Расширенные возможности Python.sa_Pyt-p

1. Что такое асинхронность?
2. Как работает параллелизм?
3. Какую роль играет модуль asyncio в Python?
4. Какие ключевые методы используются в asyncio?
5. Для чего используются корутины?
6. Какой метод asyncio позволяет программе продолжать выполнение до завершения асинхронной операции?
7. Зачем использовать потоки?
8. Где выполняются задачи в асинхронном коде?
9. В какой строке находится ошибка в коде?
10. Какую функцию выполняет команда asyncio.ensure_future?
11. Что делает эта часть кода? for site in ["site 1", "site 2", "site 3"]:
 parsers = [] asyncio.ensure_future(parser(site))
 parsers.append(asyncio.gather(*parsers))
12. Какая функция используется для запуска нескольких задач одновременно?
13. Что делает функция async def get_all_pages(sitename)?
14. Какую функцию нужно вызывать внутри функции get_page_date?
15. Какие типы данных могут быть переданы в функцию async def get_all_pages(sitename)?
16. Что является ключевым элементом асинхронного программирования в Python?
17. Найдите неверную пару ключевых слов в коде:
18. Что пропущено в следующей строке кода? start_time = time.()

19. Какой метод используется для получения данных о странице?
20. В какой строке допущена ошибка синтаксиса?
21. Какой метод используется для проверки наличия цикла обработки событий в асинхронном коде Python?
22. Как называется декоратор в библиотеке asyncio, который позволяет функциям становиться асинхронными без явного использования ключевого слова async?
23. Что такое потоки?
24. Какой метод используется для запуска потока в классе Thread?
25. Как остановить выполнение потока?
26. Что делает метод join()?
27. Для чего используются блокировки (Lock)?
28. В чем разница между Lock и RLock?
29. Что такое ThreadPoolExecutor?
30. Как планировать выполнение задачи в ThreadPoolExecutor?
31. Что делает метод submit() в ThreadPoolExecutor?
32. Что делает функция `def parser(site_name)`?
33. Какой метод используется для объединения результатов двух потоков?
34. Как изменяется значение переменной `seconds`?
35. Какое выражение используется для преобразования времени в секунды?
36. Какая библиотека используется для работы с потоками?
37. Что означает CustomThread в коде?

38 Какой метод класса CustomThread необходимо переопределить для выполнения действий в потоке?

39 Как происходит создание новых экземпляров класса CustomThread?

40 Какие изменения вносятся в код после создания объектов потоков?

41 Какие дополнительные действия выполняются в потоках после их запуска?

42 Какая функция используется для создания блокировки?

43 Какой метод используется для захвата блокировки?

44 Какой метод используется для освобождения блокировки?

45 В каком порядке выполняются команды при вызове функции increment?

46 Как должны вызываться функции increment и decrement?

47 Какую функцию выполняет метод add_done_callback в классе ThreadPoolExecutor?

48 В каком порядке выполняются задачи?

49 Какой метод используется для добавления функции в очередь задач?

50 Какой метод используется для экспорта данных в формат Batch?

51 Какой параметр является лишним в следующей строке кода? asyncio.ensure_future(async def start():

52 Что произойдет при выполнении следующего фрагмента кода? with concurrent.futures.ThreadPoolExecutor(max_workers=30) as pool:
pool.map(export_bath, range(100000))

53 Что такое ORM?

54 Какую функциональность предоставляет SQLAlchemy?

55 Как подключиться к базе данных PostgreSQL с использованием SQLAlchemy?

56 Какой метод используется для создания сессии в SQLAlchemy?

57 Какой класс используется для создания таблиц в SQLAlchemy?

58 Какие преимущества дает использование ORM?

59 Что такое sessionmaker в SQLAlchemy?

60 Что такое execute в SQLAlchemy?

61 Что такое query в SQLAlchemy?

62 Как обновить значение столбца в одной записи?

63 Как удалить одну запись из таблицы?

64 Как выбрать несколько записей с условием?

65 Как обновить значения нескольких столбцов в одной записи?

66 Как выбрать все записи из таблицы?

67 Как добавить новую запись в таблицу?

68 Типы отношений

69 Как указать имя столбца для внешней ссылки в отношении?

70 Как включить автоинкремент в столбце ID?

71 Каким образом создаются отношения между таблицами в SQLAlchemy?

72 Что такое sessionmaker в SQLAlchemy?

73 Типы отношений

74 Какие из перечисленных функций могут использоваться для работы с базой данных в Pydantic?

75 Какие из нижеперечисленных классов используются для моделирования таблиц в SQLAlchemy?

76 Что делает метод `create_all()` в SQLAlchemy?

77 Какой декоратор используется для создания эндпоинта в FastAPI, чтобы обрабатывать GET-запросы?

78 Какой параметр декоратора используется для определения модели данных для возвращаемого API-ответа в FastAPI?

79 Какой декоратор используется для выполнения функции при запуске приложения в FastAPI?

80 Какую роль играет параметр decorator response_model в FastAPI?

81 Что такое кэширование?

82 Какая функция отвечает за инициализацию кэша в FastAPI?

83 Какой декоратор используется для задания времени жизни кэша?

84 Напишите имя параметра URL для эндпоинта `/hello/{Name}`:

85 Какой из перечисленных методов HTTP используется для получения содержимого ресурса?

86 Что произойдет, если отправить запрос методу GET к эндпоинту, который не существует?

87 Какой HTTP статус код соответствует успешному завершению операции?

88 Что происходит в методе `create_all` класса `BaseTable.metadata.create_all`?

89 Что происходит при вызове import BaseTable?

90 Какой путь, по умолчанию, используется для подключения к серверу?

91 Как определяется пользователь для подключения к базе данных?

92 Какую роль выполняет функция async def on_startup в приложении?

93 Какой метод используется для запуска сессии?

94 Какой класс отвечает за создание и управление соединениями с базой данных?

95 Какую роль играет параметр `expire=30` в настройках кэша?

96 Какой метод используется для получения всех записей из таблицы пользователей?

97 Какое ключевое слово используется для определения функции-обработчика в FastAPI?

98 Какой формат данных используется для передачи данных между методом `call` и методом `init`?

99 Какой декоратор используется для кеширования результатов запросов в FastAPI?

100 Какой из перечисленных методов может использоваться для добавления новых пользователей в базу данных?

101 Какой из перечисленных классов предоставляет интерфейс для взаимодействия с файловой системой?

102 Что такое асинхронное программирование?

103 Какой ключевой элемент используется для создания асинхронных функций в Python?

104 Что делает метод `await`?

105 Какой из следующих методов используется для создания и запуска асинхронной задачи в Python?

106 Что такое `ThreadPoolExecutor`?

107 Какое ключевое слово используется для определения корутины в Python?

108 Какой метод используется для добавления задачи в очередь в `ThreadPoolExecutor`?

109 Какой из следующих методов блокирует основной поток до завершения работы потока?

110 Что происходит при использовании `asyncio.gather()`?

111 Какой из следующих подходов подходит для выполнения задач параллельно?

112 Какой из следующих методов используется для запуска асинхронной функции?

113 Что происходит, если функция помечена как `async`?

114 Какой метод используется для ожидания завершения асинхронного задания?

115 Какой метод `asyncio` используется для создания задачи?

116 Какой из следующих фреймворков лучше всего подходит для асинхронной веб-разработки в Python?

117 Какой из следующих операторов используется для управления временной задержкой в асинхронном коде?

118 Какой метод возвращает объект Future?

119 Какой из следующих операторов используется для обработки исключений в асинхронных функциях?

120 Какой из следующих методов позволяет запускать несколько корутин одновременно?

121 Что произойдет при выполнении следующего кода? x = [1, 2, 3] x.append([4, 5]) print(x)

122 Что произойдет при выполнении следующего кода? def f(x): return x + 1 print(f(5))

123 Какой вывод программы при выполнении следующего кода? x = 10 def foo(): x = 5 return x print(foo())

124 Какой результат будет выдан программой при выполнении этого кода? class MyClass: def __init__(self, value): self.value = value def get_value(self): return self.value obj = MyClass(10) print(obj.get_value())

125 Что произойдет при выполнении следующего кода? for i in range(5): print(i, end=' ')

126 Какой будет результат выполнения этой программы? x = {1, 2, 3} x.add(2) print(len(x))

127 Какой метод используется для создания асинхронного цикла событий в Python?

128 Какой из следующих методов SQLAlchemy используется для создания асинхронного соединения с базой данных?

129 Какой метод используется для выполнения асинхронного запроса в SQLAlchemy?

130 Какой из следующих вариантов является правильным способом создания асинхронной сессии в SQLAlchemy?

131 Какой из следующих методов используется для обработки ошибок в асинхронном коде?

132 Какой результат будет у переменной `result` после выполнения следующего кода? import asyncio async def delayed_return(): await asyncio.sleep(1) return "done" async def main(): result = await delayed_return() print(result) asyncio.run(main())

133 Что произойдет при выполнении следующего кода? import asyncio async def delayed_print(): await asyncio.sleep(1) print("Hello") async def main(): await asyncio.gather(delayed_print(), delayed_print()) asyncio.run(main())

134 Что такое FastAPI?

135 Какой из следующих декораторов используется для создания маршрута в FastAPI?

136 Какой метод HTTP используется для создания нового ресурса в FastAPI?

137 Какой декоратор используется для обработки PUT-запросов в FastAPI?