



Politechnika Wrocławska

Programowanie proceduralne Laboratorium #01 – Fizyka



HR EXCELLENCE IN RESEARCH

Janusz Andrzejewski



Wstęp – „Programowanie proceduralne”

- Kontakt.
- Warunki zaliczenia.
- Programowanie – dlaczego i co to jest?
- Dlaczego Python?
- Python:
 - Historia.
 - Cechy.
 - Filozofia.
- Czego będziemy się uczyli?
- Cel zajęć.
- Literatura i inne pomoce w nauce.
- Co i jak zainstalować.
- Podsumowanie oraz zadania.

Kontakt:

- janusz.andrzejewski@pwr.edu.pl
 - Pokój 319, budynek A1
- Wszystkie materiały będą dostępne na e-portalu PWr

Kiedy:

- Wykłady: CZ 11.15, sala 0.32 C13 - tygodnie parzyste
- Laboratoria: WT 7.30; PT 7.30 & 13.15 - wszystkie w sali 249 A1

	LUTY	MARZEC				KWIECIEŃ				MAJ					CZERWIEC				LIPIEC	
PN	27	6 Pn N	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26 Cz P	3	10
WT	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27 Pt P	4	11
ŚR	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21 Pn P	28	5	12
CZ	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13
PT	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14
SO	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15
N	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16
P - PARZYSTY N - NIEPARZYSTY	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P



Zaliczania:

- Obecność obowiązkowa na zajęciach.
- Będą 3 sprawdziany:
 - Na 5 zajęciach - wykłady W#1 oraz W#2
 - Na 10 zajęciach - wykłady W#3, W#4 oraz W#5
 - Na 15 zajęciach - albo 3 sprawdzian (wykłady W#6 oraz W#7) albo kolokwium poprawkowe z całości materiału
- Każdy sprawdzian będzie za 10 punktów.
- Na sprawdzianach można korzystać z własnych notatek czy prezentacji.
- Na ostatnim laboratorium jest możliwość wyboru:
 - Można pisać tylko 3 sprawdzian (materiał tylko z ostatnich 4 zajęć)
 - Można pisać tylko sprawdzian poprawkowy – obowiązuje materiał ze wszystkich zajęć, ocena ze sprawdzianu jest oceną z laboratorium, można poprawić maksymalnie na dst+

Punkty	Ocena
30+	Cel (5.5)
27-29	bdb (5.0)
24-26	db+ (4.5)
21-23	db (4.0)
19-21	dst+ (3.5)
16-18	dst (3.0)
0 - 15	ndst (2.0)

Dlaczego programowanie

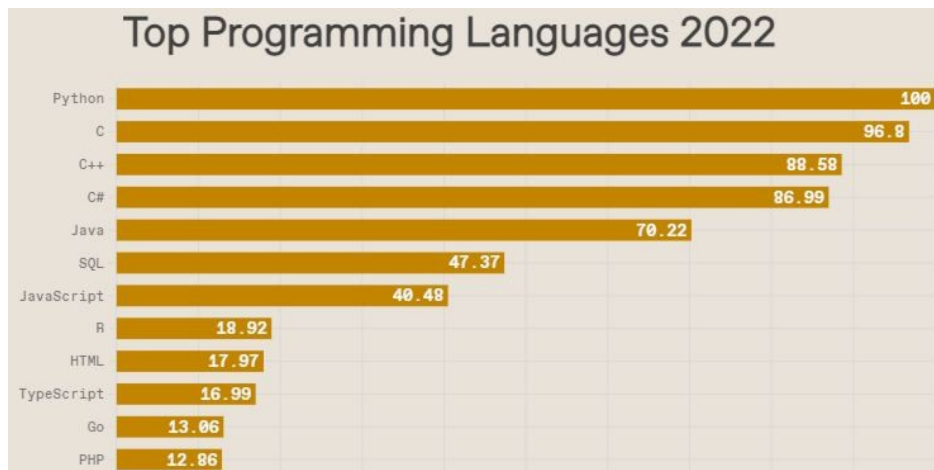
- Programowanie uczy myślenia.
- Programowanie rozwija kreatywność.
- Programowanie uczy posługiwania się zasadami.
- Programowanie to umiejętność, na którą rośnie zapotrzebowanie.
- Programowanie jest interdyscyplinarne — wykorzystuje zarówno wiedzę ścisłą jak i humanistyczną.
- Umiejętność programowania staje niezbędną dodatkową wiedzą oprócz podstawowego wykształcenia inżyniera.

Co to jest programowanie?

- 1) **Programowanie** jest to proces tworzenia programu komputerowego posługując się kodem źródłowym. Najprościej mówiąc programowanie polega na opracowaniu poleceń dla komputera –
https://www.naukowiec.org/wiedza/informatyka/programowanie-czy-m-jest_3656.html
- 2) **Programowanie** – mówienie(rozkazywanie) w sposób zrozumiały komputerowi dokładnie i precyzyjnie tego co ma robić
 - Co znaczy że umiem programować
 - 1) Znam (tzn. umiem poszukać lub wymyślić) i rozumiem algorytm – sposób rozwiązania problemu
 - 2) Znam jakiś język programowania
 - 3) Umiem zaimplementować algorytm wykorzystując cechy wybranego języka programowania
 - 4) Umiem uruchomić (skompilować, usunąć błędy składniowe) napisany program
 - 5) **Umiem przetestować, usunąć błędy semantyczne(znaczeniowe) (tzw. bugi) i sprawdzić czy program działa poprawnie**

Dlaczego Python?

- <https://spectrum.ieee.org/top-programming-languages-2022/ieee-spectrums-top-programming-languages-2022>



- <https://www.tiobe.com/tiobe-index/>

Programming Language	2022	2017	2012	2007	2002	1997	1992	1987
Python	1	5	8	7	12	28	-	-
C	2	2	1	2	2	1	1	1
Java	3	1	2	1	1	16	-	-
C++	4	3	3	3	3	2	2	6
C#	5	4	4	8	14	-	-	-



- Nazwa języka pochodzi od serialu BBC Latający Cyrk Monty Pythona.
- Twórcą jest holender **Guido van Rossum** (ur. 31 stycznia 1956 w Haarlem)

Historia

- Python (jego interpreter) zaczął powstawać od grudnia 1989 jako następca języka ABC w
- „Wewnętrzna” wersja powstawała w Centrum Wiskunde & Informatica (CWI) (Centrum Matematyki i Informatyki) w Amsterdamie — 1990
- Ostatnia wersja Pythona, która powstała w CWI to 1.2 (1995).
- Od wersji 2.1 (2001) język jest rozwijany jako projekt Open Source przez niedochodową organizację Python Software Foundation (PSF), wzorowaną na Apache Software Foundation (ASF).

Wersje Pythona: & python™

- **Python 0.9.0** – 20 luty 1991 - Guido van Rossum opublikował kod źródłowy interpretera Pythona w *alt.source*, grupie *Usenetu* zajmującej się kodem open-source
 - **Python 1.0** – 26 stycznia 1994
 - **Python 2.0** – 16 października 2000
 - Python 2.7 – 3 lipca 2010 (Koniec wsparcia 2020-01-01)
 - **Python 3.0** – 3 grudnia 2008
 - Python 3.6 – 23 grudzień 2016
 - Python 3.7 – 27 czerwca 2018 (Koniec wsparcia 2023-06-27)
 - Python 3.8 – 14 października 2019 (Koniec wsparcia 2024-10)
 - Python 3.9 – 5 października 2020 (Koniec wsparcia 2025-10)
 - Python 3.10 – 4 października 2021 (Koniec wsparcia 2026-10)
 - Python 3.11 – 24 października 2022 (Koniec wsparcia 2027-10)
 - Obecnie najnowszą wersją jest **3.11.2** – 8 luty 2023
- (Zobacz też https://en.wikipedia.org/wiki/History_of_Python)

Cechy Pythona

- Język skryptowy (interpretowany) wysokiego poziomu, ogólnego przeznaczenia, zorientowany obiektowo, dostępny za darmo.
- Prosta i czytelna składnia (poprzez wcięcia) ułatwiająca utrzymywanie, używanie i rozumienie kodu – „łatwe” czytanie kodu.
- Wbudowane wsparcie wielu praktycznych i złożonych struktur danych („nieograniczone” liczby całkowite, liczby zespolone, krotki, listy, zbiory, słowniki, napisy, ...).
- Silny dynamiczny system typów.
- Automatyczne zarządzanie pamięcią,
- Wsparcie różnych paradygmatów programowania (obektowy, strukturalny, częściowo funkcyjny),
- Rozbudowana i bogata biblioteka standardowa,
- Elastyczny i rozszerzalny,
- Bogaty w biblioteki (NumPy, SciPy, Matplotlib, ...).
- Bardzo dobra i łatwo dostępna dokumentacja

Wady Pythona

- Nie jest najszybszym z języków
- Nie ma największej ilości bibliotek
- Nie sprawdza typów zmiennych (dla jednych to wada, dla innych to zaleta)
- Słabo wspiera urządzenia mobilne
- Słabo wykorzystuje wiele procesorów naraz



Filozofia — zen of Python

- Piękne jest lepsze niż brzydkie.
- Wyrażone wprost jest lepsze niż domniemane.
- Proste jest lepsze niż złożone.
- Złożone jest lepsze niż skomplikowane.
- Płaskie jest lepsze niż wielopoziomowe.
- Rzadkie jest lepsze niż gęste.
- Czytelność się liczy.
- Sytuacje wyjątkowe nie są na tyle wyjątkowe, aby łamać reguły.
- Choć praktyczność przeważa nad konsekwencją.
- Błędy zawsze powinny być sygnalizowane.
- Chyba że zostaną celowo ukryte.
- W razie niejasności powstrzymaj pokusę zgadywania.
- Powinien być jeden — i najlepiej tylko jeden — oczywisty sposób na zrobienie danej rzeczy.
- Choć ten sposób może nie być oczywisty jeśli nie jest się Holendrem.
- Teraz jest lepsze niż nigdy.
- Chociaż nigdy jest często lepsze niż natychmiast.
- Jeśli rozwiązanie jest trudno wyjaśnić, to jest ono złym pomysłem.
- Jeśli rozwiązanie jest łatwo wyjaśnić, to może ono być dobrym pomysłem.
- Przestrzenie nazw to jeden z niesamowicie genialnych pomysłów — miejmy ich więcej



Filozofia — zalecenia

- Każdy język programowania ma swój „duch”, styl w jaki sposób powinniśmy tworzyć programy:
 - Struktura
 - Składnia
 - Typy danych
 - Wbudowane biblioteki
- Ucząc się Pythona, staraj się zrozumieć duch i filozofię tego języka
- Bądź po prostu **PYTHONISTA**



II semestr - co już każdy student powinien znać i czym umieć się posługiwać

- Posługiwanie się:
 - Liczbami
 - Ciągami (łańcuchami) znaków
 - Krotkami
 - Listami
 - Plikami
- Znajomość konstrukcji:
 - Warunkowych
 - Obliczeń cyklicznych
 - Funkcji na poziomie znanym z „matematyki”



Plan czyli czego się będziemy uczyli?

- W#1 - Podsumowanie
- - Typy mutowalne oraz niemutowalne a przekazywanie argumentów do funkcji.
- - Zasięg zmiennych - zmienne lokalne, nielokalne oraz globalne.
- - Argumenty funkcji: pozycyjne, nazwane, domyślne, rozpakowywanie, słownikowe
- - Funkcje: „first citizen object”.
- - Funkcje: funkcje wyższego rzędu.
- - Funkcje anonimowe.
- - Dopełnienia funkcji.
- - Zastosowania funkcji.
- Laboratoria: od teorii do praktyki czyli rozwiązywanie problemów związanych z omówionymi zagadnieniami na wykładzie.

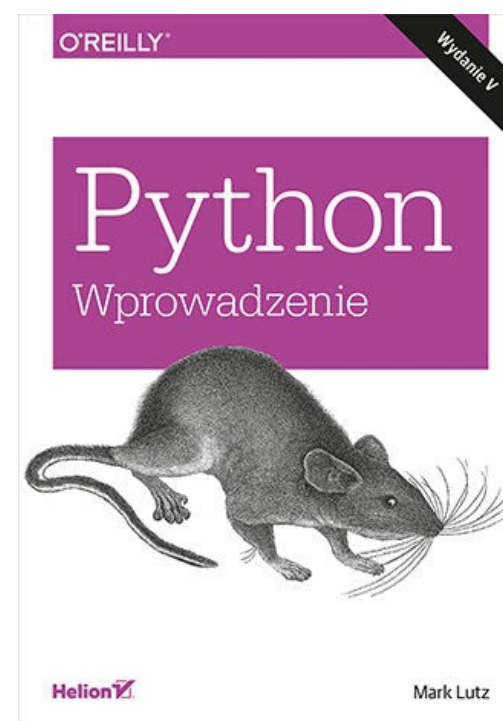
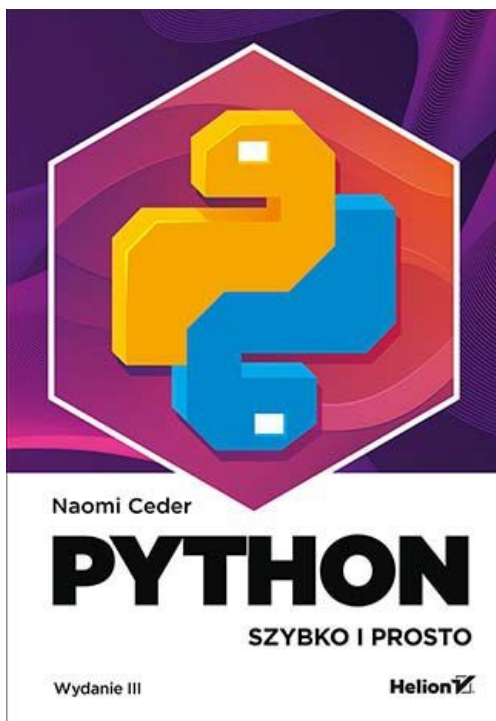
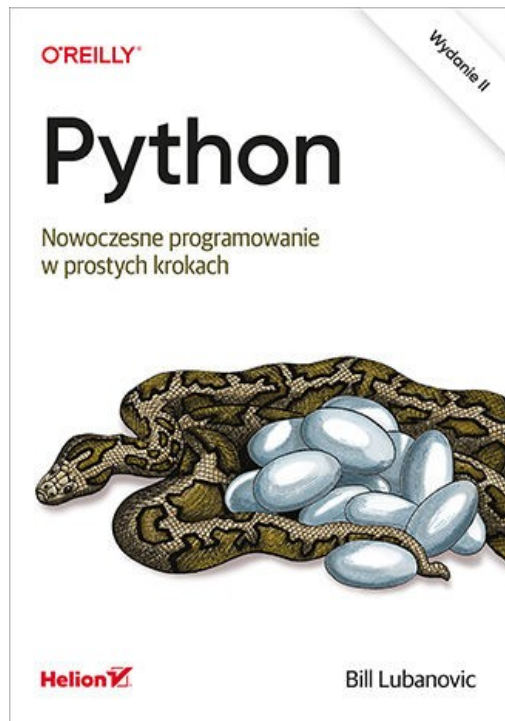
Cel zajęć:

- Ugruntowanie podstawowej znajomości języka programowania:
 - Sposoby zapamiętywania danych: liczby, listy, krotki, łańcuchy
 - Sposoby przetwarzania (obróbki) danych: pętle, instrukcje warunkowe, funkcje, instrukcje we/wy
- Rozszerzenie podstawowej znajomości języka programowania o:
 - Funkcje
 - Struktury danych: słowniki, zbiory, pliki
- Umiejętność znalezienia lub wymyślenia, zrozumienia i zastosowania (implementacji) odpowiedniego algorytmu
- Umiejętność usuwania błędów składniowych oraz znaczeniowych (semantycznych) —

[https://pl.wikipedia.org/wiki/B%C5%82%C4%85d_\(informatyka\)](https://pl.wikipedia.org/wiki/B%C5%82%C4%85d_(informatyka))



Literatura





Inne pomoce naukowe

- www.python.org
 - <https://www.python.org/doc/> - podręcznik języka („manual”)
 - <https://wiki.python.org/moin/>
- <https://learning-python.com/>
- <https://www.learnpython.org>
- <https://python-course.eu/python-tutorial/>
- <https://python.swaroopch.com/>
- <http://ricardoduarte.github.io/python-for-developers/>
- https://pl.wikibooks.org/wiki/Zanurkuj_w_Pythonie

Dlaczego IDE

Można zainstalować „gołego” Pythona ze strony <https://www.python.org>, ale:

IDE – to zintegrowane środowisko programistyczne, które:

- Pozwala na podświetlenie składni
- Pozwala na łatwe zarządzania projektami czy też samymi plikami
- Ułatwia dostęp do dokumentacji
- Pozwala na łatwiejsze usuwanie błędów
- Wymaga na początku opanowania - dodatkowa nauka
- Jest wiele różnych środowisk
 - spyder - <https://www.spyder-ide.org>
 - vscode - <https://code.visualstudio.com>
 - pycharm - <https://www.jetbrains.com/pycharm>
 - jupyter - <https://jupyter.org>



Dystrybucja Pythona — anaconda.com

The screenshot shows the Anaconda website homepage. At the top is a navigation bar with the Anaconda logo, links for Products, Pricing, Solutions, Resources, Partners, Blog, and Company, and a Contact Sales button. Below this is a banner for the 'STATE OF DATA SCIENCE 2022' report with an 'Access the Report' button. The main heading reads 'Data science technology for a better world.' followed by a paragraph about Anaconda's ease of use for Python/R data science and machine learning. A large green 'Download' button with a Windows icon is prominent, with a link to 'Wyślij nam swój ekran' (Send us your screen) above it. Below the button, it specifies 'For Windows', 'Python 3.9 • 64-Bit Graphical Installer • 594 MB', and a link to 'Get Additional Installers'. At the bottom are icons for Windows, macOS, and Linux.

← → ↻ 🔒 anaconda.com

ANACONDA Products ▾ Pricing Solutions ▾ Resources ▾ Partners ▾ Blog Company ▾ [Contact Sales](#)

STATE OF
DATA SCIENCE 2022 [Access the Report](#)

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

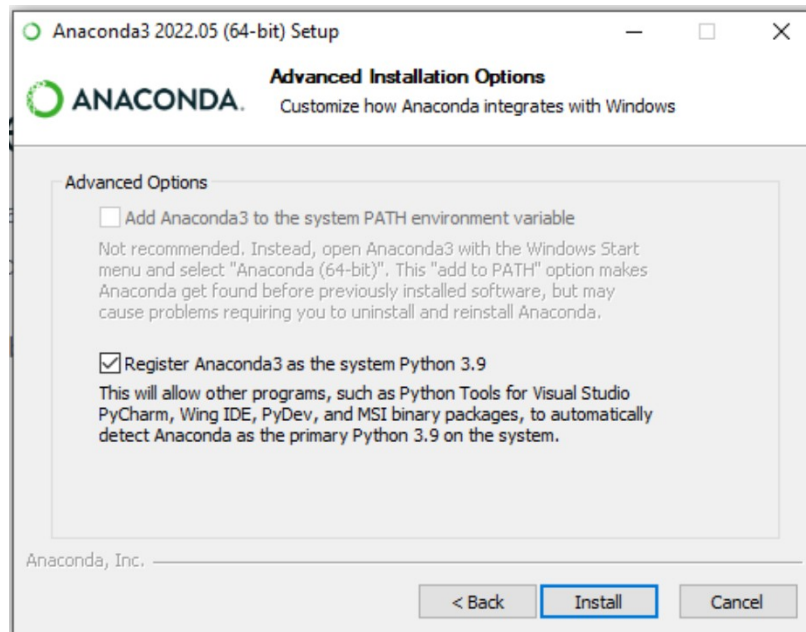
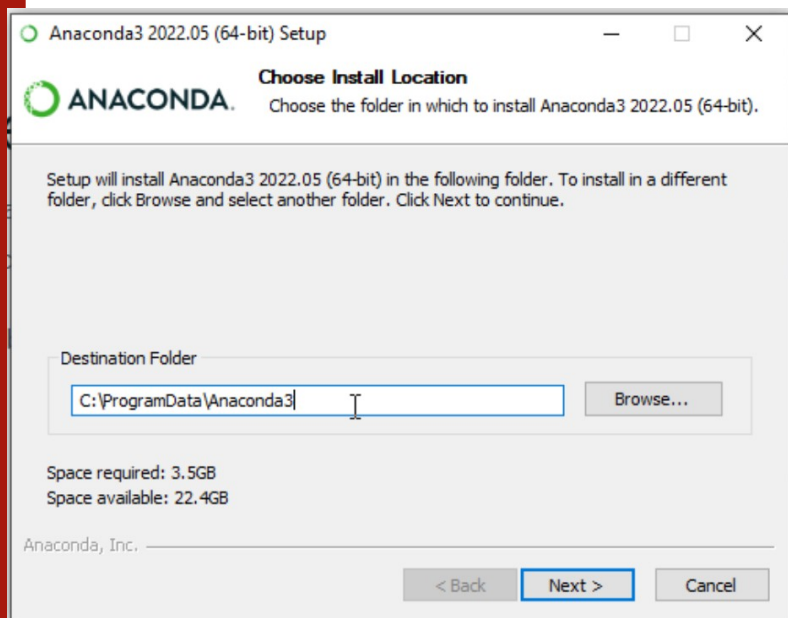
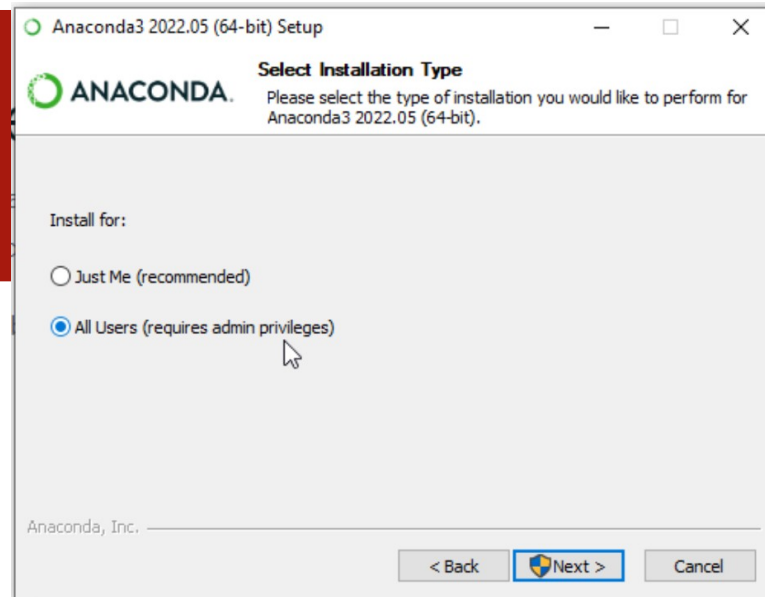
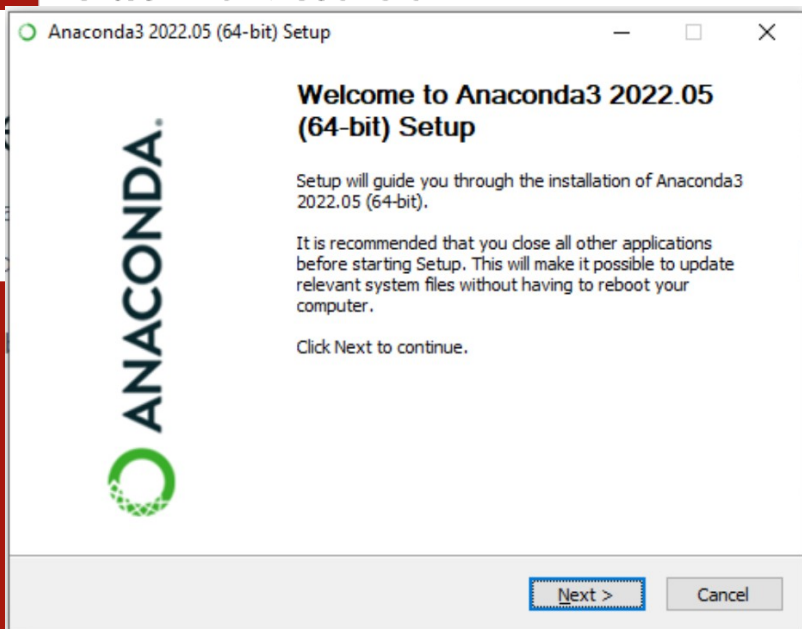
[Wyślij nam swój ekran](#)

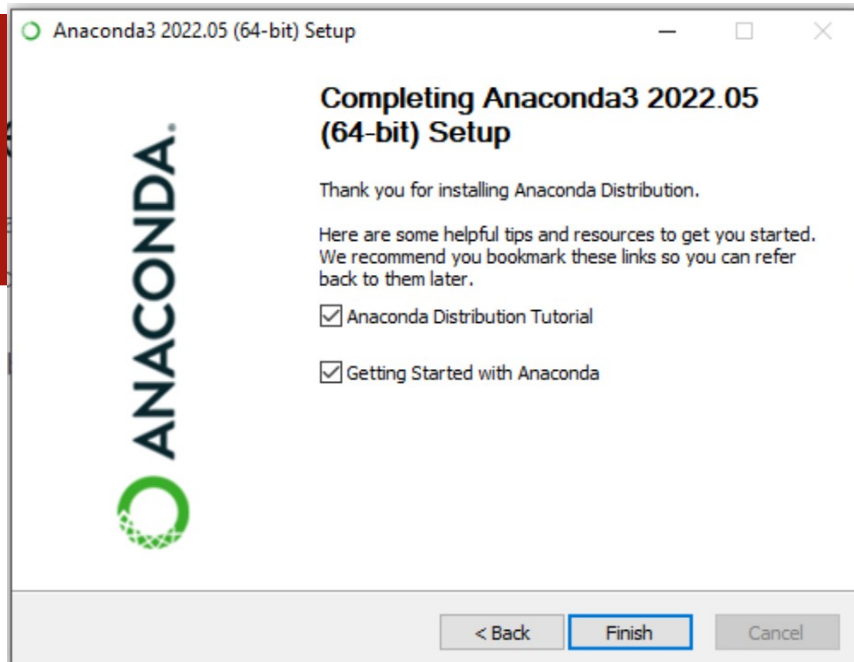
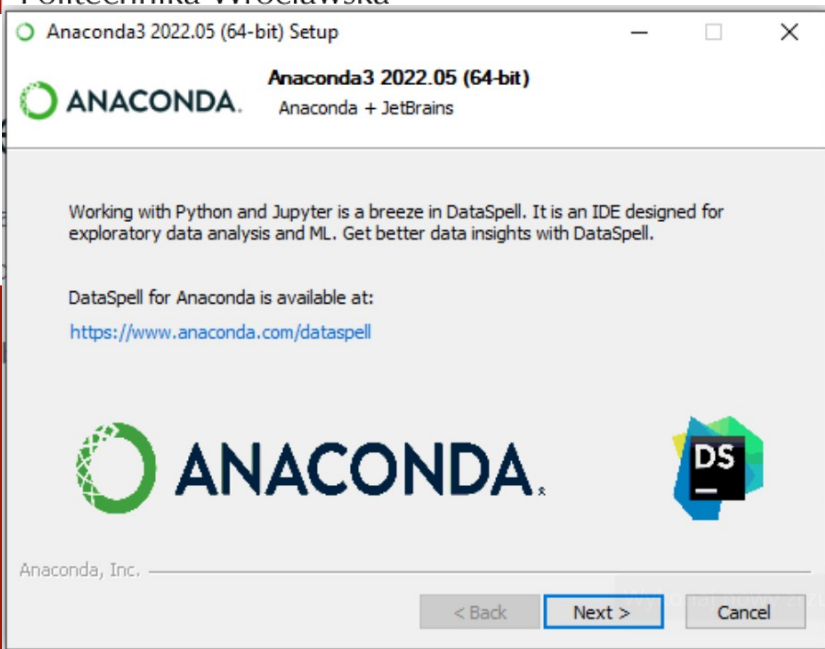
[Download](#)

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

[Get Additional Installers](#)





Installation Success

Welcome to Anaconda!

Here are some useful resources to help you get started.

Create your free [Anaconda Nucleus](#) account today to get access to training materials, how-to videos, and expert insights, all free for a limited time to Nucleus members.

Register for Free

Anaconda Distribution Tutorial

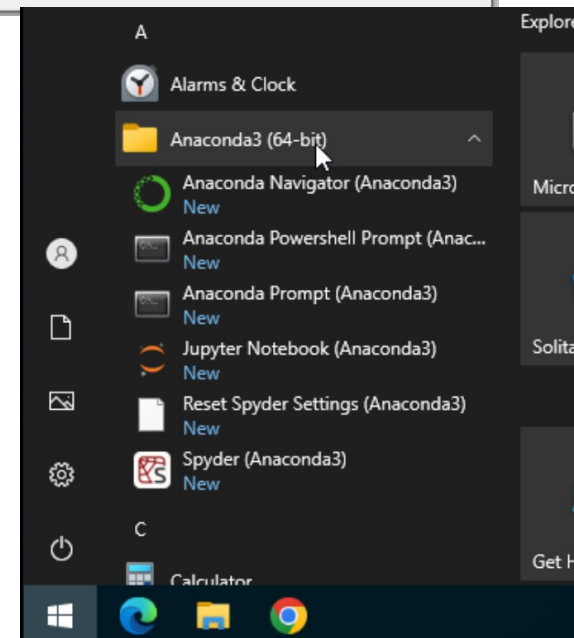
This quick 12-minute tutorial provides an introduction to help you get started using this powerful tool.

[Watch Tutorial](#)

Quick Start Guide

Learn how to use Anaconda Distribution, Anaconda Navigator, and conda with cheat sheets, FAQs, and more.

[Learn More](#)





Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Janusz Andrzejewski

temp.py x

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6
7
8
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Open automatically after writing a left object. You can activate this as > Help.

Spyder? Read our [tutorial](#)

Welcome to Spyder!

Check out our interactive tour to explore some of Spyder's panes and features.

Start tour Dismiss

New Spyder version

Spyder 5.3.3 is available!

Important note: Since you installed Spyder with Anaconda, please **don't** use **pip** to update it as that will break your installation.

Instead, run the following commands in a terminal:

```
conda update anaconda
conda install spyder=5.3.3
```

For more information visit our [installation guide](#).

☒ Check for updates at startup

OK

Console 1/A x

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:
```

Python console History

LSP Python: ready conda: base (Python 3.9.12) CRLF Mem 37%

Spyder – IDE dla Pythona

[←](#)
[→](#)
[↺](#)
<https://docs.spyder-ide.org/current/index.html>

[BLOG](#)
[DOCS](#)

VERSION

Spyder 5

Welcome to Spyder's Documentation

[Edit this page](#)

Search ...

Q

WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

▶ WORKSHOPS

FAQ

Podsumowanie — 1

- Jeśli chcesz, możesz używać na zajęciach własnego komputera z dowolnym systemem operacyjnym (Windows, macOS, Linux)
- Na sprawdzianach oddajesz(wysyłasz) kod źródłowy — który jest zwykłym plikiem tekstowym
- **Pamiętaj — programowanie to nie tylko teoria ale też praktyka. Nauka języka programowania oraz pisanie programów to tak samo ważne rzeczy które powinieneś traktować na równi.**

Podsumowanie – 2

- Zainstalować u siebie na komputerze spyder'a lub inne wybrane IDE
- Zapoznać się z dokumentacją
- Uruchomić najprostszy program w Pythonie

Zadania:

1) Napisz funkcję obliczającą silnię dla liczby naturalnej:

- Przy pomocy pętli for oraz while
- Licząc od „początku” tzn. $1*2*3*...*(n-1)*n$
- Licząc od „końca” tzn. $n*(n-1)*...*2*1$

W głównym segmencie poproś o podanie liczby naturalnej, wywołaj wszystkie funkcje oraz z odpowiednim komentarzem wyświetl wartość funkcji.

2) Napisz funkcję, która dla danego łańcucha (argument) policzy ile jest głosek bezdźwięcznych: p, t, k, f, s, ś, sz, c, ć, cz, h, ch dla zadanego ciągu znaków (Uwaga: dla słowa część mamy 3 a nie 4 spółgłoski bezdźwięczne). Umiesz to zadanie zrobić na 2 (albo więcej) istotnie różne sposoby?