.</p> <p>Может декомпозировать сложную задачу на ряд простых.</p> <p>В состоянии делегировать сложные задачи и контролировать их выполнение.</p>

В состоянии делегировать простые задачи и контролировать их выполнение.

Умеет в анализ производительности команды и способен выявлять проблемные моменты.