

# *Praktikum Ingenieurinformatik*

## *Teil 1, Programmieren*

*Termin 1*  
*"Mein erstes Programm"*

# Sicherheitsunterweisung

- Es gilt die Laborordnung
- In den Laboren nicht Rauchen, Essen und Trinken
- Fluchtwege aus dem Labor auf den Flur ins Treppenhaus
- Grüne Fluchtwegemarkierungen an der Flurdecke
- Feuerlöscher auf dem Flur, Feuermelder in beiden Treppenhäusern
- Im Brandfall keinen Aufzug benutzen (möglicher Stromausfall)
- Im Brandfall die Fenster geschlossen halten
- Informationen an den Türen: Verhalten im Brandfall, Notfall-Rufnummern, erste Hilfe

## Ein Unfall – was ist zu tun?

- Verbandskästen in den Räumen B362 , B372 , B0055 (Sekretariat)
- Notausschalter sind in allen KCA-Laboren vorhanden

Bei Hard- und Softwareproblemen (Login nicht möglich, Fragen zu Netzwerklaufwerken, Backups usw.) hilft Herr Schneider, Raum B350

1. Aufbau des KCA-Labors
2. Mein erstes Programm
3. Installation von Anaconda

## C: Lokales Laufwerk

- Festplatte des Rechners, an dem man arbeitet (lokaler Rechner),
- enthält Betriebssystem und Anwendungsprogramme,
- hier **keine Daten speichern**, nur lokal verfügbar, andere Benutzer können Dateien löschen oder ändern!



## U: Userverzeichnis/Userlaufwerk

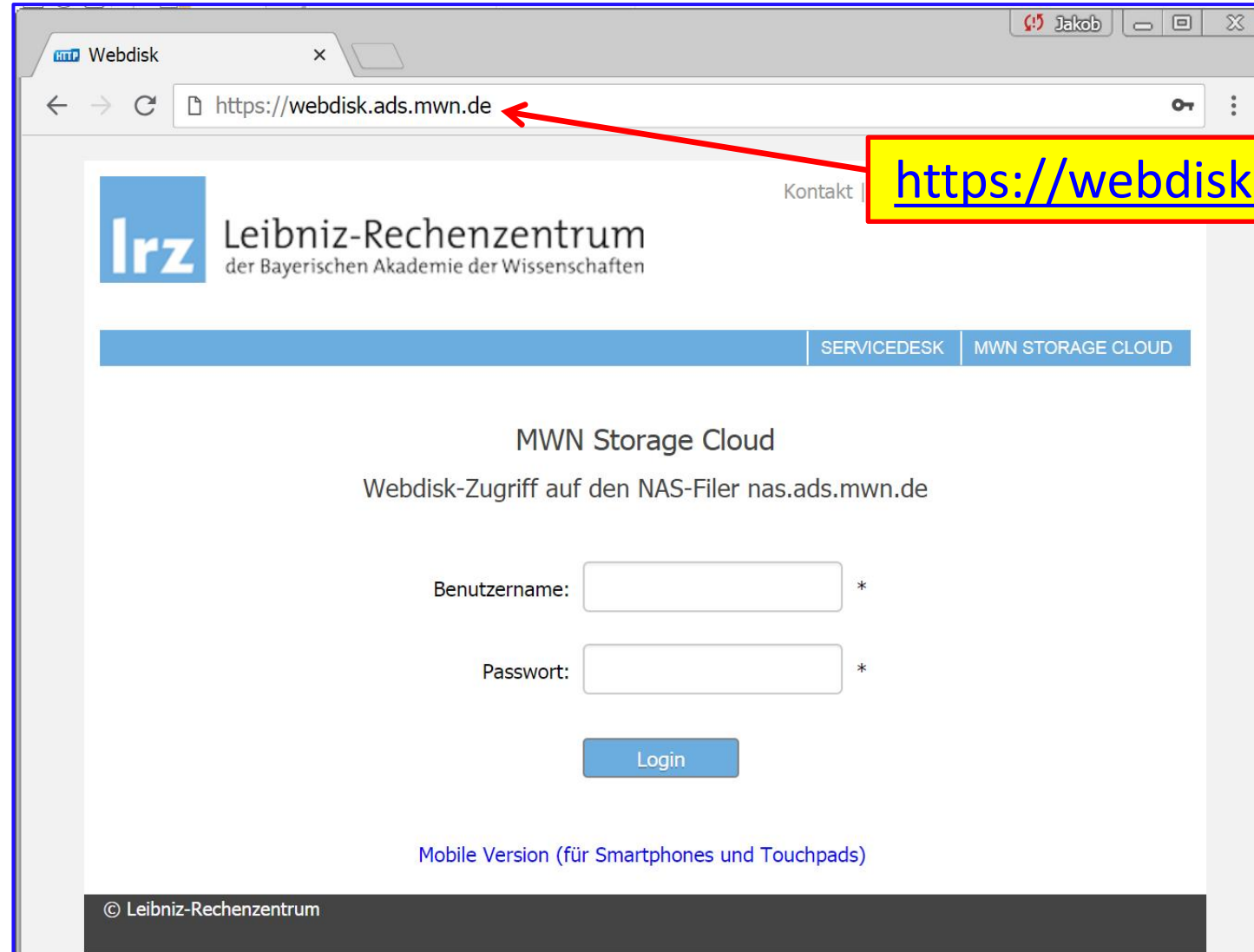
- Von FK03 für jeden Benutzer eingerichtet, jeder Benutzer besitzt sein eigenes U-Laufwerk.
- Das U-Laufwerk ist ein Netzlaufwerk, alle Daten werden regelmäßig gesichert.
- Von jedem Rechner in der FK03 ist ein Zugriff auf das U-Laufwerk möglich.
- Empfehlung: **eigene Daten hier speichern!**

## I: Transferlaufwerk

- Zum Austausch von Daten zwischen Dozenten und Studierenden.

## T: Austauschlaufwerk

- Zum temporären Ablegen oder Austauschen von Daten.



Weitere Infos:

<https://doku.lrz.de/display/PUBLIC/Cloud+Storage>

← → ↻ **Sicher** | <https://webdisk.ads.mwn.de> ☆ ⋮

Ordner Ansicht Suche Datei Vorschau / Bearbeiten Laden Herunterladen Aktualisieren

U: I:

root\Home

Name	Labels	Größe	Änderungsdatum
..			29.8.2018, 14:12:05
\$RECYCLE.BIN			17.4.2018, 14:52:19
cprog	●		29.8.2018, 14:11:58
Desktop			22.8.2018, 13:25:34
Examples			15.5.2018, 12:01:42
matlab			28.6.2018, 15:15:53
Profil	●		29.8.2018, 14:20:11
Visual Studio 2015			27.3.2018, 11:10:31
desktop.ini		402 bytes	22.8.2018, 13:25:35
InfoveranstaltungJuni2018.pdf		186.3 KB	5.6.2018, 10:08:58
quota.txt		6 bytes	1.2.2017, 10:33:37

*Dateien und Verzeichnisse:*  
*Das aktuelle Verzeichnis ist in dieser Abbildung das Verzeichnis **Home**. Das Verzeichnis Home entspricht dem U-Laufwerk.*

Ausgewählt: 6 bytes / 186.7 KBDateien: 1 / 3, Ordner: 0 / 7

1. Aufbau des KCA-Labors
2. Mein erstes Programm
3. Installation von Anaconda

Um Software einfach und schnell zu entwickeln, werden **integrierte Entwicklungsumgebungen** (engl. IDE, Integrated Development Environment) verwendet. Diese enthalten eine Vielzahl von Werkzeugen, die über eine gemeinsame grafische Benutzeroberfläche bedient werden können:

- Texteditor zum Schreiben der Quelltexte,
- Tools zur Projektverwaltung,
- Debugger für die Fehlersuche ...

Es gibt viele **Entwicklungsumgebungen für Python**:

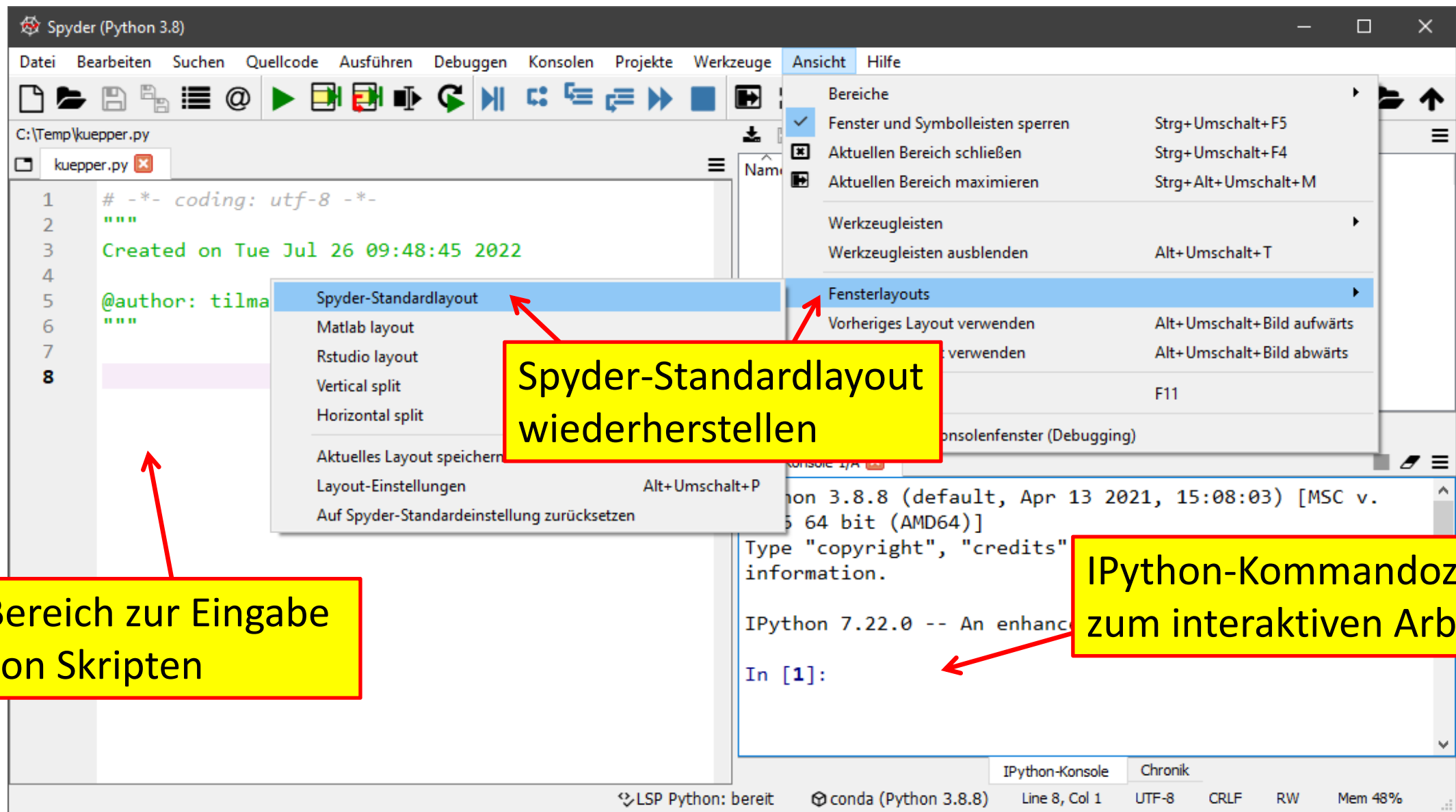
- PyCharm,
- Visual Studio Code,
- Jupyter Notebook,
- Eclipse/PyDev/LiClipse,
- Spyder ...



Spyder ist eine plattformübergreifende, integrierte Open-Source-Entwicklungsumgebung (IDE) für die wissenschaftliche Programmierung in der Programmiersprache Python. Spyder lässt sich mit einer Reihe von Paketen des wissenschaftlichen Python-Stacks integrieren, darunter NumPy, SciPy, Matplotlib, Pandas, IPython, SymPy und Cython, sowie mit anderer Open-Source-Software. Es wird unter der MIT-Lizenz veröffentlicht **und ist Teil der Anaconda-Distribution**.

Quelle: [1]





Führen Sie **verschiedene Berechnungen** aus. Testen Sie alle vier Grundrechenarten, das Potenzieren und das Setzen von Klammern. Kennt Python die Punkt-vor-Strich-Regel?

Name	Typ	Größe	Wert
x	int	1	10
y	int	1	20
z	int	1	120

Konsole 1/A

```
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: print("Hallo")
Hallo

In [2]: 3 + 4 * 5
Out[2]: 23

In [3]: x = 10

In [4]: y = 20

In [5]: z = x ** 2 + y

In [6]: print(z)
120
```

Geben Sie zunächst im interaktiven Bereich (IPython-Kommandozeile) einige Befehle ein.

- Wozu dient der whos-Befehl?
- Wie funktioniert die Hilfefunktion help()?

Das **math-Modul** enthält viele **mathematische Funktionen**. Im math-Modul finden Sie zum Beispiel die Funktion `sqrt()` zur Berechnung von Quadratwurzeln und trigonometrische Funktionen wie `sin()`, `cos()` und `tan()`.

```
In [1]: sqrt(100)
Traceback (most recent call last):

  File "<ipython-input-1-79301f9b64f1>", line 1, in <module>
    sqrt(100)
NameError: name 'sqrt' is not defined
```

Fehler: math-Modul noch nicht importiert!

```
In [2]: from math import *
```

Alle Funktionen des math-Moduls importieren

```
In [3]: sqrt(100)
Out[3]: 10.0
```

```
In [4]: z = sqrt(2)
```

```
In [5]: print(z)
1.4142135623730951
```

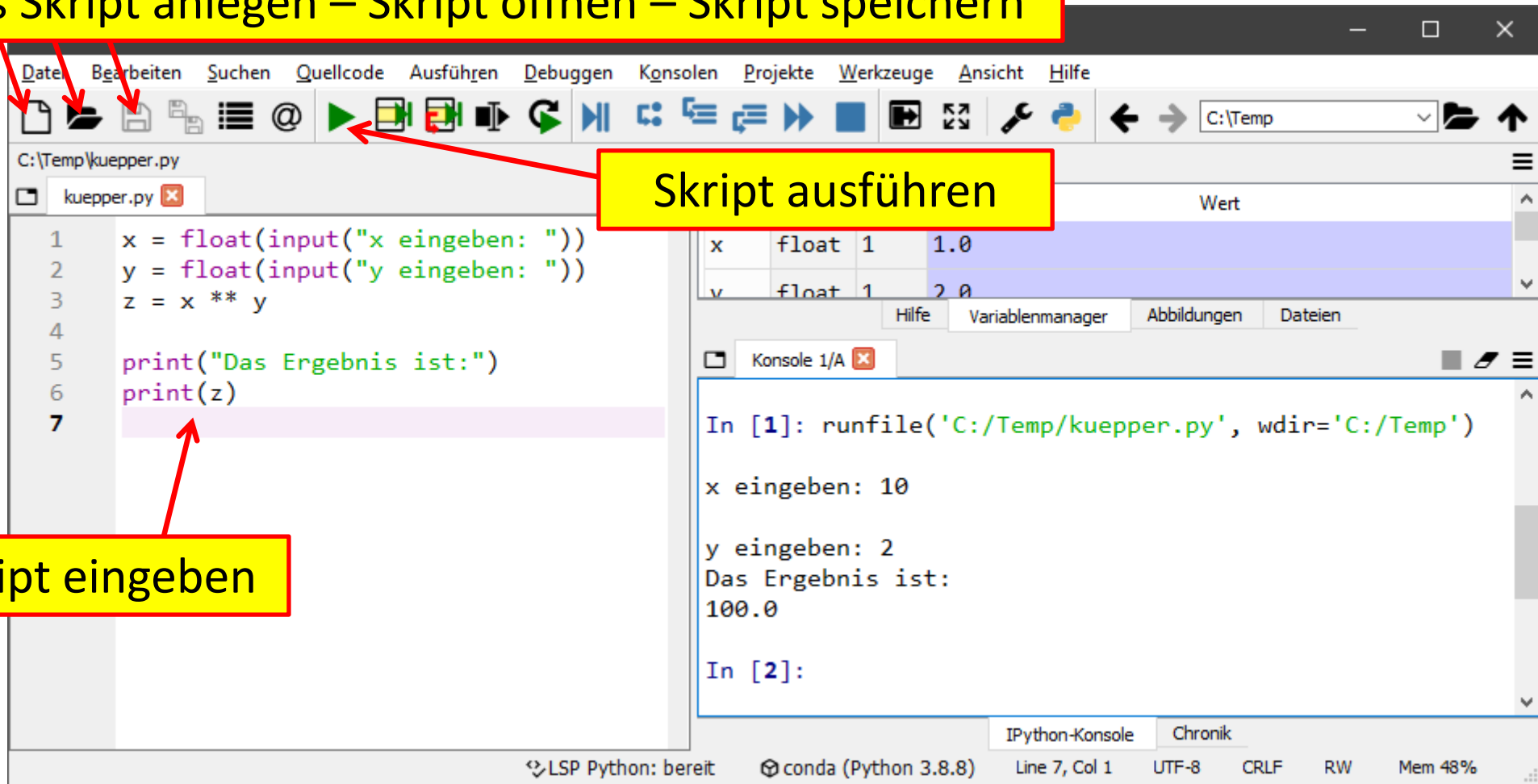
Im **cmath-Modul** gibt es eine zweite Variante der Funktion `sqrt()`, was ist der Unterschied?

Längere Befehls-Sequenzen können Sie als **Python-Skript** eingeben und abspeichern. Dadurch müssen Sie die einzelnen Befehle nicht immer wieder erneut eingeben.

Neues Skript anlegen – Skript öffnen – Skript speichern

Skript ausführen

Skript eingeben



Zur **Ausgabe von Variablen** können **f-Strings** verwendet werden.

Wenn das Skript ausgeführt wird, dann erscheinen hier die aktuellen Werte der Variablen x, y und z.

The screenshot shows the Spyder Python IDE interface. The main editor displays a Python script named 'kuepper.py' with the following code:

```
1 x = 10
2 y = 20
3 z = x + y
4
5 print(f"Das Ergebnis von {x} plus {y} ist {z}.")
6
```

Red arrows point from the yellow callout boxes to the code and the console output. Three arrows point from the top callout to the variables {x}, {y}, and {z} in the print statement. One arrow points from the bottom callout to the output text in the console.

The right sidebar shows the 'Variable Manager' tab, which displays the current values of the variables:

Name	Typ	Größe	Wert
x	int	1	10
y	int	1	20

The 'Console' tab at the bottom shows the output of the script:

```
Das Ergebnis von 10 plus 20 ist 30.
In [2]: |
```

Beachten Sie das kleine "f" vor den doppelten Anführungszeichen.

So sieht das Ergebnis auf dem Bildschirm aus.

# Aufgabe

Erstellen Sie ein neues Skript, das die **Nullstellen von quadratischen Gleichungen** berechnet und auf dem Bildschirm ausgibt:

$$ax^2 + bx + c = 0$$

Zur Berechnung der Nullstellen können Sie die Mitternachtsformel anwenden:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

## Tipps

- Zu Beginn des Skripts werden die Werte von a, b, und c über die Tastatur eingegeben.
- Danach wird zunächst die Nullstelle x1 und dann die Nullstelle x2 berechnet.
- Für die Funktion sqrt() muss das math-Modul importiert werden.
- Testen Sie Ihr Skript für  $a = 1$ ,  $b = 0$ ,  $c = 1$ : Welches Ergebnis erwarten Sie?

## Zusatzaufgabe

Erweitern Sie das Skript zur Nullstellenberechnung:  
Falls **keine reellen Nullstellen** existieren, dann soll  
auf dem Bildschirm eine entsprechende Meldung  
angezeigt werden.

```
In [1]: runfile('C:/Temp/kuepper.py', wdir='C:/Temp')
```

```
a: 1  
b: 0  
c: 1
```

keine reellen Nullstellen

```
Keine reellen Nullstellen!
```

```
In [2]: runfile('C:/Temp/kuepper.py', wdir='C:/Temp')
```

```
a: 1  
b: 0  
c: -2
```

zwei reelle Nullstellen

```
Nullstellen bei x1 = -1.4142135623730951, x2 = 1.4142135623730951
```

```
from math import *  
  
a = float(input("a: "))  
b = float(input("b: "))  
c = float(input("c: "))  
  
if b**2 - 4 * a * c < 0:  
    # Diese Befehle werden ausgeführt,  
    # falls die Bedingung erfüllt ist.  
    ...  
else:  
    # Diese Befehle werden ausgeführt,  
    # falls die Bedingung nicht erfüllt ist.  
    ...
```

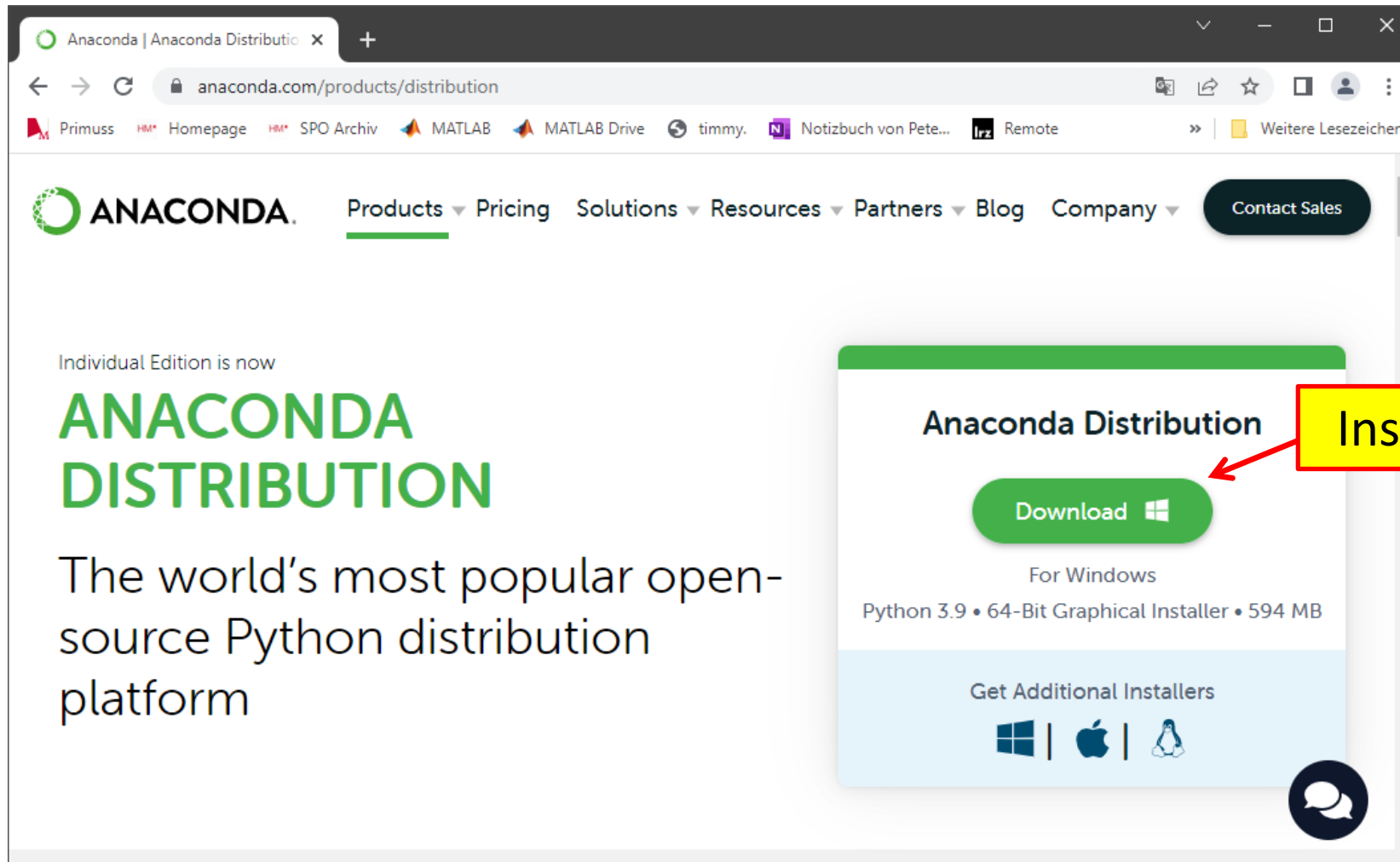
## Tipp

Mit der **if-else-Anweisung** werden  
Bedingungen überprüft: Je nachdem  
die angegebene Bedingung erfüllt ist  
oder nicht, werden unterschiedliche  
Befehle ausgeführt.

1. Aufbau des KCA-Labors
2. Mein erstes Programm
3. Installation von Anaconda



Grundsätzlich ist die Installation von Anaconda (inkl. Python, Spyder usw.) mit dem Installationsprogramm von der Webseite [www.anaconda.com/products/distribution](https://www.anaconda.com/products/distribution) problemlos möglich. Anaconda kann auf Windows-, macOS- und Linux-Rechnern installiert werden.



Installationsprogramm

Bei der **Anaconda-Version vom Mai 2022** (Anaconda3-2022.05) gibt es allerdings ein Problem mit dem input-Befehl. Dieser bringt Spyder zum Absturz, weitere Details dazu finden Sie hier:

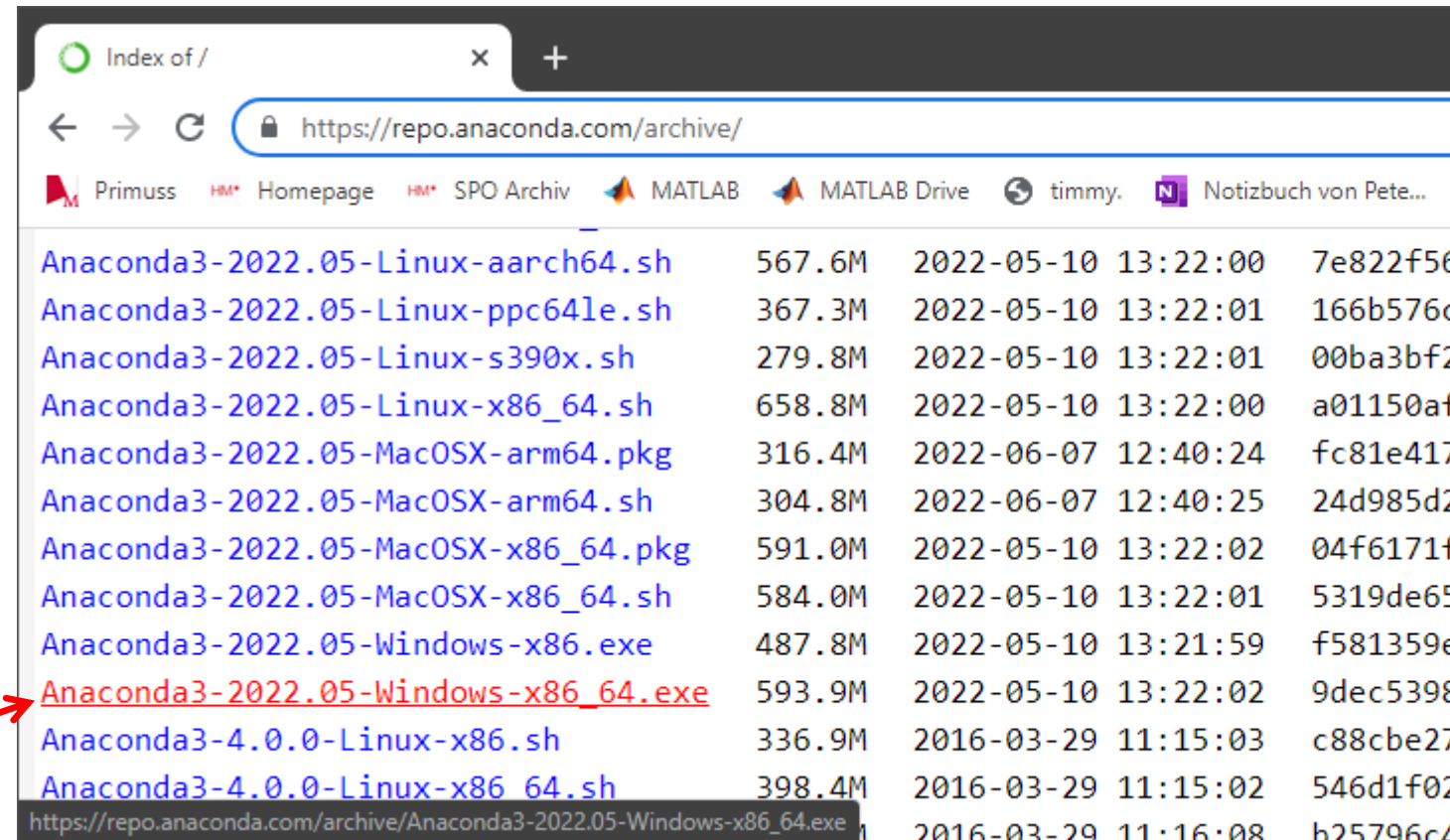
<https://github.com/spyder-ide/spyder/issues/17616>

Installieren Sie daher entweder eine ältere oder eine neuere Anaconda-Version!

Installationsprogramme für ältere Anaconda-Versionen finden Sie im Anaconda-Archiv:

<https://repo.anaconda.com/archive/>

Diese Version besser nicht installieren!



<a href="#">Anaconda3-2022.05-Linux-aarch64.sh</a>	567.6M	2022-05-10 13:22:00	7e822f56
<a href="#">Anaconda3-2022.05-Linux-ppc64le.sh</a>	367.3M	2022-05-10 13:22:01	166b576c
<a href="#">Anaconda3-2022.05-Linux-s390x.sh</a>	279.8M	2022-05-10 13:22:01	00ba3bf2
<a href="#">Anaconda3-2022.05-Linux-x86_64.sh</a>	658.8M	2022-05-10 13:22:00	a01150af
<a href="#">Anaconda3-2022.05-MacOSX-arm64.pkg</a>	316.4M	2022-06-07 12:40:24	fc81e417
<a href="#">Anaconda3-2022.05-MacOSX-arm64.sh</a>	304.8M	2022-06-07 12:40:25	24d985d2
<a href="#">Anaconda3-2022.05-MacOSX-x86_64.pkg</a>	591.0M	2022-05-10 13:22:02	04f6171f
<a href="#">Anaconda3-2022.05-MacOSX-x86_64.sh</a>	584.0M	2022-05-10 13:22:01	5319de65
<a href="#">Anaconda3-2022.05-Windows-x86.exe</a>	487.8M	2022-05-10 13:21:59	f581359e
<a href="#">Anaconda3-2022.05-Windows-x86_64.exe</a>	593.9M	2022-05-10 13:22:02	9dec5398
<a href="#">Anaconda3-4.0.0-Linux-x86.sh</a>	336.9M	2016-03-29 11:15:03	c88cbe27
<a href="#">Anaconda3-4.0.0-Linux-x86_64.sh</a>	398.4M	2016-03-29 11:15:02	546d1f02
<a href="#">https://repo.anaconda.com/archive/Anaconda3-2022.05-Windows-x86_64.exe</a>		2016-03-29 11:16:08	b25796c4

# Quellenverzeichnis

- [1] Seite „Spyder (Software)“. In: Wikipedia – Die freie Enzyklopädie.  
Bearbeitungsstand: 18. April 2021, 15:19 UTC. URL:  
[https://de.wikipedia.org/w/index.php?title=Spyder\\_\(Software\)&oldid=211066881](https://de.wikipedia.org/w/index.php?title=Spyder_(Software)&oldid=211066881)  
(Abgerufen: 27. Juli 2022, 08:23 UTC)