

Forum:[Hack Forums](#)

Thread: [Quik insight on his Pig Latin translator???](#)

Gpx45 - November 19, 2014, 12:45 pm

Hey guys, I made this translator which converts regular words into pig latin which basically is taking the first letter putting it at the end of the word and adding "ay" now this is the script:

```
Code pyg = 'ay'
```

```
original = raw_input('Enter a word:')
```

```
if len(original) > 0 and original.isalpha():
```

```
    print original
```

```
    word = original.lower()
```

```
    first = word[0]
```

```
    new_word = word + first + pyg
```

```
    print(new_word [1:len(new_word)])
```

```
else:
```

```
    print 'empty'
```

ok its works 100% however the professor wants something very specific that I don't quite understand and I really need help because its killing the cells!!!!

This is what he said: "It looks like you printed the correct translation of "dog", but make sure to set new_word equal to the slice as well."

"dog the word i had put in to translate?" it came out "ogday"

Now I don't understand what that means??? does he mean make another variable called "new_word" again and equal it to the list of the slice???

MoonCakeRuler - November 19, 2014, 3:56 pm

The 'slice' he mentions is this - [1:len(new_word)]. He basically wants you to set new_word equal to the final result.

You can do either of these:

```
Code new_word = new_word[1:]
```

Or, you can just do it when you set new_word like this:

```
Code new_word = (word + first + pyg)[1:]And then just print new_word.
```

Also, you don't need to use len(new_word). You can just do [1:] to get all the elements from 1st index onwards.

Cheers.

Gpx45 - November 19, 2014, 9:18 pm

(Nov 19, 2014 04:56 PM)MoonCakeRuler Wrote: [\[CITE\]](#) The 'slice' he mentions is this - [1:len(new_word)]. He basically wants you to set new_word equal to the final result.

You can do either of these:

```
Code new_word = new_word[1:]
```

Or, you can just do it when you set new_word like this:

```
Code new_word = (word + first + pyg)[1:]And then just print new_word.
```

Also, you don't need to use `len(new_word)`. You can just do `[1:]` to get all the elements from 1st index onwards.

Cheers.

Ahhh I fucking see it its so simple fuck!!! yes!! thanks MoonCake!!

Brandon1027 - November 22, 2014, 1:49 pm

I've made my own Pig Latin translator that is pretty easy to follow through. Feel free to use this if you decide that you like this one better.

```
Code original = raw_input('Type in a word').lower()
if len(original) > 0 and original.isalpha():
    print original
else:
    print 'Type in a word or phrase that contains only alphabetical letters and more than 0 letters.'

def pig_latin():

    first_letter = original[0]
    base_word = original[1:]
    new_word = base_word + first_letter + 'ay'

    print new_word

    return original

    pig_latin()

pig_latin()
```

ZeZil - November 22, 2014, 6:54 pm

Rofl, you talked about an instructor... This is exactly one of the very early Python lessons on codecademy. :P