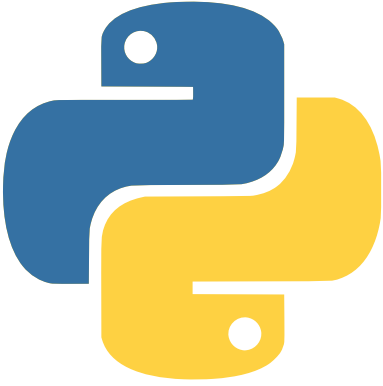
Universität Stuttgart
Projekt digit@L – BOOST. SKILLS. SUPPORT.



Programmierkurs Python

Module, Importe, und die Standardbibliothek

Dominik Göddeke

IANS SimTech

Stiftung Innovation in der Hochschullehre

Module, Importe, und die Standardbibliothek

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie
  - `time` – Zeitmessungen

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie
  - `time` – Zeitmessungen
  - `copy` – Kopieren von mutable Variablen

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie
  - `time` – Zeitmessungen
  - `copy` – Kopieren von mutable Variablen
  - `sys` – Python-Systeminformationen

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie
  - `time` – Zeitmessungen
  - `copy` – Kopieren von mutable Variablen
  - `sys` – Python-Systeminformationen
- Bisher: pragmatische Einbindung mittels `import`

Module, Importe, und die Standardbibliothek

- **Standardbibliothek**: breites Spektrum an Modulen
- Bekannte Beispiele
  - `math`, `cmath` – mathematische Funktionen wie Trigonometrie
  - `time` – Zeitmessungen
  - `copy` – Kopieren von mutable Variablen
  - `sys` – Python-Systeminformationen
- Bisher: pragmatische Einbindung mittels `import`
- Ziel jetzt: **Verständnis von und Umgang mit Modulen**

Codebeispiele: Einbinden von Modulen mit import

Was sind Module?

Was sind Module?

- **Modul**: Sammlung von Funktionen, Definitionen und (später) Klassen

Was sind Module?

- **Modul**: Sammlung von Funktionen, Definitionen und (später) Klassen
- Idee: **Gruppierung** häufig verwendeter und wiederverwendbarer Funktionalität

Was sind Module?

- **Modul**: Sammlung von Funktionen, Definitionen und (später) Klassen
- Idee: **Gruppierung** häufig verwendeter und wiederverwendbarer Funktionalität
- Technisch: ‘lediglich’ Python-Dateien in speziellen Ordnern/Verzeichnissen

Was sind Module?

- **Modul**: Sammlung von Funktionen, Definitionen und (später) Klassen
- Idee: **Gruppierung** häufig verwendeter und wiederverwendbarer Funktionalität
- Technisch: ‘lediglich’ Python-Dateien in speziellen Ordnern/Verzeichnissen
- Modul mit Namen `my_first_module` in Datei `my_first_module.py`

Beispiel

Beispiel

- Anlegen der Datei `my_first_module.py` im aktuellen Verzeichnis (dem Verzeichnis in dem das Notebook liegt), Inhalt:

```
"""
dummy module to demonstrate how to write modules.
"""

from math import exp

def function_in_module():
    print("Hello from the module!")
    print("It's pretty lonely over here!")

def another_function_in_module(x):
    return x*exp(x)
```

So geht das in Python

Diskussion

Diskussion

- Automatische Generierung einer Basis-Dokumentation

Diskussion

- Automatische Generierung einer Basis-Dokumentation
- Eigene Funktionen müssen selbst dokumentiert werden

Diskussion

- Automatische Generierung einer Basis-Dokumentation
- Eigene Funktionen müssen selbst dokumentiert werden
- Importierte Funktionen automatisch in Dokumentation übernommen

Diskussion

- Automatische Generierung einer Basis-Dokumentation
- Eigene Funktionen müssen selbst dokumentiert werden
- Importierte Funktionen automatisch in Dokumentation übernommen
- Deshalb: selektives Importieren statt pauschal alles

Diskussion

- Automatische Generierung einer Basis-Dokumentation
- Eigene Funktionen müssen selbst dokumentiert werden
- Importierte Funktionen automatisch in Dokumentation übernommen
- Deshalb: selektives Importieren statt pauschal alles
- Nutzung des eigenen Moduls genau wie jedes andere

Modulpfade

Modulpfade

- Vordefinierte Liste von Ordnern, in denen Module gesucht werden

Modulpfade

- Vordefinierte Liste von Ordnern, in denen Module gesucht werden
- **Schnell im Code . . .**

Modulpfade

- Vordefinierte Liste von Ordnern, in denen Module gesucht werden
- **Schnell im Code . . .**
- `sys.path`: normale Liste, bspw. `sys.path.append("module_folder")`

Modulpfade

- Vordefinierte Liste von Ordnern, in denen Module gesucht werden
- **Schnell im Code . . .**
- `sys.path`: normale Liste, bspw. `sys.path.append("module_folder")`
- Meist aber komfortabler über die IDE / Python-Umgebung

Laden von Modulen

Laden von Modulen

- Python Interpreter liest bei `import my_module` die Datei einmal

Laden von Modulen

- Python Interpreter liest bei `import my_module` die Datei einmal
- Kennt danach den Inhalt des Moduls

Laden von Modulen

- Python Interpreter liest bei `import my_module` die Datei einmal
- Kennt danach den Inhalt des Moduls
- Zu Demozwecken: Modul `my_module` in Datei `my_module.py` irgendwo im Suchpfad

```
"""
another dummy module
"""
def f(x):
    return x

print("Test")
```

Codebeispiel

Dokumentation von Modulen (direkt in jupyter)

Diskussion: Anwendungsmöglichkeiten für eigene Module

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen
  - Wiederverwendung von allgemeinen Funktionen / Funktionalitäten

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen
  - Wiederverwendung von allgemeinen Funktionen / Funktionalitäten
  - Bereitstellung von Bibliotheken für andere, bspw. in open source Projekten

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen
  - Wiederverwendung von allgemeinen Funktionen / Funktionalitäten
  - Bereitstellung von Bibliotheken für andere, bspw. in open source Projekten
- Wichtig dabei; **sinnvolle Gruppierung** der Funktionen in Module

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen
  - Wiederverwendung von allgemeinen Funktionen / Funktionalitäten
  - Bereitstellung von Bibliotheken für andere, bspw. in open source Projekten
- Wichtig dabei; **sinnvolle Gruppierung** der Funktionen in Module
  - Funktionen zur Ein- und Ausgabe (zum Beispiel Eingabevalidierung, Dateioperationen, spezielle Dateiformate wie CSV oder XML, ...)

Diskussion: Anwendungsmöglichkeiten für eigene Module

- **Beispiele**
  - Kurze und gut lesbare Hauptprogramme durch Auslagern von Funktionen
  - Wiederverwendung von allgemeinen Funktionen / Funktionalitäten
  - Bereitstellung von Bibliotheken für andere, bspw. in open source Projekten
- Wichtig dabei; **sinnvolle Gruppierung** der Funktionen in Module
  - Funktionen zur Ein- und Ausgabe (zum Beispiel Eingabevalidierung, Dateioperationen, spezielle Dateiformate wie CSV oder XML, ...)
  - Häufige benutzte Berechnungsvorschriften und -operationen: Lineare Algebra (Matrizen), numerische Integration, Optimierung, Nullstellensuche, . . .

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

SimTech

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

---

Autor: Dominik Göddeke, IANS, Universität Stuttgart

ZOERR

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