

Python - wprowadzenie

1. Pierwsze kroki - warsztat pracy

- 1.1 Co to jest Jupyter i jak działa?
- 1.2 Omówienie wstążki i funkcjonalności
- 1.3 Co to dedykowane środowisko i po co nam ono??
- 1.4 Instalujemy wszystko sami czy korzystamy z gotowych rozwiązań?
- 1.5 Przydatne skróty klawiszowe?
- 1.6 Co to są biblioteki i jakie dobrze znać?

2. Przygotowanie środowiska (Windows)

- 2.1 Instalowanie Pythona
- 2.2 Instalowanie bibliotek
- 2.3 Przygotowanie środowiska
- 2.4 Aktywacja jupytera

1. Pierwsze kroki - warsztat pracy



1.1 Co to jest Jupyter i jak działa?

Jupyter Notebook to open sourceowa aplikacja webowa dzięki której możemy tworzyć i dzielić się dokumentami zawierającymi:

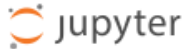
- żywy kod,
- równania,

- wizualizacje,
- załączniki

Oficjalna strona Jupyter Notebooks <https://docs.jupyter.org/en/latest/>

1.2 Omówienie wstążki i funkcjonalności?

- Home Page - Files



QuitLogout

FilesRunningClusters









Select items to perform actions on them.

UploadNew↺

☐ 0 ▾

📁 /

Name ▾Last ModifiedFile size

<input type="checkbox"/>	 Python - podstawy.ipynb	Running 3 minuty temu	2.08 kB
<input type="checkbox"/>	 Untitled.ipynb	3 dni temu	1.4 kB
<input type="checkbox"/>	 Untitled1.ipynb	4 minuty temu	2.07 kB
<input type="checkbox"/>	 DummyData	17 dni temu	2.46 GB
<input type="checkbox"/>	 DummyData.csv	17 dni temu	1.22 GB
<input type="checkbox"/>	 DummyData1.csv	2 dni temu	1.2 kB
<input type="checkbox"/>	 read.py	15 dni temu	539 B
<input type="checkbox"/>	 test.py	4 godziny temu	5.28 kB

- Home Page - Running

Files

Running

Clusters

Currently running Jupyter processes



Terminals ▾

There are no terminals running.

Notebooks ▾



Python - podstawy.ipynb

Python 3 (ipykernel)

Shutdown

kilka sekund temu

- Home Page - Clusters

Files

Running

Clusters

Clusters tab is now provided by IPython parallel. See '[IPython parallel](#)' for installation details.

- Notebook

Python - wprowadzenie

1. Pierwsze kroki - warsztat pracy

- 1.1 Co to jest Jupyter i jak działa?
- 1.2 Omówienie wstążki i funkcjonalności?

1.3 Co to dedykowane środowisko i po co nam ono?

Każdy Data Engineer, Data Scientist lub osoba programująca w Python prędzej czy później boryka się z następującymi problemami:

- konieczność wydzielenia obszaru roboczego,
- konieczność pracy na różnych wersjach pythona i bibliotekach,
- problem z zapanowaniem nad ścieżkami i zmiennymi
- negatywny wpływ bibliotek na siebie nawzajem

Dlatego na początku pracy z pythonem lub w miarę zaawansowania dobrze jest poznać metodę która pozwala na zarządzanie środowiskiem w którym będziemy pracować, oraz która pozwoli na pełną separację od innych elementów.

1.4 Instalujemy wszystko sami czy korzystamy z gotowych rozwiązań?

Pythona wraz z dodatkami można zainstalować i konfigurować na kilka sposobów:

- można zrobić to samemu ściągając oficjalne plik instalacyjny z interesującą nas wersją
- skorzystać z gotowej dystrybucji bądź dodatków do jakiegoś programu

Co lepsze? Wszystko zależy od potrzeb i stopnia zaawansowania. Natomiast z dobrych praktyk poleca się zainstalowanie wszystkiego samodzielnie ponieważ:

- budujemy wtedy nasze kompetencje
- wiemy co i gdzie się znajduje
- kontrolujemy wersje naszych instalacji

- unikamy instalacji zbędnych dodatków

Gotowe dystrybucje lub dodatki polecane są dla użytkowników:

- którzy trochę po omacku samodzielnie chcą przygotować sobie środowisko
- którym dystrybucje spełniają dodatkowe wymagania których oczekuje użytkownik
- zaawansowanych użytkowników którzy skorzystają z dystrybucji w pełni i będą potrafili je rekonfigurować

Przykłady dystrybucji i dodatków:

Anaconda

<https://www.anaconda.com/>

Start Coding Immediately

Spin up awesome data science projects anytime, anywhere!

 Code in the cloud

 Download

Get Additional Installers



Dodatki programu Visual Studio Code



EXTENSIONS: MARKETPLACE



python

**Python**

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter ...

Microsoft

564ms

**Python for VSCode**

Python language extension for vscode

Thomas Haakon Townsend

5.1M ★ 2

Install ⚠

**Python Extension Pack**

Popular Visual Studio Code extensions for Python

Don Jayamanne

4.9M ★ 4

Install

**Python Indent**

Correct Python indentation

Kevin Rose

4.5M ★ 4.5

Install

**autoDocstring - Python Docstring Generator**

Generates python docstrings automatically

Nils Werner

4.6M ★ 5

Install

**Python Environment Manager**

View and manage Python environments & packages.

Don Jayamanne

3.5M ★ 3

Install

**Python**

Extensions for Python

shiro

24K ★ 5

Install

**Python-Preview**

Provide Preview for Python Execution.

dongli

1.1M ★ 4.5

Install ⚠

**Python Extended**

Python Extended is a vscode snippet that makes it easy to write codes in pyt...

Taiwo Kareem

960K ★ 4

Install

**Python Path**

Python imports utils

661K ★ 4

1.5 Przydatne skróty klawiszowe?



AREPL for python

real-time python scratchpad
Almenon

🔗 621K ★ 5

Install



python snippets

Code snippets for python
Ferhat Yalçın

🔗 515K ★ 3

Install



Python Test Explorer for Visual Studio Code

🔗 781K ★ 4.5

Command Mode (press `Esc` to enable)

[Edit Shortcuts](#)

<code>F</code> : find and replace	<code>Shift-J</code> : extend selected cells below
<code>Ctrl-Shift-F</code> : open the command palette	<code>Ctrl-A</code> : select all cells
<code>Ctrl-Shift-P</code> : open the command palette	<code>A</code> : insert cell above
<code>Enter</code> : enter edit mode	<code>B</code> : insert cell below
<code>P</code> : open the command palette	<code>X</code> : cut selected cells
<code>Shift-Enter</code> : run cell, select below	<code>C</code> : copy selected cells
<code>Ctrl-Enter</code> : run selected cells	<code>Shift-V</code> : paste cells above
<code>Alt-Enter</code> : run cell and insert below	<code>V</code> : paste cells below
<code>Y</code> : change cell to code	<code>Z</code> : undo cell deletion
<code>M</code> : change cell to markdown	<code>D</code> , <code>D</code> : delete selected cells
<code>R</code> : change cell to raw	<code>Shift-M</code> : merge selected cells, or current cell with cell below if only one cell is selected
<code>1</code> : change cell to heading 1	<code>Ctrl-S</code> : Save and Checkpoint
<code>2</code> : change cell to heading 2	<code>S</code> : Save and Checkpoint
<code>3</code> : change cell to heading 3	<code>L</code> : toggle line numbers
<code>4</code> : change cell to heading 4	<code>O</code> : toggle output of selected cells
<code>5</code> : change cell to heading 5	<code>Shift-O</code> : toggle output scrolling of selected cells
<code>6</code> : change cell to heading 6	<code>H</code> : show keyboard shortcuts
<code>K</code> : select cell above	<code>I</code> , <code>I</code> : interrupt the kernel
<code>Up</code> : select cell above	<code>0</code> , <code>0</code> : restart the kernel (with dialog)
<code>Down</code> : select cell below	<code>Esc</code> : close the pager
<code>J</code> : select cell below	<code>Q</code> : close the pager
<code>Shift-K</code> : extend selected cells above	<code>Shift-L</code> : toggles line numbers in all cells, and persist the setting
<code>Shift-Up</code> : extend selected cells above	<code>Shift-Space</code> : scroll notebook up
<code>Shift-Down</code> : extend selected cells below	<code>Space</code> : scroll notebook down

Edit Mode (press `Enter` to enable)

<code>Tab</code> : code completion or indent	<code>Ctrl-Right</code> : go one word right
<code>Shift-Tab</code> : tooltip	<code>Ctrl-Backspace</code> : delete word before
<code>Ctrl-]</code> : indent	<code>Ctrl-Delete</code> : delete word after
<code>Ctrl-[</code> : dedent	<code>Ctrl-Y</code> : redo
<code>Ctrl-A</code> : select all	<code>Alt-U</code> : redo selection
<code>Ctrl-Z</code> : undo	<code>Ctrl-M</code> : enter command mode
<code>Ctrl-/</code> : comment	<code>Ctrl-Shift-F</code> : open the command palette
<code>Ctrl-D</code> : delete whole line	<code>Ctrl-Shift-P</code> : open the command palette
<code>Ctrl-U</code> : undo selection	<code>Esc</code> : enter command mode
<code>Insert</code> : toggle overwrite flag	<code>Shift-Enter</code> : run cell, select below
<code>Ctrl-Home</code> : go to cell start	<code>Ctrl-Enter</code> : run selected cells
<code>Ctrl-Up</code> : go to cell start	<code>Alt-Enter</code> : run cell and insert below
<code>Ctrl-End</code> : go to cell end	<code>Ctrl-Shift-Minus</code> : split cell at cursor(s)
<code>Ctrl-Down</code> : go to cell end	<code>Ctrl-S</code> : Save and Checkpoint
<code>Ctrl-Left</code> : go one word left	<code>Down</code> : move cursor down
	<code>Up</code> : move cursor up

1.6 Co to są biblioteki i jakie dobrze znać?

Biblioteka to zbiór kodu, który usprawnia codzienne zadania. Na przykład za pomocą Seaborn możesz generować wizualizacje za pomocą tylko jednego wiersza kodu zamiast pisania całej składni od podstaw.

Biblioteki są niezależnymi zbiorami które należy pobrać i zainstalować bezpośrednio w miejscu którym pracujemy.

Najbardziej popularne i użyteczne to:

TYP - Workflowy i pipeliney



Apache Airflow

apache-airflow

Biblioteka Apache-Airflow jest szeroko stosowanym programem do kolejowania, monitorania, wykonywania i zarządzania zadaniami, ładowaniami wsadowymi i orkiestracją potoków danych.

<https://airflow.apache.org/>

<https://pypi.org/project/apache-airflow/>



luigi

Podobnie jak Airflow, pozwala budować i definiować złożone potoki realizujące szereg zależności między zadaniami, zapewniając wykonanie zadań we właściwej kolejności podczas zarządzania awariami.

<https://github.com/spotify/luigi>

<https://pypi.org/project/luigi/>

perfect

Jak apache-airflow ale dedykowany do bardziej nowoczesnego stacku technologicznego.

<https://github.com/PrefectHQ/prefect/>

<https://pypi.org/project/prefect/>



kafka-python

Apache Kafka to popularna rozproszona platforma komunikacyjna używana do budowania potoków danych w czasie rzeczywistym i aplikacji do przesyłania strumieniowego, które przechowują dane i replikują je na wielu serwerach, zapewniając wysoką dostępność i trwałość w przypadku awarii serwera. Biblioteka Kafka-python zapewnia interfejs API wysokiego poziomu do tworzenia i wykorzystywania komunikatów z Apache Kafka, a także interfejsy API niższego poziomu do bardziej zaawansowanych zastosowań, takich jak przetwarzanie asynchroniczne, które ułatwia wysyłanie i odbieranie komunikatów bez blokowania głównego wątku wykonania.

<https://pypi.org/project/kafka-python>

<https://pypistats.org/packages/kafka-python>

TYP - Przetwarzanie danych



pandas

Pandas to jedna z najpopularniejszych bibliotek Pythona do pracy z małymi i średnimi zbiorami danych. Jest idealna do analizy i manipulacji danymi. Jest uważany za niezbędny ze względu na duży zbiór zaawansowanych funkcji, takich jak łączenie danych, obsługa brakujących danych, eksploracja danych i ogólna wydajność.

<https://pypi.org/project/pandas>

<https://pandas.pydata.org/>



pyarrow

Biblioteka PyArrow zapewnia API Pythona dla funkcjonalności zapewnianej przez biblioteki Arrow, wraz z narzędziami do integracji i współdziałania Arrow z pandas, NumPy i innym oprogramowaniem w ekosystemie Pythona. Dla inżynierów danych pyarrow zapewnia skalowalną bibliotekę do łatwej integracji danych z wielu źródeł w jeden, ujednolicony i duży zestaw danych w celu łatwej manipulacji i analizy.

<https://pypi.org/project/pyarrow>

<https://arrow.apache.org/>



numpy

NumPy to podstawowa biblioteka do obliczeń naukowych w Pythonie. Zapewnia obsługę tablic i macierzy, a także funkcje matematyczne do manipulowania nimi. Jest często używana do analizy numerycznej, nauki o danych i uczenia maszynowego.

<https://github.com/numpy/numpy>

<https://pypi.org/project/numpy/>



datetime

Moduł datetime dostarcza klas do manipulowania datami i godzinami.

<https://pypi.org/project/DateTime/>

<https://docs.python.org/3/library/datetime.html>

python-dateutil

Rozszerzenie dla datetime

<https://pypi.org/project/python-dateutil>

<https://github.com/dateutil/dateutil>

TYP - Współpraca z chmurami



Azure-core

Azure Core to biblioteka Pythona i interfejs API do interakcji z usługami w chmurze Azure i jest używany przez inżynierów danych do uzyskiwania dostępu do zasobów i automatyzacji zadań inżynierskich. Typowe zadania obejmują przesyłanie i monitorowanie zadań wsadowych, uzyskiwanie dostępu do baz danych, kontenerów danych i jezior danych oraz ogólne zarządzanie zasobami, takimi jak maszyny wirtualne i kontenery. Powiązaną biblioteką dla języka Python jest azure-storage-blob, biblioteka stworzona do zarządzania pobieraniem i przechowywania dużych ilości nieustrukturyzowanych danych, takich jak obrazy, audio, wideo lub tekst.

<https://pypi.org/project/azure-core>

<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/core/azure-core>



boto3

Boto3 to biblioteka Software Development Kit (SDK) dla programistów do pisania oprogramowania, które korzysta z długiej listy usług Amazon, w tym ulubionych inżynierów danych, takich jak Glue, EC2, RDS, S3, Kinesis, Redshift i Athena. Oprócz wykonywania typowych zadań, takich jak przesyłanie i pobieranie danych oraz uruchamianie i zarządzanie instancjami EC2, inżynierowie danych mogą wykorzystywać Boto3 do programowego uzyskiwania dostępu i zarządzania wieloma usługami AWS, których można używać do tworzenia potoków danych i automatyzacji zadań związanych z przepływem danych.

<https://pypi.org/project/boto3/>

<https://github.com/boto/boto3>



google-API-core

Biblioteka ta jest wspólna dla wszystkich interfejsów API chmury Google, takich jak uwierzytelnianie i autoryzacja, obsługa żądań i odpowiedzi klientów HTTP, ekstrakcja danych (Dysk Google itp.), transformacja danych i zarządzanie danymi. Dla inżynierów danych może być używany do uzyskiwania dostępu do danych z Google Cloud Storage lub BigQuery, opartej na chmurze platformy Google do przechowywania i analizy danych, lub interfejsów API uczenia maszynowego, takich jak Cloud ML Engine

<https://pypi.org/project/google-api-core>

<https://github.com/googleapis/python-api-core>

TYP - Big Data



pyspark

Apache Spark to jedna z najpopularniejszych platform inżynierii danych typu open source dzięki skalowalnej konstrukcji, która umożliwia szybkie przetwarzanie dużych ilości danych i sprawia, że idealnie nadaje się do zadań wymagających przetwarzania w czasie rzeczywistym lub analizy dużych zbiorów danych, w tym ETL, uczenia maszynowego i przetwarzanie strumieniowe. Można ją również łatwo zintegrować z innymi platformami, takimi jak Hadoop i inne platformy dużych zbiorów danych, ułatwiając inżynierom danych pracę z różnymi źródłami danych i technologiami. Biblioteka PySpark umożliwia inżynierom danych pracę z szeroką gamą źródeł i formatów danych, w tym danymi strukturalnymi, danymi nieustrukturyzowanymi i danymi przesyłanymi strumieniowo.

<https://pypi.org/project/pyspark>

<https://github.com/apache/spark/tree/master/python>



SQLAlchemy

SQLAlchemy to zestaw narzędzi Python SQL, który zapewnia interfejs wysokiego poziomu do interakcji z bazami danych. Umożliwia inżynierom danych wysyłanie zapytań do danych z bazy danych przy użyciu instrukcji podobnych do języka SQL oraz wykonywanie typowych operacji, takich jak wstawianie, aktualizowanie i usuwanie danych z bazy danych. SQLAlchemy zapewnia również obsługę mapowania obiektowo-relacyjnego (ORM), które pozwala inżynierom danych definiować strukturę ich tabel bazy danych jako klasy Pythona i mapować te klasy na rzeczywiste tabele bazy danych.

<https://pypi.org/project/SQLAlchemy>

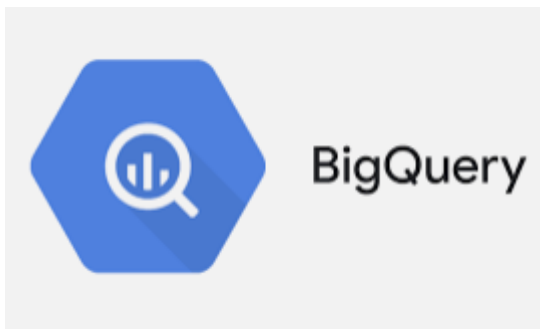


redis-py

Redis to popularny warehouse, szeroko stosowany w inżynierii danych ze względu na jego zdolność do skalowania i obsługi dużych ilości danych. Może być zainstalowany lokalnie lub jest już dostępny u głównych dostawców chmury. Redis-py to biblioteka Pythona, która umożliwia użytkownikom łączenie się z bazą danych Redis i wykonywanie różnych operacji, takich jak przechowywanie i pobieranie danych, transformacje danych i analiza danych. Redis-py może być również używany do automatyzacji zadań inżynierii danych, takich jak planowanie i integracja danych z innych źródeł, w tym wyodrębnianie danych z bazy danych lub interfejsu API i przechowywanie ich w Redis.

<https://pypi.org/project/redis>

<https://github.com/redis/redis-py>



google-cloud-bigquery

Odpowiednik Redis na platformie GCP.

<https://pypi.org/project/google-cloud-bigquery/>



dask

Dask został stworzony w celu zrównoleglenia NumPy na wielu procesorach, a teraz przekształcił się w bibliotekę ogólnego przeznaczenia do obliczeń równoległych, która obejmuje obsługę Pandas DataFrames oraz efektywne szkolenie modeli na XGBoost i scikit-learn. Inżynierowie danych dostosowali również Dask ze względu na jego wbudowane funkcje i możliwości przetwarzania równoległego, które sprawiają, że zadania związane z dużymi zbiorami danych, takie jak czyszczenie, transformacja, agregacja, analiza i eksploracja danych (obsługa Matplotlib i Seaborn) są bardziej wydajne i szybsze. Inżynierowie danych mogą również używać Dask do skalowania obciążeń za pośrednictwem rozproszonego harmonogramu, którego można używać do planowania zadań w klastrach.

<https://pypi.org/project/dask>

<https://github.com/dask/dask>



ray

W szczególności, podobnie jak Dask, Ray został zaprojektowany, aby ułatwić zrównoleglenie kodu Pythona i tworzenie rozproszonych aplikacji od podstaw. Ray nie próbuje zastąpić popularnych narzędzi do obsługi obciążeń w języku Python, ale raczej zapewnia ogólną platformę niskiego poziomu, która jest bardziej ogólną strukturą klastrową i równoległą, której można używać do tworzenia i uruchamiania dowolnego typu aplikacji rozproszonych. W związku z tym

rośnie liczba projektów, które integrują się z Rayem w celu wykorzystania akcelowanego procesora graficznego i obliczeń równoległych obok Dask, Ludwig, spaCy, Hugging Face i scikit-learn.

<https://pypi.org/project/ray>

<https://github.com/ray-project/ray>

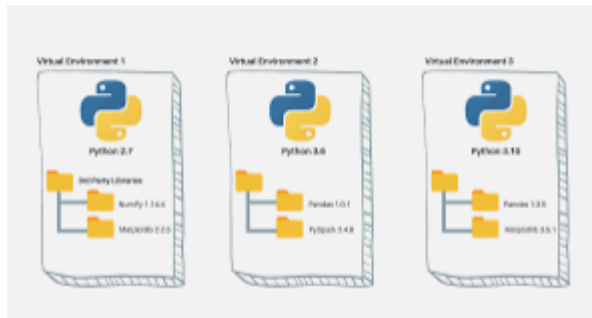
multiprocessing

Ekwiwalent dla dask.

<https://docs.python.org/3/library/multiprocessing.html>

<https://pypi.org/project/multiprocessing/>

TYP - Środowisko



virtualenv

Virtualenv to narzędzie do tworzenia oddzielnych środowisk Pythona, aby zapewnić brak ingerencji w różne konfiguracje systemu. Od Pythona 3.3 jego podzbiór został zintegrowany ze standardową biblioteką w ramach modułu venv. Virtualenv jest szczególnie ważny w przypadku projektów, które mają złożone zależności lub które muszą być uruchamiane w różnych wersjach Pythona.

<https://pypi.org/project/virtualenv>

<https://virtualenv.pypa.io/en/latest>

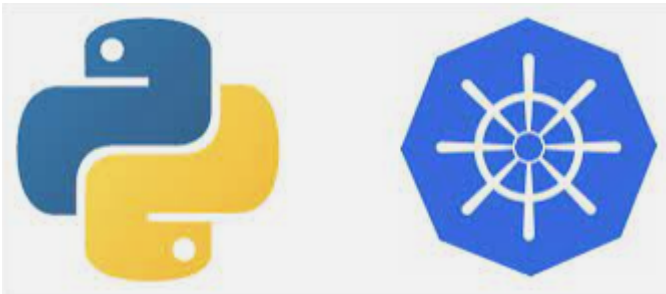


docker

Biblioteka współpracująca z Dockerem - narzędziem do konteneryzacji.

<https://pypi.org/project/docker>

<https://github.com/docker/docker-py>



kubernetes

Biblioteka współpracująca z Kubernetes - narzędziem do konteneryzacji.

<https://pypi.org/project/kubernetes>

<https://kubernetes.io>



Modin to biblioteka, która zapewnia zastępstwo dla Pandas, ale z większą wydajnością dla większych zestawów danych. Może automatycznie skalować swoje przetwarzanie na wiele rdzeni, a nawet na wiele maszyn.

<https://pypi.org/project/modin/>

<https://github.com/modin-project/modin>

TYP - Machine Learning



scikit-learn

Scikit-learn to moduł Pythona do uczenia maszynowego zbudowany na bazie SciPy i był prekursorem innych platform, takich jak PyTorch i Tensorflow. Jest to istotne dzisiaj w przypadku klasyfikacji, regresji i grupowania, a także narzędzi do wstępnego przetwarzania i inżynierii cech. Dzięki temu inżynierowie danych mogą szybko i łatwo tworzyć modele i potoki uczenia maszynowego.

<https://pypi.org/project/scikit-learn>

<https://scikit-learn.org/stable/>



tensorflow

TensorFlow to biblioteka oprogramowania typu open source do wysokowydajnych obliczeń numerycznych. Jego elastyczna architektura umożliwia łatwe wdrażanie obliczeń na różnych platformach (procesorach, procesorach graficznych, jednostkach TPU), od komputerów stacjonarnych po klastry serwerów i urządzenia mobilne i brzegowe.

<https://pypi.org/project/tensorflow>

<https://www.tensorflow.org>



keras

Keras to potężna i łatwa w użyciu bezpłatna biblioteka Pythona typu open source do opracowywania i oceny modeli deep-learning.

<https://pypi.org/project/keras>

<https://keras.io>



pytorch

Pomimo ogromnej adaptacji, TensorFlow oferował stromą krzywą uczenia się, a PyTorch został stworzony jako bardziej elastyczna i przyjazna dla użytkownika alternatywa dla innych ustalonych platform głębokiego uczenia się.

<https://pypi.org/project/torch>

<https://pytorch.org>

2.Przygotowanie środowiska (Windows)



2.1 Instalowanie Pythona

Kroki instalacji

1. Ściągamy pakiet instalacyjny bezpośrednio z oficjalnej strony python

<https://www.python.org/downloads/>

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Download the latest version for Windows

[Download Python 3.11.2](#)

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#),
[Docker images](#)

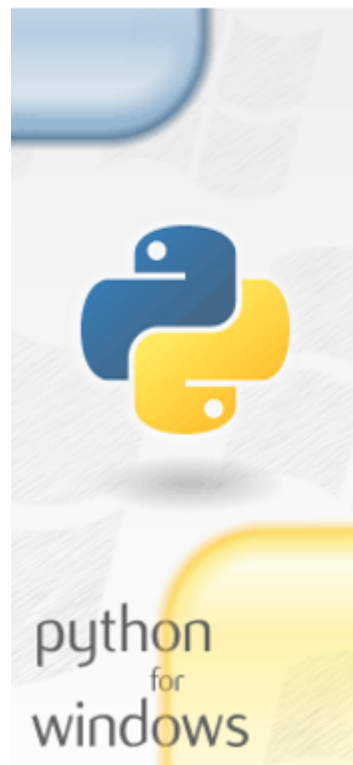


Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

2. Uruchamiamy pakiet instalacyjny i wybieramy indywidualną instalację wraz z wyborem dodania ścieżki do PATH



Install Python 3.10.9 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now

C:\Users\michal.zycki\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation

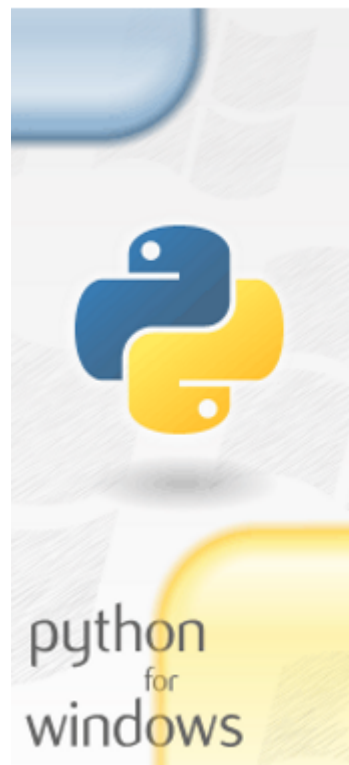
Choose location and features

☐ Use admin privileges when installing py.exe

☒ Add python.exe to PATH

Cancel

3. Zaznaczamy wszystkie opcjonalne funkcjonalności



Optional Features

☒ Documentation

Installs the Python documentation file.

☒ pip

Installs pip, which can download and install other Python packages.

☒ tcl/tk and IDLE

Installs tkinter and the IDLE development environment.

☒ Python test suite

Installs the standard library test suite.

☒ py launcher ☐ for all users (requires admin privileges)

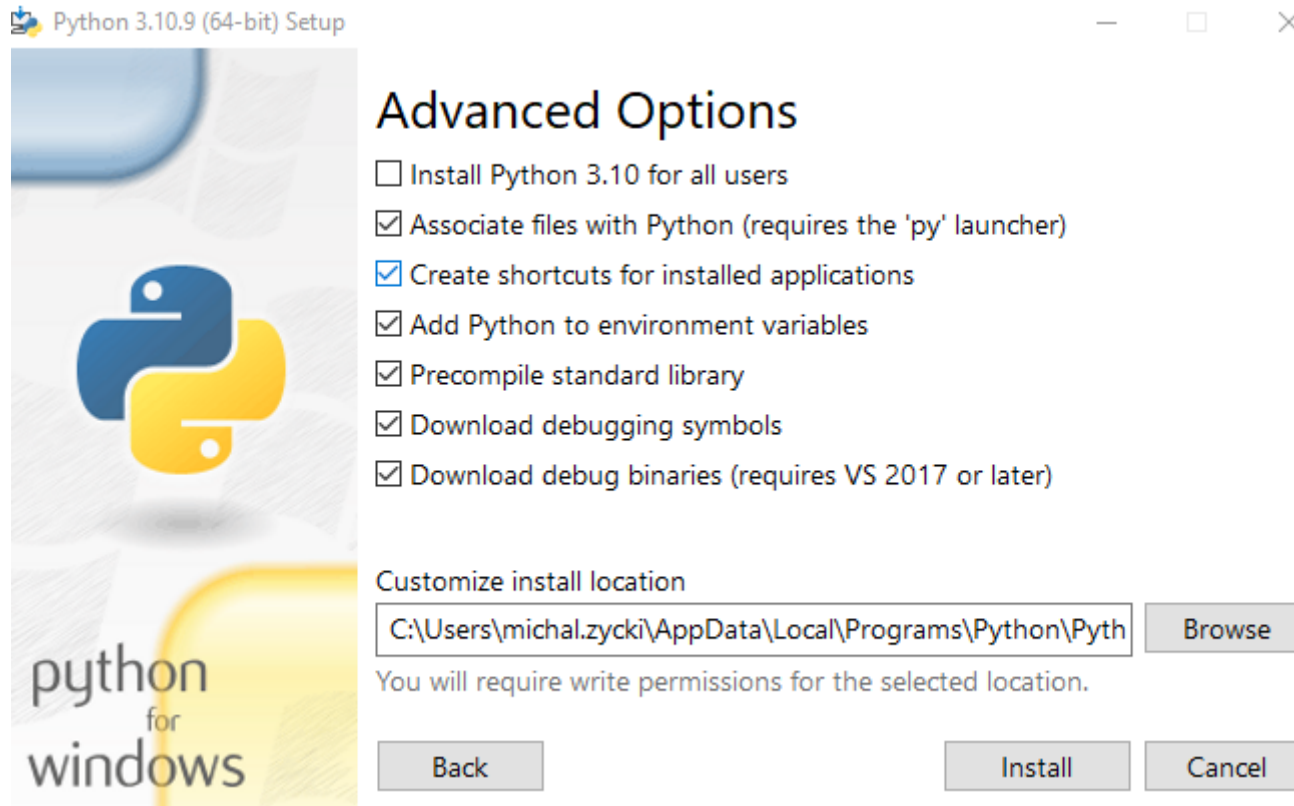
Use Programs and Features to remove the 'py' launcher.

Back

Next

Cancel

4. Instalujemy pythona w domyślnej ścieżce aplikacyjnej, zaznaczamy wszystkie dodatkowe opcje z wyjątkiem instalacji pythona dla wszystkich użytkowników



5. Finalnie instalujemy pythona

2.2 Przygotowanie środowiska

1. Tworzymy katalog/ folder w którym będziemy przechowywać pliki oraz instalować biblioteki

przykładowa ścieżka C:\BigData (unikamy raczej instalacji na partycji systemowej oraz katalogach użytkownika)

2. Otwieramy Command Prompt (cmd)

```
C:\> Developer Command Prompt for VS 2022

*****
** Visual Studio 2022 Developer Command Prompt v17.5.0
** Copyright (c) 2022 Microsoft Corporation
*****

C:\Program Files\Microsoft Visual Studio\2022\Enterprise>
```

3. Wybieramy wcześniej utworzoną przez nas lokalizację i wskazujemy ją przez komendę oraz potwierdzenie enterem

"cd C:\BigData"

```
C:\> Developer Command Prompt for VS 2022

*****
** Visual Studio 2022 Developer Command Prompt v17.5.0
** Copyright (c) 2022 Microsoft Corporation
*****

C:\Program Files\Microsoft Visual Studio\2022\Enterprise>cd C:\BigData_
```

Po zatwierdzeniu ścieżka powinna ulec zmianie na

```
C:\> Developer Command Prompt for VS 2022


*****
** Visual Studio 2022 Developer Command Prompt v17.5.0
** Copyright (c) 2022 Microsoft Corporation
*****

C:\Program Files\Microsoft Visual Studio\2022\Enterprise>cd C:\BigData
C:\BigData>
```

4. Tworzymy wirtualne środowisko przez komendę oraz potwierdzenie enterem

"python -m venv .vbigdata" (.vbigdata to przykładowa nazwa środowiska)

Po zatwierdzeniu w naszym katalogu powinien pojawić się folder

Nazwa	Data modyfikacji	Typ	Rozmiar
 .vbigdata	03.04.2023 14:42	Folder plików	

Natomiast Command Prompt będzie wyglądać w taki

```
C:\Program Files\Microsoft Visual Studio\2022\Enterprise>cd C:\BigData
C:\BigData>python -m venv .vbigdata
C:\BigData>
```

sposób

5. Będąc cały czas w naszej lokalizacji aktywujemy środowisko przez wpisanie komendy i potwierdzenie enterem

".vbigdata\Scripts\activate"

Po zatwierdzeniu środowisko zostało aktywowane a Command Prompt będzie wyglądał następująco

```
C:\Program Files\Microsoft Visual Studio\2022\Enterprise>cd C:\BigData
C:\BigData>python -m venv .vbigdata
C:\BigData>.vbigdata\Scripts\activate
(.vbigdata) C:\BigData>
```

2.3 Instalowanie bibliotek

1. Przystępujemy do instalacji kilku bibliotek oraz jupytera, cała instalacja odbywa się w tym samym Command Prompt który wykorzystywaliśmy wcześniej lub innym, ważne żebyśmy aktywowaliśmy wcześniej nasze środowisko. Instalacja odbywa się przez komendę oraz potwierdzenie enterem

przykładowo instalacja jupytera: "pip install jupyter"

Komputer domyślnie zacznie ściągać cały potrzebny zasób oraz instalować go co wyglądać będzie w podobny sposób

```
Developer Command Prompt for VS 2022
C:\BigData>.vbigdata\Scripts\activate

(.vbigdata) C:\BigData>pip install jupyter
Collecting jupyter
  Using cached jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting notebook
  Using cached notebook-6.5.3-py3-none-any.whl (529 kB)
Collecting qtconsole
  Downloading qtconsole-5.4.2-py3-none-any.whl (121 kB)
----- 121.2/121.2 kB 3.6 MB/s eta 0:00:00
Collecting jupyter-console
  Using cached jupyter_console-6.6.3-py3-none-any.whl (24 kB)
Collecting nbconvert
  Using cached nbconvert-7.2.10-py3-none-any.whl (275 kB)
Collecting ipykernel
  Downloading ipykernel-6.22.0-py3-none-any.whl (149 kB)
----- 150.0/150.0 kB 9.3 MB/s eta 0:00:00
Collecting ipywidgets
  Downloading ipywidgets-8.0.6-py3-none-any.whl (138 kB)
----- 138.3/138.3 kB 8.5 MB/s eta 0:00:00
Collecting comm>=0.1.1
  Downloading comm-0.1.3-py3-none-any.whl (6.6 kB)
Collecting debugpy>=1.6.5
  Using cached debugpy-1.6.6-py2.py3-none-any.whl (4.9 MB)
Collecting ipython>=7.23.1
  Downloading ipython-8.12.0-py3-none-any.whl (796 kB)
----- 796.4/796.4 kB 16.7 MB/s eta 0:00:00
Collecting jupyter-client>=6.1.12
  Downloading jupyter_client-8.1.0-py3-none-any.whl (102 kB)
----- 102.0/102.0 kB 2.0 MB/s eta 0:00:00
```

2. W następujący sposób instalujemy

- jupyter

- pandas

- numpy

Instalując bibliotekę pandas automatycznie zostanie ściągnięta biblioteka pomocnicza numpy, dlatego też podczas instalacji biblioteki numpy powinniśmy otrzymać komunikat że paczka jest już zainstalowana

```
(.vbigdata) C:\BigData>pip install pandas
Collecting pandas
  Using cached pandas-1.5.3-cp311-cp311-win_amd64.whl (10.3 MB)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\bigdata\.vbigdata\lib\site-packages (from pandas) (2.8.2)
Collecting pytz>=2020.1
  Downloading pytz-2023.3-py2.py3-none-any.whl (502 kB)
    ----- 502.3/502.3 kB 7.8 MB/s eta 0:00:00
Collecting numpy>=1.21.0
  Using cached numpy-1.24.2-cp311-cp311-win_amd64.whl (14.8 MB)
Requirement already satisfied: six>=1.5 in c:\bigdata\.vbigdata\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.24.2 pandas-1.5.3 pytz-2023.3
```

```
(.vbigdata) C:\BigData>pip install numpy
Requirement already satisfied: numpy in c:\bigdata\.vbigdata\lib\site-packages (1.24.2)
```

2.4 Aktywacja jupytera

1. Aby aktywować notebook wpisujemy w nasz otwarty w tym samym Command Prompt który wykorzystywaliśmy wcześniej lub innym, ważne żebyśmy aktywowali wcześniej nasze środowisko. Aktywacja odbywa się przez komendę oraz potwierdzenie enterem

"jupyter notebook"


```
C:\> Developer Command Prompt for VS 2022
-1.2.1 tornado-6.2 traitlets-5.9.0 uri-template-1.2.0 wcwidth-0.2.6 webcolors-1.13 webencodings-0.5.1 websocket-client-1
.5.1 widgetsnbextension-4.0.7

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(.vbigdata) C:\BigData>pip install pandas
Collecting pandas
  Using cached pandas-1.5.3-cp311-cp311-win_amd64.whl (10.3 MB)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\bigdata\.vbigdata\lib\site-packages (from pandas) (2.8.2)
Collecting pytz>=2020.1
  Downloading pytz-2023.3-py2.py3-none-any.whl (502 kB)
    ----- 502.3/502.3 kB 7.8 MB/s eta 0:00:00
Collecting numpy>=1.21.0
  Using cached numpy-1.24.2-cp311-cp311-win_amd64.whl (14.8 MB)
Requirement already satisfied: six>=1.5 in c:\bigdata\.vbigdata\lib\site-packages (from python-dateutil>=2.8.1->pandas)
(1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.24.2 pandas-1.5.3 pytz-2023.3

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(.vbigdata) C:\BigData>pip install numpy
Requirement already satisfied: numpy in c:\bigdata\.vbigdata\lib\site-packages (1.24.2)

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(.vbigdata) C:\BigData>jupyter notebook
```

2. Po aktywacji w przeglądarce automatycznie powinien pojawić się jupyter. Natomiast Command Prompt powinien wyglądać następująco

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾

☐ 0 ▾  /

Name ▾

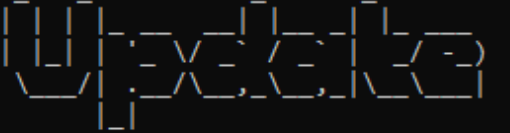
Last Modified

File size

The notebook list is empty.

```
[notice] A new release of pip available: 22.3.1 -> 23.0.1  
[notice] To update, run: python.exe -m pip install --upgrade pip
```

(.vbigdata) C:\BigData>jupyter notebook



Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.

https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

```
[I 15:06:02.195 NotebookApp] Serving notebooks from local directory: C:\BigData  
[I 15:06:02.196 NotebookApp] Jupyter Notebook 6.5.3 is running at:  
[I 15:06:02.196 NotebookApp] http://localhost:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2  
[I 15:06:02.196 NotebookApp] or http://127.0.0.1:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2  
[I 15:06:02.197 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
[C 15:06:02.235 NotebookApp]
```

To access the notebook, open this file in a browser:

file:///C:/Users/michal.zycki/AppData/Roaming/jupyter/runtime/nbsrvr-32396-open.html

Or copy and paste one of these URLs:

<http://localhost:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2>
or <http://127.0.0.1:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2>

```
0.00s - Debugger warning: It seems that frozen modules are being used, which may  
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off  
0.00s - to python to disable frozen modules.  
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
```

Jeśli jupyter nie uruchomi się automatycznie możemy skorzystać przez wklejenie linków z Command Prompt do przeglądarki które znajdują się pod "To access the notebook, open this file in a browser:" (przykładowe poniżej znajdują się w cmd)

file:///C:/Users/michal.zycki/AppData/Roaming/jupyter/runtime/nbserver-32396-open.html

<http://localhost:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2>

<http://127.0.0.1:8888/?token=bd33ff65410c6f48bb71dc943dfc9efda59b94abb50167a2>

