

# **PROGRAMOWANIE ROBOTÓW NA EKRANIE KOMPUTERA – ZGODNE Z NOWĄ PODSTAWĄ PROGRAMOWĄ.**

*Adam Jurkiewicz  
Abix Edukacja  
edukacja@cyfrowaszkoła.waw.pl  
cyfrowaszkoła.waw.pl*

*Abstract. New PP defines that student must learn how to control robot or other "object" on screen or in the real world, using visual language and text based language. How to achieve that? I want to show OpenSource platform for doing that - Free of Charge and totally legal for every teacher and student.*

## **1. Wstęp**

W nowej PP jest zapis dotyczący klas 4-6: Uczeń:

- 1) projektuje, tworzy i zapisuje w wizualnym języku programowania: [...] prosty program sterujący robotem lub innym obiektem na ekranie komputera;

Dla klas 7-8 mamy zapis: Uczeń:

- 1) projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice. W szczególności programuje algorytmy z działu I pkt 2.
- 2) projektuje, tworzy i testuje oprogramowanie sterujące robotem lub innym obiektem na ekranie lub w rzeczywistości;

Zatem mamy tu "robota" lub inny obiekt, i język "wizualny" oraz instrukcje "wejścia/wyjścia i inne", które jednoznacznie wskazują na język tekstowy. I tu pojawia się oczywiste pytanie – jak szkoła, która nie dysponuje nieograniczonym, albo nawet i ograniczonym funduszem, ma sprostać temu wyzwaniu?

## 2. Moja propozycja bezpłatnego oprogramowania

Z pomocą przychodzi tu nam oprogramowanie Open Source<sup>1</sup>, które jest szeroko wykorzystywane w całym świecie. Open Source to obok ruchu na rzecz wolnego oprogramowania<sup>2</sup> (*free software movement*), ruch na rzecz otwartego oprogramowania "sterowany" przez luźną radę starszych, w skład której wchodzi Raymond, pozostali współzałożyciele oraz takie osobistości jak Linus Torvalds, Larry Wall i Guido van Rossum.

Oprogramowanie, aby można je było nazwać wolnym, musi spełniać kilka podstawowych założeń, zawartych w definicji wolnego oprogramowania opublikowanej przez Free Software Foundation. Przysługujące użytkownikowi wolności to:

- wolność uruchamiania programu, w dowolnym celu (wolność 0)
- wolność analizowania programu oraz dostosowywania go do swoich potrzeb (wolność 1)
- wolność rozpowszechniania kopii programu (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3).

Ja korzystając z ostatniej wolności, przygotowałem projekt Reeborg<sup>3</sup>, którego autorem jest Francuz, Andree Roberge<sup>4</sup>, i utrzymuję na moim serwerze dla społeczności nauczycieli informatyki w Polsce.

**<http://robotyka.cyfrowaszkoła.waw.pl>**

To system dostępny poprzez Internet i obsługiwany poprawnie przez Mozilla Firefox. Przede wszystkim mamy tu dwa tryby pracy (i możemy łatwo je zmieniać) – patrz rzuty poniżej:

- graficzny (bloczki) – bazujący na projekcie Blockly stworzonym przez Google – idealna dla klas 4-6 – <https://developers.google.com/blockly/>
- tekstowy (python) – bazujący na projekcie Brython, implementacji Python3 dla przeglądarek WWW - idealna dla klas 7-8 - <https://github.com/brython-dev/brython>

Świat Reeborg-a jest zaprojektowany do prostego przekazywania informacji zwrotnej gdy napotyka problem (np. Reeborg uderza o ścianę, gdy próbuje ruszyć lub próbuje podnieść przedmiot którego nie ma). Wiele różnych "światów/zadań"

<sup>1</sup> [https://pl.wikipedia.org/wiki/Otwarte\\_oprogramowanie](https://pl.wikipedia.org/wiki/Otwarte_oprogramowanie)

<sup>2</sup> [https://pl.wikipedia.org/wiki/Wolne\\_oprogramowanie](https://pl.wikipedia.org/wiki/Wolne_oprogramowanie)

<sup>3</sup> <https://github.com/aroberge/reeborg>

<sup>4</sup> <https://github.com/aroberge/reeborg/blob/master/LICENSE.md>

może być tak stworzonych, aby je ukończyć należy pokierować Reeborg-a by znalazł się na określonej pozycji lub przesunął przedmioty do oznaczonych miejsc.

Mixed mode: User interface in Polish; programming language in English.  
Tryb mieszany: interfejs użytkownika w języku polskim; język programowania w języku

Mixed mode: User interface in Polish; programming language in English.  
Tryb mieszany: interfejs użytkownika w języku polskim; język programowania w języku a

Najważniejszą cechą platformy jest to, że samodzielnie oraz automatycznie zmienia kod wizualny/blokowy na Python. Dodatkowo każdy może utworzyć własny świat, który zaprogramuje według swojego uznania i potrzeb. Jeśli tylko taki świat zapisze do pliku i wyśle do mnie mailem na adres: reeborg@abixedukacja.eu – ja zobowiązuję się umieścić ten świat na platformie tak, aby wszyscy mogli z niego korzystać. A jeśli ten opis spowodował, że chcesz teraz dokładnie dowiedzieć się o platformie - proponuję mój kanał na YouTube i samouczki o Python:

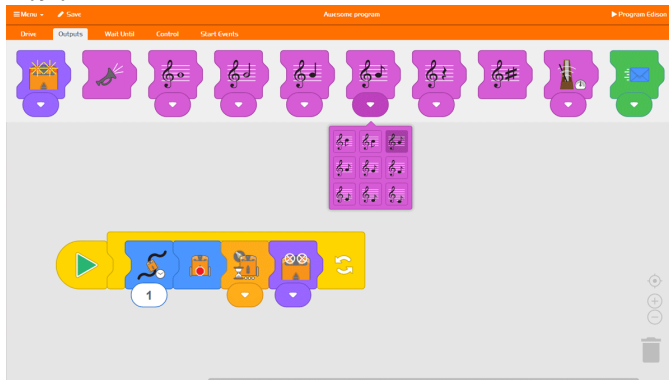
[http://bit.ly/abix\\_python101xEDU](http://bit.ly/abix_python101xEDU)

### 3. EDISON – robot programowany blokowo i tekstowo

Dodatkowo chciałbym przedstawić tu robota realnego, który ma bardzo ciekawą właściwość – może być programowany wizualnie i tekstowo, szkoła może więc używać jednego sprzętu zarówno z dziećmi małymi, jak i dużymi ;-)

<https://robotedison.pl>

Edison to projekt australijczyka, Brentona, który w 2014 roku wystartował na Kickstarterze<sup>5</sup>. Tam go poznałem, i od tego czasu uważam, że EDISON to świetny robot edukacyjny.

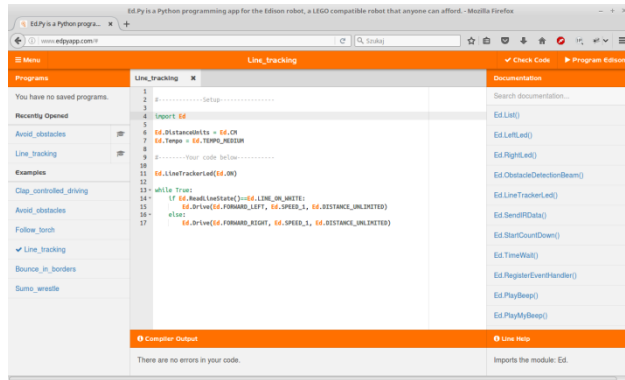


Rysunek 1 Dla klas 1-3 – platforma graficzna, tylko bloczki



Rysunek 2 dla klas 4-6 - platforma graficzna, bardziej skomplikowana

<sup>5</sup> <https://www.kickstarter.com/projects/937295081/edison-fun-robotics-for-tomorrows-inventors>



Rysunek 3 dla klas 7-8 i wyżej - Python tekstowo



Rysunek 4 dla wszystkich – możliwości współpracy z klockami LEGO (R) i rozbudowa robotów w różnych kierunkach

Internetowe środowisko programowania EdPy ma wiele funkcji wspomagających uczniów w nauce programowania tekstowego. Te funkcje obejmują:

- Linia pomocy – zapewnia wyjaśnienie w j. angielskim prawidłowych linii kodu
- Autouzupełnianie – automatycznie oferuje dopasowane propozycje opcji, gdy użytkownik zaczyna pisać kod
- Popup help – help box help zawierający opis kodu, który pojawia się podczas wpisywania (w języku angielskim)
- Pomoc tekst i przykłady – wbudowany tekst pomocy i przykłady natywne w przestrzeni programowania (w języku angielskim)

Te dwa przykłady, jeden programowy, zupełnie za darmo, oraz sprzętowy, który jednak należy zakupić, będą inspiracją do działania z młodzieżą. Do Was, nauczycieli, należy dobór rozwiązania. Każde jest jednakowo dobre, każde z powodzeniem może pomóc w nauce. To Wy wybieracie.