



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



Noch bis vor einiger Zeit gab es in der Entwickler-Community einige Debatten welche Version von Python man als Einsteiger lernen sollte: Python 2 oder Python 3.

In **2019** ist diese Frage im Grunde eindeutig beantwortet: Wer jetzt mit Python anfängt oder seine Fähigkeiten auffrischen will, sollte Python 3 nutzen. In diesem Artikel wollen wir besprechen, was **Python 3** besser macht, und warum viele Firmen zu Python 3 wechseln.

Dennoch gibt es immer noch einige Szenarien, in denen der Einsatz von Python 2 vorteilhaft sein kann. Es lohnt sich auf jeden Fall, die Geschichte der Entwicklung und die Unterschiede von **Python 2 und Python 3** zu kennen.

Schauen wir uns also an, was Python 2 von Python 3 unterscheidet, und wo du als Anfänger einen guten Einstieg in die Programmierung mit Python 3 bekommen kannst.

Die Geschichte von Python 2 und Python 3

Hier wollen wir kurz einige zeitliche Meilensteine in der Entwicklung von Python beschreiben:

- **Python 2.0** wurde zum ersten Mal im Jahr 2000 veröffentlicht. Die neueste Version, 2.7, wurde im Jahr 2010 veröffentlicht.
- **Python 3.0** wurde im Jahr 2008 veröffentlicht. Die neueste Version, 3.7, wurde 2018 veröffentlicht. Version 3.8 ist aktuell in Entwicklung.
- Obwohl Python 2.7 immer noch stark genutzt wird, geht der Trend eindeutig zum Python 3. Im Jahr 2016 nutzten 71.9% aller Projekte Python 2.7, aber diese Zahl ging 2017 bereits auf 63.7% zurück. Dies zeigt, dass die Programmiergemeinschaft für produktionsreife Projekte mittlerweile vermehrt Python 3 einsetzt.
- Die offizielle Unterstützung von Python 2 endet im Jahr 2020. Ab da wird es keine neuen Releases mehr geben.
- Tigerjython ist eine auf eine JAVA-Runtime aufgesetzte Python 2.7 Version mit grafischer Benutzeroberfläche und einem integriertem Turtle-Grafik-Modus (Schweiz) – umfangreichste und bunte Lehrbuchsammlung für Schüler

Was sind die Hauptunterschiede zwischen Python 2 und 3?

Es gibt einige Unterschiede zwischen den verschiedenen Python-Versionen. Hier wollen wir auf fünf der wichtigsten eingehen.

1. Python 2 ist die Vergangenheit, Python 3 ist die Zukunft.

Da Python 2 für über zehn Jahre die populärste Version von Python war, wird es immer noch von vielen Softwareschmieden eingesetzt.

Trotzdem sollte ein Einsteiger heute Python 3 lernen. Der Grund liegt auf der Hand: Wenn man sowieso von null startet, sollte man keine Zeit damit verlieren, etwas zu lernen, das in absehbarer Zeit obsolet sein wird.

2. Python 2 und Python 3 nutzen Bibliotheken, die zum Teil zueinander inkompatibel sind.

Da Python 3 die zukünftige Standardversion von Python ist, werden viele neue Bibliotheken exklusiv für Python 3 entwickelt. Manche der für Python 2 entwickelten Bibliotheken wiederum funktionieren nicht mit Python 3. Es ist natürlich möglich, diese Bibliotheken zu konvertieren, aber da dies oft schwierig ist, ist es meist außerhalb der Möglichkeiten eines Python-Einsteigers.

3. Python 3 hat bessere Unterstützung für Unicode.

Alle Strings in Python 3 sind standardmäßig unicode-codiert. Python 2 nutzte hier noch die ASCII-Codierung und Unicode musste hier explizit mit einem „u“ angegeben werden.

Dies ist wichtig, da Unicode deutlich vielfältiger ist als ASCII. Unicode unterstützt Zeichen aus nicht-arabischen Schriftarten, römische Zahlen, Symbole, Emojis und vieles mehr.

Wie du dir vielleicht denken kannst, bringt der Einsatz von Unicode insbesondere für internationalisierte Applikationen massive Vorteile.

4. Python 3 nutzt eine verbesserte Division von Integers

Wenn man in Python 2 zwei Zahlen ohne Nachkommastellen definiert und die beiden durcheinander dividiert, wird das Ergebnis auf die nächste ganze Zahl gerundet.

Als Beispiel schauen wir uns die Division von 7 durch 2 an. Schreiben wir in Python 2:

```
1 summe = 7/2
```

enthält die Variable sum die Zahl 2. Möchte man das richtige Ergebnis 3.5 erhalten, müsste man den Code anders schreiben:

```
1 summe = 7.0/2.0
```

Dies liegt daran, dass Python 2 den korrekten Datentyp des Ergebnisses erkennt. Python 3 dagegen erkennt, dass das Ergebnis als Float zu behandeln ist. Dies ist ein Beispiel dafür, wie Python 3 die Syntax intuitiver macht, was die Programmierung insbesondere für Einsteiger erleichtert.

5. Python 2 und Python 3 haben eine unterschiedliche Syntax für die print-

Dieser Unterschied ist rein syntaktische und wird daher von vielen als trivial eingeschätzt. Dennoch ist es einer der größten und auffälligsten Unterschiede und wahrscheinlich einer, der dir am häufigsten begegnen wird.

In Python 3 wurde die print-Anweisung durch eine print-Funktion ersetzt.

Beispiel:

```
1 print "5"
```

ist valider Code in Python 2, führt aber zu einem Syntaxfehler in Python 3. Dort würde man folgenden Code nutzen:

```
1 print("5")
```

Quelle: <https://lerneprogrammieren.de/python-2-vs-3/>