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.

[2] Wikipedia.org *Jupyter - wprowadzenie*, perseba.github.io, http://perseba.github.io/blog/jupyter-wprowadzenie.html 2015.

[3] Wikibooks.org *LaTeX*, Społeczność Wikibooks, https://pl.wikibooks.org/wiki/LaTeX 2015.

1 Pythona

1.1 Opis

Python jest interpretowanym, interaktywnym językiem programowania stworzonym przez Guido van Rossuma 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 Fundation.

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 Rossuma, 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

Jupter 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.

3 GIT

3.1 Opis

Rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.

Pierwsza wersja narzędzia Git została wydana 7 kwietnia 2005 roku, by zastąpić poprzednio używany w rozwoju Linuksa, niebędący wolnym oprogramowaniem, system kontroli wersji BitKeeper.

3.2 Historia

Prace nad Gitem rozpoczęły się po tym, jak BitKeeper, używany wtedy do rozwoju Linuksa, przestał być darmowy dla projektów o otwartym kodzie źródłowym. Torvalds szukał rozproszonego systemu kontroli wersji, który mógłby być użyty zamiast BitKeepera, głównymi kryteriami wyboru były:Wziąć przykład z CVS, czego nie robić. System powinien być rozproszony. System powinien być chroniony przed błędami w repozytorium (przypadkowymi, jak awaria twardego dysku, jak i złośliwymi, wprowadzonymi przez kogoś). System powinien być szybki. Pierwsze dwa punkty wyeliminowały wszystko prócz Monotone'a, a czwarty punkt wyeliminował wszystko, więc Torvalds postanowił napisać własny system kontroli wersji. Prace nad Gitem rozpoczęły się 3 kwietnia 2005 roku, projekt został ogłoszony 6 kwietnia, 7 kwietnia Git obsługiwał kontrolę wersji swojego własnego kodu, 18 kwietnia pierwszy raz wykonano łączenie kilku gałęzi kodu, 27 kwietnia Git został przetestowany pod względem szybkości z wynikiem 6,7 łat na sekundę, a 16 czerwca Linux 2.6.12 był hostowany przez Gita.

3.3 Najważniejsze cechy

Dobre wsparcie dla rozgałęzionego procesu tworzenia oprogramowania: jest dostępnych kilka algorytmów łączenia zmian z dwóch gałęzi, a także możliwość dodawania własnych algorytmów.

Praca off-line: każdy programista posiada własną kopię repozytorium, do której może zapisywać zmiany bez połączenia z siecią; następnie zmiany mogą być wymieniane między lokalnymi repozytoriami.

Wsparcie dla istniejących protokołów sieciowych: dane można wymieniać przez HTTP(S), FTP, rsync, SSH.

Efektywna praca z dużymi projektami: system Git według zapewnień Torvaldsa, a także według testów fundacji Mozilla, jest o rzędy wielkości szybszy niż niektóre konkurencyjne rozwiązania.

Każda rewizja to obraz całego projektu: w przeciwieństwie do innych systemów kontroli wersji, Git nie zapamiętuje zmian między kolejnymi rewizjami, lecz kompletne obrazy. Z jednej strony wymaga to nieco więcej pracy aby porównać dwie rewizje, z drugiej jednak pozwala np. na automatyczną obsługę zmian nazw plików.

4 LaTeX

4.1 Opis

LaTeX to zestaw makr zaprojektowany przez Leslie Lamporta stanowiących nadbudowę nad systemem składu tekstu TeX, będącym kompletnym językiem programowania zaimplementowanym przez Donalda Knutha. Jego celem jest umożliwienie stworzenia profesjonalnie wyglądających oraz poprawnie złożonych dokumentów, szczególnie naukowych – zawierających matematyczne wzory – oraz dokumentów działających automatycznie (szablonowo).

LaTeX znacznie różni się od programów typu WYSIWYG (ang. What You See Is What You Get – widzisz to, co dostajesz, m.in. OpenOffice.org Writer lub Microsoft Word) i jego użycie przypomina bardziej programowanie. Do podstawowych trudności dla nowych użytkowników LaTeX-a należą:

składanie dokumentów może sprawiać trudności i zabierać czas niedoświadczonym użytkownikom nie widać bezpośrednio efektu zmian musisz poznać niezbędne polecenia w LaTeX-u czasami ciężko uzyskać pożądane efekty. Z drugiej strony taki język znaczników posiada także zalety:

dokument wygląda profesjonalnie, m.in. warstwy, fonty, tabele tworzą spójną całość zachęca do tworzenia poprawnych pod względem struktury dokumentów profesjonalnie złożone dokumenty charakteryzuje mała objętość w porównaniu do zawartości w łatwy sposób można tworzyć matematyczne wzory pozwala na zaawansowaną automatyzację składania dokumentów indeksy, odwołania i bibliografie mogą być automatycznie formatowane obsługuje praktycznie każdy komputer i system operacyjny identycznie składając dokumenty na każdym dokument źródłowy jest plikiem tekstowym, zatem jego wersjonowanie jest możliwe (porównywanie wersji, przypisywanie autora do konkretnej treści/zmian). Dokument LaTeX-a jest normalnym plikiem tekstowym, który przechowuje tekst z dodatkowymi znacznikami. Kiedy plik źródłowy zostanie przetworzony LaTeX-em otrzymamy plik DVI (ang. device-independent), który potem może zostać przekonwertowany do finalnego formatu np. PDF lub PostScript, które można łatwo wydrukować. Jest również możliwe bezpośrednie utworzenie pliku w formacie PDF.