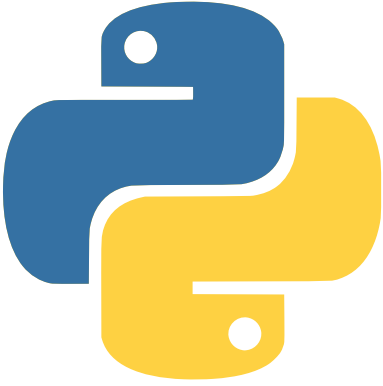
Universität Stuttgart
Projekt digit@L – BOOST. SKILLS. SUPPORT.



Programmierkurs Python

Funktionen und Rekursion

Dominik Göddeke

IANS SimTech

Stiftung Innovation in der Hochschullehre

Funktionen und Rekursion

Funktionen und Rekursion

- **Rekursiver Funktionsaufruf**: Funktion ruft sich selbst auf

Funktionen und Rekursion

- **Rekursiver Funktionsaufruf**: Funktion ruft sich selbst auf
- Wichtig: Abbruch der Rekursion, sonst „Endlosschleife“

Funktionen und Rekursion

- **Rekursiver Funktionsaufruf**: Funktion ruft sich selbst auf
- Wichtig: Abbruch der Rekursion, sonst „Endlosschleife“
- Nett: **Aufpasser**, der endlose Laufzeit verhindert

Codebeispiel

Diskussion

Diskussion

- Ziel: **Anwendungsmöglichkeiten** der Rekursion

Diskussion

- Ziel: **Anwendungsmöglichkeiten** der Rekursion
- Hilfreich: **Analogie zu Induktionsbeweis**

Diskussion

- Ziel: **Anwendungsmöglichkeiten** der Rekursion
- Hilfreich: **Analogie zu Induktionsbeweis**
- Zeige „von Hand“ Gültigkeit des **Induktionsanfangs**, entspricht bei rekursiven Funktionsaufrufen der **Abbruchbedingung** für die Rekursion

Diskussion

- Ziel: **Anwendungsmöglichkeiten** der Rekursion
- Hilfreich: **Analogie zu Induktionsbeweis**
- Zeige „von Hand“ Gültigkeit des **Induktionsanfangs**, entspricht bei rekursiven Funktionsaufrufen der **Abbruchbedingung** für die Rekursion
- Schließe von beispielsweise n auf $n + 1$ **(Induktionsschritt)**, entspricht dem tatsächlichen rekursiven Aufruf

Beispiele

- **Fakultätsfunktion**, rekursive Definition

$$n! = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ n \cdot (n-1)! & n = 2, 3, 4, \dots \end{cases}$$

Beispiele

- **Fakultätsfunktion**, rekursive Definition

$$n! = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ n \cdot (n-1)! & n = 2, 3, 4, \ldots \end{cases}$$

- **Fibonacci-Zahlen**, rekursive Definition

$$\mathsf{fib}(n) = \begin{cases} 1 & n = 1 \\ 1 & n = 2 \\ \mathsf{fib}(n-1) + \mathsf{fib}(n-2) & n = 3, 4, 5, \ldots \end{cases}$$

So geht das in Code

Diskussion

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?
- Schönere Verständnisübung: erwartete Ausgabe vorher überlegen

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?
- Schönere Verständnisübung: erwartete Ausgabe vorher überlegen
- **Probieren Sie es aus!**

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?
- Schönere Verständnisübung: erwartete Ausgabe vorher überlegen
- **Probieren Sie es aus!**
- Fortgeschrittene Verständnisübung: **Laufzeitvergleich** der rekursiven und der iterativen Implementierung

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?
- Schönere Verständnisübung: erwartete Ausgabe vorher überlegen
- **Probieren Sie es aus!**
- Fortgeschrittene Verständnisübung: **Laufzeitvergleich** der rekursiven und der iterativen Implementierung
- Ergebnis: rekursive Version deutlich teurer

Diskussion

- Schöne **Verständnisübung**: Erweiterung der Code-Schnipsel um `print()` Ausgaben: Wann wird welcher Teil des Funktionsblocks mit welchen Argumenten aufgerufen?
- Schönere Verständnisübung: erwartete Ausgabe vorher überlegen
- **Probieren Sie es aus!**
- Fortgeschrittene Verständnisübung: **Laufzeitvergleich** der rekursiven und der iterativen Implementierung
- Ergebnis: rekursive Version deutlich teurer
- Erklärung: zählen Sie die Anzahl der Additionen in beiden Fällen

Direkt im Code
Subtile Fallstricke

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