About the course

AIMS

Familiarize students with Python, the standard library, and selected external packages.

Training in finding solutions for many problems that occur in everyday scientific programming. #zadania domowe cotygodniwoe, wymagana ilośc min. 10. Praca systematyczna! Tempo jest szybkie!

Training in using best programming practices: writing readable code, testing programs, writing documentation. #czytelnośc kodu, testowanie oraz pisanie dokumentacji (np. manual)

DESCRIPTION

This course will be online-only. I will be providing online workshop material in the form of web pages and other resources. The material is arranged into weekly sets of topics. Coursework will consist of weekly homeworks and a final project. Collaboration on homeworks and projects is not permitted.

For questions about the course please contact me by my email or Teams.

EXPECTATIONS

Basic knowledge of Microsoft Windows 10 and Linux. #można się uczyc od zera :)

No prior experience with Python is required.

FINAL PROJECTS

A list of suggested projects subjects will be given later.

A typical project should include computations, file management, and data visualization. #program do naszych obliczeń + pdf lub readme

You should choose your final project in the middle of the semester.

The final result of the project is the Python code and the short documentation (PDF, HTML, TXT, or README.md file).

FEATURES

Portability (mainstream operating systems are supported).

Code readability, easy maintenance, white spaces are significant (a more uniform style is enabled).

Object-oriented approach.

Dynamic types, high-level builtin datatypes. #typy dynamiczne - możemy zmieniać typ w kodzie

Automatic memory management with garbage collection. #obiekt bez referencji jest niszczony

Reliable scalability. #

Multiple programming paradigms are supported (procedural, object-oriented, functional programming).

A comprehensive standard library ('batteries included'). #duża standardowa biblioteka - baterie sa załączone :)

A ton of useful frameworks and external libraries.

Growing Python community (many tutorials, blogs).

Many people consider using Python a pleasure!

APPLICATION DOMAINS

System programming (processes, administration tasks)

GUI programming #interfejs graficzny do programów

Web Development

Scientific programming

Machine Learning

Education (Python as the first programming language you learn)

Integration (mixing Python with other languages)

Database Management

Finance

Internet of Things

Startups

SHORTCOMINGS #wady

https://wiki.python.org/moin/GlobalInterpreterLock

https://realpython.com/python-gil/

Speed. But: **#jezyk interpretowany a nie kompilowany!**

(a) Some parts of Python run with C speed (specialized libraries). **#obliczenia numeryczne można spiąć z językiem C/C++**

(b) Coding time is reduced, many ideas can be implemented and tested (making working prototypes).

(c) Sometimes there are other slow components: internet connection, database response, etc.

(d) Faster algorithms can be used.

(e) There are tools to improve the Python performance: compilation with Cython, compilation 'just in time' with Numba, caching. **#narzędzia do przyspieszania**

Python GIL (Global Interpreter Lock). **#praca and 1 elementem naraz, bez parallel**

It is a mutex (or a lock) that protects access to Python objects, preventing multiple threads from executing Python bytecodes at once. This mutex is necessary mainly because CPython's memory management is not thread-safe [threads share the same memory].

(a) Luckily, many potentially blocking or long-running operations, such as I/O, image processing, and NumPy number crunching, happen outside the GIL.

(b) The most popular way to avoid GIL is to use a multiple processes instead of threads (a 'multiprocessing' module).

(c) One can use alternative Python interpreters, not CPython.

**PYTHON IMPLEMENTATIONS**

https://www.python.org/download/alternatives/

* CPython (the reference implementation written in C and Python). **#główny interpreter; w języku C**
* IronPython (Python written in C#, running on .NET).
* Jython (Python written in Java for the Java Virtual Machine).
* PyPy (a fast Python implementation with a JIT compiler, written in RPython [Restricted Python]).
* Stackless Python (branch of CPython supporting microthreads).
* MicroPython (Python running on micro controllers).

**INSTALLING PYTHON**

* Download from **www.python.org** [the full installer, recommended].   
  32-bit [old computers] vs 64-bit, the web installer vs the offline installer.   
  **'Install Now' option.** Admin access is not needed. Python will be installed into your user directory. The standard library, test suite, launcher and pip will be installed. Shortcuts only for the current user.   
  [ ] Add Python 3.x to PATH (option unchecked)   
  **'Customize installation' option.** Admin acces needed. Python will be installed into the Program Files directory. Optional features may be selected during installation. The standard library can be pre-compiled to bytecode. Shortcuts are available for all users.

#dodaje programik py zarządzający wersjami pythona

* Download from Microsoft Store. #prowadzący tak lubi  
  A simple installation for an educational environment. Problems with a path length limit. The 'py' command is not included. Admin access is not needed. Automatic updates provided.
* Install Anaconda or Miniconda. #**z dodatkowymi bibliotekami**
* Install Enthought Python.

Po instalacji: **python.exe -> tryb interaktywny (konsoli)**

* Zmienne przechowują referencje do obiektów danego typu

**IDLE # okienkowa wersja interpretera**

* Kod znika po zamknięciu
* File ->New File -> podświetlenie składni, F5 uruchamia kod w IDLE

**Cmd.exe -> wpisujesz „python”** u mnie nie działa

**Powershell ->** też nie działa, otwiera się Microsoft Store

### NOTEPAD++

https://notepad-plus-plus.org/

Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License.

Based on the powerful editing component Scintilla, Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size.

**Anaconda**

https://www.anaconda.com/   
There is the open-source Individual Edition (Distribution). Conda Packages (the 'conda install' command). Anaconda Installers for Python 3.9 (Windows, MacOS, Linux).

https://docs.anaconda.com/anaconda/user-guide/getting-started/   
Getting started with Anaconda

https://conda.io/en/latest/

https://www.infoworld.com/article/3564381/how-to-run-anaconda-side-by-side-with-other-pythons.html   
How to run Anaconda side by side with other Pythons

https://anaconda.cloud   
Anaconda Nucleus (Anaconda in the Cloud; registration required).

### INTRODUCTION

Anaconda Individual Edition contains conda **and Anaconda Navigator**, as well as Python and hundreds of scientific packages.

Conda works on your command line interface such as Anaconda Prompt on Windows and terminal on MacOS and Linux.

Navigator is a desktop graphical user interface that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

Anaconda Navigator can be connected to the community portal **'Anaconda Nucleus'** in order to securely store local environments in the cloud. #wersja chmurowa

### SYSTEM REQUIREMENTS

Operating system: Windows 8 or newer, 64-bit Linux.

System architecture: Windows- 64-bit x86, 32-bit x86; Linux- 64-bit x86.

Minimum 5 GB disk space to download and install.

On Windows, MacOS, and Linux, it is best to install Anaconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Anaconda system wide, which does require administrator permissions.

You do not need to uninstall other Python installations or packages before installing Anaconda.

### WINDOWS

https://docs.anaconda.com/anaconda/install/windows/

A suggested folder for Anaconda is C:\anaconda\.   
The 'default path' in Windows 10 is C:\Users\your\_username\Anaconda3\.

We recommend that you do not add Anaconda to the Windows PATH because this can interfere with other software. Instead, open Anaconda with the Start Menu and select Anaconda Prompt, or use Anaconda Navigator.

**PyCharm** can be installed with Anaconda. #IDE

The default Anaconda installation option is 'Just Me'. If you have administrator access, you can choose to install for 'All Users'. A new group AnacondaUsers should be created.

From the Start menu, click the Anaconda Navigator desktop app.

### LINUX

https://docs.anaconda.com/anaconda/install/linux/

Download the Anaconda installer for Linux (64-Bit, x86).   
Install the following extended dependencies for Qt (Debian Linux):   
apt-get install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6

The 'default path' in Linux is /home/your\_username/anaconda3.

Multi-user Anaconda installation on Linux requires sudo access. A new group should be created for Anaconda users.

Open a terminal window and type 'anaconda-navigator'.

https://docs.anaconda.com/anaconda/install/uninstall/   
Uninstalling Anaconda: (a) use simple remove, (b) use Anaconda-Clean (conda install anaconda-clean).

### CONDA

Conda is a completely separate tool to pip, virtualenv and wheel, but provides many of their combined features in terms of package management, virtual environment management and deployment of binary extensions.

# IDLE

https://docs.python.org/3/library/idle.html

### INTRODUCTION

IDLE is Python’s Integrated Development and Learning Environment.

It is coded in 100% pure Python, using the tkinter GUI toolkit.

IDLE is cross-platform: works mostly the same on Windows, Unix, and MacOS.

It has an interactive interpreter, a multi-window text editor, a debugger.

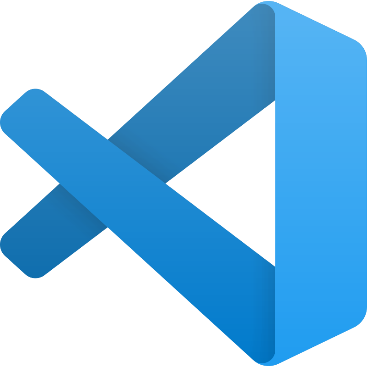
IDLE has two main window types, the Shell window and the Editor window.

### DEBIAN 10

Debian packages: idle, idle-python2.7, idle-python3.7.

**IDE**

# Visual Studio Code



https://code.visualstudio.com/

https://docs.microsoft.com/en-us/windows/python/beginners

### INTRODUCTION

By using VS Code as your text editor / integrated development environment (IDE), you can take advantage of IntelliSense (a code completion aid), Linting (helps avoid making errors in your code), Debug support (helps you find errors in your code after you run it), Code snippets (templates for small reusable code blocks), and Unit testing (testing your code's interface with different types of input).

Install the Python extension in order to work with Python.

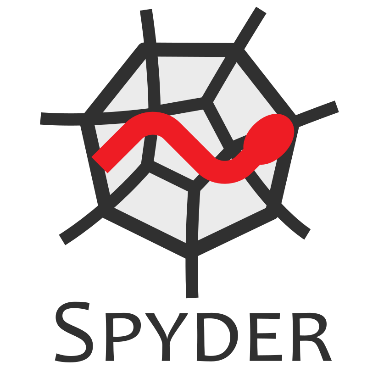
VS Code supports version control with Git.

### INSTALLING VS CODE

Windows: user installer.

Linux: deb and rpm packages.

# 2. Spyder IDE



https://www.spyder-ide.org/   
Spyder. The Scientific Python Development Environment

### INTRODUCTION

Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

### INSTALLING SPYDER

Debian Linux packages: spyder (Python 2), spyder3 (Python 3).   
[Menu Programy|Programowanie|Spyder or Spyder3]

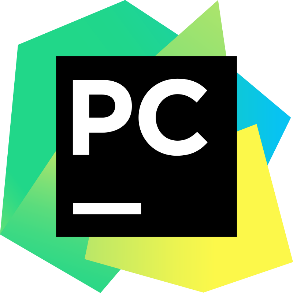
https://pypi.org/project/spyder/   
pip install spyder [for experienced users, a virtual environment recommended]

Download with Anaconda.   
Spyder is included by default in the Anaconda Python distribution.

Windows: standalone installer.

Spyder online with Binder in your web browser.

# 3. PyCharm IDE



https://www.jetbrains.com/pycharm/   
PyCharm, The Python IDE for Professional Developers.

https://www.jetbrains.com/help/pycharm/2021.1/quick-start-guide.html

https://tecadmin.net/how-to-install-pycharm-on-debian-10/   
How to Install PyCharm on Debian 10.

https://ubuntu.com/tutorials/basic-snap-usage-pl   
Ubuntu Tutorials: Podstawy korzystania ze snapów.

https://snapcraft.io/tutorials/advanced-snap-usage   
Advanced snap usage.

https://linuxiarze.pl/snap/

### INTRODUCTION

PyCharm is the Python IDE with many useful features. It has an intelligent Python assistance like smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, and rich navigation capabilities. Supported languages: Python 2.7, Python 3.6+.

PyCharm is available in community (free, open-source) and professional editions (licence).

In PyCharm you are working within the context of a project. There are three options to start working on a project:   
(1) Open an existing project   
(2) Check out a project from version control   
(3) Create a new project

### LINUX DEBIAN 10

The PyCharm community and professional editions are available as snap package. You must have Python installed on your system and sudo privileged account access.

Snap is a software packaging and deployment system developed by Canonical for the operating systems that use the Linux kernel.

Snap Store is a graphical desktop application for discovering, installing and managing snaps on Linux. Snap Store is based on GNOME Software, optimized for the Snap experience.

$ sudo apt update

$ sudo apt install snapd # install snapd

$ sudo snap install core

$ sudo snap install snap-store # install Snap Store

# /snap folder was created.

$ snap find hello # finding 'hello' applications

$ sudo snap install hello-world

$ snap find pycharm

Name Version Publisher Notes Summary

pycharm-community 2020.3.3 jetbrains✓ classic Python IDE for Professional Developers

pycharm-professional 2020.3.3 jetbrains✓ classic Python IDE for Professional Developers

pycharm-educational 2020.3.2 jetbrains✓ classic Easy and Professional Tool to Learn & Teach Programming with Python

$ snap list # list installed snaps

$ sudo snap refresh # update all snaps

$ sudo snap revert hello-world # back to the previous version

$ sudo snap remove hello-world

# Install PyCharm community edition.

$ sudo snap install pycharm-community --classic

# Run PyCharm: /snap/bin/pycharm-community

It is possible to install PyCharm from a tar.gz file: pycharm-community-2020.3.3.tar.gz (457 MB).

https://www.jetbrains.com/help/pycharm/installation-guide.html

### WINDOWS 10

Install PyCharm with Anaconda.

# Jupyter



https://jupyter.org/   
Project Jupyter

https://ipython.org/ **#enchanced jak w Mathematica!!!! Super! Można definiować makra**  
IPython. Interactive Computing

#używa przeglądarki do pracy z kodem – na serwerze WWW otwierać notatniki Jupythera

https://en.wikipedia.org/wiki/Notebook\_interface

https://wiki.debian.org/Jupyter

https://snapcraft.io/install/jupyter/debian   
How to install Project Jupyter on Debian   
(1) Enable snapd   
(2) Install Project Jupyter (sudo snap install jupyter)

https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet   
Markdown Cheatsheet by Adam Pritchard

### INTRODUCTION

The name 'Jupyter' referred to 'Julia', 'Python' and 'R', the initial languages which were implemented.

'IPython' (Interactive Python) provides a rich architecture for interactive computing in Python.

'Project Jupyter' is a spin-off project from IPython. It consists of the notebook and other language-agnostic parts of IPython. It supports execution environments in several dozen languages (Python, Julia, R, C++, Scheme, Ruby).

'Jupyter Notebook' is a web-based interactive computational environment for creating Jupyter notebook documents (JSON documents). The notebook interface is used for various computational back ends (Mathematica, MATLAB, Maple, Python, Julia, Scala). A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python).

'JupyterLab' is the next-generation web-based user interface for Project Jupyter.

An interactive session in Jupyter consists of a 'frontend' (the console, GUI or web interface in which code is entered and results displayed), and a 'kernel' (the process responsible for executing your code). The frontend and kernel communicate using the ZMQ messaging library.

### INSTALLING JUPYTER IN WINDOWS 10

https://medium.com/@kswalawage/install-python-and-jupyter-notebook-to-windows-10-64-bit-66db782e1d02

First install Python 3.9 from www.python.org.

Set 'Python39' path and 'Python39/Scripts' path to environment variable.

Install 'virtualenv' using 'pip install virtualenv'.

Upgrade pip using 'python -m pip install --upgrade pip'.

Install Jupyter using 'python -m pip install jupyter'.

Run Jupyter Notebook using 'jupyter notebook'.

Start the notebook server and popup dashboard in browser using

'localhost:8888/tree' url.

Create your first notebook using dashboard.

### EXERCISES

# IPython in Debian 10 (ipython for Python 2.7, ipython3 for Python 3.7).

$ ipython

Python 2.7.16 (default, Oct 10 2019, 22:02:15)

Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.

? -> Introduction and overview of IPython's features.

%quickref -> Quick reference.

help -> Python's own help system.

object? -> Details about 'object', use 'object??' for extra details.

In [1]: a = [1,2,3]

In [2]: a

Out[2]: [1, 2, 3]

In [3]: import math

In [4]: math.sqrt(3)

Out[4]: 1.7320508075688772

In [5]: quit()

# Google Colaboratory (Colab)



https://colab.research.google.com/notebooks/intro.ipynb **#wprowadzenie**

**dajemy plik -> nowy -> wutomatycznie robi nam w gogle dysk notebook**

https://research.google.com/colaboratory/faq.html

https://www.analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/   
Free GPUs for Everyone! Get Started with Google Colab for Machine Learning and Deep Learning

https://bulldogjob.pl/news/738-google-colab-pythonowy-obszar-roboczy-w-chmurze

### INTRODUCTION

Colab is a free Jupyter notebook environment that runs in the browser using Google Cloud.

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more.

Access to computing resources including GPUs and TPUs. Code is executed in a virtual machine private to your account.

You can run the session in an interactive Colab Notebook for 12 hours.

One more thing to keep in mind is that the dataset you upload in the Colab notebook gets deleted once the session is ended.

Colab notebooks are stored on Google Drive (15 GB of disk space), or can be loaded from GitHub.

Colab notebooks can be shared as files from Google Docs.

### EXAMPLES

Google account required.

Google Drive|Colab Notebooks|first.ipynb

Two kinds of cells: Code (Python), Text (LaTeX, Markdown).

File|New notebook   
File|Revision history

Edit|Notebook settings|Akcelerator sprzętowy: None, GPU, TPU

View|Informacje o notatniku

Insert|Cell with ...

Runtime|Run ...   
Runtime|Change Runtime Type (None, GPU, TPU)

Tools|Settings|Editor

# Using Terminal Commands on Google Colab

!pwd

!ls

!pip install library\_name

Zmienne środowiskowe