

Вопросы к экзамену (Python язык программирования) Junior developer.

Экзамен состоит из билетов, билет содержит в себе 3 основных теоретических вопроса и 1 задачу, а также дополнительные вопросы.

За экзамен можно получить 60 баллов если:

- ответить на основную теорию (+30 баллов)
- решить задачу (+20 баллов)
- ответить на доп вопросы (+10 баллов)

БИЛЕТ ФОРМИРУЕТСЯ СЛУЧАЙНЫМ ОБРАЗОМ.

НА ПОДГОТОВКУ К ОТВЕТУ ВЫДЕЛЯЕТСЯ РОВНО 1 ЧАС ВРЕМЕНИ.

РАЗРЕШАЕТСЯ 1 РАЗ ПЕРЕБРАТЬ БИЛЕТ (БЕЗ ПОТЕРИ БАЛЛОВ) ЗА КАЖДЫЙ ПОСЛЕД-Й ПЕРЕБОР **-20 БАЛЛОВ**.

ПРИ ПОДГОТОВКЕ МОЖНО ИСПОЛЬЗОВАТЬ КОНСПЕКТЫ ЛЕКЦИЙ!!! (НО ЭТО ВСЕ РАВНО НЕ ПОМОЖЕТ)

*ЗАПРЕЩЕНО ИСП-Е ИНТЕРНЕТА И ДР СРЕДСТВ. НО РАЗРЕШЕНО ПРИ РЕШЕНИИ ЗАДАЧИ ИСП-ТЬ **VSCODE**.*

1 БЛОК (основы языка)

1. Компиляция программы. Ввод/Вывод данных.
2. Переменные. Типы данных. Операции над переменными.
3. Условная конструкция IF-ELSE. Зачем нужен ELIF?
4. Виды циклов в Python. While. For.
5. Работа с файлами и строками. F-строки.
6. PER. Виды PER.
7. Коллекции в Python. Списки. Срезы.
8. Коллекции в Python. Кортежи. Множества. Словари.
9. Оценка сложности программы.
10. Алгоритмы сортировок. Сортировка пузырьком, и выбором.
11. Функции. Вызов функций, способы задания параметров (по-умолчанию, *argv, **kwargs... и др)
12. Функции. Декораторы. Понятие перегрузки функции.
13. Полезные функции min/max/...
14. Комментарии в Python, Docstring, + что такое и зачем используется Typehints?
15. Глобальные переменные.
16. Исключения и ошибки. Обработка исключений. RAISE.

2 БЛОК (ООП, модули)

1. ООП. Преимущества / недостатки ООП. SOLID принципы. Также основные понятия класса/объекта/атрибута/методы.
2. ООП. Статические и динамические атрибуты (поля). Понятие конструктора/деструктора класса, и что такое SELF?
3. ООП. Основные концепции ООП.
4. ООП. Public/Private атрибуты (поля). Public/Private методы. Что такое single underscore?
5. ООП. Соглашение NM. Основные типы методов класса. Магические методы класса.
6. ООП. Понятие стека/очереди/дека.
7. Модули и библиотеки. В чем их отличия?
8. Стандартные модули и библиотеки (OS/MATH..)
9. Внешние модули и библиотеки. PIP.
10. Парсинг данных.

3 БЛОК (АИСД и сторонние модули)

1. АИСД. Связный список. Виды связного списка.
2. АИСД. ХЕШ-таблица. Коллизии.
3. АИСД. Динамический массив данных.
4. АИСД. Деревья. Основные виды деревьев.
5. АИСД. Графы, алгоритмы.
6. Telebot. Как создать простого бота для вывода фразы Hello world? Какой алгоритм создания ботов?
7. Работа с Excel таблицами.
8. ИИ как устроен? как создается ИИ?
9. PyQT.
10. FLASK/Django.
11. Docker.
12. Интернет. HTTP. DNS. TCP/IP..
13. ОС. Работа с терминалом. Основные команды терминала.
- 14*. <самый крутой вопрос> Зачем нужен Python? Где применяется Python?

Примеры задач для самоподготовки:

1. Дан список (который задает пользователь - вводит с использованием спец. функции заполнения), найти минимальный положительный элемент в этом списке.

2. Дан словарь (который задает пользователь), создайте функцию `get_index()` и реализуйте поиск индекса первого вхождения положительного четного ключа, а после вывести значение по этому ключу (`key`).
3. Даны три кортежа, соедините три кортежа воедино, написав соответствующую функцию объединения. А также выведите длину получившегося кортежа.
4. <задача на стек данных> Напишите функцию, которая принимает на вход строку, содержащую только символы '(', ')', '{', '}', '[' и ']', и определяет, является ли расстановка скобок в этой строке корректной.
5. <задача на стек данных> Напишите функцию, которая проверяет балансировку символов в строке. В строке могут присутствовать только символы '(' и ')'. Функция должна вернуть `True`, если каждой открывающей скобке соответствует закрывающая, и `False` в противном случае.
6. Напишите функцию которая принимает список вещественных чисел и возвращает два самых наибольших отрицательных числа в этом списке данных.
7. <задача с leetcode> Учитывая список целых чисел **nums** и целое число **target**, верните индексы двух чисел так, чтобы их сумма составляла **target**. Вы можете предположить, что каждый вход будет иметь ровно одно решение, и вы не можете использовать один и тот же элемент дважды. Вы можете вернуть ответ в любом порядке.

Пример 1:

```
Ввод: nums = [2,7,11,15], target = 9
Выход: [0,1]
Объяснение: поскольку nums[0] + nums[1] == 9, мы возвращаем [0, 1].
```

Пример 2:

```
Ввод: nums = [3,2,4], цель = 6
Выход: [1,2]
```

Пример 3:

```
Ввод: nums = [3,3], цель = 6
Выход: [0,1]
```

** стоит отметить что задача может быть на любую из пройденных тем, советую повторить ООП, АИСД (деревья, хеш-таблицы).*