

TRAINEE.ТЕОРИЯ

В теоретическом блоке есть набор тем со ссылками на ресурсы/материалы для самообучения, также отмечены моменты, на которые необходимо обратить внимание при изучении.

Linux

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	nano / vim	Владеть на базовом уровне стандартным текстовым редактором nano или vim
2.	bash	уметь писать bash скрипты и разбираться в синтаксисе bash
3.	SSH	уметь работать с SSH

Python

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	Decorators	Для чего нужны декораторы
2.	Decorator with arguments	Могут ли декораторы принимать аргументы
3.	Property Decorator	Как работает декоратор property?
4.	append() and extend()	Различия между append и extend
5.	Lambda	Что такое lambda функция? Где её применяют?
6.	is vs ==	В чём отличие между сравнением через is и ==?
7.	"if smth is not None" vs "if not smth"	Почему проверка "if something is not None" более надёжна и правильна, чем "if something"?
8.	zip, filter, map, reduce	Знать, для чего нужны и как работают команды zip, filter, map, reduce.

9. №	type hint ТЕМА	type hint и чем они полезны ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
10.	Работа с файлами	Освоить работу с файлами в python
11.	context manager	Что такое контекстный менеджер? Как можно применить не только для работы с with open?
12.	time	Знать и уметь работать с библиотекой time
13.	datetime	Знать и уметь работать с библиотекой datetime
14.	os	Знать и уметь работать с библиотекой os
15.	sys	Знать и уметь работать с библиотекой sys
16.	Classes and Objects / Классы	Что такое __init__ и для чего он применяется? Что такое self? Чем отличается класс от экзепляра класса? Зачем нужны атрибуты класса и атрибуты экземпляра класса?
17.	@classmethod и @staticmethod	Что такое @classmethod и @staticmethod?
18.	Dunder or magic methods in Python Advanced Python: What Are Magic Methods? Руководство по магическим методам в Питоне	Знать основные магические методы
19.	Inheritance	Выполнить задачу на наследование
20.	Abstract classes	Понимать, что из себя представляют абстрактные классы
21.	ООП	Что такое ООП? Что такое инкапсуляция? Что такое полиморфизм? Что такое наследование?

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
22	Генераторы	Для чего нужны? Где применяются? Какая разница между генератором и итератором? Какую функцию выполняет инструкция yield?

PostgreSQL

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	Transactions	Что такое транзакции? Для чего они нужны?
2.	Indexes	Что такое индексы? Для чего они нужны?
3.	Оператор explain. Способы оптимизации	Как оценить опимизирован ли запрос?
4.	view	Что такое view? Для чего используется?
5.	Stored procedures	Что такое Stored procedures? Для чего используется?
6.	Условные операторы	Как использовать условные операторы в запросах? Какой синтаксис?
7.	Window functions	Что такое Window functions? Для чего используется?
8.	PostgreSQL & Docker	Уметь использовать связку PostgrSQL и Docker

Git

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
---	------	----------------------------

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	git-flow. Документация. Основы.	Получить представление, что такое git-flow Знать основные принципы: <ul style="list-style-type: none">- инициализация- жизненный цикл фичи- жизненный цикл релиза- жизненный цикл фикса При реализации проекта использовать git-flow

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	REST подход	Знать, для чего используется REST подход?
2.	CRUD операции	Что означает CRUD? Какие методы используются для каждой соответствующей операции?

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	Документация	Знать основные команды Уметь создавать окружение и устанавливать зависимости
2.	Примеры	Знать основные примеры применения
3.	Сравнение с другими окружениями.	Понимать отличия от venv
4.	Импорт и экспорт окружения.	Уметь импортировать и экспортировать окружения

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
---	------	----------------------------

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	<u>Основы тестирования.</u> Виды <u>отличия.</u>	Знать основные виды тестирования Понимать отличия между функциональными, юнит и интеграционным тестами
		Українська (uk) ▾ Коломієць Андрій
2.	<u>unittest</u> <u>Примеры использования.</u>	Уметь использовать библиотеку unittest
3.	<u>Mock</u> <u>Примеры использования.</u>	Что такое Mock? Для чего он нужен?

Flask

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	<u>Project structure</u>	
2.	Routing: <ul style="list-style-type: none">- <u>simple</u>- <u>with variables</u>- <u>HTTP methods</u>	
3.	<u>Static files serving</u>	
4.	<u>Templates</u>	
5.	<u>Handling request data</u>	
6.	<u>DB usage</u> <ul style="list-style-type: none">- <u>Connection</u>- <u>CRUD operations</u>	

№	Operations - SQL Alchemy ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
7.	Responses: - Redirects - Status codes	
8.	Cookies	
9.	Sessions	
10.	Logging	
11.	Testing	

Docker

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	Docker	<p>Познакомиться с экосистемой контейнеров на основе Docker</p> <p>Уметь работать с основными командами Docker:</p> <ul style="list-style-type: none">- загрузка, сборка, удаление образа- создание, запуск, остановка, удаление контейнера- просмотр установленных образов- просмотр запущенных контейнеров <p>Уметь работать с инструкциями Dockerfile</p> <p>Уметь работать с основными командами docker-compose:</p> <ul style="list-style-type: none">- сборка, запуск, остановка приложения- запуск, остановка контейнеров- просмотр запущенных контейнеров <p>Уметь работать с файлом конфигурации docker-compose.yml</p>

Nginx

№	ТЕМА	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
1.	Установка	Уметь устанавливать и использовать основные команды Nginx
2.	Конфигурация	Уметь сконфигурировать и запустить сервер
3.	Обработка запросов и регулярные выражения	<p>Понимать как Nginx обрабатывает запросы</p> <p>Регулярные выражения и приоритеты</p>

№	ТЕМА <u>Имена сервера</u>	ПОСЛЕ ПРОЧТЕНИЯ НЕОБХОДИМО
4.	<u>Управление Nginx</u>	Уметь работать с терминалом и управлять Nginx
5.	<u>Логирование</u>	Уметь настраивать логирование
6.	<u>Прокси-сервер</u>	Понимать и уметь настроить проксирование
7.	<u>Развертывание WSGI приложения (опционально)</u>	Понимать отличия Gunicorn от Nginx и получить навык их совместного конфигурирования