

Это — конструктор роадмапов

Роадмап у каждого свой. Я хочу стать более востребованным разработчиком, ты — стать менеджером, она — выучить клевую технологию. Поэтому делать один роадмап на всех людей, объединенных одной профессией, — не круто.

Мы расскажем, как тебе собрать свой собственный план обучения.

Вот 6 шагов

1

- **Найди 50 вакансий**, которые похожи на работу твоей мечты. Пожалуйста, не ограничивайся веб-пайтон-разработкой, снаружи тоже есть жизнь.

2

- **Выпиши требования** по всем вакансиям и категоризируй их. У нас получилось 20 кластеров. Это и есть блоки роадмапа.

3

- **Добавь туда то, что нужно именно тебе.** Хочешь выучить Хаскель? Писать микросервисы на Эликсире? Научиться, наконец, удаленно управлять офисной кофеваркой? Все — в список блоков. Если ты, конечно, вправду готов этим заниматься.

4

- **Нарисуйте граф зависимостей** для ваших блоков. Кажется, что пытаться в аналитику не стоит, если ты не знаешь SQL. Если вы не джуниор, редко таких зависимостей много, но их учет как раз и делает роадмап роадмапом.

5

- **Допишите каждый блок под вас.** Если технология касается работы — займитесь на ее применение. Если для фана — придумайте пет-проект. Найдите, где будете учиться. Заложите время. Проставьте дедлайны. Ну, планирование. Тайм-менеджмент. Скучно, но нужно.

6

- **Поздравляем, вы великолепны.** Как бонус — ты теперь знаешь, из чего состоят интересующие тебя области, какие там есть базворды, и как они связаны. Этого достаточно, чтобы сойти за эксперта на каком-нибудь митапе.

Пример для сеньор-веб-питониста

Мы сделали 2 из 6 выше представленных шагов для вакансий опытного веб-разработчика на Python. Дальше уже ты сам: только так получится создать персональный роадмап.

Проанализировали 80 актуальных вакансий, объединили требования в кластеры (получился 21 кластер), оставили только те блоки, которые упоминаются больше, чем в 5 вакансиях (осталось 16), и отсортировали по частоте (сверху самые часто упоминаемые).

Инфраструктура

Что это: понимание того, как код выполняется в боевом окружении, умение настроить деплой, CI/CD, логи, очереди, умение работать с очередями и все остальное, что касается того, в каком окружении исполняется код.

В каком объеме нужно знать: как минимум уметь осуществить базовую настройку деплоя, уметь чинить типичные поломки и проводить базовый тюнинг инфраструктуры. Как максимум полное понимание всех технологий и умение работать с облачными провайдерами так, чтобы настроить инфраструктуру в одиночку.

Базоворды: docker, docker-compose, Azure, AWS, GCP, CI/CD, gitlab, teamcity, Drone, Jenkins, Kubernetes, OpenStack, Ansible, MLFlow, Traefik, Nginx, Prometheus, Grafana, Elasticsearch, Logstash, Kibana, RabbitMQ, ZMQ, grpc, nameko.

Веб-фреймворки и сопутствующие технологии

Что это: умение работать с конкретными фреймворками для создания веб-приложений и знание их типичного окружения: ORM, ETL, работа с сетью и проч.

В каком объеме нужно знать: начиная от того уровня знаний, на котором вы можете комфортно разрабатывать код в заданном фреймворке и его инфраструктуре, и заканчивая глубоким пониманием принципов работы используемых технологий.

Базоворды: Flask, Django, Django REST framework, Django ORM, Django-rq, aiohttp, FastAPI, SQLAlchemy, pydantic, Celery, Redis, Sanic, Sphinx, Attrs, httpx, alembic, tornado, Falcon.

Хранилища данных

Что это: умение работать с разными хранилищами данных — как SQL, так и NoSQL.

В каком объеме нужно знать: как минимум уметь пользоваться на продвинутом уровне (сложные запросы, оптимизация, понимание принципов использования), как максимум — понимание принципов работы конкретной технологии под капотом.

Базворды: PostgreSQL, SQL (columnar vs row storage, SQL parsing, query planning, indexes, partitioning, join algorithms, filter pushdown optimizations, HLL), MySQL, Redis, MongoDB, PostGIS, ClickHouse, Cassandra.

Теоретические знания

Что это: знание того, как работают вещи в мире веба: интернет, провайдеры, облака, микросервисы, сеть, хайлоад, архитектура и все-все-все.

В каком объеме нужно знать: от среднего уровня (кажется, что без него не получится удовлетворить остальным требованиям в этом роадмапе) до глубокого понимания того, как работает, например, асинхронность в Python.

Базворды: HTTP, network, multiprocessing, WSGI, asyncio, containerization, OOP, gRPC, SOLID, SOAP, microservices architecture, design patterns.

Тестирование

Что это: умение писать юнит- и интеграционные тесты и решать основные проблемы, связанные с этим.

В каком объеме нужно знать: начиная от опыта написания юнит-тестов и заканчивая умением с нуля в одиночку настроить весь процесс TDD.

Базворды: unit testing, TDD, pytest.

Работа в команде

Что это: все, что делает вас хорошим командным игроком: от умения пользоваться Гитом и Джирой до понимания аджайла, умения проводить код-ревью или писать стайл-гайды.

В каком объеме нужно знать: если у вас есть пара лет опыта работы в такой команде, то, скорее всего, вы удовлетворяете минимальным требованиям этого пункта.

Базворды: Jira, Confluence, BitBucket, git, distributed team, agile, development process, style guides.

Другие языки программирования

Что это: часто от сеньора ждут, что он будет знать не только Python. Во-первых, в стеке компании могут быть другие языки, а во-вторых, это правда расширяет кругозор. Это редко бывает обязательным требованием, почти всегда это опциональный пункт.

В каком объеме нужно знать: от знакомства и умения сравнить с любимым Питоном до уровня мидл-разработчика.

Базворды: Rust, C, C++, Ruby, JS, TS, Java, Go, Perl, Lua, Kafka, c#, PHP, Kotlin.

Архитектура

Что это: умение выбрать архитектуру проекта целиком, отдельного куска или отдельного модуля.

В каком объеме нужно знать: начиная с базовых принципов и заканчивая полноценными знаниями для построения и масштабирования нагруженных приложений.

Базворды: serverless, highload, microservice architecture, SOA, KISS, DRY, SOLID.

Продвинутый пользователь ПК

Что это: умение пользоваться ОС — командная строка, настройка локального окружения, знание основ работы unix. Это не шутка, это правда пишут в требованиях для сеньора!

В каком объеме нужно знать: вы можете развернуть типичный проект локально, грепнуть логи и выполнить другие повседневные задачи на своем ПК.

Базворды: bash, GNU toolchain, packaging, git, linux.

Английский

Что это: вы должны уметь коммуницировать не только на русском, но и на английском.

В каком объеме нужно знать: обычно, если компания целиком находится в СНГ, то от вас хотят "технический английский" (читать документацию, что-то криво писать). Иначе может произойти что угодно от требования подтверждения знаний английского не ниже B1 до требования бегло говорить без акцента (если, например, предстоит общаться с клиентами или заказчиками).

Фронтенд

Что это: знания того, кто и как использует ваш API и что происходит в браузере.

В каком объеме нужно знать: от азов HTML/CSS, чтобы поправить съехавшую кнопку в админке, до полноценного умения разрабатывать современный фронтенд на каком-нибудь React.

Базворды: HTML, CSS, JavaScript, React, Vue, Vuex.

Коммуникация

Что это: все, что связано с говорением словами через рот.

В каком объеме нужно знать: от аргументации своего решения задачи до корпоративных принципов общения с внешними заказчиками.

Азы machine learning и data engineering

Что это: понимание того, что происходит в отделе DS, их архитектуры и проблематики.

В каком объеме нужно знать: обычно в базовом: верхнеуровневое понимание того, что как работает, какие бывают фреймворки и зачем они нужны, желание в этом разбираться.

Базворды: PyTorch, TensorFlow.

Дизайн API

Что это: умение построить нормальный API: целостный, безопасный, логичный.

В каком объеме нужно знать: достаточно для того, чтобы с нуля спроектировать и поддерживать актуальность API, внешнего или внутреннего.

Базворды: API design, REST, GraphQL, Swagger, openapi.

Писать читаемый код

Что это: вы пишете код, с которым потом можно жить.

В каком объеме нужно знать: в зависимости от культуры и специфики компании это или не нужно вообще или нужно уметь об этом что-то сказать. Наверное, это многое говорит о нашей отрасли.

Сделайте себе роадмап и живите счастливо

Теперь у тебя есть все, чтобы собрать свой
личный роадмап.

Пара советов напоследок:

Не используй пример выше как свой роадмап. В нем нет
того, что нравится именно тебе, а чем персональнее
роадмап, тем он эффективнее.

Не закливайся на одной области. Современная
разработка — это куда больше, чем перекладывание json на
каком-нибудь Flask.

Поделитесь вашим роадмапом в группе MoscowPython
(<https://www.facebook.com/groups/367203686641943>).
Чем больше таких штук есть в открытом доступе, тем лучше
будет наше сообщество.



Moscow ★
Python Conf++
2021