

---

QA  
מע

אלכס גורבצ'וב  
АЛЕКСЕЙ ГОРБАЧЁВ



---

PYTHON  
PYTHON



# Python



## Методы в Python. **\*\*kwargs**.

В Python, **\*\*kwargs** (или распаковка словаря) - это особая синтаксическая конструкция, которая позволяет передавать произвольное количество именованных аргументов в функцию. Имя **kwargs** является соглашением, хотя можно использовать и другое имя, но **\*\*kwargs** - распространенное именование, которое стало устоявшимся.

Когда определена функция с **\*\*kwargs** в списке аргументов, это означает, что функция принимает дополнительные именованные аргументы, которые не были определены заранее в списке аргументов функции. Аргументы передаются в виде словаря, где ключи - это имена аргументов, а значения - значения аргументов.

# Python



Методы в Python. **\*\*kwargs**.

Пример использования **\*\*kwargs**:

```
def print_kwargs(**kwargs):  
    for key, value in kwargs.items():  
        print(f"{key} : {value}")
```

```
# Вызов функции с различными именованными аргументами  
print_kwargs(name="Alice", age=30, city="New York")
```

Словарь будет следующим:

```
kwargs = {"name": "Alice", "age": 30, "city": "New York"}
```

# Python



## Задание.

- Написать функцию `square`, принимающую 1 аргумент - сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.
- Написать функцию `season`, принимающую 1 аргумент - номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень).
- Пользователь делает вклад в размере `X` рублей сроком на `years` лет под 10% годовых (каждый год размер его вклада увеличивается на 10%. Эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты). Написать функцию `bank`, принимающая аргументы `X` и `years`, и возвращающую сумму, которая будет на счету пользователя.

# Python



## Функции для работы со списками.

- `append()`: Добавляет элемент в конец списка.
- `extend()`: Расширяет список, добавляя элементы другого списка.
- `insert()`: Вставляет элемент на указанную позицию в списке.
- `remove()`: Удаляет первый элемент с указанным значением из списка.
- `pop()`: Удаляет и возвращает элемент с указанным индексом (по умолчанию - последний элемент).
- `clear()`: Очищает список, удаляя все его элементы.
- `index()`: Возвращает индекс первого вхождения элемента с указанным значением.
- `count()`: Возвращает количество вхождений элемента с указанным значением.
- `sort()`: Сортирует список по возрастанию (по умолчанию) или по кастомному ключу.
- `reverse()`: Изменяет порядок элементов списка на обратный.
- `copy()`: Создает копию списка.
- `len()`: Возвращает количество элементов в списке.
- `in` и `not in`: Используются для проверки наличия элемента в списке.

# Python



## Задание.

- Напишите функцию `change(lst)`, которая принимает список и меняет местами его первый и последний элемент. В исходном списке минимум 2 элемента.
- Иван решил создать самый большой словарь в мире. Для этого он придумал функцию `biggest_dict(**kwargs)`, которая принимает неограниченное количество параметров «ключ: значение» и обновляет созданный им словарь `my_dict`, состоящий всего из одного элемента «first\_one» со значением «we can do it». Воссоздайте эту функцию.
- Создайте функцию `three_args()`, которая принимает 1, 2 или 3 строго ключевых параметра. В результате ее работы на печать в консоль выводятся значения переданных переменных, но только если они не равны `None`. Получим, например, следующее сообщение: Переданы аргументы: `var1 = 2, var3 = 10`.



Thanks for your time 😊