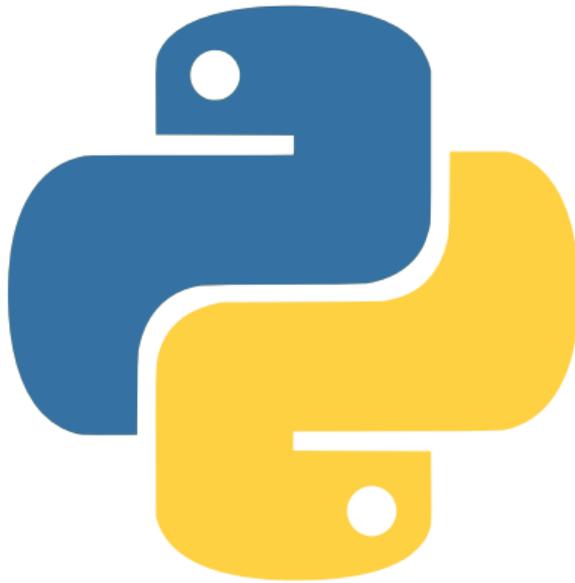


**Universität Stuttgart**

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik  
Göddeke

# Programmierkurs Python

SymPy: Weitere  
Anwendungsbereiche

# SymPy: Weitere Anwendungsbereiche

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten
  - Bestimmung von Ableitungen

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten
  - Bestimmung von Ableitungen
  - Bestimmung von Integralen

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten
  - Bestimmung von Ableitungen
  - Bestimmung von Integralen
  - Lineare Gleichungssysteme

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten
  - Bestimmung von Ableitungen
  - Bestimmung von Integralen
  - Lineare Gleichungssysteme
  - Nichtlineare Gleichungssysteme

# Weitere Anwendungsbereiche

- In diesem Abschnitt: einige Möglichkeiten aus der Schulmathematik
  - Bestimmung von Grenzwerten
  - Bestimmung von Ableitungen
  - Bestimmung von Integralen
  - Lineare Gleichungssysteme
  - Nichtlineare Gleichungssysteme
- Guter Einstiegspunkt zur weiteren Lektüre: **Tutorialsammlung**
  - <https://docs.sympy.org/latest/tutorial>

# Bestimmung von Grenzwerten

# Bestimmung von Grenzwerten

- Funktion `limit()`, folgende Argumente

# Bestimmung von Grenzwerten

- Funktion `limit()`, folgende Argumente
- Funktion, für die ein Grenzwert bestimmt werden soll, als symbolischer Ausdruck

# Bestimmung von Grenzwerten

- Funktion `limit()`, folgende Argumente
- Funktion, für die ein Grenzwert bestimmt werden soll, als symbolischer Ausdruck
- Symbolische Variable, die zum Grenzwert getrieben werden soll

# Bestimmung von Grenzwerten

- Funktion `limit()`, folgende Argumente
- Funktion, für die ein Grenzwert bestimmt werden soll, als symbolischer Ausdruck
- Symbolische Variable, die zum Grenzwert getrieben werden soll
- Wert (symbolisch oder numerisch), gegen den der Grenzübergang erfolgt

# Bestimmung von Grenzwerten

- Funktion `limit()`, folgende Argumente
- Funktion, für die ein Grenzwert bestimmt werden soll, als symbolischer Ausdruck
- Symbolische Variable, die zum Grenzwert getrieben werden soll
- Wert (symbolisch oder numerisch), gegen den der Grenzübergang erfolgt
- Beispiel, benötigt Regel von de l'Hopital

# Codebeispiel

# Bestimmung von Ableitungen

# Bestimmung von Ableitungen

- Funktion `diff()`, folgende Argumente

# Bestimmung von Ableitungen

- Funktion `diff()`, folgende Argumente
- Funktion, für die die Ableitung bestimmt werden soll

# Bestimmung von Ableitungen

- Funktion `diff()`, folgende Argumente
- Funktion, für die die Ableitung bestimmt werden soll
- Symbolische Variable, nach der differenziert wird

# Codebeispiel

# Bestimmung von Stammfunktionen und Integralen

# Bestimmung von Stammfunktionen und Integralen

- Funktion `integrate()`, folgende Argumente

# Bestimmung von Stammfunktionen und Integralen

- Funktion `integrate()`, folgende Argumente
- Funktion, die integriert werden soll

# Bestimmung von Stammfunktionen und Integralen

- Funktion `integrate()`, folgende Argumente
- Funktion, die integriert werden soll
- Integrationsgrenzen und Integrationsvariable, in Form eines Tupels

# Bestimmung von Stammfunktionen und Integralen

- Funktion `integrate()`, folgende Argumente
- Funktion, die integriert werden soll
- Integrationsgrenzen und Integrationsvariable, in Form eines Tupels
- Falls nur ein Symbol angegeben: unbestimmtes Integral

# Codebeispiel

# Lösung linearer und nichtlinearer Gleichungssysteme

# Lösung linearer und nichtlinearer Gleichungssysteme

- Funktion `solve()`

# Lösung linearer und nichtlinearer Gleichungssysteme

- Funktion `solve()`
- Erstes Argument: **Gleichung** (`Equation`, kurz `Eq`)

# Lösung linearer und nichtlinearer Gleichungssysteme

- Funktion `solve()`
- Erstes Argument: **Gleichung** (`Equation`, kurz `Eq`)
  - Symbolischer Ausdruck mit Gleichheitszeichen und rechter Seite

# Lösung linearer und nichtlinearer Gleichungssysteme

- Funktion `solve()`
- Erstes Argument: **Gleichung** (`Equation`, kurz `Eq`)
  - Symbolischer Ausdruck mit Gleichheitszeichen und rechter Seite
- Zweites Argument: symbolische Variable, nach der aufgelöst werden soll

# Lösung linearer und nichtlinearer Gleichungssysteme

- Funktion `solve()`
- Erstes Argument: **Gleichung** (`Equation`, kurz `Eq`)
  - Symbolischer Ausdruck mit Gleichheitszeichen und rechter Seite
- Zweites Argument: symbolische Variable, nach der aufgelöst werden soll
- Codebeispiel: Reproduktion der pq-Formel

# Codebeispiel

# Impressum, Danksagung und Quellen



Stiftung  
Innovation in der  
Hochschullehre



Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, <https://stiftung-hochschullehre.de>

Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie

---

Autor: Dominik Göddeke, IANS, Universität Stuttgart



Weitere Quellen:

- Logos Universität Stuttgart, IANS, SimTech: Universität Stuttgart, alle Rechte vorbehalten
  - Logo Python: <https://freesvg.org/387>, CC-0
  - Logo Stiftung: Stiftung Innovation in der Hochschullehre, alle Rechte vorbehalten
  - Logo ZOERR: Universität Tübingen, alle Rechte vorbehalten
- 



Veröffentlicht auf dem Zentralen OER Repositorium Baden-Württemberg, <https://www.zoerr.de>

---