

Python w używaniu Thymio

Czym jest Python?

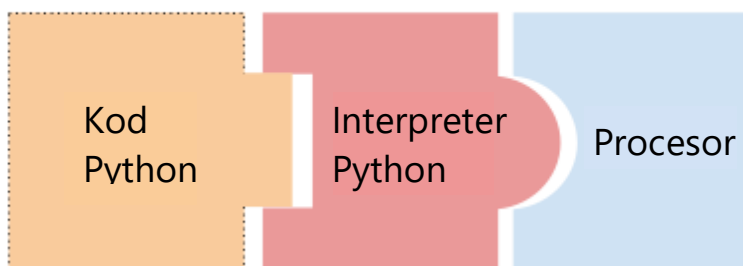
Najprawdopodobniej słyście o języku programowania Python. Cieszy się on bardzo dużą popularnością w dzisiejszych czasach. Python jest:

- **Językiem wysokiego poziomu** (co oznacza, że jest on zdecydowanie prostszy do zrozumienia przez ludzi, w porównaniu do języków niskiego poziomu).
- **Obiektowym** (czyli jest oparty na atrybutach) językiem programowania.

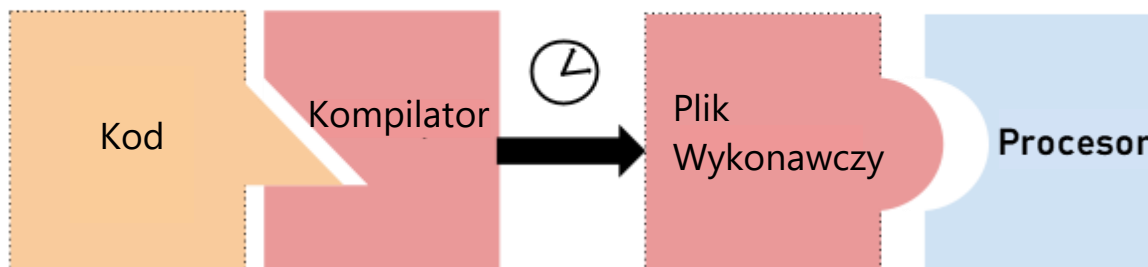
Jego składnia jest relatywnie prosta. Aktualnie jest szeroko stosowany w wielu domenach i posiada bardzo dużą społeczność z mnogą liczbą bibliotek.

Python jest także **językiem interpretowanym**, co oznacza, iż na komputerze zawsze musi być **interpreter**, który uruchamia kod bezpośrednio, bez kompilacji, w przeciwieństwie do języka takiego jak C++, w którym najpierw potrzebny jest **kompilator** do utworzenia pliku wykonywalnego, lecz ten plik wykonywalny można uruchomić nawet bez zainstalowanego kompilatora.

Język Interpretowany



Język Kompilowany



Python w korzystaniu z Thymio

W przypadku Thymio, **Robot nie posiada interpretera Python**. Thymio używa języka Aseba i posiada Wirtualną Maszynę Aseba (VM) o małej wielkości. Aby Thymio mógł uruchomić wgrany program, musi on być w kodzie bajtowym Aseba. Jeżeli program jest stworzony w VPL lub w Aseba, skrypty Aseba zostają skompilowane na kod bajtowy Aseba, przez Menadżer Urządzeń Thymio (TDM) i wysłany do robota.

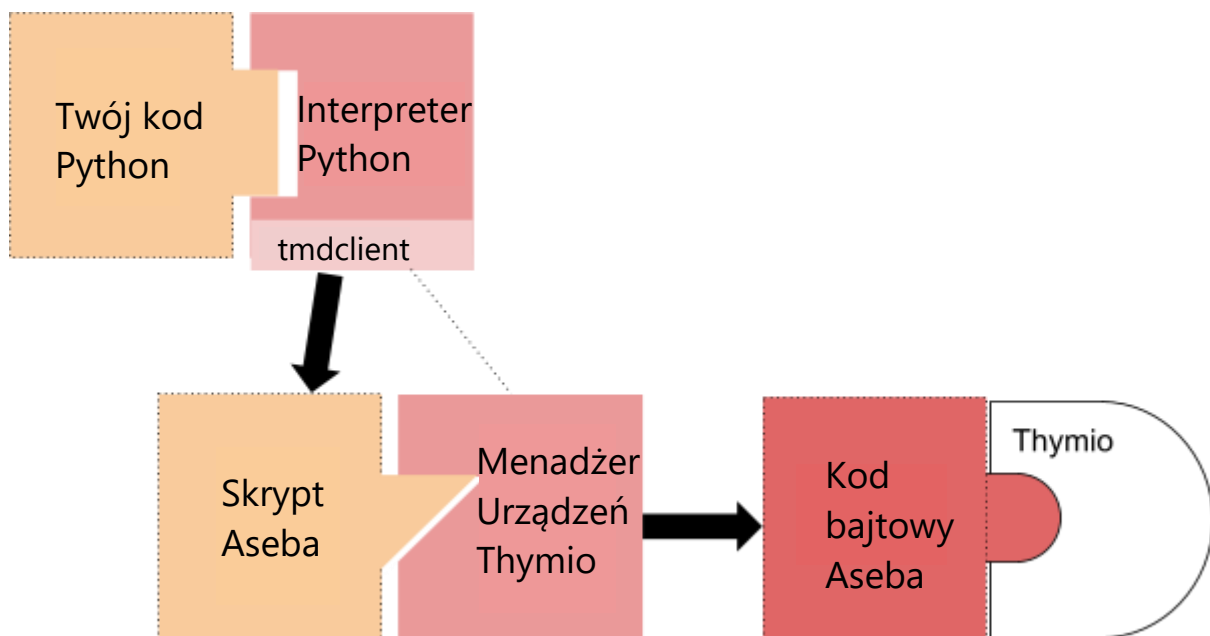
Ze względu na tą sytuację, został stworzony **moduł Python (tmdclient)**, który pozwala **komunikować się językowi Python z Menadżerem Urządzeń Thymio (TDM)**. Moduł pozwala na:

- Dostęp do zmiennych robota.
- Zamieniać kod Python na kod Aseba.
- Wysłać kod Aseba do TDM, co następnie wysyła odpowiedni kod bajtowy do robota.

Dzięki temu modułowi, nie jest wymagane uczenie się składni języka Aseba, do programowania Thymio. Istnieją polecenia Python, odpowiadające poleceniom API Aseba Thymio oraz inne dodatkowe funkcjonalności:

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

<https://github.com/epfl-mobots/tdm-python>



Notatniki Jupyter

Materiały i instruktaże, z kodem Python, które tutaj znajdziesz, są w formie Notatników Jupyter. Jupyter Lab jest aplikacją internetową, uruchamianą z poziomu przeglądarki i pozwala Ci tworzyć dokumenty zawierające różne rodzaje komórek, takich jak komórki Markdown, komórki kodu, które mogą zostać bezpośrednio wykonane itp. ([więcej informacji](#)).

W skrócie oznacza to, że zamiast korzystać z terminala w którym wpisujesz i uruchamiasz komendy Python, będziesz używał dokumentu, podzielonego na komórki, gdzie będziesz wpisywał kod, opis funkcji i wklejał obrazy.

Wymagania przedinstalacyjne i Instalacja

W tej sekcji znajdziesz informacje na temat wymagań i innych czynności, o które należy zadbać przed rozpoczęciem korzystania z Notatników Jupyter.

Thymio Suite

Instalacja Thymio Suite jest wymagana, ponieważ sterowniki dla Thymio i TDM są niezbędne dla tej aplikacji.

<https://www.thymio.org/program/>

Python 3

Aby wykonywać kod Python, na twoim komputerze musi się znajdować **interpreter**. Nie jest on wbudowany w Thymio Suite, więc należy zainstalować go oddzielnie.

Najnowszą wersję Python, możesz znaleźć tutaj (wymagana jest wersja **3.8** lub nowsza):

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

Ta instalacja zapewni Ci także system **pip**, który jest wymagany poniżej.

Rekomendowane jest zaznaczenie okienka **Add Python to PATH**, co pozwoli Ci na wpisywanie krótszych komand.



JupyterLab

JupyterLab pozwoli Ci otwierać Notatniki Jupyter w przeglądarce i zajmie się interpretacją kodu Python. Aby go zainstalować, użyjemy pip, narzędzia które zainstalowałeś wraz z Python. Otwórz terminal (w Windows wyszukaj program „cmd.exe”, w MacOS aplikacja nazywa się Terminal) i wpisz:

```
python3 -m pip install --upgrade pip
```

aby upewnić się, czy posiadasz aktualną wersję. W zależności od twojej instalacji, w komendzie może być „python3” lub „python”.

Invite de commandes - jupyter-lab

```
C:\Users\ >python3 -m pip install --upgrade pip
Requirement already satisfied: pip in c:\program files\windowsapps\pythonsoftwaref
bz5n2kfra8p0\lib\site-packages (21.1.1)
Collecting pip
Using pip 21.2.4 from https://pypi.org/whl/ (1.6 MB)
```

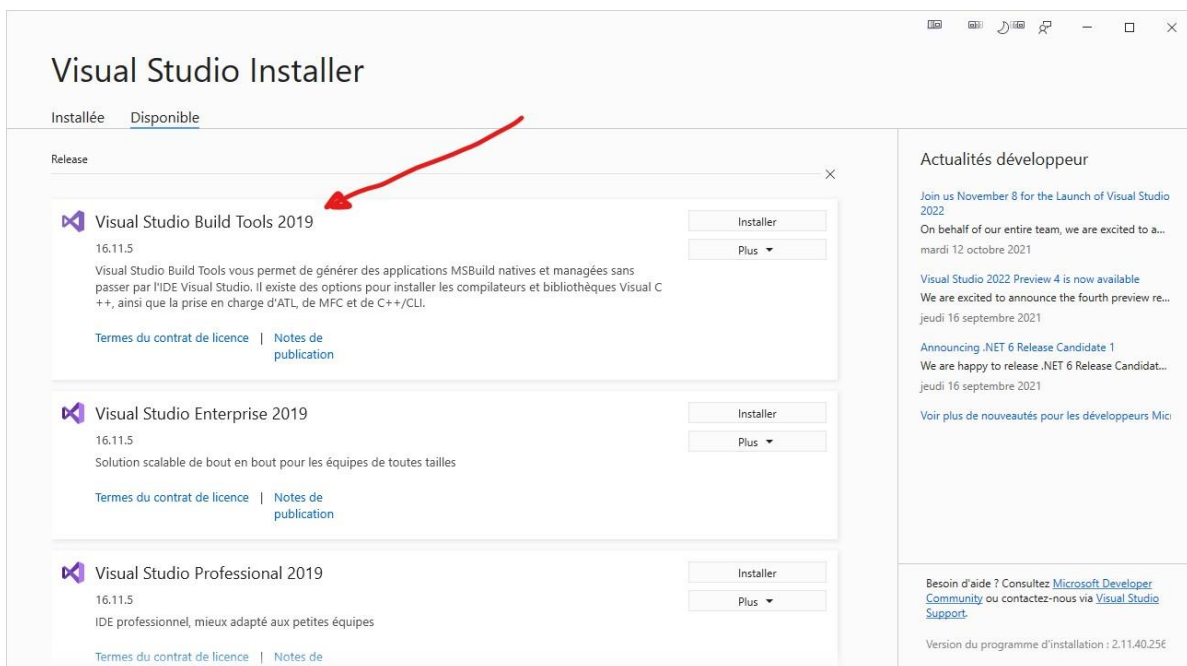
Następnie wpisz:

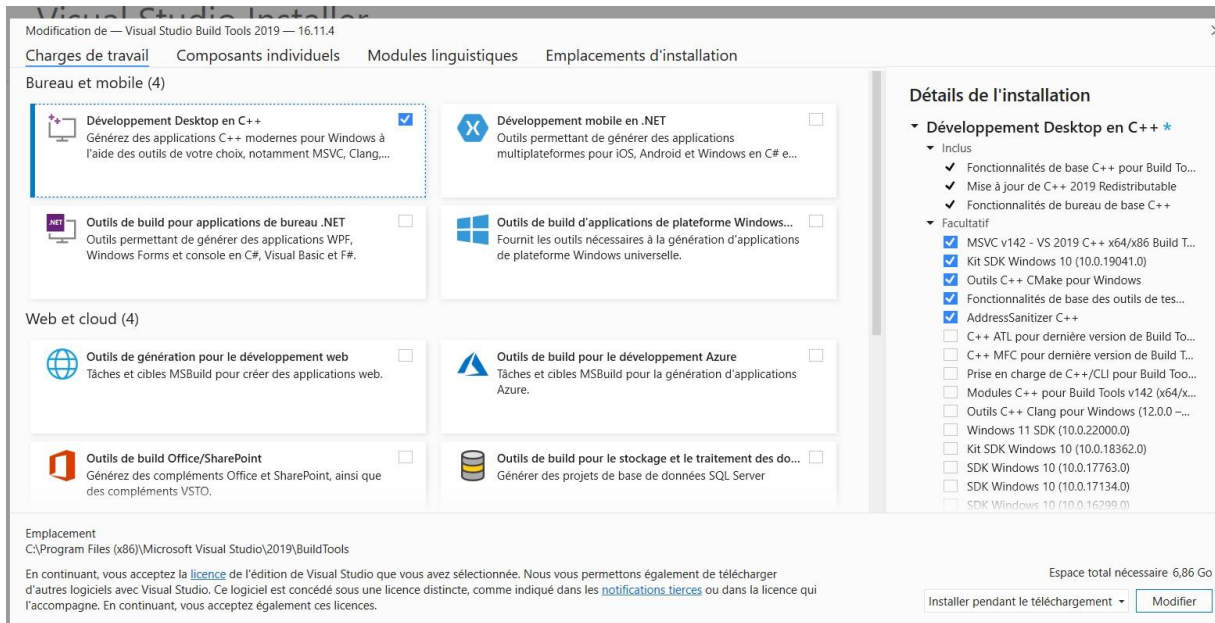
```
python3 -m pip install jupyterlab
```

Jeżeli w czasie instalacji pojawi się komunikat błędu: `error: Microsoft Visual C++ 14.0 is required.`

Należy pobrać i zainstalować następujący program:

<https://visualstudio.microsoft.com/fr/visual-cpp-build-tools/>





TDMclient

Aby zainstalować TMDclient, wejdź w terminal i wpisz (pamiętaj, że w zależności od twojego urządzenia, będziesz musiał użyć „python” zamiast „python3”):

```
python3 -m pip install tdmclient
```

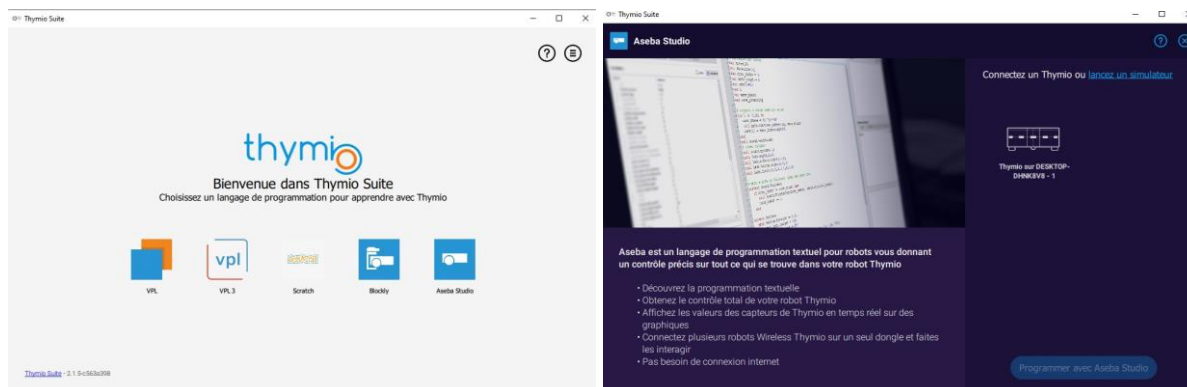
Nasze Notatniki

Przygotowaliśmy notatnik, który pomoże Ci się zapoznać z Python na Thymio (język angielski):

https://drive.google.com/file/d/1bJGHCR2mp_b47FweznHiBYKlexFXhz7A/view?usp=sharing

Pierwsze Kroki

1. **Podłącz swojego Thymio** do komputera.
2. **Uruchom Thymio Suite**. Pozostaw okno powitalne otworzone, aby upewnić się czy TDM jest uruchomiony. Jeżeli chcesz zobaczyć, które roboty są podłączone, możesz nacisnąć na jeden z języków programowania, aby wyświetlić listę.



Nie zamykaj Thymio Suite. Pozostaw go na ekranie powitalnym lub liście robotów.

3. **Otwórz Terminal i wpisz następującą komendę**, aby otworzyć Jupyter:

```
jupyter-lab
```

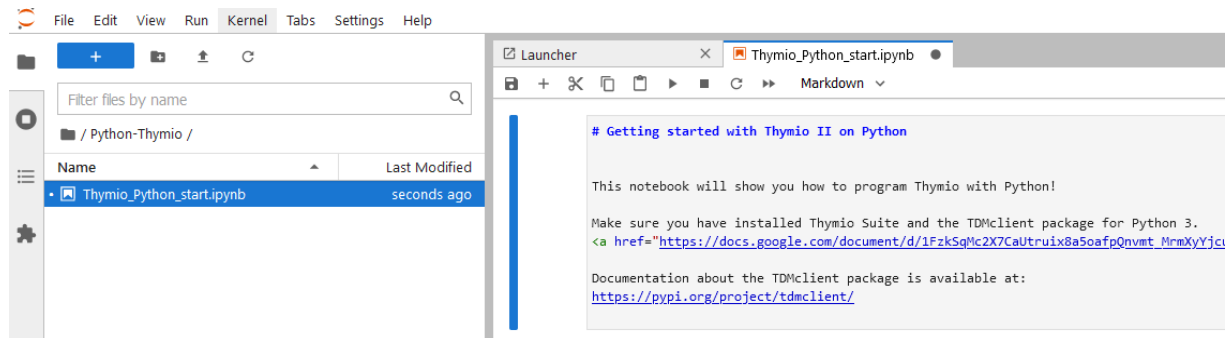
JupyterLab otworzy się w twojej przeglądarce.

4. Po lewej stronie możesz przeglądać swoje pliki. Odszukaj pobrany notatnik i kliknij dwa razy na *Thymio_Python_start.ipynb* aby go otworzyć.

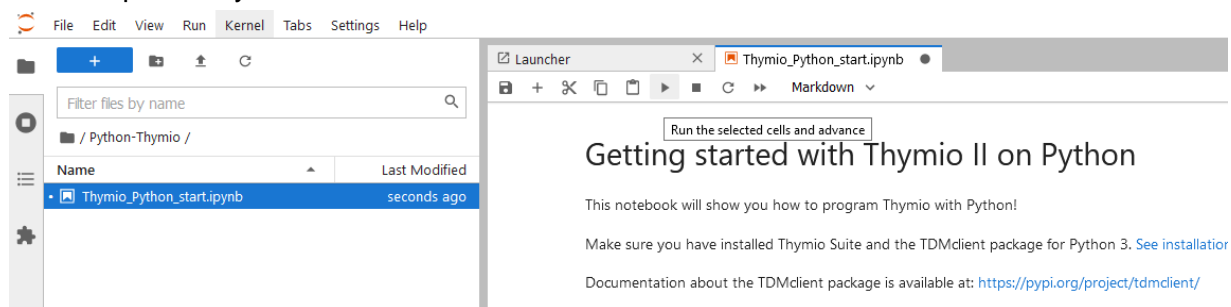
Na temat Jupyter Notebooks

Po uruchomieniu JupyterLab i otwarciu naszego notatnika, zobaczysz przed sobą strukturę tego programu.

Jeżeli dwukrotnie klikniesz na pierwszą komórkę z tekstem, zmieni się jej wygląd (zobaczysz kod źródłowy) i będziesz mógł ją edytować. Jest to komórka Markdown. Niektóre znaczniki, pozwolą Ci na formatowanie tekstu (na przykład: # aby wyznaczyć tytuły).



Jeżeli wykonasz kod tej komórki (przycisk odtwarzania na górze interfejsu lub Shift+Enter), zostanie pokazany rezultat kodu.



Inny typ komórek, z których będziemy korzystać, to komórki kodu. Zawierają one kod języka Python. Gdy je wykonasz, interpreter Python wykona zawarte w nich komendy.

```
[ ]: my_measure = prox_horizontal[2]
     print(my_measure)
```

Czasami zobaczysz wynik wykonania kodu. Jeżeli komórka jest aktualnie wykonywana, zobaczysz * w nawiasie obok niej. Po zakończeniu tego procesu, pojawi się tam liczba.

```
[2]: my_measure = prox_horizontal[2]
     print(my_measure)
0
```

Wyniki wykonanego kodu, są zachowywane pomiędzy komórkami. Jeżeli nadasz wartość zmiennej, i użyjesz ją później w innej komórce, wartość będzie zapamiętana.



Naciskając ten przycisk, możesz ponownie uruchomić jądro aby zacząć od początku.