

1. [Теоретический минимум по Computer Science, Фило Владстон Феррейра](#)

1. 4.09.23 - 10.09.23

1. Основы
2. Вычислительная сложность
3. Стратегия
4. Данные

2. 11.09.23 - 17.09.23

1. Алгоритмы
2. Базы данных
3. Компьютеры
4. Программирование + заключение, приложения

2. [Теоретический минимум по Computer Science. Сети, криптография, data science, Фило Владстон Феррейра](#)

1. 18.09.23 - 24.09.23

1. Связи
2. Обмен данными

2. 25.09.23 - 01.10.23

1. Безопасность
2. Анализ данных

3. 16.10.23 - 22.10.23

1. Машинное обучение
2. Шаблоны
3. Приложения

3. [Чистый код, Роберт Мартин](#)

1. 23.10.23 - 29.10.23

1. Чистый код
2. Содержательные имена
3. Функции
4. Комментарии

2. 30.10.23 - 05.11.23

1. Форматирование
2. Объекты и структуры данных
3. Обработка ошибок

- 4. Границы
- 5. Модульные тесты
- 6. Классы

3. 06.11.23 - 12.11.23

- 1. Системы
- 2. Формирование архитектуры
- 3. Многопоточность

4. 13.11.23 - 19.11.23

- 1. Последовательное очищение
- 2. Внутреннее строение JUnit
- 3. Переработка SerialDate
- 4. Запахи и эвристические правила

4. [Python к вершинам мастерства, Лучано Рамальо](#)

1. 20.11.23 - 26.11.23

- 1. Модель данных в языке Python
- 2. Массив последовательностей
- 3. Словари и множества

2. 27.11.23 - 03.12.23

- 1. Unicode-текст и байты
- 2. Построители классов данных
- 3. Ссылки на объекты, изменяемость и повторное использование

3. 04.12.23 - 10.12.23

- 1. Функции как полноправные объекты
- 2. Аннотации типов в функциях
- 3. Декораторы и замыкания

4. 11.12.23 - 17.12.23

- 1. Реализация паттернов проектирования с помощью полноправных функций
- 2. Объект в духе Python
- 3. Специальные методы для последовательностей
- 4. Интерфейс, протоколы и ABC

5. 18.12.23 - 24.12.23

- 1. Наследование: к добру или к худу
- 2. Еще об аннотациях типов
- 3. Перегрузка операторов

6. **25.12.23 - 31.12.23**

1. Итераторы, генераторы и классические сопрограммы
2. Блоки with, match и else
3. Модели конкурентности в Python

7. **01.01.24 - 07.01.24**

1. Конкурентные исполнители
2. Асинхронное программирование

8. **08.01.24 - 14.01.24**

1. Динамические атрибуты и свойства
2. Дескрипторы атрибутов
3. Метапрограммирование классов

5. [Asyncio и конкурентное программирование, Мэттью Фаулер](#)

1. **15.01.24 - 21.01.24**

1. Первое знакомство с asyncio
2. Основы asyncio
3. Первое приложение asyncio

2. **22.01.24 - 28.01.24**

1. Конкурентные веб-запросы
2. Неблокирующие драйверы баз данных
3. Счетные задачи

3. **29.01.24-04.02.24**

1. Решение проблем блокирования с помощью потоков
2. Потоки данных
3. Веб-приложения
4. Микросервисы

4. **05.02.24 - 11.02.24**

1. Синхронизация
2. Асинхронные очереди
3. Управление процессами
4. Продвинутое использование asyncio

6. [Чистая Архитектура, Роберт Мартин](#)

1. **12.02.24 - 18.02.24**

1. Что такое дизайн архитектура?
2. История о двух ценностях
3. Обзор парадигм

4. Структурное программирование
5. Объектно-ориентированное программирование
6. Функциональное программирование
7. Принцип единственной ответственности

2. 19.02.24 - 25.02.24

1. Принцип открытости / закрытости
2. Принцип подстановки Барбары Лисков
3. Принцип разделения интерфейсов
4. Принцип инверсии зависимости
5. Компоненты
6. Связность компонентов
7. Сочетаемость компонентов
8. Что такое архитектура
9. Независимость

3. 26.02.24 - 03.03.24

1. Границы: проведение разделяющих линий
2. Анатомия границ
3. Политика и уровень
4. Бизнес-правила
5. Кричащая архитектура
6. Чистая архитектура
7. Презентаторы и скромные объекты
8. Неполные границы
9. Уровни и границы
10. Главный компонент
11. Службы: большие и малые

4. 04.03.24 - 10.03.24

1. Границы тестов
2. Чистая встраиваемая архитектура
3. База данных - это деталь
4. Веб - это деталь
5. Фреймворки - это деталь
6. Практический пример: продажа видео
7. Недостающая глава

8. Приложение

7. [Высоконагруженные приложения. Программирование, масштабирование, поддержка, Мартин Клепман](#)

1. 11.03.24 - 17.03.24

1. Надежные, масштабируемые и удобные в сопровождении приложения
2. Модели данных и языки запросов

2. 18.03.24 - 24.03.24

1. Подсистемы хранения и извлечения данных
2. Кодирование и эволюция
3. Репликация

3. 25.03.24 - 31.03.24

1. Секционирование
2. Транзакции

4. 01.04.24 - 07.04.24

1. Проблемы распределенных систем
2. Согласованность и консенсус

5. 08.04.24 - 14.04.24

1. Пакетная обработка

6. 15.04.24 - 21.04.24

1. Поточковая обработка
2. Будущее информационных систем

8. [Python. Разработка на основе тестирования, Гарри Персиваль](#)

1. 22.04.24 - 28.04.24

1. Настройка Django с использованием функционального теста
2. Расширение функционального теста при помощи модуля unittest
3. Тестирование простой домашней страницы при помощи модульных тестов
4. И что же делать со всеми этими тестами?
5. Сохранение вводимых пользователем данных: тестирование баз данных

2. 29.04.24 - 05.05.24

1. Усовершенствование функциональных тестов: обеспечение изоляции и удаление методов sleep
2. Работа в инкрементном режиме

3. Придание привлекательного вида: макет, стилевое оформление сайте и что тут тестировать
4. Тестирование развертывания с использованием промежуточного сайта

3. 06.05.24 - 12.05.24

1. Переход к развертыванию, готовому к эксплуатации
2. Автоматизация развертывания с помощью Fabric
3. Разделение тестов на многочисленные файлы и обобщенный помощник ожидания
4. Валидация на уровне базы данных
5. Простая форма
6. более развитые формы

4. 13.05.24 - 19.05.24

1. Пробуем окунуться, очень робко, в JavaScript
2. Развертывание нового программного кода
3. Аутентификация пользователя, импульсное исследование и внедрение его результатов
4. Использование имитация для тестирования внешних зависимостей или сокращение дублирования
5. Тестовые фикстуры и декоратор для явных ожиданий

5. 20.05.24 - 26.05.24

1. Отладка на стороне сервера
2. Завершение приложения "Мои списки": TDD с подходом снаружи внутрь
3. Изоляция тестов и "слушание своих тестов"
4. Непрерывная интеграция
5. Социально значимый кусок, шаблон проектирования "Страница" и упражнения для читателя
6. Быстрые тесты, медленные сетсы и горячие поля

6. 27.05.24 - 02.06.24

1. Повинуйтесь Билли-тестировщику!
2. Приложение A: PythonAnywhere
3. Приложение B: Представления на основе классов в Django
4. Приложение C: Обеспечение работы серверной среды при помощи Ansible

5. Приложение D: Тестирование миграций базы данных
6. Приложение E: Разработка на основе поведение BDD
7. Приложение F: Создание REST API: JSON, Ajax и имитирование на JavaScript
8. Приложение G: Django-Rest-Framework
9. Приложение H: Шпаргалка
10. Приложение I: Что делать дальше
11. Приложение J: Примеры исходного кода