Politechnika Wrocławska

Wstęp do programowania
Laboratorium #01 — Fizyka

Janusz Andrzejewski

Wstęp:

- Podstawowe informacje
- 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.

Podstawowe informacje:

- janusz.andrzejewski@pwr.edu.pl
  - Pokój 319, budynek A1
- Wszystkie materiały będą dostępne na e-portalu PWr, stronach domowych:
  https://andrzejewski.wppt.pwr.edu.pl/
  https://tarnowski.wppt.pwr.edu.pl/
- Koordynatorem zajęć (dobór zakresu materiału, listy zadań) jest prowadzący wykład – prof. Karol Tarnowski
- Zajęcia odbywają się w środy, w sali 250 A1.

Pomoc psychiczna/psychologiczna:

- wszystkie informacje na temat możliwości pomocy znajdują się na stronie https://ddo.pwr.edu.pl/
- każdy student, który potrzebuje pomocy w przypadku powyższych problemów powinien zwrócić się do DDO - Działu Dostępności - bud. C13, p. 1.09 oraz 1.07, tel.(71)320-43-20
- każdy student, który potrzebuje pomocy w przypadku powyższych problemów może zwrócić się do Liderów Dostępności na naszym wydziale:
  - Damian Siedlecki p. 342 L1, damian.siedlecki@pwr.edu.pl
  - Daniel Brandyk p. 17 A1 tel. (71)320-42-79, daniel.brandyk@pwr.edu.pl
  - Agnieszka Dębowska p. 227a A1, tel. (71)320-42-59, agnieszka.debowska@pwr.edu.pl
- **wszystkie informacje objęte są tajemnicą.**
- uczelnia posiada Poradnię Psychologiczną - w przypadku uzasadnionym, liderzy pomogą umówić wizytę w tym samym dniu.

Zaliczenie:*

Punkty	Ocena
100+	Cel (5.5)
90-100	bdb (5.0)
80-89	db+ (4.5)
70-79	db (4.0)
60-69	dst+ (3.5)
50-59	dst (3.0)
0 - 49	ndst (2.0)

- Obecność obowiązkowa na zajęciach.
- Każdy sprawdzian będzie za 30 punktów.
- Będą 2+1 sprawdziany:
  - 1 sprawdzian - mniej więcej na 5 zajęciach
  - 2 sprawdzian - mniej więcej na 10 zajęciach
  - 3 sprawdzian - na ostatnim laboratorium, jest możliwość wyboru:
    - Można pisać tylko 3-ci sprawdzian (materiał tylko z ostatnich ~4 zajęć)
    - Można pisać tylko sprawdzian poprawkowy — obowiązuje materiał ze wszystkich zajęć, ocena tylko z tego sprawdzianu jest oceną z laboratorium, można poprawić maksymalnie na db.
- Dodatkowe 10+ punktów będą przyznawane za aktywność podczas zajęć laboratoryjnych.

* Dokładne szczegóły lub ewentualne zmiany zostaną podane na wykładzie.

Kolokwia:

1) Na kolokwiach **nie wolno** korzystać z internetu lub jakiejkolwiek pomocy zdalnej, sztucznej inteligencji, telefonów komórkowych, własnych laptopów itp. rzeczy
2) Na kolokwiach zadania rozwiązujemy tylko na komputerach PWr dostępnych w sali
3) Na kolokwiach można korzystać z własnych notatek czy prezentacji – w formie klasycznej lub elektronicznej.
4) Każde zadanie stanowi osobny plik tekstowy, który musi zawierać w pierwszych linijkach jako komentarz:
   - Imię i nazwisko
   - Nr indeksu
   - Nr komputera na którym się rozwiązuje zadania

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-czym-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://www.tiobe.com/tiobe-index/

Programming Language	2024	2019	2014	2009	2004	1999	1994	1989
Python	1	4	8	6	11	27	22	-
C	2	2	1	2	2	1	1	1
C++	3	3	4	3	3	2	2	3
Java	4	1	2	1	1	13	-	-
C#	5	6	5	8	8	30	-	-
JavaScript	6	8	9	9	9	20	-	-
Visual Basic	7	19	-	-	-	-	-	-
PHP	8	7	6	5	6	-	-	-
SQL	9	9	-	-	7	-	-	-
Assembly language	10	11	-	-	-	-	-	-
Objective-C	27	10	3	38	48	-	-	-
Lisp	33	28	14	17	15	12	7	2
(Visual) Basic	-	-	7	4	5	3	3	7

Tiobe - „dynamika”:

Feb 2024	Feb 2023	Change	Programming Language	Ratings	Change
1	1		Python	15.16%	-0.32%
2	2		C	10.97%	-4.41%
3	**3**		**C++**	**10.53%**	**-3.40%**
4	4		Java	8.88%	-4.33%
5	5		C#	7.53%	+1.15%
6	7	↑	JavaScript	3.17%	+0.64%
7	8	↑	SQL	1.82%	-0.30%
8	11	↑	Go	1.73%	+0.61%
9	6	↓	Visual Basic	1.52%	-2.62%
10	10		PHP	1.51%	+0.21%
11	24	⇈	Fortran	1.40%	+0.82%
12	14	↑	Delphi/Object Pascal	1.40%	+0.45%
13	13		MATLAB	1.26%	+0.27%
14	9	⇊	Assembly language	1.19%	-0.19%
15	18	↑	Scratch	1.18%	+0.42%

python™

- 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
- „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

- **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)
  - Python 3.12 — 2 października 2023 (wsparcie do 2028-10)
- Obecnie „najnowszą” wersją jest **3.12.2 — 6 luty 2024**

(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 (obiektowy, 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 jednocześnie – problem z GIL'em (Global Interpreter Lock)

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

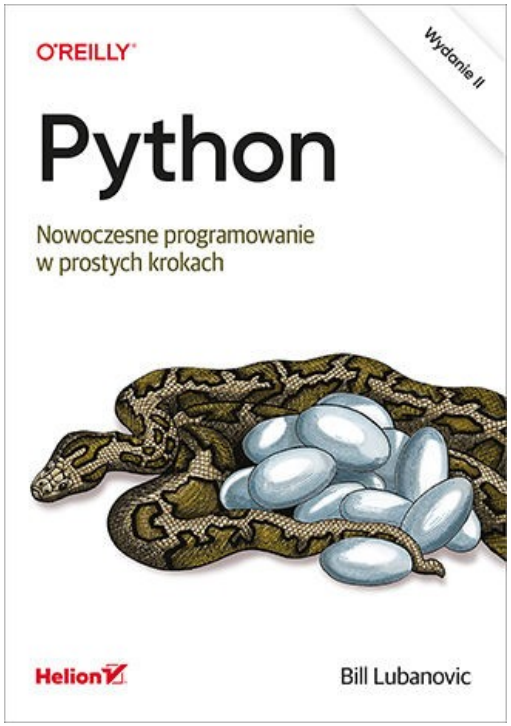
Filozofia — zalecenia

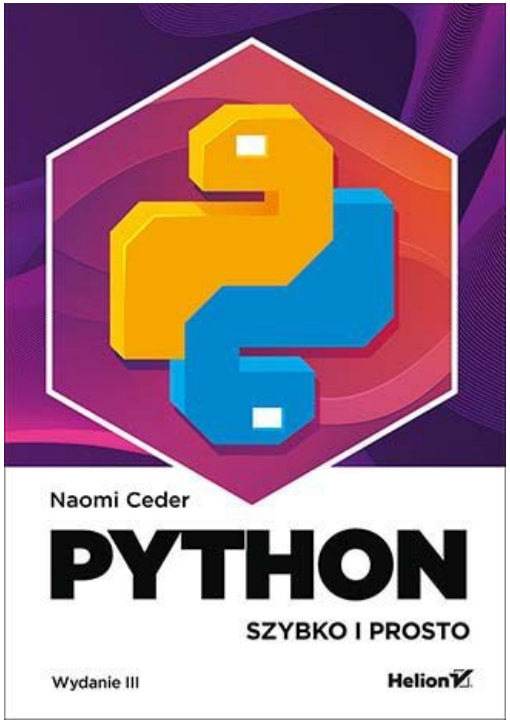
- Każdy język programowania ma swój „duch”, styl w jaki sposób powinniśmy tworzyć programy:
  - Struktura
  - Składnia
  - Typy danych
  - Wbudowane biblioteki
- Ucząc się Pythona, staraj się zrozumieć duch i filozofię tego języka
- Bądź po prostu **PYTHONISTĄ**

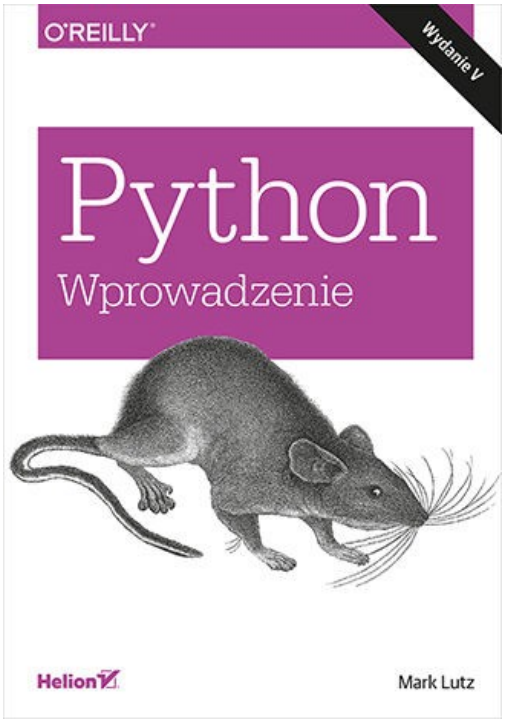
Cel zajęć:

- Opanowanie podstawowej znajomości języka programowania — sposoby zapamiętywania danych oraz ich przetwarzania (obróbki), podstawowych schematów programowania.
- 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)

Literatura


O'REILLY
Wydanie II
Python
Nowoczesne programowanie
w prostych krokach
Helion
Bill Lubanovic


Naomi Ceder
PYTHON
SZYBKO I PROSTO
Wydanie III
Helion


O'REILLY
Wydanie V
Python
Wprowadzenie
Helion
Mark Lutz

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

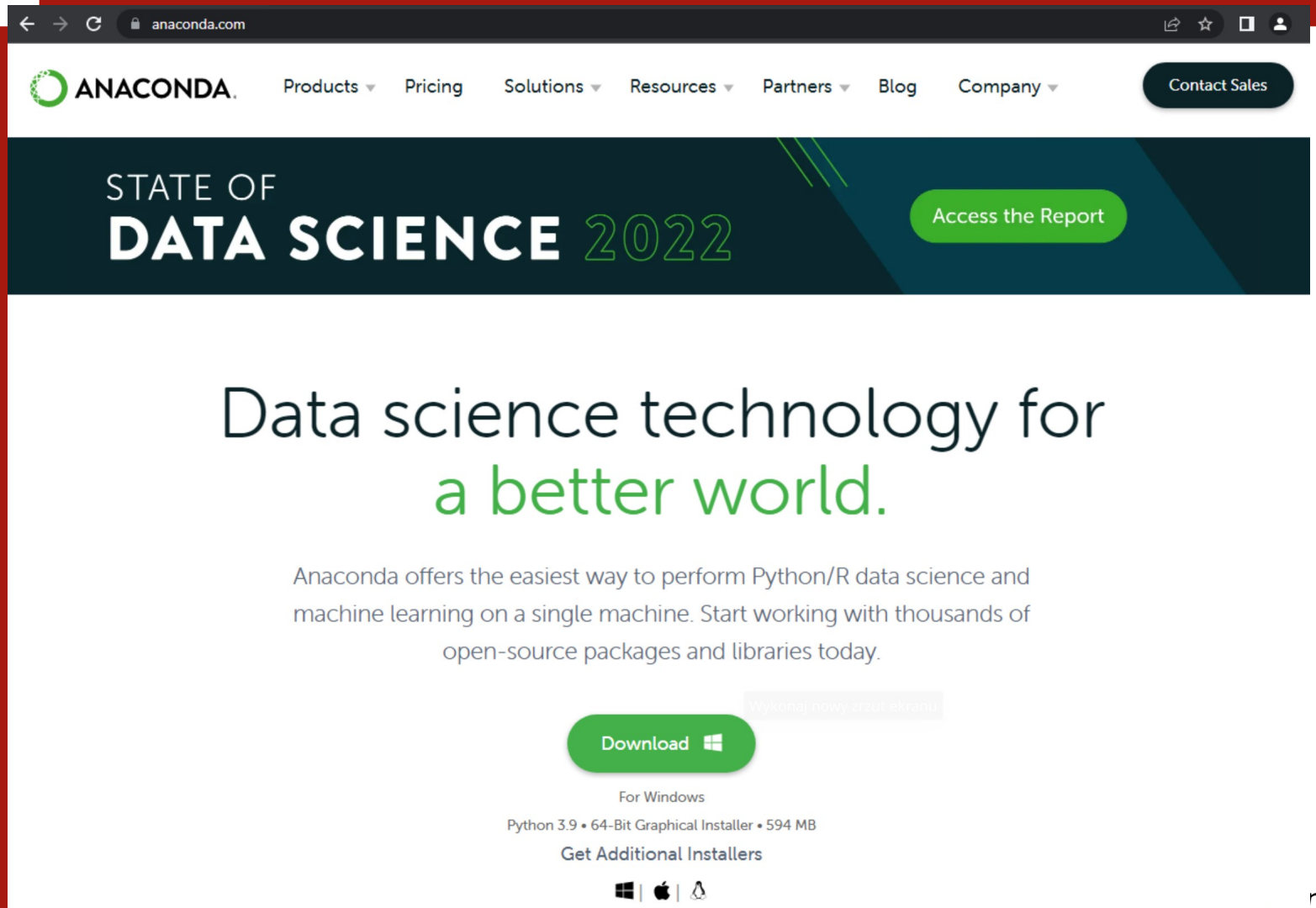
Dlaczego IDE

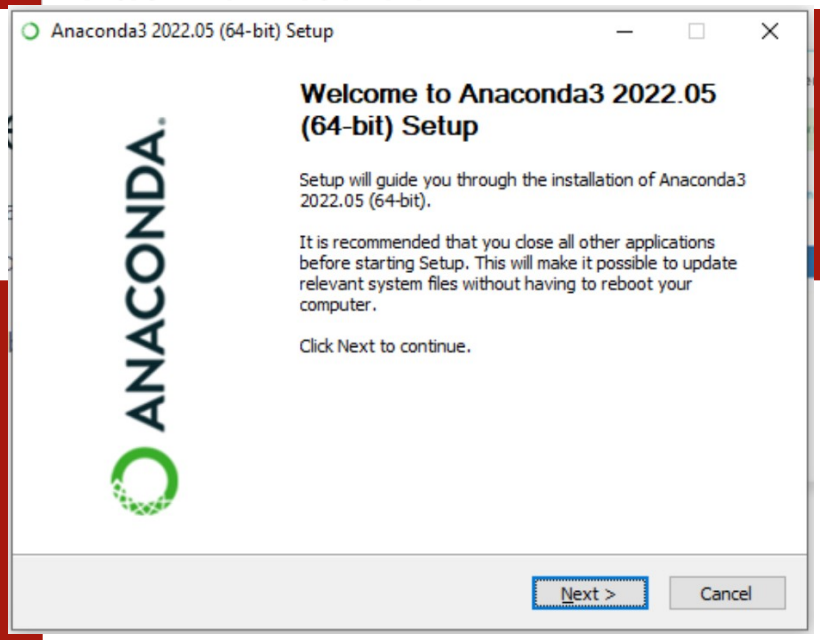
Można zainstalować „gołego” Pythona ze strony https://www.python.org, ale:

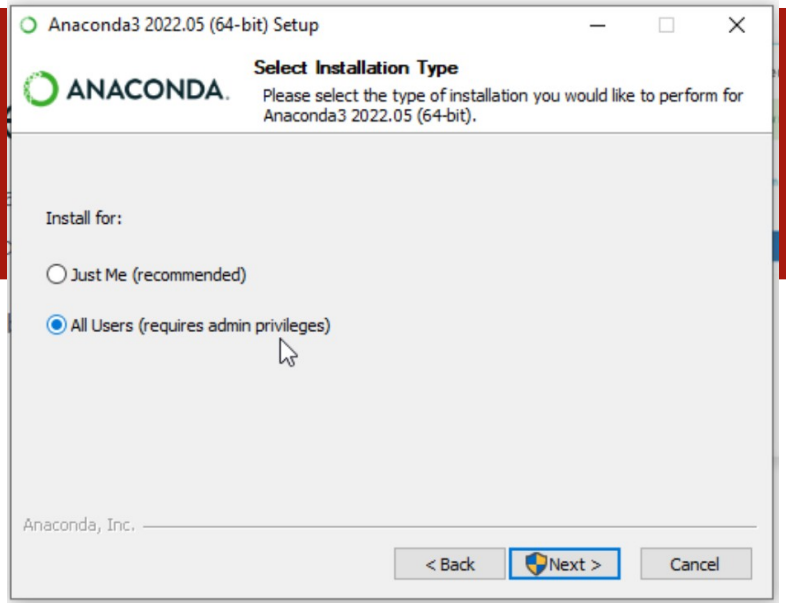
IDE — to zintegrowane środowisko programistyczne, które:

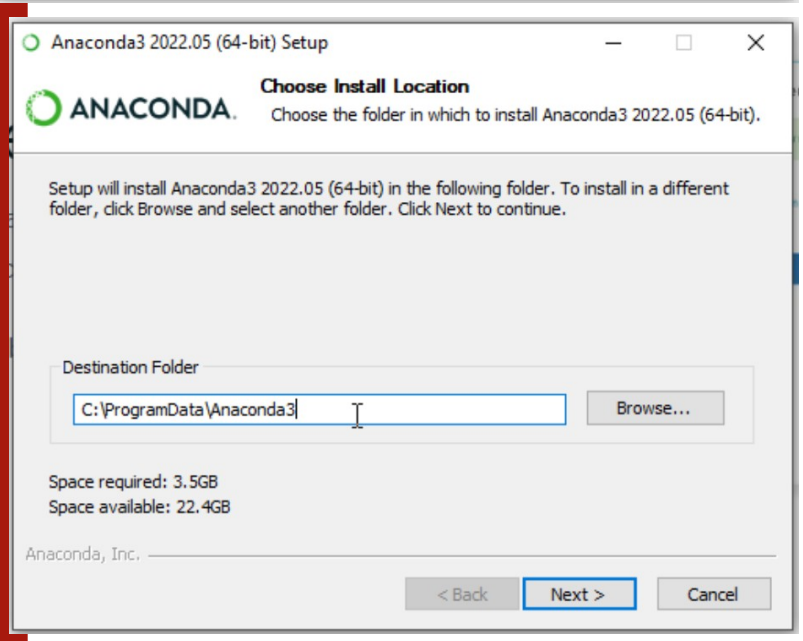
- Pozwala na podświetlenie składni
- Pozwala na łatwe zarządzania projektami czy też samymi plikami
- Ułatwia dostęp do dokumentacji
- Pozwala na łatwiejsze usuwanie błędów
- Wymaga na początku opanowania – dodatkowa nauka
- Jest wiele różnych środowisk
  - spyder - https://www.spyder-ide.org
  - vscode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

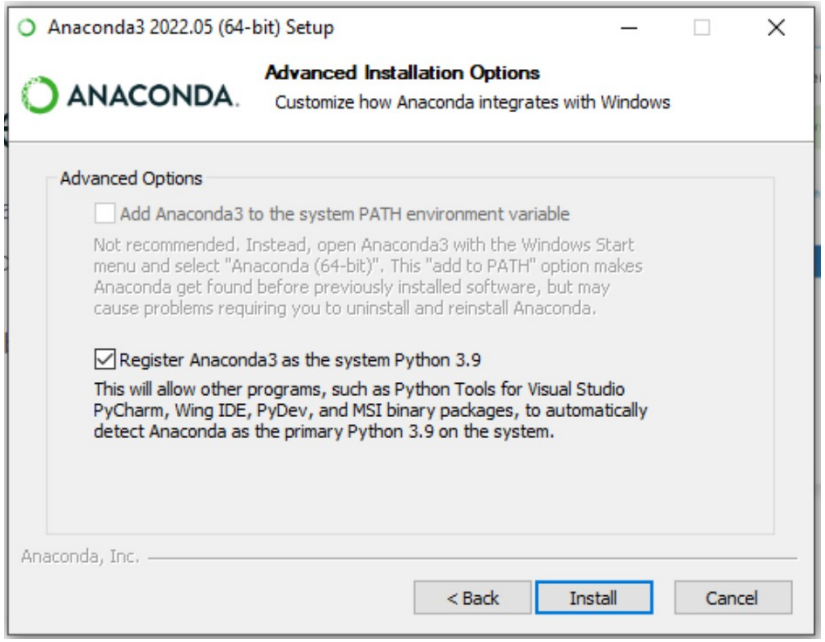
Dystrybucja Pythona — anaconda.com

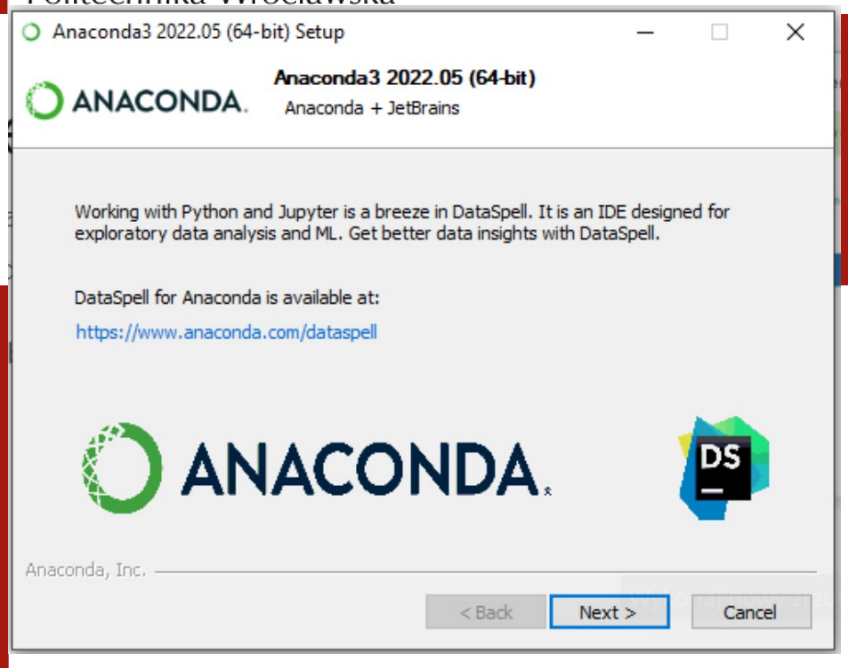


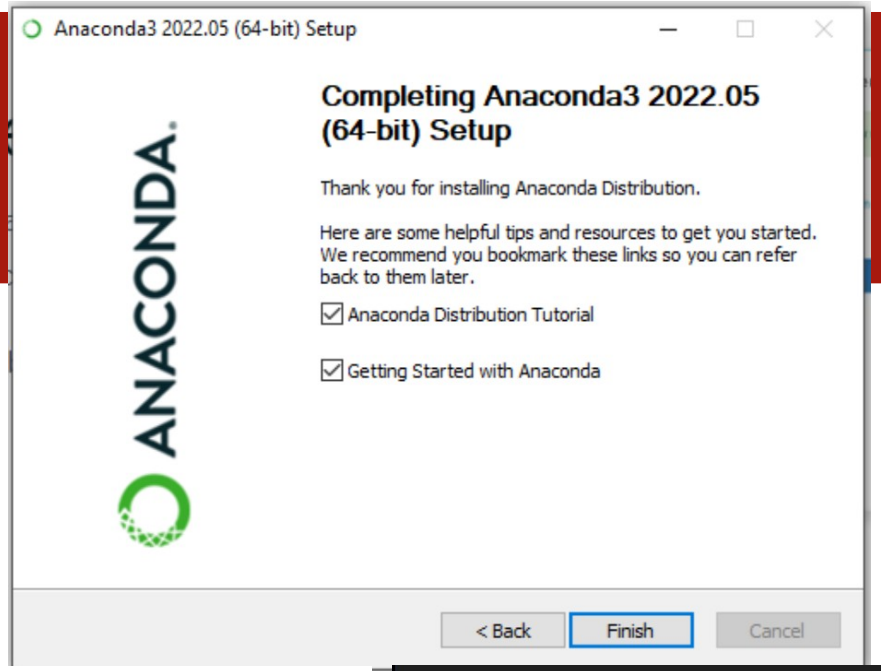

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Welcome to Anaconda3 2022.05 (64-bit) Setup
Setup will guide you through the installation of Anaconda3 2022.05 (64-bit).
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

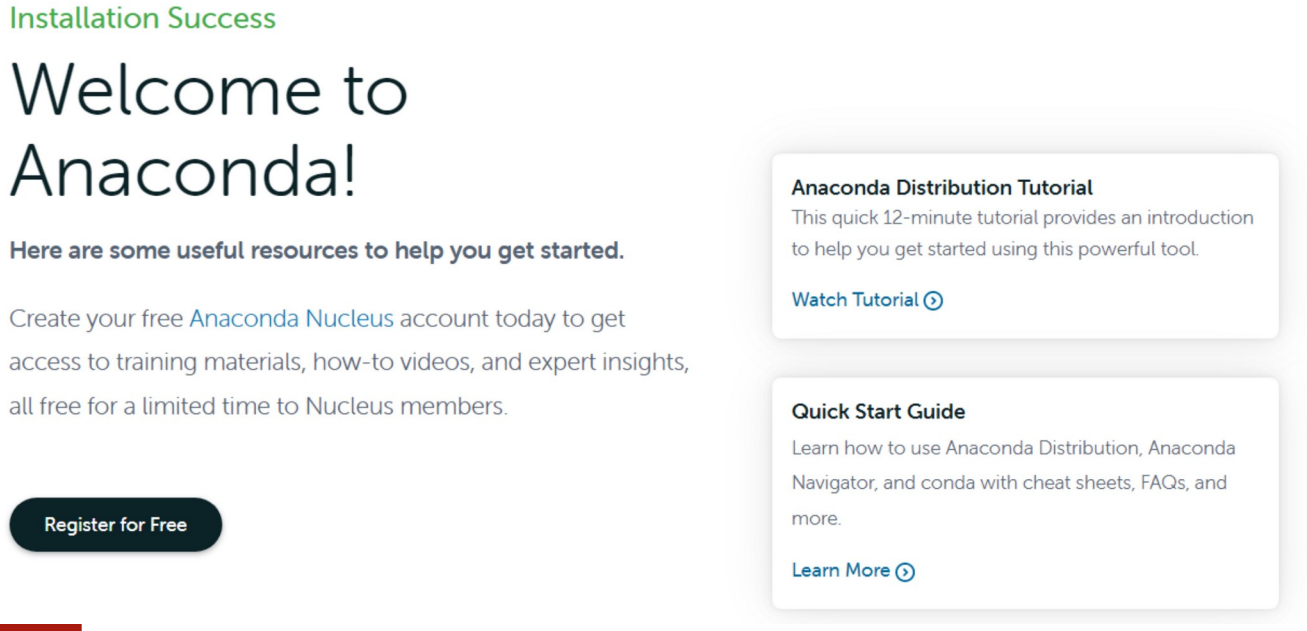

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2022.05 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

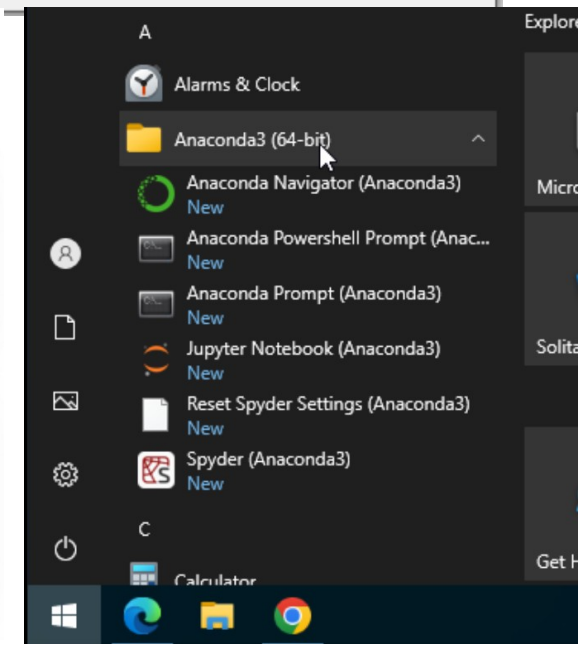

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Choose Install Location
Choose the folder in which to install Anaconda3 2022.05 (64-bit).
Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\ProgramData\Anaconda3
Browse...
Space required: 3.5GB
Space available: 22.4GB
Anaconda, Inc.
< Back
Next >
Cancel

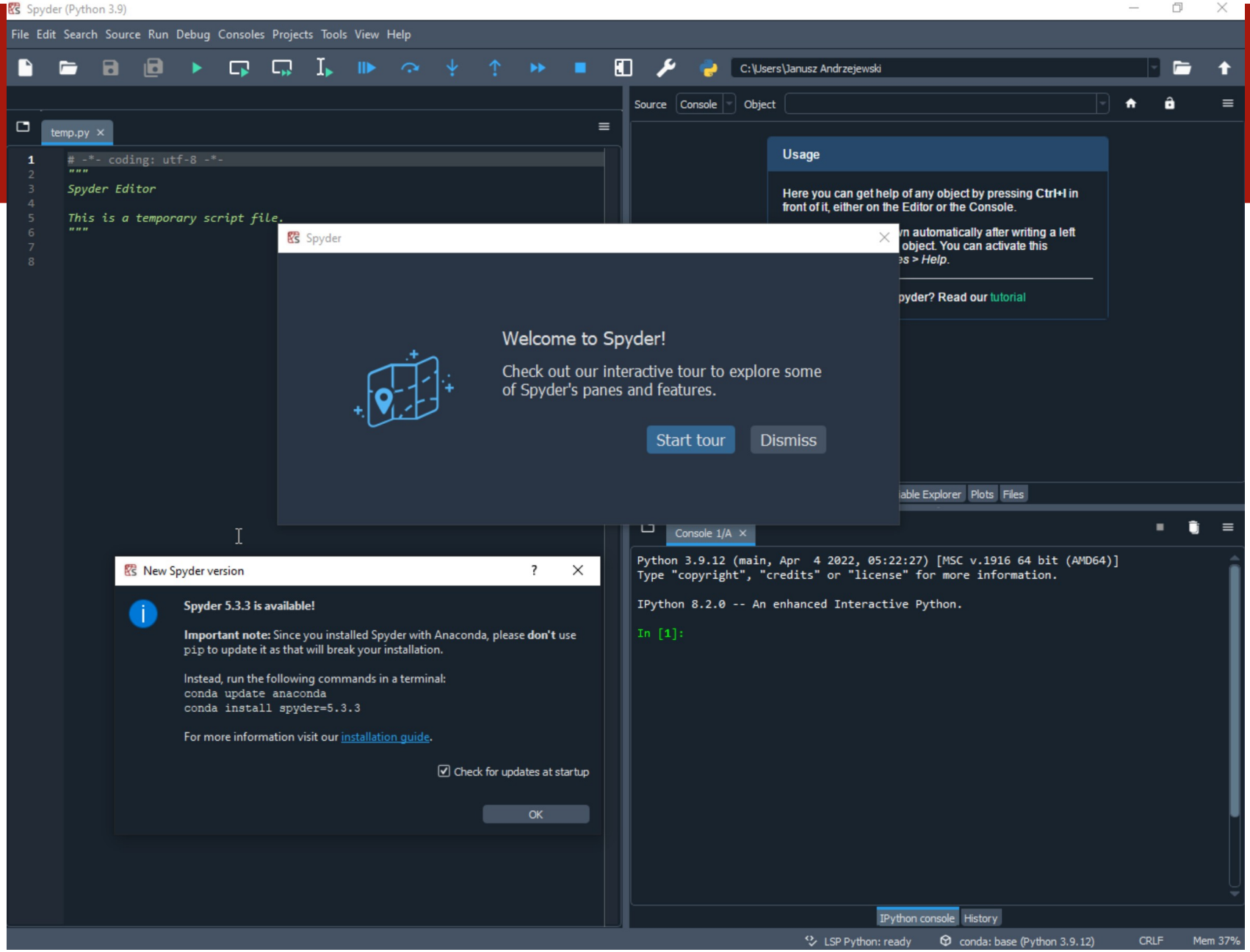

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda3 to the system PATH environment variable
Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda3 as the system Python 3.9
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.
Anaconda, Inc.
< Back
Install
Cancel


Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Anaconda3 2022.05 (64-bit)
Anaconda + JetBrains
Working with Python and Jupyter is a breeze in DataSpell. It is an IDE designed for exploratory data analysis and ML. Get better data insights with DataSpell.
DataSpell for Anaconda is available at:
https://www.anaconda.com/dataspell
ANACONDA.
DS
Anaconda, Inc.
< Back
Next >
Cancel

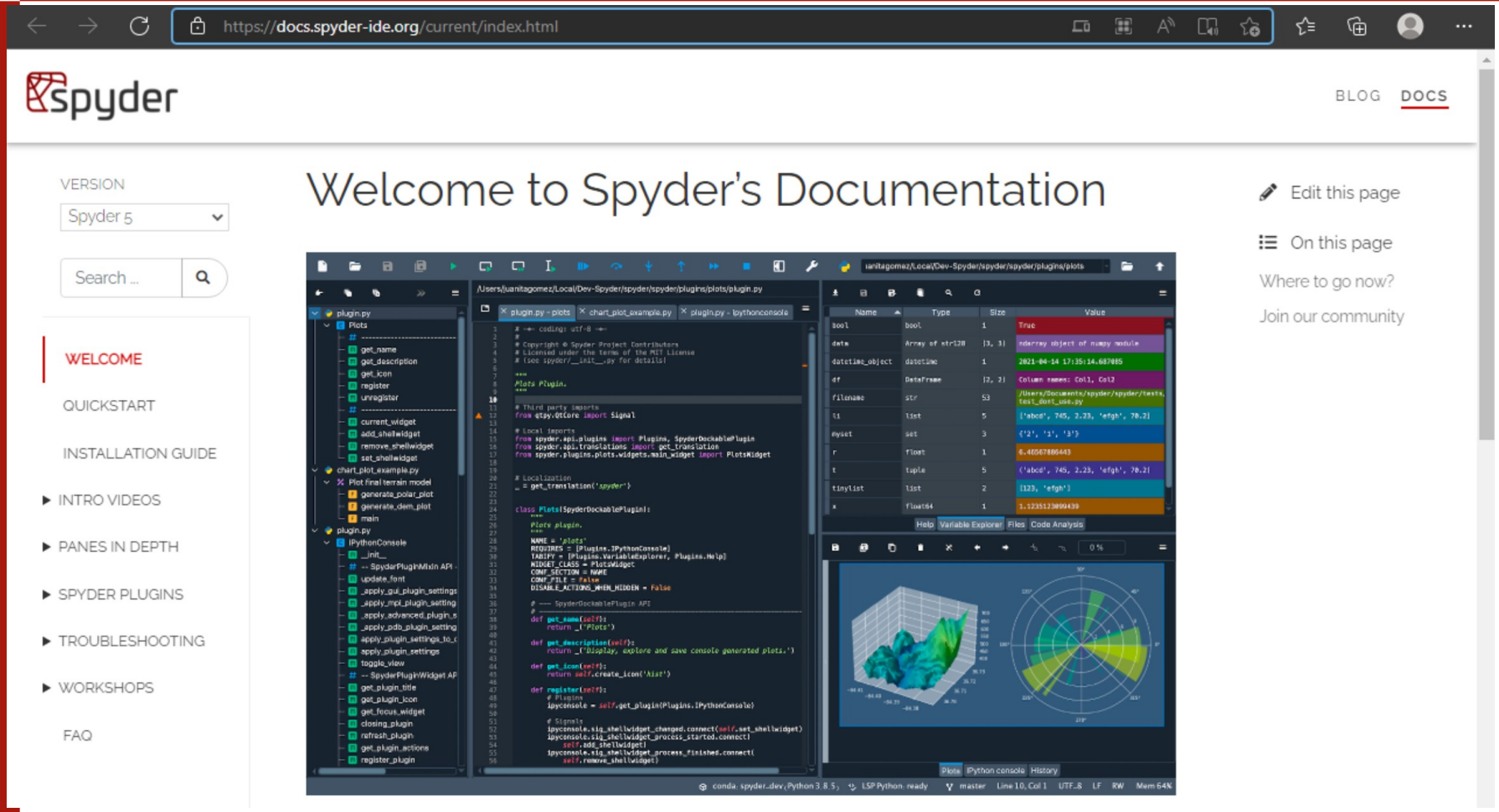

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Completing Anaconda3 2022.05 (64-bit) Setup
Thank you for installing Anaconda Distribution.
Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.
Anaconda Distribution Tutorial
Getting Started with Anaconda
< Back
Finish
Cancel


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


A
Alarms & Clock
Anaconda3 (64-bit)
Anaconda Navigator (Anaconda3)
New
Anaconda Powershell Prompt (Anac...
New
Anaconda Prompt (Anaconda3)
New
Jupyter Notebook (Anaconda3)
New
Reset Spyder Settings (Anaconda3)
New
Spyder (Anaconda3)
New
C
Calculator


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Janusz Andrzejewski
temp.py
-*- coding: utf-8 -*-
"""
Spyder Editor
This is a temporary script file.
"""
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Spyder
Welcome to Spyder!
Check out our interactive tour to explore some of Spyder's panes and features.
Start tour
Dismiss
Plots Files
Console 1/A
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]:
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
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
CRLF
Mem 37%

Spyder — IDE dla Pythona



Podsumowanie:

- 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.**