

100 вопросов на собеседование Python

Основы программирования на Python

1. Что такое переменная в Python?
2. Как объявить и инициализировать переменную?
3. Какие типы данных есть в Python?
4. Что такое список в Python и как его создать?
5. Как добавить элемент в список?
6. Как удалить элемент из списка?
7. Что такое кортеж и чем он отличается от списка?
8. Как создать кортеж?
9. Что такое словарь и как его создать?
10. Как добавить элемент в словарь?
11. Как удалить элемент из словаря?
12. Что такое множество в Python и как его создать?
13. Как добавить элемент в множество?
14. Как удалить элемент из множества?
15. Как узнать длину строки, списка, кортежа или словаря?
16. Что такое условные операторы в Python?
17. Как используется оператор if?
18. Как использовать оператор elif?
19. Как использовать оператор else?
20. Как работают циклы for в Python?
21. Как работают циклы while в Python?
22. Что такое функция в Python?
23. Как определить функцию в Python?
24. Как передавать параметры в функцию?
25. Как вернуть значение из функции?
26. Что такое lambda-функция?
27. Чем отличаются локальные и глобальные переменные?
28. Как работает оператор return?
29. Что такое модуль в Python и как его импортировать?

30. Как работает конструкция try-except?
31. Как выбрасывать исключения?
32. Что такое итераторы и генераторы?
33. Как работает генератор списка (list comprehension)?
34. Что такое декоратор и как его использовать?
35. Как работает функция map()?
36. Как работает функция filter()?
37. Как работает функция reduce()?
38. Что такое объектно-ориентированное программирование (ООП)?
39. Как создать класс в Python?
40. Как создать объект класса?
41. Что такое метод класса?
42. Как работают наследование и полиморфизм?
43. Что такое магические методы (например, init)?
44. Что такое инкапсуляция и как её реализовать?
45. Что такое абстрактные классы?
46. Как работают свойства (property) в Python?
47. Как работают статические методы и методы класса?
48. Что такое self в методах класса?
49. Как проверить тип объекта?
50. Как преобразовать строку в число и наоборот?

Работа с базами данных и SQL

51. Что такое база данных?
52. Какие виды баз данных вы знаете?
53. Что такое SQL?
54. Какие операции можно выполнять с помощью SQL?
55. Как создать таблицу в базе данных?
56. Как вставить данные в таблицу?
57. Как обновить данные в таблице?
58. Как удалить данные из таблицы?
59. Как выбрать данные из таблицы?
60. Что такое JOIN в SQL?
61. Какие типы JOIN существуют?

62. Что такое индекс в базе данных?
63. Что такое первичный ключ?
64. Что такое внешний ключ?
65. Как создать индекс для таблицы?
66. Как сделать выборку уникальных значений?
67. Что такое транзакция в базе данных?
68. Как работают команды COMMIT и ROLLBACK?
69. Как создать и удалить базу данных?
70. Что такое нормализация базы данных?
71. Какие уровни нормализации вы знаете?
72. Что такое денормализация?
73. Как соединить Python с базой данных?
74. Какие библиотеки Python используются для работы с SQL?
75. Как установить соединение с базой данных в Python?
76. Как выполнить SQL-запрос из Python?
77. Как обработать результаты запроса в Python?
78. Что такое ORM (Object-Relational Mapping)?
79. Какие ORM-библиотеки вы знаете?
80. Как работает SQLAlchemy?
81. Как создать модель в SQLAlchemy?
82. Как вставить данные с использованием SQLAlchemy?
83. Как обновить данные с использованием SQLAlchemy?
84. Как удалить данные с использованием SQLAlchemy?
85. Как сделать выборку данных с использованием SQLAlchemy?

Работа с виртуальными машинами

86. Что такое виртуальная машина?
87. Какие типы виртуализаций вы знаете?
88. Что такое Docker?
89. Как работает Docker?
90. Что такое образ (image) в Docker?
91. Как создать Docker-образ?

92. Что такое контейнер в Docker?
93. Как запустить контейнер из Docker-образа?
94. Как остановить Docker-контейнер?
95. Как удалить Docker-контейнер?
96. Как использовать Dockerfile?
97. Как работает команда docker-compose?
98. Какие команды Docker вы знаете?
99. Как просмотреть запущенные контейнеры?
100. Как подключиться к работающему контейнеру?

Бонус: 33 вопроса по парсингу

Requests

1. Что такое библиотека requests и для чего она используется?
2. Как отправить GET-запрос с использованием библиотеки requests?
3. Как отправить POST-запрос с использованием библиотеки requests?
4. Как передавать параметры в запросе requests?
5. Как добавить заголовки в запрос requests?
6. Как работать с cookies в библиотеке requests?
7. Как обрабатывать ошибки при отправке запросов с requests?
8. Как получить содержимое ответа в виде строки или байтов?

BeautifulSoup

9. Что такое библиотека BeautifulSoup и для чего она используется?
10. Как создать объект BeautifulSoup из HTML-документа?
11. Какие парсеры можно использовать с BeautifulSoup?

12. Как найти элемент по тегу с использованием BeautifulSoup?
13. Как найти элемент по классу или ID с использованием BeautifulSoup?
14. Как найти все элементы по тегу с использованием BeautifulSoup?
15. Как извлечь текст из элемента с использованием BeautifulSoup?
16. Как извлечь атрибут элемента (например, href или src) с использованием BeautifulSoup?

Selenium

17. Что такое Selenium и для чего он используется?
18. Как инициализировать веб-драйвер в Selenium?
19. Как открыть веб-страницу с использованием Selenium?
20. Как найти элемент на странице с использованием Selenium?
21. Как взаимодействовать с элементами (например, кликнуть, ввести текст) с использованием Selenium?
22. Как ждать появления элемента на странице с использованием Selenium?
23. Как переключаться между окнами или вкладками в Selenium?
24. Как сделать скриншот страницы с использованием Selenium?

Scrapy

25. Что такое Scrapy и для чего он используется?
26. Как создать новый проект в Scrapy?
27. Как создать нового паука (spider) в Scrapy?
28. Как определить стартовые URL в пауке Scrapy?
29. Как парсить ответ страницы в методе parse паука Scrapy?
30. Как сохранять данные, собранные пауком Scrapy?

31. Как настроить промежуточные слои (middlewares) в Scrapy?
32. Как обрабатывать ошибки и перенаправления в Scrapy?
33. Как использовать селекторы XPath и CSS в Scrapy для извлечения данных?