

# Prácticas de SAR

## 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](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” inicial), simplemente se agrega “yay” al final de la palabra.

# Ejemplos

---

## Ejemplos de traducción a “Pig Latin”

- mess  $\Rightarrow$  essmay
- father  $\Rightarrow$  atherfay
- Rwanda  $\Rightarrow$  Andarway
- choir  $\Rightarrow$  oirchay
- ant  $\Rightarrow$  antyay
- 4G  $\Rightarrow$  4G

# ¿Qué debo hacer?

---

## Escribe un programa en python que traduzca a Pig Latin.

- El programa DEBE llamarse `SAR_p1_piglatin.py`.
- El programa tendrá dos modos de funcionamiento:
  1. Si no recibe argumentos, entrará en un bucle pidiendo frases por teclado y traduciéndolas. El programa continuará hasta que introduzcamos una cadena vacía.
  2. Si le pasamos como argumento el nombre de un fichero, el programa creará otro fichero con la traducción:
    - El nombre del nuevo fichero se formará añadiendo el sufijo `_latin` al nombre del fichero original.
    - Se debe respetar la extensión del fichero original.

*Ejemplo: `test.txt` generaría `test_latin.txt`*

# ¿Qué debo hacer?

---

Se debe tener en cuenta:

- La traducción de la frase se hace palabra a palabra.
- Debe respetar unos determinados signos de puntuación:

".,;?!"

- Se deben respetar las mayúsculas en estos 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:

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

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

ENGLISH: brandy and a fried egg on top and Spam

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

ENGLISH: 4G and spam

PIG LATIN: 4G andyay amspay

ENGLISH:



# ¿Qué debo hacer?, Plantilla

---

```
import re
import sys
from typing import Text
from os.path import isfile

class Translator():

    def __init__(self, punt:Optional[Text]=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:
            punt = ". , ; ? ! "
        self.re = re.compile(r"(\w+)([" + punt + r"]*)")
```

# ¿Qué debo hacer?, Plantilla

---

```
def translate_word(self, word:Text) -> Text:
    """
    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
    """
    new_word = word # SUSTITUIR ESTA PARTE
    return new_word

def translate_sentence(self, sentence:Text) -> Text:
    """
    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
    """
    new_sentence = sentence # SUSTITUIR ESTA PARTE
    return new_sentence
```

# ¿Qué debo hacer?, Plantilla

---

```
def translate_file(self, filename:Text):
    """
    Recibe un fichero y crea otro con su traducción a Pig Latin

    :param filename: el nombre del fichero que se debe traducir
    :return: None
    """

    if not isfile(filename):
        print(f'{filename} no existe o no es un nombre de
fichero', file=sys.stderr)

    # COMPLETAR ESTA PARTE
```

# ¿Qué debo hacer?, Plantilla

---

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

# ¿Qué debo hacer?, Algunas pistas

---

```
>>> "4".isalpha()
False
>>> "4G"[0]
'4'
>>> "4G"[0].isalpha()
False
>>> "hola2"[0].isalpha()
True
>>> "Hola".isupper()
False
>>> "HOLA".isupper()
True
>>> "Hola"[0].isupper()
True
>>> "hola".upper()
'HOLA'
>>> "hola".capitalize()
'Hola'
```