



Программирование на языке Python.dor_БАК_24-172-Б

- 1 Первая версия языка Python была выпущена в ...
- 2 Новая версия ... несовместима с предыдущими версиями
- 3 Преимуществом Python перед некоторыми другими языками программирования является ...
- 4 Python 3.0 был выпущен в ...
- 5 Для ... данных в Python используется функция print();
- 6 Байт-код Python хранится в файле с расширением ...
- 7 PyCharm Community Edition представляет собой ...
- 8 Установите соответствие между понятиями и их определениями:
- 9 Установите правильную последовательность процессов выполнения программы на Python:
- 10 Неверно, что Python поддерживает парадигму ... программирования
- 11 Переменные в Python могут содержать ...
- 12 Метод, который используется для удаления переменной в Python, -...
- 13 При работе с кортежами в Python изменение значения элемента приводит к ...
- 14 В Python метод ... служит для разделения строки по пробелам
- 15 В Python метод ... служит для преобразования строки в нижний регистр
- 16 Установите соответствие между типами данных и их описаниями:
- 17 Расположите операторы по их приоритету выполнения, начиная с самого высокого:





- 18 Python оператор ... служит для проверки равенства двух объектов
- 19 Метод ... в Python служит для получения среза списка с первого до третьего элемента включительно
- 20 Метод range(5) в Python вернет ...
- 21 Простые инструкции в Python ...
- 22 В Python оператор ... позволяет пропустить оставшуюся часть кода в текущей итерации цикла и перейти к следующей
- 23 В Python инструкция assert используется для ...
- 24 Функция zip в Python ...
- 25 Оператор в Python, используемый для прерывания выполнения цикла, – это ...
- 26 В Python при выполнении тернарного оператора x if условие else y, если условие ложно, результатом будет ...
- 27 В Python при использовании конструкции while важно удостовериться, что ...
- 28 Функция range в Python генерирует ...
- 29 Расположите строки кода в таком порядке, чтобы в результате исполнения этого кода была выведена сумма квадратов четных чисел в диапазоне от 1 до 10:
- 30 Сопоставьте каждое исключение в Python с его определением:
- 31 Создать пакет в Python – значит ...
- 32 В Python оператор ... позволяет вернуть значение из функции
- 33 В Python для импортирования модуля math необходимо использовать конструкцию ...
- 34 В Python импортировать все содержимое модуля os можно с помощью конструкции ...
- 35 Модуль ... предоставляет функции для работы с регулярными выражениями в Python
- 36 В Python для импорта модуля с псевдонимом используется конструкция ...





- 37) В Python вызвать функцию из модуля, который находится в пакете, можно с помощью конструкции ...
- 38) В Python правильное использование аннотации типов в функции - ...
- 39) Расположите строки кода в правильном порядке, чтобы в результате исполнения этого кода была произведена пузырьковая сортировка на Python:
- 40) Сопоставьте функции в Python с описаниями их использования:
- 41) Основной принцип объектно-ориентированного программирования: «...»
- 42) В Python внутри методов класса первый параметр – всегда ...
- 43) В Python механизм, позволяющий создать новый класс на основе существующего, унаследовав все его атрибуты и методы, – это ...
- 44) Объект, являющийся атрибутом другого объекта, – это ...
- 45) Механизм для изменения поведения операторов для пользовательских классов – это ... операторов
- 46) Метод, предоставляющий строковое представление объекта для пользователя, – это метод ____
- 47) При множественном наследовании Python использует для поиска методов алгоритм:
- 48) Способность объекта использовать методы и атрибуты различных классов – это ...
- 49) Соотнесите методы перегрузки операторов в Python с их функциями:
- 50) Расположите фрагменты кода, определяющие иерархию классов, в том порядке, в котором Python будет искать метод в классе X (порядок разрешения методов (MRO) для класса X):
- 51) Различие между классами Tk и TopLevel в Tkinter заключается в том, что Tk ...
- 52) Модуль для работы с Tkinter – это ...
- 53) Метод для создания главного цикла обработки событий в Tkinter – это ...





- 54 Виджет для выбора элемента из выпадающего списка в модуле ttk – это ...
- 55 Модуль Tkinter, добавляющий улучшенные виджеты и стили, – это модуль ...
- 56 Класс, предназначенный для создания и управления группами виджетов в Tkinter, – это класс ...
- 57 Менеджер геометрии для размещения виджетов в конкретной позиции в родительском виджете:
- 58 Метод для привязки функции к событию нажатия клавиши в Tkinter – это ...
- 59 Упорядочьте шаги для создания простого окна в Tkinter, от начального к завершающему:
- 60 Соотнесите виджеты Tkinter с их функциональным назначением:
- 61 Преимущество Django перед Flask – ...
- 62 Стандартный метод для запуска сервера разработки в Django – ...
- 63 Команда для создания нового приложения в Django – ...
- 64 Основная задача веб-фреймворков в Python – ...
- 65 Тип веб-фреймворков как набора инструментов в Python, который позволяет разработчику создать полноценное веб-приложение от начала до конца, – это ... веб-фреймворки
- 66 в Python метод функции path(), который указывается для подключения маршрутов уровня приложения, – это ...
- 67 Класс модели Django, используемый для работы с таблицами базы данных, – это ...
- 68 Компонент Django, отвечающий за взаимодействие с базой данных, – это ...
- 69 Расположите в правильном порядке части URL-адреса, начиная с первой:
- 70 Соотнесите функции Django с их описаниями:
- 71 Уровень изоляции, который используется по умолчанию в SQLite, – ...





- 72) Уровень изоляции, который блокирует базу данных для всех объектов чтения и записи до завершения транзакции, – ...
- 73) SQL-запрос, который используется для создания таблицы в SQLite, – ...
- 74) Ключевой аспект, который отличает SQLite от MySQL, – ...
- 75) SQL-оператор для вставки данных в таблицу ...
- 76) Метод ... в sqlite3 используется для извлечения всех строк из результата запроса
- 77) Метод, который используется для подключения к базе данных SQLite в Python, – это ...
- 78) Атрибут `row_factory`, который используется для доступа к данным по именам столбцов, – это ...
- 79) Упорядочьте этапы работы с транзакциями в SQLite:
- 80) Соотнесите описание функционала с правильным классом или функцией:
- 81) Блоки кода в Python обозначаются ...
- 82) Основным преимуществом интерпретируемых языков, таких как Python, является ...
- 83) ... – это фреймворк Python, который используется для создания сложных веб-приложений ...
- 84) Виртуальная машина Python (PVM) отвечает за ...
- 85) Неверно, что фреймворк ... относится к веб-разработке на Python
- 86) Для ... данных в Python используется функция `input ()`
- 87) Команда, используемая для проверки установленной версии Python, – это `python --help` – ...
- 88) Jupyter Notebook используется как среда для ...
- 89) Концом инструкции в Python является ...
- 90) Установите соответствие между понятиями и их определениями:





- 91 Система управления пакетами для Python, которая используется для установки и управления библиотеками и зависимостями: ...
- 92 Символ использующийся для создания комментариев в Python, – это ...
- 93 Команда для установки Python через Homebrew на macOS: ... `install python3`
- 94 Установите правильную последовательность процессов выполнения программы на Python:
- 95 При присваивании одного списка другой переменной, ...
- 96 Функция, возвращающая длину строки в Python
- 97 В результате выполнения кода (см. ниже) будет выведено ...
`a = 5`
`b = 3`
`result = a << 1`
`print(result)`
- 98 Метод для добавления одного элемента в конец списка
- 99 Метод, который удаляет и возвращает последнюю пару «ключ:значение» из словаря в виде кортежа
- 100 Строка `text` после выполнения кода (см. рисунок ниже) будет выглядеть ...
`text = "Python"`
`text = text * 3`
- 101 Метод для преобразования списка в строку с возможностью указания разделителя – ...()
- 102 Метод, преобразующий строку в список символов, – ...()
- 103 Оператор, выполняющий деление с остатком, – ...
- 104 Метод для получения уникальных элементов из списка – ...
- 105 Установите соответствие между типами данных и их описаниями:
- 106 Расположите операторы в порядке убывания приоритета выполнения, начиная с самого высокого приоритета:
- 107 Обработку исключений в Python позволяет реализовать инструкция `try except` ...
- 108 Оператор ... используется для перебора элементов списка в Python
- 109 Инструкция `pass` в Python ...





- 110) Оператор в Python, который используется для проверки нескольких условий поочередно, – это if-elif-...
- 111) В результате выполнения приведенного ниже кода на экран будет выведено: ... `d = {1: 'a', '2': ['b', {'c': 3}], (5, '6'): ('f', 'g')}`
`print(len(d))`
- 112) Установите соответствие операторов в Python и их характеристик:
- 113) В результате выполнения приведенного ниже кода на экран будет выведено: ... `lst = [] print(lst[0])`
- 114) Оператор, используемый для создания контекста выполнения кода, – это оператор ...
- 115) В результате выполнения приведенного ниже кода на экран будет выведено: ... `try: print("Начало блока try") 1 / 0 except ZeroDivisionError: print("Возникло исключение ZeroDivisionError") finally: print("Блок finally выполнен")`
- 116) Установите правильную последовательность строк кода, чтобы в результате его исполнения была выведена сумма квадратов четных чисел в диапазоне от 1 до 10:
- 117) Исключение ... возникает при невозможности импортировать модуль или найти атрибут в модуле
- 118) Установите соответствие исключений в Python и их характеристик:
- 119) Python ищет модули для импорта в ...
- 120) Обновить импортированный модуль без перезапуска интерпретатора ...
- 121) Чтобы задать значение по умолчанию для параметра функции, необходимо ввести: ...
- 122) Добавление директории в путь поиска модулей: ...
- 123) Чтобы передать переменное количество аргументов функции, необходимо ввести: ...
- 124) Текстовый файл ... используется для управления зависимостями проекта в Python
- 125) Метод ... используется для получения текущего рабочего каталога в Python
- 126) Метод ... сортировки используется для сортировки списка в Python на месте





- (127) Обработка исключения при импорте модуля выглядит следующим образом: ...
- (128) Для импорта конкретной функцию или переменной из модуля используется конструкция ...
- (129) В результате выполнения приведенного ниже кода на экран будет выведено ... `def add(*args): return sum(args) print(add(1, 2, 3))`
- (130) Установите правильный порядок фрагментов кода, чтобы в результате его исполнения была произведена пузырьковая сортировка на Python
- (131) Сопоставьте функции в Python и их назначение:
- (132) Механизм включения объектов одного класса в качестве атрибутов другого класса – это ...
- (133) Декоратор для создания статического метода – ...
- (134) Метод, определенный внутри класса и связанный с самим классом, а не с его экземпляром, – это ...
- (135) Специальные функции, принимающие класс в качестве аргумента и возвращающие его измененную версию, – это ...
- (136) Процесс создания конкретных объектов на основе класса – это ...
- (137) Метод в Python, возвращающий строковое представление объекта для отладки, – это метод `__str__`
- (138) Метод, позволяющий в Python дочернему классу вызвать метод родительского класса, – это метод `super()`
- (139) Конструктор класса в Python автоматически вызывается при создании нового экземпляра и называется `__init__`
- (140) Декоратор `@...` в Python позволяет методам класса работать с самим классом, а не с его экземплярами
- (141) Для того чтобы в Python пользовательский класс поддерживал операцию индексации (например, `obj[0]`), необходимо реализовать метод `__getitem__`
- (142) Соотнесите методы перегрузки операторов в Python с их функциями:
- (143) Расположите фрагменты кода, определяющие иерархию классов, в том порядке, в котором Python будет искать метод в классе X (порядок разрешения методов (MRO) для класса X):





- 144 Основная роль цикла обработки событий в событийно-ориентированном программировании – ...
- 145 При запуске основного цикла обработки событий в Tkinter происходит ...
- 146 Базовый класс для всех виджетов в Tkinter – ...
- 147 Виджет в Tkinter, предназначенный для отображения какой-либо надписи без возможности редактирования пользователем, – это ...
- 148 Класс, предоставляющий методы для рисования графических элементов в Tkinter, – это ...
- 149 Метод, который размещает виджеты в виде блоков в Tkinter, – это ...
- 150 Виджет, используемый для создания кнопок в Tkinter, – это ...
- 151 Библиотека Python, которая позволяет создавать пользовательские интерфейсы с поддержкой мультитач и написана на сочетании Python и Cython, – это ...
- 152 Класс в Tkinter, который предоставляет возможность работы с таблицами и списками, – это ...
- 153 Параметр, используемый для привязки функции к событию нажатия кнопки при создании кнопки в Tkinter, – это ...
- 154 Установите правильный порядок шагов создания простого окна в Tkinter:
- 155 Соотнесите виджеты Tkinter с их функциональным назначением:
- 156 Файл URL маршрутов проекта, определяющий, какие контроллеры (views) будут обрабатывать запросы для каждого URL адреса, – это ...
- 157 Команда для создания новой миграции базы данных в Django – python manage.py: ...
- 158 Основной паттерн архитектуры, используемый в Django: ...
- 159 Способ подключения HTML-шаблонов в Django: ...
- 160 HTML-тег, отвечающий за создание гиперссылок, – это тег





- 161) Безопасный протокол передачи данных, который поддерживает шифрование посредством криптографических протоколов SSL и TLS, – это протокол ...
- 162) Раздел HTML-документа, содержащий основной контент страницы, – это раздел
- 163) Модуль или файл, в котором содержатся настройки приложения в проекте Django: ...
- 164) Файл `__init__.py` указывает на то, что папка является пакетом Python
- 165) Соотнесите команды Django с их назначением:
- 166) Соотнесите функции Django с их описаниями:
- 167) Расположите в правильной последовательности части URL-адреса, слева направо:
- 168) Функция, которая используется для выполнения SQL-запросов в Python с помощью `sqlite3`, – ...
- 169) Название функции в `sqlite3`, которая устанавливает объект `Row` в качестве типа для результирующего набора запроса, – ...
- 170) При вызове метода `rollback` в контексте `SQLite` ...
- 171) Метод, который необходимо вызвать для сохранения изменений в базе данных `SQLite` после выполнения запросов, – это метод ...
- 172) Команда `show tables` служит для получения списка таблиц в базе данных `MySQL`.
- 173) Тип данных, используемый для хранения строк в `SQLite`, – ...
- 174) Для обозначения позиционного параметра в SQL-запросе используется символ ...
- 175) Метод для выполнения запросов в виде списка строк – это метод ...
- 176) Функция для установления соединения с базой данных `SQLite` – это функция ...
- 177) Упорядочьте этапы работы с транзакциями в `SQLite`:
- 178) Соотнесите описание функционала с правильным классом или функцией:





- 179) Необходим скрипт на Python, который принимает от пользователя имя и выводит приветственное сообщение. Напишите данный скрипт.
- 180) Необходима программа на Python, которая правильно считывает возраст пользователя и выводит его год рождения. Приведите данную программу.
- 181) Необходима программа на Python, которая правильно считывает строку текста и выводит количество слов в этой строке. Приведите данную программу.
- 182) Необходим скрипт на Python, который верно заменяет символ «o» в строке Python на «\$». Напишите данный скрипт.
- 183) Необходим скрипт на Python, который удаляет повторяющиеся элементы из списка [2, 4, 5, 1, 2, 7, 8, 4, 1]. Напишите данный скрипт.
- 184) Необходима программа на Python, которая принимает от пользователя два числа и проверяет следующие условия (выводит результаты проверок): - Оба числа положительные. - Хотя бы одно из чисел отрицательное. - Оба числа делятся на 5 без остатка. Приведите данную программу.
- 185) Ниже приведен скрипт на Python: `string = "Python упрощает программирование с помощью простого синтаксиса"` `for index in range(len(string)):` `if index % 2 == 0:` `print(string[index], end="")` Что является результатом вывода этого скрипта?
- 186) Ниже приведен код на Python:
`month = int(input("Введите номер месяца (от 1 до 12): "))`
`if 1 <= month <= 12:` `if month in [12, 1, 2]:` `season = "Зима"`
`elif month in [3, 4, 5]:` `season = "Весна"`
`elif month in [6, 7, 8]:` `season = "Лето"` `else:`
`season = "Осень"` `print("Время года:", season)` `else:`
`print("Ошибка: введите номер месяца от 1 до 12.")` Что выполняет этот код?
- 187) Требуется программа на Python, которая запрашивает у пользователя число и пытается вывести его квадрат. Если пользователь вводит нечисловое значение, программа обрабатывает ошибку и выводит сообщение об ошибке. И в любом случае программа должна вывести сообщение о завершении работы. Приведите такую программу.
- 188) В Python существует функция, которая проверяет, содержится ли элемент в списке. Приведите эту функцию.





- 189) Ниже приведен следующий код в Python: `def infinite_generator():
 num = 1
 while True:
 yield num
 num += 1
def generator_2(num):
 for i in infinite_generator():
 yield num ** i
gen = generator_2(3)
for _ in range(4):
 result = next(gen)
 print(result)` Какой результат будет выведен при выполнении цикла в последней итерации?
- 190) Ниже приведен декоратор в Python. `def log_function_call(func):
def wrapper(*args, **kwargs):
 print(f"Функция {func.__name__} была вызвана с аргументами: {args} и {kwargs}")
 return func(*args, **kwargs)
return wrapper` Что делает данный декоратор?
- 191) Ниже приведен код, написанный на Python. `class Animal:
 def __init__(self, name):
 self.name = name
 print(f"Создано животное: {self.name}")
 def speak(self):
 return f"{self.name} издает звук"
class Dog(Animal):
 def __init__(self, name, breed):
 super().__init__(name)
 self.breed = breed
 print(f"Порода собаки: {self.breed}")
 def speak(self):
 return f"{self.name} лает"
class Cat(Animal):
 def speak(self):
 return f"{self.name} мяукает"
def make_animal_speak(animal):
 print(animal.speak())` # Создаем объекты `dog = Dog("Барбос", "Овчарка")
cat = Cat("Мурка")
make_animal_speak(dog)
make_animal_speak(cat)` Какой результат будет напечатан после выполнения данного кода?
- 192) Ниже приведен код на Python. `from abc import ABC, abstractmethod
class Animal(ABC):
 def __init__(self, name):
 self.name = name
 @abstractmethod
 def sound(self):
 pass
class Dog(Animal):
 def sound(self):
 return "Гав"
class Cat(Animal):
 def sound(self):
 return "Мяу"` # Создание экземпляра класса `Animal animal = Animal("Барбос")` Что произойдет при попытке создания экземпляра класса `Animal` в следующем коде?
- 193) Ниже приведен код на Python. `class Vector:
 def __init__(self, x, y):
 self.x = x
 self.y = y
 def __add__(self, other):
 if isinstance(other, Vector):
 return Vector(self.x + other.x, self.y + other.y)
 return NotImplemented
 def __repr__(self):
 return f"Vector({self.x}, {self.y})"` `v1 = Vector(1, 2)
v2 = Vector(3, 4)
result = v1 + v2
print(result)` Что произойдет при перегрузке оператора `+` для класса `Vector` в следующем коде?
- 194) Ниже приведен код с использованием Tkinter для создания интерфейса. `import tkinter as tk
def say_hello():
 print("Привет!")
root = tk.Tk()
root.title("Пример окна")
btn = tk.Button(root, text="Нажми меня", command=say_hello)
btn.pack()
root.mainloop()` Что произойдет, если выполнить данный код?





- 195) Ниже приведен код с использованием Tkinter. `import tkinter as tk`
`class MyApp: def __init__(self, master): self.master = master`
`master.title("MyApp") self.label = tk.Label(master,`
`text="Hello, Tkinter!") self.label.pack() self.button =`
`tk.Button(master, text="Click Me", command=self.change_text)`
`self.button.pack() def change_text(self):`
`self.label.config(text="Text Changed!") root = tk.Tk() app =`
`MyApp(root) root.mainloop()` Что произойдет при запуске приложения?
- 196) Ниже приведен код с использованием Tkinter. `import tkinter as tk`
`class MyApp: def __init__(self, master): self.master = master`
`master.title("MyApp") self.frame = tk.Frame(master)`
`self.frame.pack() self.canvas = tk.Canvas(self.frame, width=100,`
`height=100, bg='blue') self.canvas.pack() self.button =`
`tk.Button(self.frame, text="Draw", command=self.draw_rectangle)`
`self.button.pack() def draw_rectangle(self):`
`self.canvas.create_rectangle(10, 10, 90, 90, fill='red')` `root = tk.Tk() app =`
`MyApp(root) root.mainloop()` Что произойдет при запуске приложения?
- 197) Ниже представлен код на Python. `from django.db import models`
`class Person(models.Model): name =`
`models.CharField(max_length=100) age = models.IntegerField()`
`def __str__(self): return f"{self.name}, Age: {self.age}"` Что
делает метод `__str__` в модели Django Person в этом коде?
- 198) Ниже представлен код на Python. `from django.http import`
`HttpResponse from django.shortcuts import render def`
`greeting(request): name = request.GET.get('name', 'Гость') return`
`render(request, 'greeting.html', {'name': name})` Что произойдет,
если этот код выполнится при обращении к URL
`http://example.com/greeting?name=Иван?`
- 199) Ниже представлен код на Python. `from django.http import`
`HttpResponse from .models import Bb def index(request): s =`
`'Объявления\r\n\r\n' for bb in Bb.objects.order_by('-published'):`
`s += bb.title + '\r\n' + bb.content + '\r\n\r\n' return HttpResponse(s,`
`content_type='text/plain; charset=utf-8')` Какой результат будет
возвращен пользователю при запросе по URL, который вызывает
функцию `index`?





- 200) Ниже представлен код. `import sqlite3 # Подключение к базе данных SQLite conn = sqlite3.connect('example.db') cursor = conn.cursor() # Выполнение запросов SQLite cursor.execute('CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT)') cursor.execute('INSERT INTO users (name) VALUES (?)', ('Alice',)) conn.commit() # Проверка содержимого таблицы cursor.execute('SELECT * FROM users') rows = cursor.fetchall() # Закрытие соединения conn.close() print(rows)` Что выведет этот код?
- 201) Ниже представлен код, который использует именованные параметры в запросе к базе данных SQLite. `import sqlite3 def fetch_tasks_by_status_and_project_named(db_name, status, project_name): connection = sqlite3.connect(db_name) # Установка row_factory для получения данных в виде словарей connection.row_factory = sqlite3.Row cursor = connection.cursor() # Определение запроса с именованными параметрами и явным указанием колонок query = """ SELECT id, priority, description, status, deadline, completed, project FROM tasks WHERE status = :status AND project = :project """ # Выполнение запроса с использованием именованных параметров cursor.execute(query, {"status": status, "project": project_name}) rows = cursor.fetchall() for row in rows: print(f"ID: {row['id']}, Priority: {row['priority']}, Description: {row['description']}, " f"Status: {row['status']}, Deadline: {row['deadline']}, Completed: {row['completed']}, " f"Project: {row['project']}") connection.close() if __name__ == "__main__": fetch_tasks_by_status_and_project_named("projects_and_tasks.db", "pending", "Learn Python")` Что произойдет, если в запросе используются неверные имена именованных параметров (например, :stat вместо :status)?





- 202) Ниже представлен код, в котором используется управление транзакциями для добавления нового проекта и задач в базу данных. `import sqlite3 def demo_transactions(db_name): connection = sqlite3.connect(db_name) cursor = connection.cursor() try: cursor.execute('INSERT INTO projects (name, description, deadline) VALUES ('Project Y', 'Описание проекта Y', '2024-11-01')') cursor.execute('INSERT INTO tasks (priority, description, status, deadline, completed, project) VALUES (1, 'Задача 1 для Project Y', 'pending', '2024-10-15', NULL, 'Project Y'), (2, 'Задача 2 для Project Y', 'in progress', '2024-10-30', NULL, 'Project Y')') raise sqlite3.Error("Искусственная ошибка для демонстрации rollback.") connection.commit() print("Изменения сохранены в базе данных.") except sqlite3.Error as e: # В случае ошибки откат всех изменений print(f"Ошибка: {e}. Откат изменений...") connection.rollback() finally: connection.close() if __name__ == "__main__": demo_transactions('projects_and_tasks.db')` Что произойдет, если возникает ошибка при выполнении транзакции?
- 203) Блоки кода в Python обозначаются ...
- 204) Основным преимуществом интерпретируемых языков, таких как Python, является ...
- 205) ... – это фреймворк Python, который используется для создания сложных веб-приложений ...
- 206) Виртуальная машина Python (PVM) отвечает за ...
- 207) Неверно, что фреймворк ... относится к веб-разработке на Python
- 208) Для ... данных в Python используется функция `input()`
- 209) Команда, используемая для проверки установленной версии Python, – это `python --help` ...
- 210) Jupyter Notebook используется как среда для ...
- 211) Концом инструкции в Python является ...
- 212) Установите соответствие между понятиями и их определениями:
- 213) Система управления пакетами для Python, которая используется для установки и управления библиотеками и зависимостями: ...





- 214) Символ использующийся для создания комментариев в Python, – это ...
- 215) Команда для установки Python через Homebrew на macOS: ... `install python3`
- 216) Установите правильную последовательность процессов выполнения программы на Python:
- 217) При присваивании одного списка другой переменной, ...
- 218) Функция, возвращающая длину строки в Python
- 219) В результате выполнения кода (см. ниже) будет выведено ...
`a = 5 b = 3 result = a << 1 print(result)`
- 220) Метод для добавления одного элемента в конец списка
- 221) Метод, который удаляет и возвращает последнюю пару «ключ:значение» из словаря в виде кортежа
- 222) Строка `text` после выполнения кода (см. рисунок ниже) будет выглядеть ...
`text = "Python" text = text * 3`
- 223) Метод для преобразования списка в строку с возможностью указания разделителя – ...()
- 224) Метод, преобразующий строку в список символов, – ...()
- 225) Оператор, выполняющий деление с остатком, – ...
- 226) Метод для получения уникальных элементов из списка – ...
- 227) Установите соответствие между типами данных и их описаниями:
- 228) Расположите операторы в порядке убывания приоритета выполнения, начиная с самого высокого приоритета:
- 229) Обработку исключений в Python позволяет реализовать инструкция `try except` ...
- 230) Оператор ... используется для перебора элементов списка в Python
- 231) Инструкция `pass` в Python ...
- 232) Оператор в Python, который используется для проверки нескольких условий поочередно, – это `if-elif-...`





- 233) В результате выполнения приведенного ниже кода на экран будет выведено: ... `d = {1: 'a', '2': ['b', {'c': 3}], (5, '6'): ('f', 'g')}`
`print(len(d))`
- 234) Установите соответствие операторов в Python и их характеристик:
- 235) В результате выполнения приведенного ниже кода на экран будет выведено: ... `lst = [] print(lst[0])`
- 236) Оператор, используемый для создания контекста выполнения кода, – это оператор ...
- 237) В результате выполнения приведенного ниже кода на экран будет выведено: ... `try: print("Начало блока try") 1 / 0`
`except ZeroDivisionError:`
`print("Возникло исключение ZeroDivisionError") finally:`
`print("Блок finally выполнен")`
- 238) Установите правильную последовательность строк кода, чтобы в результате его исполнения была выведена сумма квадратов четных чисел в диапазоне от 1 до 10:
- 239) Исключение ... возникает при невозможности импортировать модуль или найти атрибут в модуле
- 240) Установите соответствие исключений в Python и их характеристик:
- 241) Python ищет модули для импорта в ...
- 242) Обновить импортированный модуль без перезапуска интерпретатора ...
- 243) Чтобы задать значение по умолчанию для параметра функции, необходимо ввести: ...
- 244) Добавление директории в путь поиска модулей: ...
- 245) Чтобы передать переменное количество аргументов функции, необходимо ввести: ...
- 246) Текстовый файл ... используется для управления зависимостями проекта в Python
- 247) Метод ... используется для получения текущего рабочего каталога в Python
- 248) Метод ... сортировки используется для сортировки списка в Python на месте





- (249) Обработка исключения при импорте модуля выглядит следующим образом: ...
- (250) Для импорта конкретной функцию или переменной из модуля используется конструкция ...
- (251) В результате выполнения приведенного ниже кода на экран будет выведено ...
`def add(*args):
 return sum(args)
print(add(1, 2, 3))`
- (252) Установите правильный порядок фрагментов кода, чтобы в результате его исполнения была произведена пузырьковая сортировка на Python
- (253) Сопоставьте функции в Python и их назначение:

