

## Список вопросов к зачёту

1. Преимущества и недостатки Python, области его применения.
2. Кортежи. Основные функции, методы, операторы для работы с кортежами.
3. Списки. Основные функции, методы, операторы для работы со списками. Срезы.
4. Словари. Понятие ключей и значений. Создание словарей. Основные функции, методы, операторы для работы со словарями.
5. Какие есть способы форматирования строк в Python? Приведите примеры.
6. Специфика работы библиотеки pickle. Отличия pickle от JSON.
7. Как работает механизм сериализации и десериализации объектов в Python?
8. Какие существуют кодировки файлов? Почему важно правильно указывать кодировку при работе с файлами?
9. Сравните списковые включения и генераторные выражения. В чем их сходства и различия? Приведите примеры использования каждого из них.
10. Способы передачи аргументов в функцию.
11. Что такое фикстура? Для чего она нужна?
12. Правило LEGB.
13. Почему следует избегать модификации изменяемых аргументов?
14. Назначение и синтаксис конструкции yield в Python.
15. Что такое аннотации типов в Python? Зачем они нужны?
16. Что такое декоратор? Синтаксис и примеры использования.
17. Что такое множественное наследование? Какие проблемы могут возникнуть при множественном наследовании?
18. Как работают @property в Python? Чем они отличаются от обычных методов?
19. Как работает механизм перегрузки операторов в Python? Когда это полезно?
20. Что такое методы класса и статические методы? Чем они отличаются от методов экземпляра?
21. Что такое контекстные менеджеры и зачем они нужны? Объясните принцип работы оператора with.
22. Как создать собственный контекстный менеджер с помощью класса?
23. Обработка исключений с помощью блоков try, except.
24. Как Python обрабатывает исключения? В чем разница между raise и assert?

25. Как создать собственные классы исключений в Python? Для чего это может потребоваться? Приведите примеры.
26. Что такое Global Interpreter Lock (GIL)? Его влияние на многопоточность и способы обхода ограничений.
27. Что такое асинхронное программирование? В чем его отличие от синхронного? Приведите примеры, где асинхронное программирование предпочтительнее синхронного.
28. Объясните ключевые понятия: Future, Task, Event Loop. Как они взаимодействуют в асинхронных программах на Python? Приведите примеры использования этих понятий.
29. Что такое корутины (coroutines)? Как они связаны с асинхронностью? В чем их отличие от обычных функций?
30. В чем отличие корутин (coroutines) от генераторных функций?
31. Что такое async и await? Для чего они используются? Как они влияют на выполнение кода?
32. Как обрабатывать исключения в асинхронном коде? Какие особенности у обработки исключений в асинхронных функциях?
33. Как организовать асинхронную обработку нескольких задач?
34. Какие есть паттерны проектирования для асинхронных приложений?
35. В чем разница между параллельным и асинхронным программированием?