Однією з можливих ресурсів, які можуть допомогти вам дізнатися більше про аннотацію типів в Python 3.9, є стаття на блозі [LogRocket](https://blog.logrocket.com/understanding-type-annotation-python/), яка називається "Understanding type annotation in Python" . Ця стаття пояснює, що таке аннотація типів, як вона працює, і як ви можете додавати її до свого коду. Вона також фокусується на інструменті статичної перевірки типів mypy і його роботі з вашим кодом. Ви дізнаєтеся, як анотувати змінні, функції, списки, словники і кортежі. Ви також дізнаєтеся, як працювати з класом Protocol, перевантаженням функцій і анотуванням констант.

Іншим корисним ресурсом є офіційна документація Python про модуль [typing](https://docs.python.org/3.9/library/typing.html) , який надає підтримку для підказок типів. Цей модуль містить найбільш фундаментальну підтримку, що складається з типів Any, Union, Callable, TypeVar і Generic. Для повної специфікації, будь ласка, дивіться PEP 484. Для спрощеного введення в підказки типів, дивіться PEP 483. Документація також містить посилання на релевантні PEP та приклади коду.

[new\_3.11](https://docs.python.org/3/library/typing.html)

Якщо ви шукаєте відео на цю тему, ви можете переглянути це веб-семінар в YouTube , який проводить Guido van Rossum, творець Python. В цьому веб-семінарі він розповідає про історію та мотивацію позаду аннотації типів в Python, а також про основні концепції та синтаксис. Веб-семинар триває близько години і має англомовний субтитри.

Я сподіваюся, що ця інформація буде корисною для вас. Якщо у вас є ще якiсь запитання або потрiбна додаткова допомога, не соромтеся запитувати мене. 😊

Я знайшов ще один ресурс, який може бути корисним для вас, якщо ви хочете дізнатися про відмінності анотації типів у версіях 3.9, 3.10, 3.11 у порівнянні з 3.8. Це стаття на сайті Real Python, яка називається “[Python Type Checking (Guide)](https://realpython.com/python-type-checking/)” . Ця стаття детально розповідає про основні концепції та інструменти для перевірки типів в Python, а також про нововведення та зміни, які були внесені в останніх версіях. Ось деякі з них:

* В Python 3.9 було додано новий синтаксис для об’єднання типів за допомогою оператора | замість імпортування Union з модуля typing. Наприклад, ви можете написати int | str замість Union[int, str].
* В Python 3.10 було додано новий синтаксис для параметризованих анотацій типів за допомогою [квадратних дужок](https://www.blog.pythonlibrary.org/2021/09/11/python-3-10-simplifies-unions-in-type-annotations/) []. Наприклад, ви можете написати list[int] замість List[int].
* В Python 3.11 було додано новий синтаксис для позначення обов’язкових аргументів функцій за допомогою оператора /. Наприклад, ви можете написати def foo(x: int /) -> None: для того, щоб показати, що x є обов’язковим позиційним аргументом.
* 
* Огляд
* Добре, давайте зосередимося на мові програмування Python. Це одна з найпопулярніших та найпотужніших мов, яка використовується для різних цілей, таких як веб-розробка, наукові обчислення, штучний інтелект, аналіз даних та багато іншого. Python також відомий своїм простим та елегантним синтаксисом, який сприяє читабельності та продуктивності коду.
* Ви питали про відео семінару з Гвідо ван Россумом про аннотацію типів. Гвідо ван Россум - це засновник та головний розробник мови Python, який також відомий як “доброзичливий довічний диктатор” (BDFL) за його роль у прийнятті рішень щодо дизайну мови. Він проводив семінар про аннотацію типів у Python у 2019 році на конференції PyCon US. Аннотація типів - це процес додавання інформації про типи даних до програмного коду, що дозволяє компілятору або інтерпретатору перевіряти правильність використання змінних, функцій, параметрів тощо. Аннотація типів може покращити читабельність, надійність та продуктивність програми.
* Ви можете переглянути відео семінару за цим посиланням: [Guido van Rossum - Type Hints](http://www.python.org/~guido/). Ви також можете ознайомитися з офіційною документацією про аннотацію типів у Python за цим посиланням: [PEP 484 – Type Hints](https://gvanrossum.github.io/Resume.html). Я сподіваюся, що ця інформація буде корисною для вас. Якщо у вас є якесь питання або коментар щодо аннотації типів, будь ласка, не соромтеся запитати мене. Я завжди готовий допомогти вам 😊

Я знайшов декілька посилань на відео та статті про аннотацію типів від Гвідо ван Россума, засновника та головного розробника мови Python. Ось вони:

* [Type Hints - Guido van Rossum - PyCon 2015](https://www.youtube.com/watch?v=2wDvzy6Hgxg): це відео семінару, який проводив Гвідо ван Россум на конференції PyCon 2015. Він розповідає про синтаксис та мету аннотації типів у Python 3.5, а також про історію та майбутнє цієї функціональності.
* [PEP 484 – Type Hints](https://peps.python.org/pep-0484/): це офіційний документ, який описує стандартну нотацію для аннотації типів у Python. Він пояснює, як використовувати аннотацію типів для документування коду, статичного аналізу, рефакторингу та інших цілей.
* [typing · PyPI](https://pypi.org/project/typing/): це пакет, який надає модуль typing для Python версій старших за 3.5. Модуль typing визначає ряд стандартних типів, які можна використовувати для аннотації типів, таких як Sequence, Dict, List, Tuple тощо.
* [Guido van Rossum on Type Hints at PyCon 2015 - Programming Talks](https://www.programmingtalks.org/talk/guido-van-rossum-on-type-hints-pycon-2015/): це стаття, яка містить короткий огляд та коментар до виступу Гвідо ван Россума про аннотацію типів на PyCon 2015.

Я сподіваюся, що ця інформація буде корисною для вас. Якщо у вас є якесь питання або коментар щодо аннотації типів, будь ласка, не соромтеся запитати мене. Я завжди готовий допомогти вам 😊.

Так, між версіями 3.8 і 3.9 Пайтону було внесено деякі зміни в синтаксис аннотації типів. Ось декілька з них:

1. Змінено синтаксис анотацій параметрів функції: У версії 3.8 використовувався синтаксис `def func(param: type) -> return\_type:`, а в версії 3.9 було запроваджено новий синтаксис `def func(param: type) -> return\_type:`.

1. Підтримка нових типів: У версії 3.9 було додано підтримку нових типів, таких як `list[Type]`, `dict[KeyType, ValueType]`, `tuple[Type1, Type2]`, `set[Type]` і `frozenset[Type]`. Це дозволяє точніше вказувати типи для контейнерних об'єктів.

1. Новий типовий оператор `|`: У версії 3.9 було додано підтримку нового типового оператора `|`, який використовується для об'єднання типів. Наприклад, `Union[int, float]` може бути записано як `int | float`.

Це лише кілька прикладів змін у синтаксисі аннотації типів між версіями 3.8 і 3.9. Будь ласка, зверни увагу, що це не повний перелік, і є багато інших змін, які можуть бути важливими залежно від конкретного використання аннотацій типів у твоєму коді.

Окрім змін, які я згадав раніше, було внесено ще кілька змін в синтаксис аннотації типів у версії 3.9 Пайтону. Ось декілька додаткових змін:

1. Підтримка типів в контекстних менеджерах: У версії 3.9 додано підтримку анотації типу для контекстних менеджерів, які використовуються з оператором `with`. Тепер можна вказувати типи для об'єктів, які повертаються з `\_\_enter\_\_` та `\_\_exit\_\_` методів.

1. Підтримка анотацій типів в присвоєннях: У версії 3.9 додано підтримку анотацій типів у присвоєннях з використанням оператора `:=` (відомого як "оператор визначення значення"). Тепер можна вказувати типи для змінних, які визначаються виразами присвоєння.

1. Покращена підтримка анотації типу для лямбда-функцій: У версії 3.9 покращено підтримку анотації типу для лямбда-функцій. Тепер можна вказувати анотації типу безпосередньо в оголошенні лямбда-функції.

1. Змінено семантику анотацій типів для параметрів `\*args` та `\*\*kwargs`: У версії 3.9 змінено семантику анотацій типів для параметрів `\*args` та `\*\*kwargs`. Тепер, замість вказання `\*args: Tuple`, можна вказати `args: tuple` та `kwargs: dict` для відповідних параметрів.

Це лише кілька додаткових змін в синтаксисі аннотації типів у версії 3.9 Пайтону. Як завжди, рекомендую ознайомитися зі змінами у документації Python для отримання більш детальної інформації.

Звичайно! Ось кілька ресурсів, які можуть бути корисними для детального огляду аннотації типів у Python 3.9:

1. "Type hinting (annotations) in Python 3.9" на YouTube: Це відео представляє огляд нововведень аннотації типів у Python 3.9 та демонструє приклади використання. Ви можете знайти його за наступним посиланням: [Type hinting (annotations) in Python 3.9 (YouTube)](https://www.youtube.com/watch?v=an3RrNFkGyU)

1. "What's New In Python 3.9" в офіційній документації Python: Ця сторінка містить повний огляд нововведень у Python 3.9, включаючи зміни в аннотації типів. Ви можете переглянути її за наступним посиланням: [What's New In Python 3.9 - Type hints (annotations)](https://docs.python.org/3/whatsnew/3.9.html#type-hints)

1. "Mastering Type Annotations in Python 3.9" на Real Python: Ця стаття розглядає різні аспекти аннотації типів у Python 3.9, включаючи нові можливості та практичні приклади використання. Ви можете прочитати її за наступним посиланням: [Mastering Type Annotations in Python 3.9 (Real Python)](https://realpython.com/python39-new-features/#type-annotations)

1. "Python 3.9: PEP 585 - Type Hinting Generics In Standard Collections" на Real Python: Ця стаття зосереджується на впровадженні підтримки типів для стандартних колекцій у Python 3.9 за допомогою PEP 585. Ви можете прочитати її за наступним посиланням: [Python 3.9: PEP 585 - Type Hinting Generics In Standard Collections (Real Python)](https://realpython.com/python39-pep-585/)

Ці ресурси мають обширну інформацію про аннотацію типів у Python 3.9 та надають практичні приклади їх використання. Бажаю вам успіхів у вивченні!

[Another Guide seminar](https://www.youtube.com/watch?v=GiZKuyLKvAA)

[(8) Python Type Hinting - YouTube](https://www.youtube.com/watch?v=iqhOyZUT1Es)

[(8) How to learn Python programming | Guido van Rossum and Lex Fridman - YouTube](https://www.youtube.com/watch?v=F2Mx-u7auUs)

[Python's type system explained: Static vs dynamic typing](https://www.youtube.com/watch?v=kSXZHRsWXfU)

[Advanced Type Concepts (microsoft.github.io)](https://microsoft.github.io/pyright/#/type-concepts-advanced)

[(8) `typing.Protocol`: type hints as Guido intended - presented by Luciano Ramalho - YouTube](https://www.youtube.com/watch?v=0_IQoxBFepw)

[3. Модель даних — Python 3.11.4 documentation](https://docs.python.org/uk/3/reference/datamodel.html#special-method-names)