

Eine Sprache für Leute mit Deadlines

Python ist nicht gleich Python

Python ist kein Spielzeug – außer Sie spielen damit

Kapitel 1

Orientierung

Bevor es losgeht, brauchen Sie erst mal eine Python-Lizenz. Diese kostet als Einzelplatz-Lizenz 7543,20 €. Alternativ können Sie auch eine Volumenlizenz für eine Bepreisung pro Prozessorkern erwerben. Wenn Sie einen Großrechner mit mehr als 64 Kernen haben, bekommen Sie Rabatt. Umgerechnet kostet die Lizenz dann 5380 €.

Bevor Sie jetzt verärgert das Buch in die Ecke pfeffern, möchten wir uns entschuldigen. Das war nicht ganz ernst gemeint. Python gibt's für lau. Aber auch sonst gibt es viele Dinge, die für Python sprechen.

Motivation

In diesem Buch geht es um die Programmiersprache Python. Es soll Sie motivieren, über den Tellerrand zu blicken, und Ihnen helfen, die Freude an der Programmierung wiederzufinden, wenn Sie sie im Dschungel zwischen fleischfressenden Ameisen und todbringenden Moskitos wie TypeScript-Typen, Java-Objekten, C-Pointern und C++-Templates verloren haben.

Was ist Python?

Python ist eine vielseitige Programmiersprache, die durch Einfachheit und Eleganz besticht.

Warum sollte ich Python lernen?

Wir glauben fest daran, dass sich Ihr Programmier-*Karate* in C, Java oder anderen Sprachen verbessert, wenn Sie erlernen, wie Python manche Dinge regelt.

Egal woher Sie gerade kommen, Python ist einen Besuch wert – und wer weiß, vielleicht möchten Sie ja sogar noch etwas länger bleiben.

Wenn Sie beruflich mit anderen Sprachen arbeiten, kann Python Ihnen interessante Design-Patterns aufzeigen. Vielleicht ist Python aber auch eine sinnvolle Ergänzung zu Ihrem bestehenden Werkzeugkasten. Beispielsweise könnten Sie mit Python eine elegante Test-Umgebung zaubern, wenn Sie viel mit SQL-Datenbanken arbeiten, oder Sie verwenden eines der Dutzende in Python geschriebener Tools.

Python lediglich die zweite Geige spielen zu lassen, wäre aber Verschwendung, denn Python ist als Programmiersprache und -umgebung durchaus ernstzunehmen. Es gibt Hunderte Bibliotheken und Frameworks für alle möglichen Anwendungsfälle, von Server-Monitoring, Webseiten und Computerspielen über Micro-Controller-Steuerung bis hin zu Tests und Automatisierung – es ist für alle was dabei.

Was ist an Python so besonders?

Es gibt viele Dinge, die Python zu etwas Besonderem machen:

- ✓ Einfache, ausdrucksstarke und elegante Syntax
- ✓ Schnell (... genug)
- ✓ Läuft auf allen Betriebssystemen
- ✓ Unglaublich vielseitig (vom Roboter bis zur Webseite ist alles möglich)
- ✓ Die Benutzung macht Spaß
- ✓ Python spielt auch gerne mit den anderen Kindern (Sprachen)
- ✓ Gigantische Community
- ✓ Wenn Sie es sich vorstellen können, dann gibt es dafür schon ein Modul

Python wohnt eine besondere Philosophie inne, die Sie auf der ersten Schummelseite im Buch finden. Das sogenannte »Zen of Python« fasst in 20 Sentenzen die Prinzipien zusammen, die Pythons Design über die Jahre lenkten. Tatsächlich enthält das Gedicht nur 19 Zeilen – die letzte wird vielleicht eines Tages noch aufgeschrieben.

Seit wann gibt es Python?

Vielleicht länger, als Sie dachten!

Von C und Java haben Sie sicherlich schon einmal gehört oder sogar schon Erfahrungen damit gesammelt – immerhin sind diese Sprachen (gefühlte) uralte. Die Sprache C wurde 1972 entwickelt, Java hingegen wurde 1995 in die Welt geworfen. Sicher überrascht es Sie zu erfahren, dass die erste Version von Python bereits im Jahr 1991 erschien – damit ist Python sogar 4 Jahre älter als Java. Trotzdem wirkt Python eventuell wie ein neuzeitliches Phänomen.

Als Lernsprache war Java lange der Platzhirsch an den Unis und wenn Sie sich auf Jobplattformen wie Xing oder LinkedIn nach Java-Programmierjobs umschauchen, so werden Sie

nicht lange suchen müssen; Python-Jobs hingegen sind immer noch eher rar gesät. Python befindet sich jedoch auf der Überholspur und ist an US-Universitäten inzwischen Lernsprache Nummer eins – da ist es eine Frage der Zeit, bis der Jobmarkt diesen Trend widerspiegelt.

Warum ist Python weniger verbreitet als andere Sprachen?

Schwer zu sagen. Firmenpolitik?

Es mag dafür verschiedene Gründe geben, auf die wir nicht *alle* eingehen möchten. Allerdings glauben wir, dass ein Faktor besonders wichtig war: Java wurde von Sun Microsystems entwickelt (eine Firma, die es heute nicht mehr gibt), Python aber wurde von Guido van Rossum erfunden (einem holländischen Informatiker, den es heute durchaus noch gibt).

In der Geschichte spielen sicher andere Faktoren eine wichtige Rolle, doch diese Besonderheit möchten wir hervorheben. Java ist sozusagen ein Kind der Unternehmenswelt, Python hingegen entstand als Spaßprojekt eines Einzelnen. Bis heute merkt man das, denn Python konnte sich – von einer Community vorangetrieben – ohne den Druck eines Marktes entwickeln, der auf Unternehmen und Konsortien einwirkt und sie behäbig macht.

An C# erkennt man dies sehr gut. Diese Sprache wurde 2001 von Microsoft entwickelt und enthält immer noch *goto*-Statements – die Edsger W. Dijkstra in seinem Artikel »Go To Statement Considered Harmful« bereits 1968 öffentlich schmähte. Um den Markt nicht unnötig zu verwirren, darf eine neue Programmiersprache nicht zu abgefahren aussehen, sonst lehnen Programmierer sie ab. C# sieht daher aus wie Java und enthält Altlasten wie das *goto*-Statement.



Verstehen Sie uns nicht falsch – sowohl Johannes als auch Horst programmieren für ihr Leben gerne und sind große Fans von C#. Jedoch entstammt Python einer anderen Kultur und bringt ein bestimmtes Gedankengut mit, das uns bis heute positiv beeinflusst.

Welche Sprachfeatures bringt Python mit?

Natürlich ist auch Python nicht im luftleeren Raum entstanden und borgt sich Konzepte sowie Syntax von anderen Programmiersprachen. Allerdings gibt es in Python nur Sprachfeatures, die die Community haben wollte. Sie werden sich daher sicher wundern, dass es manche Features in Python gar nicht gibt, die Sie aus anderen Sprachen kennen, wie etwa statische Typisierung, Switch-Case-Anweisungen, Inkrement-Operatoren (zum Beispiel `++`) oder ein Semikolon am Ende der Zeile.

Sie werden sich aber auch wundern, was Python stattdessen mitbringt:

- ✓ Alles ist ein Objekt
- ✓ Alles ist dynamisch
- ✓ Code wird durch Einrücken gruppiert, nicht durch geschweifte Klammern

- ✓ Viele Operatoren sind Schlüsselworte statt Symbole
- ✓ List Comprehensions
- ✓ Generatoren
- ✓ Dekoratoren

Natürlich hat auch Python einige seltsame Eigenarten. Beispielsweise wussten die Entwickler früher nicht, ob sie `abkürz`, `unter_striche` oder `camelCase` zur Benennung verwenden sollten. Auch konnten sie sich oft nicht für ein bestimmtes Programmierparadigma entscheiden (prozedural? objektorientiert?). In der Praxis fällt dies aber kaum ins Gewicht.

Ist Python eine Skript- oder eine Programmiersprache?

Die Antwort auf diese Frage lautet »ja«.

Python wird in der Regel durch einen *Interpreter* ausgeführt, also ist es eigentlich eine *Skriptsprache*. Allerdings wird wie bei Java ein Byte-Code erzeugt und interpretiert. Python ist außerdem eine *Turing-vollständige* Sprache, also können Sie damit theoretisch alles programmieren, was Sie möchten.

Ich habe gehört, dass Python langsam ist. Stimmt das?

Das kommt drauf an, aber eigentlich nicht.

Im normalen Sprachgebrauch bezeichnet Python hauptsächlich die Sprache und der Interpreter ist gleich mitgemeint, allerdings kommt es ganz auf den Interpreter an, wie schnell Ihre Programme später laufen.

Wenn Sie Python von der Stange installieren, bekommen Sie natürlich einen (in C geschriebenen) Interpreter mitgeliefert. Dessen Anspruch ist vor allem, eine gut verständliche Referenz zu bieten, und nicht so sehr, Geschwindigkeitsrekorde zu brechen. Allerdings müssen Sie diesen auch gar nicht nutzen.

Manche Interpreter bringen einen Just-in-Time-Compiler mit, der die Ausführung beschleunigt (zum Beispiel *PyPy*), und es ist durchaus möglich, Python-Code zu kompilieren.

Die Ausführung durch einen Interpreter hat prinzipbedingt eine langsamere Geschwindigkeit zur Folge, etwa im Gegensatz zu C, C++ oder blankem Assembler, aber selbst wenn Sie mal mit Performance-Problemen zu kämpfen haben, ist es nicht zu spät, die kritischen Teile in einer anderen Sprache zu schreiben. Diesen Weg geht beispielsweise *SciPy*, eine Python-Bibliothek für wissenschaftliche Berechnungen, bei der kritische Stellen in Fortran geschrieben wurden.

Die Sprache selbst ist also nicht langsam, denn es steht Ihnen offen, einen anderen Interpreter zu verwenden. Für die meisten Anwendungsfälle ist aber der Standard-Interpreter schnell genug.

Anwendungsgebiete

Python benutzen alle Leute, die Deadlines haben und guten Code schreiben möchten, ohne sich dabei von komplizierter Syntax und verwirrender Semantik ablenken zu lassen, darunter WissenschaftlerInnen, Web-EntwicklerInnen, TesterInnen und EntwicklerInnen.

Wer verwendet Python?

Verschiedene große Player setzen heute auf Python, darunter etwa Finanzinstitute (Deutsche Bank, Goldman Sachs, PayPal), Medienkonzerne (Netflix), Tech-Firmen (Facebook, Google, Amazon, Microsoft, Dropbox) und Forschungseinrichtungen (NASA, das Deutsche Zentrum für Luft- und Raumfahrt oder Universitäten).

Kann man mit Python Geld verdienen?

Auf jeden Fall! Im Moment herrscht Goldgräberstimmung!

Im Gegensatz zu Java-Projekten sind Python-Projekte rarer gesät, allerdings werden es immer mehr.

Natürlich ist Java immer noch die 0815-Allzweckwaffe, wenn es um Programmierung geht, aber besonders in der Forschung wird verstärkt auf Python gesetzt. Da Python flexibel ist und einfach zu erlernen, gibt es hier viele interessante Projekte zum Thema Künstliche Intelligenz, Machine Learning, aber auch Data Science und Statistik. Da die Schnittmenge dieser Verfahren mit Business-Intelligence-Anwendungen groß ist, kommt Python auch in der freien Wirtschaft zum Einsatz.

Auch der Computerspielmärkte ist in den letzten Jahren zu einer Milliarden-Dollar-Industrie geworden und Python wird oft als Skriptsprache für Computerspiele eingesetzt; auch manche CAD-Anwendungen nutzen Python zur Automatisierung.

Wenn Firmen auch oft nicht ihr Kerngeschäft mit Python erledigen, so wird in Job-Angeboten und Ausschreibungen Python oft als zusätzlich gewünschte Qualifikation aufgeführt. Es gibt also genug für Python-Programmierende zu tun und der Einstieg oder sogar Umstieg lohnt sich.



Sowohl Horst als auch Johannes haben bisher in größeren und kleineren Firmen mit Python gearbeitet und ihren Lebensunterhalt mit Python verdient. Dabei waren Projekte für Verlage, Universitäten, Medienkonzerne, Verbraucherportale und viele weitere.

Was kann ich mit Python machen?

Python ist nicht nur geeignet, um abstrakte Parser für domänenspezifische Sprachen oder Webseiten zu bauen, sondern befähigt Sie genauso gut, in Echtzeitanwendungen Sicherheitslücken einzubauen, indem Sie den Speicher einfach selber verwalten (nur weil es geht, heißt das nicht, dass Sie das tun sollten – aber es geht). Python ist unglaublich vielseitig!

Sollte ich nicht doch lieber eine andere Sprache lernen?

Es spricht nichts dagegen, wenn Sie sich andere Sprachen angucken – das empfehlen wir sogar. Auch dazu finden Sie Ausgaben in der »Für Dummies«-Serie, zum Beispiel »JavaScript für Dummies« von Andy Harris. Nun jedoch, wo Sie das Python-Buch schon in der Hand haben, empfehlen wir Ihnen natürlich, erst mal mit Python anzufangen.