Polish Python: A Short Report from a Short Experiment

Jakub Swacha
Department of IT in Management, University of Szczecin, Poland
jakub.swacha@usz.edu.pl

Abstract

Using a programming language based on English can pose an obstacle for learning programming, especially at its early stage, for students who do not understand English. In this paper, however, we report on an experiment in which higher-education students who have some knowledge of both Python and English were asked to solve programming exercises in a Polish-language-based version of Python. The results of the survey performed after the experiment indicate that even among the students who both know English and learned the original Python language, there is a group of students who appreciate the advantages of the translated version.

2012 ACM Subject Classification Social and professional topics → Computing education; Software and its engineering → General programming languages

Keywords and phrases programming language education, programming language localization, programming language translation, programming language vocabulary

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.25

1 Introduction

As a result of the overwhelming contribution of English-speaking researchers to the conception and development of computer science, almost every popular programming language used nowadays has a vocabulary based on this language [15]. This can be seen as an obstacle for learning programming, especially at its early stage, for students whose native language is not English [11]. In their case, the difficulty of understanding programs is augmented by the fact that keywords and standard library function names mean nothing to them. Even in the case of students who speak English as learned language, they are additionally burdened with translating the words to the language in which they think.

In order to let everyone write programs with the same level of ease as native English speakers can do, one needs to provide programming languages based on various natural languages. Although there were numerous attempts to develop new localized programming languages (see e.g. Algorithmi [9] or Phoenix [1]) and to localize existing languages (such as Logo as exemplified by various language kits for MSWLogo [8]), so far none of them gained notable popularity, maybe with the sole exception of Scratch [12], ranked 25th in the current TIOBE index [15].

In this paper, we investigate an attempt of translating one of the most popular programming languages of these days, Python, to the Polish language, and report the reaction of higher-education students who were asked to use it to solve just two programming exercises. As the experiment lasted short, and the survey answers obtained from the involved students were not many, the results presented here can only be interpreted to a limited degree, but they can certainly be seen as an indicator of the students' interest in non-English-based, translated programming languages, and possibly a motivator for future work on them.

The remainder of this paper is organized as follows. Section 2 gives a glimpse of the historical and still existing non-English programming languages with a focus on Polish-based languages. Next, the assumptions and the form of the experiment involving the students are

First International Computer Programming Education Conference (ICPEC 2020).
Editors: Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões; Article No. 25; pp. 25:1–25:6

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

described, along with the unsophisticated technique used to translate Python vocabulary to Polish language and the scope of translation. Then the survey results are presented and discussed. The final section concludes.

2 Related Work

The benefits of writing programs using a language based on the programmer's native language are especially valuable in the case of natural languages based on non-Latin script, where programmers can struggle even with deciphering characters they are not accustomed to. A number of programming languages were developed based on natural languages using such scripts – the examples from different computing epochs are Russian Rapira [7], developed in the 1980s for educational purposes, Chinese Eyuyan, first released in early 2000s, and still active [6], or Arabic Phoenix, released only as recently as 2019 [1].

While the benefit of using alphabet familiar to the programmer does not apply to the Polish language which is based on Latin script (with some additions), the argument of using words of a language known to the programmer remains valid. It was even more valid in the times of communist rule in Poland (1945-1989), when foreign language education in most primary schools was limited to the Russian language, and only a very small part of the population had any knowledge of English. Probably the earliest programming language based on Polish was SAKO, developed for the first Polish-built computers of the late 1950's and the early 1960's [17]. Over twenty years later, when the 8-bit microcomputers found their way into Polish homes, the Logo programming language has been translated, first for Atari XL/XE [16], then for Elwro 800 Junior, a Polish ZX Spectrum clone [3]. Contemporarily, Logomocja, a Polish version of Imagine Logo is still used for educational purposes [4].

In 2008, Rey, an educational programming language using Polish words was released. It was based on Java syntax and implemented many modern object-oriented-programming language features [2]. Unfortunately, Rey did not manage to reach a wide audience, which was, however, achieved by another educational language: Scratch, translated to Polish by Weronika Łabaj, Jan Baryła, Kris Kopera, Aleksandra Kopczynska, Marek Nowicki and Tomasz Ho-Janecki [10].

Probably the most interesting approach to the problem of programming language localization is the one introduced in Citrine, version 0.7, whose vocabulary is automatically translated between natural languages [5].

3 The experiment

3.1 Motivation

While the advantages of using a programming language similar to programmers' native language are easy to defend in the case of young pupils having no prior programming experience and limited or no knowledge of the base natural language, it is far from obvious if it could be considered valuable by higher-education students, who not only know the base natural language but also mastered the translated programming language at least at the pre-intermediate level. Note that a positive answer to this question makes it worthwhile to develop localized programming languages aiming at something more than introductory programming education, possibly even professional programming.

While a definite answer to this question is beyond the scope of this author's intended effort, as it would require a fully-fledged translation of a programming language and a long-term observation involving a significant number of students, a preliminary answer could be obtained in a very simple way as described below.

3.2 Execution

The experiment was performed at the end of winter semester of academic year 2019/2020. Two simple programming exercises were first prepared. The first exercise addressed the topics of text input/output and loop control and the second one –defining classes and methods. The exemplary solutions for the two exercises in Polish Python are presented in Listings 1 and 2.

Exercise 1. Write a program that asks the user to enter a password and checks whether it is `Mniam!`. If the password is wrong, the user is asked to correct it, but if he/she fails for the third time, the program should close.

Listing 1 Exemplary solution for Exercise 1.

```
dla _ w zakres(3):
    kod = wczytaj('Podaj tajny kod obiadowy: ')
    gdy kod == 'Mniam!':
        pisz ('To jest dobre hasło!')
        przerwij
    inaczej:
        pisz ('To nie jest dobre hasło!')
inaczej:
    pisz ('Wyczerpano dostępną liczbę prób.')
```

Exercise 2. Write a `Square` class which features one field (side length) and two methods (area and perimeter). Create an object of this class and call both its methods.

Listing 2 Exemplary solution for Exercise 2.

```
klasa Kwadrat:
    a = 1.0
    funkcja obwod(f):
        powrót f.a*4
    funkcja pole(f):
        powrót f.a**2
k = Kwadrat()
pisz (k.obwod(), k.pole())
```

Three groups counting together over 80 students were asked to read a short introduction to the translated Python language including a list of words translated to Polish (see Table 1), then solve the exercises using the translated version of the language. The list of translated words comprised all the Python keywords, most built-in functions as well as several module functions and methods that were chosen considering the typical solutions of the exercises used in the experiment. In order to edit and test their solutions, the students were given access to a modified version of this author's web-based interactive learning environment for Python described in [14]. The modification comprised in using a simple RegExp to replace all occurrences of the original Python words listed in Table 1 with their respective Polish translations.

Eventually, having successfully finished the exercises, they were asked to answer a survey consisting of just four questions:

1. General attitude to Python featuring Polish words.
2. The effect of translation on code readability.
3. The perceived value of translation for learning programming.
4. The willingness to write longer programs using the translated language.

The whole experiment took less than an hour.

Table 1 The scope of translation: Python words translated to Polish.

Original	Translated	Original	Translated	Original	Translated
add	dodaj	and	i	append	dostaw
as	jako	assert	sprawdź	await	oczekuj
break	przerwij	capitalize	dużą	ceil	sufit
center	centruj	choice	wybierz_losowo	chr	znak
class	klasa	clear	czyść	continue	kontynuuj
copy	kopiuj	count	licz	def	funkcja
degrees	stopnie	del	skasuj	dict	słownik
difference	różnica	discard	wyrzuć	elif	ale gdy
else	inaczej	except	wyjątek	extend	wydłuż
False	Fałsz	finally	finalnie	find	znajdź
float	dziesiętna	floor	podłoga	for	dla
from	z	frozen_set	stały_zbiór	global	globalne
if	gdy	in	w	index	pozycja
input	wczytaj	insert	wstaw	int	liczba
intersection	przecięcie	is	to	join	złącz
keys	klucze	len	długość	list	lista
lower	małe	math	matematyka	None	Nic
nonlocal	nielokalne	not	nie	or	lub
ord	kod	pass	pas	print	pisz
radians	radiany	raise	podnieś	randint	liczba_losowa
random	losowe	range	zakres	raw_input	wczytaj_tekst
remove	usuń	replace	podmień	return	powrót
reverse	odwróć	rfind	znajdź_od_tyłu	rjust	do_prawej
round	zaokrągl	set	zbiór	setdefault	ustaw_ domyślną
shuffle	pomieszaj	sort	sortuj	sorted	posortowana
split	podziel	sqrt	pierwiastek	str	napis
sum	suma	swapcase	zamień_litery	symmetric_ difference	różnica_ symetryczna
title	tytuł	True	Prawda	try	spróbuj
tuple	krotka	type	typ	union	połączenie
upper	duże	values	wartości	while	dopóki
with	używając	yield	zwróć	___init___	___inicjuj___
___str___	___napis___				

4 Survey results

The participation in the survey was not compulsory and some of the students were busy working on their assignments behind time. Altogether 48 survey responses were received (one omitting the answer to the first question).

For question 1, the allowed answers ranged from 1 (Nonsense) to 5 (Highly interesting). The distribution of students' answers to this question is presented in Fig. 1.

While about half of the surveyed students evaluated the idea negatively, there is about one-third of them who considered such an approach as interesting.

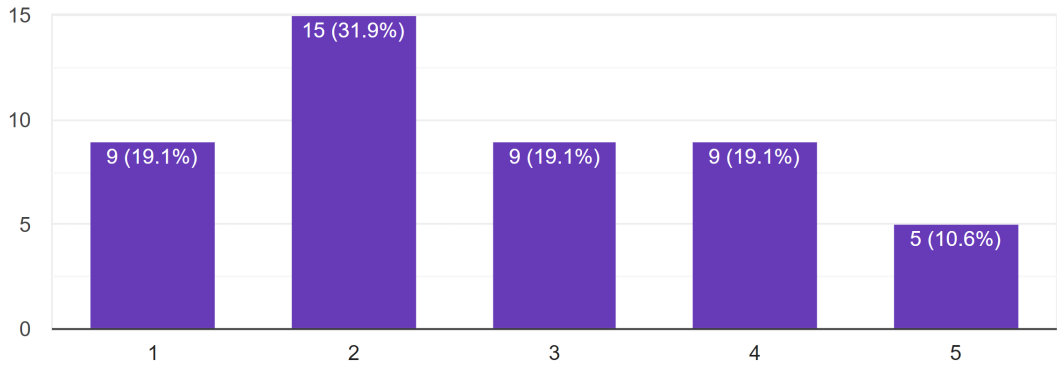

Figure 1 Overall evaluation of the idea of translating Python to Polish.

For question 2, about half of the students declared the effect of translation on code readability as negative, whereas only about 15% considered it positive. In the context that the students were accustomed to the standard, English-based version of Python, it is worth noting that there was a non-marginal group of students who appreciated the translation to their native language.

With regard to question 3, almost one-third of the students agreed with the benefits of translation for introductory programming learning. About one-sixth considered it as pointless because students should know English. According to half of the respondents the downside is that it would make it difficult to switch to the standard, English-based Python at the later stage of learning.

Finally, four-fifths of the students would not like to write any longer program using the translated Python, the remaining 20% of the surveyed students had opposite opinion.

5 Conclusion

The experiment described in this paper revealed that even among the higher-education students with prior experience with at least one English-based programming language, and some command of English, there is a group which finds the translated version of program more readable. This provides some motivation to future work on non-English-based, translated programming languages.

While the technical approach used in the experiment (using a simple RegExp) shows how easy it is to implement program translation in the world of browser-accessed interpreters, applying the translation to a stand-alone interpreter would require a more sophisticated approach, though there are ready-made solutions for doing it (see e.g. [13]).

One problem that would be difficult to tackle in real world is the scope of translation. Leaving popular libraries untranslated would leave the code look only partly-translated. On the other hand, in practice, it is impossible to translate them wholly, considering that they keep being developed and updated every day. Perhaps, the solution could be automatic translation, though it is doubtful whether its reliability matches the strict translation requirements necessary for the translated programs to run correctly.

References

1. Youssef Bassil. Phoenix – the Arabic object-oriented programming language. *International Journal of Computer Trends and Technology*, 67(2):7–11, 2019.
2. Paweł Baszuro and Jakub Swacha. Rey: An educational programming language. In R.T. Mittermeir and M.M. Sysło, editors, *Informatics education: Contributing across the curriculum*, pages 1–9, Toruń, 2008. WMiI UMK/PTI.
3. Wojciech Cellary and Krzysztof Pielesiak. *Leksykon Logo*. WNT, Warsaw, 1990.
4. Wojciech Czerski. Nowe sposoby nauki programowania w edukacji wczesnoszkolnej. *Dydaktyka Informatyki*, 13:129–134, 2018. doi:10.15584/di.2018.13.16.
5. Gabor de Mooij. Citrine, 2020. accessed on 20 Jan 2020. URL: http://citrine-lang.org.
6. DYWT.COM.CN. Eyuyan 5.9, 2019. accessed on 20 Jan 2020. URL: http://www.eyuyan.com/pdown.htm.
7. Laura Eileen Goodin. Teaching a nation to compute: The Rapira project and Soviet Information Technology education. Master's thesis, The American University, Washington, 1991.
8. Brian Harvey and George Mills. MSWLogo, an educational programming language, 2020. accessed on 20 Jan 2020. URL: http://www.softronix.com/logo.html.
9. António Manso, Luís Lopes, Célio Gonçalo Marques, Raquel Guedes, and Paulo Santos. Algorithmi: Bridging the algorithms to natural and programming languages, 2020. accessed on 20 Jan 2020. URL: http://algorithmi.ipt.pt/assets/papers/Final_Paperalgorithmi_2019_versao_EN.pdf.
10. MIT Scratch Team. Translators, 2020. accessed on 20 Jan 2020. URL: http://en.scratch-wiki.info/wiki/Translators.
11. Yizhou Qian and James D. Lehman. Correlates of success in introductory programming: A study with middle school students. *Journal of Education and Learning*, 5(2), 2016. doi:10.5539/jel.v5n2p73.
12. Mitchel Resnick. Reviving Papert's dream. *Educational Technology*, 52(4):42–46, 2012.
13. Stack Overflow. Can you add new statements to Python's syntax?, 2018. accessed on 20 Jan 2020. URL: http://stackoverflow.com/questions/214881/can-you-add-new-statements-to-pythons-syntax.
14. Jakub Swacha. Development and evaluation of an interactive Python course. In *ICERI2018 Proceedings*, pages 456–466, Seville, 2018. IATED.
15. TIOBE. TIOBE Index for January 2020, 2020. accessed on 20 Jan 2020. URL: http://www.tiobe.com/tiobe-index/.
16. Wojciech Zientara. Polskie Logo, 1986. accessed on 20 Jan 2020. URL: http://archive.org/details/a8b_Polskie_Logo_Wersja_Kasetowa_1986_W.Zientara_of_Bajtek_pl.
17. Leon Łukaszewicz and Antoni Mazurkiewicz. SAKO – an automatic coding system. *Annual Review in Automatic Programming*, 2:161–176, 1961.