**Challenge 1**

[www.pythonchallenge.com/pc/def/map.html](http://www.pythonchallenge.com/pc/def/map.html)



everybody thinks twice before solving this.

g fmnc wms bgblr rpylqjyrc gr zw fylb. rfyrq ufyr amknsrcpq ypc dmp. bmgle gr gl zw fylb gq glcddgagclr ylb rfyr'q ufw rfgq rcvr gq qm jmle. sqgle qrpgle.kyicrpylq() gq pcamkkclbcb. lmu ynnjw ml rfc spj.

Considering the picture and the page title ‘What about making trans?’ and even the link for the page …map.html, it looks like we need to decode the gibberish. Notice that on the notepad the decodes are two letters apart: K(L)M, O(P)Q, E(F)G – hint says … thinks twice …

So, pychallenge1.py just shifts the encoded stuff two letters a -> c, b -> d, etc.

To process the gibberish, I just copy/pasted into the source code and made it a list of sentences. (There is a single quote that needs to be escaped \’.)

This the message decoded:

i hope you didnt translate it by hand.

thats what computers are for.

doing it in by hand is inefficient and that's why this text is so long.

using string.maketrans() is recommended.now apply on the url.

I didn’t know about **string.maketrans().** Looking it up, it uses a translation table. It seems useful but too much trouble for this problem, since we’ve already decoded the gibberish. Shifting ‘map’ 2 letters gives ‘ocr’ and …ocr.html comes up as the next challenge.

I decided to see if I could figure out how to get the gibberish from the web page in python using **requests.get(url).text.**

The final programs that does everything all at once is pychallenge1.py. pcutils has a new function to get a string. It is kind of kludge-y, and I will try to develop it better when I use it more.