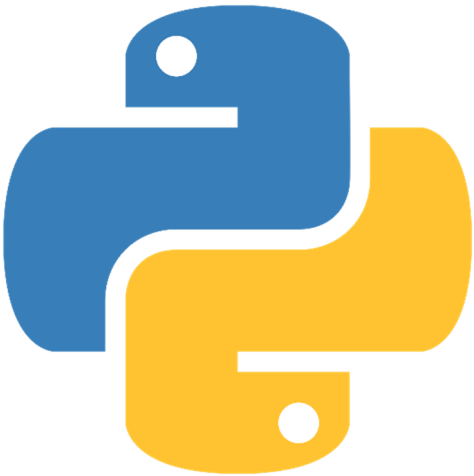
Podstawy Python 3:

Tabele informatyczne

Dominik Matusiak



Podstawy Python 3: Tabele informatyczne

dr Dominik Matusiak

d.matusiak@op.pl

Wydanie 4 (uaktualnione)
ISBN 978-83-964183-9-5
Wydawnictwo Eclipse, Zgierz, 2026 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), zastosowanie w elektronice i robotyce (np. Microbit i PyBoard oparte o MicroPython, CircuitPython – odmiana poprzedniego, Pybricks – Lego Mindstorms i in.).

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 z Buildozer, BeeWare; QPython, Pydroid, SL4A – Scripting Layer For Android, p4a – python for android, PyQt, Chaquopy, Flet, pyqtdeploy), brak natywnej obsługi makr składniowych, nienajlepsze dokumentacje, mało zrozumiałe informacje o błędach, brak oficjalnego edytora wizualnego jak w Visual Basic (ale dostępny np. Python Tkinter Designer, KivyMD Builder, wxGlade, PyQT Designer, PySide6, MD Python Designer, PyUiBuilder, P4D – Python for Delphi, PAGE – Python Automatic GUI Generator, Gluonix Designer, B4X + PyBridge, Pygubu Designer, Anvil – aplikacje online), 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 asemblerowych (ale dostępne np. biblioteki PeachPy, PyAsm), wiele bibliotek jest niedopracowanych, nierozwijanych (porzucone projekty), słabo udokumentowanych.

Interpreter

W IDLE możliwe jest bezpośrednie wpisywanie i wykonywanie poleceń, jak również tworzenie nowych skryptów lub uruchamianie już napisanych.

F5 − uruchomienie edytowanego skryptu
control + c − zatrzymanie działania skryptu
help (print) # wyświetlenie pomocy na temat polecenia "print"
Skrypty można również uruchamiać z linii poleceń:
python c:\skrypt.py

Przykładowe IDE i rozbudowane edytory tekstów

Pisanie programów w Python ułatwiają m.in. następujące programy: IDLE (dołączony do interpretera), PyCharm (dużo funkcji), Spyder, Visual Studio Code, Atom, Sublime Text, Thonny, Eclipse + PyDev, Visual Studio + PTVS (Python Tools for Visual Studio), Ninja-IDE, Notepad++, Wing, Geany.
Jupyter Notebook – interfejs webowy dla Python, w którym można pisać, wizualizować i wykonywać kod (podzielony na komórki).

Komentarze

```
Print ("tekst") # komentarz
# Komentarz2
'''
Brak komentarzy wieloliniowych w Python. Jako substytut można zastosować trzy apostrofy.
'''
```

Spis treści

Wprowadzenie	2
Wybrane zalety Python	3
Wybrane wady Python	3
Interpreter	4
Przykładowe IDE i rozbudowane edytory tekstów	4
Komentarze	4
Operatory	5
Rozdzielanie komend i kodu	6
Zmienne	6
Wielokrotne przypisanie	6
Wyświetlanie danych w konsoli	7
Minimalny program	7
Pobieranie danych od użytkownika	7
Funkcje	8
Instrukcje warunkowe	9
Importowanie bibliotek	10
Instrukcje iteracyjne (pętle)	12
Przerywanie pętli/programu	12
Pauza i timer	13
Czyszczenie ekranu	13
Podstawowe funkcje matematyczne	13
Generowanie liczb pseudolosowych	14
Manipulowanie zmiennymi tekstowymi	14
Praca na listach	15
Wykorzystanie słowników	16
Wykorzystanie funkcjonalności klawiatury	17
Grafika żółwia	17
Biblioteka graficzna PIL	20
Dźwięki	20
Biblioteka Colorama	21
Obsługa plików	21
Obsługa błędów	23
Wczytanie parametru z linii poleceń	23
Funkcje statystyczne	23
Biblioteka Tkinter	24

OOP (Object-Oriented Programming) 25
Biblioteka Numpy – deklarowanie tablicy 26
Tryb tekstowy – biblioteka BearLibTerminal 26
Jak przyspieszyć działanie skryptów Python? 27
Typ danych „bytearray” 27
Biblioteka Pygame 28
Polecenie „eval” i „exec” 32
Biblioteka PySimpleGUI 32
Biblioteka Matplotlib 33
Polecenie „match” 35
Przykładowe różnice między Python 2 i 3 35
Double-Ended Queue 36
Biblioteka Openpyxl 36
Pyinstaller 37
Biblioteka Itertools 38
Przetwarzanie równoległe 38
Aplikacje mobilne i nie tylko – Beeware 39
Biblioteki decimal i fractions 42
Biblioteka pandas 42
Pyarmor 43
CadQuery – projektowanie modeli 3D 45
Mypy 46
Nuitka 47
Przykładowe programy 48
Literatura uzupełniająca 60
Przydatne linki 61
Inne dzieła autora 61
Tabela kodów ASCII 0-127 62

Notatki

Wydanie 4
ISBN 978-83-964183-9-5
Wydawnictwo Eclipse, Zgierz, 2026 r.

