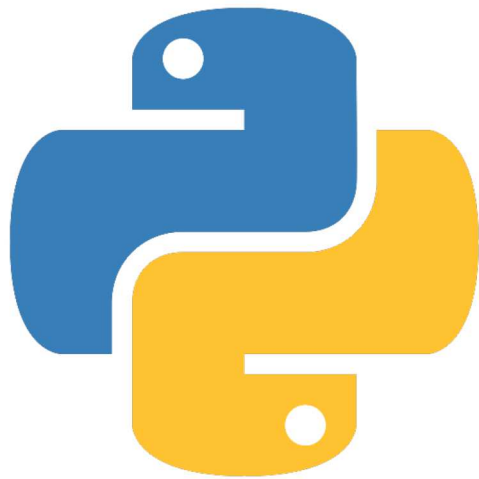


Podstawy Python 3:
Tabele informatyczne
Dominik Matusiak



Podstawy Python 3: Tabele informatyczne

Dominik Matusiak

d.matusiak@op.pl

Wydanie 3 (uzupełnione)

ISBN 978-83-964183-7-1

Wydawnictwo Eclipse, Zgierz, 2024 r.

Źródło ryciny na okładce: www.python.org/community/logos/



Wprowadzenie

Python jest wieloparadygmatowym, dynamicznie (nie deklaruje się typu danych dla zmiennej) i silnie typowanym, interpretowanym, obiektowym językiem programowania wysokiego poziomu i ogólnego przeznaczenia. Jest jednym z najpopularniejszych języków (obok C i Java). Jest rozwijany jako projekt open source (www.python.org). Jego składnia jest zwięzła i przejrzysta. Python dostępny jest na różne systemy operacyjne. Dostępnych jest wiele bibliotek rozszerzających jego funkcjonalność (np. na pypi.org – Python Package Index). Obok Pascala, Javy i C++ jest językiem programowania dopuszczonym do użytku na maturze z informatyki. Python standardowo wyposażony jest w proste, zintegrowane środowisko programistyczne (IDE – Integrated Development Environment) nazywane IDLE (Integrated Development and Learning Environment). Bogatym w funkcje środowiskiem programistycznym jest np. PyCharm, Visual Studio Code. Pliki źródłowe w Pythonie mają rozszerzenie „py”. Gotowe skrypty można łączyć w jeden plik wykonywalny za pomocą dodatków "py2exe", "PyInstaller" (np. „pyinstaller -F c:\skrypt.py”), "Auto PY to EXE" czy "cx_Freeze"; nie jest to jednak kompilacja, gdyż kod źródłowy nadal jest wykonywany przez interpreter. Dostępne są również interpretery w wersji online (np. www.programiz.com/python-programming/online-compiler/, www.online-python.com). Skrypty można zaciemnić, tj. utrudnić ich czytanie i zrozumienie bez zmiany funkcjonalności, za pomocą np. dodatku "pyarmor" (np. „pyarmor g c:\skrypt.py”), usług online (np. development-tools.net/python-obfuscator/). Nieoficjalnym kompilatorem Python jest program Nuitka (konwertujący Python do C++, a następnie do kodu maszynowego). Język Python został również zaimplementowany dla .NET (IronPython) i Java (Jython). W dalszej części zostaną omówione podstawowe cechy i komendy Pythona wraz z przykładami.

Wybrane zalety Python:

Prosta i zwięzła składnia, open source, duża dostępność bibliotek, przenośność (dostępność na różne platformy), różnorodność zastosowań (np. aplikacje webowe, analiza danych, uczenie maszynowe, aplikacje desktopowe), dopuszczony do użytku na maturze, możliwość pisania programów dla UEFI (Unified Extensible Firmware Interface).

Wybrane wady Python:

Brak oficjalnego kompilatora oraz wolniejsze działanie programów, brak natywnej obsługi tablic, brak komentarzy wieloliniowych, duże zużycie pamięci, niemutowalne zmienne tekstowe (ale alternatywnie można zastosować typ danych „bytearray”), dynamiczne typowanie utrudniające wykrywanie błędów, ubogie oficjalne IDE (IDLE), ograniczone zastosowanie dla aplikacji mobilnych (jednakże dostępne są np. frameworki Kivy, BeeWare; QPython, Pydroid, SL4A – Scripting Layer For Android, p4a – python for android, PyQt, Chaquopy), brak natywnej obsługi makr składniowych, nienajlepsze dokumentacje, mało zrozumiałe informacje o błędach, brak oficjalnego edytora wizualnego jak w Visual Basic (ale dostępny np. wxGlade, QT Designer, MD Python Designer), brak wsparcia nowszych wersji Python dla starszych systemów Windows, to samo polecenie nieco odmiennie może funkcjonować w różnych wersjach Python 3, brak oficjalnego wsparcia dla Android (w tym Android TV) i Tizen TV, brak oficjalnego wsparcia dla wstawek assemblerowych (ale dostępne np. biblioteki PeachPy, PyAsm).

Spis treści

| | |
|---|----|
| Wprowadzenie | 2 |
| Wybrane zalety Python | 3 |
| Wybrane wady Python | 3 |
| Interpreter | 3 |
| Przykładowe IDE i rozbudowane edytory tekstów | 4 |
| Komentarze | 4 |
| Operatory | 4 |
| Rozdzielanie komend i kodu | 5 |
| Zmienne | 5 |
| Wielokrotne przypisanie | 6 |
| Wyświetlanie danych w konsoli | 6 |
| Minimalny program | 6 |
| Pobieranie danych od użytkownika | 6 |
| Funkcje | 7 |
| Instrukcje warunkowe | 7 |
| Importowanie bibliotek | 8 |
| Instrukcje iteracyjne (pętle) | 10 |
| Przerywanie pętli/programu | 10 |
| Pauza i timer | 10 |
| Czyszczenie ekranu | 11 |
| Podstawowe funkcje matematyczne | 11 |
| Generowanie liczb pseudolosowych | 11 |
| Manipulowanie zmiennymi tekstowymi | 12 |
| Praca na listach | 13 |
| Wykorzystanie słowników | 14 |
| Wykorzystanie funkcjonalności klawiatury | 15 |
| Grafika żółwia | 15 |
| Biblioteka graficzna PIL | 17 |
| Dźwięki | 18 |
| Biblioteka Colorama | 18 |
| Obsługa plików | 19 |
| Obsługa błędów | 20 |
| Uruchomienie innego skryptu | 21 |
| Wczytanie parametru z linii poleceń | 21 |
| Funkcje statystyczne | 21 |
| Biblioteka Tkinter | 22 |
| OOP (object-oriented programming) | 23 |
| Biblioteka Numpy – deklarowanie tablicy | 24 |
| Tryb tekstowy – biblioteka BearLibTerminal | 24 |
| Jak przyspieszyć działanie skryptów Python? | 25 |
| Typ danych „bytearray” | 26 |
| Biblioteka Pygame | 26 |
| Polecenie „eval” | 28 |
| Biblioteka PySimpleGUI | 29 |
| Biblioteka Matplotlib | 31 |
| Polecenie „match” | 32 |
| Przykładowe różnice między Python 2 i 3 | 32 |
| Przykładowe programy | 32 |
| Literatura uzupełniająca | 39 |
| Przydatne linki | 40 |
| Tabela kodów ASCII 0-127 | 41 |