▯ ▯ ▯ ▯ ▯ ▯ ----- Original Message -----▯ > From: Ian Kelly < ian.g.kelly@gmail.com>▯ > To: Python <python-list@python.org>▯ > Cc: ▯ > Sent: Thursday, June 5, 2014 10:18 PM▯ > Subject: Re: Unicode and Python - how often do you index strings?▯ > ▯ > On Thu, Jun 5, 2014 at 1:58 PM, Paul Rubin <no.email@nospam.invalid> ▯ > wrote:▯ >> Ryan Hiebert < ryan@ryanhiebert.com> writes:▯ >>> How so? I was using line=line[:-1] for removing the trailing newline, ▯ > and▯ >>> just replaced it with rstrip('\n'). What are you doing ▯ > differently?▯ >> ▯ >> rstrip removes all the newlines off the end, whether there are zero or▯ >> multiple.? In perl the difference is chomp vs chop.? line=line[:-1]▯ >> removes one character, that might or might not be a newline.▯ > ▯ > Given the description that the input string is "a textfile line", if▯ > it has multiple newlines then it's invalid.▯ > [> Personally I tend toward rstrip('\r\n') so that I don't have ▯ > to worry▯ > about files with alternative line terminators.▯ ▯ I tend to use: s.rstrip (os.linesep)▯ ▯ > If you want to be really picky about removing exactly one line▯ > terminator, then this captures all the relatively modern variations:▯ > re.sub('\r?\n$|\n?\r$', line, '', count=1)▯ ▯ or perhaps: re.sub ("[^ \S]+$", "", line)