iter-tools1

September 11, 2018

In [34]:
```
import itertools as it
# https://www.geeksforgeeks.org/iterator-functions-in-python-set-1/
# https://www.youtube.com/watch?v=xK7E2YmjyAc
```

In [35]:
```
def simple_func(a, b):        # Two parameters needed for it.accumulator
    return (a+b)/(a-b)
```

In [36]:
```
a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
b = ['zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
c = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

In [37]:
```
# chain()
# groups together multiple lists into one single iterable

x = []
for i in it.chain(a, b, c):
    x.append(i)

print(x)        # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'zero', 'one', 'two', 'three', 'four
print(list(it.chain(a[5:], c[5:])))        # [5, 6, 7, 8, 9, 'f', 'g', 'h', 'i', 'j']
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven',
[5, 6, 7, 8, 9, 'f', 'g', 'h', 'i', 'j']
```

In [38]:
```
# chain.from_iterable()

print(list(it.chain.from_iterable( [a[:3], c[:3]] )))        # Similar as chain() but
```

```
[0, 1, 2, 'a', 'b', 'c']
```

In [39]:
```
# accumulate(iter, func=sum)

print(list(it.accumulate(a, simple_func)))
print(list(it.accumulate(a)))        # [0, 1, 3, 6, 10, 15, 21, 28, 36, 45]
```

```
[0, -1.0, -0.3333333333333333, -0.7999999999999999, -0.6666666666666667, -0.7647058823529411,
[0, 1, 3, 6, 10, 15, 21, 28, 36, 45]
```

```
In [40]: # compress(iter, selector)
         # selectively picks the values to print from the passed container according to the bo
         container = 'abcdefghijklm'
         selector = [1,0,0,0,0,1,0,0,1,0,0,0,0]
         print(list(it.compress(container, selector)))
```

```
['a', 'f', 'i']
```

```
In [41]: # dropwhile(func, seq)
         # starts printing the characters only after the func. in argument returns false

         print (list(it.dropwhile(lambda x : x%2==0, [2, 4, 5, 7, 8, 2])))
```

```
[5, 7, 8, 2]
```

```
In [42]: # filterfalse(func, seq)
         # prints only values that return false for the passed function.

         print (list(it.filterfalse(lambda x : x%2==0, [2, 4, 5, 7, 8, 2])))
```

```
[5, 7]
```