

МОДУЛЬ `itertools`

Размещения с повторениями

Для работы с размещениями с повторениями есть функция `itertools.product`:

`itertools.product(итерируемый_объект, repeat=длина_размещения)`.

Функция `product` возвращает размещениями с повторениями для элементов передаваемых итерируемых объектов длиной, указанной в параметре `repeat`.

Если значение `repeat` не задано, то для нескольких итерируемых объектов комбинации, где на первом месте будут расположены элементы первого итерируемого объекта, на втором – второго и т.д. Если же в качестве аргументов был подан только один итерируемый объект, то в качестве результата последовательно вернуться кортежи из одного элемента.

Для последовательности `s` из 3 элементов – `[s1, s2, s3]`. Результатом работы функции `product(s, repeat=2)` будет итератор, который вернет следующие кортежи всегда в таком порядке, независимо от того, есть ли в последовательности повторяющиеся элементы:

`(s1, s1), (s1, s2), (s1, s3), (s2, s1), (s2, s2), (s2, s3), (s3, s1), (s3, s2), (s3, s3)`

Примеры:

```
1 from itertools import product
2 print(list(product('123', repeat=2)))
3 # [('1', '1'), ('1', '2'), ('1', '3'), ('2', '1'), ('2', '2'), ('2', '3'), ('3', '1'), ('3', '2'), ('3', '3')]
4 print(list(product('12', '34'))) # repeat=1 [('1', '3'), ('1', '4'), ('2', '3'), ('2', '4')]
```

Размещения без повторений

Функция, возвращающая все размещения длиной `r` элементов последовательности – `itertools.permutations`:

`itertools.permutations(итерируемый_объект, r=длина_размещения)`

Предположим, что у нас имеет последовательность `s` из 3 элементов – `[s1, s2, s3]`. Результатом работы функции `permutations(s, r=2)` будет итератор, который вернет следующие кортежи всегда в таком порядке, независимо от того, есть ли в последовательности повторяющиеся элементы:

`(s1, s2, s3), (s1, s3, s2), (s2, s1, s3), (s2, s3, s1), (s3, s1, s2), (s3, s2, s1)`

Примеры:

```
1 from itertools import permutations
2 print(list(permutations('123', r=2))) # [('1', '2'), ('1', '3'), ('2', '1'), ('2', '3'), ('3', '1'), ('3', '2')]
```

Сочетания

Функция для получения сочетаний без повторений –
itertools.combinations:

itertools.combinations (итерируемый_объект, r=длина_сочетания)

Количество элементов в сочетании устанавливается через параметр *r*, который является обязательным параметром. Для последовательности *s* из 3 элементов – [s1, s2, s3]. Результатом работы функции combinations(*s*, *r*=2) будет итератор, который вернет следующие кортежи всегда в таком порядке, независимо от того, есть ли в последовательности повторяющиеся элементы:

(s1, s2), (s1, s3), (s2, s3)

Примеры:

1	from itertools import combinations
2	print(list(combinations('123', r=2))) # [('1', '2'), ('1', '3'), ('2', '3')]