

Python Intro

Lasst uns mit Python durchstarten

1.

Intro

Was ist Python
und was wird
damit
gemacht?

Wer?

Guido van Rossum ist der Kopf hinter Python und hat die erste Version am 20. Februar 1991 herausgegeben.

Während man Python eher als gigantische Schlange kennt, ist der Name eigentlich eine Anspielung auf **Monty Python's Flying Circus**.



Wer?

Eines der absolut bemerkenswerten Fakten über Python ist, dass es ursprünglich die Arbeit von nur einer Person war.

Normalerweise werden neue Programmiersprachen von grossen Unternehmen entwickelt und veröffentlicht, die viele Fachleute beschäftigen.

Python ist eine Ausnahme.



Wer?

Natürlich hat Guido van Rossum nicht alle Python-Komponenten **selbst** entwickelt und weiterentwickelt.

Python wird von der **Python Software Foundation** verwaltet, einer gemeinnützigen Organisation, die sich der Entwicklung, Verbesserung, Erweiterung und Verbreitung **der Sprache Python** und ihrer Umgebung widmet.



Was?

Python ist eine **weit verbreitete, interpretierte, objektorientierte** und **high-level** Programmiersprache mit dynamischer Semantik, die für die allgemeine Programmierung verwendet wird.

Sie ist **allgegenwärtig**, und die Menschen benutzen täglich zahlreiche Geräte, auf denen Python läuft, ob sie es merken oder nicht.

Frage: Sind Sie schon einmal mit Python in Berührung gekommen?

Warum?

Es gibt eine Reihe von Gründen, die Python grossartig machen:

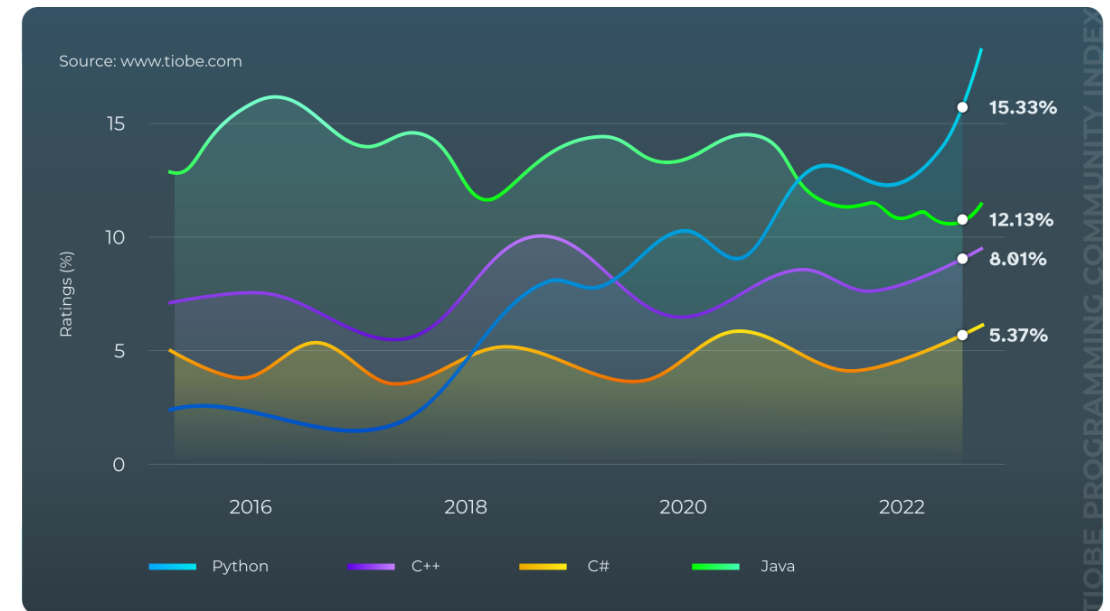
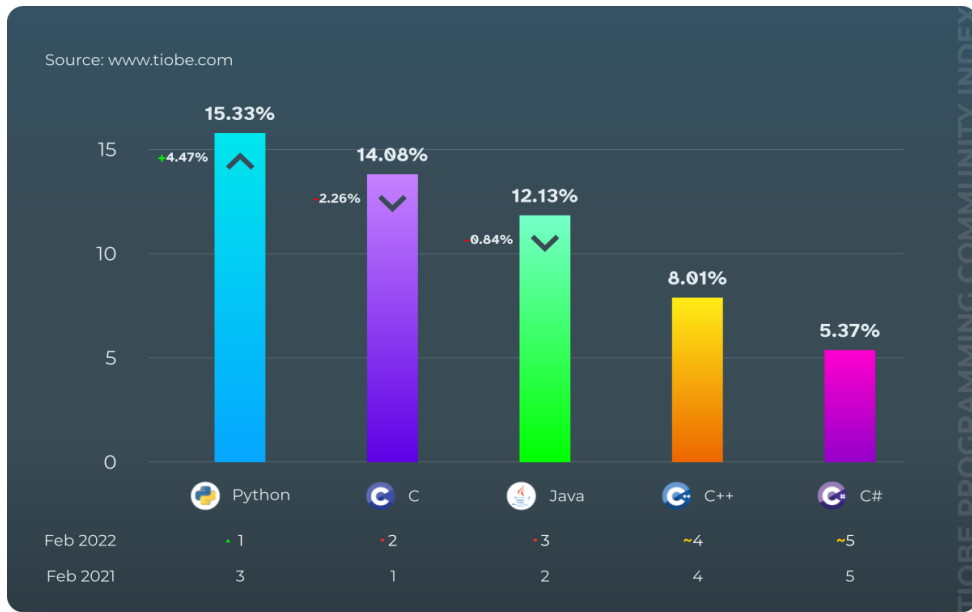
- **Es ist (eher) leicht zu erlernen**
 - Die Zeit, die benötigt wird, um Python zu lernen, ist kürzer als bei vielen anderen Sprachen
- **Der Prozess um Software zu schreiben ist oft kürzer**
 - Viele Bibliotheken und gute Dokumentationen sind vorhanden
- **Die Beschaffung, Installation und der Einsatz ist Einfach**
 - Python ist kostenlos, offen und plattformübergreifend nutzbar

Ziele von Python

Im Jahr 1999 definierte Guido van Rossum seine Ziele für Python:

- Eine einfache und intuitive Sprache, die genauso leistungsfähig ist wie die der grossen Konkurrenten
- Open Source, sodass jeder zu ihrer Entwicklung beitragen kann
- Code, der so verständlich ist wie **simples Englisch**
- Geeignet sein für alltägliche Aufgaben, was kurze Entwicklungszeiten ermöglicht.

Etwas Statistik





Python ist **keine junge Sprache**.

Sie ist **ausgereift und vertrauenswürdig**.

Sie ist **keine Eintagsfliege**.

Die Zeit, die man mit dem Erlernen von Python
verbringt, ist eine **ausgezeichnete
Investition**.

2.

Geschichte

Etwas
Geschichte
zu Python

Python 1.0

Python erreichte im Januar 1994 die Version 1.0. Die wichtigsten neuen Funktionen in dieser Version waren die funktionalen Programmierwerkzeuge **lambda, map, filter und reduce**.

Van Rossum erklärte, dass «Python lambda, reduce(), filter() und map() dank eines Lisp-Hackers erhielt, der diese Funktionen vermisste und **funktionsierende Patches einreichte**».

Python 2.0

Mit Python 2.0, das im Oktober 2000 veröffentlicht wurde, wurden **list comprehensions** eingeführt, eine Funktion, die von den funktionalen Programmiersprachen SETL und Haskell übernommen wurde.

Die Syntax von Python für dieses Konstrukt ist der von Haskell sehr ähnlich, abgesehen von Haskells Vorliebe für Interpunktionszeichen und Pythons Vorliebe für **alphabetische Schlüsselwörter**.

Mit Python 2.0 wurde auch ein **Garbage Collector** eingeführt.

Python 2.7

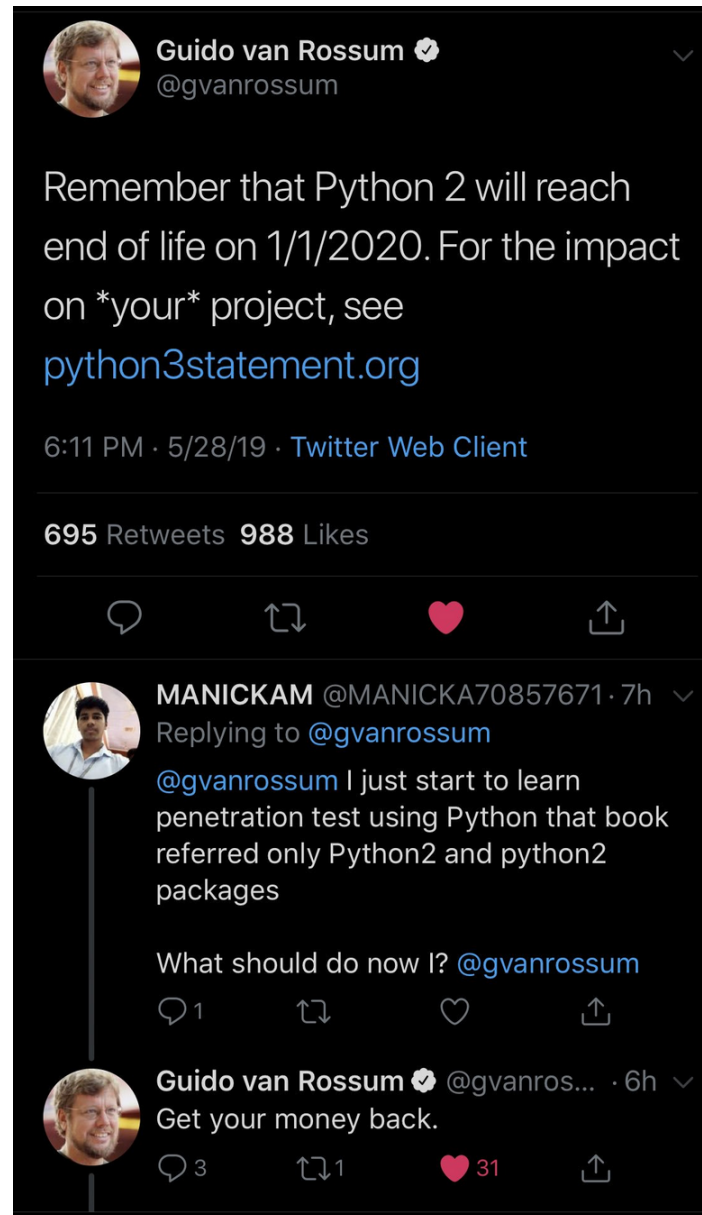
Im November 2014 wurde bekannt gegeben, dass Python 2.7 bis 2020 unterstützt werden würde, aber die Benutzer wurden aufgefordert, **so schnell wie möglich auf Python 3 umzusteigen**.

Die Unterstützung für Python 2.7 endete am 1. Januar 2020, **zusammen mit dem Einfrieren des 2.7-Entwicklungszweigs**.

Eine letzte Version, 2.7.18, wurde am 20. April 2020 veröffentlicht und enthielt Korrekturen für kritische Fehler und Release-Blocker.

Dies markierte das Ende des Lebenszyklus von Python 2.

Python 2.7



A screenshot of a Twitter thread on a dark background. The top tweet is from Guido van Rossum (@gvanrossum), a verified account, posted at 6:11 PM on 5/28/19 via the Twitter Web Client. The tweet text reads: "Remember that Python 2 will reach end of life on 1/1/2020. For the impact on *your* project, see python3statement.org". It has 695 retweets and 988 likes. Below this is a reply from MANICKAM (@MANICKA70857671) posted 7 hours ago, replying to @gvanrossum. The reply text says: "@gvanrossum I just start to learn penetration test using Python that book referred only Python2 and python2 packages". Below the reply is another tweet from Guido van Rossum (@gvanros...) posted 6 hours ago, which says: "Get your money back." This tweet has 3 replies, 1 retweet, and 31 likes. The interface includes icons for replying, retweeting, liking, and sharing.

Guido van Rossum ✓ @gvanrossum
Remember that Python 2 will reach end of life on 1/1/2020. For the impact on *your* project, see python3statement.org
6:11 PM · 5/28/19 · Twitter Web Client
695 Retweets 988 Likes

MANICKAM @MANICKA70857671 · 7h
Replying to @gvanrossum
@gvanrossum I just start to learn penetration test using Python that book referred only Python2 and python2 packages
What should do now I? @gvanrossum

Guido van Rossum ✓ @gvanros... · 6h
Get your money back.

Python 2.7 versus Python 3.x

	Python 2	Python 3
"Print" Keyword	In Python 2 wird print als Anweisung und nicht als Funktion betrachtet.	In Python 3 wird print als Funktion und nicht als Anweisung betrachtet.
Division ganzer Zahlen	Bei der Division von zwei ganzen Zahlen erhalten wir in Python 2 einen ganzzahligen Wert. Zum Beispiel ergibt 7/2 in Python 2 den Wert 3.	Bei der Division von zwei ganzen Zahlen erhalten wir in Python 3 eine Fließkommazahl. Zum Beispiel ergibt 7/2 in Python 3 den Wert 3,5.
Rückwärtskompatibilität	Python-2-Codes können mit viel Aufwand auf Python 3 portiert werden.	Python 3 ist nicht rückwärts kompatibel mit Python 2.

Python 3.x

Python 3.0 (auch «Python 3000» oder «Py3K» genannt) wurde am 3. Dezember 2008 veröffentlicht.

Es wurde entwickelt, um grundlegende Designfehler in der Sprache zu korrigieren – die erforderlichen Änderungen konnten nicht unter Beibehaltung der vollständigen Abwärtskompatibilität mit der 2.x-Serie implementiert werden, was eine neue Hauptversionsnummer erforderlich machte.

Das Leitprinzip von Python 3 war: **«reduce feature duplication by removing old ways of doing things»**.

3.

Syntax

Einige
Beispiele zu
Beginn

Der Beweis, warum Python so beliebt ist

Java

```
public class HelloWorld{  
    public static void main(String args[]){  
        System.out.println("Hello World!");  
    }  
}
```

C++

```
#include <iostream>  
using namespace std;  
int main() {  
    cout << "Hello World!" << endl;  
    return 0;  
}
```

Der Beweis, warum Python so beliebt ist

Und jetzt das Gleiche in Python:

Der Beweis, warum Python so beliebt ist

Und jetzt das Gleiche in Python:

```
Python
```

```
print('Hello World')
```

Warum Python?

- Einfache Syntax
- High-Level-Programmiersprache
- Plattformübergreifend nutzbar
- Interpretiert
- Objektorientiert
- Viele Bibliotheken verfügbar

Python Interpreter

- Stellt sicher, dass der Code korrekt geschrieben ist (Syntaxprüfung)
- Kompiliert danach den Python-Code und führt ihn auch aus
- Es sind unterschiedliche Python-Interpreter-Versionen verfügbar
 - Ein Python 3.11-Interpreter kann Python 3.8-Code kompilieren
 - Ein Python 3.8-Interpreter kann möglicherweise nicht in der Lage, Python 3.11-Code zu übersetzen
 - Python 3.11 bietet neue Funktionen, die der 3.10- oder 3.8-Interpreter nicht kennt

Ein geheimnisvolles erstes Programm


Sehen wir uns unser erstes Codebeispiel an:

```
1 nr_1 = float(input("Please enter a number (step 1 of 3): "))
2 nr_2 = float(input("Please enter a number (step 2 of 3): "))
3 nr_3 = float(input("Please enter a number (step 3 of 3): "))
4
5 result = nr_1 + nr_2 + nr_3
6
7 print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Können Sie erkennen, was das Programm tun wird?

Ein geheimnisvolles erstes Programm

Sehen wir uns unser erstes Codebeispiel an:




```
1 nr_1 = float(input("Please enter a number (step 1 of 3): "))
2 nr_2 = float(input("Please enter a number (step 2 of 3): "))
3 nr_3 = float(input("Please enter a number (step 3 of 3): "))
4
5 result = nr_1 + nr_2 + nr_3
6
7 print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Können Sie erkennen, was das Programm tun wird?

Ein geheimnisvolles erstes Programm

Sehen wir uns unser erstes Codebeispiel an:

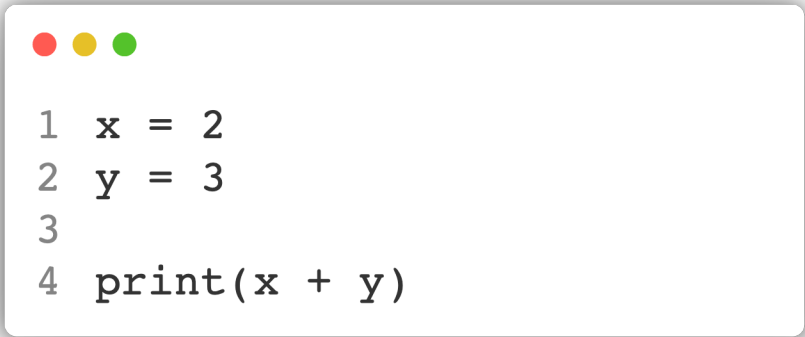


```
1 nr_1 = float(input("Please enter a number (step 1 of 3): "))
2 nr_2 = float(input("Please enter a number (step 2 of 3): "))
3 nr_3 = float(input("Please enter a number (step 3 of 3): "))
4
5 result = nr_1 + nr_2 + nr_3
6
7 print(f"The result of {nr_1} + {nr_2} + {nr_3} is {result}")
```

Können Sie erkennen, was das Programm tun wird?

Ein weiteres Code-Beispiel

Hier ein noch einfacheres Beispiel:



```
1 x = 2
2 y = 3
3
4 print(x + y)
```

Auch cool an Python is die sogenannte REPL (Read-Eval-Print Loop), damit können wir Python direkt in der Konsole ausprobieren!

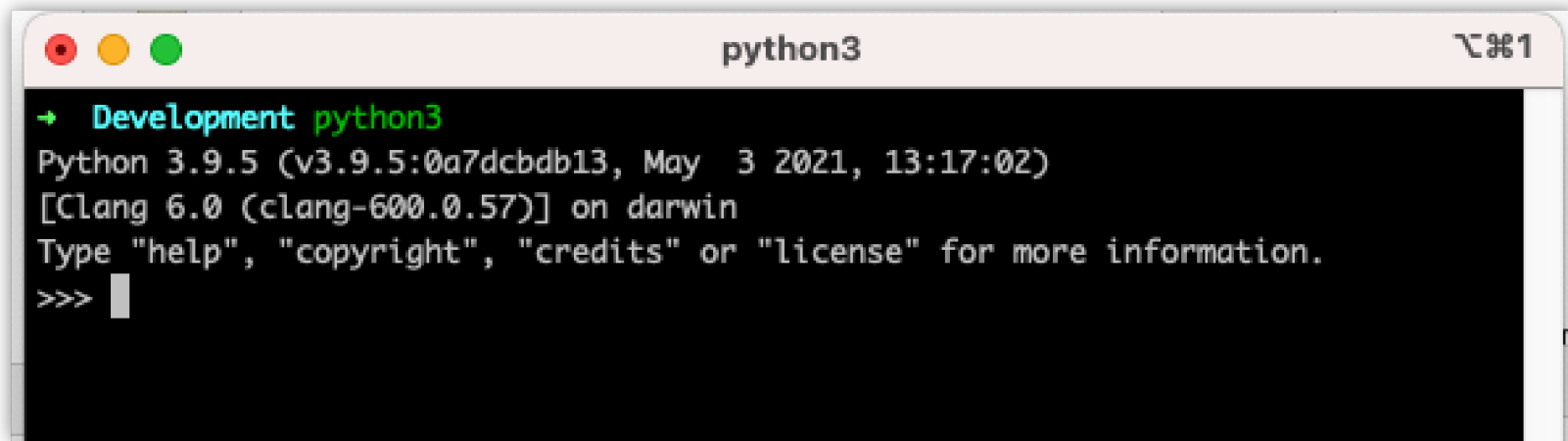
Lassen Sie uns Ihre CMD, PowerShell oder eine andere Konsole starten.

Ein weiteres Code-Beispiel

Windows: Python.exe or Python3.exe

Mac: Python or Python3

Die Version spielt keine Rolle!

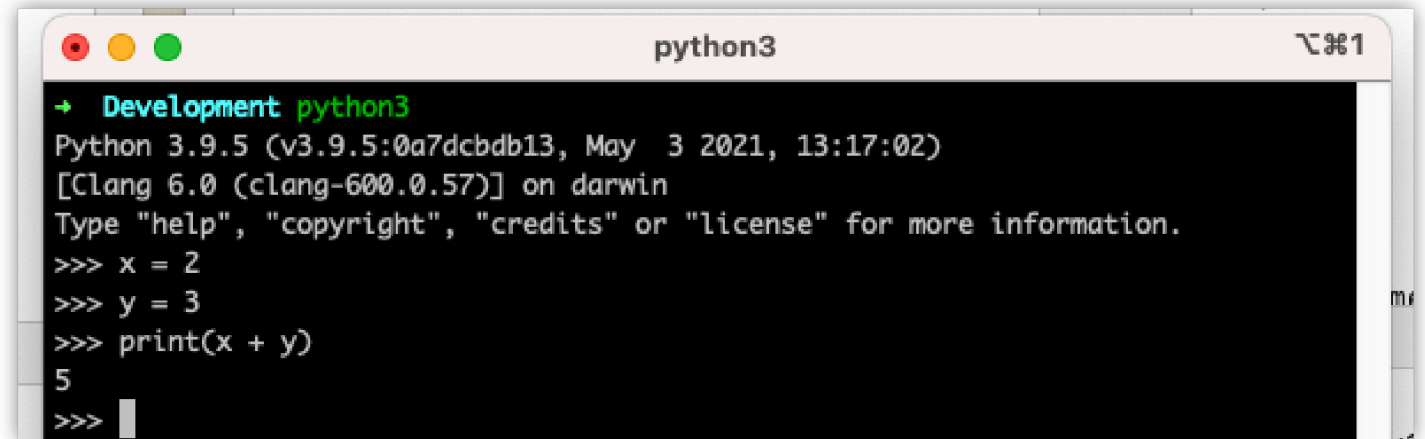


```
→ Development python3
Python 3.9.5 (v3.9.5:0a7dcdb13, May  3 2021, 13:17:02)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Ein weiteres Code-Beispiel

Bitte geben Sie nun Zeile für Zeile ein:

```
1 x = 2
2 y = 3
3
4 print(x + y)
```



The screenshot shows a terminal window titled "python3" with a standard macOS window header (red, yellow, green buttons). The terminal output is as follows:

```
+ Development python3
Python 3.9.5 (v3.9.5:0a7dcdbd13, May  3 2021, 13:17:02)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 2
>>> y = 3
>>> print(x + y)
5
>>> 
```

Herzlichen Glückwunsch

Sie haben gerade Ihr erstes Python-Skript geschrieben!

Viele weitere werden nun folgen :-)



Ende

Das war alles für dieses Kapitel
