itertools-combino

September 12, 2018

In [1]:
```
# Combinatoric Iterators

import itertools as it
```

In [2]:
```
# product(iter1, iter2)
# cartesian product of the two iterable containers

print (list(it.product('AB','12')))
```

```
[('A', '1'), ('A', '2'), ('B', '1'), ('B', '2')]
```

In [3]:
```
# permutations(iter, group_size)
# prints all possible permutation of all elements

print (list(it.permutations('abc',1)), end='\n\n')
print (list(it.permutations('abc',2)), end='\n\n')
print (list(it.permutations('abc',3)), end='\n\n')
```

```
[('a',), ('b',), ('c',)]

[('a', 'b'), ('a', 'c'), ('b', 'a'), ('b', 'c'), ('c', 'a'), ('c', 'b')]

[('a', 'b', 'c'), ('a', 'c', 'b'), ('b', 'a', 'c'), ('b', 'c', 'a'), ('c', 'a', 'b'), ('c', 'b
```

In [4]:
```
# combinations(iterable, group_size)
# This iterator prints all the possible combinations(without replacement) of the conta

print (list(it.combinations('123',2)), end='\n\n')
```

```
[('1', '2'), ('1', '3'), ('2', '3')]
```

In [5]:
```
# combinations_with_replacement(iterable, group_size)
# to print every combination with replacement

print (list(it.combinations_with_replacement('123',2)), end='\n\n')
```

```
[('1', '1'), ('1', '2'), ('1', '3'), ('2', '2'), ('2', '3'), ('3', '3')]
```