

Prácticas de SAR

Sistemas de Almacenamiento y Recuperación de información

Práctica 1. Pig Latin

Traductor de “Pig Latin”

Descripción del problema

Un grupo de agentes de la policía del Vaticano está planeando un viaje a los Estados Unidos. Dado que sólo hablan Pig Latin, tendrán que traducir un montón de frases en inglés.

Escriba un programa fácil para traducir de Inglés a Pig Latin.

(Tim Wilson’s Python Programming Tips)

¿Qué es Pig Latin?

El Pig Latin es un juego con el idioma inglés.

“Good Morning” en Pig Latin se dice “Oodgay Orningmay”. El Pig Latin lo usan los niños para divertirse o para conversar secretamente sobre adultos u otros niños. Recíprocamente, los adultos a veces lo usan para hablar de temas sensibles que quieren que los niños no entiendan.

Los turistas anglohablantes a veces usan el Pig Latin para disimular sus conversaciones cuando viajan por países donde el inglés es el segundo idioma.

http://es.wikipedia.org/wiki/Pig_Latin

Traductor de “Pig Latin”

Reglas **simplificadas** de conversión a Pig Latin:

- No tratemos palabras con acentos.
- La traducción se hace palabra a palabra.
- Las palabras traducidas se separan por un único espacio en blanco.
- Si una palabra no comienza con una letra se deja igual.
- Para las palabras que comienzan por consonantes, se mueven todas las consonantes antes de la primera vocal al final y se agrega la sílaba “ay”.
- Para palabras que comienzan por vocal (también consideramos vocal la “y”), simplemente se agrega “yay” al final de la palabra.

Ejemplos

Ejemplos de traducción a “Pig Latin”

- mess: essmay
- father: atherfay
- Rwanda: Andarway

- choir: oirchay
- ant: antyay
- 4G: 4G

Ejercicio

¿Qué debo hacer?

Escribe un programa en python que traduzca a Pig Latin.

El programa tendrá dos modos de funcionamiento:

- Si no recibe argumentos, entrará en un bucle pidiendo frases por teclado y traduciéndolas. El programa continuará hasta que introduzcamos una cadena vacía.
- Si le pasamos como argumento un fichero, el programa creará otro añadiendo la "_latin" al final del nombre del fichero. Se debe respetar la extensión del fichero original.

¿Qué debo hacer?

Se debe tener en cuenta:

- La traducción de la frase se hace palabra a palabra.
- Se deben respetar los siguientes signos de puntuación: ‘,’ ‘;’, ‘:’, ‘?’ ‘!’.
- Se deben respetar las mayúsculas en dos casos:
 - si una palabra empieza con mayúscula su traducción también debe empezar con mayúscula.
 - si una palabra está toda en mayúsculas su traducción debe estar toda en mayúsculas.
 - otras mayúsculas pueden ser ignoradas.

¿Qué debo hacer?

Ejemplo de funcionamiento

- Sin argumento:

```
$ python pig_latin.py
```

ENGLISH: Spam, SPAM, Spam, Egg and Spam;

PIG LATIN: Amspay, AMSPAY, Amspay, Eggyay andyay Amspay;

ENGLISH: brandy and a fried egg on top and Spam

PIG LATIN: andybray andyay ayay iedfray eggyay onyay optay andyay Amspay

ENGLISH: 4G and spam

PIG LATIN: 4G andyay amspay

ENGLISH:

- Con un argumento:

```
$ python pig_latin.py "Spam, Spam, Spam, Egg and Spam;"
```

Amspay, Amspay, Amspay, Eggyay andyay Amspay;

¿Qué debo hacer?, Plantilla

```
#!/usr/bin/env python
#! -*- encoding: utf8 -*-
# 1.1.- Pig Latin

import sys

def piglatin_word(word):
    """
    Esta funcion recibe una palabra en ingles y la traduce a Pig Latin

    :param word: la palabra que se debe pasar a Pig Latin
    :return: la palabra traducida
    """
    # completar
    return word
```

¿Qué debo hacer?, Plantilla

```
#!/usr/bin/env python
#! -*- encoding: utf8 -*-

# 1.- Pig Latin

import sys
import re

class Translator():

    def __init__(self, punt=None):
        """
        Constructor de la clase Translator

        :param punt(opcional): una cadena con los signos de puntuación
                                que se deben respetar
        :return: el objeto de tipo Translator
        """
        if punt is None:
            self.re = re.compile("(\w+)([.,;?!]*)")
        else:
            self.re = re.compile("(\w+)(["+punt+"]*)")
```

¿Qué debo hacer?, Plantilla

```
def translate_word(self, word):
    """
    Este método recibe una palabra en inglés y la traduce a Pig Latin

    :param word: la palabra que se debe pasar a Pig Latin
    :return: la palabra traducida
    """

    # sustituir
    new_word = word

    return new_word

def translate_sentence(self, sentence):
    """
    Este método recibe una frase en inglés y la traduce a Pig Latin

    :param sentence: la frase que se debe pasar a Pig Latin
    :return: la frase traducida
```

```

    """

    # sustituir
    new_sentence = sentence

    return result

def translate_file(self, filename):
    """
    Este método recibe un fichero y crea otro con su traducción a Pig Latin

    :param filename: el nombre del fichero que se debe traducir
    :return: True / False
    """

    # rellenar

if __name__ == "__main__":
    if len(sys.argv) > 2:
        print('Syntax: python %s [filename]' % sys.argv[0])
        exit
    else:
        t = Translator()
        if len(sys.argv) == 2:
            t.translate_file(sys.argv[1])
        else:
            while True:
                sentence = input("ENGLISH: ")
                if len(sentence) < 2:
                    break
                print("PIG LATIN:", t.translate_sentence(sentence))

```

¿Qué debo hacer?, Plantilla

```

def translate_file(self, filename):
    """
    Este método recibe un fichero y crea otro con su traducción a Pig Latin

    :param filename: el nombre del fichero que se debe traducir
    :return: True / False
    """

    # rellenar
    pass

if __name__ == "__main__":
    if len(sys.argv) > 2:
        print('Syntax: python %s [filename]' % sys.argv[0])
        exit
    else:
        t = Translator()
        if len(sys.argv) == 2:
            t.translate_file(sys.argv[1])
        else:
            while True:
                sentence = input("ENGLISH: ")
                if len(sentence) < 2:
                    break
                print("PIG LATIN:", t.translate_sentence(sentence))

```