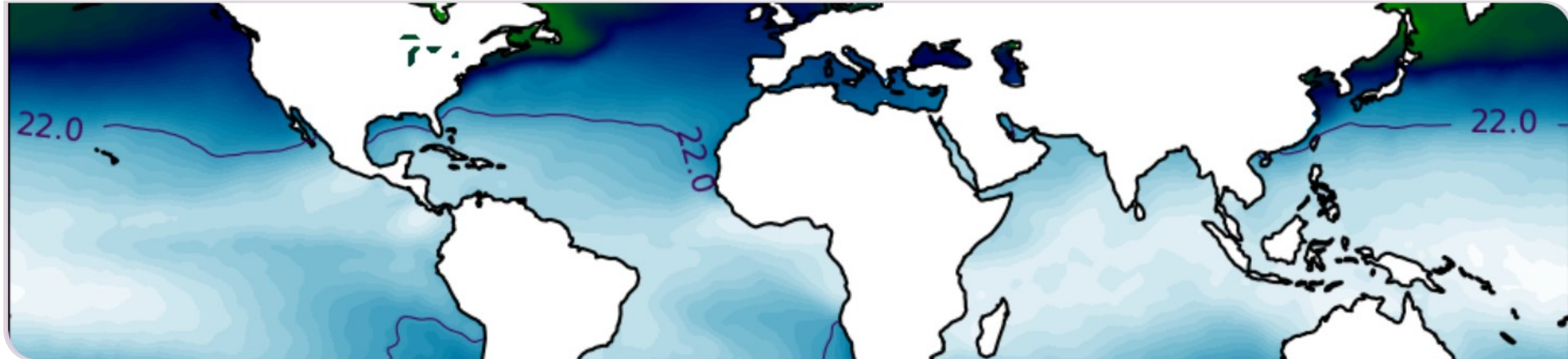


M.Sc. Modul Geodatenanalyse 1

Einführung in Python

Gabriel Rau



Inhalt

- Vorbereitung der Entwicklungsumgebung
- Zahlensysteme
- Datenspeicherung
- Aufbau eines Computers
- Die Python Programmiersprache
- Modularer Aufbau: Bibliotheken und Pakete
- Fehler und „Debugging“ eines Programms
- Umgang mit *Jupyter* Notebooks

Vorbereitung

Anaconda für Python

Anaconda ist erhältlich unter folgendem Link:
<https://www.anaconda.com/products/individual>

Bitte für euer System downloaden und installieren!

Windows

Python 3.8

64-Bit Graphical Installer (457 MB)

32-Bit Graphical Installer (403 MB)

MacOS

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

Linux

Python 3.8

64-Bit (x86) Installer (529 MB)

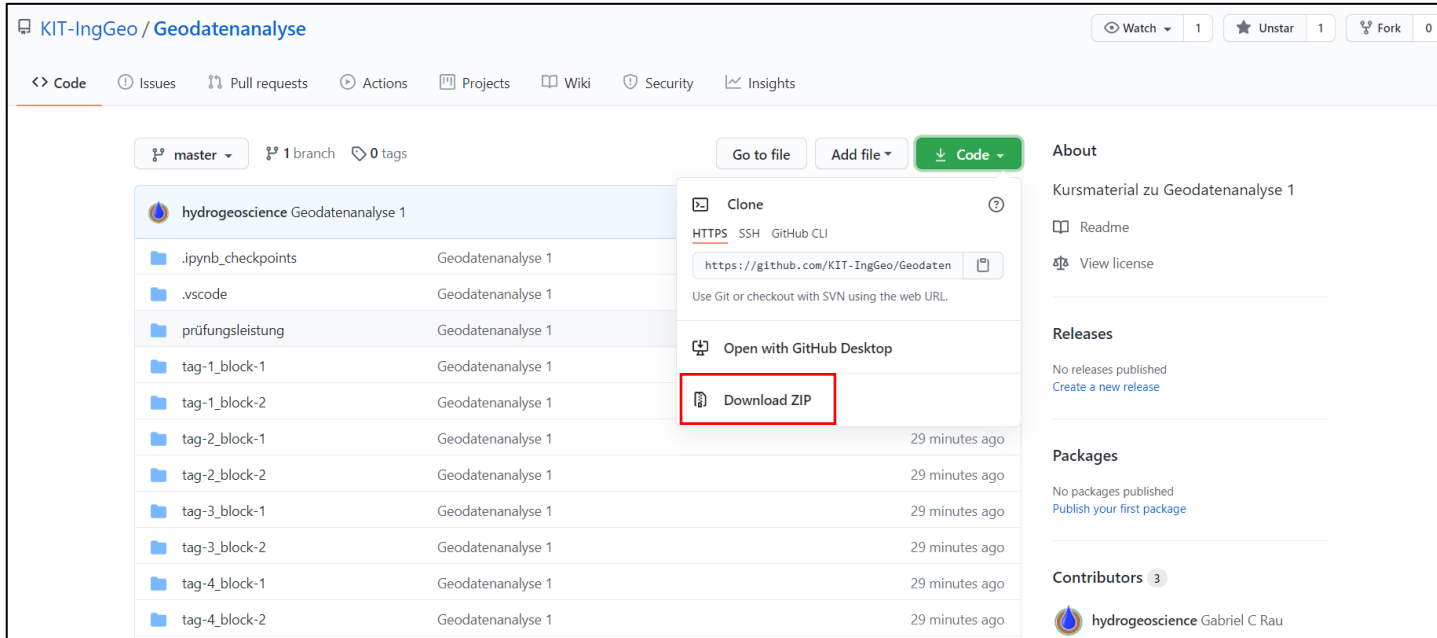
64-Bit (Power8 and Power9) Installer (279 MB)

Kursmaterial Geodatenanalyse 1

- Alles Kursmaterial ist aktualisiert auf *GitHub* erhältlich
- Bitte regelmäßig als ZIP herunterladen und lokal speichern
- **Einmalig:** Zuerst muss die Analyseumgebung eingerichtet werden
- **Für jede Session:** *Jupyter* Server starten
- **Hinweis:** Eine genaue Anleitung hierfür befindet sich auf GitHub
- Kursmaterial: <https://github.com/KIT-IngGeo/Geodatenanalyse>

Kursmaterial Geodatenanalyse 1

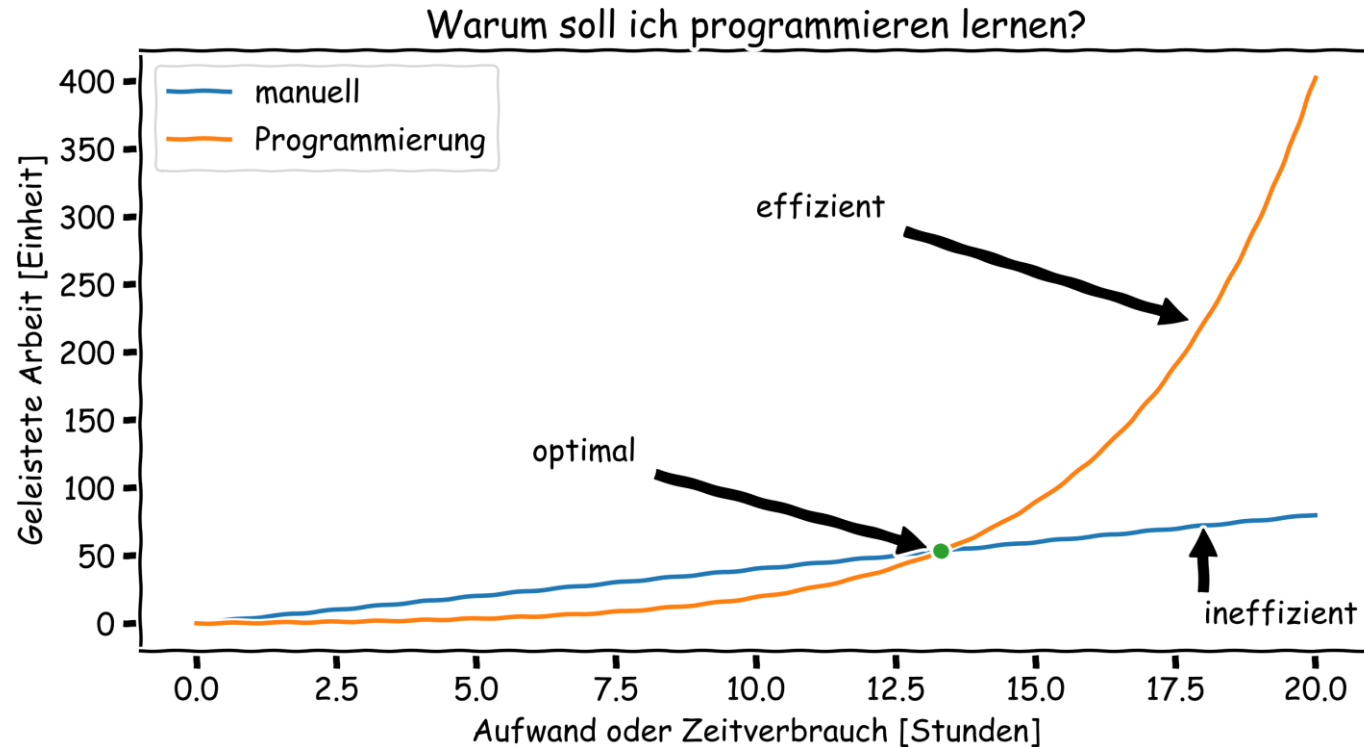
Kursmaterial als ZIP herunterladen



The screenshot shows the GitHub interface for the repository 'KIT-IngGeo / Geodatenanalyse'. The 'Code' tab is selected, displaying a file list on the left and a download menu on the right. The file list includes folders like '.ipynb_checkpoints', '.vscode', 'prüfungslleistung', and several 'tag-1_block-1', 'tag-1_block-2', 'tag-2_block-1', 'tag-2_block-2', 'tag-3_block-1', 'tag-3_block-2', 'tag-4_block-1', and 'tag-4_block-2' folders, all dated '29 minutes ago'. The download menu on the right offers options: 'Clone' (with HTTPS, SSH, and GitHub CLI links), 'Open with GitHub Desktop', and 'Download ZIP' (which is highlighted with a red box). The right sidebar contains sections for 'About', 'Releases', 'Packages', and 'Contributors'.

Einführung

Warum programmieren ...?



Zahlensysteme

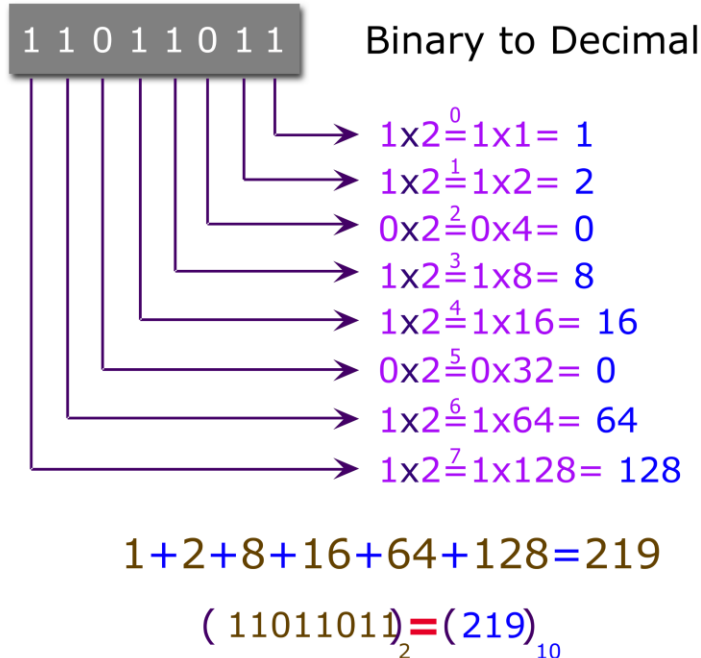
Zehnersystem – Basis 10 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

- Haben wir in der Schule gelernt
- Verwenden wir jeden Tag

Binärsystem - Basis 2 (0, 1)

- Das Rechensystem eines Computers
- Nur 0 (*False*) oder 1 (*True*) möglich
- Kann durch mehrere “Bits” erweitert werden
- Ein Byte = 7 Bits, z.B. 1010010

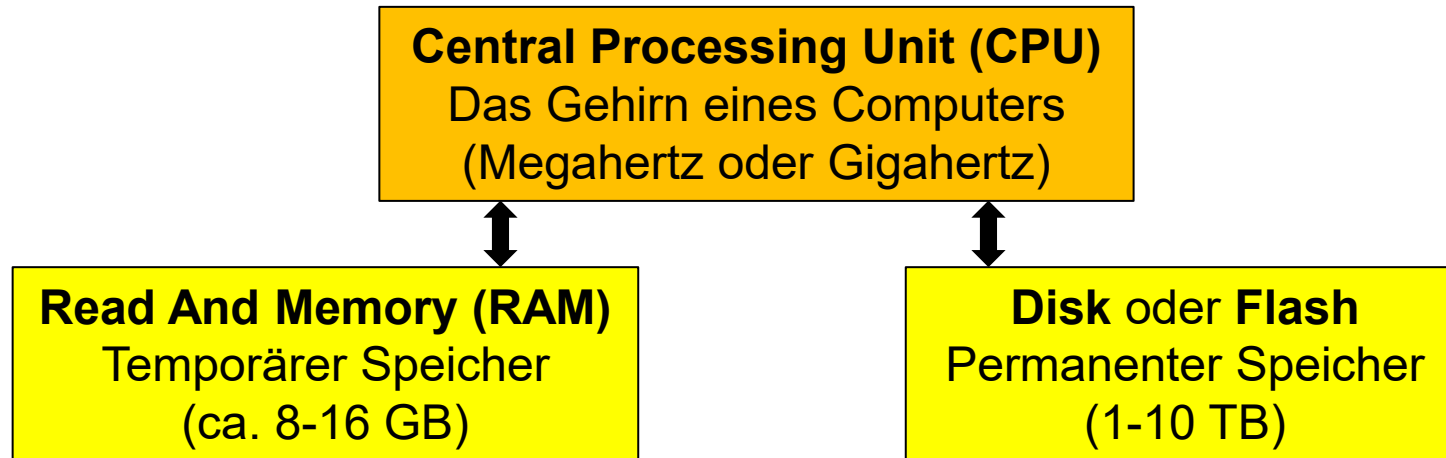
Umrechnung zwischen Binär- und Zehnersystem



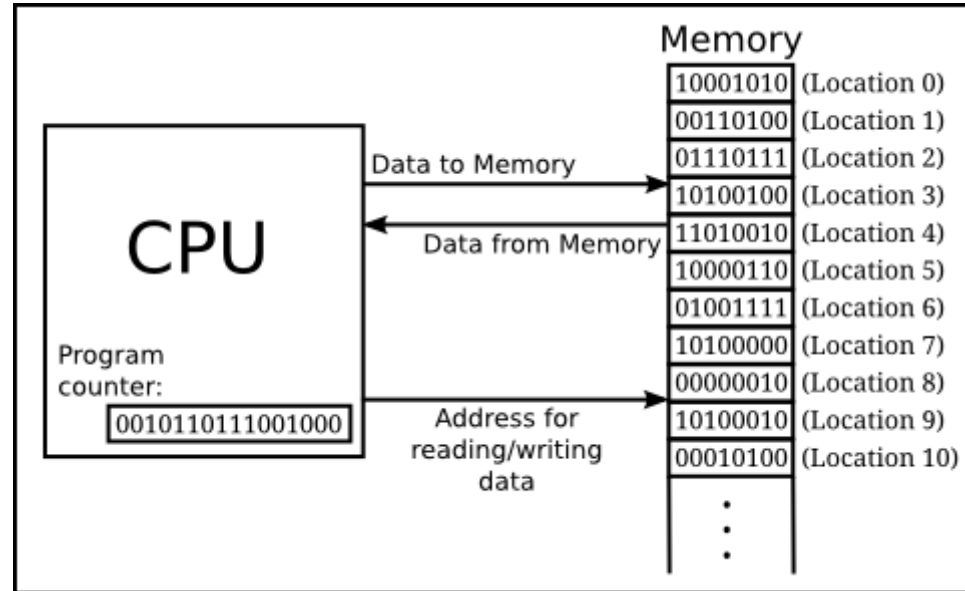
- Alle Prozesse in einem PC laufen im Binärsystem ab
- Für die Interaktion benötigt es also eine Umrechnung zwischen beiden Zahlensystemen
- Glücklicherweise passiert das automatisch
- Allerdings ist ein wenig Wissen sehr sinnvoll

Funktionen eines Computers

Überblick über die Funktionsweise eines Computers



Wie arbeitet ein Computer?



Quelle: [HWS Maths and Computer](#)

1 I DC3 ä; ± SUB ä NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL
 NUL NUL NUL FF NUL NUL NUL pÿÿÿ
 NUL NUL NUL SO NUL NUL NUL SI NUL NUL NUL DLB NUL NUL NUL DC1 NUL N
 NUL NUL NUL NUL NUL NUL SOH NUL NUL NUL STX NUL NUL NUL ETX NUL NUL
 NUL NUL NUL VT NUL NUL NUL FF NUL NUL NUL
 NUL NUL NUL SO NUL NUL NUL SI NUL NUL NUL DLB NUL NUL NUL DC1 NUL N
 NUL NUL NUL EOT SOH NUL NUL VT NUL NUL NUL DLB SOH NUL NUL
 NUL NUL NUL ES SOH NUL NUL FF NUL NUL NUL (SOH NUL NUL SO NUL NUL N
 ?) Eu°; DLB=ii1i-¿æ5ðEþSTXBEFFDC4p>@yÑZfDC1 K i^êYENQæðð°
 ßÜ0qRsd
 OG;Hs=Vt2(S'ÄÄ(Ü qäh°ðiIQ1i¿EOTCZG_1-@JwgðsÈ íf, (SOH-°;
 R3ä,, _ETXSOHCAN=\\4cHKÜ+YÖ
 ?CY;USB°Pð°QRFç£ä¿@C-VFFNAKIM-ES'÷ICIdRSFSK°kNaUe2x-Íé
 ßæ°ZCSHN;GÊETBYUJ B,,ÜÏpu
 ?;Q?è;;j¿)°ÄqUAÄBÓ;CANØVT \ù.ÑiNULSÍ°xy+-ETXsö6h iYêw:G
 ZNUL°nKBEFFBSLOZþBEFF-e*frhí27è,,°8iYEOTI°ENQ?FEIjÊæ"xêv
 ETXø0DLBFFSOH...""°Sj[ÄiNULÜ2ä,,È[E°A:uYíêéù!°»...pB]
 FaEY°OT°p8apÇK>*|ØG|q4ZESUBÄSiIo(ÍxtSOEM7y3)çxßü°çPSI
 þ°6Ä,,.hêfDC3jNÈht-|°?ð°CUBÄ,[ðDETXYðRdøU(BSCðUWÉŽ¿ÊETBäq
 é=ØÈÑðÄÄ-AL~
 fOSUB¿EOT;+HŽ4ùFSŽñ9÷ðSYNENQEM,°Y#1DC3'-pÜi¿!EOTäÄXC
 '°)ÊBELUSÄÄP°aÜ48i!äè?¶æ°FF°-y
 f°æ:µL÷--DC1-ŽŠjÇ°Að;|yK)1B-YANÄNAKVTbYDC3EBCæTN-7wxiiE
 çÄÄt,æERS°ÄS.Ü9Li,q4ZIE+ãBEFFøRS3±æ...°qIgQY;
 æDC1u"ivT#yî!°p°,µ-

Übersetzung zwischen Computer und Mensch

Übersetzungstabelle von binär in menschenlesbare Zeichen

■ ASCII = American Standard Code for Information Interchange

Decimal - Binary - Octal - Hex – ASCII Conversion Chart																			
Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r



Was ist *Python*?

Python ([ˈpʰaɪθn], [ˈpʰaɪθən], auf Deutsch auch [ˈpʰyːtɔn]) ist eine universelle, üblicherweise interpretierte, höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern. So werden beispielsweise Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert.

Quelle: [Wikipedia](#)

Hintergrund zu *Python*?

- Erscheinungsjahr 1991
- Designer oder Vater ist *Guido van Rossum* (Holland)
- Plattformunabhängig
- Momentane Version ist 3.9
- Lizenz: *Python Software Foundation*
(ähnlich der GNU General Public)

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

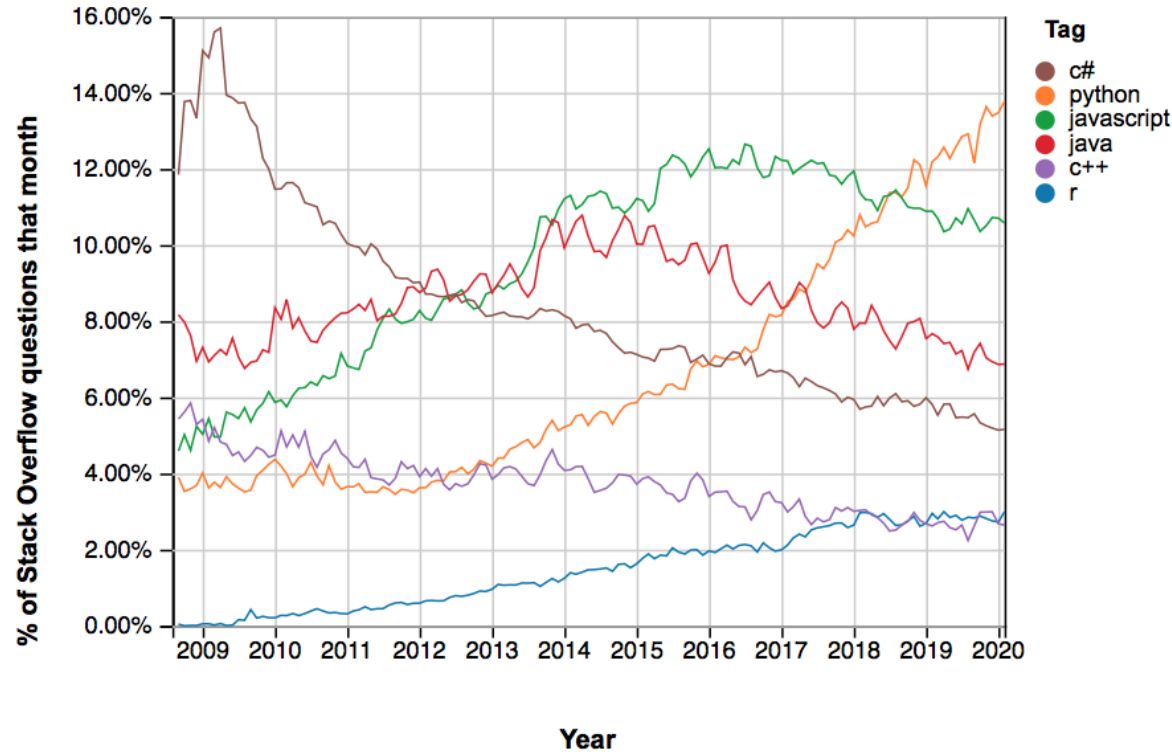


Quelle: [Wikipedia](#)

Warum *Python*?

- Es ist eine alte und weit entwickelte Sprache
- Sehr einfache und intuitive Syntax – einfach zu lernen
- Große Einsatzbandbreite – vom Scripting bis zur Softwareentwicklung
- Flexible Implementierung von Paketen
- Nicht gebunden an eine teure Software-Lizenz
- Einsatz von Python in der Datenwissenschaft ist ein wachsendes Feld

Interesse an verschiedenen Sprachen



Modularer Aufbau

- Python ist modular. D.h., Module werden nach Bedarf geladen
- Damit ist es flexibel einsetzbar

Code Beispiel:

```
# ganzes Modul  
import math  
  
# alle Methoden  
from math import *  
  
# einzelne Methoden  
from math import sin
```

Was ist eine Methode?

- Methode (oder Funktionen) sind eine Zusammenfassung an Code für bestimmte Aufgaben

Code Beispiel:

```
# Funktion definieren
def function(a, b, c):
    return a+b+c

# Funktionen verwenden
function(1, 2, 3)

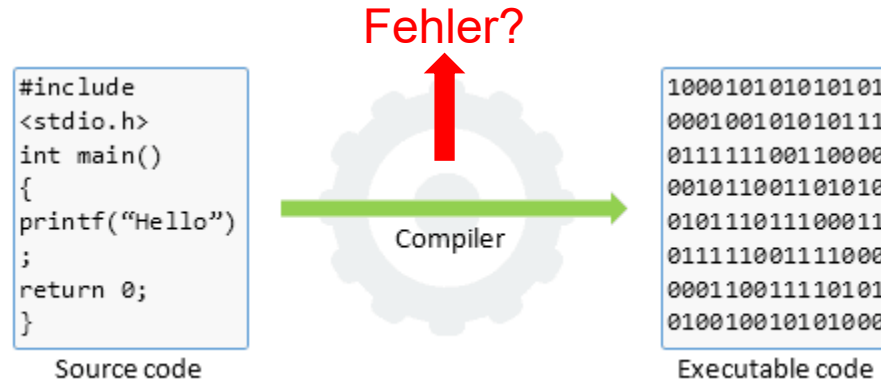
# Ergebnis: 6
```

Was ist ein Compiler?

Ein Compiler (auch Kompilierer; von englisch compile ‚zusammentragen‘ bzw. lateinisch compilare ‚aufhäufen‘) ist ein Computerprogramm, das Quellcodes einer bestimmten Programmiersprache in eine Form übersetzt, die von einem Computer (direkter) ausgeführt werden kann. Daraus entsteht ein mehr oder weniger direkt ausführbares Programm. Davon zu unterscheiden sind Interpreter, etwa für frühe Versionen von BASIC, die keinen Maschinencode erzeugen.

Quelle [Wikipedia](#)

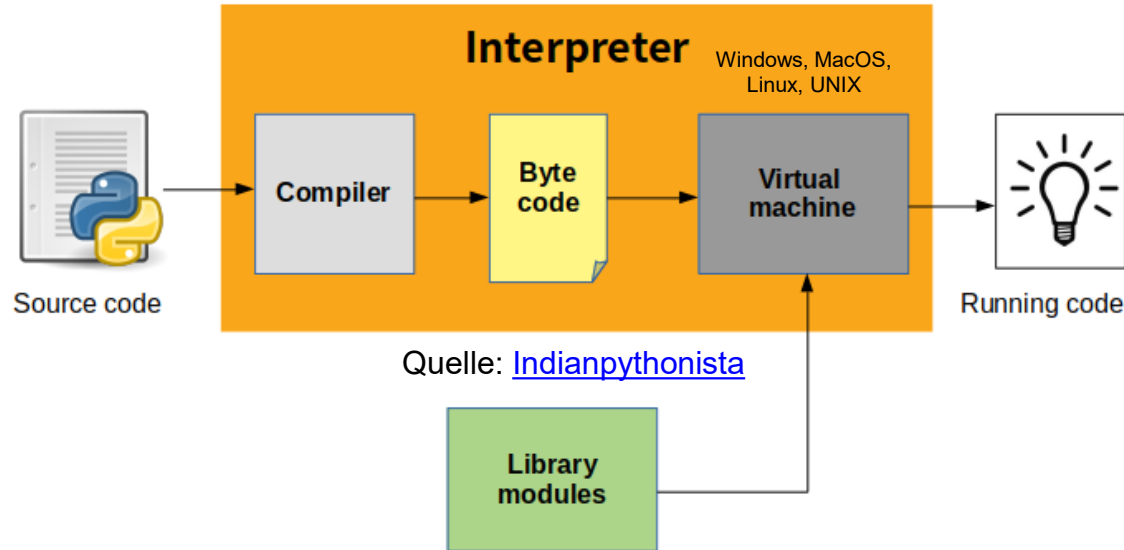
Grobe Funktionsweise eines Compilers



Quelle: [Geeksgod](#)

- Quellcode wird in Maschinencode übersetzt
- Maschinencode ist die binäre Sprache von Prozessoren

Python Übersetzer



- Quellcode wird in Maschinencode übersetzt
- Maschinencode wird zusammen mit Modulen ausgeführt

Was ist bei Fehlern? Debugging ...

- Bei Fehlern stoppt das Programm
- Fehler werden vom Compiler angezeigt
- Fehler sind oft ineinander verschachtelt (engl. „nested error“)

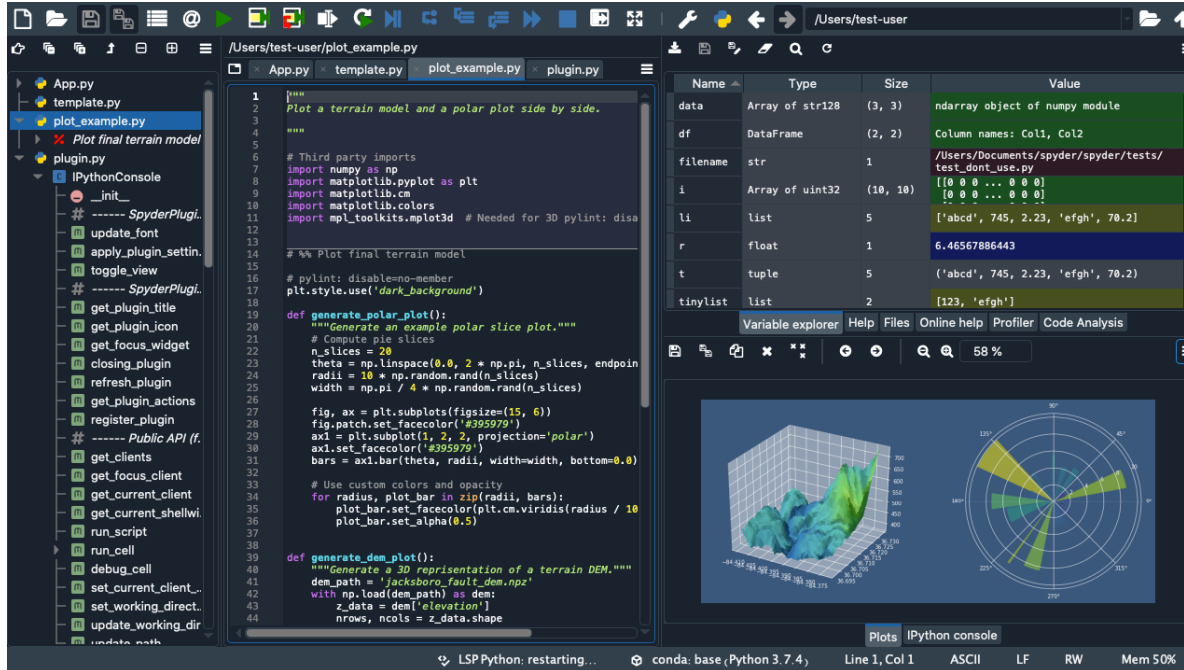
- Fehlersuche = engl. Debugging

- Anfänger verbringen ihre meiste Zeit mit Debugging!
- Wenn die Fehlermeldung nicht hilft, dann bitte Google fragen

Entwicklungsumgebung

- Programme werden am besten in einer Entwicklungsumgebung (engl. *Integrated Development Environment* (IDE) geschrieben
- IDE's Bestandteile:
 - Grafische Benutzeroberfläche
 - Editor zur Textbearbeitung
 - Integrierter Compiler
- Es gibt sehr viele unterschiedliche IDE's
- Jede IDE sieht anders aus und hat Vor- und Nachteile

Spyder IDE



The screenshot displays the Spyder IDE interface with the following components:

- Left Panel (File Explorer):** Shows a project structure with files like `App.py`, `template.py`, `plot_example.py`, and `plugin.py`. The `IPythonConsole` is also visible.
- Editor:** Displays the `plot_example.py` file. The code includes imports for `numpy`, `matplotlib.pyplot`, `matplotlib.cm`, and `mpl_toolkits.mplot3d`. It defines two functions: `generate_polar_plot()` and `generate_dem_plot()`. The `generate_polar_plot()` function generates a 3D surface plot and a corresponding polar plot. The `generate_dem_plot()` function loads a DEM file and generates a 3D surface plot.
- Right Panel (Variable Explorer):** Shows a table of variables in the current namespace. The variables and their values are:

Name	Type	Size	Value
data	Array of str128	(3, 3)	ndarray object of numpy module
df	DataFrame	(2, 2)	Column names: Col1, Col2
filename	str	1	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	Array of uint32	(10, 10)	[[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ...
li	list	5	['abcd', 745, 2.23, 'efgh', 70.2]
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 70.2)
tinylst	list	2	[123, 'efgh']
- Bottom Panel:** Contains the `IPython console` and a `Plots` area. The `Plots` area shows a 3D surface plot of a terrain model and a corresponding polar plot.

<https://www.spyder-ide.org>

Was ist *Jupyter*?

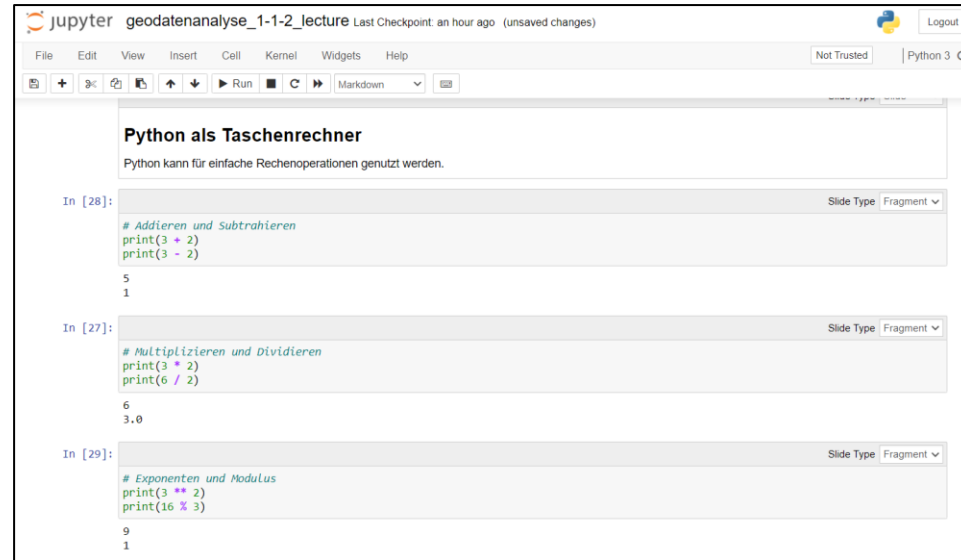
Project Jupyter ([ˈdʒuːpɪtər]) ist eine Non-Profit-Organisation, die gegründet wurde, um „Open-Source-Software, offene Standards und Services für interaktives Arbeiten mit Dutzenden Programmiersprachen zu entwickeln“. Der Name Jupyter bezieht sich auf die drei wesentlichen Programmiersprachen Julia, Python und R und ist auch eine Hommage an Galileos Notizbucheinträge zur Entdeckung der Jupitermonde, bei denen Galileo Beobachtungen und Messdaten zusammen mit Metadaten darstellte.



Quelle: [Wikipedia](#)

Jupyter Notebook

- Jupyter Notebooks sind Notizbücher, welche Programmierung und Erklärungen in einem Arbeitsfluss erlauben
- Zellen können entweder mit Text oder mit Code belegt werden
- Variablen werden von Zelle zu Zelle bewahrt
- Damit lässt sich ein Arbeitsfluss als Notizbuch herstellen
- Man kann ein Notizbuch exportieren



The screenshot displays a Jupyter Notebook titled "geodatenanalyse_1-1-2_lecture". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, navigation, and execution. The notebook content is organized into three code cells, each with a title and a description:

- Python als Taschenrechner**
Python kann für einfache Rechenoperationen genutzt werden.
- In [28]:** *# Addieren und Subtrahieren*
`print(3 + 2)`
`print(3 - 2)`
Output: 5, 1
- In [27]:** *# Multiplizieren und Dividieren*
`print(3 * 2)`
`print(6 / 2)`
Output: 6, 3.0
- In [29]:** *# Exponenten und Modulus*
`print(3 ** 2)`
`print(16 % 3)`
Output: 9, 1

