



Кортеж и множество (tuple и set)

Tuple

tuple это контейнер хранящий в себе элементы разделенные запятой. Элементы находятся между круглыми скобками `()`. Кортеж похожи на списки. Разница в том, что кортеж не изменяемый объект. Содержание кортежа нельзя изменить после того, как он создан.

```
tpl = ('Blue', 3)
print(tpl)

>> ('Blue', 3)
##
lst = ['Blue', 3]
tpl = tuple(lst)
print(tpl)

>> ('Blue', 3)
```

Пустой кортеж можно создать при помощи открытой-закрытой скобки `()` или функции **tuple()** без аргументов.

```
tpl = ()
print(tpl)

>> ()
##
tpl = tuple()
print(tpl)

>> ()
```

Индексинг и слайсинг работает как в списках.

```
tpl = (1, True, 'Hello')
print(tpl[1])

>> True
```

Set

set это коллекция не повторяющихся элементов. Элементы в множестве находятся между закругленных скобок. Порядок в множествах не важен. Из-за этого индексирование в множествах не работает. Дубликаты в множествах удаляются и остается только один из одинаковых элементов.

Пустое множество создается при помощи функции **set()** без аргументов.

```
st = {1, 2, 3, 4, 5}

>> {1, 2, 3, 4, 5}
##
lst = [1, 2, 2, 3, 3, 3]
st = set(lst)
print(st)

>> {1, 2, 3}
##
st = set()
print(st)

>> set()
##
st = {1, 2, 3, 4}
print(st[2])

>> Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    st[2]
TypeError: 'set' object is not subscriptable
```

Полезные функции

x in st

Возвращает `True` если элемент `x` существует внутри множества `st`. В противном случае возвращается `False`.

```
st = {1, 2, 3, 4}
print(1 in st)

>> True
```

`x not in st`

Возвращает `True` если элемент `x` не существует внутри множества `st`. В противном случае возвращается `False`.

```
st = {1, 2, 3, 4}
print(5 not in st)

>> True
```

`len(st)`

Возвращает длину множества `st`.

```
st = {1, 2, 3, 4, 5}
print(len(st))

>> 5
```

`st.isdisjoint(other)`

Возвращает `True`, если у множества `st` нет общих элементов с множеством `other`. В противном случае возвращает `False`.

- `other: set` → Множество

```
st1 = {1, 2, 3, 4}
st2 = {5, 6, 7, 8}
print(st1.isdisjoint())

>> True
```

`st.issubset(other)`

`st <= other`

Возвращает `True`, если все элементы множества `st` также являются элементами множества `other`. В противном случае возвращает `False`.

- *other*: `set` → Множество

```
st = {1, 2, 3, 4}
other = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

print(st.issubset(other))
print(st <= other)

>> True
>> True
```

`st < other`

То же самое что и `st <= other and st != other`.

```
st = {1, 2, 3, 4}
other = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

print(st < other)

>> True
##
st = {1, 2, 3, 4}
other = {1, 2, 3, 4}
```

```
print(st < other)

>> False
```

`st.issuperset(other)`

`st >= other`

Возвращает `True`, если все элементы множества `other` также являются элементами множества `st`. В противном случае возвращает `False`.

```
st = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
other = {1, 2, 3, 4}

print(st.issuperset())
print(st >= other)

>> True
>> True
```

`st > other`

То же самое что и `st >= other and st != other`.

```
st = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
other = {1, 2, 3, 4}

print(st > other)

>> True
##
st = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
other = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

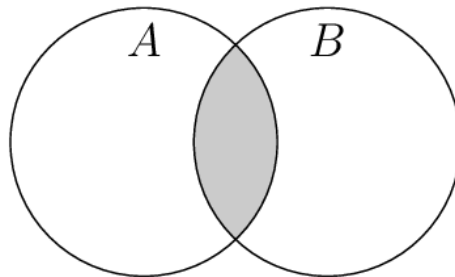
print(st > other)

>> False
```

st.intersection(*others)

st & other & ...

Возвращает пересечение множеств `st` и `others`. Пересечением двух множеств `A` и `B` является множество состоящие из элементов которые находятся и в `A` и в `B`, без дубликатов.



`others` может быть несколько множеств разделенные запятой.

```
# Страны входящие в НАТО
nato_set = {'Czech Rep', 'Iceland', 'Poland', 'Turkey', 'Estonia', 'Belgium', 'Canada', 'Norway', 'Germany', 'Greece', 'Montenegro', 'Albania', 'Bulgaria', 'Croatia', 'Portugal', 'North Macedonia', 'Spain', 'United States', 'United Kingdom', 'Romania', 'France', 'Denmark', 'Slovakia', 'Slovenia', 'Netherlands', 'Lithuania', 'Italy', 'Latvia', 'Hungary', 'Luxembourg'}

# Страны входящие в ЕС
eu_set = {'Austria', 'Lithuania', 'Republic of Cyprus', 'Belgium', 'Latvia', 'Portugal', 'Slovakia', 'Denmark', 'France', 'Estonia', 'Finland', 'Czech Republic', 'Spain', 'Ireland', 'Croatia', 'Bulgaria', 'Hungary', 'Slovenia', 'Netherlands', 'Malta', 'Luxembourg', 'Greece', 'Italy', 'Germany', 'Poland', 'Sweden', 'Romania'}

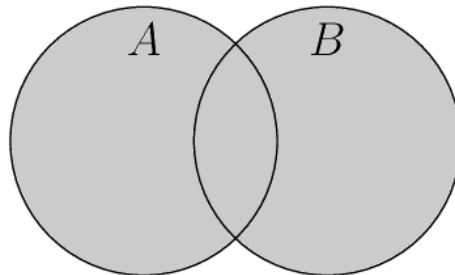
print(nato_set.intersection(eu_set)) # Распечатает страны входящие и в ЕС и в НАТО

>> {'Slovenia', 'Poland', 'Latvia', 'Romania', 'Bulgaria', 'Hungary', 'France', 'Greece', 'Spain', 'Lithuania', 'Belgium', 'Estonia', 'Portugal', 'Slovakia', 'Italy', 'Croatia', 'Germany', 'Netherlands', 'Denmark', 'Luxembourg'}
```

st.union(*others)

st | other | ...

Возвращает объединение множеств `st` и `others`.
Объединением двух множеств `A` и `B` является множество состоящее из всех элементов `A` и `B`, без дубликатов.



`others` может быть несколько множеств разделенные запятой.

```
# Страны входящие в НАТО
nato_set = {'Czech Rep', 'Iceland', 'Poland', 'Turkey', 'Estonia', 'Belgium', 'Canada', 'Norway', 'Germany', 'Greece', 'Montenegro', 'Albania', 'Bulgaria', 'Croatia', 'Portugal', 'North Macedonia', 'Spain', 'United States', 'United Kingdom', 'Romania', 'France', 'Denmark', 'Slovakia', 'Slovenia', 'Netherlands', 'Lithuania', 'Italy', 'Latvia', 'Hungary', 'Luxembourg'}

# Страны входящие в ЕС
eu_set = {'Austria', 'Lithuania', 'Republic of Cyprus', 'Belgium', 'Latvia', 'Portugal', 'Slovakia', 'Denmark', 'France', 'Estonia', 'Finland', 'Czech Republic', 'Spain', 'Ireland', 'Croatia', 'Bulgaria', 'Hungary', 'Slovenia', 'Netherlands', 'Malta', 'Luxembourg', 'Greece', 'Italy', 'Germany', 'Poland', 'Sweden', 'Romania'}

print(nato_set.union(eu_set)) # Распечатает все страны входящие в ЕС и в НАТО

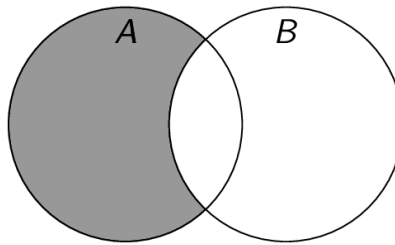
>> {'Netherlands', 'Italy', 'Ireland', 'Montenegro', 'Slovenia', 'Turkey', 'United States', 'Portugal', 'Germany', 'Estonia', 'Greece', 'Croatia', 'Iceland', 'Romania', 'France', 'Finland', 'Canada', 'Denmark', 'Spain', 'Belgium', 'Luxembourg', 'Bulgaria', 'Czech Republic', 'Austria', 'Norway', 'Hungary', 'Slovakia', 'Malta', 'Czech Rep', 'North Macedonia', 'Republic of Cyprus', 'Albania', 'Poland', 'Sweden', 'United Kingdom', 'Lithuania', 'Latvia'}
```

`st.difference(*others)`

`st` - `other` - ...

Возвращает разницу множеств `st` и `other`. Разницей множеств `A` и `B` является множеством элементов которые находятся в

A но не **B**.



others может быть несколько множеств разделенные запятой.

```
# Страны входящие в НАТО
nato_set = {'Czech Rep', 'Iceland', 'Poland', 'Turkey', 'Estonia', 'Belgium', 'Canada', 'Norway', 'Germany', 'Greece', 'Montenegro', 'Albania', 'Bulgaria', 'Croatia', 'Portugal', 'North Macedonia', 'Spain', 'United States', 'United Kingdom', 'Romania', 'France', 'Denmark', 'Slovakia', 'Slovenia', 'Netherlands', 'Lithuania', 'Italy', 'Latvia', 'Hungary', 'Luxembourg'}

# Страны входящие в ЕС
eu_set = {'Austria', 'Lithuania', 'Republic of Cyprus', 'Belgium', 'Latvia', 'Portugal', 'Slovakia', 'Denmark', 'France', 'Estonia', 'Finland', 'Czech Republic', 'Spain', 'Ireland', 'Croatia', 'Bulgaria', 'Hungary', 'Slovenia', 'Netherlands', 'Malta', 'Luxembourg', 'Greece', 'Italy', 'Germany', 'Poland', 'Sweden', 'Romania'}

print(nato_set.difference(eu_set)) # Распечатает страны входящие только в НАТО и не входящие в ЕС

>> {'Iceland', 'Albania', 'North Macedonia', 'Canada', 'Norway', 'Montenegro', 'United Kingdom', 'Czech Rep', 'United States', 'Turkey'}
```