**6 kyu**

**Range Parser**

14492% of 49141 of147[DeepSpace](https://www.codewars.com/users/DeepSpace)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/range-parser/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/range-parser/solutions/python)

[Discourse (13)](https://www.codewars.com/kata/range-parser/discuss/python)

* Add to Collection
* |
* Share this kata:

In this Kata you are to implement a function that parses a string which is composed from tokens of the form 'n1-n2,n3,n4-n5:n6' where 'nX' is a positive integer. Each token represent a different range:

'n1-n2' represents the range n1 to n2 (inclusive in both ends). 'n3' represents the single integer n3. 'n4-n5:n6' represents the range n4 to n5 (inclusive in both ends) but only includes every other n6 integer.

Notes:

1. The input string doesn't not have to include all the token types.
2. All integers are assumed to be positive.
3. Tokens may be separated by ',', ', ' or ,.

Some examples:

'1-10,14, 20-25:2' -> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 20, 22, 24]

'5-10' -> [5, 6, 7, 8, 9, 10]

'2' -> [2]

The output should be a list of integers.

[**https://www.codewars.com/kata/range-parser/python**](https://www.codewars.com/kata/range-parser/python)

**def range\_parser(string):**

**res = []**

**for range\_ in string.split(','):**

**first\_last, \_, step = range\_.partition(':')**

**first, \_, last = first\_last.partition('-')**

**res += range(int(first), int(last or first) + 1, int(step or 1))**

**return res**

**def** my\_split(s, seps):

    res = [s]

**for** sep **in** seps:

        s, res = res, []

**for** seq **in** s:

            res += seq.split(sep)

**return** res

**def** range\_parser(s):

    rangos = s.split(",")

    ans = []

**for** i **in** range(0, len(rangos)):

        r = rangos[i].strip()

**if**("-" **in** r **and** ":" **in** r):

            partes = my\_split( r, ["-",":"])

**for** x **in** range(int(partes[0]), int(partes[1]) + 1, int(partes[2])):

                ans.append(x)

**elif**("-" **in** r):

            partes = r.split("-")

**for** x **in** range(int(partes[0]), int(partes[1]) + 1):

                ans.append(x)

**else**:

            ans.append(int(r))

**return** ans

s = "1-10,14, 20-25:2"

**print**(range\_parser(s))