



**Faculty of Geodesy  
and Cartography**

WARSAW UNIVERSITY OF TECHNOLOGY

# INFORMATYKA GEODEZYJNA - WYKŁADY/ĆWICZENIA, ROK AKAD. 2021-2022

WYK. 14: SYSTEMY ZARZĄDZANIA BIBLIOTEKAMI I ŚRODOWISKIEM  
PROGRAMISTYCZNYM

---

Kinga Węzka  
[kinga.wezka@pw.edu.pl](mailto:kinga.wezka@pw.edu.pl)  
Zakład Geodezji i Astronomii Geodezyjnej

**Warsaw University  
of Technology**





1. System zarządzania pakietami: CONDA
  - Środowiska wirtualne – tworzenie, dostosowanie
  - Conda – podstawowe komendy
2. System zarządzania pakietami: PIP (ang. package installer for Python)
  - PIP – podstawowe komendy
3. Repozytoria pakietów: Anaconda repo i cloud PyPI
4. PIP versus CONDA



- **Conda to system zarządzania pakietami** – *opensource* – pozwala zarządzać pakietami, zależnościami i środowiskiem dla dowolnego języka – Python, R, Ruby, Lua, Java, JavaScript, C C ++, FORTRAN
- **Conda działa w systemach Windows, macOS i Linux** – instaluje, uruchamia i aktualizuje pakiety oraz ich zależności.
- Conda może bezpośrednio instalować pakiety Pythona, a także interpreter Pythona.
- **Conda jest również menedżerem środowisk pracy.** Za pomocą kilku poleceń można skonfigurować całkowicie oddzielne środowisko do uruchamiania różnych wersji Pythona.
- Oficjalna strona: <https://conda.io/projects/conda/en/latest/index.html>
- Dokumentacja: <https://conda.io/projects/conda/en/latest/user-guide>



- Nowe instalacje oraz ich testowanie zaleca się wykonywać w środowiskach wirtualnych:
  - tworzymy dystrybucje wyspecjalizowane do konkretnych zadań (konkretna wersja Python, konkretne pakiety i ich wersje),
  - zapobiegamy konfliktom między bibliotekami,
  - utrzymujemy porządek

- Tworzenie środowiska:

```
1 conda create --name myenv
```

```
1 conda create -n myenv python=3.6
```

- Aktywacja środowiska:

```
1 activate myenv
```

- Instalacja bibliotek w danym środowisku:

```
1 conda install -name myenv scipy
```

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>



<https://conda.io/docs/commands.html>

#### ■ Lista zainstalowanych bibliotek:

```
1 conda list
```

#### ■ Wyszukiwanie bibliotek:

```
1 anaconda search -t conda library_name
```

#### ■ Instalowanie bibliotek:

```
1 conda install nazwa_pakietu
```

#### ■ Instalowanie bibliotek z kanału (chanel) – lokalizacja:

```
1 conda install -c nazwa_kanal_u nazwa_pakietu
```

#### ■ Pomoc na temat komendy conda:

```
1 conda install -- help
```



- Jest menadżerem rekomendowanym przez PPA (ang. *Python Packaging Authority's* do instalowania pakietów z PYPI (ang. *Python Package Index*)
- Pip może zainstalować wszystko z repozytorium PYPI
- Oficjalna strona: <https://pip.pypa.io/en/stable/>
- Lista komend: [https://pip.pypa.io/en/stable/reference/pip\\_list/](https://pip.pypa.io/en/stable/reference/pip_list/)
- Dokumentacja: : <https://pypi.org/project/pip/>
- PIP nie tworzy środowisk wirtualnych, potrzebne jest dodatkowe narzędzie: *virtualenv*, *venv*



[https://pip.pypa.io/en/stable/reference/pip\\_list/](https://pip.pypa.io/en/stable/reference/pip_list/)

```
1 pip install SomePackage          # latest version
2 pip install SomePackage==1.0.4   # specific version
3 pip install 'SomePackage>=1.0.4' # minimum version
4
5 pip install -r requirements.txt   # lista plików wymaganych
```

Opcje wiersza poleceń pip można ustawić za pomocą zmiennych środowiskowych

```
1 pip install -vvv
```



## Repozytorium Anaconda

- Anaconda repository: <https://repo.anaconda.com/>
- Anaconda cloud: <https://anaconda.org/>, <https://anaconda.org/anaconda>

## Repozytorium PyPI (Python Package Index)

- <https://pypi.org/>





Conda i pip są często uważane za równorzędne narzędzia. Chociaż niektóre ich funkcje pokrywają się, zostały one zaprojektowane i powinny być używane do różnych celów.

- **Pip** to narzędzie zalecane przez **Python Packaging Authority** do instalowania pakietów z **Python Package Index, PyPI**. Pip instaluje oprogramowanie Python **spakowane jako wheel** lub **dystrybucje źródłowe** – może wymagać, aby system miał zainstalowane kompatybilne kompilatory i biblioteki, zanim wywołanie pip zakończy się powodzeniem.
- **Conda** jest narzędziem do pakowania i instalatorem, posiada więcej funkcjonalności niż pip; obsługuje zależności bibliotek poza pakietami Pythona, a także same pakiety Pythona, tworzy również wirtualne środowiska, tak jak **virtualenv**.
- Conda wprowadza nowy format pakowania (pip nie zainstaluje formatu pakietów conda). Można używać tych dwóch narzędzi obok siebie (instalując pip: **conda install pip**).
- **Conde** należy porównać do **Buildout**, innego narzędzia, które pozwala obsługiwać zadania instalacyjne zarówno w Pythonie, jak i poza Pythonem.
- **Pip** instaluje tylko pakiety Pythona, **conda** instaluje pakiety innych języków.
  - przed użyciem pip należy zainstalować interpreter języka Python;
  - conda bezpośrednio instaluje pakiety Pythona, a także interpreter Pythona.



- Kluczową różnicą między tymi dwoma narzędziami jest to, że conda ma możliwość tworzenia izolowanych/wirtualnych środowisk programistycznych:
  - Conda ma możliwość tworzenia środowisk z różnymi wersjami Pythona i/lub zainstalowanymi w nich pakietami.
  - Pip nie ma wbudowanej obsługi środowisk, ale polega raczej na innych narzędziach, takich jak **virtualenv** lub **venv**, aby tworzyć izolowane środowiska. Narzędzia takie jak **pipenv**, **poetry** i **virtualenv** zapewniają ujednoliconą metodę pracy z tymi środowiskami.
- Pip i conda różnią się również sposobem spełniania zależności i zgodności w środowisku:
  - pip instaluje zależności w szeregowej pętli rekurencyjnej (po kolei). Nie podejmuje żadnych działań, aby zapewnić, że zależności wszystkich pakietów są spełnione jednocześnie. Może to prowadzić do zepsucia środowisk, jeśli pakiety zainstalowane wcześniej w danej kolejności mają niezgodne wersje zależności w stosunku do pakietów zainstalowanych później.
  - conda używa 'SAT solvera' (satisfiability – SAT – solver), aby sprawdzić, czy spełnione są wszystkie wymagania wszystkich pakietów zainstalowanych w środowisku. To sprawdzenie może zająć więcej czasu, ale pomaga zapobiegać błędom w danym środowisku programistycznym. Dopóki metadane pakietu dotyczące zależności są poprawne, conda w przewidywalny sposób utworzy środowiska robocze.

Więcej: <https://www.anaconda.com/blog/understanding-conda-and-pip>



Funkcja	PIP	CONDA
Zarządzanie	wheel lub źródło	binarne (format condy)
Wymaga kompilatorów	może wymagać	nie
Pakiety języków	tylko Python	każdy
Tworzenie środowisk	nie ( <i>virtualenv</i> , <i>venv</i> )	tak
Sprawdzanie zależności	nie	tak
Źródło pakietów	PyPI: <a href="https://pypi.org/">pypi.org/</a>	Anaconda repo: <a href="https://repo.anaconda.com/">repo.anaconda.com/</a>
Dostępne pakiety	> 150000	> 1500



- `https://conda.io/projects/conda/en/latest/user-guide/getting-started.html`



Dziękuję za uwagę

Kinga Węzka [kinga.wezka@pw.edu.pl](mailto:kinga.wezka@pw.edu.pl)