

Warum ist Python so beliebt?

(C) 2019-2021 T.Birnthaler OSTC GmbH

=====

- * Ursprünglich als Lehrsprache konzipiert, daher leicht zu erlernen
(auch für technisch Orientierte ohne formale Programmierausbildung).
- * Syntax ist relativ einfach lesbar/schreibbar und konsistent
(verglichen mit anderen Programmiersprachen wie z.B. C++)
+ Nur 35 Schlüsselworte (--> Bedeutung aller anderen Bezeichner änderbar)
- * Vereinheitlichtes Programmiermodell:
 - + ALLES ist ein OBJEKT (auch Werte elementarer Datentypen,
Funktionen, Datentypen, Klassen, Module)
 - + Variablen sind immer INITIALISIERT
 - + Es gibt nur CALL BY REFERENCE und RETURN BY REFERENCE
 - + Automatische Speicherverwaltung per REFERENCE COUNTING + GARBAGE COLLECTION
 - + DUCK TYPING ("if it looks and behaves like a duck, it's a duck")
 - + Vieles per NAMENSKONVENTION geregelt
 - + Einrückung ist Teil der Syntax und wird erzwungen (BLOCK INDENTATION)
- * Nichts wird verboten,
alles ist möglich,
weniges wird erzwungen,
vieles per Konvention geregelt (PEP 8).
- * Ausgezeichnete Fehlermeldungen zur Übersetzungs- und zur Laufzeit.
- * Darf durch Einbau neuer Fähigkeiten und Eigenschaften nicht langsamer werden
(lehrreiche Erfahrung aus dem Übergang von Python 2 nach 3).
- * Multiparadigmen-sprache: Unterstützt strukturierte, objektorientierte,
funktionale, deklarative und aspektorientierte Programmierung.
- * Viele moderne Programmierkonzepte werden unterstützt:
 - + Iteratoren, Generatoren und Koroutinen
 - + Metafunktionen / Higher-order Funktionen
 - + Closures
 - + Dekorateur
 - + lambda, map, filter, reduce
 - + Asynchrone Programmierung
 - + Introspection, Reflection, Selfdescription
 - + Deskriptoren
 - + Name Spaces
 - + Kontext Manager
 - + Operator Overloading
 - + Properties
 - + Wrapper-Funktionen und Currying
 - + Abstract Base Classes
 - + Metaklassen
 - + Dynamische Klassen (Monkey Patching)
 - + (Multiple) Inheritance
 - + Traits und Mixins
 - + Type Annotation / Type Hints
 - + Comprehensions
 - + Dynamic Typing
 - + Meta-Programming
 - + Positional and Keyword Arguments
 - + Exception Handling
 - + Domain Specific Languages (DSL)
 - + ...
- * Umfangreiche und ausgereifte Standardbibliothek wird mitgeliefert.
("Batteries included")
- * Python und seine Standardbibliothek sind portabel.
- * Umfangreiche und ausgezeichnete Online-Dokumentation der eigentlichen
Sprache, der Standardbibliothek sowie des Entwicklungsprozesses (PEPs).
- * Systematische Programm-Dokumentation per "Docstring" eingebaut (pydoc).
- * Über 30 Jahre weiterentwickelt und gereift.
- * Lizenz ist sehr freizügig (MIT),
Community agiert sehr vernünftig,
eine Kommerzialisierung ist nicht zu befürchten.
- * Der Umfang der Python-Laufzeitumgebung ist relativ klein (etwa 200 MB),
sie lässt sich somit leicht ausliefern und in andere Systeme einbetten.
- * In anderen Programmiersprachen geschriebene Bibliotheken lassen sich
relativ leicht mit einer Python-Schnittstelle ausstatten und nutzen

(man merkt gar nicht, dass man Fremdcode aufruft).

- * "Lingua Franca" zur Nutzung vieler Bibliotheken.
("Glue Language" = Bindeglied zwischen allem).
- * Zentrales Repository PyPi mit frei verfügbaren Bibliotheken, aus dem diese einfach inkl. Abhängigkeiten installiert werden können (pip, easy_install).
- * Viele frei verfügbare Bibliotheken für aktuell relevante Themen vorhanden:
 - + AI/KI (Artificial Intelligence/Künstliche Intelligenz)
 - + ML (Machine Learning)
 - + Neuronale Netze (Deep Learning)
 - + Data Science
 - + Statistik
 - + Simulation
 - + Number Crunching (NumPy)
 - + Symbolische Mathematik (SymPy)
 - + Configuration Management
 - + Container Management
 - + Software Deployment
 - + Ticket Systeme
 - + TDD (Test Driven Development)
 - + BDD (Behaviour Driven Development)
 - + IoT (Internet of Things)
 - + Controller (RaspberryPi)
 - + Web Frameworks
 - + GUI Toolkits
- * Ersetzt zunehmend kommerzielle System wie Matlab und Labview.
- * Kritische Masse an Verwendern ist erreicht.
- * War einfach zur rechten Zeit am rechten Ort!