

Presentatie Python

Dit is de begeleidende tekst bij de presentatie over Python (16 september 2017).

Inhoud

- Ontstaansgeschiedenis
- Kenmerken
- Voordelen
- Nadelen
- Voorbeelden
- Beschikbaarheid
- Versie nummers
- Verschillen versies 2 en 3
- Referenties

Ontstaansgeschiedenis

Python is eind 80'er jaren bedacht en geïmplementeerd in C door Guido van Rossum bij de Stichting Mathematisch Centrum, het huidige Centrum voor Wiskunde & Informatica (CWI) in Amsterdam. Hij baseerde zijn ontwerp op de ABC programmeertaal, waar hij bij het CWI ook aan heeft bijgedragen. Hij is de originele auteur van de referentie implementatie van Python.

Tegenwoordig bestaat er een grote Python community van enthousiaste gebruikers. De leden daarvan noemen hem de Benevolent Dictator for Life (BDFL). Of een nieuwe feature wel of niet wordt toegevoegd wordt nog steeds door hem bepaald.

Guido van Rossum publiceerde de eerste release versie 0.9.0 in 1991 met de CWI licentie. Dit is een open source licentie, ook al bestond deze term in die tijd nog niet. Deze eerste versie kende al classes met inheritance, modules, exceptions, functies en basis datatypes zoals list, dict en str.

Versie 1.0 verscheen in 1994. In 2000 verscheen de laatste versie 1.6 in de 1.x reeks. Deze werd in datzelfde jaar opgevolgd door versie 2.0.

In 2001 werd de Python Software Foundation (PSF) opgericht, een niet commerciële stichting. De ontwikkeling werd meer open, anderen konden eenvoudiger bijdragen aan de ontwikkeling van Python. Nog steeds regelt de PSF de voortgang van de ontwikkeling van Python. Sindsdien zijn er velen die bijgedragen hebben, de Python community is groot en groeit nog steeds.

Vanaf versie 2.1 kreeg Python een andere licentie, de Python Software Foundation License. Deze licentie staat het gebruik en verspreiding toe zonder source code, ook voor commercieel gebruik. De licentie is compatible met Copyleft licenties zoals de GNU General Public License (GPL). De PSF is de copyright houder van de Python code en documentatie.

Versie 3.0 verscheen in 2008 ongeveer tegelijk met versie 2.6. De meest recente release is Python 3.6. Versie 3.6.0 is van eind 2016, de meest recente 3.6.2 is van juli 2017. Python 2.7.0 verscheen in 2010, de meest recente 2.7.13 is van eind 2016, binnenkort verschijnt 2.7.14.

eigenaar	jaar	v1	v2	v3	enkele features, zie What's New
CWI	1991	0.9.0			eerste release: classes, modules, exceptions, list, dict, str
CWI	1994	1.0			
CNRI	1999	1.5.2			
CNRI, BeOpen	2000	1.6	2.0		unicode, garbage collection
CNRI, PSF	2001	1.6.1	2.1, 2.2		geneste scopes, weak references, descriptors, generators
PSF	2003		2.3		set datatype, source encoding, zip file import
PSF	2004		2.4		decorators, decimal datatype
PSF	2006		2.5		conditionele expressies, with statement
PSF	2008		2.6	3.0	unicode, print functie, string formatting, migratie
PSF	2009			3.1	ordered dictionary
PSF	2010		2.7		sommige features van Python 3
PSF	2011			3.2	stabiele ABI (application binary interface) voor C modules
PSF	2012			3.3	yield from, virtual environments, exception hierarchy
PSF	2014			3.4	enum, asyncio, ensurepip: voor externe packages installatie
PSF	2015			3.5	coroutines: async en await, matrix @ operator
PSF	2016			3.6	literals: formatted string en underscore
PSF	2017		2.7.14	3.6.2	security

Kenmerken

Python is een script taal en is ontwikkeld om de productiviteit van programmeurs te verhogen.

Standaard Python is een interpreter en is daarom minder snel dan andere gecompileerde talen. Omdat Python modules ook in C kunnen worden ontwikkeld is het snelheidsverschil vaak minder problematisch.

In Python zijn er zijn geen declaraties nodig, de types worden afgeleid uit de context. Python is type-safe en heeft sterke, dynamische types (strong and dynamic). Een ongeldige bewerking resulteert in een exception. Een buffer overflow of een array benaderen verder dan het einde zijn niet toegestaan Dit in tegenstelling tot C of C++ waar zulke fouten goed mogelijk is, vaak valt dit dan onder 'undefined behaviour'.

De programmeertaal Python ondersteunt verschillende manieren van programmeren (paradigma's):

- procedureel, imperatief
- object geïntereerd met multiple inheritance
- functioneel, declaratief

Python wordt met name gebruikt voor:

- systeembeheer, zoals het (in batch) verwerken van bestanden
- wetenschappelijke software, vanwege diverse lang doorontwikkelde modules: Numpy, Scipy, Jupyter notebooks
- websites, via de gestandaardiseerde WSGI interface: Django, Flask, Apache met mod_wsgi.

Je kunt de Python interpreter direct gebruiken als REPL (Read-Eval-Print-Loop). Of je gebruikt een bestand met Python code en voert deze uit door de .py bestandsnaam op de commandline mee te geven.

De naam Python komt van de Britse TV komedie Monty Python's Flying Circus. Er worden in de Python community diverse verwijzingen gebruikt naar dit TV programma, zoals "the Cheeseshop", "Unladen Swallow" of "Spam and Eggs". Er zijn ook verwijzingen naar Python, de soort slang, zoals in het logo van Python, "Anaconda" of "Boa Constructor".

Voordelen

Python code is vaak eenvoudiger, leesbaarder en compacter dan andere programmeertalen voor dezelfde functionaliteit. Zo zijn er geen declaraties nodig, geen braces { } of begin/end statements. Er is minder discussie over de code layout of stijl, omdat de indentatie via spaties en tabs (whitespace) gedaan wordt. Dit alles zorgt voor een hogere productiviteit, een programmeur kan in minder tijd meer functionaliteit in een programma stoppen.

De ingebouwde datatypes zijn: bool, int, float, complex, list, tuple, set, dict, str, bytes, bytearray. Deze hebben relatief weinig beperkingen. Zo kunnen bijvoorbeeld gehele getallen in een int veel groter worden dan 128 bits. Er is geen onderscheid tussen 32 of 64 bits, verschil maken tussen signed of unsigned is niet nodig. Er zijn in Python zelf bijna nooit overflow problemen.

Bij de standaard installatie worden al veel modules en packages meegeleverd, meer dan 200. Python wordt daarom een taal met "batteries included" genoemd. Een package is een aantal bij elkaar behorende modules. Zo bestaat de html package uit de html module en de html.parser en html.entities submodules.

Daarnaast zijn er nog veel meer packages beschikbaar, deze zijn te installeren met het pip programma. Pip gebruikt standaard de Python Package Index website (PyPI, ook wel de Cheeseshop genoemd) via pypi.python.org. Daarop staan vele bekende en onbekende, kleine en grote packages. Op dit moment zijn er meer dan 116.000 packages beschikbaar in PyPI.

Python is portable en beschikbaar op veel platforms. Naast Windows, Mac OS X en Linux ook voor Raspberry Pi. Python bestaat wel maar is minder gebruikelijk op Android en Apple iOS.

De documentatie is uitgebreid en van hoge kwaliteit. De help is als Windows help bestand (.chm) te

downloaden en ook online te raadplegen.

Heel veel aanvullende informatie is te vinden op internet. Er bestaan veel boeken en online websites met Python artikelen, vaak ook gratis. Naast python.org bestaan er veel websites met (ook) Python informatie. Om een probleem op te lossen is het vaak voldoende om even te zoeken op Python met enkele extra keywords.

Nadelen

Standaard is Python niet zo snel, omdat de code door een interpreter wordt uitgevoerd heeft het programma geen hoge performance. Dit kan verbeterd worden door het gebruik van C modules, zoals bijvoorbeeld NumPy.

Python heeft geen goede multithreading vanwege de Global Interpreter Lock (GIL). Python code wordt binnen deze GIL uitgevoerd, waardoor parallel Python uitvoeren in hetzelfde proces niet mogelijk is. Multithreading werkt wel maar is vaak niet echt parallel. Modules geschreven in C kunnen toch echt parallel werken, deze moeten hiervoor wel speciaal aangepast worden. Dit kan ook als voordeel worden gezien: in multithreaded Python code is vaak geen locking nodig.

Python kent geen goede Grafische User Interface (GUI) module. De meegeleverde standaard GUI module is gebaseerd op de Tcl/Tk toolkit. De Python source code die je hiervoor schrijft is niet ideaal. Er bestaan alternatieve modules, gebaseerd op C libraries zoals Gtk, Qt of wxWidgets, maar deze zijn nog niet goed genoeg. De code om zo'n GUI te programmeren is minder 'Pythonic', soms moet je ook de C documentatie nakijken voor meer details.

Verder kent Python een luxe probleem: er is vaak te veel informatie beschikbaar, er zijn dan te veel keuzes. Het is ondoenlijk om alles te weten van Python, de vele modules en wat er allemaal mogelijk is.

Over Python is niet zo veel informatie te krijgen in het Nederlands. Er bestaat wel een gratis Nederlandstalig boek "De Programmeursleerling: Leren coderen met Python 3" geschreven door Pieter Spronck. Dit boek is gratis te downloaden in pdf formaat, ook in het Engels.

De meeste andere programmeertalen hebben hetzelfde probleem. Documentatie voor een programmeertaal is meestal in het Engels. Het Nederlandse taalgebied is maar klein en de meeste Nederlanders kunnen goed genoeg overweg met Engels. Wil je verder met programmeren dan moet je toch goed Engels kennen.

Voorbeelden

Deze voorbeelden zijn uit te proberen door Python via een command line prompt te runnen. Bijvoorbeeld via de command prompt (cmd.exe) op Windows python.exe of py.exe. Ook kan de standaard Integrated Development and Learning Environment (IDLE) gebruikt worden:

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. Alle rechten voorbehouden.
```

```
D:\Test>py
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

>>> print("Hello, world!")
Hello, world!

>>> print(hex(2 ** 142 + 1))
0x4000000000000000000000000000000000000000000000000000000000000001

>>> (3 + 2j) * (2 + 3j)
13j

>>> txt = "z # t 3 A 5 7 2"
>>> lijst = txt.split()
>>> lijst.sort()
>>> txt2 = '.'.join(lijst)
>>> print(txt2)
#.2.3.5.7.A.t.z
```

lets langere programma's kunnen beter als bestand opgeslagen worden en dan op de command line meegegeven worden. Als voorbeeld de code in bestand test.py:

```
with open('test.py') as infile:
    for line in infile:
        print(line.upper(), end = '') # line already has \n
```

Resultaat:

```
D:\Test>py test.py
WITH OPEN('TEST.PY') AS INFILE:
    FOR LINE IN INFILE:
        PRINT(LINE.UPPER(), END = '') # LINE ALREADY HAS \N
```

Beschikbaarheid

Er bestaan verschillende platform edities. Op www.python.org is een installatie download beschikbaar voor Windows en Mac OS X. Voor Windows is er zowel een 32 bits (x86) en een 64 bits (x86-64) variant beschikbaar. Tegelijkertijd een 32 bits en een 64 bits editie van dezelfde major.minor versie gebruiken op dezelfde computer gaat soms niet goed. De 64 bits installatie verwijdert de 32 bits variant. Bij de meeste Linux distributies kun je Python installeren met het gebruikelijke installatie programma, zoals apt of yum. Je kunt ook de source code downloaden en gebruiken bij een installatie.

Er bestaan ook andere distributies van standaard Python buiten die van de PSF. Deze zijn vaak

aangepast, komen met veel meer bijgeleverde modules, of de installatie is eenvoudiger. De meeste distributies zijn voor wetenschappelijk gebruikers. Zo zijn er bijvoorbeeld Anaconda, Python(x,y) of ActivePython.

De standaard Python implementatie van de PSF op www.python.org wordt ook wel CPython genoemd omdat deze is geschreven in C. Omdat Python zo populair is bestaan er daarnaast nog andere Python implementaties zoals:

- PyPy (JIT implementatie, geschreven in Python)
- MicroPython (voor microcontollers, geschreven in C)
- IronPython (voor .NET, geschreven in C#)
- Jython (voor de Java Virtual Machine (JVM), geschreven in Java).

Versienummers

Versies van Python worden aangeduid met major.minor.micro. Soms volgt nog een a (alpha), b (beta) of rc (release candidate) toevoeging, zoals 2.7.14rc1. Van (major) versie 2 bestaan bijvoorbeeld 2.3 of 2.7, en 3.6 is van (major) versie 3. Versie 3.6 is begonnen bij 3.6.0 en is op dit moment bij 3.6.2.

Een major.minor versie kun je installeren en gebruiken naast een versie met een ander major.minor versie. Je kunt dus versie 3.6.2 gebruiken naast versie 3.5.4 of 2.7.13 op dezelfde computer. Een bepaalde major.minor versie kan aanzienlijk verschillen in features ten opzichte van een vorige major.minor versie.

Een gewijzigde micro versie betekent een bug fix versie. Dan zijn er geen verschillen in features. Bij installatie wordt de vorige versie overschreven. Zo verscheen Python 3.6.0 eind 2016. Bij versie 3.6.1 van maart 2017 en de meest recente versie 3.6.2 van juli 2017 zijn er diverse problemen opgelost.

Verschillen versies 2 en 3

Een probleem dat langzaam aan steeds minder wordt zijn de verschillen tussen Python versie 2 en 3. Code geschreven voor Python 3.x is (meestal) niet uitwisselbaar met Python 2.x Omdat er nog zoveel code bestaat voor Python 2 wordt Python 2.7 nog steeds ondersteund. Waarschijnlijk zal dat in 2022 stoppen.

Het is daarom beter om Python 3 te gebruiken, indien mogelijk. Gebruik Python 2 alleen wanneer je nog Python modules wilt gebruiken waarvan geen Python 3 versie bestaat. Meestal zijn dit modules waarvoor al lang geen updates meer worden gegeven. Omdat er dan ook geen bugs worden opgelost is het gebruik meestal niet handig. Dan is het beter om een alternatief te zoeken.

Er zijn vele verschillen tussen versie 2 en versie 3 van Python. Een opvallende wijziging is de manier van gebruik van "print". In Python 2 is print een statement en hoeven er geen haakjes gebruikt te worden In Python 3 is print een expressie en moeten er wel haakjes gebruikt worden. Hieraan kun je vaak herkennen of code nog Python 2 gebruikt:

```
print "hello world" # dit is python 2
```

werkt niet in python 3,

```
print("hello world") # dit is python 3
```

werkt ook in python 2.

Een ander belangrijk verschil zijn de string types en het gebruik van Unicode. Unicode is nuttig voor het werken met karakters in niet westerse talen, zoals Arabisch, Chinees, Grieks, Hebreeuws, Japans of Russisch. In Python 2.x bestaan strings (het str type) nog uit bytes en bestaat er een apart unicode string type. In versie 3.x zijn strings Unicode karakters (code points), en is er een nieuw bytes type. Bij string types die bytes gebruiken moet je goed uitkijken welke encoding er wordt gebruikt. Er bestaan veel encodings zoals ascii, utf8, latin1, iso8859-1. Het door elkaar heen gebruiken van verschillende encodings in hetzelfde programma is soms lastig. Niet elk karakter bestaat in elke encoding. Bij unicode string types en utf encodings zijn wel alle karakters mogelijk. Doordat in Python 3 het standaard string type Unicode is heb je minder problemen met encodings. Toch moet je ook dan goed uitkijken, bijvoorbeeld bij webpagina's die nog andere encodings gebruiken. Het beste is om voor bestanden UTF-8 te gebruiken indien mogelijk.

Referenties

- <https://www.python.org/> de standaard Python website
- <https://docs.python.org/> de uitgebreide Python documentatie
- <https://www.python.org/downloads/release/python-362/> de nieuwste Python 3.6.2 downloads
- <https://pypi.python.org/pypi> de Python Package Index, met meer dan 116.000 packages
- <http://www.spronck.net/pythonbook/dutchindex.xhtml> gratis boek "De Programmeursleerling: Leren coderen met Python 3" geschreven door Pieter Spronck.