Программирование в лингвистике

Повторение

Поговорим о Python

1. На каком уровне ЯП находится Python?

Низкий | Средний | **Высокий**

2. К какому типу ЯП относится Python?

Компилируемые | **Интерпретируемые**

3. В какой парадигме по умолчанию работает Python?

Объектно-ориентированная | Функциональная | Структурная

Что такое среда разработки (IDE)? Какие IDE Вы знаете?

IDE - программный комплекс, предназначенный для написания и тестирования программ на ЯП.

К IDE для Python относятся PyCharm, Visual Studio Code, IDLE, Google Colaboratory, Jupyter Notebook, Wing и другие.

Чем отличается интерактивный режим работы с Python (консоль) от скриптового?

В консоли команды вводятся по одной, получившийся код никуда не сохраняется.

В скриптовом режиме код пишется полностью и выполняется целиком после сохранения.

Что такое .py и .ipynb? В чем их отличия?

Py-файлы предполагают выполнение кода целиком, а тетрадки (ipynb) позволяют запускать ячейки кода произвольно.
Также в тетрадках есть markdown-ячейки, позволяющие показывать оформленные комментарии.

x = 1

Что здесь что в этом коде?

x – название создаваемой переменной.

= – оператор присваивания.

1 – объект типа int, который попадает в переменную x.

Какие встроенные типы данных Вы знаете?

Неизменяемые

int, float, complex (числовые), bool (булево значение), str (строки), bytes (байты), tuple (кортежи), None,

list (списки), dict (словари), set (множества).

Изменяемые

В чем разница между изменяемыми и неизменяемыми типами данных?

Объекты изменяемого типа данных можно изменить по частям.

Объекты неизменяемого типа данных можно только перезаписать.

Какие операции можно выполнять с числами?

Арифметические: сложение (+), вычитание (-), умножение (*), деление (/), целочисленное деление (//), остаток от деления (%), возведение в степень (**).

Операции сравнения: больше (>), меньше (<), больше или равно (>=), меньше или равно (<=), равно (==), не равно (!=).

Какой тип данных будут возвращать операции **/**, **//** и **%**?

Деление (/) всегда возвращает float: 4 / 2 = 2.0

Целочисленное деление (//) и остаток от деления (%) возвращают int, если оба операнда – int, в ином случае float: 4 // 2 = 2, **НО** 4.0 // 2 = 2.0

Какие итерируемые типы данных Вы знаете?
Чем они отличны от неитерируемых?

Объекты итерируемого типа данных можно перебрать по частям (к примеру, в цикле for).

К итерируемым типам данных относятся строки, списки, множества, словари, кортежи.

Какие методы строк Вы можете назвать?
Какой тип данных они возвращают?

Возвращают str: replace, strip (rstrip, lstrip), upper, lower, capitalize, title

Возвращают int: find (rfind), index, count

Возвращают bool: startswith, endswith, isupper, islower, istitle, isalpha, isdigit, isspace, isalnum

Чем отличны f-строки от обычных строк?

Префикс f позволяет форматировать строку, вставляя значение переменной или результат операции непосредственно в нее в требуемом виде.

f’5 / 2 = {5 / 2}’ аналогично ‘5 / 2 =’, 5 / 2

Что такое индексы у строки?
Зачем нужны срезы? Из чего они состоят?

Индексы – порядковые номера символов в строке.

Срезы позволяют получить часть строки, используя индексы символов. Три элемента среза – начало, конец и шаг.

‘qwerty’.
Как взять из этой строки ‘wer’?

С помощью срезов:
‘qwerty’[1:4]

Как проитерироваться по строке и вывести каждый символ по отдельности?

В циклах for и while по индексам, либо в цикле for поэлементно.

Хитрый способ с распаковкой:
print(*string, sep=\n)

В чем разница между циклами for и while?

Цикл for принимает итерируемый объект и перебирает его. Почти всегда имеет окончание.

Цикл while принимает условие и повторяет команды до тех пор, пока условие не будет являться False.

for i in range(len(**lst**)) и for i in **lst**
В чем разница?

В первом варианте мы итерируемся по индексам списка.

Во втором варианте мы итерируемся по элементам списка.

Какие методы списков Вы можете назвать?

Append (добавление), clear (очищение), copy (копирование), count (подсчет), extend (расширение), index (поиск), insert (вставка), pop и remove (удаление), reverse (обращение), sort (сортировка).

В каком/каких случаях изменится список «a»?

a = [1, 2, 3], **b = a.copy()**, b[0] = 5

a = [1, 2, 3,], **b = a**, b[0] = 5

a = [1, 2, 3,], **b = a[:]**, b[0] = 5

Чем множество отличается от списка?

Элементы множества не упорядочены и уникальны. Во множестве могут находиться только неизменяемые объекты.

Обращение к элементу множества происходит быстрее, чем к элементу списка.

Для чего обычно используются множества?

Множества используются для хранения уникальных объектов.

Так, они часто используются для хранения тегов.

Какие методы множеств Вы можете назвать?

union (объединение |), intersection (пересечение &), difference (разность -), symmetric difference (симметрическая разность ^).

Проверки: равенство (==), isdisjoint (неравенство), issubset и issuperset (подмножества - <= и >=)

А также remove, pop и clear.

Опишите словари в Python. Как их можно задавать?

Словари - неупорядоченные коллекции пар объектов с доступом по ключу. Словари итерируемые и изменяемые.

Их можно задавать явно в коде {ключ: значение}, преобразованием списка кортежей (также используя zip и enumerate), с помощью fromkeys или генератором словарей.

Как можно итерироваться по словарю?

По ключам (keys), по значениям (values), по парам (ключ, значение) (items)

for k, v in dct.items() и for k in dct
В чем разница?

В первом случае мы итерируемся сразу по распакованным ключам и значениям словаря, а во втором – только по ключам.

Зачем нужны функции?
Как их определять?

Функции нужны для оптимизации внешнего вида кода.
Их часто используют тогда, когда нужно повторить
одни и те же действия несколько раз.

Функции объявляются с помощью def:
def func(*args, **kwargs):

Чем функция отличается от метода?

Функции применимы к объектам разных типов, в то время как методы принадлежат только определенному встроенному типу данных или пользовательскому классу.

Что такое параметры и аргументы функции?

Параметры функции – это переменные, которые передаются в функцию при ее вызове.

Аргументы функции – это объекты, передаваемые в параметры функции при ее вызове.

Что значит передавать аргументы в функцию позиционально? А по ключу?

Позициональная передача аргументов предполагает, что объекты, передаваемые в функцию, должны находиться в строгой последовательности.

Передача по ключу разрешает передавать аргументы в произвольном порядке, но с указанием keyword.

Как задать необязательный параметр функции, как sep у print?

С помощью ключа и дефолтного значения:
def func(arg=1)

Что возвращает функция, если у нее нет return?

Объект NoneType.

Среди таких функций - print, а также все in-place методы.

Как Python работает с файлами?

Python работает с файлами через систему.
Для работы с файлами у Python есть дескриптор файла – специальный класс, который позволяет читать данные из файла – IOWrapper.

Какие параметры есть у функции open? Зачем они нужны?

У open() есть три параметра: path, mode и encoding. Путь подается в виде строки и является обязательным параметром, режим настраивает Python для определенной работы с файлом, кодировка позволяет избежать проблем с чтением и записью символов.

Как прочитать файл? Назовите 3 способа.

Можно прочитать файл целиком (read), получить список строк (readlines), а также в цикле итерироваться по строкам (for line in file).

Как записать файл?

Можно записать в файл строку (write), список строк (writelines), а также использовать print() для записи:
print(**any**, file=file)

Что будет, если не закрыть файл после окончания работы с ним?

Данные не запишутся, а файл останется в оперативной памяти.