Uniformly Partitioning A String Using Textwrap

2025-11-13

Table of contents

Today I Learned added on 2025-11-13, learned on 2025-11-08; edited on 2026-06-14.

A merchant on Ebay sent me a list of item ID codes (each 12 characters long) as a newline separated list. When I copied and pasted the list into a text file (to have two separate windows open at once), the characters were concatenated into a single string — how annoying! I tried different types of copying-and-pasting (pasting using `Cmd-V` and `Cmd-Shift-V`) to no avail. I then wondered if Python could help me avoid manually cutting out each 12 character string from the single long string. The answer is "Yes!" and there is a package called `textwrap` that can help with this task specifically.

```
# concatenated str
c_str = "432853242343242343242342343243276575"

# partition size
p_size  = 12

# uniformly partitioned string; ['432853242343', '242343242342',
 ↪  '343243276575']
up_str = [c_str[i:i+p_size] for i in range(0, len(c_str),  p_size) if
 ↪  len(c_str) % p_size == 0]
```

The simpler method is to use Python's `textwrap` package:

```
import textwrap


c_str = "432853242343242343242342343243276575"

p_size  = 12

if len(c_str) % p_size == 0:
    # ['432853242343', '242343242342', '343243276575']
    up_str = textwrap.wrap(c_str, p_size)
```