[CS 11] Prac 8c – Word Wrap

Problem Statement

You are making a utility program to make your documentation and comments nicer. As part of this, you would like to have **word wrap**—the ability to display text such that the words fit within only c columns of characters, for some given c. For example, consider the following text:

```
They were all dead.

The final   gunshot
was an exclamation mark on everything that had led to this
point.   I released
my
finger  from  the  trigger,
and then it was over.
```

Displayed in $c = 25$ columns:

```
They were all dead. The
final gunshot was an
exclamation mark on
everything that had led
to this point. I released
my finger from the
trigger, and then it was
over.
```

Notice that each line has at most 25 characters, and there are no leading or trailing spaces, or two consecutive spaces.

Given some text and the value of c, display the text with at most c columns.

If there is a token with more than c characters, simply display that token on its own line. It is okay for lines like that to exceed c characters.

Note that we can only change whitespace—spaces and newline (a.k.a. "enter") characters. Anything other than those must be preserved, including case and punctuation.

Submit solution [CS 11] Practice 8

My submissions

Points: 230 (partial)
Time limit: 4.0s
Memory limit: 1G

Author: kvatienza (Kevin Atienza)

Problem type

Allowed languages ~~NONE~~, py3

Task Details

Your task is to implement a function called `wrap`. This function has two parameters:

- the first is a `str` representing the text.
- the second is the `int` c, the number of columns

The function must return a `list` of `str`s denoting the individual lines of the text, wrapped into c columns.

Restrictions

(See 8a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 600.

Example Calls

Example 1 Function Call

```
wrap("""They were all dead.

The final   gunshot
was an exclamation mark on everything that had led to this
point.   I released
my
finger  from  the  trigger,
and then it was over.
""", 25)
```

Example 1 Return Value

```
[
    "They were all dead. The",
    "final gunshot was an",
    "exclamation mark on",
    "everything that had led",
    "to this point. I released",
    "my finger from the",
    "trigger, and then it was",
    "over.",
]
```

Example 2 Function Call

```
wrap("""They were all dead.

The final   gunshot
was an exclamation mark on everything that had led to this
point.   I released
my
finger  from  the  trigger,
and then it was over.
""", 16)
```

Example 2 Return Value

```
[
    "They were all",
    "dead. The final",
    "gunshot was an",
    "exclamation mark",
    "on everything",
    "that had led to",
    "this point. I",
    "released my",
    "finger from the",
    "trigger, and",
    "then it was",
    "over.",
]
```

Example 3 Function Call

```
wrap("""They were all dead.

The final   gunshot
was an exclamation mark on everything that had led to this
point.   I released
my
finger  from  the  trigger,
and then it was over.
""", 9)
```

Example 3 Return Value

```
[
    "They were",
    "all dead.",
    "The final",
    "gunshot",
    "was an",
    "exclamation",
    "mark on",
    "everything",
    "that had",
    "led to",
    "this",
    "point. I",
    "released",
    "my finger",
    "from the",
    "trigger,",
    "and then",
    "it was",
    "over.",
]
```

Constraints

- The function `wrap` will be called at most $60{,}000$ times.
- The total lengths of all texts will be at most $250{,}000$.
- Each text will have a length of at most $250{,}000$.
- The text will have at least one non-whitespace character.
- $1 \le c \le 200$
- Each word in the text is a nonempty string of at most 300 English letters, spaces, newline characters (`\n`), or one of the characters: `.,!?-` .

Scoring

- You get 30 ❤ points if you solve all test cases where:
  - the length of the text is ≤ 160,
  - the total lengths of all text is ≤ 1600.
  - $c \le 80$
- You get 30 ❤ points if you solve all test cases where:
  - the length of the text is $\le 4{,}000$,
  - the total lengths of all text is $\le 8{,}000$.
  - $c \le 80$
- You get 120 ❤ points if you solve all test cases where:
  - $c \le 80$
- You get 50 ❤ points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.