

# VIERTE ÜBUNG

ZUR EINFÜHRUNG IN DIE PROGRAMMIERUNG FÜR COMPUTERLINGUISTEN

# TWEEDBACK

- [Imu.twbk.de](http://Imu.twbk.de)
- Lesson ID: ???



Audience

Speaker

## Participate in a lecture

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



**PARTICIPATE**

# WIEDERHOLUNG: PIPING

- Was ist der richtige Operator?
- `echo "Ni! Ni! Ni!" ? knight.txt`
  - a. |
  - b. >
  - c. <



# WIEDERHOLUNG: PIPING

- Was ist der richtige Operator?
- `echo "Ni! Ni! Ni!" ? knight.txt`
  - a. `|`
  - b. `>`
  - c. `<`



# WIEDERHOLUNG: WHILE SCHLEIFEN

■ Was gibt das Programm aus?

■ `x = -3`

```
while (x < 4)
```

```
    x += 2
```

```
    print(x, end=' ')
```

a. -3 -1 1 3

b. -1 1 3 5

c. -3 -2 -1 0 1 2 3

d. -2 -1 0 1 2 3 4



# WIEDERHOLUNG: WHILE SCHLEIFEN

■ Was gibt das Programm aus?

■ `x = -3`

```
while (x < 4)
```

```
    x += 2
```

```
    print(x, end=' ')
```

a. -3 -1 1 3

b. -1 1 3 5

c. -3 -2 -1 0 1 2 3

d. -2 -1 0 1 2 3 4



# WIEDERHOLUNG: SLICING

- Was ist das Ergebnis des Ausdrucks?
- `[0,2,4,8,10,12,14,16][0:5]`
  - a. `[0,12]`
  - b. `[12,14,16]`
  - c. `[0,2,4,8,10]`
  - d. `[0,2,4,8,10,12]`



# WIEDERHOLUNG: SLICING

- Was ist das Ergebnis des Ausdrucks?
- `[0,2,4,8,10,12,14,16][0:5]`
  - a. `[0,12]`
  - b. `[12,14,16]`
  - c. `[0,2,4,8,10]`
  - d. `[0,2,4,8,10,12]`



# FOR SCHLEIFEN

- Mit For Schleifen kann man die Elemente einer Liste durchgehen (iterieren)
- Man vermeidet komplizierte, unlesbare while schleifen

```
zahlen = [5,6,7,8]
```

```
position = 0
```

```
while (position < len(zahlen)):
```

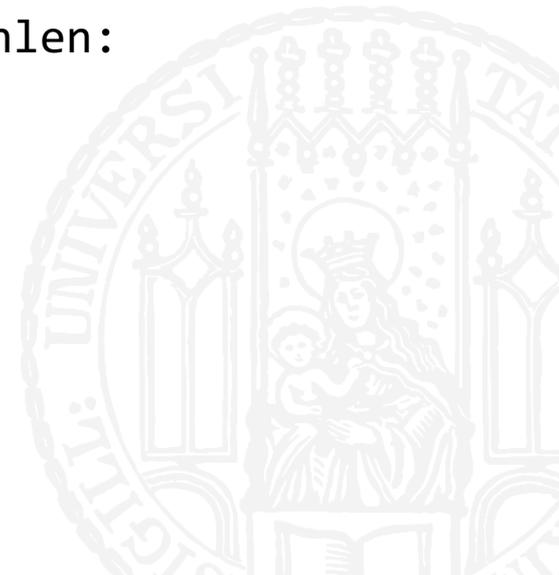
```
    z = zahlen[position]
```

```
    print(z)
```

```
zahlen = [5,6,7,8]
```

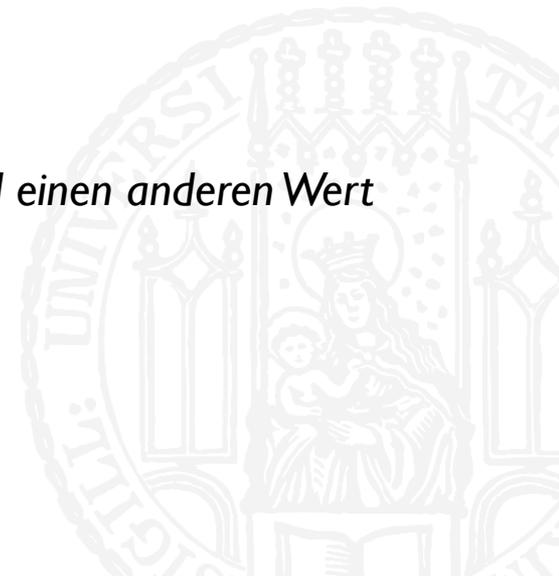
```
for z in zahlen:
```

```
    print(z)
```



# FOR SCHLEIFEN

- **for z in zahlen:**  
    **print(z)**
- **for** leitet die Schleife ein
- **z** ist die Variable die nacheinander alle Werte annimmt
- **in** kündigt die Liste an
- **zahlen** ist die Liste aus der die Werte kommen
- **:** kündigt den codeblock an
- *„Führe den folgenden Code immer wieder aus und lass z jedes mal einen anderen Wert aus der Liste sein“*



# FOR SCHLEIFEN

- `zweierpotenzen = [1,2,4,8,16]`  
`for a in zweierpotenzen:`  
    `print(a)`
- 1
- 2
- 4
- 8
- 16



# QUIZ

- Was gibt das Programm aus?
- ```
zahlen= [2,3,5,7,11]  
for p in zahlen:  
    print(p, end=' ')
```
- a. 2 3 5 7 11
- b. 2 2 2 2 2
- c. 11 11 11 11 11
- d. 11 7 5 3 2



# QUIZ

- Was gibt das Programm aus?
- ```
zahlen= [2,3,5,7,11]  
for p in zahlen:  
    print(p, end=' ')
```
- a. 2 3 5 7 11
- b. 2 2 2 2 2
- c. 11 11 11 11 11
- d. 11 7 5 3 2



# RANGES

- Mit `range(x)` kann man sich automatisch eine Range von Zahlen generieren lassen
- Ranges verhalten sich wie Listen, sind aber keine Listen (!)
- `range(5)`  
`[0,1,2,3,4]`
- `range(10,15)[4]`  
`14`
- ```
for x in range(5)
    print(x, end=' ')
0 1 2 3 4
```



# RANGES

- Die Parameter verhalten sich wie die Parameter beim Slicing
- `range(3) ~ [0, 1, 2]`
- `range(4, 9) ~ [4, 5, 6, 7, 8]`
- `range(10, 0) ~ []`
- `range(0, 10, 2) ~ [0, 2, 4, 6, 8]`



# ACHTUNG SELTSAM!

- Um eine Range “rückwärts“ zu definieren (oder einen Teil einer Liste rückwärts zu slicen) muss im dritten Argument ein Minus stehen **und** die ersten beiden Argumente vertauscht sein!
- `range(0,10,-1)`
- `range(10,0)`
- `range(10,0,-1)` `[10,9,8,7,6,5,4,3,2,1]`



# QUIZ

■ Was gibt das Programm aus?

■ `for x in range(10):`  
    `print(x, end=' ')`

a. 10

b. 10 0

c. 0 1 2 3 4 5 6 7 8 9 10

d. 0 1 2 3 4 5 6 7 8 9



# QUIZ

■ Was gibt das Programm aus?

■ `for x in range(0,10):`  
    `print(x, end=' ')`

a. 10

b. 10 0

c. 0 1 2 3 4 5 6 7 8 9 10

d. 0 1 2 3 4 5 6 7 8 9



# QUIZ

- Was gibt das Programm aus?
- `for x in range(0,10,2):`  
    `print(x, end=' ')`
- a. 0 10 2
- b. 0 2 4 8
- c. 0 2 4 8 10
- d. (nichts)



# QUIZ

- Was gibt das Programm aus?
- `for x in range(0,10,2):`  
    `print(x, end=' ')`
- a. 0 10 2
- b. 0 2 4 8
- c. 0 2 4 8 10
- d. (nichts)



# QUIZ

- Was gibt das Programm aus?
- `for x in range(0,10,-2):`  
    `print(x, end=' ')`
- a. 0 2 4 8
- b. 8 4 2 0
- c. 0 2 4 8 10
- d. (nichts)



# QUIZ

- Was gibt das Programm aus?
- `for x in range(0,10,-2):`  
    `print(x, end=' ')`
- a. 0 2 4 8
- b. 8 4 2 0
- c. 0 2 4 8 10
- d. (nichts)



# QUIZ

- Was gibt das Programm aus?
- `for x in range(7,2,-1):`  
    `print(x, end=' ')`
- a. 7 6 5 4 3
- b. 2 3 4 5 6
- c. 3 4 5 6 7
- d. (nichts)



# QUIZ

- Was gibt das Programm aus?
- `for x in range(7,2,-1):`  
    `print(x, end=' ')`
- a. 7 6 5 4 3
- b. 2 3 4 5 6
- c. 3 4 5 6 7
- d. (nichts)



# IMMUTABLES VS MUTABLES

- Es gibt zwei Arten von Typen in Python: Mutables und Immutables
- Nur Variablen deren Typ mutable ist, können durch Methoden verändert werden
- Variablen deren Typ immutable ist, können nur neu belegt werden
- Zahlen sind immutable:
  - `i = 3`
  - `i = i + 4`
- Listen sind mutable:
  - `l = [0,7,2,5,1]`
  - `l.sort()`



# IMMUTABLES VS MUTABLES

## Immutable

- int
- float
- string
- boolean
- range

## Mutable

- list



# MUSTERLÖSUNG 4-1

- Schrieben sei ein Programm, das zwei Vornamen von der Tastatur einliest und in den Variablen mit den Namen `name1` und `name2` speichert. Die Eingaben sollen folgendermaßen ausgegeben werden, falls z.B. Anna und Sepp eingegeben wurden:
- `Der erste Vorname ist Anna und der zweite Vorname ist Sepp.`
- `Auf Wiedersehen.`



# MUSTERLÖSUNG 4-1

```
#!/usr/bin/python3
#Aufgabe 4-1
#Autorin: Leonie Weißweiler
name1 = input('Geben Sie den ersten Vornamen ein\n')
name2 = input('Geben Sie den zweiten Vornamen ein\n')
print('Der erste Vorname ist ', name1,
      ' und der zweite Vorname ist ', name2 ,
      '.\nAuf Wiedersehen.', sep='')
```



## MUSTERLÖSUNG 4-2

- Schreiben Sie ein Programm, das vier Namen einliest und in einer Liste speichert. Dann soll das Programm den alphabetisch größten und kleinsten Namen ausgeben.



# MUSTERLÖSUNG 4-2

```
#!/usr/bin/python3
#Aufgabe 4-2
#Autorin: Leonie Weißweiler
namen = []
namen.append(input('Geben Sie den ersten Namen ein\n'))
namen.append(input('Geben Sie den zweiten Namen ein\n'))
namen.append(input('Geben Sie den dritten Namen ein\n'))
namen.append(input('Geben Sie den vierten Namen ein\n'))
namen.sort()
print('Alphabetisch größter Name:', namen[-1])
print('Alphabetisch kleinster Name:',namen[0])
```



## MUSTERLÖSUNG 4-3

- Schreiben Sie ein Programm, das den Benutzer nur ein einziges Mal auffordert, eine im Programm intern definierte Zahl zu erraten. Falls die eingegebene Zahl nicht mit der internen Zahl identisch ist, soll eine Fehlermeldung ausgegeben werden aus der hervorgeht, ob die eingegebene Zahl größer oder kleiner als die intern gespeicherte Zahl ist. Sind sie identisch, dann soll eine Erfolgsmeldung ausgegeben werden.



# MUSTERLÖSUNG 4-3

```
#!/usr/bin/python3
#Aufgabe 4-3
#Autorin: Leonie Weißweiler
passwort = 42
guess = int(input('Geben Sie die Zahl ein\n'))

if(guess==passwort):
    print('Zahl korrekt.')
elif(guess<passwort):
    print('Eingegebene Zahl zu klein')
else:
    print('Eingegebene Zahl zu groß')
```



## MUSTERLÖSUNG 4-4

- Erweitern Sie das Programm aus Aufgabe 4-3 dahingehend, dass der Benutzer nur maximal fünfmal gefragt wird. Nach jedem Fehlversuch soll ausgegeben werden, wie viele Versuche der Benutzer noch hat.



# MUSTERLÖSUNG 4-4

```
#!/usr/bin/python3
#Aufgabe 4-4
#Autorin: Leonie Weißweiler
zahl = 42
versuche = 5
while(versuche>0):
    versuche = versuche-1
    guess = int(input('Geben Sie die Zahl ein\n'))
    if(guess==zahl):
        print('Zahl korrekt.')
        versuche = 0
    elif(guess<zahl):
        print('Eingegebene Zahl zu klein. Noch',versuche,'Versuche.')
    else:
        print('Eingegebene Zahl zu groß. Noch',versuche,'Versuche.')
```



## MUSTERLÖSUNG 4-5

- Erweitern Sie das Programm aus Aufgabe 4-4 dahingehend, dass Sie nicht nach einer intern definierten Zahl, sondern nach einem intern definierten Wort gefragt werden.



# MUSTERLÖSUNG 4-5

```
#!/usr/bin/python3
#Aufgabe 4-5
#Autorin: Leonie Weißweiler
passwort = 'spambaconandeggs'
versuche = 5
while(versuche>0):
    versuche = versuche-1
    guess = input('Geben Sie das Passwort ein\n')
    guess = guess.lower()
    if(guess==passwort):
        print('Passwort korrekt.')
        versuche = 0
    else:
        print('Eingegebenes Passwort falsch. Noch',versuche,'Versuche.')
```

