

## Developer Stories: Guido van Rossum, Python-Vater, netter Diktator und Problemlöser

*Deutschsprachiges Videotranskript*

**Frage:** Woran arbeitest du bei Microsoft?

**Guido van Rossum:** Erstens das Projekt, das ich vor etwa einem Jahr bei Microsoft begonnen habe, bei dem ich ein Team von Leuten gebildet habe, um den Python-Interpreter schneller zu machen. Und wir haben dort Ergebnisse erzielt, bei denen die nächste Version von Python - Python 3.11, die im Oktober dieses Jahres herauskommen wird - nach unserem Benchmark etwa 20 bis 25 Prozent schneller ist als Python 3.10, das letzten Oktober herauskam. Andere Projekte, an denen ich arbeite, sind interne Beratungsprojekte, bei denen ich anderen Teams bei Microsoft helfe, Python in öffentliche oder für Entwickler\*innen bestimmte Microsoft-Produkte zu integrieren.

**Frage:** Du wurdest als der wohlwollende Diktator für das Leben in der Python-Community bezeichnet - was ist damit gemeint?

**Guido van Rossum:** Meine Rolle als wohlwollender Diktator auf Lebenszeit - das war eigentlich ein Scherztitel, der in den späten 90ern speziell für mich erfunden wurde. Und dann tauchte er in den Aufzeichnungen eines Treffens von 1996 oder 1997 auf, bei dem eine Gruppe von Leuten zusammenkam und einige lokale Python-Aktivitäten plante. Und weil wir in einer heiteren Stimmung waren, haben wir uns alle gegenseitig Scherztitel zugewiesen, und einer der Anwesenden war wirklich gut darin, alberne Witze zu improvisieren. Ich glaube, zuerst war ich der ‚vorläufige wohlwollende Diktator auf Lebenszeit‘, und dann war es auch eine Anerkennung, weil es seine Sprache ist, weil er sie geschaffen hat und er immer noch hier ist, und das Design und die Implementierung anleitet. Später, in den frühen 00er Jahren, als sich die Dinge änderten und ich ein paar Mal den Arbeitsplatz wechselte, begannen die Leute, sich auf denselben Begriff zu beziehen, abgekürzt als BDFL. Und schließlich wurde es zu einem Begriff, der auch für einzelne Leiter\*innen eines Open-Source-Projekts verwendet wurde, die die ursprünglichen Schöpfer\*innen oder die aktuell leitenden Betreuer\*innen waren. Derzeit hat Python also keinen BDFL mehr. Ich bin 2018 zurückgetreten, glaube ich. Wir haben einen Lenkungsrat, der aus fünf Personen besteht, die jährlich von der Gruppe der Kernentwickler\*innen gewählt werden, von denen wir, ich weiß nicht, ungefähr hundert oder so haben.

**Frage:** In welcher Weise bist du heute noch aktiv an der Entwicklung von Python beteiligt?

**Guido van Rossum:** Nun, bei Python sind die Themen, an denen ich derzeit beteiligt bin, meist die Themen, die ich im letzten Jahrzehnt begonnen habe, zum Beispiel eine asyncio-

Bibliothek, und ich bin immer noch gelegentlich an Verbesserungen daran beteiligt. Die letzte Sache, die wir zum Beispiel in 3.11 hinzugefügt haben, sind Task-Gruppen, die eine Möglichkeit darstellen, Tasks auszuführen und darauf zu warten, dass sie ordnungsgemäß beendet werden. Eine große Sache, die ich kurz nach dem Asyncio-Zeug angefangen habe, sind Typ-Annotationen, und das ist ein Bereich, in dem es eine Menge Sprachdesign-Aktivitäten gibt, die ich versuche, so gut ich kann zu leiten, um den Leuten zu helfen, gute Ideen zu entwickeln, um diese Ideen in eine Syntax zu bringen, die für Python-Benutzer\*innen akzeptabel und verständlich ist. Und schließlich beaufsichtige ich das schnellere C-Python-Projekt. Glücklicherweise haben wir jetzt eine\*n Vollzeitmanager\*in eingestellt, so dass das Projekt auch dann laufen kann, wenn ich nicht jeden Tag allen die Hand halte.

**Frage:** Wohin steuert die Entwicklung von Python?

**Guido van Rossum:** Die Python-Entwicklung ist generell evolutionär. Python macht langsame und vorsichtige Schritte in Richtung neuer Funktionen und Versionen. Der derzeitige Lenkungsausschuss ist zwar offen für alle möglichen Ideen zur Verbesserung der Sprache, legt aber auch großen Wert auf die Erhaltung der Rückwärtskompatibilität. Und so ist die Leistung eine großartige Sache, an der man arbeiten kann, weil man alles schneller machen kann, ohne die APIs zu ändern, ohne den Code der Leute zu zerstören - obwohl es sich in der Praxis als schwieriger erweist als in der Theorie. Was neue Funktionen angeht, so wollen wir nach und nach eine verbesserte Syntax für type annotations und typing im Allgemeinen einführen. Das ist immer noch ein kontroverses Thema. Ich sehe das als ein Programm, das in den nächsten fünf Jahren laufen wird. Es gibt Leute, die hart daran arbeiten, Python zu einer Web-Assembly zu kompilieren, ein Gebiet, über das ich sehr wenig weiß.