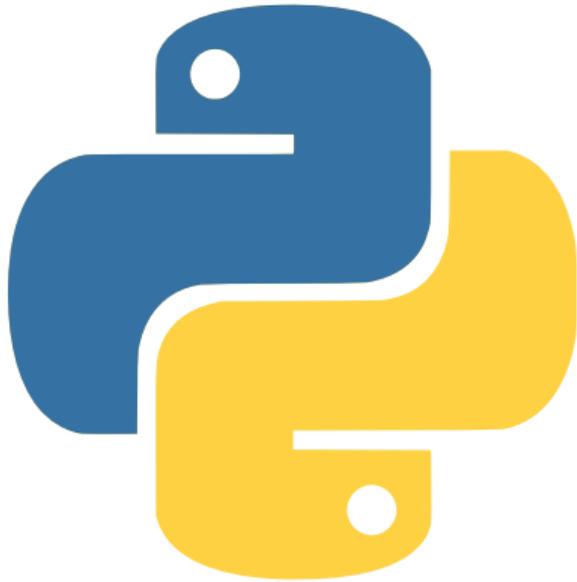




Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

Beispiele: Bedingte Ausführung

Beispiele: Bedingte Ausführung

Beispiel 1: Verbesserung des Primzahltests

- Idee: Kombination des Primzahltests mit Erkenntnis aus letzter Miniübung

Beispiel 1: Verbesserung des Primzahltests

- Idee: Kombination des Primzahltests mit Erkenntnis aus letzter Miniübung
- Insbesondere: `break` basierend auf `if`

Beispiel 1: Verbesserung des Primzahltests

- Idee: Kombination des Primzahltests mit Erkenntnis aus letzter Miniübung
- Insbesondere: `break` basierend auf `if`
- **Direkt zum Code ...**

Beispiel 2: Schaltjahre

- **Definition eines Schaltjahrs**

Beispiel 2: Schaltjahre

- **Definition eines Schaltjahrs**
 - Das Jahr muss durch 4 teilbar sein

Beispiel 2: Schaltjahre

- **Definition eines Schaltjahrs**
 - Das Jahr muss durch 4 teilbar sein
 - Säkulärjahre, also die Jahre, die ein Jahrhundert abschließen sind keine Schaltjahre (Beispiel 1800, 1900, 2100 und 2200)

Beispiel 2: Schaltjahre

- **Definition eines Schaltjahrs**
 - Das Jahr muss durch 4 teilbar sein
 - Säkulärjahre, also die Jahre, die ein Jahrhundert abschließen sind keine Schaltjahre (Beispiel 1800, 1900, 2100 und 2200)
 - Ist die Jahreszahl durch 400 teilbar, so ist es doch ein Schaltjahr

Beispiel 2: Schaltjahre

- **Definition eines Schaltjahrs**
 - Das Jahr muss durch 4 teilbar sein
 - Säkulärjahre, also die Jahre, die ein Jahrhundert abschließen sind keine Schaltjahre (Beispiel 1800, 1900, 2100 und 2200)
 - Ist die Jahreszahl durch 400 teilbar, so ist es doch ein Schaltjahr
- Logik sollte durch `if` und Vergleiche einfach zu implementieren sein

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode
- **Verteilung von Anweisungen** über mehrere Zeilen

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode
- **Verteilung von Anweisungen** über mehrere Zeilen
- Erfordert Klammern um den gesamten booleschen Ausdruck

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode
- **Verteilung von Anweisungen** über mehrere Zeilen
- Erfordert Klammern um den gesamten booleschen Ausdruck
- **Alternative**: Andeutung einer neuen Zeile mit Backslash

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode
- **Verteilung von Anweisungen** über mehrere Zeilen
- Erfordert Klammern um den gesamten booleschen Ausdruck
- **Alternative**: Andeutung einer neuen Zeile mit Backslash
- **Konvention**: Zeilenlänge in Programmen sollte **80 Zeichen** nicht überschreiten

Beispiel 2: Schaltjahre

- Bedingung verspricht **schwer lesbar** zu werden
- Deshalb: **neue Funktionalität** zur Verbesserung der Lesbarkeit von Quellcode
- **Verteilung von Anweisungen** über mehrere Zeilen
- Erfordert Klammern um den gesamten booleschen Ausdruck
- **Alternative**: Andeutung einer neuen Zeile mit Backslash
- **Konvention**: Zeilenlänge in Programmen sollte **80 Zeichen** nicht überschreiten
- **Einrückungen** zur weiteren Verbesserung der Lesbarkeit

Realisierung in Python

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 Repository Baden-Württemberg, <https://www.zoerr.de>