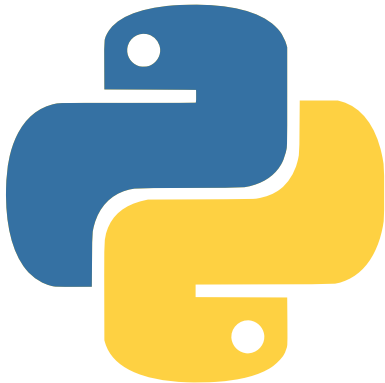




Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

List Comprehensions

List Comprehensions

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**
- Beispiele: Listen, Dictionaries und Mengen

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**
- Beispiele: Listen, Dictionaries und Mengen
- Allgemeine Syntax

`Anweisungen for Element in IterierbaresObjekt if ElementBedingung`

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**
- Beispiele: Listen, Dictionaries und Mengen
- Allgemeine Syntax
`Anweisungen for Element in IterierbaresObjekt if ElementBedingung`
- `if` Teil optional

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**
- Beispiele: Listen, Dictionaries und Mengen
- Allgemeine Syntax
`Anweisungen for Element in IterierbaresObjekt if ElementBedingung`
- `if` Teil optional
- Eröffnen keine neuen Möglichkeiten im Vergleich zu Schleifen

Motivation

- **Comprehensions**: spezielle Möglichkeit für **iterierbare Objekte**
- Beispiele: Listen, Dictionaries und Mengen
- Allgemeine Syntax
`Anweisungen for Element in IterierbaresObjekt if ElementBedingung`
- `if` Teil optional
- Eröffnen keine neuen Möglichkeiten im Vergleich zu Schleifen
- Aber: viele Dinge **sehr bequem und effizient**

List Comprehensions

- In diesem Abschnitt: Comprehension-Ausdrücke über Listen

List Comprehensions

- In diesem Abschnitt: Comprehension-Ausdrücke über Listen
- Allgemeiner Syntax in eckige Klammern eingebettet

List Comprehensions

- In diesem Abschnitt: Comprehension-Ausdrücke über Listen
- Allgemeiner Syntax in eckige Klammern eingebettet
- Macht Sinn: `list = []` Initialisator für leere Listen, `list = [ 1,2,3,4 ]` erzeugt initialisierte Liste

Codebeispiele

Tupel

Tupel

- Im letzten Beispiel: `tuple`

Tupel

- Im letzten Beispiel: `tuple`
- Sinnvoll für **Sequenzen fester Länge**

Tupel

- Im letzten Beispiel: `tuple`
- Sinnvoll für **Sequenzen fester Länge**
- Syntax: Komma-separierte Liste von Variablen, durch runde Klammern eingeschlossen

Tupel

- Im letzten Beispiel: `tuple`
- Sinnvoll für **Sequenzen fester Länge**
- Syntax: Komma-separierte Liste von Variablen, durch runde Klammern eingeschlossen
- Wie Strings: **nicht veränderbar** („immutable“)

Tupel

- Im letzten Beispiel: `tuple`
- Sinnvoll für **Sequenzen fester Länge**
- Syntax: Komma-separierte Liste von Variablen, durch runde Klammern eingeschlossen
- Wie Strings: **nicht veränderbar** („immutable“)
- Später im Detail

Codebeispiel

Ausführlichere Beispiele

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>