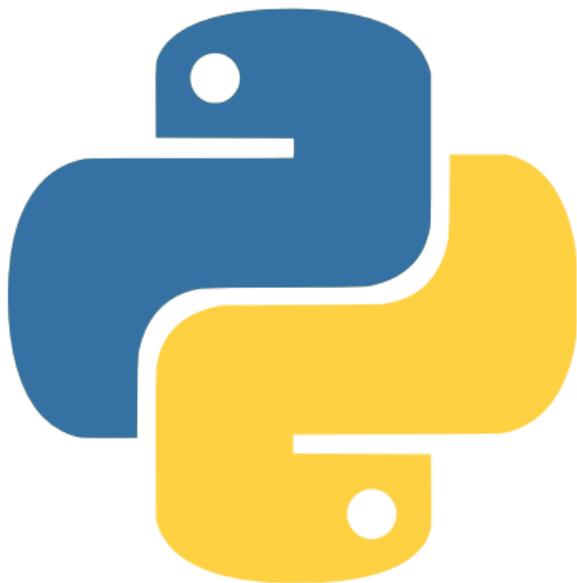


Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

NumPy: Einleitung

NumPy: Einleitung

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen
 - Auswertung von Funktionen in vielen Datenpunkten

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen
 - Auswertung von Funktionen in vielen Datenpunkten
- Beispiele für **Anwendungen solcher Operationen**

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen
 - Auswertung von Funktionen in vielen Datenpunkten
- Beispiele für **Anwendungen solcher Operationen**
 - Schulmathematik: Gleichungen mit mehreren Unbekannten, Schnitt von Geraden und Ebenen

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen
 - Auswertung von Funktionen in vielen Datenpunkten
- Beispiele für **Anwendungen solcher Operationen**
 - Schulmathematik: Gleichungen mit mehreren Unbekannten, Schnitt von Geraden und Ebenen
 - Bildverarbeitung: Freistellen, Kantenschärfung, Deblurring

Einleitung

- **Vektoren** und **Matrizen**: Standardwerkzeug der (numerischen) linearen Algebra und allen darauf aufbauenden Anwendungen
- Bereits in der **Schulmathematik** nicht wegzudenken, auch wenn sie erst in späteren Jahrgangsstufen so genannt werden
- Beispiele für **Operationen mit Vektoren und Matrizen**
 - Matrix-Vektor und Matrix-Matrix Multiplikationen
 - Lösen von linearen Gleichungssystemen
 - Lösen von Eigenwertproblemen
 - Auswertung von Funktionen in vielen Datenpunkten
- Beispiele für **Anwendungen solcher Operationen**
 - Schulmathematik: Gleichungen mit mehreren Unbekannten, Schnitt von Geraden und Ebenen
 - Bildverarbeitung: Freistellen, Kantenschärfung, Deblurring
 - Computerspiele und virtual reality, Navigationsprogramme, Machinelles Lernen

Einleitung



<https://raw.githubusercontent.com/numpy/numpy/main/branding/logo/primary/numpylogo.svg>, BSD 3-clause license, Copyright (c) 2005-2023, NumPy Developers

- All dies bietet das Paket **NumPy**

Einleitung



<https://raw.githubusercontent.com/numpy/numpy/main/branding/logo/primary/numpylogo.svg>, BSD 3-clause license, Copyright (c) 2005-2023, NumPy Developers

- All dies bietet das Paket **NumPy**
- Viel effizienter, als wir das je implementieren könnten

Realisierung mit Bordmitteln

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**
 - Beispiel Gauß-Elimination aus der Schule: Lösungsvektor eines linearen Gleichungssystems nicht komponentenweise berechenbar, ohne permanente Neuanlegung des Vektors in neuer Größe

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**
 - Beispiel Gauß-Elimination aus der Schule: Lösungsvektor eines linearen Gleichungssystems nicht komponentenweise berechenbar, ohne permanente Neuanlegung des Vektors in neuer Größe
- Fokus also nun auf **Listen**, neues Problem

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**
 - Beispiel Gauß-Elimination aus der Schule: Lösungsvektor eines linearen Gleichungssystems nicht komponentenweise berechenbar, ohne permanente Neuanlegung des Vektors in neuer Größe
- Fokus also nun auf **Listen**, neues Problem
 - Keine Operationen wie Vektor-Vektor Addition im Funktionsumfang von Listen

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**
 - Beispiel Gauß-Elimination aus der Schule: Lösungsvektor eines linearen Gleichungssystems nicht komponentenweise berechenbar, ohne permanente Neuanlegung des Vektors in neuer Größe
- Fokus also nun auf **Listen**, neues Problem
 - Keine Operationen wie Vektor-Vektor Addition im Funktionsumfang von Listen
 - Operationen müssen händisch (fehleranfällig!) selbst programmiert werden

Realisierung mit Bordmitteln

- Naheliegende Idee: `list` oder `tuple`
- Prinzipiell geeignet zur **Realisierung von Vektoren in Python**
- Problem: `tuple` **immutable**
 - Beispiel Gauß-Elimination aus der Schule: Lösungsvektor eines linearen Gleichungssystems nicht komponentenweise berechenbar, ohne permanente Neuanlegung des Vektors in neuer Größe
- Fokus also nun auf **Listen**, neues Problem
 - Keine Operationen wie Vektor-Vektor Addition im Funktionsumfang von Listen
 - Operationen müssen händisch (fehleranfällig!) selbst programmiert werden
 - Erwartung an gute Programmierumgebung, insbesondere mit Erfahrung aus dem Kurs bisher: muss es schon geben

Codebeispiele

Diskussion

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen
 - Dynamisch erweiterbar oder kürzbar

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen
 - Dynamisch erweiterbar oder kürzbar
- Bauchgefühl: **overkill** für Vektoren und Matrizen

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen
 - Dynamisch erweiterbar oder kürzbar
- Bauchgefühl: **overkill** für Vektoren und Matrizen
 - Beliebige aber feste Größe

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen
 - Dynamisch erweiterbar oder kürzbar
- Bauchgefühl: **overkill** für Vektoren und Matrizen
 - Beliebige aber feste Größe
 - Einträge immer aus derselben Klasse, üblicherweise `float` oder `complex`

Diskussion

- **Umständliche Realisierung** von Standardoperationen mit Matrizen und Vektoren auf der Basis von Listen
 - Kein $a + b$ zur Addition, sondern Funktionsaufruf
 - Noch OK, wenn in einem Modul weggekapselt
- Schärfere Nachdenken: Liste ein **sehr allgemeiner Datentyp**
 - Einträge mit unterschiedlichen Datentypen
 - Dynamisch erweiterbar oder kürzbar
- Bauchgefühl: **overkill** für Vektoren und Matrizen
 - Beliebige aber feste Größe
 - Einträge immer aus derselben Klasse, üblicherweise `float` oder `complex`
 - Operationen mit (großen) Listen relativ langsam

Lösung

- **Wunschzettel: spezieller Datentyp**

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen
- Und darauf zugeschnittene Sammlung üblicher Operationen

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen
- Und darauf zugeschnittene Sammlung üblicher Operationen
- Tatsächlich: dieser **Datentyp ist Grundlage von NumPy**

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen
- Und darauf zugeschnittene Sammlung üblicher Operationen
- Tatsächlich: dieser **Datentyp ist Grundlage von NumPy**
- `ndarray`, beliebig dimensionales sogenanntes Array

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen
- Und darauf zugeschnittene Sammlung üblicher Operationen
- Tatsächlich: dieser **Datentyp ist Grundlage von NumPy**
- `ndarray`, beliebig dimensionales sogenanntes Array
- Eindimensionales `ndarray`: Vektor

Lösung

- **Wunschzettel: spezieller Datentyp**
- Effizient auch für großen Vektoren und Matrizen
- Und darauf zugeschnittene Sammlung üblicher Operationen
- Tatsächlich: dieser **Datentyp ist Grundlage von NumPy**
- `ndarray`, beliebig dimensionales sogenanntes Array
- Eindimensionales `ndarray`: Vektor
- Zweidimensionales `ndarray`: Matrix, usw.

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>
