ТЕксктовые методы

<https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html>

<https://www.python.org/> - основная документация по Python

colors = 'red blue green'

**print**(colors.**split**()[1])

# blue

animal = 'bear,1,0,0,1,0,0,1,1,1,1,0,0,4,0,0,1,1'

print(animal.split(','))

* \n — символ переноса строки;
* \t — символ горизонтальной табуляции;
* \v — символ вертикальной табуляции.

print(poem.split('\n'))

print('The {} currency rate on the date {} is {}'.format(currency, cur\_date, rate))

print('The {0} currency rate on the date {2} is {1}'.format(currency, rate, cur\_date))

print('The {} currency rate on the date {} is {:.3f}'.format(currency, cur\_date, rate))

#The EUR currency rate on the date 20.06.2021 is 86.370

f-строки использовать предпочтительней

print(f'The {currency} currency rate on the date {cur\_date} is {rate}')

#The EUR currency rate on the date 20.06.2021 is 86.37

print(f'The {currency} currency rate on the date {cur\_date} is {rate\*1.2}')

#The EUR currency rate on the date 20.06.2021 is 103.644

print(f'The {currency} currency rate on the date {cur\_date} is {rate:.3f}')

#The EUR currency rate on the date 20.06.2021 is 86.370

d = 12942021.24910

d\_str = str(d)

print(d\_str.split('.'))

# ['12942021', '2491']

* как создавать пустые списки;
* как записывать в них значения;
* какие типы данных и какой вложенности могут храниться в списках;
* какие бывают методы для работы со списками:
  + .append() — добавление элементов;
  + .remove() — удаление элементов;
  + .clear() — очистка списка;
  + .count() — определение количества вхождений определённого элемента;
  + .extend() — расширение одного списка другим.
* как сортировать и "переворачивать" списки.
* научились создавать словари;
* узнали о ключах и значениях, а также об их роли в словаре;
* изучили, какие типы данных могут храниться в словаре, какие из них могут выступать в качестве ключей, а какие — в качестве значений;
* научились доставать из словаря только ключи или только значения;
* изучили методы работы со словарями:
  + .clear() —  очистка словаря;
  + .update() —  обновление словаря;
  + .pop() —  удаление элементов из словаря;
* узнали, как можно обращаться к элементам словаря (используя квадратные скобки или метод .get());
* поняли, что можно приписывать значения по умолчанию тем ключам, которых, возможно, нет в словаре.
* научились создавать и модифицировать содержимое множеств:
  + .add();
  + .discard();
  + .remove();
* изучили объединение(.union()) и пересечение (.intersection()) множеств;
* познакомились с операцией .difference(), позволяющей понять, какие элементы присутствуют в одном множестве и отсутствуют в другом;
* узнали, что операция .issubset() позволяет увидеть вхождение элементов одного множества в другое.

Тернарный условный оператор применяется только в том случае, если внутри условного оператора присваивается значение переменной. Если действий внутри блоков больше, использовать тернарный оператор невозможно.

result = a **if** a > b **else** b

**Отлов всех ошибок**

**try**:

value\_1 = my\_dict[key\_1]

value\_2 = my\_dict[key\_2]

value\_1, value\_2 = float(value\_1), float(value\_2)

new\_value = value\_1 + value\_2

**except** Exception **as** e:

print('Код завершился с ошибкой: {}'.format(e))

В завершение мы предлагаем вам ознакомиться с дополнительными источниками, где вы сможете найти ещё больше практических задач на отработку синтаксиса *Python*:

* + [150 простейших задач по Python на отработку синтаксиса](http://www.kodesource.top/python-exercises/python-basic-exercises.php)
  + [Задачи по Python от Pythonist](https://pythonist.ru/category/tasks/)
  + [«5 классических задач по Python для начинающих с решениями»](https://proglib.io/p/5-klassicheskih-zadach-po-python-dlya-nachinayushchih-s-resheniyami-2022-03-04)

**Pandas**

[**https://habr.com/ru/articles/196980/**](https://habr.com/ru/articles/196980/)

[**https://lms-cdn.skillfactory.ru/assets/courseware/v1/0d324f2647f4ec15da909e5d6c083873/asset-v1:SkillFactory+DST-3.0+28FEB2021+type@asset+block/Pandas\_info.pdf**](https://lms-cdn.skillfactory.ru/assets/courseware/v1/0d324f2647f4ec15da909e5d6c083873/asset-v1:SkillFactory+DST-3.0+28FEB2021+type@asset+block/Pandas_info.pdf)

Однако мы составили небольшую **инструкцию по преобразованию признаков**:

* если признак распределён нормально, то его необходимо стандартизировать;
* если признак распределён ненормально, его необходимо нормализовать;
* если разброс значений небольшой, то можно обойтись без преобразования данных.