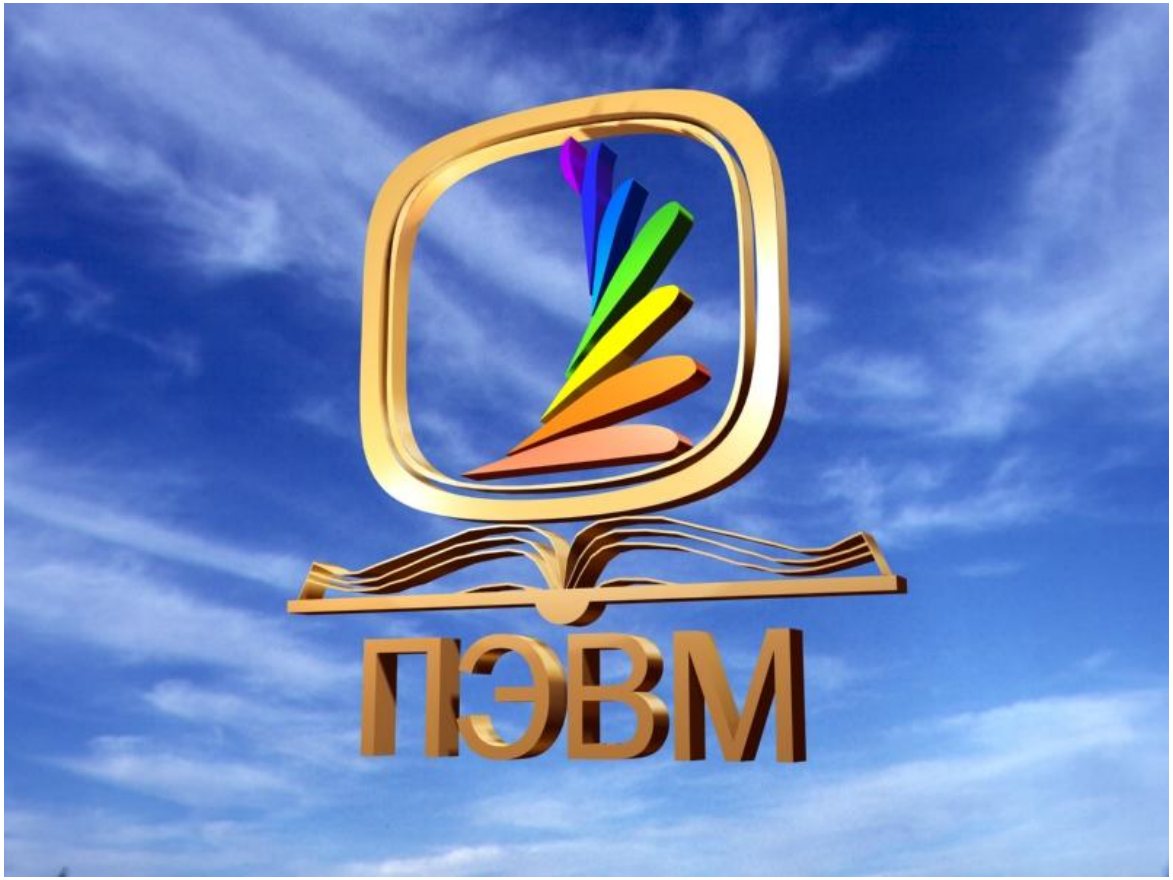
# Разрежённое лучше плотного



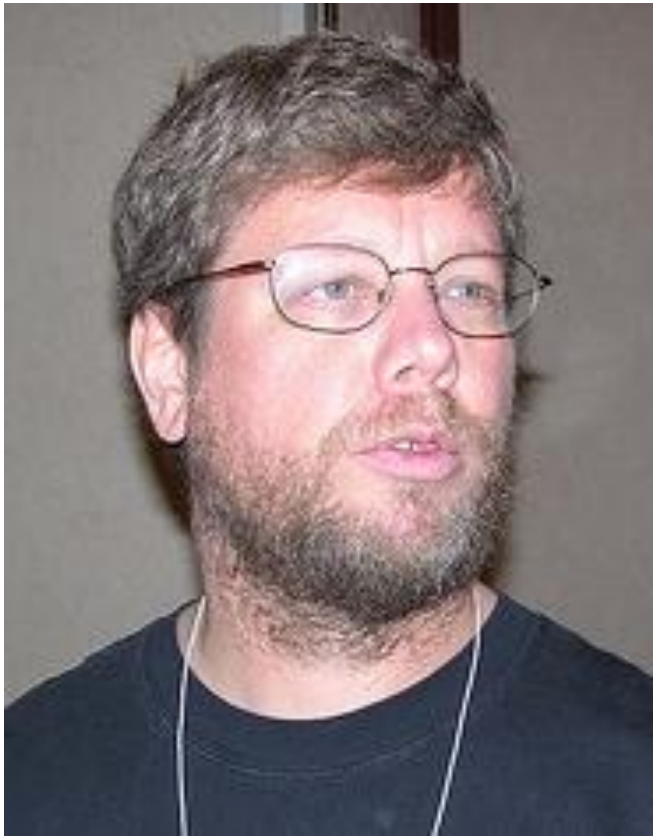

# Явное лучше неявного

Гвидо ван Россум

# Язык программирования Python
(Лекция 12. Кортежи)

**Валеева Н.Х. – кан. пед. наук,
преподаватель ц/к ПЭВМ**

# Язык программирования Python был создан примерно в 1991 году голландцем Гвидо ван Россумом.



Официальный сайт http://python.org

# Кортеж (tuple)

**Кортежи в Python - упорядоченные неизменяемые совокупности объектов произвольных типов, заключенные в круглые скобки**

**Например:**

(23, 656, -20, 67, -45) *# кортеж целых чисел*

(4.15, 5.93, 6.45, 9.3, 10.0, 11.6) *# кортеж из дробных чисел*

("Katy", "Sergei", "Oleg", "Dasha") *# кортеж из строк*

("Москва", "Титова", 12, 148.4) *# смешанный кортеж*

([0, 0, 0], [0, 0, 1], (0, 1, 0), 100) *# кортеж, состоящий из списков, кортежей и числа*

**Кортеж, по сути - *неизменяемый* список.**

# Особенности кортежа


- Кортеж защищен от изменений, как намеренных (что плохо), так и случайных (что хорошо). То есть «защита от дурака».
- Имеет меньший размер:

```
>>> a = (1, 2, 3, 4, 5, 6)
>>> b = [1, 2, 3, 4, 5, 6]
>>> a.__sizeof__()      #  36
>>> b.__sizeof__()      #  44
```

- Кортежи работают быстрее, чем списки.
- Кортежи можно использовать в качестве ключей словаря

# Кортежи и списки


Основное отличие между кортежами и списками состоит в том, что кортежи не могут быть изменены. На практике это означает, что у них нет методов, которые бы позволили их изменить. У списков есть такие методы, как append(), extend(), insert(), remove(), и pop().

У кортежей ни одного из этих методов нет!

# Способы создания кортежей


1. **Пустой кортеж:**

```
>>> a = ()
```

или

```
>>> b = tuple()
```

2. **Одноэлементный кортеж:**

```
>>> a = (5, )
>>> print(a)
(5, )
```

# Способы создания кортежей

**3. Произвольный кортеж можно создать простым перечислением элементов:**

```
>>> a = (1, 2, 3, 4, 5)
>>> print(a)
(1, 2, 3, 4, 5)
```

***Или***

```
>>> a = tuple((1, 2, 3, 4))  # скобки!
>>> print(a)
(1, 2, 3, 4)
```

***Или***

```
>>> a = ('hello, world!')
>>> a
('h', 'e', 'l', 'l', 'o', ',', ' ', 'w', 'o', 'r', 'l', 'd', '!')
```

# Доступ к элементам кортежа


Осуществляется через индекс:

```
>>> a = (1, 2, 3, 4, 5)
>>> print(a[0])
1
>>> print(a[1:3])
(2, 3)
>>> a[1] = 3
TypeError: 'tuple' object does not support item assignment
```

# Удаление кортежей


```
>>> a = (1, 2, 3, 4, 5)
>>> del a[0]
TypeError: 'tuple' object doesn't support item
   deletion

>>> del a
>>> print(a)
NameError: name 'a' is not defined
```

# Кортежи. Выводы.


1. Вы *не можете* добавить элементы к кортежу. Кортежи не имеют методов append() или extend().
2. Вы *не можете* удалять элементы из кортежа. Кортежи не имеют методов remove() или pop().
3. Вы *можете* искать элементы в кортежи, поскольку это не изменяет кортеж.
4. Вы также *можете* использовать оператор in, чтобы проверить существует ли элемент в кортеже.

# Кортеж □=□ Список


Функция **tuple()** принимает список и возвращает кортеж из всех его элементов:

```
a = [1, 2, 3, 4, 5]     # наш список
b = tuple(a)          # преобразование в кортеж
print(b)            # вывод кортежа (1, 2, 3, 4, 5)
```

Функция **list()** принимает кортеж и возвращает список:

```
a = (1, 2, 3, 4, 5, [1, 2])   # наш кортеж
b = list(a)                 # преобразование в список
print(b)            # вывод списка [1, 2, 3, 4, 5, [1, 2]]
```

***Иначе, tuple() замораживает список,***
***а list() размораживает кортеж.***

# Базовые операторы кортежей

| *Выражение* | *Результат* | *Название* |
|---|---|---|
| len((1, 2, 3)) | 3 | длина |
| (1, 2, 3) + (4, 5, 6) | (1, 2, 3, 4, 5, 6) | конкатенация |
| (‘Hi!’) * 4 | (‘Hi!’, ‘Hi!’, ‘Hi!’, ‘Hi!’) | повторение |
| 3 in (1, 2, 3) | True | вхождение |
| for x in (1, 2, 3):<br>print (x) | 1<br>2<br>3 | итерация |
| a = (‘end’,’End’,’END’)<br>b = a[2]<br>c = a[-2]<br>g = a[1:] | <br>END<br>End<br>(‘End’, ‘END’) | срезы |

# Методы кортежей


1. **cmp(tuple1, tuple2) -** сравнение элементов двух кортежей;
2. **len(tuple) -** количество элементов в кортеже;
3. **max(tuple)** - получить наибольший элемент кортежа;
4. **min(tuple)** - получить наименьший элемент кортежа;
5. **sorted(tuple) –** отсортировать кортеж;
6. **. tuple.index()** – получить индекс указанного элемента кортежа;
7. **tuple.count()** - получить количество вхождений в кортеж указанного элемента

# Задача


Дана последовательность фамилий сотрудников фирмы с годами их рождения.

Выделить из этого списка кортеж фамилий, подсчитать количество, отсортировать и вывести в текстовый файл fio.txt в столбик

А также выделить кортеж годов рождения и определить возраст самого молодого и пожилого работников.

# Скрипт

```
# Кортежи
t = ("Бахарев",1963 ,"Салин",1985,"Ардов",1981,"Козлов",1997,"Шудрик",1975)
f = t[0::2] # кортеж фамилий
g = t[1::2] # кортеж годов рождения

print('Количество сотрудников - ', len(f))
tf = tuple(sorted(f))
print('Список работников по алфавиту')
print(tf)
fail= open('fio.txt', 'w')
for a in tf:
    fail.write(a+'\n')
fail.close()

mn = 2018 - max(g)
mx = 2018 - min(g)
print('Возраст самого молодого сотрудника равен ',mn)
print('Возраст самого пожилого сотрудника равен ',mx)
```


Количество сотрудников - 5

Список работников по алфавиту

['Ардов', 'Бахарев', 'Козлов', 'Салин', 'Шудрик']

Возраст самого молодого сотрудника равен 21

Возраст самого пожилого сотрудника равен 55

# Выполнить задание:

**Задание 1: создайте кортеж, в котором храниться информация о результатах квалификационных выступлений 10 фигуристов (фамилия, балл) . На соревнования допускаются только те, кто получил результаты выше среднего. Составьте программу, которая определяет число спортсменов, прошедших квалификацию и выводит их фамилии.**

**Задание 2: Составьте функцию, которая будет удалять элемент кортежа. Составьте программу, которая удаляет из кортежа максимальный и минимальный элементы и выводит оставшиеся результаты в порядке убывания.**

**Задание 3: Напишите функцию get_score(student, hw, exam), которая принимает на вход имя студента (student), кортеж, содержащий его оценки за домашние задания (hw) (у всех разные оценки и количество), а также его оценку за экзамен (exam), и выдает строку**

**<Student>, ваш рейтинг равен <grade>.**

**Student — имя студента, grade — его итоговый балл-рейтинг за курс, являющаяся целым числом.**

**Известно, что итоговая оценка по курсу считается так: Итог=0.4·ДЗ + 0.6·экзамен, где ДЗ - среднее арифметическое оценок за домашние задания. Итог должен быть округлен до целого числа.**

**Пример:**

**get_score(«Олег", (6, 7, 8, 9, 5, 4), 6)**
**Олег, ваш рейтинг равен 6.**