

Тема занятия

# Списки и кортежи



# План

# занятия



Введение

Теория

Списки

Теория

Кортежи

Практика

Список покупок

Практика

Злодей

Домашнее задание

Итоги

# Проверяем домашнее задание

## 1. Трудности

Какие трудности возникли при написании кода?

## 2. Обратная связь

Понравилось использовать срезы строк? Трудно ли было считать индексы?

## 3. Расскажи

Какой информации не хватило для выполнения задания?

## 4. Соб@чка

Какой индекс был у собачки?

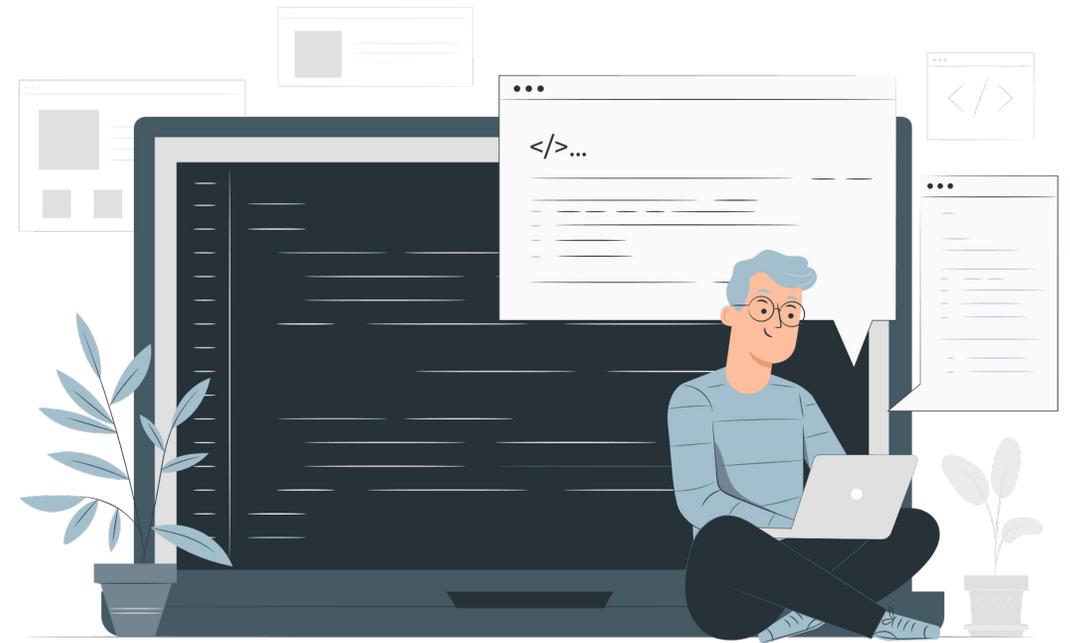


# Введение

Списки и кортежи это очень популярные и важные типы данных в Python.

Без них не обходится ни одна серьезная программа.

Сегодня мы узнаем, что они такое и чем похожи на строки из предыдущего урока.



Теория

# СПИСКИ

## Теория

Списки используются для хранения и использования множества разных данных в одной переменной.

Они очень похожи на список покупок, потому что работают по одному принципу.

1. Йогурт
2. Туалетная бумага
3. Яйца
4. Соль
5. Чай
6. Салфетки

1. Мука
2. Соль
3. Сода пищевая
4. Сахар
5. Яйца
6. Молоко

## Теория

В Python списки обозначаются **квадратными скобками**, а элементы внутри записываются через запятую

```
to_buy = ["Мука", "Сахар", "Яблоки"]
```

В списки можно записать любые объекты, а не только строки. И даже выражения – они посчитаются и запишутся в список.

```
to_buy = [435, "Сахар", True, 245.516]  
cell = [349*45, 34+11]
```



## Теория

Списки обладают свойством индексации и срезов, прямо как строки, только вместо букв тут выступают элементы.

```
print(to_buy [0])  
print(to_buy[1:3])
```

```
435  
['Сахар', True]
```



## Теория

Главное отличие списков — возможность менять значения элементов.

Это делает их незаменимыми помощниками в написании сложных программ.

Для изменения значения нужно обратиться по индексу к нужному элементу.

```
to_buy = [435, "Сахар", True, 245.516]
print(to_buy)
to_buy[1] = "Соль"
print(to_buy)
```

```
[435, 'Сахар', True, 245.516]
[435, 'Соль', True, 245.516]
```

В коде видно,  
что второй элемент изменился

## Теория

Также мы можем  
добавлять значения в список.

Делается это с помощью метода **append()**.

```
shop = ["Аты-баты", "Шли солдаты", "Аты-Баты"]
shop.append("На базар")
print(shop)

['Аты-баты', 'Шли солдаты', 'Аты-Баты', 'На
базар']
```

Удаление из списка можно выполнить  
с помощью оператора **del**:

```
shop = ["Аты-Баты", "Шли солдаты"]
del shop[0]
print(shop)

['Шли солдаты']
```

После **del** нужно указать название списка  
и индекс того элемента в квадратных скобках,  
который нужно удалить.

Теория

# Кортежи

## Теория

Кортежем называют список, который нельзя редактировать (неизменяемый).

Кортеж создается точно так же, как и список, но вместо квадратных скобок, кортеж использует круглые.

Если мы попробуем изменить его, то получим ошибку, которая сообщит нам, что кортежи нельзя редактировать.

```
to_buy = (435, "Сахар", True, 245.516)
print(to_buy)
to_buy[1] = "Соль"
print(to_buy)
```

```
(435, 'Сахар', True, 245.516)
```

**TypeError**

Traceback (most recent call last)

Cell In[5], line 3

```
1 to_buy = (435, "Сахар", True, 245.516)
2 print(to_buy)
-----> 3 to_buy[1] = "Соль"
4 print(to_buy)
```

**TypeError:** 'tuple' object does not support item assignment

# Теория

Возникает вопрос:  
А зачем тогда нужны кортежи?



## Теория

Не во всех случаях есть необходимость менять элементы, например, список имен врагов в игре можно задать в кортеже.

Это сильно влияет на скорость и производительность кода, потому что кортеж быстрее работает и занимает меньше места, чем список.



## Теория

Не во всех случаях есть необходимость менять элементы, например, список имен врагов в игре можно задать в кортеже.

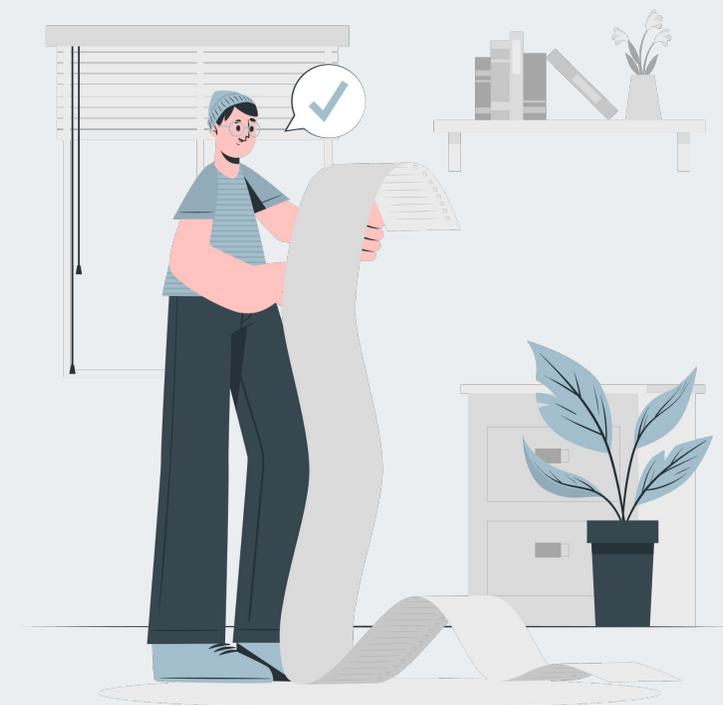
Это сильно влияет на скорость и производительность кода, потому что кортеж быстрее работает и занимает меньше места, чем список.

Совет: используй кортежи, когда знаешь, что элементы не будут меняться во время работы кода.



Практика

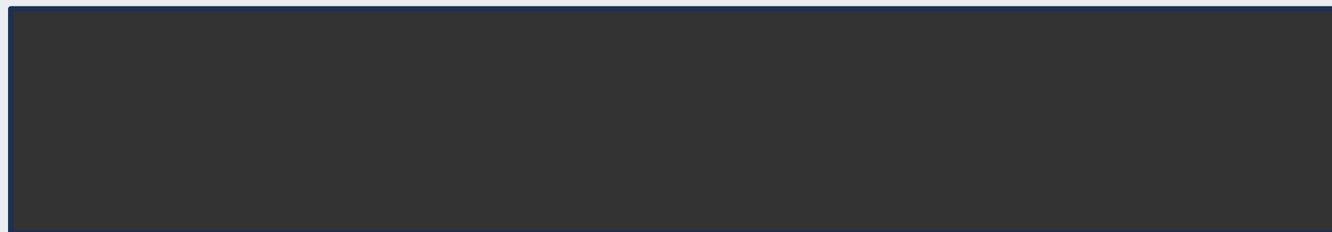
# СПИСОК ПОКУПОК



## Практика



Составь список покупок не менее чем из 5 товаров. Выведи на экран только последние 3 покупки



## Практика



Составь список покупок не менее чем из 5 товаров. Выведи на экран только последние 3 покупки

```
to_buy = ["Мука", "Сахар", "Яблоки", "Какао", "Апельсины"]  
print(to_buy[2:5])
```

```
['Яблоки', 'Какао', 'Апельсины']
```

Практика

# Злодей



## Практика

Попробуй составить, прочитать и отредактировать кортеж.

Что из этого получится, а что нет?

```
to_buy = ("Мука", "Сахар", "Яблоки")
print(to_buy[0])
to_buy[0] = "Мандарины"
print(to_buy[0])
```

Мука

**TypeError**

Traceback (most recent call last)

Cell In[9], line 3

```
1 to_buy = ("Мука", "Сахар", "Яблоки")
2 print (to_buy[0])
-----> 3 to_buy[0] = "Мандарины"
4 print (to_buy [0])
```

**TypeError:** 'tuple' object does not support item assignment

# Домашнее задание

Необходимо составить список из 5 текстовых значений (тип данных string)

Выведи в консоль каждую первую букву каждого элемента списка.

## Подсказка:

нужно будет ставить две пары квадратных скобок (`data[[]]`)



# Итоги

---

**1.** Что нового сегодня было на занятии?

---

**2.** В чём схожесть списков со строками? В чём различия?

---

**3.** Как ты думаешь, где еще ты бы мог применить эти знания?

---

**4.** Для чего ты можешь использовать кортежи?

