

# Opis Pythona+jupyter oraz Git+LaTeX

Karol Błędziński

3 grudnia 2018

## Literatura

- [1] Wikipedia.org      *Python*,      Społeczeństwo      Wikipedia,  
<https://pl.wikipedia.org/wiki/Python> 2002.

# 1 Pythona

## 1.1 Opis

Python jest interpretowanym, interaktywnym językiem programowania stworzonym przez Guido van Rossum w 1990 roku. Posiada w pełni dynamiczny system typów i automatyczne zarządzanie pamięcią, jest zatem podobny do takich języków, jak Tcl, Perl, Scheme czy Ruby. Python rozwijany jest jako projekt Open Source, zarządzany przez niedochodową Python Software Foundation.

## 1.2 Rozwój języka

Pythona stworzył we wczesnych latach 90. Guido van Rossum – jako następcę języka ABC, stworzonego w Centrum voor Wiskunde en Informatica (CWI – Centrum Matematyki i Informatyki w Amsterdamie). Van Rossum jest głównym twórcą Pythona, choć spory wkład w jego rozwój pochodzi od innych osób. Z racji kluczowej roli, jaką van Rossum pełni przy podejmowaniu ważnych decyzji projektowych, często określa się go przydomkiem „Benevolent Dictator for Life” (BDFL).

Nazwa języka nie pochodzi od zwierzęcia lecz od serialu komediowego emitowanego w latach siedemdziesiątych przez BBC – „Monty Python’s Flying Circus” (Latający cyrk Monty Pythona). Projektant, będąc fanem serialu i poszukując nazwy krótkiej, unikalnej i nieco tajemniczej, uznał tę za świetną.

Wersja 1.2 była ostatnią wydaną przez CWI. Od 1995 roku Van Rossum kontynuował pracę nad Pythonem w Corporation for National Research Initiatives (CNRI) w Reston w Wirginii, gdzie wydał kilka wersji Pythona, do 1.6 włącznie. W 2000 roku van Rossum i zespół pracujący nad rozwojem jądra Pythona przenieśli się do BeOpen.com by założyć zespół BeOpen PythonLabs. Pierwszą i jedyną wersją wydaną przez BeOpen.com był Python 2.0.

Po wydaniu wersji 1.6 i opuszczeniu CNRI przez van Rossum, który zajął się programowaniem komercyjnym, uznano za wysoce pożądane, by Pythona można było używać z oprogramowaniem dostępnym na licencji GPL. CNRI i Free Software Foundation (FSF) podjęły wspólny wysiłek w celu odpowiedniej modyfikacji licencji Pythona. Wersja 1.6.1 była zasadniczo identyczna z wersją 1.6, z wyjątkiem kilku drobnych poprawek oraz licencji, dzięki której późniejsze wersje mogły być zgodne z licencją GPL. Python 2.1 pochodzi zarówno od wersji 1.6.1, jak i 2.0.

Po wydaniu Pythona 2.0 przez BeOpen.com Guido van Rossum i inni programiści z PythonLabs przeszli do Digital Creations. Cała własność intelektualna dodana od tego momentu, począwszy od Pythona 2.1 (wraz z wersjami alpha i beta), jest własnością Python Software Foundation (PSF), niedochodowej organizacji wzorowanej na Apache Software Foundation.

### 1.3 Filozofia Pythona

Python realizuje jednocześnie kilka paradygmatów. Podobnie do C++, a w przeciwieństwie do Smalltalka nie wymusza jednego stylu programowania, pozwalając na stosowanie różnych. W Pythonie możliwe jest programowanie obiektowe, programowanie strukturalne i programowanie funkcyjne. Typy sprawdzane są dynamicznie, a do zarządzania pamięcią stosuje się garbage collection.

Choć w jego popularyzacji kładzie się nacisk na różnice w stosunku do Perla, Python jest pod wieloma względami do niego podobny. Jednakże projektanci Pythona odrzucili złożoną składnię Perla na rzecz bardziej oszczędnej i – ich zdaniem – bardziej czytelnej. Mimo że podobnie do Perla, Python jest czasem klasyfikowany jako język skryptowy, wykorzystuje się go do tworzenia dużych projektów jak serwer aplikacji Zope, system wymiany plików Mojo Nation czy nawet oprogramowanie klasy ERP – OpenERP.

## 2 jupyter

### 2.1 Opis

Jupyter to w pierwszej kolejności przeglądarkowe środowisko do pisania skryptów. Pierwotnym projektem na bazie którego powstał Jupyter jest IPYTHON, którego ostatnim wydaniem jako osobnego programu była wersja 3.0. Od wersji 4.0 IPython funkcjonuje już jako podstawowy kernel Jupytera, o czym przeczytamy dalej.

### 2.2 Markdown

Komórki notesowe mają swoje typy. Domyślnym typem jest oczywiście komórka na wpisanie kodu. Drugim ciekawym typem jest komórka interpretująca MARKDOWN. Dzięki temu w Jupyterze można ładnie komentować kod, jak także obiekty wynikowe.