1. Генератор и итератор?

Итератор является более общей концепцией, чем генератор, и представляет собой любой объект, класс которого имеет методы __next__ и __iter__. Генератор – это итератор, который обычно создается путем вызова функции, содержащей не менее одного оператора yield. Это ключевое слово действует аналогично return, но возвращает объект-генератор.

Между ними существуют тонкие различия:

- Для генератора мы написали функцию, а для итератора можно использовать встроенные функции iter() и next().
- Для генератора используется ключевое слово yield для выдачи по одному объекту за раз.
- В генераторе может быть сколько угодно операторов yield.
- Генератор сохраняет текущее состояние локальных переменных (local variables) каждый раз, когда yield приостанавливает цикл (loop). Итератор не использует локальные переменные, он работает только с итерируемым объектом (iterable).
- Итератор можно использовать с помощью класса, а генератор — нет.
- Генераторы работают быстро, компактно и проще.
- Итераторы экономнее потребляют память.

2. Лямбда или анонимная функция?
Расскажите про выражения лямбда (lambda expressions). Где они могут пригодиться?

Если нужно написать функцию с одним выражением, то можно обойтись без определения и сделать ее анонимной. Выражение лямбда может принимать входные данные и возвращать значения. Пример функции, представленной в виде выражения лямбда:

В данном примере входные данные обозначаются переменными a и b. То есть, если a > b, то возвращается a, в противном случае возвращается b. В качестве аргументов используются 3 и 3.5.

Лямбда позволяет обойтись без входных данных.

В Python анонимные функции создаются при помощи лямбда-выражений. Такие выражения удобно использовать в местах, где ожидается функция с достаточно ограниченной задачей.

3. Типы данных

1. *None* (неопределенное значение переменной)
2. Логические переменные (*Boolean Type*)
3. Числа (*Numeric Type*)
    1. *int* – целое число
    2. *float* – число с плавающей точкой
    3. *complex* – комплексное число
4. Списки (*Sequence Type*)
    1. *list* – список
    2. *tuple* – кортеж
    3. *range* – диапазон
5. Строки (*Text Sequence Type*)
    1. *str*
6. Бинарные списки (*Binary Sequence Types*)
    1. *bytes* – байты
    2. *bytearray* – массивы байт
    3. *memoryview* – специальные объекты для доступа к внутренним данным объекта через protocol buffer
7. Множества (*Set Types*)
    1. *set* – множество
    2. *frozenset* – неизменяемое множество
8. Словари (*Mapping Types*)
    1. *dict* – словарь

4. Магические методы?

Так называют специальные методы, обрамленные двумя подчеркиваниями. Магические методы представляют простой способ заставить объекты вести себя аналогично встроенным типам. Это, в частности, позволяет стандартизировать поведение базовых операторов с экземплярами класса.

5. Разница между __new__ и __init__?

Метод __new__ используется, когда нужно управлять процессом создания нового экземпляра, а __init__ – когда контролируется его инициализация. Поэтому new возвращает новый экземпляр класса, а init – ничего.

6. Что такое id или id()?

Возвращает идентификатор переданного объекта, уникальный на время его существования.

7. Как получить список всех атрибутов объекта или что такое dir()?

В простейшем виде с помощью функции dir().
Для объектов, класс которых не определил __dir__(), функция попытается определить атрибуты по данным __dict__.
Возвращаемый список может включать не все атрибуты, особенно в случаях с переопределенным __getattr__().
Функция help() показывает строку документации и справку для ее аргумента
Функция dir() возвращает список, содержащий пространство имен в объекте

8. Что быстрее dict, list, set, tuple?

Средняя временная сложность поиска в множествах и словарях соответствует O(1), в случае последовательностей O(n). Кортежи – это неизменяемый тип, поэтому они могут давать выигрыш в скорости перед списками.

9. Какая разница между одинарным (_) и двойным (__) подчеркиванием? Почему игнорируются имена-идентификаторы, которые начинаются с символа подчеркивания?

В питоне не реализована концепция скрытой переменной (private variable), поэтому принято декларировать скрытые переменные первым символом в виде нижнего подчеркивания.

Одинарным подчеркиванием задаются частные переменные, функции, методы и классы. Двойное подчеркивание применяется для искажения имен атрибутов в классе (вызвать такой метод стандартным образом не получится).

10. Как в Python воплощены public, private, protected методы?
Этот вопрос напрямую связан с предыдущим. Все компоненты класса Python по умолчанию являются открытыми (public). Для защищенных (protected) методов по соглашению Python добавляется префикс одиночного подчеркивания, для закрытых (private) методов – префикс двойного подчеркивания

11. Что такое MRO?

MRO – сокращение от method resolution order. То есть это способ разрешения проблемы множественного наследования классов. Для конструирования линеаризации класса в версиях Python с новым стилем классов используется алгоритм C3 линеаризации.

12. Для чего в Python используются слоты?

Магический атрибут __slots__ позволяет задать ограниченный набор атрибутов, которыми будет обладать экземпляр класса. За счет такого ограничения можно повысить скорость работы при доступе к атрибутам и сэкономить место в памяти.

13. Что такое контекстные менеджеры? Где они применяются? Как создать свой контекстный менеджер? Что такое with в питоне?

Данная инструкция обеспечивает исполнение кода очистки после исполнения программы. Например, можно использовать ее для открытия файла, совершить с ним какие-то действия и автоматически закрыть файл после завершения работы. Аналогичным образом можно открывать соединение с базой данных и автоматически его закрывать. Код очистки исполняется даже в случае, когда появляется исключение (exception).

Контекстные менеджеры – это конструкции, которые упрощают работу с тем или иным интерфейсом. Например, работу с файлами или базами данных. Создаются они с помощью оператора with. Для создания собственного класса контекстного менеджера используется библиотека contextlib.

14. Что такое декораторы с параметрами?

В качестве декоратора можно использовать выражение, значение которого – функция, принимающая и возвращающая функцию. А значит, можно создавать декораторы с параметрами (фабрики декораторов). Подробно с примерами.

15. Чем отличаются и как используются декораторы @classmethod и @staticmethod?

Базовые декораторы classmethod, staticmethod используются для методов, определённых внутри классов. В метод класса первым аргументом передаётся класс. Аналогично метод экземпляра в первом аргументе получает сам экземпляр. Статичный метод используется в том случае, когда метод не имеет доступа к тому, что представляет собой класс или объект класса.

16. Что такое дескриптор?

Дескриптор – атрибут объекта, чьё поведение при доступе переопределяется методами __get__, __set__ и __delete__. Если определен хотя бы один из этих методов, объект становится дескриптором.

17. Что такое метаклассы в Python?

В Python классы являются объектами, поэтому они сами должны чем-то генерироваться. Эти конструкции представляют собой своеобразные «классы классов» и называются метаклассами. Примером встроенного метакласса является type. В основном метаклассы используются для создания API. Подробнее читайте в нашей публикации.

18. Как вы тестируете код? Что такое mock? Что такое Pytest?

Для модульного тестирования используется unittest или Pytest. Mock – это специальный модуль (ставший недавно частью стандартной библиотеки) для тестирования без существенной адаптации кода под тесты.

19. Как проводится отладка программ на Python?

В Python есть модуль pdb. Он позволяет пошагово провести отладку программы. В версии 3.7 появилась функция breakpoint(), которая также облегчает дебаггинг.

20. Что такое PEP8?

Соглашение о том, как писать код на Python. Нередко среда разработки после соответствующей настройки позволяет автоматически поддерживать программный код в соответствии с этими правилами или теми, что приняты в компании.

21. Какие есть программы для проверки стиля кода? Каковы их преимущества и недостатки?

Скажем, pylint, pyflakes. Наиболее популярные инструменты мы уже описали и сравнили.

22. В чём состоит отличие процессов от потоков?

Модули – это subprocess и threading. Использование нескольких процессов аналогично использованию нескольких независимых программ, обмен данными организован через каналы. Если приложение должно выполнять несколько задач в

одно и то же время, используются потоки (threads). В этом случае для ограничения доступа потоков к памяти в Python имеется блокировщик GIL.

23. Что такое AsyncIO? Когда его имеет смысл использовать?

В отличие от потоков, в AsyncIO переключение между сопрограммами происходит лишь тогда, когда сопрограмма ожидает завершения внешней операции. AsyncIO подойдет, если приложение большую часть времени тратит на чтение/запись данных, а не их обработку, то есть, например, для работы с веб-сокетами.

24. В чем заключается проблема циклических зависимостей? Как ее решить?

Циклические зависимости обычно служат признаком некачественного проектирования системы. Временное решение – перенести импорт во вложенные области видимости. В частности, определения функций.

25. Как реализовать шаблон Singleton на Python? Какие есть альтернативы?

Стандартный пример описан в примерах PEP-0318. Менее удобна для тестирования реализация через декоратор, элегантнее вариант через метаклассы.

26. Какие шаблоны проектирования вы еще знаете? Какими пользовались?

Фабричный метод, абстрактная фабрика, прототип, компоновщик, итератор.

27. Основные фичи питона?

Если питон оказался первым языком в опыте программирования, нужно иметь общее понимание о нем. Какие у него основные признаки:

- это интерпретируемый язык

- в нем динамическая типизация данных

- это объектно-ориентированный язык

- он лаконичный и внешне простой

- распространяется бесплатно

- у него большое сообщество

28. В чем разница между списками (list) и кортежами (tuple)?

Основная разница в том, что список может изменяться (mutable), а кортеж не может (immutable).

29. Как в питоне работает трёхместный (тернарный) оператор?

В питоне есть такие выражения:

[если верно] if [выражение] else [если неверно]

То есть, когда выражение верное (True), то исполняется код [если верно]. В остальных случаях исполняется код [если неверно]. Например:

30. Питон чувствителен к регистру?

Язык считается чувствительным к регистру в случае, если он различает имена "myname" и "Myname". То есть, если он отслеживает разницу регистра (между верхним и нижним). Посмотрим, как с этим в питоне.

31. Предельно допустимая длина идентификатора в питоне?

В питоне идентификатор может быть любой длины. Помимо этого есть несколько правил для присвоения имен:

- первым символом может быть нижнее подчеркивание (_), символы A-Z или a-z;

- остальная часть имени может состоять из символов A-Z/a-z/_/0-9;

- не забываем, что питон чувствителен к регистру;

- в качестве имени нельзя использовать ключевые слова (keywords):

and, def, False, import, not, True, as, del, finally, in, or, try, assert, elif, for, is, pass, while, break, else, from, lambda, print, with, class, except, global, None, raise, yield, continue, exec, if, nonlocal, return.

32. Как можно преобразовать строку (string) в нижний регистр (lowercase)?

Для этого используется метод lower(). Для преобразования в верхний регистр (uppercase) используется метод upper(). Еще есть методы isupper() (все символы в верхнем регистре) и islower() (все символы в нижнем регистре), которые проверяют регистр всех символов имени. Еще есть метод istitle(), который проверяет строку на стиль заголовка (все слова должны начинаться с символа в верхнем регистре)

33. Для чего нужен pass (pass statement) в питоне? Зачем нужны break и continue?

Они используются для управления последовательностью операций: break останавливает исполнение цикла и переводит исполнение на следующий блок кода, continue как бы перепрыгивает на следующую итерацию цикла и не прекращает его исполнение. Иногда нужно, чтобы код не давал никакого результата и не показывал ошибку, например, если еще не готово, но нужно иметь синтаксический корректный код. Можно поставить pass. Кроме него есть break (break statement), которое разрывает цикл. Наконец, есть continue (continue statement), которое перешагивает на следующую итерацию.

34. Как получить список из всех ключей словаря (dictionary keys)?

На такие вопросы нужно отвечать детально, с примерами. Данная задача выполняется с помощью функции keys()

35. Что такое срез?

Срез — это методика, которая позволяет получить часть списка, кортежа или строки.

36. Как пишутся комментарии в питоне?

Для этого используется символ #. Все, что написано на строке после него, считается комментарием и игнорируется. Комментарии используются для объяснения цели написанного кода. Многострочных комментариев в прямом смысле слова в питоне нет.

37. Как проверить, что все символы строки относятся к алфавитно-цифровым?

Для этого используется метод isalnum().

38. Как перевести первый символ строки в верхний регистр?

Для этого есть метод capitalize()

39. Все знают, что сегодня питон в моде. Но истинное принятие новой технологии подразумевает понимание ее недостатков. Что вы можете сказать по этому поводу? Минусы python?

Какие в питоне есть ограничения:

- интерпретируемая природа питона снижает скорость исполнения программы

- его не выгодно использовать для мобильных устройств и браузеров

- будучи языком с динамической типизацией данных, он использует утиную типизацию; в связи с этим появляются ошибки исполнения (runtime errors);

- в нем слабо развиты возможности доступа к базам данных; поэтому питон не идеальный вариант для приложений с очень большими базами данных;

- низкие требования на входе, то есть свои силы в питоне может попробовать каждый; это иногда снижает качество кода;

- у питона индивидуально выраженный стиль.

40. Как в питоне узнать, в какой мы сейчас директории? Как узнать текущую директорию в питоне?

Для этого используется функция *os.getcwd()*. Она импортируется из модуля os

41. Что такое приглашение интерпретатора (interpreter prompt)?

Когда мы заходим в интерпретатор питона, то видим следующую строку:

>>>

42. Что нужно сделать, чтобы функция возвращала значение?

Для этого используется ключевое слово return.

43. Что такое блок?

Когда мы пишем предложение (statement), нам нужно завершить первую строку двоеточием, а под ним написать блок кода, который исполняется в рамках этого предложения. Каждая строка блока пишется с одинаковым отступом.

44. Если мы не поставим двоеточие в конце строки для цикла "do-while", он все равно сработает?

В питоне такой цикл не реализован. Это вопрос из тех, которые с подвохом, когда упоминают элементы других языков.

45. В каких областях питон имеет преимущество, лучше всего использовать?

Лучше всего питон использовать в следующих областях:

- веб-приложения

- графические интерфейсы пользователя для настольных ПК

- научные и арифметические приложения

- разработка ПО

- разработка программ обучения

- приложения для бизнеса

- сетевые приложения

- игры, 3D-графика

46. Можете назвать десять встроенных функций питона?

Функция complex() создает комплексное число

Функция eval() исполняет строку.0) — min(2,3)

Функция filter() отфильтровывает элементы, для которых заданное условие верно.

Функция format() помогает задать формат строки:

Функция hash() возвращает хэш-значение объекта:

Функция hex() преобразовывает число в шестнадцатеричное число:

Функция input() читает ввод и возвращает строку:

Функция len() возвращает число, показывающее длину строки:

Функция locals() возвращает словарь с локальной таблицей имен

Функция open() открывает файл

47. Как конвертировать список в строку?

Для этого подойдет метод join()

48. Как убрать из списка дубликат элемента?

Для этого можно конвертировать список во множество (set)

49. Можете объяснить жизненный цикл треда?

Общими словами, цикл выглядит так:

- сначала создается класс, который подменяет метод исполнения класса в треде, и создаем экземпляр (instance) для этого класса;

- вызываем start(), который готовит тред к исполнению;

- переводим тред в состояние исполнения;

- можно вызвать разные методы, например sleep() и join(), которые переводят тред в режим ожидания;

- когда режим ожидания или исполнения прекращается, другие ожидающие треды подготавливаются к исполнению;

- после завершения исполнения тред останавливается.

50. Что такое словарь (dictionary)?

Словарь содержит пары типа "ключ: значение"

Расскажите про арифметические операторы //, %, и **

Оператор // выполняет целочисленное деление и возвращает целую часть числа, полученного в результате операции

Оператор ** возводит в степень

Оператор % возвращает результат деления по модулю, то есть остаток после деления

51. Что вам известно про операторы сравнения в питоне?

Данные операторы сравнивают значения между собой.

Оператор "меньше" (<): если значение с левой стороны от оператора меньше, он возвращает True:

Оператор "больше" (>): если значение с левой стороны от оператора больше, он возвращает True:

Оператор "меньше или равно" (<=): если значение с левой стороны от оператора меньше значения с правой стороны или равно ему, он возвращает True:

Оператор "больше или равно" (>=): если значение с левой стороны от оператора больше значения с правой стороны или равно ему, он возвращает True:

Оператор равенства (==): если значения равны, он возвращает True:

Оператор неравенства (!=): если значения не равны, он возвращает True:

51. Что такое операторы присвоения в питоне?

Все арифметические операторы можно комбинировать с символом присвоения =, *=

52. Расскажите про логические операторы в питоне.

Всего их три: and, or, not. Как логический оператор можно использовать XOR ^

53. Что такое оператор принадлежности?

Это операторы in и not in. Они показывают, является ли одно значение частью другого.

54. Расскажите про операторы тождественности.

Операторы is и is not показывают, являются ли два значения идентичными.

55. Что такое битовые операторы?

Данные операторы выполняют операции в битовом формате.

% ^ >>

54. Что такое строка документации (docstring)?

Она вносится первой строкой в блок, определяющий содержание функции, класса или метода. Содержит описание их цели и способа исполнения. Обозначается тремя одинарными или двойными кавычками с каждой стороны.

__doc__

55. Как можно конвертировать строку в число?

Если строка содержит только числовые символы, можно использовать функцию int()

56. Как можно принять результат ввода на клавиатуре?

Если пользователь что-то вводит с помощью клавиатуры, можно использовать функцию input(). В качестве аргумента можно задать данной функции текст запроса на ввод. Результат ввода всегда является строкой.

57. Что такое рекурсия? Может ли рекурсия создавать сложности? Какие преимущества у рекурсии?

Это когда функция вызывает саму себя. При этом она должна иметь базовое условие, чтобы не создать бесконечный цикл

Разумеется:

- Приходится чаще вызывать функцию.
- Каждый вызов функции сохраняет переменную состояния в программном стеке, то есть растет потребление памяти, что в итоге может стать причиной переполнения памяти.
- Вызовы функции отнимают время.

Рекурсия помогает:

- экономить усилия на выполнение задачи,
- сократить объем кода по сравнению с циклами,
- легче воспринимать код.

58. Что делает функция zip()?

Возвращает итератор с кортежами:

В данном случае она совмещает элементы двух списков и создает из них кортежи. Работает не только со списками.

59. Как посчитать длину строки (string)?

Для этого вызываем функцию len()

60. Расскажите про генераторы списков (list comprehension).

Они позволяют создавать списки с помощью одной строки кода

61. Как можно получить все значения из словаря?

Для этого используется метод values()

62. Как можно переключить регистр строки?

Можно использовать метод swapcase(), предусмотренный для класса str

63. Для чего используется bytes()?

Это встроенная функция питона, которая возвращает неизменяемый байтовый объект

64. Что такое оператор контроля последовательности (control flow statement)?

Обычно программа в питоне начинает исполнение с первой строки. После нее программа однократно исполняет каждое предложение. Когда будет исполнено последнее предложение, программа прекращается. Также контроль последовательности помогает усложнить обычный порядок исполнения программы.

65. Как работать с числами, которые не входят в десятичную систему счисления?

В питоне можно вводить бинарные, восьмеричные и шестнадцатеричные числа.

Бинарные. Это числа, составленные из 0 и 1. Для ввода в бинарном формате, используется префикс 0b или 0B:

Число можно преобразовать в бинарный формат с помощью функции bin():

Восьмеричные числа могут состоять из цифр от 0 до 7, также используется префикс 0o или 0O:

Шестнадцатеричные числа могут состоять из цифр от 0 до 15, также используется префикс 0x или 0X:

66. Чем Python отличается от Java? Сравнение Python с другим языком

Если сравнивать Python и Java:

- Java быстрее

- Python использует отступы, а Java нужны скобки

- в Python динамическая типизация, а в Java — статическая

- Python — простой и лаконичный, а Java — многословный язык

- Python — интерпретируемый язык

- Java не зависит от используемой платформы

- в Java есть интерфейс JDBC, который улучшает доступ к базам данных

67. Как выйти из бесконечного цикла?

Можно нажать комбинацию клавиш Ctrl+C, которая прерывает исполнение.

68. Как исполняется код в питоне?

Файлы питона сначала компилируются в байткод, который затем исполняется

69. Расскажите, какой в питоне механизм передачи параметров.

В питоне используется передача параметров по ссылке. Если изменить параметр внутри функции, то это отразится на выводе функции. Однако, если использовать в качестве параметров литералы (строки, числа, кортежы), то они передаются по значению (потому что они не изменяемые).

70. Чем файл .pyc отличается от .py?

Оба файла содержат байткод, но .pyc является компилированной версией файла питона. Его байткод не зависит от платформы, поэтому он исполняется на всех платформах, которые поддерживают формат .pyc.

71. Что делает питон объектно-ориентированным? ООП

Он следует парадигме объектно-ориентированного программирования, которая построена вокруг классов (classes) и их экземпляров (instances). Это позволяет реализовать следующие функции:

- сокрытие внутренней структуры данных

- абстракция

- наследование

- полиморфизм (способность выбирать правильный метод в зависимости от типа данных)

- ограничение доступа к данным

72. Когда в блоке try-except исполняется элемент else?

В блоке if-else элемент else исполняется в случае, если условие в операторе if (if statement) является неверным (False). А вот в блоке try-except элемент else исполняется только в случае, если элемент try не выдает исключение.

73. Что такое переменная PYTHONPATH?

PYTHONPATH — эта переменная сообщает интерпретатору путь до файлов модуля, импортированных в программу. Поэтому она должна включать в себя директорию с библиотекой-источником питона и директории с исходным кодом питона. Переменную PYTHONPATH можно назначить самостоятельно, однако обычно ее предустанавливает установщик питона.

74. Расскажите про функции join() и split() в Python.

Функция join() позволяет соединять символы строки (string), чередуя с указанным символом. Функция split() позволяет разделить строку, чередуя символы с указанным символом.

75. Приведите несколько методов, с помощью которых можно реализовать в питоне функционально ориентированное программирование.

Несколько методов могут помочь с итерацией по списку (list).

filter() может отфильтровать несколько значений на основе условия.

map() применяет функцию к каждому элементу итерируемого объекта.

reduce() продолжает уменьшать последовательность (sequence) парами, пока не будет достигнуто единичное значение.

75. Можно ли сказать, что del и remove() — это одно и то же? Что это такое, в целом?

del и remove() — это методы для списков, они нужны для удаления элементов

del позволяет удалять элементы под конкретным индексом, а remove() позволяет удалять элементы на основе их значения.

76. Какие различия есть между методами для списков append() и extend()?

Метод append() добавляет элемент к концу списка, а метод extend() добавляет к концу списка переданный ему итерируемый объект (iterable).

77. Какие есть режимы обработки файлов в Python?

Предусмотрены следующие режимы:

- только чтение – ‘r’
- только запись – ‘w’
- чтение-запись – ‘rw’
- добавление в конце – ‘a’

Можно открыть текстовый файл с опцией ‘t’. Поэтому, чтобы открыть текстовый файл для чтения, можно использовать режим ‘rt’. Точно так же для бинарных файлов используется ‘b’.

78. Что делает функция map()?

Функция map() возвращает итератор, который применяет функцию, переданную ей в первом аргументе, ко всем элементам итерируемого объекта (iterable), переданного ей во втором аргументе. Можно показать пример?

79. Расскажите про try, raise и finally.

Это ключевые слова (keywords) для обработки исключений (exception handling). Потенциально рискованный код помещается в блок try, оператор raise (raise statement) используется для прямого вызова ошибки, а в блоке finally находится код, который исполняется в любом случае.

80. Что случится, если не обработать ошибку в блоке except?

Если этого не сделать, программа завершится. Затем она отправит трассу исполнения на sys.stderr.

81. Как можно преобразовать целое число (integer) в символ Unicode?

Для этого просто нужна встроенная функция chr(x).

82. Если строка (string) начинается с пробела, как его убрать?

Такой пробел можно убрать с помощью метода lstrip().

Такой пробел можно убрать с помощью метода lstrip().

В этой строке пробелы стояли как в начале, так и в конце. Функция lstrip() убрала крайний слева пробел из строки. Если мы захотим убрать пробел из хвоста, то воспользуемся функцией rstrip().\

83. Что за функция enumerate() в Python?

Функция enumerate() осуществляет итерацию вдоль последовательности (sequence), извлекает индекс и его значение.

84. В каком случае while уместнее, чем for?

В целом, for подойдет во всех случаях, когда применим while, однако есть несколько ситуаций, когда с циклом while проще:

- Простые повторяющиеся циклы
- Когда не нужно осуществлять итерацию вдоль списка элементов (например, записи в базе данных и символы строки.

85. Объясните разницу между полной копией (deep copy) и поверхностной копией (shallow copy).

Полное копирование создает новый объект-копию. То есть, если внести изменение в копию объекта, то с первоначальным объектом ничего не случится. В Python для этого используется функция `deepcopy()` с помощью импорта из модуля `copy`.

Поверхностная копия копирует на новый объект ту ссылку, которая закреплена на первоначальном объекте. Поэтому если внести изменение в копию, то оно распространится на первоначальный объект. Данный функционал реализуется с помощью функции `copy()`.

86. Можно ли сказать, что массив (array) NumPy лучше списка (list)?

Массивы NumPy имеют три преимущества перед списками:

- Они быстрее
- Они потребляют меньше памяти
- С ними удобнее работать

87. Как можно отслеживать разные версии кода?

Для этого используется контроль версий (version control). Одним из возможных инструментов контроля является Git.

88. Можно ли осуществить динамическую загрузку модуля в Python?

При динамической загрузке модули загружаются только когда они становятся нужны. Такой подход — медленный, но он помогает эффективнее использовать память. В Python для этого можно использовать модуль `importlib`:

89. Какие методы/функции мы используем для определения типа экземпляра (type of instance) и наследования (inheritance)?

Для этого используются `type()`, `isinstance()` и `issubclass()`.

1. type() используется для определени типа объекта.

2. isinstance() принимает два аргумента: значение (value) и тип (type). Если значение относится к соответствующему типу, то возвращается True. Если нет, то возвращается False.

3. issubclass() принимает два класса (classes) в качестве аргументов (arguments). Если второй наследует из первого, то возвращается True. Если нет, то возвращается False.

90. Методы (methods) и конструкторы (constructors) — это одно и то же или нет?

Разница между ними очень тонкая, но важная:

- Название конструктора должно соответствовать названию класса, а метод можно называть как угодно.
- Конструктор исполняется при создании объекта, а метод исполняется при его вызове.
- Конструктор исполняется один раз для каждого объекта, а метод можно вызывать по одному объекту неограниченно.
- Конструкторы используются для определения (define) и инициализации не статических переменных. Методы используются для осуществления операций в рамках бизнес-логики.

91. Что понимается под модулем в питоне?

Модуль — это скрипт, в котором определяются операторы импорта (import statements), функции (functions), классы (classes) и переменные (variables). Файлы ZIP и DLL тоже могут быть модулями. Название модуля хранится в глобальной переменной (global variable) в виде строки (string).

92. Какие в питоне есть модули для работы с файлами?

Питон предлагает следующие библиотеки и модули для обработки текстов и двоичных файлов:

os

os.path

shutil

93. Можете коротко объяснить, как используются модули sqlite3, ctypes, pickle, traceback и itertools.

- sqlite3 помогает обрабатывать базы данных, например SQLite
- ctypes позволяет создавать в питоне типы данных из Си и обрабатывать их
- pickle позволяет переносить любые структуры данных во внешние файлы
- traceback позволяет извлекать, форматировать и выводить на печать трассы вызовов (stack traces)
- itertools помогает работать с перестановками (permutations), комбинациями (combinations) и другими итерируемыми объектами (iterables).

94. Расскажите про наследование (inheritance) в Python.

Когда один класс наследует из другого, его называют дочерним/производным/подклассом (child/derived/sub class), который наследует из родительского/базового/супер класса (parent/base/super class). Он наследует/получает все атрибуты и методы.

Наследование позволяет повторно использовать код и облегчает создание и дальнейшую работу приложений (applications). В Python поддерживаются следующие виды наследования:

- Единичное наследование (Single Inheritance) — класс наследует из одного базового класса.
- Множественное наследование (Multiple Inheritance) — класс наследует из двух или нескольких базовых классов.
- Многоуровневое наследование (Multilevel Inheritance) — класс наследует из базового класса, который, в свою очередь, наследует из другого базового класса.
- Иерархическое наследование (Hierarchical Inheritance) — два класса или несколько классов наследуют из одного базового класса (single base class).

- Гибридное наследование (Hybrid Inheritance) — сочетание двух или нескольких видов наследования.

95. Объясните, как в Python осуществляется управление памятью.

В Python объекты и структуры данных (data structures) находятся в закрытой динамически выделяемой области (private heap), которая управляется менеджером памяти Python. Он делегирует часть работы программам распределения ресурсов (allocators), закрепленным за конкретными объектами, и одновременно с этим следит, чтобы они не выходили за пределы динамически выделяемой области. По факту данной областью управляет интерпретатор (interpreter). Пользователь никак не контролирует данный процесс, даже когда манипулирует ссылками объектов на блоки памяти внутри динамической области. Менеджер памяти Python распределяет пространство динамической области среди объектов и другие внутренние буферы по требованию.

96. Как можно сделать скрипт Python, исполняемый в Unix?

Для этого должны выполняться два условия:

- Файл скрипта должен быть в исполняемом режиме.
- Первая строка должен начинаться с решетки (хэша, hash(#)), например: #!/usr/local/bin/python

97. Что такое временная подмена (Monkey Patching)?

Она модифицирует класс или модуль во время выполнения (at runtime), то есть представляет собой динамическую модификацию (dynamic modification).