

# ÜBUNG 3

ZUR EINFÜHRUNG IN DIE PROGRAMMIERUNG FÜR COMPUTERLINGUISTEN

# ICH KANN IMMER NOCH NICHT GEDANKEN LESEN

[Start](#) [Termine](#) [Übungen](#) [Links](#) [Fragen](#) [Antworten](#) [Feedback](#)

## Feedback

Hier könnt ihr anonym Feedback zur Tutorübung, zu den Lösungen, zu den Folien usw. geben.

Gib hier deine Anmerkungen/Kritik ein...

Abschicken

<http://www.cip.ifi.lmu.de/~weissweiler/#feedback>



# FEHLERKORREKTUR

- Es stellt sich heraus, dass `==` für alle Datentypen definiert ist
- Es gibt für inkompatible Datentypen (wie `String` und `int`) **false** zurück
- Für `<`, `>` und ähnliches wird trotzdem ein Fehler ausgegeben



# DIDAKTISCHES EXPERIMENT

- [lmu.twbk.de](http://lmu.twbk.de)
- Lesson ID: **8v5**
- Panik-Knopf
- Quiz
- Fragen stellen und auch selber Fragen beantworten



Audience

Speaker

## Participate in a lecture

To participate, please enter the Lesson-ID provided by your docent.



**PARTICIPATE**

# WIEDERHOLUNG: PFADE

■ Welcher der Pfade ist identisch zu `../test/../../abc`?

A. `../test/../../abc`

B. `./../abc`

C. `../../abc`

D. `test/../../abc`



# WIEDERHOLUNG: PFADE

■ Welcher der Pfade ist identisch zu `../test/../../abc`?

A. `../test/../../abc`

B. `./../abc`

C. `../../abc`

D. `test/../../abc`



# WIEDERHOLUNG: PFADE

■ Ist `../test/../../abc` ein relativer Pfad?

A. Ja

B. Nein



# WIEDERHOLUNG: PFADE

■ Ist `../test/../../abc` ein relativer Pfad?

A. Ja

B. Nein





# WIEDERHOLUNG: IF-ANWEISUNG

■ Was gibt das Programm aus?

- A. 1.0
- B. True
- C. False
- D. 1
- E. 2

```
a = 2.0
b = 2

if a/b == 2:
    print (a/b)
elif a*b > 3:
    print (a//b)
else:
    print (a > b)
```



# WIEDERHOLUNG: IF-ANWEISUNG

■ Was gibt das Programm aus?

- A. 1.0
- B. true
- C. false
- D. 1
- E. 2

```
a = 2.0
b = 2

if a/b == 2:
    print (a/b)
elif a*b > 3:
    print (a//b)
else:
    print (a > b)
```



# PIPING

- Mit “|” wird in Bash der output eines Befehls als Eingabe einem anderen Befehl übergeben
- Es kopiert die Ausgabe des linken Programmes und “tippt” sie als Eingabe in das recht ein
- Unendlich verkettbar

```
Leonie@Laptop $ cat inhalt.txt
```

```
c  
a  
f  
b
```

```
Leonie@Laptop $ cat inhalt.txt | sort
```

```
a  
b  
c  
f
```

```
Leonie@Laptop $ cat inhalt.txt | sort | ...
```

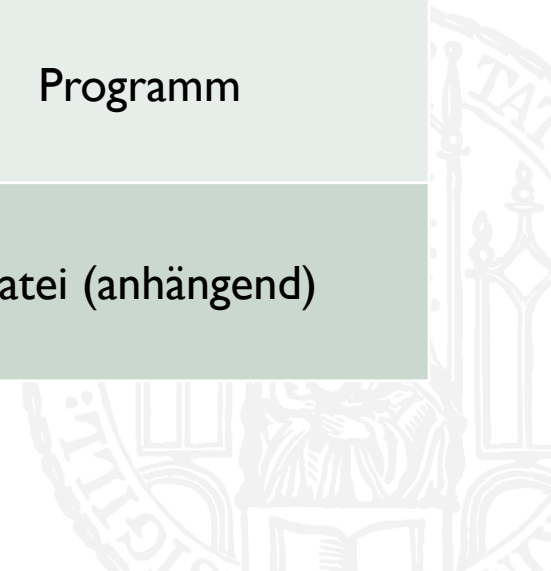
# REDIRECTING

- Mit “>” wird in Bash die Ausgabe des Programmes abgefangen und in eine Datei geschrieben
- Eine evtl. existierende Datei wird **überschrieben**
- `cat datei.txt > datei2.txt`
- Mit “>>” wird in Bash die Ausgabe des Programmes abgefangen und an eine Datei angehängt
- `echo “Hallo” >> begrüßungen.tx`
- `echo “Hi” >> begrüßungen.txt`



# PIPING VS REDIRECTING

Befehl	Piping	Redirecting	Appending
Operator		>	>>
Quelle	Programm	Programm	Programm
Ziel	Programm	Datei (überschreibend)	Datei (anhängend)



# QUIZ

- Was ist der richtige Operator?
  - `cat test.txt ? test2.txt`
- A.** | (Piping)
- B.** > (Redirecting)



# QUIZ

- Was ist der richtige Operator?
  - `cat test.txt ? test2.txt`
- A. | (Piping)
- B. > (Redirecting)



# QUIZ

- Was ist der richtige Operator?
- `cat test.txt ? sort`
- A. | (Piping)
- B. > (Redirecting)





# QUIZ

- Was ist der richtige Operator?
  - `cat test.txt ? sort`
- A. | (Piping)
- B. > (Redirecting)



# QUIZ

- Was ist der richtige Operator?
- `cat test.txt | sort ? sorted.txt`
- A. | (Piping)
- B. > (Redirecting)



# QUIZ

- Was ist der richtige Operator?
  - `cat test.txt | sort ? sorted.txt`
- A. | (Piping)
- B. > (Redirecting)



# WHILE-SCHLEIFE

- While-Schleifen kann man benutzen, um Anweisungen zu wiederholen

- `x = 0`  
`while (x < 5):`  
    `print(x)`  
    `x = x + 1`

- `y = 5`  
`while (y > 0):`  
    `print(y)`  
    `y = y + 1`



# LISTEN

- Bis jetzt kennen wir normale Variablen
- Sie können sich genau eine Sache merken, und wenn wir ihnen eine andere übergeben, vergessen sie die erste.
- Wenn wir jetzt aber mehrere zusammengehörige Werte gleichzeitig in einer Variable speichern wollen, brauchen wir eine **Liste**
- $x_0 = 0$   
 $x_1 = 0$   
 $x_2 = 0$   
 $x_3 = 0$   
 $x \dots$



# LISTEN

```
zahlen = [4,9,42]
```

```
print (zahlen[0])  
4
```

0	1	2
4	9	42

```
zahlen[0] = 11
```

```
print (zahlen[3])  
Fehler
```

0	1	2
11	9	42

```
zahlen[3] = 0  
Fehler
```



# QUIZ ZU LISTENZUGRIFFEN

- Es gibt die Liste:
- `buchstaben = ['c', 'i', 's']`
- Was ergibt der Zugriff `buchstaben[1]`?

- A. 'c'
- B. 'i'
- C. 's'
- D. Fehler



# QUIZ ZU LISTENZUGRIFFEN

- Es gibt die Liste:
- `buchstaben = ['c', 'i', 's']`
- Was ergibt der Zugriff `buchstaben[1]`?

A. 'c'

B. 'i'

C. 's'

D. Fehler





# QUIZ ZU LISTENZUGRIFFEN

- Es gibt die Liste:
  - `buchstaben = ['c', 'i', 's']`
  - Was ergibt der Zugriff `buchstaben[3]`?
- A. 'c'
- B. 'i'
- C. 's'
- D. Fehler



# QUIZ ZU LISTENZUGRIFFEN

- Es gibt die Liste:
- `buchstaben = ['c', 'i', 's']`
- Was ergibt der Zugriff `buchstaben[3]`?

A. 'c'

B. 'i'

C. 's'

D. Fehler



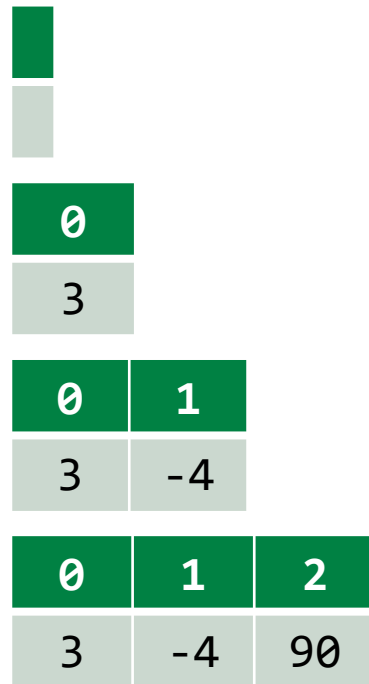
# LISTEN

```
zahlen = []
```

```
zahlen.append(3)
```

```
zahlen.append(-4)
```

```
zahlen.append(90)
```



# LISTEN

```
len(zahlen)
```

3

0	1	2
3	-4	90

```
print(zahlen[-1])
```

90

```
zahl = 3
```

3

```
zahl.append(4)
```

*Fehler*

3



# SLICING

- Mit slicing kann man sich eine “Scheibe” aus einem Array “schneiden”
- Der linke Index wird hierbei “eingeschlossen”, der rechte “ausgeschlossen”
- zahlen = [1, 2, 4, 8, 16, 32]
- zahlen[2:4]
- zahlen[1:-1]
- zahlen[3:]
- zahlen[:-3]

0	1	2	3	4	5
1	2	4	8	16	32



# GROSSES SLICING QUIZ

- Es gibt die List
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[3:6]?
- A.** [9,16,25,36]
- B.** [9,16,25]
- C.** [4,9,16,25]
- D.** [4,9,16]



# GROSSES SLICING QUIZ

- Es gibt die List
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[3:6]?
- A. [9,16,25,36]
- B. [9,16,25]
- C. [4,9,16,25]
- D. [4,9,16]



# GROSSES SLICING QUIZ

- Es gibt die List
- zahlen = [0,1,4,9,16,25,36,49]
- Was ist zahlen[4:]?
  - A. [16,25,36,49]
  - B. [16,25,36]
  - C. [9,16,25,36,49]
  - D. [9,16,25,36]





# GROSSES SLICING QUIZ

- Es gibt die List
- zahlen = [0,1,4,9,16,25,36,49]
- Was ist zahlen[4:]?  
**A. [16,25,36,49]**  
B. [16,25,36]  
C. [9,16,25,36,49]  
D. [9,16,25,36]



# GROSSES SLICING QUIZ

- Es gibt die List
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[: -1]?
- A.** [0,1,4,9,16,25,36,49]
- B.** [0,1,4,9,16,25,36]
- C.** [0]
- D.** [0,1]



# GROSSES SLICING QUIZ

- Es gibt die List
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[: -1]?
- A. [0,1,4,9,16,25,36,49]
- B. [0,1,4,9,16,25,36]
- C. [0]
- D. [0,1]



# MEHR SLICING

- zahlen = [1,2,4,8,16,32]
  - zahlen[::2]
- “jedes zweite”*

0	1	2	3	4	5
1	2	4	8	16	32



# MEHR SLICING

- zahlen = [1,2,4,8,16,32]
- zahlen[::2]
- zahlen[::-1]

*“Jedes erste von hinten”*

0	1	2	3	4	5
1	2	4	8	16	32



# MEHR SLICING

- zahlen = [1,2,4,8,16,32]
- zahlen[::2]
- zahlen[::-1]
- zahlen[:: -2]

*“jedes zweite von hinten”*

0	1	2	3	4	5
1	2	4	8	16	32



# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[::3]?
- A.** [0,4,16,36]
- B.** [1,16,49]
- C.** [0,9,36]
- D.** [4,25]



# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[::3]?
- A.** [0,4,16,36]
- B.** [1,16,49]
- C.** [0,9,36]
- D.** [4,25]





# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[1:6:2]?
- A.** [0,4,9,16,36]
- B.** [1,4,9,16,25,36]
- C.** [4,16,36]
- D.** [1,9,25]



# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[1:6:2]?
- A.** [0,4,9,16,36]
- B.** [1,4,9,16,25,36]
- C.** [4,16,36]
- D.** [1,9,25]



# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[::-1]?
- A.** [49,36,25,16,9,4,1,0]
- B.** [0,1,4,9,16,25,36]
- C.** [1,4,9,16,25,36]
- D.** [36,25,16,9,4,1,0]



# GROSSES SLICING QUIZ

- Es gibt die Liste
  - zahlen = [0,1,4,9,16,25,36,49]
  - Was ist zahlen[::-1]?
- A. [49,36,25,16,9,4,1,0]
- B. [0,1,4,9,16,25,36]
- C. [1,4,9,16,25,36]
- D. [36,25,16,9,4,1,0]



# MUSTERLÖSUNG: 3-1

- Kreieren Sie ein Verzeichnis Übungsblatt3 in Ihrem Homeverzeichnis

```
Leonie@Laptop:~ $ mkdir Übungsblatt3
```



# MUSTERLÖSUNG: 3-2

- Sichern Sie im Verzeichnis Übungsblatt3 den Inhalt des Artikels im Wochenmagazin "der Freitag" über google in der Datei artikel.txt; siehe Artikel: <https://www.freitag.de/autoren/der-freitag/google-ganz-verstehen>

```
Leonie@Laptop:~ $ cd Übungsblatt3  
Leonie@Laptop:Übungsblatt3 $ kate artikel.txt
```



# MUSTERLÖSUNG: 3-3

- Lesen Sie die man pages der Befehle head, tail und wc

```
Leonie@Laptop:~ $ man head  
Leonie@Laptop:~ $ man tail  
Leonie@Laptop:~ $ man wc
```



# MUSTERLÖSUNG: 3-4

- Geben Sie die ersten drei Zeilen der Datei artikel.txt auf der Konsole aus.

```
Leonie@Laptop:~ $ head -3 artikel.txt
Google ganz verstehen
Plädoyer Wer im Internet surft, weiß noch lange nicht, was das für unsere
Gesellschaft bedeutet. Ein Schulfach muss her
Der bildungspolitische Skandal liegt nicht allein darin, dass es in unserer
„Mediengesellschaft“ kein Schulfach Medienbildung gibt. Er beginnt mit einer
verfehlten Idee von Medienbildung. Was an deutschen Schulen und
Weiterbildungseinrichtungen vereinzelt und fakultativ unter dem Label Medienpass
oder Medienführerschein vermittelt wird, zielt vor allem auf den fachkundigen,
effektiven Umgang mit den Medien.
```





# MUSTERLÖSUNG: 3-5

- Geben Sie die letzten drei Zeilen der Datei artikel.txt auf der Konsole aus.

```
Leonie@Laptop:~ $ tail -3 artikel.txt
```

```
Fehlende Debatte
```

```
Gewiss, es gibt die Enquete-Kommission des Deutschen Bundestags Internet und digitale Gesellschaft, und es gibt zunehmend entsprechende Berichte und Rubriken in den traditionellen Massenmedien. Es ist nicht so, dass man nicht Bescheid wissen könnte. Dennoch: Es fehlt eine öffentliche Debatte darüber, welche Art von Gesellschaft wir unseren Nachkommen hinterlassen wollen und wie viel Gestaltungsspielraum wir dabei der jüngeren Generation überlassen
```



# MUSTERLÖSUNG: 3-6

- Geben Sie die Anzahl der Zeilen und Wörter der Datei artikel.txt auf der Konsole aus.

```
Leonie@Laptop:~ $ wc -l -w artikel.txt
36 1091 artikel.txt
Leonie@Laptop:~ $ wc artikel.txt
36 1091 8553 artikel.txt
```



# MUSTERLÖSUNG: 3-7

- Kopieren Sie die Datei artikel.txt in text\_2.txt!

```
Leonie@Laptop:~ $ cp artikel.txt text_2.txt
```



# MUSTERLÖSUNG: 3-8

- Kopieren Sie die ersten 3 Zeilen der Datei text\_2.txt in text\_anfang.txt (Tipp: Redirecting Operator >)

```
Leonie@Laptop:~ $ head -3 text_2.txt > text_anfang.txt
```



# MUSTERLÖSUNG: 3-9

- Kopieren Sie die letzten 3 Zeilen der Datei text\_2.txt in text\_ende.txt (Tipp: Redirecting Operator >)

```
Leonie@Laptop:~ $ tail -3 text_2.txt > text_ende.txt
```



# MUSTERLÖSUNG: 3-10

- Schreiben Sie ein Programm, das drei Zahlen in den Variablen mit den Namen zahl1, zahl2 und zahl3 speichert, den Mittelwert der drei Zahlen berechnet und den Mittelwert in der Variable mit dem Namen mittel speichert. Geben Sie den Mittelwert auf dem Terminal aus.



# MUSTERLÖSUNG: 3-10

```
#!/usr/bin/python3
#Aufgabe 3-10
#Autorin: Leonie Weißweiler
zahl1 = int(input('Geben Sie die erste Zahl ein\n'))
zahl2 = int(input('Geben Sie die zweite Zahl ein\n'))
zahl3 = int(input('Geben Sie die dritte Zahl ein\n'))
mittel = (zahl1+zahl2+zahl3)/3
print('Der Mittelwert ist:', mittel)
```



# MUSTERLÖSUNG: 3-11

- Schreiben Sie ein Programm, das die Fakultät der Zahl 4 berechnet, speichert und ausgibt.





# MUSTERLÖSUNG: 3-11

```
#!/usr/bin/python3
#Aufgabe 3-11
# Autorin: Leonie Weißweiler
vier_fakultät = 4*3*2*1
print ('Vier Fakultät ist:', vier_fakultät)
```



# MUSTERLÖSUNG: 3-12

- Schreiben Sie ein Programm, das zwei Zahlen in den Variablen mit den Namen zahl1 und zahl2 speichert und den Wert der Variablen zahl1 und zahl2 vertauscht. Nach dem Vertauschen sollen der Wert der Variablen auf dem Terminal ausgegeben werden.



# MUSTERLÖSUNG: 3-12

```
#!/usr/bin/python3
#Aufgabe 3-12
#Autorin: Leonie Weißweiler
zahl1 = input('Geben Sie eine Zahl ein\n')
zahl2 = input('Geben Sie eine zweite Zahl ein\n')
temp = zahl1
zahl1 = zahl2
zahl2 = temp
print ('Vertauschte Zahlen: Zahl 1: ', zahl1, ' Zahl2: ', zahl2)
```



# MUSTERLÖSUNG: 3-13

- Schreiben Sie ein Programm, das drei Zahlen einliest, sie in einer Liste speichert, den Mittelwert berechnet und diesen ausgibt.



# MUSTERLÖSUNG: 3-13

```
#!/usr/bin/python3
#Aufgabe 3-13
#Autorin: Leonie Weißweiler

zahlen = []
zahlen.append(int(input('Geben Sie die erste Zahl ein\n')))
zahlen.append(int(input('Geben Sie die zweite Zahl ein\n')))
zahlen.append(int(input('Geben Sie die dritte Zahl ein\n')))
mittel = (zahlen[0]+zahlen[1]+zahlen[2])/3
print('Der Mittelwert der drei Zahlen ist', mittel)
```



# MUSTERLÖSUNG: 3-14

- Lesen Sie eine Textzeile von der Konsole ein und geben Sie die Textzeile 5 mal auf der Konsole aus:
- a) Die Zeilen sollen ohne Zeilenumbruch hintereinander ausgegeben werden
- b) Die Zeilen sollen mit Zeilenumbruch auf der Konsole ausgegeben werden



# MUSTERLÖSUNG: 3-14

```
#!/usr/bin/python3
#Aufgabe 3-14-a
#Autorin: Leonie Weißweiler
zeile = input('Geben Sie eine Textzeile ein\n')
i = 0
while (i<5):
    print(zeile, end='')
    i = i+1
print('\n')
```



# MUSTERLÖSUNG: 3-14

```
#!/usr/bin/python3
#Aufgabe 3-14-a
#Autorin: Leonie Weißweiler
zeile = input('Geben Sie eine Textzeile ein\n')
i = 0
while (i<5):
    print(zeile)
    i = i+1
```





# MUSTERLÖSUNG: 3-15

- Schreiben Sie folgendes Python Programm ab und starten Sie es. Was druckt das Programm aus?



# MUSTERLÖSUNG: 3-15

```
#!/usr/bin/python
#Aufgabe 3-15
#Autorin: Leonie Weißweiler
i = 5
while(i>=0):
    j = i*i
    print(i)
    print(j)
    i = i-1
```

