

# 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)
```



# 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)



# 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



# 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]



# 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]**



# 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]`



# 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



# 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)



# WIEDERHOLUNG: FOR SCHLEIFEN

7R8

- Was gibt das Programm aus?
- `noten = ['a', 'cis', 'f', 'g', 'd']`  
`for p in noten:`  
    `print(p, end='')`

- a. `acisfgd`
- b. `a cis f g d`
- c. `n o t e n`
- d. `p p p p p`



# WIEDERHOLUNG: FOR SCHLEIFEN

7R8

- Was gibt das Programm aus?
- `noten = ['a', 'cis', 'f', 'g', 'd']`  
`for p in noten:`  
 `print(p, end='')`

a. `acisfgd`

b. `a cis f g d`

c. `n o t e n`

d. `p p p p p`



# WIEDERHOLUNG: RANGES

7R8

■ Was gibt das Programm aus?

■ `for x in range(4):`  
    `print(x, end=' ')`

a. 0 1 2 3

b. 4

c. 3

d. 0 1 2 3 4



# WIEDERHOLUNG: RANGES

7R8

- Was gibt das Programm aus?

- `for x in range(4):`  
    `print(x, end=' ')`

a. 0 1 2 3

b. 4

c. 3

d. 0 1 2 3 4



# QUIZ

7R8

- Was gibt das Programm aus?
- `for x in range(10,40,5):`  
    `print(x, end=' ')`
- a. 15 20 25 30 35 40
- b. 10 15 20 25 30 35 40
- c. 10 15 20 25 30 35
- d. 40 35 30 25 20 15 10



# QUIZ

7R8

- Was gibt das Programm aus?
- `for x in range(10,40,5):`  
    `print(x, end=' ')`
- a. 15 20 25 30 35 40
- b. 10 15 20 25 30 35 40
- c. 10 15 20 25 30 35
- d. 40 35 30 25 20 15 10



# QUIZ

7R8

- Was gibt das Programm aus?
- `for x in range(0,10,-2):`  
    `print(x, end=' ')`
- a. 10 8 6 4 2 0
- b. 0 2 4 8 8
- c. 0 2 4 6 8 10
- d. (nichts)



# QUIZ

7R8

- Was gibt das Programm aus?
- `for x in range(0,10,-2):`  
    `print(x, end=' ')`
- a. 10 8 6 4 2 0
- b. 0 2 4 8 8
- c. 0 2 4 6 8 10
- d. (nichts)



# QUIZ

7R8

- Welches Statement funktioniert?
  - a. `open = datei('knights.txt', 'r')`
  - b. `open(datei, 'knights.txt', 'r')`
  - c. `open('knights.txt', 'r')`
  - d. `datei = open('knights.txt', 'r')`



# QUIZ

7R8

- Welches Statement funktioniert?
  - a. `open = datei('knights.txt', 'r')`
  - b. `open(datei, 'knights.txt', 'r')`
  - c. `open('knights.txt', 'r')`
  - d. `datei = open('knights.txt', 'r')`



# QUIZ

7R8

- Welchen Datentypen haben die Variablen `datei` und `inhalt`?
- ```
datei = open('knights.txt', 'r')  
inhalt = datei.read()  
datei.close()
```
- a. `datei`: `filehandle`      `inhalt`: `filehandle`
- b. `datei`: `string`            `inhalt`: `string`
- c. `datei`: `filehandle`        `inhalt`: `string`
- d. `datei`: `string`             `inhalt`: `filehandle`



# QUIZ

7R8

- Welchen Datentypen haben die Variablen `datei` und `inhalt`?
- ```
datei = open('knights.txt', 'r')  
inhalt = datei.read()  
datei.close()
```
- a. `datei`: `filehandle`      `inhalt`: `filehandle`
- b. `datei`: `string`              `inhalt`: `string`
- c. `datei`: `filehandle`        `inhalt`: `string`
- d. `datei`: `string`              `inhalt`: `filehandle`



# QUIZ

7R8

- Die Datei `african-swallow.txt` enthält zu Beginn den Text "Cocounts". Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('african-swallow.txt', 'a')  
datei.write(' and moose')  
datei.close()
```
- a. "Cocounuts"
- b. "Coconuts and moose"
- c. " and moose"
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `african-swallow.txt` enthält zu Beginn den Text "Cocounts". Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('african-swallow.txt', 'a')  
datei.write(' and moose')  
datei.close()
```
- a. "Cocounuts"
- b. "Coconuts and moose"
- c. " and moose"
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `snacks.txt` enthält zu Beginn den Text "Lerchenzungen". Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('snacks.txt', 'w')  
datei.write('Otternasen')  
datei.write('Ozelothmilch')  
datei.close()
```
- a. "Lerchenzungen Otternasen Ozelothmilch"
- b. "OtternasenOzelothmilch"
- c. "Otternasen  
Ozelothmilch"
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `snacks.txt` enthält zu Beginn den Text "Lerchenzungen". Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('snacks.txt', 'w')  
datei.write('Otternasen')  
datei.write('Ozelothmilch')  
datei.close()
```
- a. "Lerchenzungen Otternasen Ozelothmilch"
- b. "OtternasenOzelothmilch"
- c. "Otternasen  
Ozelothmilch"
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `graffito.txt` enthält zu Beginn den Text “Romanes eunt domus”. Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('graffito.txt', 'w')
datei.write(500)
datei.close()
```
- a. “Romanes eunt domus500“
- b. “500“
- c. “Romanes eunt domus  
500“
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `graffito.txt` enthält zu Beginn den Text “Romanes eunt domus”. Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('graffito.txt', 'w')  
datei.write(500) ←  
datei.close()
```
- a. “Romanes eunt domus500“
- b. “500“
- c. “Romanes eunt domus  
500“
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `graffito.txt` enthält zu Beginn den Text “Romanes eunt domus”. Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('graffito.txt', 'r')  
datei.write('500')  
datei.close()
```
- a. “Romanes eunt domus500“
- b. “500“
- c. “Romanes eunt domus  
500“
- d. Programm funktioniert nicht



# QUIZ

7R8

- Die Datei `graffito.txt` enthält zu Beginn den Text “Romanes eunt domus”. Welchen Text enthält sie nach der Ausführung des Programms?
- ```
datei = open('graffito.txt', 'r')
datei.write('500')
datei.close()
```


- a. “Romanes eunt domus500“
- b. “500“
- c. “Romanes eunt domus  
500“
- d. Programm funktioniert nicht



# QUIZ

- Was ist **00110** im Dezimalsystem?
- a) 2
- b) 5
- c) 6
- d) 11



# QUIZ

- Was ist **00110** im Dezimalsystem?
- a) 2
- b) 5
- c) 6
- d) 11



# QUIZ

- Was ist **10010** im Dezimalsystem?
  - 36
  - 18
  - 10
  - 9



# QUIZ

- Was ist **10010** im Dezimalsystem?
- a) 36
- b) 18
- c) 10
- d) 9



# QUIZ

■ Was ist 10 im Binärformat?

- a) 0010
- b) 0110
- c) 1010
- d) 0101



# QUIZ

■ Was ist 10 im Binärformat?

- a) 0010
- b) 0110
- c) 1010
- d) 0101



# QUIZ

■ Was ist 29 im Binärformat?

- a) 10110
- b) 10111
- c) 11101
- d) 11011



# QUIZ

■ Was ist 29 im Binärformat?

- a) 10110
- b) 10111
- c) 11101
- d) 11011



# QUIZ

■ Was ist Oktal 29 im Binärsystem?

a) 010 110

b) 110 111

c) 011 101

d) 111 011



# QUIZ

- Was ist Oktal 29 im Binärformat?



# QUIZ

■ Was ist Oktal 22 im Binärsystem?

- a) 010 110
- b) 110 011
- c) 011 011
- d) 010 010



# QUIZ

■ Was ist Oktal 22 im Binärsystem?

- a) 010 110
- b) 110 011
- c) 011 011
- d) 010 010



# QUIZ

- Was ist Binär 101 011 010 im Oktalsystem?
  - a) 221
  - b) 532
  - c) 364
  - d) 521



# QUIZ

- Was ist Binär 101 011 010 im Oktalsystem?
  - a) 221
  - b) 532
  - c) 364
  - d) 521



# QUIZ

- Was ist Binär 101011010 im Hexadezimalsystem?
- a) 15A
- b) AD0
- c) 532
- d) 111AA



# QUIZ

- Was ist Binär 101011010 im Hexadezimalsystem?
  - a) 15A
  - b) AD0
  - c) 532
  - d) 111AA



# QUIZ

- Was ist Hexadezimal DEADAFFE in Binär?
  - a) 1101 1110 1010 1101 1010 1111 1111 1110
  - b) 1110 1101 1100 1110 1100 1111 1111 1101
  - c) 0011 1101 1010 0011 1010 1111 1111 1101
  - d) 0011 1110 1010 0011 1010 1111 1111 1110



# QUIZ

- Was ist Hexadezimal DEADAFFE in Binär?
  - 1101 1110 1010 1101 1010 1111 1111 1110
  - 1110 1101 1100 1110 1100 1111 1111 1101
  - 0011 1101 1010 0011 1010 1111 1111 1101
  - 0011 1110 1010 0011 1010 1111 1111 1110



# QUIZ

- Was ist `"Tux Müde. Tux Schlafen".split("u")` ?
- a) `["Tu", "x Müde. Tu", "x Schlafen"]`
- b) `["T", "ux Müde. T", "ux Schlafen"]`
- c) `["T", "ux M", "de. T", "x Schlafen"]`
- d) `["T", "x Müde. T", "x Schlafen"]`



# QUIZ

- Was ist `"Tux Müde. Tux Schlafen".split("u")` ?
- a) `["Tu", "x Müde. Tu", "x Schlafen"]`
- b) `["T", "ux Müde. T", "ux Schlafen"]`
- c) `["T", "ux M", "de. T", "x Schlafen"]`
- d) `["T", "x Müde. T", "x Schlafen"]`



# WIEDERHOLUNG: BINÄR

71G

- Was ist **101010** im Dezimalsystem?
- a) 10
- b) 26
- c) 42
- d) 84



# WIEDERHOLUNG: BINÄR

71G

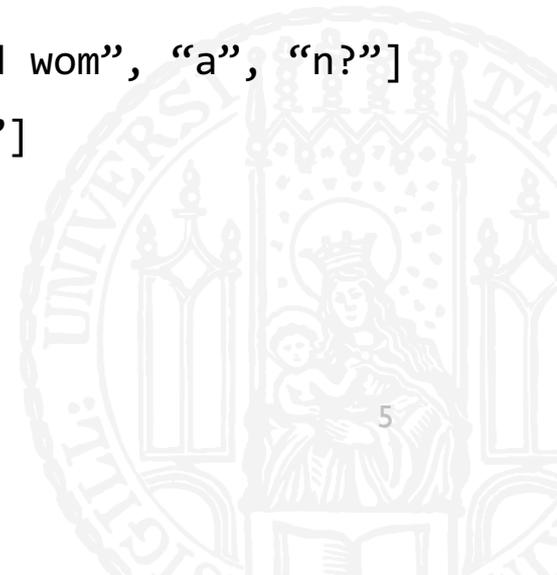
- Was ist **101010** im Dezimalsystem?
- a) 10
- b) 26
- c) 42
- d) 84



# WIEDERHOLUNG: SPLIT

71G

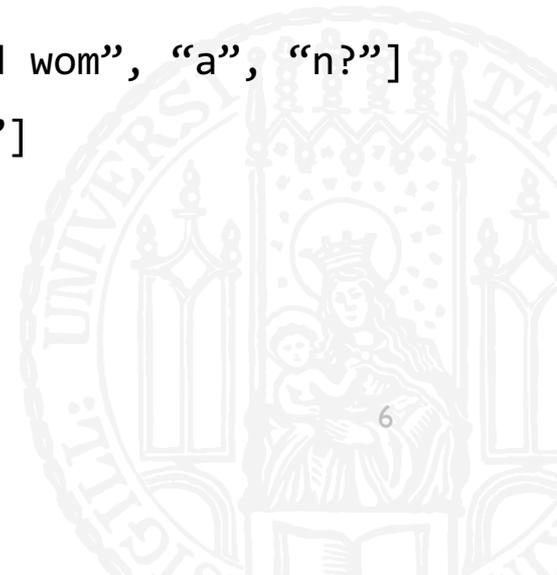
- Was ist `"Are you saying ni to that old woman?".split("a")` ?
- a) [`"re you s"`, `"ying ni to th"`, `"t old wom"`, `"n?"`]
- b) [`"Are you s"`, `"ying ni to th"`, `"t old wom"`, `"n?"`]
- c) [`"A"`, `"re you s"`, `"a"`, `"ying ni to th"`, `"a"`, `"t old wom"`, `"a"`, `"n?"`]
- d) [`"Are you s"`, `"aying ni to th"`, `"at old wom"`, `"an?"`]



# WIEDERHOLUNG: SPLIT

71G

- Was ist `"Are you saying ni to that old woman?".split("a")` ?
  - a) [`"re you s"`, `"ying ni to th"`, `"t old wom"`, `"n?"`]
  - b) [`"Are you s"`, `"ying ni to th"`, `"t old wom"`, `"n?"`]
  - c) [`"A"`, `"re you s"`, `"a"`, `"ying ni to th"`, `"a"`, `"t old wom"`, `"a"`, `"n?"`]
  - d) [`"Are you s"`, `"aying ni to th"`, `"at old wom"`, `"an?"`]



# QUIZ

71G

■ Was ist **B** in ASCII?

- a) 100010
- b) 10010
- c) 1000010
- d) 10000010



# QUIZ

71G

■ Was ist **B** in ASCII?

- a) 100010
- b) 10010
- c) 1000010
- d) 10000010



# QUIZ

71G

- Wieviele Iso-Latin Buchstaben sind hier?

01000101 11111100 01000111 01100011 11110110

- a) 3
- b) 4
- c) 2
- d) 1



# QUIZ

71G

- Wieviele Iso-Latin Buchstaben sind hier?

01000101 11111100 01000111 01100011 11110110

- a) 3
- b) 4
- c) 2
- d) 1



# QUIZ

71G

- Was ist hier passiert?

**Das ist nämlich doof!**

- a) ISO-Latin als ASCII angezeigt
- b) ASCII als ISO-Latin angezeigt
- c) UTF-8 als ISO-Latin angezeigt
- d) UTF-8 als ASCII angezeigt



# QUIZ

71G

- Was ist hier passiert?

**Das ist nämlich doof!**

- a) ISO-Latin als ASCII angezeigt
- b) ASCII als ISO-Latin angezeigt
- c) **UTF-8 als ISO-Latin angezeigt**
- d) UTF-8 als ASCII angezeigt



# QUIZ

71G

- Aus „abcdefg“ ist „愀戀振摠攀昀□“ geworden. Was ist passiert?
  - a) UTF-8 als UTF-16LE angezeigt
  - b) UTF-16LE als UTF-9 angezeigt
  - c) UTF-16LE als UTF-16BE angezeigt
  - d) UTF-16 als ISO-Latin angezeigt



# QUIZ

71G

- Aus „abcdefg“ ist „愀戀振摠攀昀□“ geworden. Was ist passiert?
  - a) UTF-8 als UTF-16LE angezeigt
  - b) UTF-16LE als UTF-9 angezeigt
  - c) UTF-16LE als UTF-16BE angezeigt
  - d) UTF-16 als ISO-Latin angezeigt



# QUIZ

71G

```
Leonie@Laptop $ hexdump -C datei.txt
00000000  4c 00 61 00 74 00 65 00  69 00 6e 00 20 00 69 00  |L.a.t.e.i.n. .i.|
00000010  73 00 74 00 20 00 65 00  69 00 6e 00 65 00 20 00  |s.t. .e.i.n.e. .|
00000020  74 00 6f 00 74 00 65 00  20 00 53 00 70 00 72 00  |t.o.t.e. .S.p.r.|
00000030  61 00 63 00 68 00 65 00  0a 00                                |a.c.h.e...|
0000003a
```

■ Welches Encoding hat die Datei?

- a) ISO-Latin
- b) UTF-8
- c) UTF-16LE
- d) UTF-16BE



# QUIZ

71G

```
Leonie@Laptop $ hexdump -C datei.txt
00000000  4c 00 61 00 74 00 65 00  69 00 6e 00 20 00 69 00  |L.a.t.e.i.n. .i.|
00000010  73 00 74 00 20 00 65 00  69 00 6e 00 65 00 20 00  |s.t. .e.i.n.e. .|
00000020  74 00 6f 00 74 00 65 00  20 00 53 00 70 00 72 00  |t.o.t.e. .S.p.r.|
00000030  61 00 63 00 68 00 65 00  0a 00                                |a.c.h.e...|
0000003a
```

■ Welches Encoding hat die Datei?

- a) ISO-Latin
- b) UTF-8
- c) UTF-16LE
- d) UTF-16BE



# QUIZ

71G

- Was ist das Ergebnis dieses Codes?
- ```
dict = {}  
dict['a'] = 2  
dict['b'] = 3  
dict['a'] = 4  
print( dict['a'] + dict['b'] )
```

- a) 4
- b) 5
- c) 6
- d) 7



# QUIZ

71G

- Was ist das Ergebnis dieses Codes?
- ```
dict = {}  
dict['a'] = 2  
dict['b'] = 3  
dict['a'] = 4  
print( dict['a'] + dict['b'] )
```

- a) 4
- b) 5
- c) 6
- d) 7



# QUIZ

71G

■ Wie gibt man die Werte eines Dictionaries sortiert nach den Keys aus?

- a) `for key in sorted(dict)`  
`print(key)`
- b) `for key in sorted(dict.keys())`  
`print(key)`
- c) `for key in sorted(dict.keys())`  
`print(dict[key])`
- d) `for key, value in sorted(dict.items())`  
`print(value)`



# QUIZ

71G

■ Wie gibt man die **Werte** eines Dictionaries **sortiert nach den Keys** aus?

- a) `for key in sorted(dict)  
print(key)`
- b) `for key in sorted(dict.keys())  
print(key)`
- c) `for key in sorted(dict.keys())  
print(dict[key])`
- d) `for key, value in sorted(dict.keys())  
print(value)`



# WIEDERHOLUNG : ENCODINGS

GZZ

- Was ist hier passiert?

## Anti terror Änderung

- a) ISO-Latin als ASCII angezeigt
- b) ASCII als ISO-Latin angezeigt
- c) UTF-8 als ISO-Latin angezeigt
- d) UTF-8 als ASCII angezeigt



# WIEDERHOLUNG : ENCODINGS

GZZ

- Was ist hier passiert?

## Anti terror Änderung

- a) ISO-Latin als ASCII angezeigt
- b) ASCII als ISO-Latin angezeigt
- c) UTF-8 als ISO-Latin angezeigt
- d) UTF-8 als ASCII angezeigt



# QUIZ

GZZ

- Aus  
„Das ist nicht mein Traumjob“ ist  
„𐄂𐄃𐄄𐄅𐄆𐄇𐄈𐄉𐄊𐄋𐄌𐄍𐄎𐄏𐄐𐄑𐄒𐄓𐄔𐄕𐄖𐄗𐄘𐄙𐄚𐄛𐄜𐄝𐄞𐄟𐄠𐄡𐄢𐄣𐄤𐄥𐄦𐄧𐄨𐄩𐄪𐄫𐄬𐄭𐄮𐄯𐄰𐄱𐄲𐄳𐄴𐄵𐄶𐄷𐄸𐄹𐄺𐄻𐄼𐄽𐄾𐄿𐅀𐅁𐅂𐅃𐅄𐅅𐅆𐅇𐅈𐅉𐅊𐅋𐅌𐅍𐅎𐅏𐅐𐅑𐅒𐅓𐅔𐅕𐅖𐅗𐅘𐅙𐅚𐅛𐅜𐅝𐅞𐅟𐅠𐅡𐅢𐅣𐅤𐅥𐅦𐅧𐅨𐅩𐅪𐅫𐅬𐅭𐅮𐅯𐅰𐅱𐅲𐅳𐅴𐅵𐅶𐅷𐅸𐅹𐅺𐅻𐅼𐅽𐅾𐅿𐆀𐆁𐆂𐆃𐆄𐆅𐆆𐆇𐆈𐆉𐆊𐆋𐆌𐆍𐆎𐆏𐆐𐆑𐆒𐆓𐆔𐆕𐆖𐆗𐆘𐆙𐆚𐆛𐆜𐆝𐆞𐆟𐆠𐆡𐆢𐆣𐆤𐆥𐆦𐆧𐆨𐆩𐆪𐆫𐆬𐆭𐆮𐆯𐆰𐆱𐆲𐆳𐆴𐆵𐆶𐆷𐆸𐆹𐆺𐆻𐆼𐆽𐆾𐆿𐇀𐇁𐇂𐇃𐇄𐇅𐇆𐇇𐇈𐇉𐇊𐇋𐇌𐇍𐇎𐇏𐇐𐇑𐇒𐇓𐇔𐇕𐇖𐇗𐇘𐇙𐇚𐇛𐇜𐇝𐇞𐇟𐇠𐇡𐇢𐇣𐇤𐇥𐇦𐇧𐇨𐇩𐇪𐇫𐇬𐇭𐇮𐇯𐇰𐇱𐇲𐇳𐇴𐇵𐇶𐇷𐇸𐇹𐇺𐇻𐇼𐇽𐇾𐇿𐈀𐈁𐈂𐈃𐈄𐈅𐈆𐈇𐈈𐈉𐈊𐈋𐈌𐈍𐈎𐈏𐈐𐈑𐈒𐈓𐈔𐈕𐈖𐈗𐈘𐈙𐈚𐈛𐈜𐈝𐈞𐈟𐈠𐈡𐈢𐈣𐈤𐈥𐈦𐈧𐈨𐈩𐈪𐈫𐈬𐈭𐈮𐈯𐈰𐈱𐈲𐈳𐈴𐈵𐈶𐈷𐈸𐈹𐈺𐈻𐈼𐈽𐈾𐈿𐉀𐉁𐉂𐉃𐉄𐉅𐉆𐉇𐉈𐉉𐉊𐉋𐉌𐉍𐉎𐉏𐉐𐉑𐉒𐉓𐉔𐉕𐉖𐉗𐉘𐉙𐉚𐉛𐉜𐉝𐉞𐉟𐉠𐉡𐉢𐉣𐉤𐉥𐉦𐉧𐉨𐉩𐉪𐉫𐉬𐉭𐉮𐉯𐉰𐉱𐉲𐉳𐉴𐉵𐉶𐉷𐉸𐉹𐉺𐉻𐉼𐉽𐉾𐉿𐊀𐊁𐊂𐊃𐊄𐊅𐊆𐊇𐊈𐊉𐊊𐊋𐊌𐊍𐊎𐊏𐊐𐊑𐊒𐊓𐊔𐊕𐊖𐊗𐊘𐊙𐊚𐊛𐊜𐊝𐊞𐊟𐊠𐊡𐊢𐊣𐊤𐊥𐊦𐊧𐊨𐊩𐊪𐊫𐊬𐊭𐊮𐊯𐊰𐊱𐊲𐊳𐊴𐊵𐊶𐊷𐊸𐊹𐊺𐊻𐊼𐊽𐊾𐊿𐋀𐋁𐋂𐋃𐋄𐋅𐋆𐋇𐋈𐋉𐋊𐋋𐋌𐋍𐋎𐋏𐋐𐋑𐋒𐋓𐋔𐋕𐋖𐋗𐋘𐋙𐋚𐋛𐋜𐋝𐋞𐋟𐋠𐋡𐋢𐋣𐋤𐋥𐋦𐋧𐋨𐋩𐋪𐋫𐋬𐋭𐋮𐋯𐋰𐋱𐋲𐋳𐋴𐋵𐋶𐋷𐋸𐋹𐋺𐋻𐋼𐋽𐋾𐋿𐌀𐌁𐌂𐌃𐌄𐌅𐌆𐌇𐌈𐌉𐌊𐌋𐌌𐌍𐌎𐌏𐌐𐌑𐌒𐌓𐌔𐌕𐌖𐌗𐌘𐌙𐌚𐌛𐌜𐌝𐌞𐌟𐌠𐌡𐌢𐌣𐌤𐌥𐌦𐌧𐌨𐌩𐌪𐌫𐌬𐌭𐌮𐌯𐌰𐌱𐌲𐌳𐌴𐌵𐌶𐌷𐌸𐌹𐌺𐌻𐌼𐌽𐌾𐌿𐍀𐍁𐍂𐍃𐍄𐍅𐍆𐍇𐍈𐍉𐍊𐍋𐍌𐍍𐍎𐍏𐍐𐍑𐍒𐍓𐍔𐍕𐍖𐍗𐍘𐍙𐍚𐍛𐍜𐍝𐍞𐍟𐍠𐍡𐍢𐍣𐍤𐍥𐍦𐍧𐍨𐍩𐍪𐍫𐍬𐍭𐍮𐍯𐍰𐍱𐍲𐍳𐍴𐍵𐍶𐍷𐍸𐍹𐍺𐍻𐍼𐍽𐍾𐍿𐎀𐎁𐎂𐎃𐎄𐎅𐎆𐎇𐎈𐎉𐎊𐎋𐎌𐎍𐎎𐎏𐎐𐎑𐎒𐎓𐎔𐎕𐎖𐎗𐎘𐎙𐎚𐎛𐎜𐎝𐎞𐎟𐎠𐎡𐎢𐎣𐎤𐎥𐎦𐎧𐎨𐎩𐎪𐎫𐎬𐎭𐎮𐎯𐎰𐎱𐎲𐎳𐎴𐎵𐎶𐎷𐎸𐎹𐎺𐎻𐎼𐎽𐎾𐎿𐏀𐏁𐏂𐏃𐏄𐏅𐏆𐏇𐏈𐏉𐏊𐏋𐏌𐏍𐏎𐏏𐏐𐏑𐏒𐏓𐏔𐏕𐏖𐏗𐏘𐏙𐏚𐏛𐏜𐏝𐏞𐏟𐏠𐏡𐏢𐏣𐏤𐏥𐏦𐏧𐏨𐏩𐏪𐏫𐏬𐏭𐏮𐏯𐏰𐏱𐏲𐏳𐏴𐏵𐏶𐏷𐏸𐏹𐏺𐏻𐏼𐏽𐏾𐏿𐐀𐐁𐐂𐐃𐐄𐐅𐐆𐐇𐐈𐐉𐐊𐐋𐐌𐐍𐐎𐐏𐐐𐐑𐐒𐐓𐐔𐐕𐐖𐐗𐐘𐐙𐐚𐐛𐐜𐐝𐐞𐐟𐐠𐐡𐐢𐐣𐐤𐐥𐐦𐐧𐐨𐐩𐐪𐐫𐐬𐐭𐐮𐐯𐐰𐐱𐐲𐐳𐐴𐐵𐐶𐐷𐐸𐐹𐐺𐐻𐐼𐐽𐐾𐐿𐑀𐑁𐑂𐑃𐑄𐑅𐑆𐑇𐑈𐑉𐑊𐑋𐑌𐑍𐑎𐑏𐑐𐑑𐑒𐑓𐑔𐑕𐑖𐑗𐑘𐑙𐑚𐑛𐑜𐑝𐑞𐑟𐑠𐑡𐑢𐑣𐑤𐑥𐑦𐑧𐑨𐑩𐑪𐑫𐑬𐑭𐑮𐑯𐑰𐑱𐑲𐑳𐑴𐑵𐑶𐑷𐑸𐑹𐑺𐑻𐑼𐑽𐑾𐑿𐒀𐒁𐒂𐒃𐒄𐒅𐒆𐒇𐒈𐒉𐒊𐒋𐒌𐒍𐒎𐒏𐒐𐒑𐒒𐒓𐒔𐒕𐒖𐒗𐒘𐒙𐒚𐒛𐒜𐒝𐒞𐒟𐒠𐒡𐒢𐒣𐒤𐒥𐒦𐒧𐒨𐒩𐒪𐒫𐒬𐒭𐒮𐒯𐒰𐒱𐒲𐒳𐒴𐒵𐒶𐒷𐒸𐒹𐒺𐒻𐒼𐒽𐒾𐒿𐓀𐓁𐓂𐓃𐓄𐓅𐓆𐓇𐓈𐓉𐓊𐓋𐓌𐓍𐓎𐓏𐓐𐓑𐓒𐓓𐓔𐓕𐓖𐓗𐓘𐓙𐓚𐓛𐓜𐓝𐓞𐓟𐓠𐓡𐓢𐓣𐓤𐓥𐓦𐓧𐓨𐓩𐓪𐓫𐓬𐓭𐓮𐓯𐓰𐓱𐓲𐓳𐓴𐓵𐓶𐓷𐓸𐓹𐓺𐓻𐓼𐓽𐓾𐓿𐔀𐔁𐔂𐔃𐔄𐔅𐔆𐔇𐔈𐔉𐔊𐔋𐔌𐔍𐔎𐔏𐔐𐔑𐔒𐔓𐔔𐔕𐔖𐔗𐔘𐔙𐔚𐔛𐔜𐔝𐔞𐔟𐔠𐔡𐔢𐔣𐔤𐔥𐔦𐔧𐔨𐔩𐔪𐔫𐔬𐔭𐔮𐔯𐔰𐔱𐔲𐔳𐔴𐔵𐔶𐔷𐔸𐔹𐔺𐔻𐔼𐔽𐔾𐔿𐕀𐕁𐕂𐕃𐕄𐕅𐕆𐕇𐕈𐕉𐕊𐕋𐕌𐕍𐕎𐕏𐕐𐕑𐕒𐕓𐕔𐕕𐕖𐕗𐕘𐕙𐕚𐕛𐕜𐕝𐕞𐕟𐕠𐕡𐕢𐕣𐕤𐕥𐕦𐕧𐕨𐕩𐕪𐕫𐕬𐕭𐕮𐕯𐕰𐕱𐕲𐕳𐕴𐕵𐕶𐕷𐕸𐕹𐕺𐕻𐕼𐕽𐕾𐕿𐖀𐖁𐖂𐖃𐖄𐖅𐖆𐖇𐖈𐖉𐖊𐖋𐖌𐖍𐖎𐖏𐖐𐖑𐖒𐖓𐖔𐖕𐖖𐖗𐖘𐖙𐖚𐖛𐖜𐖝𐖞𐖟𐖠𐖡𐖢𐖣𐖤𐖥𐖦𐖧𐖨𐖩𐖪𐖫𐖬𐖭𐖮𐖯𐖰𐖱𐖲𐖳𐖴𐖵𐖶𐖷𐖸𐖹𐖺𐖻𐖼𐖽𐖾𐖿𐗀𐗁𐗂𐗃𐗄𐗅𐗆𐗇𐗈𐗉𐗊𐗋𐗌𐗍𐗎𐗏𐗐𐗑𐗒𐗓𐗔𐗕𐗖𐗗𐗘𐗙𐗚𐗛𐗜𐗝𐗞𐗟𐗠𐗡𐗢𐗣𐗤𐗥𐗦𐗧𐗨𐗩𐗪𐗫𐗬𐗭𐗮𐗯𐗰𐗱𐗲𐗳𐗴𐗵𐗶𐗷𐗸𐗹𐗺𐗻𐗼𐗽𐗾𐗿𐘀𐘁𐘂𐘃𐘄𐘅𐘆𐘇𐘈𐘉𐘊𐘋𐘌𐘍𐘎𐘏𐘐𐘑𐘒𐘓𐘔𐘕𐘖𐘗𐘘𐘙𐘚𐘛𐘜𐘝𐘞𐘟𐘠𐘡𐘢𐘣𐘤𐘥𐘦𐘧𐘨𐘩𐘪𐘫𐘬𐘭𐘮𐘯𐘰𐘱𐘲𐘳𐘴𐘵𐘶𐘷𐘸𐘹𐘺𐘻𐘼𐘽𐘾𐘿𐙀𐙁𐙂𐙃𐙄𐙅𐙆𐙇𐙈𐙉𐙊𐙋𐙌𐙍𐙎𐙏𐙐𐙑𐙒𐙓𐙔𐙕𐙖𐙗𐙘𐙙𐙚𐙛𐙜𐙝𐙞𐙟𐙠𐙡𐙢𐙣𐙤𐙥𐙦𐙧𐙨𐙩𐙪𐙫𐙬𐙭𐙮𐙯𐙰𐙱𐙲𐙳𐙴𐙵𐙶𐙷𐙸𐙹𐙺𐙻𐙼𐙽𐙾𐙿𐚀𐚁𐚂𐚃𐚄𐚅𐚆𐚇𐚈𐚉𐚊𐚋𐚌𐚍𐚎𐚏𐚐𐚑𐚒𐚓𐚔𐚕𐚖𐚗𐚘𐚙𐚚𐚛𐚜𐚝𐚞𐚟𐚠𐚡𐚢𐚣𐚤𐚥𐚦𐚧𐚨𐚩𐚪𐚫𐚬𐚭𐚮𐚯𐚰𐚱𐚲𐚳𐚴𐚵𐚶𐚷𐚸𐚹𐚺𐚻𐚼𐚽𐚾𐚿𐛀𐛁𐛂𐛃𐛄𐛅𐛆𐛇𐛈𐛉𐛊𐛋𐛌𐛍𐛎𐛏𐛐𐛑𐛒𐛓𐛔𐛕𐛖𐛗𐛘𐛙𐛚𐛛𐛜𐛝𐛞𐛟𐛠𐛡𐛢𐛣𐛤𐛥𐛦𐛧𐛨𐛩𐛪𐛫𐛬𐛭𐛮𐛯𐛰𐛱𐛲𐛳𐛴𐛵𐛶𐛷𐛸𐛹𐛺𐛻𐛼𐛽𐛾𐛿𐜀𐜁𐜂𐜃𐜄𐜅𐜆𐜇𐜈𐜉𐜊𐜋𐜌𐜍𐜎𐜏𐜐𐜑𐜒𐜓𐜔𐜕𐜖𐜗𐜘𐜙𐜚𐜛𐜜𐜝𐜞𐜟𐜠𐜡𐜢𐜣𐜤𐜥𐜦𐜧𐜨𐜩𐜪𐜫𐜬𐜭𐜮𐜯𐜰𐜱𐜲𐜳𐜴𐜵𐜶𐜷𐜸𐜹𐜺𐜻𐜼𐜽𐜾𐜿𐝀𐝁𐝂𐝃𐝄𐝅𐝆𐝇𐝈𐝉𐝊𐝋𐝌𐝍𐝎𐝏𐝐𐝑𐝒𐝓𐝔𐝕𐝖𐝗𐝘𐝙𐝚𐝛𐝜𐝝𐝞𐝟𐝠𐝡𐝢𐝣𐝤𐝥𐝦𐝧𐝨𐝩𐝪𐝫𐝬𐝭𐝮𐝯𐝰𐝱𐝲𐝳𐝴𐝵𐝶𐝷𐝸𐝹𐝺𐝻𐝼𐝽𐝾𐝿𐞀𐞁𐞂𐞃𐞄𐞅𐞆𐞇𐞈𐞉𐞊𐞋𐞌𐞍𐞎𐞏𐞐𐞑𐞒𐞓𐞔𐞕𐞖𐞗𐞘𐞙𐞚𐞛𐞜𐞝𐞞𐞟𐞠𐞡𐞢𐞣𐞤𐞥𐞦𐞧𐞨𐞩𐞪𐞫𐞬𐞭𐞮𐞯𐞰𐞱𐞲𐞳𐞴𐞵𐞶𐞷𐞸𐞹𐞺𐞻𐞼𐞽𐞾𐞿𐟀𐟁𐟂𐟃𐟄𐟅𐟆𐟇𐟈𐟉𐟊𐟋𐟌𐟍𐟎𐟏𐟐𐟑𐟒𐟓𐟔𐟕𐟖𐟗𐟘𐟙𐟚𐟛𐟜𐟝𐟞𐟟𐟠𐟡𐟢𐟣𐟤𐟥𐟦𐟧𐟨𐟩𐟪𐟫𐟬𐟭𐟮𐟯𐟰𐟱𐟲𐟳𐟴𐟵𐟶𐟷𐟸𐟹𐟺𐟻𐟼𐟽𐟾𐟿𐠀𐠁𐠂𐠃𐠄𐠅𐠆𐠇𐠈𐠉𐠊𐠋𐠌𐠍𐠎𐠏𐠐𐠑𐠒𐠓𐠔𐠕𐠖𐠗𐠘𐠙𐠚𐠛𐠜𐠝𐠞𐠟𐠠𐠡𐠢𐠣𐠤𐠥𐠦𐠧𐠨𐠩𐠪𐠫𐠬𐠭𐠮𐠯𐠰𐠱𐠲𐠳𐠴𐠵𐠶𐠷𐠸𐠹𐠺𐠻𐠼𐠽𐠾𐠿𐡀𐡁𐡂𐡃𐡄𐡅𐡆𐡇𐡈𐡉𐡊𐡋𐡌𐡍𐡎𐡏𐡐𐡑𐡒𐡓𐡔𐡕𐡖𐡗𐡘𐡙𐡚𐡛𐡜𐡝𐡞𐡟𐡠𐡡𐡢𐡣𐡤𐡥𐡦𐡧𐡨𐡩𐡪𐡫𐡬𐡭𐡮𐡯𐡰𐡱𐡲𐡳𐡴𐡵𐡶𐡷𐡸𐡹𐡺𐡻𐡼𐡽𐡾𐡿𐢀𐢁𐢂𐢃𐢄𐢅𐢆𐢇𐢈𐢉𐢊𐢋𐢌𐢍𐢎𐢏𐢐𐢑𐢒𐢓𐢔𐢕𐢖𐢗𐢘𐢙𐢚𐢛𐢜𐢝𐢞𐢟𐢠𐢡𐢢𐢣𐢤𐢥𐢦𐢧𐢨𐢩𐢪𐢫𐢬𐢭𐢮𐢯𐢰𐢱𐢲𐢳𐢴𐢵𐢶𐢷𐢸𐢹𐢺𐢻𐢼𐢽𐢾𐢿𐣀𐣁𐣂𐣃𐣄𐣅𐣆𐣇𐣈𐣉𐣊𐣋𐣌𐣍𐣎𐣏𐣐𐣑𐣒𐣓𐣔𐣕𐣖𐣗𐣘𐣙𐣚𐣛𐣜𐣝𐣞𐣟𐣠𐣡𐣢𐣣𐣤𐣥𐣦𐣧𐣨𐣩𐣪𐣫𐣬𐣭𐣮𐣯𐣰𐣱𐣲𐣳𐣴𐣵𐣶𐣷𐣸𐣹𐣺𐣻𐣼𐣽𐣾𐣿𐤀𐤁𐤂𐤃𐤄𐤅𐤆𐤇𐤈𐤉𐤊𐤋𐤌𐤍𐤎𐤏𐤐𐤑𐤒𐤓𐤔𐤕𐤖𐤗𐤘𐤙𐤚𐤛𐤜𐤝𐤞𐤟𐤠𐤡𐤢𐤣𐤤𐤥𐤦𐤧𐤨𐤩𐤪𐤫𐤬𐤭𐤮𐤯𐤰𐤱𐤲𐤳𐤴𐤵𐤶𐤷𐤸𐤹𐤺𐤻𐤼𐤽𐤾𐤿𐥀𐥁𐥂𐥃𐥄𐥅𐥆𐥇𐥈𐥉𐥊𐥋𐥌𐥍𐥎𐥏𐥐𐥑𐥒𐥓𐥔𐥕𐥖𐥗𐥘𐥙𐥚𐥛𐥜𐥝𐥞𐥟𐥠𐥡𐥢𐥣𐥤𐥥𐥦𐥧𐥨𐥩𐥪𐥫𐥬𐥭𐥮𐥯𐥰𐥱𐥲𐥳𐥴𐥵𐥶𐥷𐥸𐥹𐥺𐥻𐥼𐥽𐥾𐥿𐦀𐦁𐦂𐦃𐦄𐦅𐦆𐦇𐦈𐦉𐦊𐦋𐦌𐦍𐦎𐦏𐦐𐦑𐦒𐦓𐦔𐦕𐦖𐦗𐦘𐦙𐦚𐦛𐦜𐦝𐦞𐦟𐦠𐦡𐦢𐦣𐦤𐦥𐦦𐦧𐦨𐦩𐦪𐦫𐦬𐦭𐦮𐦯𐦰𐦱𐦲𐦳𐦴𐦵𐦶𐦷𐦸𐦹𐦺𐦻𐦼𐦽𐦾𐦿𐧀𐧁𐧂𐧃𐧄𐧅𐧆𐧇𐧈𐧉𐧊𐧋𐧌𐧍𐧎𐧏𐧐𐧑𐧒𐧓𐧔𐧕𐧖𐧗𐧘𐧙𐧚𐧛𐧜𐧝𐧞𐧟𐧠𐧡𐧢𐧣𐧤𐧥𐧦𐧧𐧨𐧩𐧪𐧫𐧬𐧭𐧮𐧯𐧰𐧱𐧲𐧳𐧴𐧵𐧶𐧷𐧸𐧹𐧺𐧻𐧼𐧽𐧾𐧿𐨀𐨁𐨂𐨃𐨄𐨅𐨆𐨇𐨈𐨉𐨊𐨋𐨌𐨍𐨎𐨏𐨐𐨑𐨒𐨓𐨔𐨕𐨖𐨗𐨘𐨙𐨚𐨛𐨜𐨝𐨞𐨟𐨠𐨡𐨢𐨣𐨤𐨥𐨦𐨧𐨨𐨩𐨪𐨫𐨬𐨭𐨮𐨯𐨰𐨱𐨲𐨳𐨴𐨵𐨶𐨷𐨹𐨺𐨸𐨻𐨼𐨽𐨾𐨿𐩀𐩁𐩂𐩃𐩄𐩅𐩆𐩇𐩈𐩉𐩊𐩋𐩌𐩍𐩎𐩏𐩐𐩑𐩒𐩓𐩔𐩕𐩖𐩗𐩘𐩙𐩚𐩛𐩜𐩝𐩞𐩟𐩠𐩡𐩢𐩣𐩤𐩥𐩦𐩧𐩨𐩩𐩪𐩫𐩬𐩭𐩮𐩯𐩰𐩱𐩲𐩳𐩴𐩵𐩶𐩷𐩸𐩹𐩺𐩻𐩼𐩽𐩾𐩿𐪀𐪁𐪂𐪃𐪄𐪅𐪆𐪇𐪈𐪉𐪊𐪋𐪌𐪍𐪎𐪏𐪐𐪑𐪒𐪓𐪔𐪕𐪖𐪗𐪘𐪙𐪚𐪛𐪜𐪝𐪞𐪟𐪠𐪡𐪢𐪣𐪤𐪥𐪦𐪧𐪨𐪩𐪪𐪫𐪬𐪭𐪮𐪯𐪰𐪱𐪲𐪳𐪴𐪵𐪶𐪷𐪸𐪹𐪺𐪻𐪼𐪽𐪾𐪿𐫀𐫁𐫂𐫃𐫄𐫅𐫆𐫇𐫈𐫉𐫊𐫋𐫌𐫍𐫎𐫏𐫐𐫑𐫒𐫓𐫔𐫕𐫖𐫗𐫘𐫙𐫚𐫛𐫜𐫝𐫞𐫟𐫠𐫡𐫢𐫣𐫤𐫦𐫥𐫧𐫨𐫩𐫪𐫫𐫬𐫭𐫮𐫯𐫰𐫱𐫲𐫳𐫴𐫵𐫶𐫷𐫸𐫹𐫺𐫻𐫼𐫽𐫾𐫿𐬀𐬁𐬂𐬃𐬄𐬅𐬆𐬇𐬈𐬉𐬊𐬋𐬌𐬍𐬎𐬏𐬐𐬑𐬒𐬓𐬔𐬕𐬖𐬗𐬘𐬙𐬚𐬛𐬜𐬝𐬞𐬟𐬠𐬡𐬢𐬣𐬤𐬥𐬦𐬧𐬨𐬩𐬪𐬫𐬬𐬭𐬮𐬯𐬰𐬱𐬲𐬳𐬴𐬵𐬶𐬷𐬸𐬹𐬺𐬻𐬼𐬽𐬾𐬿𐭀𐭁𐭂𐭃𐭄𐭅𐭆𐭇𐭈𐭉𐭊𐭋𐭌𐭍𐭎𐭏𐭐𐭑𐭒𐭓𐭔𐭕𐭖𐭗𐭘𐭙𐭚𐭛𐭜𐭝𐭞𐭟𐭠𐭡𐭢𐭣𐭤𐭥𐭦𐭧𐭨𐭩𐭪𐭫𐭬𐭭𐭮𐭯𐭰𐭱𐭲𐭳𐭴𐭵𐭶𐭷𐭸𐭹𐭺𐭻𐭼𐭽𐭾𐭿𐮀𐮁𐮂𐮃𐮄𐮅𐮆𐮇𐮈𐮉𐮊𐮋𐮌𐮍𐮎𐮏𐮐𐮑𐮒𐮓𐮔𐮕𐮖𐮗𐮘𐮙𐮚𐮛𐮜𐮝𐮞𐮟𐮠𐮡𐮢𐮣𐮤𐮥𐮦𐮧𐮨𐮩𐮪𐮫𐮬𐮭𐮮𐮯𐮰𐮱𐮲𐮳𐮴𐮵𐮶𐮷𐮸𐮹𐮺𐮻𐮼𐮽𐮾𐮿𐯀𐯁𐯂𐯃𐯄𐯅𐯆𐯇𐯈𐯉𐯊𐯋𐯌𐯍𐯎𐯏𐯐𐯑𐯒𐯓𐯔𐯕𐯖𐯗𐯘𐯙𐯚𐯛𐯜𐯝𐯞𐯟𐯠𐯡𐯢𐯣𐯤𐯥𐯦𐯧𐯨𐯩𐯪𐯫𐯬𐯭𐯮𐯯𐯰𐯱𐯲𐯳𐯴𐯵𐯶𐯷𐯸𐯹𐯺𐯻𐯼𐯽𐯾𐯿𐰀𐰁𐰂𐰃𐰄𐰅𐰆𐰇𐰈𐰉𐰊𐰋𐰌𐰍𐰎𐰏𐰐𐰑𐰒𐰓𐰔𐰕𐰖𐰗𐰘𐰙𐰚𐰛𐰜𐰝𐰞𐰟𐰠𐰡𐰢𐰣𐰤𐰥𐰦𐰧𐰨𐰩𐰪𐰫𐰬𐰭𐰮𐰯𐰰𐰱𐰲𐰳𐰴𐰵𐰶𐰷𐰸𐰹𐰺𐰻𐰼𐰽𐰾𐰿𐱀𐱁𐱂𐱃𐱄𐱅𐱆𐱇𐱈𐱉𐱊𐱋𐱌𐱍𐱎𐱏𐱐𐱑𐱒𐱓𐱔𐱕𐱖𐱗𐱘𐱙𐱚𐱛𐱜𐱝𐱞𐱟𐱠𐱡𐱢𐱣𐱤𐱥𐱦𐱧𐱨𐱩𐱪𐱫𐱬𐱭𐱮𐱯𐱰𐱱𐱲𐱳𐱴𐱵𐱶𐱷𐱸𐱹𐱺𐱻𐱼𐱽𐱾𐱿𐲀𐲁𐲂𐲃𐲄𐲅𐲆𐲇𐲈𐲉𐲊𐲋𐲌𐲍𐲎𐲏𐲐𐲑𐲒𐲓𐲔𐲕𐲖𐲗𐲘𐲙𐲚𐲛𐲜𐲝𐲞𐲟𐲠𐲡𐲢𐲣𐲤𐲥𐲦𐲧𐲨𐲩𐲪𐲫𐲬𐲭𐲮𐲯𐲰𐲱𐲲𐲳𐲴𐲵𐲶𐲷𐲸𐲹𐲺𐲻𐲼𐲽𐲾𐲿𐳀𐳁𐳂𐳃𐳄𐳅𐳆𐳇𐳈𐳉𐳊𐳋𐳌𐳍𐳎𐳏𐳐𐳑𐳒𐳓𐳔𐳕𐳖𐳗𐳘𐳙𐳚𐳛𐳜𐳝𐳞𐳟𐳠𐳡𐳢𐳣𐳤𐳥𐳦𐳧𐳨𐳩𐳪𐳫𐳬𐳭𐳮𐳯𐳰𐳱𐳲𐳳𐳴𐳵𐳶𐳷𐳸



# QUIZ

GZZ

```
Leonie@Laptop $ hexdump -C datei.txt
00000000  41 6d 20 45 6e 64 65 20 69 73 74 20 65 73 20 61 |Am Ende ist es a|
00000010  62 65 72 20 45 67 61 6c 20 6f 62 20 64 69 65 20 |ber Egal ob die |
00000020  4c 65 75 74 65 20 22 73 69 6e 6e 20 6d 61 63 68 |Leute "sinn mach|
00000030  65 6e 22 20 6f 64 65 72 20 22 73 69 6e 6e 20 65 |en" oder "sinn e|
00000040  72 67 65 62 65 6e 22 20 73 61 67 65 6e 2c 20 64 |rgeben" sagen, d|
00000050  65 6e 6e 20 73 69 65 20 72 65 64 65 6e 20 6d 65 |enn sie reden me|
00000060  69 73 74 65 6e 73 20 73 6f 77 69 65 73 6f 20 6e |istens sowieso n|
00000070  75 72 20 55 6e 73 69 6e 6e 21 0a |ur Unsinn!.|
0000007b
```

■ Welches Encoding hat die Datei?

- a) ISO-Latin
- b) UTF-16LE
- c) UTF-16BE



# QUIZ

GZZ

```
Leonie@Laptop $ hexdump -C datei.txt
00000000  41 6d 20 45 6e 64 65 20 69 73 74 20 65 73 20 61 |Am Ende ist es a|
00000010  62 65 72 20 45 67 61 6c 20 6f 62 20 64 69 65 20 |ber Egal ob die |
00000020  4c 65 75 74 65 20 22 73 69 6e 6e 20 6d 61 63 68 |Leute "sinn mach|
00000030  65 6e 22 20 6f 64 65 72 20 22 73 69 6e 6e 20 65 |en" oder "sinn e|
00000040  72 67 65 62 65 6e 22 20 73 61 67 65 6e 2c 20 64 |rgeben" sagen, d|
00000050  65 6e 6e 20 73 69 65 20 72 65 64 65 6e 20 6d 65 |enn sie reden me|
00000060  69 73 74 65 6e 73 20 73 6f 77 69 65 73 6f 20 6e |istens sowieso n|
00000070  75 72 20 55 6e 73 69 6e 6e 21 0a |ur Unsinn!.|
0000007b
```

■ Welches Encoding hat die Datei?

- a) ISO-Latin
- b) UTF-16LE
- c) UTF-16BE



# QUIZ

GZZ

- Was matcht der folgende Regex?
- '.o.....'
- a) Goethe
- b) Schiller
- c) Kant
- d) Lipschitz



# QUIZ

GZZ

■ Was matcht der folgende Regex?

■ '.o.....'

- a) Goethe
- b) Schiller
- c) Kant
- d) Lipschitz



# QUIZ

GZZ

■ Was matcht der folgende Regex?

■ `'..[m-z][^n]'`

- a) Seit
- b) Sein
- c) Kann
- d) Muss



# QUIZ

GZZ

■ Was matcht der folgende Regex?

■ `'..[m-z][^n]'`

- a) Seit
- b) Sein
- c) Kann
- d) Muss



# QUIZ

GZZ

- Was matcht der folgende Regex?
- '[A-Z][äöü]st'
- a) Übst
- b) Obst
- c) Löst
- d) Dust



# QUIZ

GZZ

- Was matcht der folgende Regex?
- '[A-Z][äöü]st'
- a) Übst
- b) Obst
- c) Löst
- d) Dust



# QUIZ

GZZ

- Welche der folgenden Strings werden von diesem Regex gematcht: „test“, „tttt“, „eeee“, „a“
- `'[a-z]{2,4}'`
- a) „a“
- b) „tttt“, „eeee“
- c) „test“
- d) „test“, „tttt“, „eeee“



# QUIZ

GZZ

- Welche der folgenden Strings werden von diesem Regex gematcht: „test“, „tttt“, „eeee“, „a“
- `'[a-z]{2,4}'`
- a) „a“
- b) „tttt“, „eeee“
- c) „test“
- d) „test“, „tttt“, „eeee“



# QUIZ

GZZ

- Was matcht der folgende Regex?
- `'.{1,3}'`
- a) Das
- b) Dies
- c) Sonst
- d) Selbst



# QUIZ

GZZ

- Was matcht der folgende Regex?
- `'.{1,3}'`
- a) Das
- b) Dies
- c) Sonst
- d) Selbst



# QUIZ

GZZ

■ Was matcht der folgende Regex?

■ 'a+.\*'

a) alle

b) Aal

c) Egaaal

d) Alles



# QUIZ

GZZ

■ Was matcht der folgende Regex?

■ 'a+.\*'

a) alle

b) Aal

c) Egaaal

d) Alles



# QUIZ

GZZ

- Was matcht der folgende Regex?
- 'x?B+.\*'
- a) Xenial
- b) Bauen
- c) Xylophon
- d) basteln



# QUIZ

GZZ

- Was matcht der folgende Regex?
- 'x?B+.\*'
- a) Xenial
- b) Bauen
- c) Xylophon
- d) basteln



# QUIZ

GZZ

- Was matcht der folgende Regex in diesem Text?
  - `'^\w+'`
  - "Wir verkaufen nicht nur Obst!  
Auch Döner!"
- a) „Wir“
  - b) „Döner!“
  - c) „Wir“ „verkaufen“ „nicht“ „nur“ „Auch“ „Döner“
  - d) „Wir“ „verkaufen“ „nicht“ „nur“ Obst“ „Auch“ Döner“



# QUIZ

GZZ

- Was matcht der folgende Regex in diesem Text?
  - `'^\w+'`
  - "Wir verkaufen nicht nur Obst!  
Auch Döner!"
- a) „Wir“
  - b) „Döner!“
  - c) „Wir“ „verkaufen“ „nicht“ „nur“ „Auch“ „Döner“
  - d) „Wir“ „verkaufen“ „nicht“ „nur“ Obst“ „Auch“ Döner“



# QUIZ

GZZ

- Was matcht der folgende Regex in diesem Text?
  - `'\w+@\w+\.\w{2,3}'`
  - “Ihr könnt Fragen zum Tutorium, Aufgaben, Python, RegExs etc. jederzeit an [leonie.weissweiler@campus.lmu.de](mailto:leonie.weissweiler@campus.lmu.de), oder an [kaiser@gmail.de](mailto:kaiser@gmail.de) (insbesondere Fragen zur Korrektur der Hausaufgaben) stellen.“
- a) nichts
  - b) „leonie.weissweiler@campus.lmu.de“
  - c) „kaiser@gmail.de“
  - d) „kaiser@gmail.de“ „leonie.weissweiler@campus.lmu.de“



# QUIZ

GZZ

- Was matcht der folgende Regex in diesem Text?
  - `'\w+@\w+\.\w{2,3}'`
  - “Ihr könnt Fragen zum Tutorium, Aufgaben, Python, RegExs etc. jederzeit an [leonie.weissweiler@campus.lmu.de](mailto:leonie.weissweiler@campus.lmu.de), oder an [kaiser@gmail.de](mailto:kaiser@gmail.de) (insbesondere Fragen zur Korrektur der Hausaufgaben) stellen.“
- a) nichts
  - b) „leonie.weissweiler@campus.lmu.de“
  - c) „kaiser@gmail.de“
  - d) „kaiser@gmail.de“ „leonie.weissweiler@campus.lmu.de“



# QUIZ

G35

- Was darf der Nutzer bei `r-xrw--wx`
  - a) Lesen
  - b) Lesen und Schreiben
  - c) Lesen und Ausführen
  - d) Schreiben und Ausführen



# QUIZ

G35

- Was darf der Nutzer bei `r-xrw--wx`
  - a) Lesen
  - b) Lesen und Schreiben
  - c) Lesen und Ausführen
  - d) Schreiben und Ausführen



# QUIZ

G35

- Was darf die Gruppe bei `rwX-wXrwx`
  - a) Lesen
  - b) Schreiben und Ausführen
  - c) Lesen und Ausführen
  - d) Lesen und Schreiben und Ausführen



# QUIZ

G35

- Was darf die Gruppe bei `rwX-wXrwx`
  - a) Lesen
  - b) Schreiben und Ausführen
  - c) Lesen und Ausführen
  - d) Lesen und Schreiben und Ausführen



# QUIZ

G35

- Was matcht 'x+?\w' auf "xxxxx" zuerst?
- a) **xxxxx**
- b) **xxxxx**
- c) **xxxxx**
- d) **xxxxx**



# QUIZ

G35

- Was matcht 'x+?\w' auf "xxxxx" zuerst?
- a) **xxxxx**
- b) **xxxxx**
- c) **xxxxx**
- d) **xxxxx**



# QUIZ

G35

- Was matcht `'(\w+ )+z'` auf "Sowas ist zu zart" zuerst?
  - a) **Sowas ist** zu zart
  - b) **Sowas ist zu** zart
  - c) **Sowas ist zu** zart
  - d) **Sowas ist zu zart**



# QUIZ

G35

- Was matcht `'(\w+ )+z'` auf "Sowas ist zu zart" zuerst?
  - a) **Sowas ist** zu zart
  - b) **Sowas ist zu** zart
  - c) **Sowas ist zu zart**
  - d) **Sowas ist zu zart**



- Was matcht '(\w+) \1' auf "ein kleiner kleiner Wald ist ein Baum" zuerst?
  - a) **ein kleiner** kleiner Wald ist ein Baum
  - b) **ein** kleiner kleiner Wald ist **ein** Baum
  - c) **ein kleiner kleiner Wald ist ein** Baum
  - d) ein **kleiner kleiner** Wald ist ein Baum
  - e) Die Regex enthält einen Syntaxfehler



- Was matcht '(\w+) \1' auf "ein kleiner kleiner Wald ist ein Baum" zuerst?
  - a) **ein kleiner** kleiner Wald ist ein Baum
  - b) ein kleiner kleiner Wald ist **ein** Baum
  - c) **ein kleiner kleiner Wald ist ein** Baum
  - d) ein **kleiner kleiner Wald ist ein** Baum
  - e) Die Regex enthält einen Syntaxfehler



# QUIZ

G35

- Was gibt der folgende Code aus?
- `def magic(n):  
 print(n*3 - 5)`

`magic(4)`

- a) 4
- b) 7
- c) -1; -2
- d) 11



# QUIZ

G35

- Was gibt der folgende Code aus?
- `def magic(n):  
 print(n*3 - 5)`

`magic(4)`

- a) 4
- b) 7
- c) -1; -2
- d) 11



# QUIZ

G35

■ Was gibt der folgende Code aus?

```
def magic(x):  
    print(x*7 - 2)  
    return x*7 + 9
```

```
a = magic(3)
```

- a) 19
- b) 30
- c) 19;30
- d) 3



# QUIZ

G35

■ Was gibt der folgende Code aus?

```
def magic(x):  
    print(x*7 - 2)  
    return x*7 + 9
```

```
a = magic(3)
```

- a) 19
- b) 30
- c) 19;30
- d) 3



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(x):  
    a = x*x  
    return x
```

```
print(magic(5))
```

- a) 5
- b) 10
- c) 25
- d) 30



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(x):  
    a = x*x  
    return x
```

```
print(magic(5))
```

- a) 5
- b) 10
- c) 25
- d) 30



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(a,b):  
    a = b  
    return a+b
```

```
print(magic(4,5))
```

- a) 8
- b) 9
- c) 10
- d) 11



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(a,b):  
    a = b  
    return a+b
```

```
print(magic(4,5))
```

- a) 8
- b) 9
- c) 10
- d) 11



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(a):  
    a = 5  
    print(7)  
    return 6
```

magic(9)

- a) 5
- b) 6
- c) 7
- d) 9



# WIEDERHOLUNG: FUNKTIONEN

GY7

- Was gibt der Code aus?

```
def magic(a):  
    a = 5  
    print(7)  
    return 6
```

magic(9)

- a) 5
- b) 6
- c) 7
- d) 9



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
def magic(t):  
    t.append(9)
```

```
z = [3,4,5]  
magic(z)  
print(z)
```

- a) [3,4]
- b) [4,5]
- c) [3,4,5]
- d) [3,4,5,9]



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
def magic(t):  
    t.append(9)
```

```
z = [3,4,5]  
magic(z)  
print(z)
```

- a) [3,4]
- b) [4,5]
- c) [3,4,5]
- d) [3,4,5,9]



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
def magic(c):  
    c = 7  
    print(c)
```

```
t = 3  
magic(t)
```

- a) 3
- b) 6
- c) 7
- d) 12



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
def magic(c):  
    c = 7  
    print(c)
```

```
t = 3  
magic(t)
```

- a) 3
- b) 6
- c) 7
- d) 12



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
x = 4
def magic():
    x = 3
```

```
magic()
print(x)
```

- a) 3
- b) 4
- c) 7
- d) 8



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
x = 4
def magic():
    x = 3
```

```
magic()
print(x)
```

- a) 3
- b) 4
- c) 7
- d) 8



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
x = 4
def magic():
    global x
    x = x + 3
```

```
magic()
magic()
print(x)
```

- a) 3
- b) 4
- c) 7
- d) 10



# CALL BY REFERENCE/ CALL BY VALUE

GY7

- Was gibt der Code aus?

```
x = 4
def magic():
    global x
    x = x + 3
```

```
magic()
magic()
```

```
print(x)
```

- a) 3
- b) 4
- c) 7
- d) 10



# CALL BY REFERENCE/ CALL BY VALUE

GVA

- Was gibt der Code aus?

```
def brian(n):  
    n = 2*n  
    z = 4+n  
    print(n)
```

```
x = 3  
z = 4  
brian(x)  
print(z)
```

- a) 3;4
- b) 6;4
- c) 3;10
- d) 6;10



# CALL BY REFERENCE/ CALL BY VALUE

GVA

- Was gibt der Code aus?

```
def brian(n):  
    n = 2*n  
    z = 4+n  
    print(n)
```

```
x = 3  
z = 4  
brian(x)  
print(z)
```

- a) 3;4
- b) 6;4
- c) 3;10
- d) 6;10



# CALL BY REFERENCE/ CALL BY VALUE

GVA

- Was gibt der Code aus?

```
def youneverknow(t):  
    t.append(1)  
    print(t)
```

```
z = [7,6,5]  
youneverknow(z)  
print(z)
```

- a) [7,6,5], [7,6,5]
- b) [7,6,5,1], [7,6,5]
- c) [7,6,5], [7,6,5,1]
- d) [7,6,5,1], [7,6,5,1]



# CALL BY REFERENCE/ CALL BY VALUE

GVA

- Was gibt der Code aus?

```
def youneverknow(t):  
    t.append(1)  
    print(t)
```

```
z = [7,6,5]  
youneverknow(z)  
print(z)
```

- a) [7,6,5], [7,6,5]
- b) [7,6,5,1], [7,6,5]
- c) [7,6,5], [7,6,5,1]
- d) [7,6,5,1], [7,6,5,1]



# QUIZ

GVA

- Was gibt der Code aus?

```
def magic(u):  
    if u == 0:  
        return 1  
    else:  
        return 2 * magic(u-1)  
print( magic(5) )
```

- a) 1
- b) 2
- c) 16
- d) 32



# QUIZ

GVA

- Was gibt der Code aus?

```
def magic(u):  
    if u == 0:  
        return 1  
    else:  
        return 2 * magic(u-1)  
print( magic(5) )
```

- a) 1
- b) 2
- c) 16
- d) 32



# QUIZ

GVA

- Was gibt der Code aus?

```
def magic(t):  
    if t == 1:  
        return 1  
    else:  
        return magic(t/2) + magic(t/2)  
print( magic(8) )
```

- a) 2
- b) 4
- c) 8
- d) 12



# QUIZ

GVA

- Was gibt der Code aus?

```
def magic(t):  
    if t == 1:  
        return 1  
    else:  
        return magic(t/2) + magic(t/2)  
print( magic(8) )
```

- a) 2
- b) 4
- c) 8
- d) 12



# WIEDERHOLUNG

65Y

- Was gibt der Code aus?

```
def magic(t):  
    if t >= 16:  
        return 1  
    else:  
        return magic(t*2) * 2  
print( magic(4) )
```

- a) 1
- b) 2
- c) 4
- d) 8



# WIEDERHOLUNG

65Y

- Was gibt der Code aus?

```
def magic(t):  
    if t >= 16:  
        return 1  
    else:  
        return magic(t*2) * 2  
print( magic(4) )
```

- a) 1
- b) 2
- c) 4
- d) 8

