

Tryb interaktywny, uzyskiwanie pomocy, pisanie aplikacji

Teraz, gdy masz zainstalowaną Anacondę i VS Code, możesz zacząć głębiej pisać kod Pythona. Tutaj pokrótce omówimy funkcje interaktywne, pomocy i edycji kodu VS Code i Jupyter Notebook, aby wykorzystać to, czego się nauczyłeś. Większość z was prawdopodobnie chce rozpocząć bardziej zaawansowane tematy, takie jak analiza danych, sztuczna inteligencja, robotyka itp. Ale nauka będzie łatwiejsza, jeśli dobrze zrozumiesz wiele dostępnych narzędzi i umiejętności korzystania z nich.

Korzystanie z trybu interaktywnego Python

Wielu nauczycieli i autorów zasugeruje, abyś wypróbował rzeczy od razu po pytaniu w Pythonie i zakłada, że już wiesz, jak się tam dostać. Widzieliśmy wielu sfrustrowanych początkujących narzekających, że próby aktywności zalecane w niektórych samouczkach nigdy dla nich nie działają. Frustracja często wynika z faktu, że wpisują i wykonują kod w niewłaściwym miejscu. W Anaconda terminal w VS Code jest doskonałym miejscem do pisania kodu Python. Tu zaczniemy od tego.

Otwarcie terminalu

Aby używać Pythona interaktywnie z Anacondą, zalecamy wykonanie następujących kroków:

1. Otwórz Anaconda Navigator, a następnie otwórz VS Code, klikając przycisk Uruchom na stronie głównej Anaconda.
2. Jeśli nie widzisz panelu Terminal u dołu okna VS Code, wybierz View ⇌ Terminal z paska menu VS Code.
3. Jeśli słowa Terminal nie są podświetlone w górnej części okienka, kliknij Terminal



Pierwszy monit, jaki widzisz, zwykle dotyczy systemu operacyjnego komputera i prawdopodobnie pokazuje nazwę użytkownika konta, na którym się znajdujesz. Na przykład na komputerze Mac może to wyglądać jak Alans-Air: ~ alan \$, ale z nazwą twój komputer zamiast Alans-Air. W systemie Windows prawdopodobnie będzie to C: \ Users \ Alan > z nazwą użytkownika zamiast Alan i być może ścieżką inną niż C: \ Users.

Uzyskiwanie wersji Python

W wierszu polecenia systemu operacyjnego możesz to wpisać i nacisnąć Enter, aby zobaczyć, jakiej wersji Pythona używasz. Zauważ, że przed pierwszym łącznikiem jest spacja i nie ma innych spacji.

```
python --version
```

Powinieneś zobaczyć coś takiego jak Python 3.xx (gdzie x to liczby reprezentujące używaną wersję Pythona). Jeśli zamiast tego zobaczysz komunikat o błędzie, nie jesteś dokładnie tam, gdzie musisz być. Chcesz się upewnić, że uruchom kod VS z poziomu Anacondy, nie tylko z Launchpada lub z menu Start. Wpisz `python --wersja` w panelu Terminal kodu VS i naciśnij ponownie Enter. Jeśli nadal nie działa,

wyberz View ⇨ Command Palette z kodu VS pasek menu, wpisz python, wybierz Python: wybierz Interpreter, a następnie wybierz interpreter Pythona pobrany za pomocą Anacondy.

KOLORY I IKONY W KODZIE VS

Domyślnie terminal VS Code wyświetla biały tekst na czarnym tle. Odwrócimy te kolory, tylko dlatego, że uważamy, że ciemność pod światło lepiej sprawdza się w przypadku takiego druku. Możesz użyć dowolnego schematu kolorów, który Ci się podoba. Jeśli chcesz po prostu zmienić kolor na czarno-biały, jak pokazano w tej książce, użyj jednej z następujących metod:

Na komputerze Mac wybierz Code ⇨ Preferences ⇨ Theme ⇨ Light (Visual Studio).

W systemie Windows wybierz File ⇨ Preferences ⇨ Color Theme ⇨ Light (Visual Studio).

Jeśli chcesz, aby Twoje ikony w kodzie VS były zgodne z tymi, których używamy, musisz pobrać i zainstalować Material Icon Theme. Możesz także pobrać motyw Kolor materiału i wypróbować go. Nie będziemy go używać, ponieważ nie gra się dobrze po wydrukowaniu na papierze. Ale możesz spróbować. Wykonaj następujące kroki:

1. Kliknij Extensions w lewym panelu.
2. Wpisz material, poszukaj Material Icon Theme i kliknij opcję Install.
3. Kliknij Reload ponownie na dowolnym wybranym rozszerzeniu, aby zainstalować oba rozszerzenia. Jeśli w prawym dolnym rogu zobaczysz monit z pytaniem, czy chcesz aktywować ikony, kliknij Activate.
4. Wybierz File (w systemie Windows) lub Code (na komputerze Mac), a następnie Preferences ⇨ File Icon Theme, a następnie kliknij Material Icon Theme.
5. Jeśli chcesz wypróbować motyw kolorów, otwórz File (w systemie Windows) lub Code (na komputerze Mac), a następnie wybierz Preferences ⇨ Color Theme i kliknij Material Icon Theme..

Jeśli w dowolnym momencie zmienisz zdanie na temat motywu kolorów, powtórz krok 5 powyżej i wybierz coś innego niż Material Icon Theme.

Przechodzenie do interpretera języka Python

Gdy możesz wpisać python --wersja i nie pojawia się błąd, wiesz, że jesteś gotowy do pracy z Pythonem w VS Code. Stamtąd możesz dostać się do interpretera Pythona, wprowadzając polecenie

```
python
```

Kiedy my lub ktokolwiek inny powie „wprowadź polecenie”, oznacza to, że musisz wpisać polecenie, a następnie nacisnąć klawisz Enter. Nic się nie dzieje, dopóki nie naciśniesz Enter. Więc jeśli po prostu wpiszesz polecenie i zaczekasz, aż coś się wydarzy, będziesz czekał bardzo długo. Powinieneś zobaczyć trochę informacji o używanej wersji Pythona oraz monit >>>, który reprezentuje interpreter Pythona.

Wprowadzanie poleceń

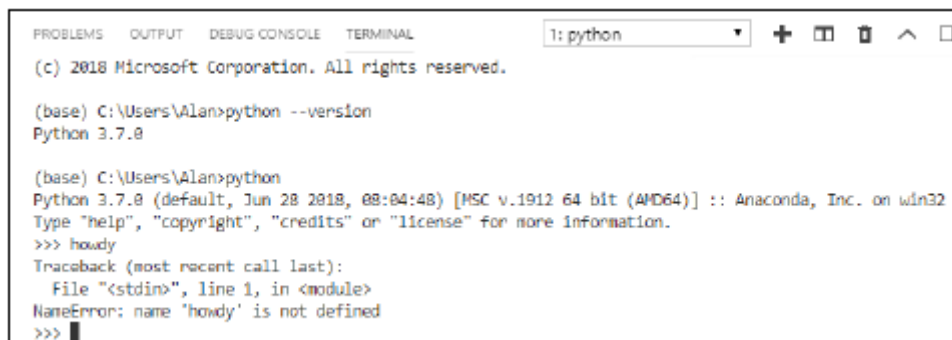
Wprowadzanie poleceń w interpretatorze Pythona jest takie samo, jak pisanie w dowolnym innym miejscu. Musisz poprawnie wpisać polecenie, a następnie nacisnąć klawisz Enter. Jeśli przeliterujesz coś złego w poleceniu, prawdopodobnie zobaczysz komunikat o błędzie, który jest tylko tłumaczem mówiącym, że nie rozumie, co masz na myśli. Ale nie martw się, nic nie możesz złamać. Na przykład, jeśli wprowadzisz polecenie

howdy

UWAGA O PyLint

PyLint to funkcja Anacondy, która pomaga znaleźć i uniknąć błędów w kodzie. Zwykle jest domyślnie włączony. Chociaż w przeszłości otrzymywaliśmy różne wyniki z różnymi wersjami VS Code. Możliwe, że przy pierwszej próbie użycia Pythona zobaczysz niektóre komunikaty w prawym dolnym rogu VS Code. Nie przejmuj się, jeśli ich nie widzisz. Jeśli jednak tak jest, oto jak możesz odpowiedzieć: Jeśli zobaczysz komunikat o Python Language Server, kliknij Wypróbuj teraz, a następnie kliknij Załaduj ponownie. Jeśli zobaczysz komunikat, że Linter PyLint nie jest zainstalowany, kliknij Zainstaluj. Jeśli zobaczysz Wybierz środowisko Python w lewym dolnym rogu okna VS Code, kliknij to i wybierz opcję Anakonda z menu rozwijanego w pobliżu górnego środka. Jeśli widzisz wiele opcji Anakondy, wybierz tę bez największego numeru wersji

Po naciśnięciu klawisza Enter na ekranie pojawia się błąd techniczny, który próbuje powiedzieć, że nie wie, co to znaczy „howdy”, więc nie może tego zrobić. Ale znowu nic się nie zepsuło. Właśnie wróciłeś do innego >>> monitu, w którym możesz spróbować ponownie, jak pokazano na rysunku

A screenshot of a terminal window with a dark background. The window title bar shows '1: python'. The terminal output is as follows:

```
(c) 2018 Microsoft Corporation. All rights reserved.

(base) C:\Users\Alan>python --version
Python 3.7.0

(base) C:\Users\Alan>python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> howdy
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'howdy' is not defined
>>>
```

Korzystanie z wbudowanej pomocy Pythona

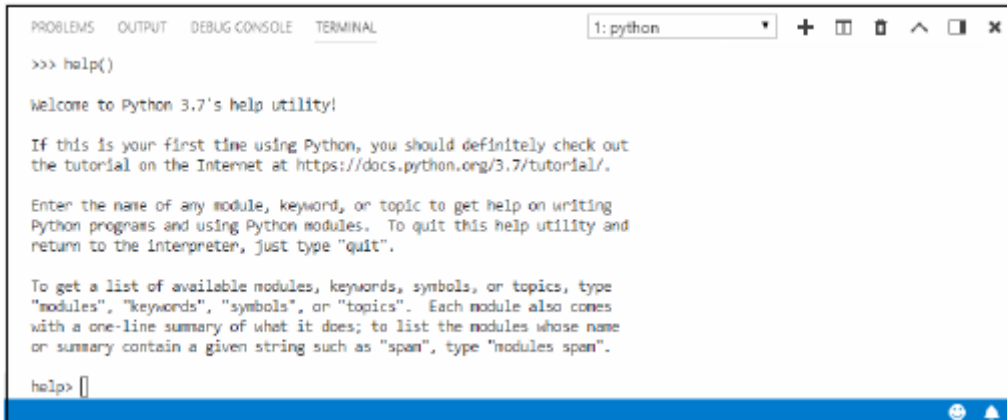
W jednej z podpowiedzi na ekranie wspomniano, że można wpisać help jako komentarz w interpretatorze Pythona. Pamiętaj, że nie wpisujesz cudzysłowu, a jedynie słowo pomoc (a następnie jak zwykle naciśnij Enter). Tym razem zobaczysz

Type help() for interactive help, or help(object) for help about object.

Zauważ, że tym razem każą Ci wpisać help, po którym następuje pusta para nawiasów, lub pomoc z konkretnym słowem w nawiasach (podany jest przykład). Mimo że na początku zdania używają słowa „typ”, chcą wprowadzić polecenie. . . wpisz go i naciśnij klawisz Enter. Śmiało i wprowadź

help()

Zauważ, że w linii nie ma spacji. Po naciśnięciu klawisza Enter ekran zawiera informacje o korzystaniu z interaktywnej pomocy Pythona, jak pokazano na rysunku



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:python + [ ] ^ [ ] x
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".

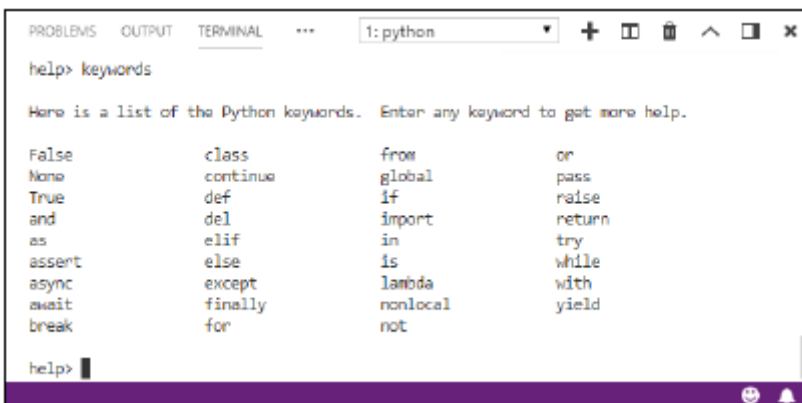
To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "span", type "modules span".

help> [ ]
```

Widok help> w dolnej części okna informuje, że nie jesteś już w powłoce systemu operacyjnego lub interpreterze Pythona (który zawsze pokazuje >>>), ale znajduje się teraz w nowym obszarze zapewniającym pomoc. Wszelkie wpisywane tutaj polecenia powinny być rozpoznawane przez pomoc. Jako początkujący prawdopodobnie ich nie znasz. Ale do przyszłych odniesień wiedz, że możesz użyć tego monitu pomocy, aby przypomnieć o różnych częściach Pythona. Na przykład Python używa określonych słów kluczowych, które mają specjalne znaczenie w języku. Otrzymasz ich listę, po prostu wpisując

keywords

po znaku help> i naciśnięciu Enter, a zobaczysz listę słów kluczowych, jak poniżej.



```
PROBLEMS OUTPUT TERMINAL ... 1:python + [ ] ^ [ ] x
help> keywords

Here is a list of the Python keywords. Enter any keyword to get more help.

False      class      from       or
None       continue  global     pass
True       def        if         raise
and        del        import     return
as         elif       in         try
assert     else      is         while
async      except    lambda    with
await     finally  nonlocal  yield
break     for       not

help> [ ]
```

Powyżej listy słów kluczowych znajduje się informacja, że możesz wpisać dowolne słowo kluczowe w help>, aby uzyskać więcej informacji o tym słowie kluczowym. Na przykład wprowadzenie słowa kluczowego class dostarcza informacji o klasach języka Python, jak pokazano na rysunku

```
PROBLEMS OUTPUT TERMINAL ... 1:python + [ ] [ ] ^ [ ] x
Class definitions
*****

A class definition defines a class object (see section The standard
type hierarchy):

    classdef ::= [decorators] "class" classname [inheritance] ":" suite
    inheritance ::= "(" [argument_list] ")"
    classname ::= identifier

A class definition is an executable statement. The inheritance list
usually gives a list of base classes (see Metaclasses for more
advanced uses), so each item in the list should evaluate to a class
object which allows subclassing. Classes without an inheritance list
-- More --
```

Nie są to zajęcia, do których uczęszczasz w szkole; raczej są to te, które tworzysz w Pythonie (po opanowaniu podstaw i przygotowaniu się do przejścia na bardziej zaawansowane tematy). Nie trzeba dodawać, że cały techniczny żargon w tekście pomocy sprawi, że przeciętny początkujący będzie całkowicie oszołomiony. Ale chodzi tutaj o to, że do przyszłych odniesień, kiedy uczysz się różnych rzeczy w Pythonie, możesz użyć interaktywnej pomocy Pythona do przypomnienia tych pojęć w razie potrzeby

--More-- na dole tekstu nie jest pytaniem, w którym wpisujesz polecenia. Zamiast tego informuje Cię, że będzie więcej tekstu. Naciśnij spację. Może być kilka stron informacji. Za każdym razem, gdy widzisz - Więcej -, możesz nacisnąć klawisz Enter, aby przejść do następnej strony. W końcu wrócisz do monitu `help>` i wtedy wiesz, że doszedłeś do końca tej pomocy.

Wyjście z interaktywnej pomocy

Aby wyjść z interaktywnej pomocy i powrócić do pytania w Pythonie, wpisz literę `q` (aby wyjść) lub naciśnij `Ctrl + Z`, a następnie naciśnij `Enter`. Powinieneś powrócić do monitu `>>>`. W wierszu `>>>` możesz wpisać `exit()` lub `python`. Aby wyjść z monitu Python i wrócić do systemu operacyjnego, wpisz polecenie `exit()` i naciśnij klawisz `Enter`. Pamiętaj, że jeśli popełnisz błąd, na przykład pomijając nawiasy, otrzymasz pomoc na ekranie. Na przykład, jeśli wprowadzisz `exit` i naciśniesz `Enter`, zobaczysz Użyj `exit()` lub `Ctrl-Z` plus `Return`, aby wyjść. To informuje, że aby wyjść z wiersza polecenia Python, należy wpisać `exit()` (w nawiasach, bez spacji) lub nacisnąć `Ctrl + Z`, a następnie nacisnąć `Enter`. Będziesz wiedział, że opuściłeś interpreter Pythona, gdy zobaczysz monit systemu operacyjnego zamiast `>>>` na końcu okna terminala, jak na rysunku.

```
PROBLEMS OUTPUT TERMINAL ... 1:powershell + [ ] [ ] ^ [ ] x
See also: **PEP 3115** - Metaclasses in Python 3 **PEP 3129** -
Class Decorators

Related help topics: CLASSES, SPECIALMETHODS

help> q

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)". Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>> exit
Use exit() or Ctrl-Z plus Return to exit
>>> exit()
PS C:\Users\Alan>
```

Wyszukiwanie określonych tematów pomocy online

Wbudowana pomoc Pythona jest nieco archaiczna, ale może ci pomóc, gdy potrzebujesz szybkiego przypomnienia o zapomnianym słowie kluczowym Python. Ale jeśli jesteś online, prawdopodobnie lepiej po prostu poszukać w sieci pomocy. Możesz zacząć od strony <https://www.youtube.com/>, jeśli konkretnie szukasz filmów, a jeśli nie, <https://stackoverflow.com/> to dobre miejsce do zadawania pytań i szukania pomocy. I oczywiście zawsze są Google, Bing i inne wyszukiwarki. Niezależnie od tego, czego używasz do wyszukiwania, pamiętaj, aby rozpocząć wyszukiwanie słowem python lub python 3. Wiele języków programowania ma podobne pojęcia i słowa kluczowe, więc jeśli nie określisz języka Python w zapytaniu, nie mówiąc, jakie wyniki możesz uzyskać.

Wiele darmowych ścią

Innym dobrym źródłem informacji dla uczniów są niezliczone ściąawki dostępne online za darmo. Za każdym razem, gdy zaczynasz czuć się przytłoczony wszystkimi możliwościami języka takiego jak Python, ściąawka podsumowująca wszystko na jednej stronie, może naprawdę pomóc sprowadzić rzeczy do łatwiejszego do zarządzania (i mniej zastraszającego) rozmiaru. Oczywiście tak naprawdę nie „oszukujesz” za pomocą ściąawki, chyba że nam to zrobisz podczas testu, na który masz odpowiedzieć z pamięci. Ale pisanie kodu w prawdziwym życiu różni się znacznie od odpowiadania na pytania wielokrotnego wyboru. To, co często nazywamy ściąawką w świecie technologii, jest tak naprawdę kolejnym narzędziem, które pomoże ci się uczyć. Jest ich wiele, a to, co ci się podoba, zależy od twojego stylu uczenia się. Aby zobaczyć, co jest dostępne, po prostu przejdź do Google lub Bing lub dowolną wyszukiwarkę, która Ci się podoba i wyszukaj darmową ściąawkę Pythona 3. Większość z nich jest w formacie, który można pobrać, wydrukować i mieć pod ręką, gdy poznasz nieskończone możliwości pisania kodu w Pythonie.

Tworzenie obszaru roboczego programowania Python

Chociaż tryby interaktywne i pomoc online oraz reszta są z pewnością przyzwoitymi narzędziami wsparcia, większość ludzi chce używać Pythona do tworzenia aplikacji. Osobiście uznaliśmy to za najłatwiejsze, jeśli skonfigurujesz środowisko programistyczne VS Code specjalnie do nauki i programowania w języku Python. Możesz skonfigurować inne środowiska programistyczne dla innych rodzajów kodowania, takich jak HTML, CSS i JavaScript dla Internetu, dostrajanie w miarę upływu czasu, aby uzyskać najlepszą obsługę niezależnie od języka, w którym pracujesz. Często przełączamy się między komputerami Mac i Windows, a więc mamy dla każdego jedno środowisko programistyczne. Ale jeśli będziesz pracować ściśle z jednego komputera, możesz umieścić swoje środowisko na dysku twardym komputera, a nie na dysku w chmurze. W VS Code używają terminu workspace do zdefiniowania tego, co nazywamy środowiskiem programistycznym. Jest to w zasadzie konkretny interpreter języka Python, którego używasz, a także wszelkie dodatkowe rozszerzenia, które gromadzisz, aby ułatwić naukę i wykonywanie zadań. Aby ułatwić dostęp do tych obszarów roboczych z dowolnego komputera podłączonego do Internetu na świecie mamy folder na dysku w chmurze (OneDrive) o nazwie VS Code Workspaces. Możesz jednak przechowywać swoje miejsca pracy w dowolnym miejscu - nawet na własnym komputerze, jeśli nie masz lub nie chcesz korzystać z dysku w chmurze. Ale jeśli chcesz utworzyć folder do przechowywania obszarów roboczych, zrób to teraz, zanim przejdziesz do kolejnych kroków. Następnie

1. Jeśli kod VS nie jest jeszcze otwarty, otwórz go w Anaconda.
2. Choose File ⇔ Save Workspace As..
3. Przejdź do folderu, w którym chcesz zapisać ustawienia obszaru roboczego.

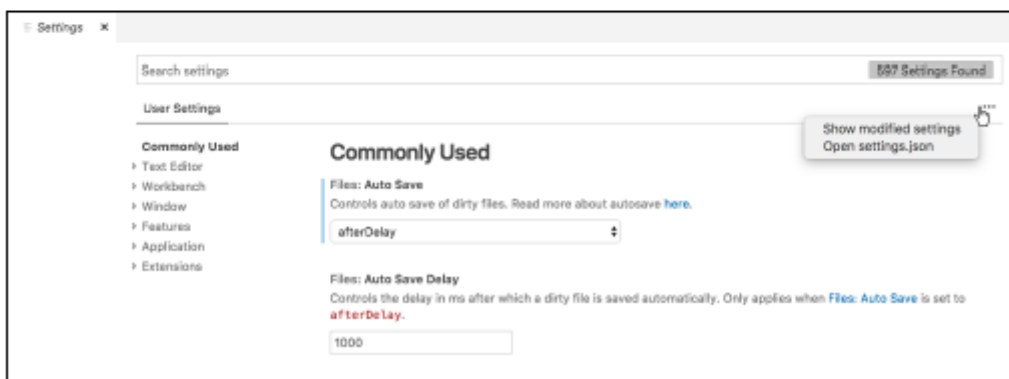
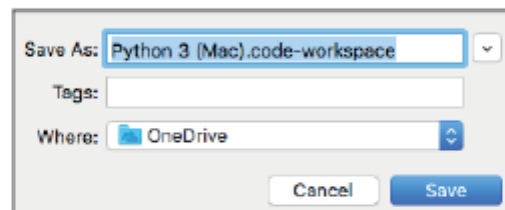
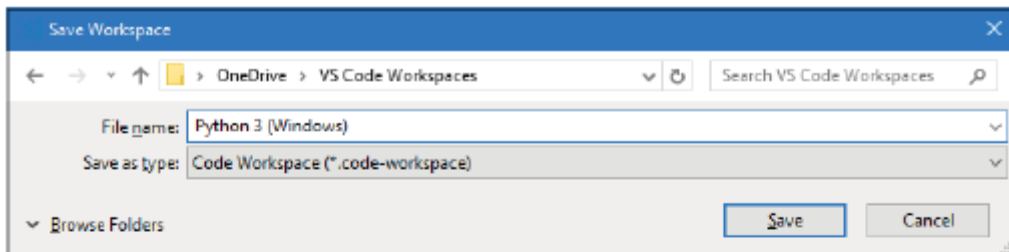
4. Wpisz nazwę obszaru roboczego. Używamy Pythona 3, a następnie Maca lub Windowsa, w zależności od typu komputera, na którym jest on włączony, jak pokazano na rysunku poniżej.

- Na komputerze Mac wybierz Code ⇌ Preferences ⇌ Settings.
- W systemie Windows wybierz File ⇌ Preferences ⇌ Settings.

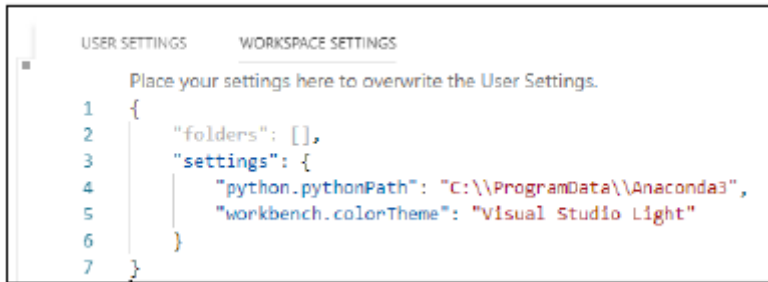
5. Jeśli zobaczysz stronę podobną do tej z rysunku, kliknij trzy kropki w prawym górnym rogu i wybierz Open Settings.json, jak pokazano na rysunku kolejnym.

6. W następnym oknie wybierz cały wiersz kodu rozpoczynający się od Pythona. pythonpath (informuje VS Code, gdzie znaleźć interpreter Pythona na twoim komputerze). Możesz również wybrać dowolne inne wiersze poleceń, które chcesz włączyć do obszaru roboczego, ale nie wybieraj nawiasów klamrowych.

7. Kliknij Ustawienia obszaru roboczego nad właśnie wybranym kodem.



8. Kliknij między nawiasami klamrowymi ustawienia i wklej tam wiersze kodu, jak pokazano na rysunku. Pamiętaj, że twoja ścieżka do Pythona może nie wyglądać tak, jak na obrazku

A screenshot of the VS Code settings editor showing the 'WORKSPACE SETTINGS' tab. The text 'Place your settings here to overwrite the User Settings.' is at the top. Below it, a JSON configuration is shown with line numbers 1 through 7 on the left. The configuration includes an empty 'folders' array and a 'settings' object with 'python.pythonPath' set to 'C:\\ProgramData\\Anaconda3\\' and 'workbench.colorTheme' set to 'Visual Studio Light'.

```
1  {
2    "folders": [],
3    "settings": {
4      "python.pythonPath": "C:\\ProgramData\\Anaconda3\\",
5      "workbench.colorTheme": "Visual Studio Light"
6    }
7  }
```

9. Wybierz File ⇨ Save z menu Kod VS.

10. Zamknij karty Ustawienia i Ustawienia użytkownika, klikając X po prawej stronie każdej karty.

11. Zamknij kod VS.

12. Zamknij Anakondę.

Za chwilę zobaczysz, jak skorzystać z nowych ustawień przestrzeni roboczej. Ale najpierw warto utworzyć katalog (folder) do przechowywania całego kodu, który napiszesz w tej książce, aby łatwo było go znaleźć, kiedy chcesz przejrzeć.

Tworzenie folderu dla twojego kodu Python

Następnie tworzymy folder, w którym można przechowywać cały kod Pythona, który zapiszesz, dzięki czemu wszystko jest razem w jednym miejscu i łatwo je znaleźć, gdy tego potrzebujesz. Możesz umieścić ten folder w dowolnym miejscu. Alan ponownie użyje dysku w chmurze (OneDrive), aby mógł się do niego dostać z dowolnego komputera. Ale możesz umieścić swoje w dowolnym miejscu. Nazwie swój folder AIO Python, ale możesz nazwać swój, jak chcesz. Kroki są takie same, jak w przypadku każdego innego folderu, nie ma w nim nic specjalnego. W systemie Windows możesz przejść do folderu, który będzie zawierał nowy folder. Na komputerze Mac kliknij prawym przyciskiem myszy puste miejsce w folderze i wybierz Nowy folder. W systemie Windows kliknij prawym przyciskiem myszy puste miejsce w folderze i wybierz Nowy ⇨ Folder. Wpisz nazwę folderu i naciśnij klawisz Enter. Na koniec chcesz powiązać utworzony folder kodu z właśnie utworzonym obszarem roboczym VS Code, aby za każdym razem, gdy pracujesz w folderze AIO Python, używałś poprawnego interpretera Python i innych ustawień związanych z Pythonem wybierz z czasem pliki w folderze kodu. Oto jak:

1. Otwórz Anacondę i uruchom stamtąd kod VS.

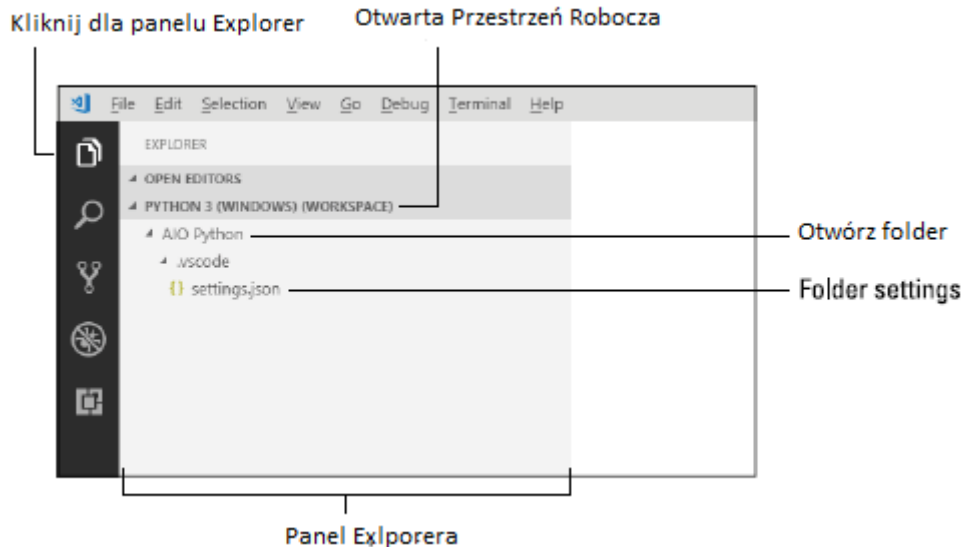
2. Z menu VS Code wybierz File ⇨ Open Workspace.

3. Przejdź do folderu, w którym zapisałeś obszar roboczy i stamtąd otwórz obszar roboczy.

4. Wybierz File ⇨ Add Folder do obszaru roboczego.

5. Przejdź do folderu, w którym utworzono folder AIO Python, kliknij ikonę tego folderu i wybierz opcję Dodaj.

Pasek eksploratora w kodzie VS pokazuje, że masz otwarty obszar roboczy, a pod nim widać, że masz również otwarty folder, jak pokazano na rysunku. Jeśli zobaczysz coś zupełnie innego w lewym okienku, kliknij ikonę Eksploratora w lewym górnym rogu (obok strzałki na rysunku), aby upewnić się, że przeglądasz to okienko.



Pasek Otwartych edytorów zawiera listę aktualnie otwartych plików (po prawej stronie). Na przykład jeśli widzisz tam Witamy, oznacza to, że przeglądasz stronę powitalną. Aby zamknąć tę stronę, kliknij X obok jej nazwy w okienku lub karcie Eksploratora. Za każdym razem, gdy chcesz ponownie otworzyć tę stronę powitalną, wybierz Help ⇌ Welcome from the VS Code menu bar. Ikona .vscode to tylko podfolder, który przechowuje twoje ustawienia dla tego obszaru roboczego, a settings.json to plik w tym folderze, który zawiera te ustawienia. Trójkąt obok każdej nazwy pozwala rozwinąć lub zwinąć tę listę. Więc jeśli coś jest ukryte, kliknij trójkąt obok elementu, aby go rozwinąć i zobaczyć, co zawiera. Jeśli obok nazw folderów widzisz symbol inny niż trójkąt lub w ogóle go nie ma, być może używasz motywu ikony innego niż domyślny. Nie martw się, po prostu kliknij po lewej stronie dowolnego folderu, aby go rozwinąć / zwinąć, nawet jeśli nie ma w nim żadnego symbolu. Zaletą tego podejścia jest to, że za każdym razem, gdy chcesz pracować z Pythonem w VS Code, wszystko co musisz zrobić, to wykonać następujące kroki:

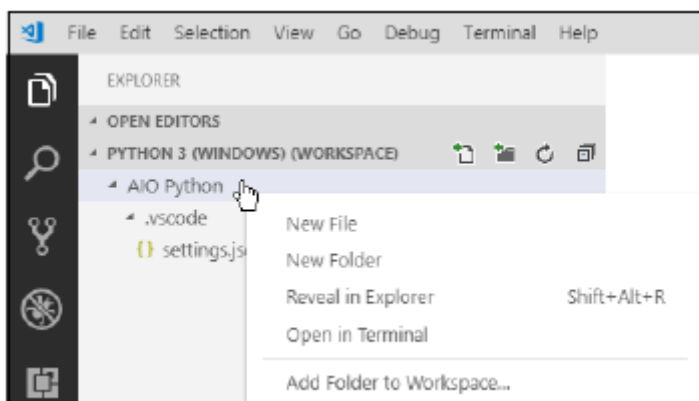
1. Jeśli zamknąłeś VS Code, uruchom go z Anaconda Navigator.
2. Wybierz File ⇌ Open Workspace z menu Kod VS.
3. Otwórz swój obszar roboczy.

Obszar roboczy i wszystkie foldery powiązane z tym obszarem roboczym otworzą się i jesteś gotowy do pracy.

Wpisywanie, edycja i debugowanie kodu w języku Python

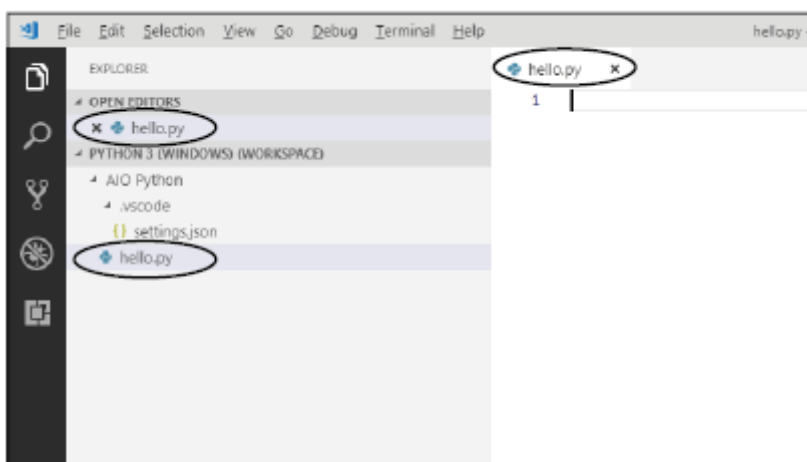
Najprawdopodobniej zdecydowana większość kodu, który napiszesz, będziesz pisać i redagować. Będzie to zwykły plik tekstowy z rozszerzeniem .py. Sugerujemy, aby przechowywać wszystkie pliki tworzone w tym folderze AIO Python, które powinny być widoczne za każdym razem, gdy masz VS Code i otwarty obszar roboczy Python 3. Aby w dowolnym momencie utworzyć plik .py, wykonaj następujące kroki:

1. Jeśli jeszcze tego nie zrobiłeś, otwórz VS Code w swoim obszarze roboczym Python 3.
2. Jeśli okienko Eksploratora nie jest otwarte, kliknij ikonę Eksploratora w lewym górnym rogu VS Code.
3. Aby utworzyć nowy plik w folderze AIO Python, kliknij prawym przyciskiem myszy AIO Python i wybierz Nowy plik.



4. Wpisz nazwę pliku z rozszerzeniem .py (hello.py dla pierwszego) i naciśnij ENTER

Nowy plik zostanie otwarty i jego nazwa będzie widoczna u góry karty po prawej stronie. W tym większym obszarze należy wpisać kod w języku Python. Zobaczysz także hello.py w sekcji Otwarte edytory. To tylko wygoda, więc gdy masz wiele otwartych plików, możesz wybrać jeden z nich, aby wysunąć go na pierwszy plan, klikając jego nazwę. Nazwa pliku pojawia się również pod nazwą folderu Pythona AUO w okienku Eksploratora, ponieważ tam jest przechowywana, jak pokazano na rysunku



Pisanie kodu w języku Python

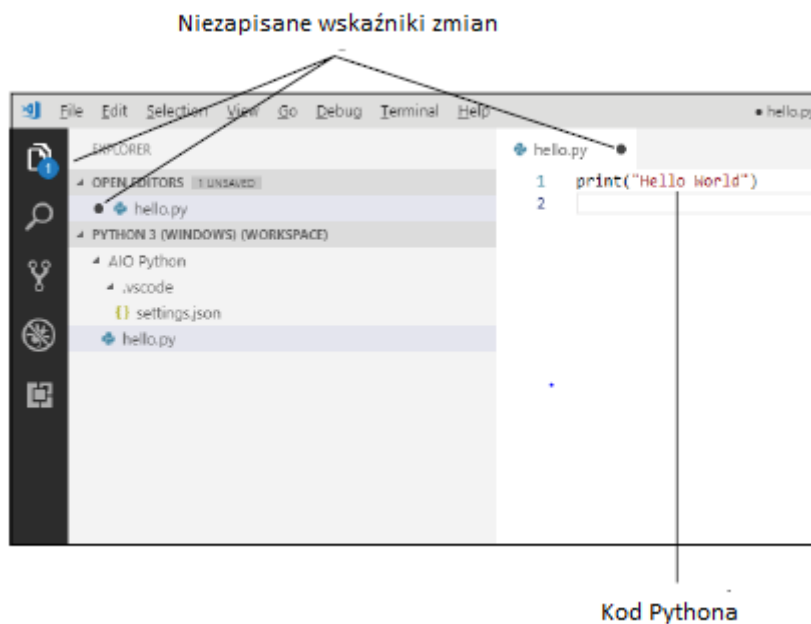
Teraz, gdy masz otwarty plik py, możesz go użyć do napisania kodu w języku Python. Jak to jest typowe podczas nauki nowego języka programowania, zaczniesz od wpisania prostego programu Hello World. Oto kroki:

1. Kliknij obok 1 w obszarze edycji.
2. Wpisz (dokładnie) `print(„Hello World”)`, a podczas pisania możesz zauważyć tekst pojawiający się na ekranie. To jest tekst intellisense, który wykrywa to, co piszesz i wyświetla informacje o tym słowie kluczowym. Nie musisz jednak nic z tym robić, po prostu pisz dalej.
3. Naciśnij klawisz Enter po wpisaniu linii.

Nowa linia kodu pokazuje, ale nie wykonuje się. Jest tak, ponieważ zazwyczaj nie wpisujesz i nie uruchamiasz jednego wiersza kodu naraz. Możesz również zauważyć kilka innych zmian, jak pokazano na rysunku :

* Ikona Eksploratora pokazuje kółko 1, wskazując, że masz jedną niezapisaną zmianę.

* Nazwa hello.py w zakładce i w obszarze Otwórz edytory pokazuje kropkę, wskazującą, że w pliku nie zapisano zmian.

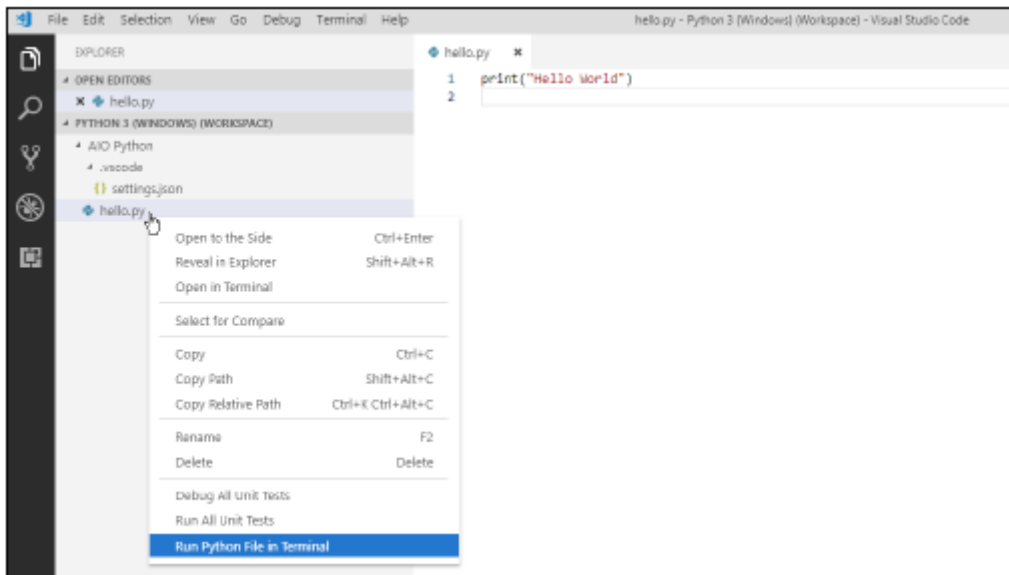


Zapisywanie kodu

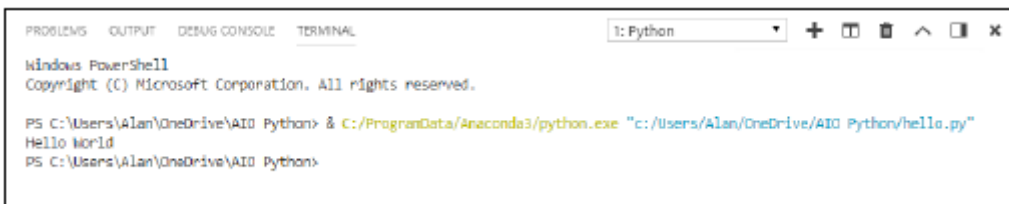
Kod wpisany w VS Code jest zapisywany automatycznie. Jest na to kilka sposobów. Jednym z nich jest próba zapamiętania, aby oszczędzać za każdym razem, gdy wprowadzasz zmiany, które warto zapisać. Najłatwiej to zrobić, wybierając File ⇨ Save z paska menu VS Code lub naciśnij Ctrl + S w systemie Windows lub Command + S na komputerze Mac. Osobiście wolimy korzystać z funkcji AutoSave, która automatycznie zapisuje wprowadzone zmiany, dzięki czemu nie musisz pamiętać. Aby włączyć automatyczne zapisywanie, wystarczy wybrać File ⇨ AutoSave z paska menu VS Code. Gdy zobaczysz znacznik wyboru obok Auto Save w menu File, oznacza to, że Auto Save jest włączony, więc nie masz pamiętać, aby zapisać każdą zmianę. Jeśli zdecydujesz, że nie chcesz już używać Auto Save w przyszłości, po prostu wybierz File ⇨ Auto Save ponownie z paska menu, aby usunąć zaznaczenie i wyłączyć Auto Save.

Uruchamianie Pythona w VS Code

Aby przetestować kod Python w VS Code, musisz go uruchomić. Najłatwiejszym sposobem na to w VS Code jest prawdopodobnie kliknięcie prawym przyciskiem myszy nazwy pliku (hello.py w tym przykładzie) i wybranie Uruchom plik Python w terminalu, jak pokazano na rysunku



Panel Terminal otwiera się u dołu okna VS Code. Zobaczysz wiersz polecenia, a następnie komentarz, aby uruchomić kod w interpreterze języka Python (python.exe). A poniżej zobaczysz wynik programu: słowa Hello World, w tym przykładzie, a następnie kolejny monit, jak na rysunku.



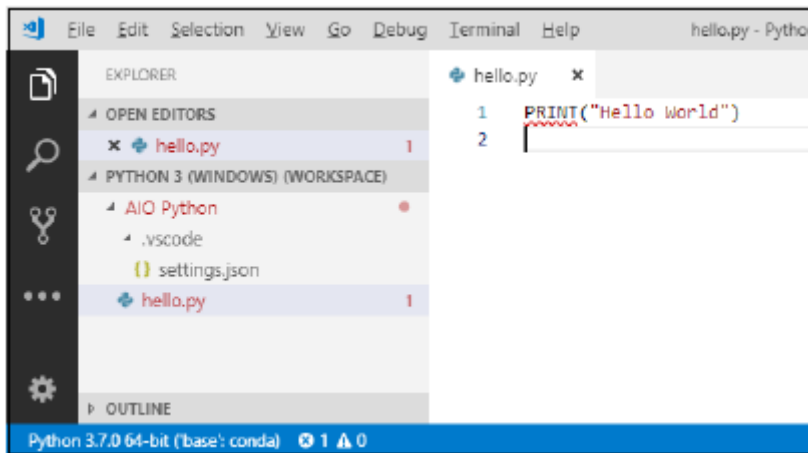
To nie jest najbardziej ekscytująca aplikacja na świecie, ale przynajmniej teraz wiesz, jak pisać, zapisywać i uruchamiać programy w języku Python w VS Code, i to jest umiejętność, której będziesz często używać. Jeśli używasz programu PowerShell w oknie terminala, możesz zobaczyć komunikat o przejściu do wiersza polecenia. O ile nie jesteś ekspertem PowerShell i naprawdę go nie potrzebujesz (z jakiegokolwiek powodu), możesz równie dobrze kliknąć opcję Użyj wiersza polecenia, jeśli zobaczysz tę opcję, aby monit nie przeszkadzał w przyszłości.

Proste debugowanie

Kiedy uczysz się pisanie kodu, musisz popełnić wiele błędów. Dobra wiadomość jest taka, że to nic wielkiego. Nic nie zniszczysz ani nie zniszczysz. Kod po prostu nie działa zgodnie z oczekiwaniami. Zanim nawet spróbujesz uruchomić jakiś kod, możesz zobaczyć kilka wskaźników na ekranie wskazujących na błąd w kodzie:

- * Nazwa folderu i pliku zawierającego błąd będzie czerwona w okienku Eksploratora.
- * Liczba błędów w pliku będzie wyświetlana na czerwono obok nazwy pliku na pasku Eksploratora.
- * Całkowita liczba błędów pojawi się obok zakreślonego X w lewym dolnym rogu okna CS Code.
- * Zły kod prawdopodobnie będzie miał czerwoną falę podkreśloną poniżej.

Rysunek poniższy pokazuje przykład, w którym wpisaliśmy PRINT wielkimi literami, co nie jest dozwolone w Pythonie. W Pythonie rozróżniana jest wielkość liter, więc gdy pokazujemy polecenie wpisywania małymi literami, oznacza to, że musisz także wpisywać je małymi literami.

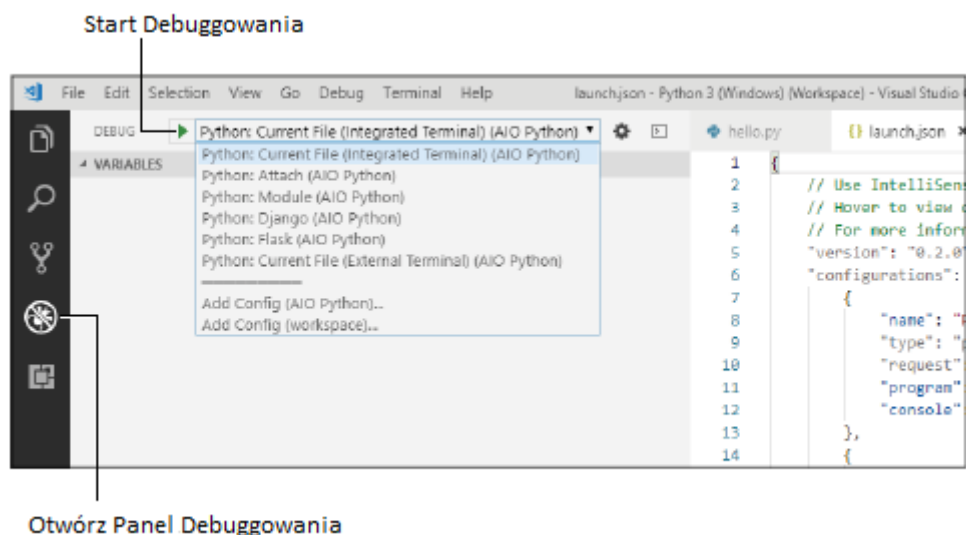


Aby uruchomić plik w terminalu, musisz naprawić błąd. Dotknięcie wskaźnika myszy do słowa z czerwoną falistą linią poniżej daje krótki (choć wysoce techniczny) opis problemu. W przykładzie pokazanym na obrazku wpisaliśmy po prostu PRINT (wielkie litery) zamiast print (małe litery). Aby więc naprawić błąd, po prostu zamieniamy PRINT na print, a następnie zapisujemy zmianę (chyba że włączyłeś Auto Save). Następnie możesz kliknąć prawym przyciskiem myszy i wybrać Uruchom plik Python w terminalu, aby uruchomić poprawiony kod.

Debugger VS Code Python

VS Code ma również wbudowany debugger, który może pomóc podczas pracy z bardziej złożonymi programami, a także zapewnia inny sposób uruchamiania programów Python w VS Code. W tej chwili nie będziemy robić niczego bardzo złożonego. Ale z pewnością nie ma nic złego w skonfigurowaniu i przygotowaniu debugera w ramach rozwoju przestrzeni roboczej Pythona. Aby to zrobić teraz, wykonaj następujące kroki:

1. Kliknij ikonę Debug po lewej stronie okienka Eksploratora.
2. Obok opcji Brak konfiguracji u góry panelu kliknij ikonę koła zębatego (która prawdopodobnie pokazuje teraz czerwoną kropkę, ponieważ nie określiłeś jeszcze debugera). Nowy plik o nazwie launch.json otworzy się po prawej stronie.
3. Kliknij otwórz menu rozwijane u góry panelu i wybierz Python: bieżący plik (terminal zintegrowany) (Python AIO), jak pokazano na rysunku



4. Zamknij plik `launch.json`, klikając X na zakładce.

Odtąd, jako alternatywę do korzystania z metody kliknięcia prawym przyciskiem myszy, aby uruchomić kod Pythona, możesz korzystać z okienka debugowania. Ale najpierw musisz wiedzieć, który plik zostanie uruchomiony. Na razie jest to łatwe, ponieważ `hello.py` jest jedynym, który mamy. W przyszłości, gdy możesz mieć kilka otwartych, możesz powiedzieć, który jest otwarty, ponieważ pojawia się w obszarze edycji po prawej stronie okienka eksploracji. Aby go uruchomić, w okienku debugowania kliknij strzałkę Rozpocznij debugowanie u góry okienka debugowania, która pokazuje się jako zielony trójkąt. Po kliknięciu przycisku Rozpocznij debugowanie kod Pythona będzie działał tak samo, jak po wybraniu opcji Uruchom plik Python w terminalu. Jednak potrwa to trochę dłużej, a na terminalu pojawi się więcej tekstu. Jednak dopóki nie ma żadnych faktycznych błędów, pod tym wszystkim powinien pojawić się wynik (Hello World) z wiersza polecenia dla systemu operacyjnego. Jeśli wydaje się to dużo do zapamiętania, na razie wystarczy pamiętać, że za każdym razem, gdy chcesz uruchomić jakiś kod Python w VS Code, możesz wykonać jedną z następujących czynności, w zależności od tego, która z tych opcji jest dla Ciebie najwygodniejsza:

- * Kliknij prawym przyciskiem myszy nazwę pliku `.py` i wybierz polecenie Uruchom plik w terminalu.
- * Kliknij nazwę pliku `.py` na pasku Eksploratora, aby wybrać ten plik, kliknij opcję Debuguj, a następnie kliknij opcję Rozpocznij debugowanie w górnej części okienka debugowania.

Jeśli pamiętasz te dwie rzeczy, jesteś na dobrej drodze do zarabiania na Pythonie i masz dobre środowisko do pracy.

Pisanie kodu w notatniku Jupyter

W Części 1 dowiedziałeś się o notatnikach Jupyter jako innym sposobie pisania i uruchamiania kodu w języku Python. Tu chcielibyśmy wykorzystać to, czego się tam nauczyłeś, pokazując, jak tworzyć, zapisywać i otwierać notesy Jupyter. Możesz oczywiście zapisać swoje zeszyty Jupyter w dowolnym miejscu, używając dowolnych nazw plików. W naszym przykładzie roboczym utworzymy podfolder o nazwie Notatki Jupyter w folderze AIO Python, aby wszystko było razem.

Tworzenie folderu dla Jupyter Notebook

Folder Jupyter Notebooks nie różni się niczym innym, więc możesz go utworzyć przy użyciu dowolnej metody, której zwykle używasz w systemie operacyjnym. Umieścimy nasz w utworzonym przez nas folderze AIO Python, aby wszystkie pliki tej książki były w jednym miejscu. Aby to zrobić, wykonaj następujące czynności:

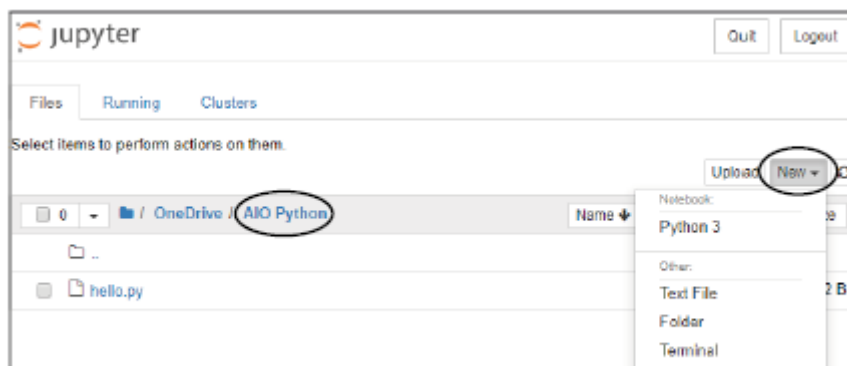
1. Otwórz folder AIO Python w Finderze (Mac) lub Explorersze (Windows).
2. Kliknij prawym przyciskiem myszy puste miejsce w tym folderze i wybierz New ⇨ Folder (w systemie Windows) lub New Folder na komputerze Mac.
3. Wpisz Jupyter Notebooks jako nazwę folderu i naciśnij klawisz Enter. Teraz, gdy masz folder, w którym możesz zapisać notesy Jupyter, możesz utworzyć notatnik, jak omówiono poniżej.

Tworzenie i zapisywanie notatnika Jupyter

Aby utworzyć notatnik Jupyter i zapisać go w folderze, wykonaj następujące kroki:

1. Otwórz Anakondę (jeśli jeszcze nie jest otwarta) i stamtąd uruchom Notatniki Jupyter.
2. Na pierwszej stronie przejdź do folderu Notatniki Jupyter utworzonego w poprzedniej sekcji. Powinieneś zobaczyć coś takiego: Lista notatników jest pusta, ponieważ folder jest pusty.
3. Kliknij Nowy i wybierz Python 3.
4. W górnej części nowego otwartego notatnika kliknij Bez tytułu, wpisz nową nazwę 01 Notatnik i kliknij Zmień nazwę.

To wszystko, notatnik jest tworzony i zapisywany w folderze AIO Python, jak pokazano na rysunku.



Wpisywanie i uruchamianie kodu w notatniku

Gdy notatnik jest otwarty, możesz wpisać kod Python w dowolnej komórce Code i tekst w dowolnej komórce Markdown. Gdy zobaczysz słowo Code w menu rozwijanym na pasku narzędzi poniżej paska menu, aktywna komórka służy do pisania kodu. Aby spróbować, wykonaj następujące czynności:

1. Kliknij komórkę Kod (po prawej stronie In []): i wpisz print („Hello World”) i nie zapomnij używać małych liter do wyświetlenia słów.
2. Aby uruchomić kod, przytrzymaj klawisz Alt (w systemie Windows) lub klawisz Opcja (na komputerze Mac) i naciśnij klawisz Enter lub kliknij trójkąt Play po lewej stronie słowa In.

Dane wyjściowe z kodu są wyświetlane poniżej komórki, a inna komórka otwiera się poniżej

Dodanie tekstu Markdown

Jak wspomniano, do notatników Jupyter można dodawać tekst (a właściwie zdjęcia i wideo). Nie musisz używać żadnego specjalnego kodu do pisania zwykłego tekstu. Jeśli chcesz przeprowadzić formatowanie lub dodać zdjęcia i filmy, musisz użyć kodu Markdown. Markdown to popularny język znaczników, coś w rodzaju znacznie uproszczonego HTML. Chociaż nie możemy przejść do długiego samouczka na temat Markdown tutaj w książce Python, możemy powiedzieć, że prawdopodobnie łatwiej jest wpisać treść Markdown w VS Code, a następnie skopiować / wkleić ją do komórki Markdown niż jest aby wpisać bezpośrednio w komórce Markdown w Jupyter. Upewnij się tylko, że pracując z Markdown, wybierasz Markdown z menu rozwijanego na pasku narzędzi. Rysunek poniższy pokazuje, gdzie dodaliśmy trochę Markdown i tekst do komórki Markdown w Jupyter.



Aby uruchomić komórkę zawierającą Markdown, kliknij komórkę, a następnie kliknij Uruchom na pasku narzędzi. Kod jest renderowany na tekst i dowolną inną treść umieszczoną w komórce. Aby zmienić kod w komórce Kod, wystarczy kliknąć komórkę i wpisać kod normalnie. Aby zmienić zawartość komórki Markdown, najpierw dwukrotnie kliknij tekst lub puste miejsce w komórce, aby ponownie zobaczyć kod, a następnie wprowadź zmiany. Po wprowadzeniu zmian kliknij Uruchom ponownie. Pamiętaj, że tylko komórka zawierająca kursor uruchomi się ponownie. Jeśli chcesz uruchomić wszystkie komórki w notatniku, użyj przycisku podwójnego trójkąta nieco po prawej stronie przycisku Uruchom.

Zapisywanie i otwieranie notatników

Aby zapisać notatnik Jupyter, kliknij przycisk Zapisz na pasku narzędzi lub wybierz File ⇔ Save and Checkpoint z menu. Aby zamknąć notatnik, wybierz z menu File ⇔ Save and Halt. Za każdym razem, gdy chcesz ponownie otworzyć notatnik w przyszłości, otwórz Anacondę i stamtąd uruchom Notatnik Jupyter. Następnie przejdź do zapisanego pliku i kliknij jego nazwę pliku. Nazwa pliku prawdopodobnie będzie zawierać rozszerzenie `.ipynb`, ponieważ jest to standard w przypadku notatników Jupyter. Warto zauważyć, że po otwarciu folderu AIO Python w VS Code zobaczysz nowy folder Notatniki Jupyter w okienku Eksploratora. Pamiętaj, że umieściliśmy tam ten folder, aby uporządkować pliki. Naprawdę nie ma powodu, aby otwierać ten folder w VS Code. Folder nosi nazwę Notatniki Jupyter, ponieważ dotyczy tylko plików tworzonych i zarządzanych w aplikacji Notatnik Jupyter. Ok, więc zagłęбилиśmy się tutaj nieco w VS Code i Notatnik Jupyter, głównie po to, abyś mógł zapisywać i otwierać pliki Pythona i Notatniki Jupyter w dowolnym momencie w przyszłych częściach. Wszystkie te umiejętności okażą się przydatne, gdy zaczniesz głębiej pisać kod Python. Do zobaczenia tam!