

Перечень вопросов к экзамену
по дисциплине «Разработка интернет-приложений»
(осенний семестр 2020 года)

1. Условные операторы в языке Python.
2. Цикл for в языке Python.
3. Цикл while в языке Python.
4. Операторы break и continue в языке Python.
5. Использование комментариев в языке Python.
6. Обработка исключений в языке Python.
7. Создание функций в языке Python. Оператор return.
8. Лямбда-выражения в языке Python.
9. Функции в языке Python. Использование параметров *args.
10. Функции в языке Python. Использование параметров **kwargs.
11. Функции в языке Python. Совместное использование параметров *args и **kwargs.
12. Использование виртуальных окружений для языка Python.
13. Создание модулей и пакетов в языке Python.
14. Импорт модулей и пакетов в языке Python.
15. Проверка на истинность для произвольных типов данных в языке Python.
16. Тип данных Null Object (NoneType) в языке Python.
17. Логический тип данных в языке Python.
18. Операторы and и or как операторы «короткого замыкания» (short-circuit operators).
19. Операторы сравнения в языке Python.
20. Целый тип данных в языке Python.
21. Представление целых чисел в различных системах счисления в языке Python.
22. Использование битовых операций для целых чисел в языке Python.
23. Вещественный тип данных в языке Python.
24. Использование модуля decimal для вещественного типа данных в языке Python.
25. Использование модуля fractions для вещественного типа данных в языке Python.

26. Основные операции с целыми и вещественными числами в языке Python.
27. Комплексный тип данных в языке Python.
28. Создание строк в языке Python с использованием апострофов, кавычек и тройных кавычек (апострофов).
29. Основные методы для работы со строками в языке Python.
30. Индексирование строк (срезы, slices) в языке Python.
31. Форматирование строк (строковая интерполяция) в языке Python.
32. Байтовые строки (bytes) в языке Python.
33. Объявление списков в языке Python.
34. Индексирование списков (срезы, slices) в языке Python.
35. Обход элементов списка в обратном порядке с использованием индексирования списков в языке Python.
36. Описание матриц с использованием списков в языке Python.
37. Как реализовать конкатенацию списков в языке Python?
38. Как реализовать умножение (повторение) списков в языке Python?
39. Как проверить наличие элемента в списке в языке Python?
40. Как добавить элементы в конец списка в языке Python?
41. Как добавить элемент в заданную позицию списка в языке Python?
42. Как получить индекс элемента с заданным значением в списке в языке Python?
43. Использование метода sort для сортировки списка в языке Python.
44. Использование функции sorted для сортировки списка в языке Python.
45. В чем разница между использованием метода sort и функции sorted для сортировки списка в языке Python?
46. Как реализовать удаление элемента (элементов) из списка в языке Python?
47. Использование списка в режиме стека в языке Python.
48. Использование списка в режиме очереди в языке Python.
49. Объявление кортежей в языке Python.
50. Основные сходства и различия между списками и кортежами в языке Python.
51. Преобразование из кортежа в список и из списка в кортеж в языке Python.
52. Способы доступа к элементам кортежей в языке Python.

53. Как проверить наличие элемента в кортеже в языке Python?
54. Как получить индекс элемента с заданным значением в кортеже в языке Python?
55. Как реализовать конкатенацию кортежей в языке Python?
56. Как реализовать умножение (повторение) кортежей в языке Python?
57. Сортировка кортежей в языке Python.
58. Напишите фрагмент программы, показывающий, что списки являются изменяемым (mutable) типом данных, а кортежи являются неизменяемым (immutable) типом данных в языке Python.
59. Объявление множеств в языке Python.
60. Основные сходства и различия между списками и множествами в языке Python.
61. Основные операции над множествами в языке Python.
62. Объявление словаря в языке Python.
63. Основные сходства и различия между списками и словарями в языке Python.
64. Способы получения значений словаря в языке Python.
65. Использование list comprehension в языке Python.
66. Использование set comprehension в языке Python.
67. Использование dict comprehension в языке Python.
68. Использование функции map для обработки коллекций в языке Python.
69. Использование функции filter для обработки коллекций в языке Python.
70. Использование функции zip для обработки коллекций в языке Python.
71. Использование функций foldl и foldr для обработки коллекций в языке Python.
72. Для чего используются декораторы функций в языке Python. Приведите пример простого декоратора.
73. Как создать декоратор с параметрами для функции в языке Python.
74. Что такое итераторы и для чего они используются в языке Python. Приведите пример простого итератора.
75. Что такое генераторы и для чего они используются в языке Python. Какие две разновидности генераторов существуют в языке Python.
76. Генераторные выражения в языке Python.
77. Генераторные функции в языке Python.

78. Что такое контекстные менеджеры и для чего они используются в языке Python. Приведите пример контекстного менеджера.
79. Создание контекстного менеджера на основе класса с методами `__enter__` и `__exit__` в языке Python.
80. Создание контекстного менеджера на основе библиотеки `contextlib` (конструкция `yield`) в языке Python.
81. Использование функции `open` для открытия файла в языке Python.
82. Открытие файла с использованием контекстного менеджера в языке Python.
83. Чтение и обработка текстовых файлов в языке Python.
84. Чтение и обработка файлов в формате JSON в языке Python.
85. Представление класса в качестве словаря (`dict`) в языке Python.
86. Создание класса в языке Python. Приведите пример создания класса.
87. Как создать конструктор класса в языке Python?
88. Как создать в классе на языке Python метод, принадлежащий объекту?
89. Как создать метод класса на языке Python?
90. Как создать статический метод в классе на языке Python?
91. Как реализуется наследование классов в языке Python?
92. Как вызвать конструктор базового класса в языке Python?
93. Как документировать классы на языке Python с помощью строк документации? Как получить строку документации с использованием конструкции `__doc__`?
94. Список специальных атрибутов в языке Python? Назначение основных атрибутов.
95. Как произвести дизассемблирование кода в языке Python?
96. Аннотации типов в языке Python. Для чего они используются, приведите примеры.
97. Абстрактные классы в языке Python.
98. Использование свойств (`property`) в классах на языке Python.
99. Как реализуется множественное наследование классов в языке Python?
100. Проблема ромбовидного наследования. Приведите пример на языке Python.
101. Классы данных (записи) в языке Python.
102. Как реализуется перегрузка операторов в языке Python.
103. Как реализуется контроль доступа к элементам класса в языке Python.

104. Что такое модульное тестирование, для чего оно используется?
105. Особенности подхода test-driven development (TDD).
106. Особенности подхода behavior-driven development (BDD).
107. Использование языка Gherkin для описания сценариев в подходе behavior-driven development (BDD).
108. Основные сходства и различия между подходами test-driven development (TDD) и behavior-driven development (BDD).
109. Особенности подхода IoC/DI (Inversion of Control/Dependency injection).
110. Что такое принципы SOLID, для чего они используются.
111. Принципы SOLID, принцип S.
112. Принципы SOLID, принцип O.
113. Принципы SOLID, принцип L.
114. Принципы SOLID, принцип I.
115. Принципы SOLID, принцип D.
116. Что такое Mock-объекты, как они используются при создании тестов.
117. Библиотека unittest. Пример создания простого теста.
118. Библиотека unittest. Методы assert...
119. Библиотека unittest. Метод subTest.
120. Библиотека unittest. Методы setUp и tearDown.
121. Трехзвенная архитектура веб-приложения.
122. Шаблон проектирования Model-View-Controller.
123. Шаблон проектирования Model-View-Presenter.
124. Основные компоненты технологии WWW (HTML, HTTP, URI).
125. Для чего используются URI, URL, URN, IRI. Как эти технологии соотносятся между собой.
126. Что такое схемы URI (URL) и для чего они используются.
127. Схемы URI (URL). Схема HTTP.
128. Протокол HTTP. Общая структура протокола.
129. Протокол HTTP. Структура запроса.
130. Протокол HTTP. Структура ответа.

131. Основные методы протокола HTTP. Сходства и различия между методами GET и POST.
132. Коды состояния (статус-коды) протокола HTTP.
133. Сессии в HTTP-протоколе. Cookie.
134. Язык разметки HTML. Структура HTML-документа.
135. Язык разметки HTML. Основные типы конструкций: тэги, элементы, атрибуты, комментарии.
136. Язык разметки HTML. Основные элементы разметки текста и заголовков.
137. Язык разметки HTML. Создание гипертекстовых ссылок с использованием абсолютных и относительных URI (URL).
138. Язык разметки HTML. Элементы создания списков.
139. Язык разметки HTML. Элементы создания таблиц.
140. Язык разметки HTML. Элементы создания изображений.
141. Язык разметки HTML. Элементы семантической разметки.
142. Язык разметки HTML. Элементы секции HEAD.
143. Технология CSS (Cascading Style Sheets). Стилиевые правила и их структура.
144. Технология CSS. Способы включения стилей в документ HTML.
145. Технология CSS. Класс в качестве селектора (класс-селектор).
146. Технология CSS. Id в качестве селектора (Id-селектор).
147. Технология CSS. Контекстные селекторы.
148. Технология CSS. Псевдоклассы и псевдоэлементы.
149. Технология CSS. Основные принципы каскадирования.
150. Технология CSS. Определение значения специфичности при каскадировании.