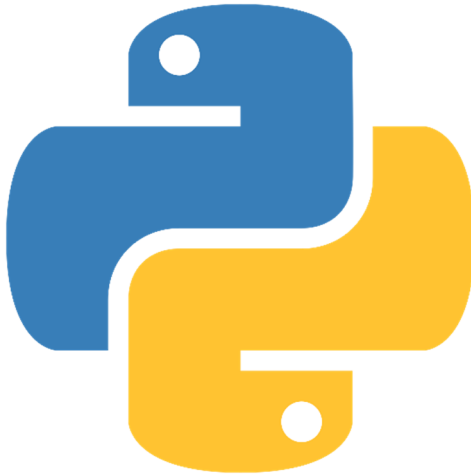


# **Podstawy Python 3:**

## **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/](http://www.python.org/community/logos/)

ISBN 978-83-964183-7-1



9 788396 418371

## 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](http://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](http://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/](http://www.programiz.com/python-programming/online-compiler/), [www.online-python.com](http://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/](http://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, 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, Gluonix Designer, PyUiBuilder), 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).

## 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.

## Komentarze

```
Print ("tekst") # komentarz
```

```
# Komentarz2
```

```
"""
```

Brak komentarzy wieloliniowych w Python. Jako substytut można zastosować trzy apostrofy.

```
"""
```

## Operatory

*Arytmetyczne:*

+      dodawanie

-      odejmowanie

\*      mnożenie

/      dzielenie

//     dzielenie całkowite (odrzućenie części ułamkowej wyniku)

%      reszta z dzielenia (modulo)

\*\*     potęgowanie (można również wykorzystać do wyciągania pierwiastka, gdyż:  $n\sqrt{a} = a^{1/n}$ )

## Przypisania:

```
a = (2 * 5) + 3
```

```
zmienna_tekst = "tekst" # Python rozróżnia wielkość liter.
```

```
# To samo co: zmienna_tekst = 'tekst'
```

```
a += 5 # to samo co: a = a + 5; podobnie działają: -=, *=, /=, //, **=
```

## Relacji:

```
==      równy
```

```
!=      różny
```

```
<       mniejszy
```

```
>       większy
```

```
<=      mniejszy bądź równy
```

```
>=      większy bądź równy
```

## Operacji na bitach (logiczne):

```
&       koniunkcja
```

```
|       alternatywa
```

```
^       alternatywa rozłączna (XOR – exclusive or, można stosować do prostego  
szyfrowania danych)
```

## Rozdzielanie komend i kodu

```
liczba = 1; print (liczba)
```

```
# Komendy można zapisywać w jednej linii, jeśli rozdzieli się je średnikiem.
```

```
print ("tekst" + \
```

```
"tekst2") # Kod można rozdzielić na więcej linii, jeśli zastosuje się lewy ukośnik.
```

## Zmienne

Python obsługuje m.in. zmienne całkowite (integer), zmiennie-przecinkowe (float), tekstowe / łańcuchowe (string), listy, słowniki, krotki.

```
liczba = 0 # całkowita
```

```
liczba2 = 1_000_000 # to samo co: liczba2 = 1000000
```

```
liczba3 = 0.0 # zmiennoprzecinkowa
```

```
liczba = "0b111" # zapis liczby binarnej
```

```
liczba2 = "0xFF" # zapis liczby w systemie szesnastkowym
```

```
tekst = "tekst"
```

```
# to samo co tekst: tekst = 'tekst', tego typu zmienne nie są modyfikowalne  
# (mutowalne)
```