

PROPÄDEUTIKUM INFORMATIK

SOSE 2018

Martin Mehlhose

WIDERHOLUNG

- For-Schleifen
- While-Schleifen

SCHLEIFEN

FOR-SCHLEIFE

- Syntax: **for var in Iterable: Anweisungen**
- var ist eine selbstgewählte Variable
- Iterable kann jede Collection von Daten sein, die man durchlaufen kann. z.B.: Listen, Tupel, usw.
- Für uns vorerst nur Zahlen

BEISPIEL 10 MAL HELLO WORLD

AUSGEBEN

```
for count in range(10):  
    print("HelloWorld")
```

FUNKTION RANGE()

- Die Funktion Range gibt grundsätzlich eine Sequenz von Zahlen zurück
- `range(n)` liefert alle Zahlen von 0 bis $n-1$
- `range(start, ende)` liefert alle Zahlen von start bis stop-1
- `range(start, stop, step)` wie bisher, aber mit step kann die Schrittweite beeinflusst werden

GIB ALLE GERADEN ZAHLEN BIS 100 AUS

```
for count in range (0, 100, 2)  
    print(count)
```

ODER MIT IF-ANWEISUNG

```
for count in range(100):  
    if(count % 2 == 0):  
        print(count)
```

WHILE-SCHLEIFE

- Syntax: **while(condition): statements**
- condition : beliebiger boolescher Ausdruck. Die schleife wird so lange ausgeführt, wie dieser zu True ausgewertet wird.
- Wichtig: Bei der Abbruchbedingung aufpassen, dass keine Endlosschleifen entstehen!

GIB ALLE ZAHLEN BIS 100 AUS

```
count =0

while(count <= 100):
    print(count)
    count = count +1

# Endlosschleife
while(True):
    print(count)
```

SCHLEIFEN

- Zwei besondere Anweisungen
- **continue** Der aktuelle Schleifendurchlauf wird beendet und der Schleifenrumpf von vorn begonnen
- **break** Die Schleife wird verlassen

LISTEN

LISTEN

- Listen dienen dem "effizientem" Speichern großer Datenmengen
- Listen werden durch den [] Operator markiert. Zum Beispiel [1,2,3,4,5,6,7]
- Die Daten müssen nicht alle den selben Typ haben

LISTEN

- Zugriff auf Listen Elemente über den Index.
Achtung Listen sind 0 basiert!
- Zugriff erfolgt dann über eckige Klammern, z.B.
`list[0]` für das erste Listenelement
- Die Elemente können über den Index sowohl
ausgelesen als auch bearbeitet werden
- sollen alle Elemente einer Liste bearbeitet werden,
dass am besten eine for-Schleife benutzen

LISTEN

- Wichtige Funktionen:
- `len(list)` : gibt sie Länge der Liste zurück
- `mylist.pop(i)` : entfernt das i'te Element aus der Liste
- `mylist.append(i)` : hängt i an das Ende der Liste an
- Zusammenfügen von zwei Listen über + Operator

MASCHINENZAHLEN

MASCHINENZAHLEN

- Wir rechnen hier mit 8 bit Zahlen
- Also Zahlen im Bereich von -128 bis + 127
- Umwandlung Dez in Bin durch iteratives teilen durch 2 und merken der Reste

MASCHINENZAHLEN

- Umrechnung von Bin nach Dec durch Multiplikation der letzten 7 bit mit ihrer jeweiligen Wertigkeit
- Beispiel: 0 0 0 0 1 0 1 0

MASCHINENZAHLEN

- Umrechnung von Bin nach Dec durch Multiplikation der letzten 7 bit mit ihrer jeweiligen Wertigkeit
- Beispiel: 0 0 0 0 1 0 1 0
- Lösung: $0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 = 2 + 8 = 10$