Коллекции. Функции. Работа с файлами.

Лавприт Сингх-Пальчевская
младший научный сотрудник МГУ им. Ломоносова,
кафедра биоинженерии

Проверка связи

Отправьте «+», если меня видно и слышно

Если у вас нет звука или изображения:

- перезагрузите страницу
- попробуйте зайти заново
- откройте трансляцию в другом браузере

О чем поговорим сегодня

МФТИ

1. Рассмотрим встроенные в Python коллекции
2. Обсудим тему функций приёмы работы с файлами на Python
3. Потренируемся в решении задач

МФТИ

Коллекции в Python

Коллекции в Python

Коллекция — структура данных (“переменная-контейнер”), которая включает в себя некоторое количество объектов *одного* или *разных* типов и позволяет обращаться к ним.

Не путать с модулем collections

Встроенные в Python коллекции:

- список
- кортеж
- множество
- словарь

`collections` - это модуль, который предоставляет специализированные типы данных, на основе словарей, кортежей, множеств, списков, т.е. модуль, содержащий набор “объектов-оберток” для различных типов коллекций.

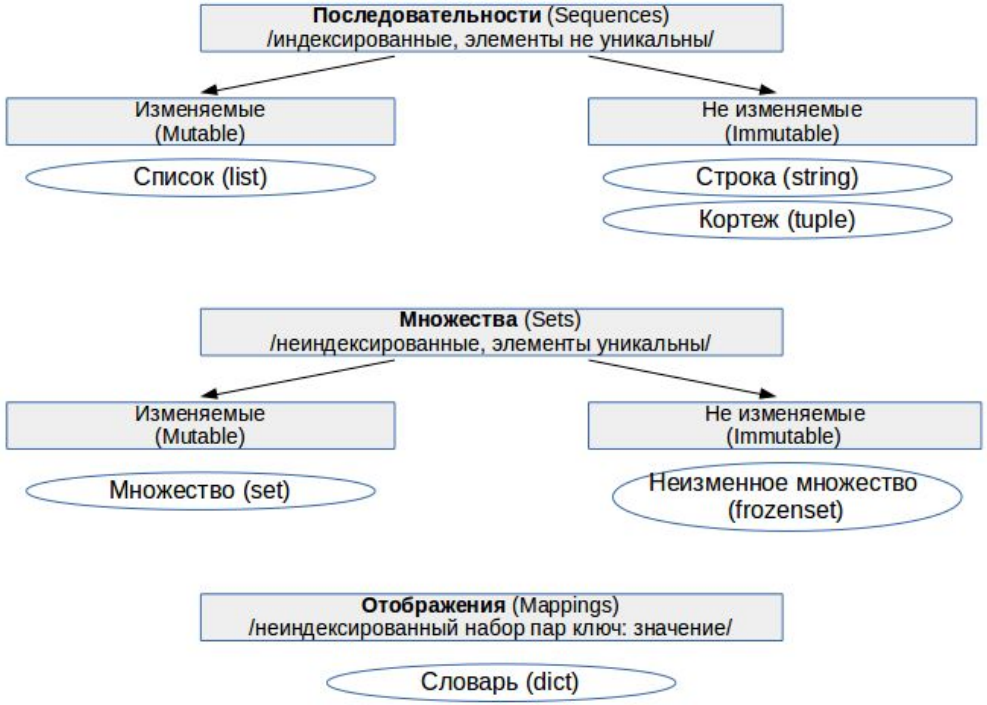
Полный скрипт: https://colab.research.google.com/drive/1MBrGcxRUaJ29bHQnQvStHswCP-_li-DK?usp=sharing

Коллекции в Python

Согласно документации *нет* понятия коллекций.

Однако, есть такие понятия, как:

- последовательности (sequences)
- множества (sets)
- отображения (mappings)



Документация: https://docs.python.org/3/library/stdtypes.html

Встроенные в Python коллекции

МФТИ

	Обозначение	Возможность изменения	Упорядоченность элементов	Уникальность элементов
Список list	[]	да	да	нет
Кортеж tuple	()	нет	да	нет
Множество set	{}	да	нет	да
Словарь dict	{} содержит пары элементов `{key: value}`	`key` — нет `value` — да	нет	`key` — да `value` — нет

Строки (str) *не* являются коллекцией; **строки** - это *упорядоченные не изменяемые* последовательности.

Пример использования коллекций

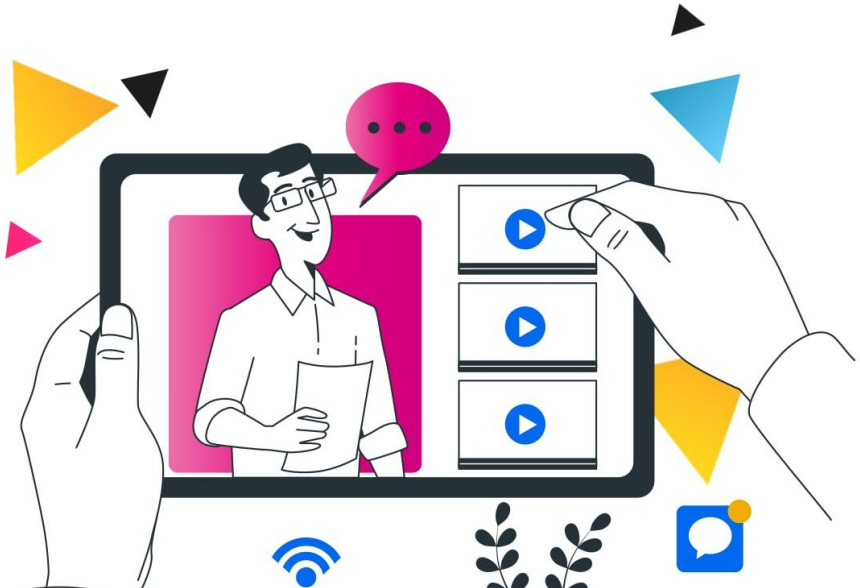
Для хранения информации о товарах в корзине можно использовать список словарей

```
cart = [{"product": "Milk", "quantity": 1, "price": 62.9},
        {"product": "Bread", "quantity": 2, "price": 1.6},
        {"product": "Chocolate", "quantity": 1, "price": 89.5},
        {"product": "Fish", "quantity": 1.2, "price": 189.2},
        {"product": "Tea", "quantity": 4, "price": 69.5}]
s = 0
for product in cart:
 s += product["price"]*product["quantity"]
print(f'Общая стоимость товаров в продуктовой корзине{s} рублей')
```

```
Общая стоимость товаров в продуктовой корзине 278.0 рублей
```

Полный скрипт: https://colab.research.google.com/drive/1MBrGcxRUaJ29bHQnQvStHswCP-_li-DK?usp=sharing

Ваши вопросы?



МФТИ

Функции в Python

Функция

Функции в Python - это ***объект***, который принимает на вход *аргументы* и *возвращает* значение.

Функции делятся на 2 типа:

- **встроенные функции**, которые являются частью Python (`raw_input`(), `type`(), `float`(), `max`(), `min`(), `int`(), `str`(), …)
- **функции, объявленные пользователем**, которые объявляются и определяются пользователем для конкретной цели

Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Функция

Функции в Python - это ***объект***, который принимает на вход *аргументы* и *возвращает* значение.

Функции делятся на 2 типа:

- **встроенные функции**, которые являются частью Python (`raw_input`(), `type`(), `float`(), `max`(), `min`(), `int`(), `str`(), …)
- **функции, объявленные пользователем**, которые объявляются и определяются пользователем для конкретной цели

ВАЖНО: имена встроенных функций являются зарезервированными словами, которые запрещено использовать в качестве названия своих переменных, функций или классов (***в противном случае, встроенная функция перезапишется!!!***)

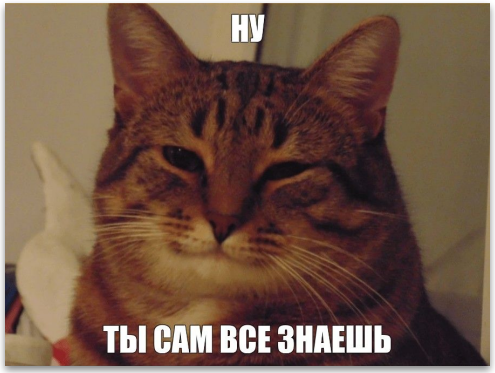
Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Определение функций

Именные функции

```
def greet(lang, count=1):
 if lang == 'es':
   return ['Hola' for i in range(count)]
 elif lang == 'fr':
   return ['Bonjour' for i in range(count)]
 else:
   return ['Hello' for i in range(count)]

greet('fr', count=3)
```



Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Определение функций

обязательный параметр

опциональный параметр

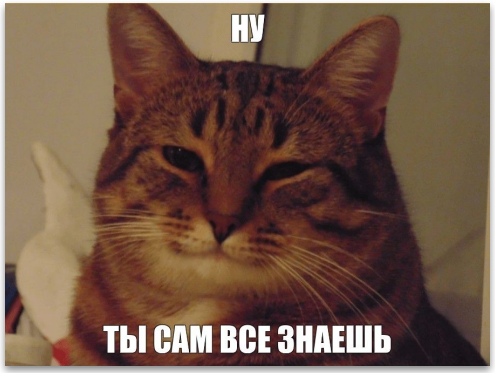
Именные функции

```
def greet(lang, count=1):
 if lang == 'es':
   return ['Hola' for i in range(count)]
 elif lang == 'fr':
   return ['Bonjour' for i in range(count)]
 else:
   return ['Hello' for i in range(count)]

greet('fr', count=3)
```

оператор выхода из функции

вызов функции



Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

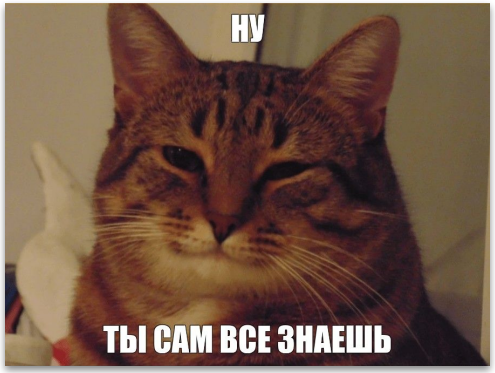
Определение функций

МФТИ

Именные функции

```
def greet(lang, count=1):
 if lang == 'es':
   return ['Hola' for i in range(count)]
 elif lang == 'fr':
   return ['Bonjour' for i in range(count)]
 else:
   return ['Hello' for i in range(count)]

greet('fr', count=3)
```

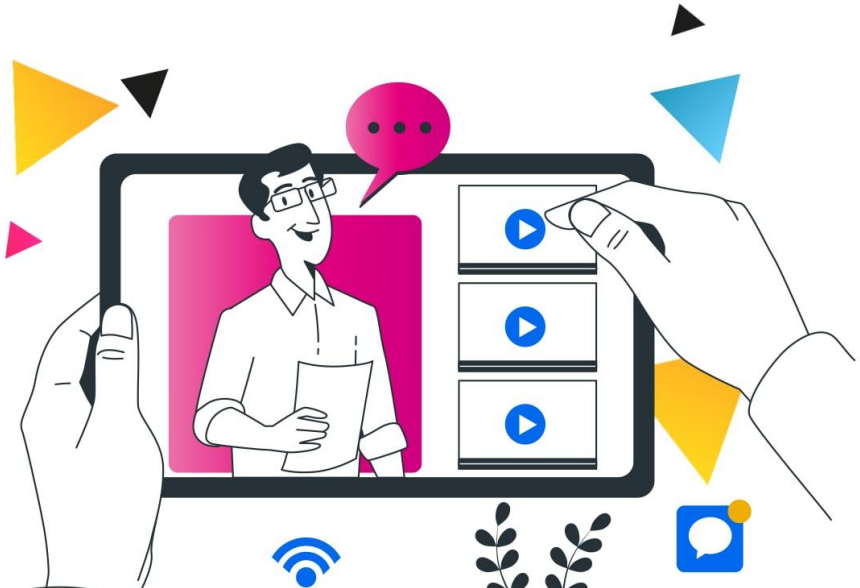


Лямбда-функции

```
f = lambda x: x**2
(lambda x: x*2)(12)
map(lambda x: x**2, my_list)
```

Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Ваши вопросы?



Работа с файлами

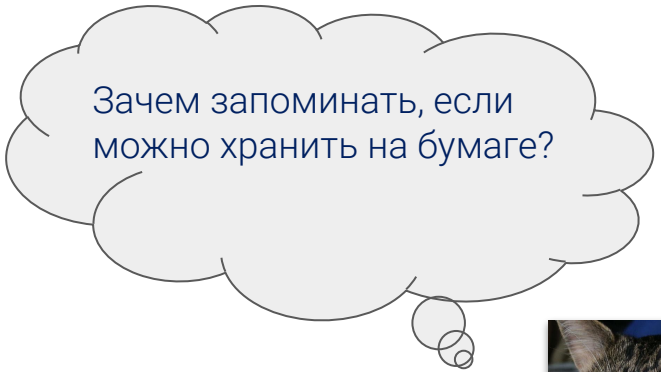
Работа с файлами

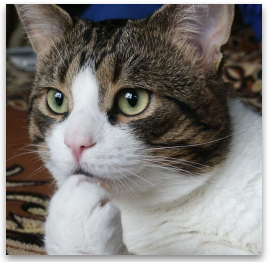
Любые данные, используемые для анализа необходимо где-то ***хранить***. Один из распространенных способов - **файлы**.

Существует два типа файлов:

- текстовые;
- двоичные.

При записи на определенный носитель информации все файлы являются двоичными.





Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Приёмы работы с файлами

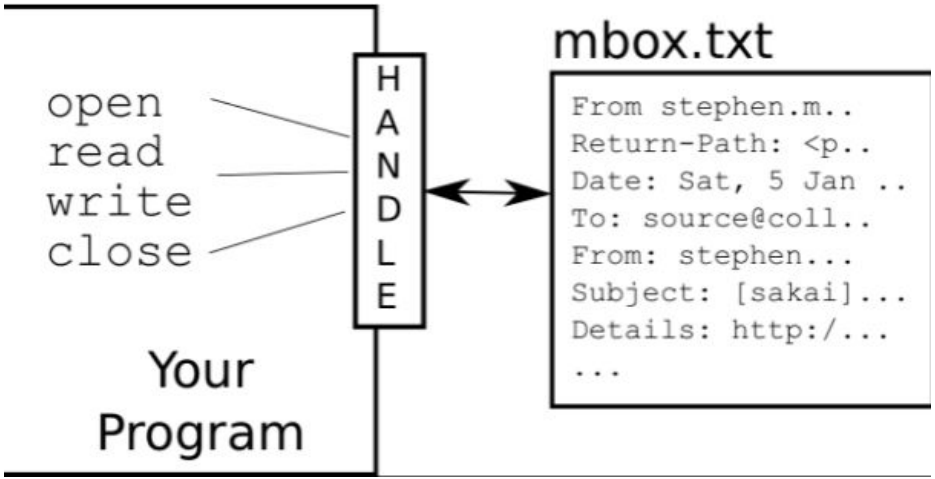
1

```
fhand = open('mbox.txt', 'r')
print(fhand)
fhand.close()
```

2

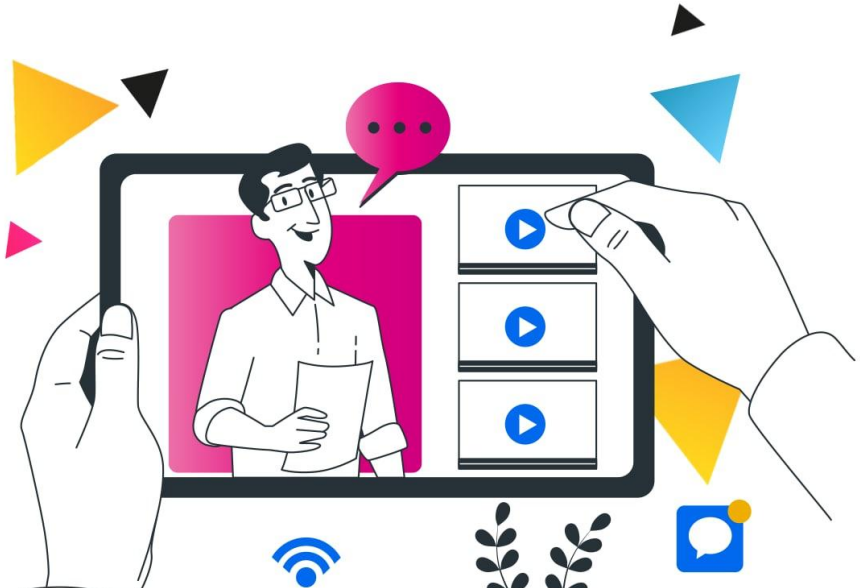
```
with open('mbox.txt', 'r') as f:
  print(f)
```

```
<_io.TextIOWrapper name='mbox.txt' mode='r' encoding='UTF-8'>
```



Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Ваши вопросы?



Практика в решении задач

Задача 1

Есть словарь, хранящий информацию о координатах точек в двумерном пространстве:

```
points_dict = {
 'A': (0.1, 25.445),
 'B': (1.51, 11.5),
 'C': (85.01, 52.8),
 'D': (13.141, 78.547),
 'E': (94.1, 0.5),
 'F': (0.15, 35.05),
 'G': (4., 11.745),
 'H': (6.14, 8.4),
 }
```

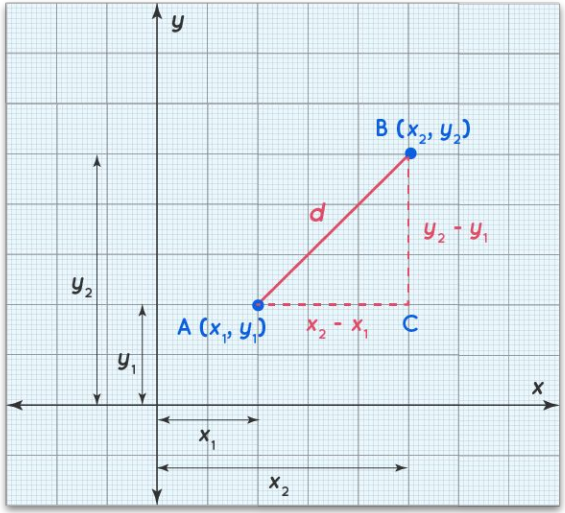
Пользователь вводит имена точек (примеры ввода: "A", "B", "a", "b", ...). Рассчитайте евклидово расстояние между указанными пользователем точками и распечатайте результат на экран.

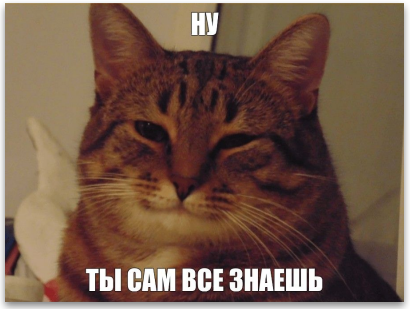
В случае, если введено имя, отсутствующее в словаре (для получения ключей используйте метод словаря keys()). Выведите сообщение об ошибке *"Одно из введенных имен отсутствует в списке. Примеры ввода: "A", "B", "a", "b", ..."*.

После вывода сообщения об ошибке необходимо осуществлять автоматический запрос пользователю для ввода имен точек.

Решение задачи 1

```
points_dict = {
 'A': (0.1, 25.445),
 'B': (1.51, 11.5),
 'C': (85.01, 52.8),
 'D': (13.141, 78.547),
 'E': (94.1, 0.5),
 'F': (0.15, 35.05),
 'G': (4., 11.745),
 'H': (6.14, 8.4),
 }
```






```
x, y = input('Введите имя точки: ').upper(), input('Введите имя точки: ').upper()
while (x not in points_dict.keys()) or (y not in points_dict.keys()):
  print('Одно из введенных имен отсутствует в списке. Примеры ввода: "A", "B", "a", "b", ...')
  x, y = input('Введите имя точки: ').upper(), input('Введите имя точки: ').upper()
print(((points_dict[y][0]-points_dict[x][0])**2+(points_dict[y][1]-points_dict[x][1])**2)**.5)
```

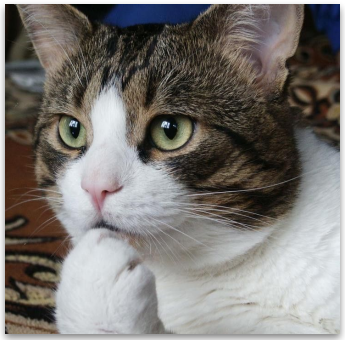
Задача 2

Имеется список списков строк. Необходимо написать функцию, которая преобразует его к одной строке, состоящей из всех элементов этого списка, разделенных некоторым символом.

Например: `[['1', '2'], ['3', '4']]`

будет преобразован к виду: `'1 2 3 4'`,

если разделитель – пробел.



Решение задачи 2

Имеется список списков строк. Необходимо написать функцию, которая преобразует его к одной строке, состоящей из всех элементов этого списка, разделенных некоторым символом.

Например: `[['1', '2'], ['3', '4']]`

будет преобразован к виду: `'1 2 3 4'`,

если разделитель – пробел.


```
my_list = [['1', '2'], ['3', '4']]
', '.join([item for sublist in my_list for item in sublist])
```

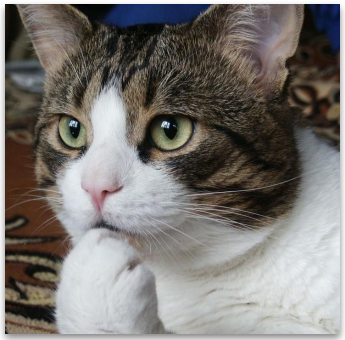
Полный скрипт: https://colab.research.google.com/drive/1uBVjtPBi0_OidTUdXVBMjw7OGobejTGb?usp=sharing

Задача 3*

Имеется список списков строк неограниченной вложенности. Необходимо написать функцию, которая преобразует его к одной строке, состоящей из всех элементов этого списка, разделенных некоторым символом. Например:

```
[[['1', '2'], ['1', '2']], [['3', '4'], ['3', '4']]]
```

будет преобразован к виду (если разделитель - пробел): `'1 2 1 2 3 4 3 4'`.



Решение задачи 3*

Имеется список списков строк неограниченной вложенности. Необходимо написать функцию, которая преобразует его к одной строке, состоящей из всех элементов этого списка, разделенных некоторым символом. Например:


```
[[['1', '2'], ['1', '2']], [['3', '4'], ['3', '4']]]
```

будет преобразован к виду (если разделитель - пробел): `'1 2 1 2 3 4 3 4'`.

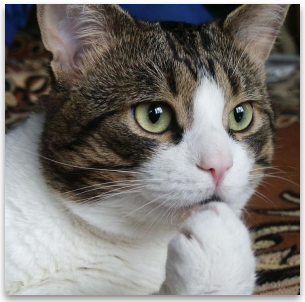
```
def magic(x, splitter=' '):
  if type(x) != list: return x
  return splitter.join([magic(i)for i in x])
```

Полный скрипт: https://colab.research.google.com/drive/1uBVjtPBi0_OidTUdXVBMjw7OGobejTGb?usp=sharing

Приёмы работы с файлами

Практический пример из анализа данных в биологии

Необходимо получить только “коровую” часть последовательностей, т.е. обрезать по 10 аминокислот с каждого конца.



Пример содержимого файла

```
>Mus|NP_612184.1|OO_H1.8 Mus_OO_H1.8_19923865
MAPGSVSSVSSSSFPSRDTSPSGSCGLPGADKPGPSCRRIQAGQRNPTML
HMVLEALKAREARQGTSVVAIKVYIQHKYPTVDTTRFKYLLKQALETGVR
RGLLTRPAHSKAKGATGSFKLVPKPKTKKACAPKAGRGAAGAKETGSKKS
GLLKKDQVGKATMEKGQKRRAYPCKAATLEMAPKKAKAKPKEVRKAPLKQ
DKAAGAPLTANGGQKVKRSGSRQEANAHGKTKGEKSKPLASKVQNSVASL
AKRKMADMAHTVTVVQGAETVQETKVPTPSQDIGHKVQPIPRVRKAKTPE
NTQA
>Homo|NP_722575.1|OO_H1.8 Homo_OO_H1.8_24475863
MAPGSVTSDISPSSTSTAGSSRSPESEKPGPSHGGVPPGGPSHSSLPVGR
RHPPVLRMVLEALQAGEQRRGTSVAAIKLYILHKYPTVDVLRFKYLLKQA
LATGMRRGLLARPLNSKARGATGSFKLVPKHKKKIQPRKMAPATAPRRAG
EAKGKGPKKPSEAKEDPPNVGKVKKAAKRPAKVQKPPPKPGAATEKARKQ
GGAAKDTRAQSGEARKVPPKPDKAMRAPSSAGGLSRKAKAKGSRSSQGDA
EAYRKTKAESKSSKPTASKVKNGAASPTKKKVVAKAKAPKAGQGPNTKAA
APAKGSGSKVVPAHLSRKTEAPKGPRKAGLPIKASSSKVSSQRAEA
>Homo|NP_001295191|OO_H1.8
MAPATAPRRAGEAKGKGPKKPSEAKEDPPNVGKVKKAAKRPAKVQKPPPK
PGAATEKARKQGGAAKDTRAQSGEARKVPPKPDKAMRAPSSAGGLSRKAK
AKGSRSSQGDAEAYRKTKAESKSSKPTASKVKNGAASPTKKKVVAKAKAP
KAGQGPNTKAAAPAKGSGSKVVPAHLSRKTEAPKGPRKAGLPIKASSSKVS
SQRAEA
>Trypanosoma|XP_846259.1|H2A.Z Trypanosoma_H2A.Z_72391930
MSLTGDDAVPQAPLVGGVAMSPEQASALTGGKLGGKAVGPAHGKGKGKGK
GKRGGKTGGKAGRRDKMTRAARADLNFPVGRIHSRLKDGLNRKQRCGASA
AIYCAALLEYLTSEVIELAGAAAKAQKTERIKPRHLLLAIRGDEELNQIV
NATIARGG
VVPFVHKSLEKKIIKKSKRGS
```

Приёмы работы с файлами

МФТИ

Практический пример из анализа данных в биологии

Необходимо получить только “коровую” часть последовательностей, т.е. обрезать по 10 аминокислот с каждого конца.

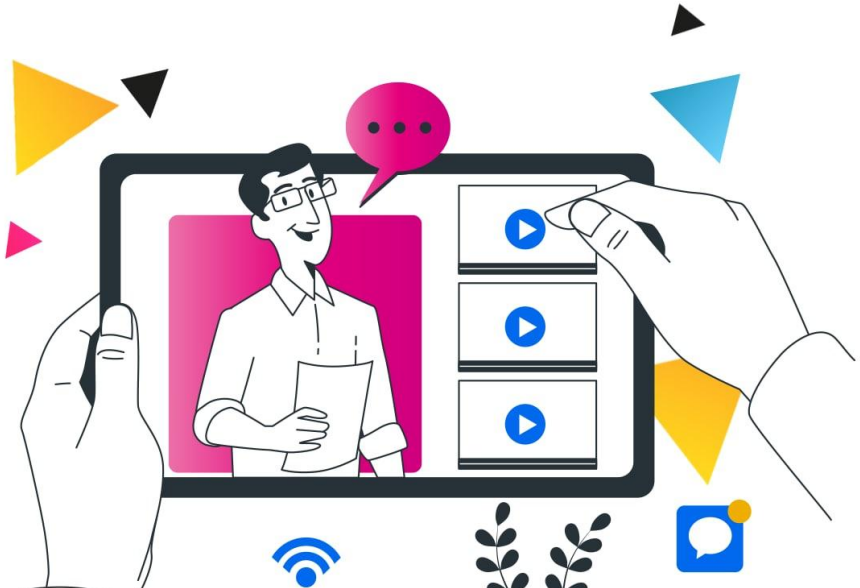
```
with open('my_fasta.fasta', 'r') as f, \
    open('new_fasta.fasta', 'w') as
new_f:
 for line in f.read().split('>')[1:]:
   line_split = line.split('\n', 1)
   seq = line_split[1].replace('\n', '')
   new_f.write(f'>{line_split[0]}\n')
   new_f.write(f'{seq[11:-10]}\n')
```

Пример содержимого файла

```
>Mus|NP_612184.1|OO_H1.8 Mus_OO_H1.8_19923865
MAPGSVSSVSSSSFPSRDTSPSGSCGLPGADKPGPSCRRIQAGQRNPTML
HMVLEALKAREARQGTSVVAIKVYIQHKYPTVDTTRFKYLLKQALETGVR
RGLLTRPAHSKAKGATGSFKLVPKPKTKKACAPKAGRGAAGAKETGSKKS
GLLKKDQVGKATMEKGQKRRAYPCKAATLEMAPKKAKAKPKEVRKAPLKQ
DKAAGAPLTANGGQKVKRSGSRQEANAHGKTKGEKSKPLASKVQNSVASL
AKRKMADMAHTVTVVQGAETVQETKVPTPSQDIGHKVQPIPRVRKAKTPE
NTQA
>Homo|NP_722575.1|OO_H1.8 Homo_OO_H1.8_24475863
MAPGSVTSDISPSSTSTAGSSRSPESEKPGPSHGGVPPGGPSHSSLPVGR
RHPPVLRMVLEALQAGEQRRGTSVAAIKLYILHKYPTVDVLRFKYLLKQA
LATGMRRGLLARPLNSKARGATGSFKLVPKHKKKIQPRKMAPATAPRRAG
EAKGKGPKKPSEAKEDPPNVGKVKKAAKRPAKVQKPPPKPGAATEKARKQ
GGAAKDTRAQSGEARKVPPKPDKAMRAPSSAGGLSRKAKAKGSRSSQGDA
EAYRKTKAESKSSKPTASKVKNGAASPTKKKVVAKAKAPKAGQGPNTKAA
APAKGSGSKVVPAHLSRKTEAPKGPRKAGLPIKASSSKVSSQRAEA
>Homo|NP_001295191|OO_H1.8
MAPATAPRRAGEAKGKGPKKPSEAKEDPPNVGKVKKAAKRPAKVQKPPPK
PGAATEKARKQGGAAKDTRAQSGEARKVPPKPDKAMRAPSSAGGLSRKAK
AKGSRSSQGDAEAYRKTKAESKSSKPTASKVKNGAASPTKKKVVAKAKAP
KAGQGPNTKAAAPAKGSGSKVVPAHLSRKTEAPKGPRKAGLPIKASSSKVS
SQRAEA
>Trypanosoma|XP_846259.1|H2A.Z Trypanosoma_H2A.Z_72391930
MSLTGDDAVPQAPLVGGVAMSPEQASALTGGKLGGKAVGPAHGKGKGKGK
GKRGGKTGGKAGRRDKMTRAARADLNFPVGRIHSRLKDGLNRKQRCGASA
AIYCAALLEYLTSEVIELAGAAAKAQKTERIKPRHLLLAIRGDEELNQIV
NATIARGG
VVPFVHKSLEKKIIKKSKRGS
```

Полный скрипт: https://colab.research.google.com/drive/1LIAp75-7ndNh6iuva22n4JW0sz8MILWT?usp=sharing

Ваши вопросы?



Итоги занятия

Итоги занятия

1. Обобщили материал по встроенным в Python коллекциям
2. Обобщили материал по работе с функциями и файлами в Python
3. Отработали задачи на использование коллекций
4. Отработали на практике работу с функциями и файлами

Дополнительные материалы по теме занятия

МФТИ

1. Полный скрипт семинара здесь: https://colab.research.google.com/drive/1MBrGcxRUaJ29bHQnQvStHswCP-_li-DK?usp=sharing
2. Список: https://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html
3. Кортеж: https://pythonworld.ru/tipy-dannyx-v-python/kortezhi-tuple.html
4. Множество: https://pythonworld.ru/tipy-dannyx-v-python/mnozhestva-set-i-frozenset.html
5. Словарь: https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html

Ваши вопросы?