

Re: textwrap.dedent replaces tabs?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-12/msg03238.html>

- *From:* Frederic Rentsch <anthra.norell@xxxxxxxxxxx>
 - *Date:* Fri, 22 Dec 2006 22:23:26 +0100
-

Tom Plunket wrote:

Frederic Rentsch wrote:

Well, there is that small problem that there are leading tabs that I want stripped. I guess I could manually replace all tabs with eight spaces (as opposed to 'correct' tab stops), and then replace them when done, but it's probably just as easy to write a non-destructive dedent.

This should do the trick:

```
>>> Dedent = re.compile ('^\s+')
>>> for line in lines: print Dedent.sub ("", line)
```

The fact that this doesn't do what dedent() does makes it not useful. Stripping all leading spaces from text is as easy as calling lstrip() on each line:

My goodness! How right your are.

```
text = '\n'.join([line.lstrip() for line in text.split('\n')])
```

alas, that isn't what I am looking for, nor is that what textwrap.dedent() is intended to do.

-tom!

Following a call to dedent () it shouldn't be hard to translate leading groups of so many spaces back to tabs. But this is probably not what you want. If I understand your problem, you want to restore the dedented line to its original composition if spaces and tabs are mixed and this doesn't work because the information doesn't survive dedent (). Could the information perhaps be passed around dedent ()? Like this: make a copy of your

Re: textwrap.dedent replaces tabs?

lines and translate the copy's tabs to so many (8?) marker bytes (e.g. ascii 0). Dedent the originals. Left-strip each of the marked line copies to the length of its dedented original and translate the marked groups back to tabs.

Frederic