

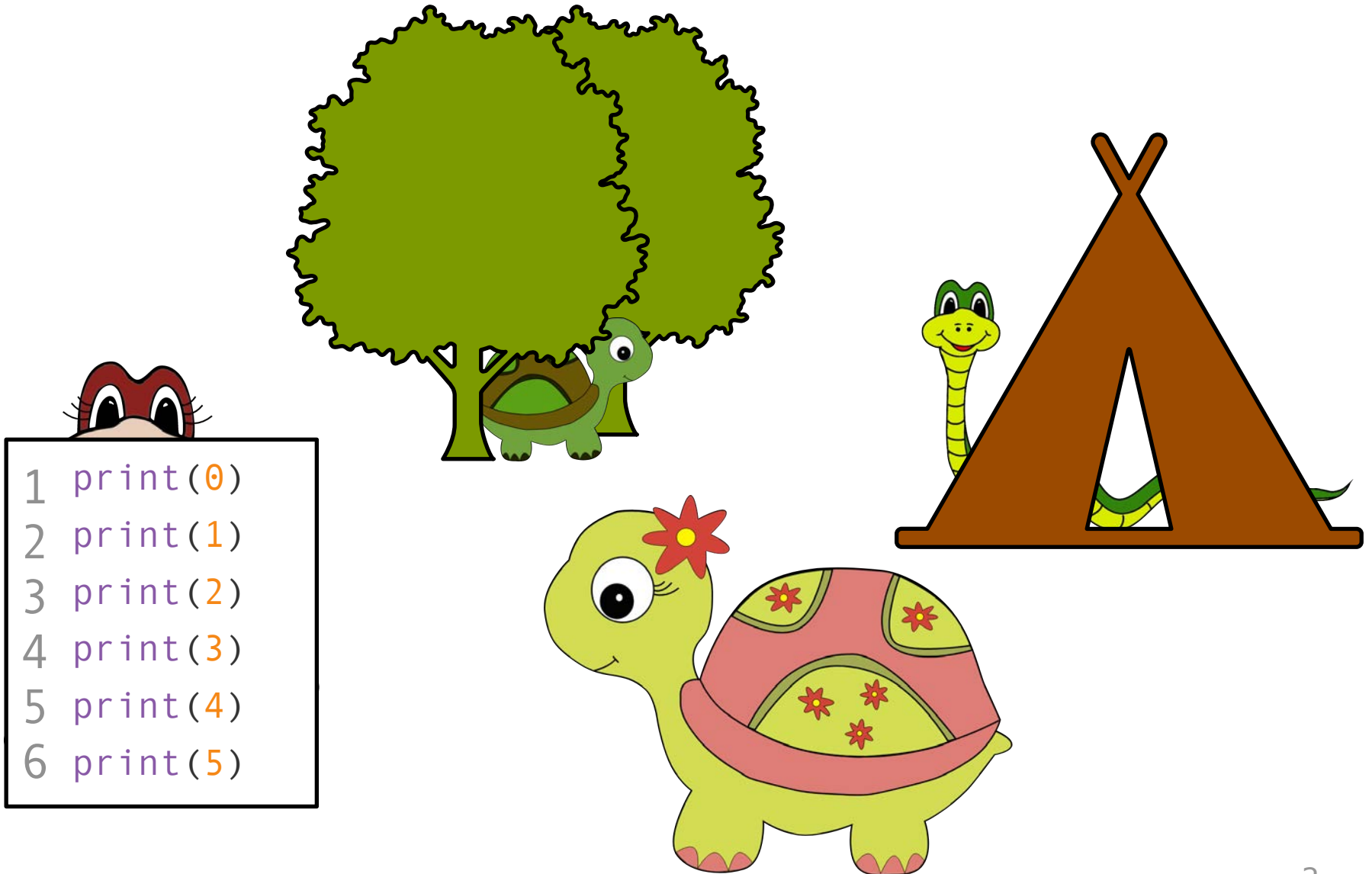


Programmieren  
lernen mit Python

# Schleifen

**Kira Grammel, Nina Ihde, Sebastian Serth & Selina Reinhard**  
Hasso-Plattner-Institut  
Universität Potsdam

# Story



# Programmablauf

---

```
1 zahl = 0
2 #TODO: Anweisung zum Wiederholen des folgenden Codes einfügen
3     print(zahl)
4     zahl = zahl + 1
```

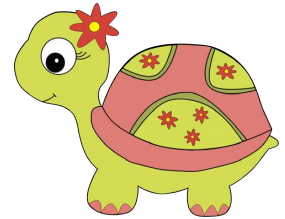
```
0
1
2
```

## Schleifen

- Können mehrere ähnliche Aktionen ausführen
- Dazu werden Teile des Codes mehrfach wiederholt

# For-Schleifen

```
1 for zahl in (0, 1, 2, 3, 4, 5):  
2     print(zahl)
```



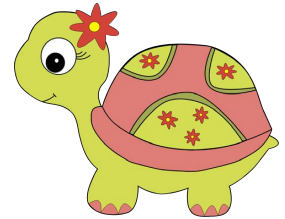
```
0  
1  
2  
3  
4  
5
```

## For-Schleifen

- `for` heißt "für"
- `zahl` ist eine Variable, die nacheinander die verschiedenen Werte in der Aufzählung annimmt
- Die Aktion wird für jede Belegung von `zahl` einmal ausgeführt

# Zahlenfolgen

```
1 for zahl in range(6):  
2     print(zahl)
```



```
0  
1  
2  
3  
4  
5
```

## **range()**

- `range` heißt "Reihe"
- Gibt eine Aufzählung von Zahlen zurück
- Die Zahlen beginnen bei 0 und enden vor der übergebenen Zahl
- `range(6)` gibt (0, 1, 2, 3, 4, 5) zurück

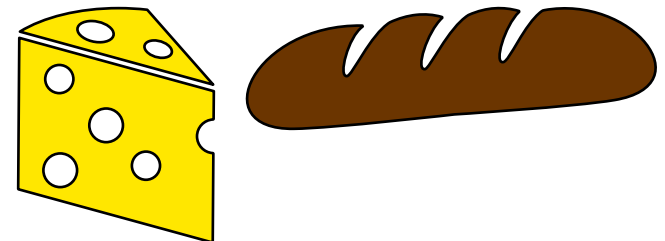
# For-Schleifen

```
1 for wort in ("bald", "gibt", "es", "essen"):  
2     print(wort)
```

```
bald  
gibt  
es  
essen
```

## For-Schleifen

- Können auch Strings in der Aufzählung verwenden



# Summe berechnen

```

1 summe = 0
2 for alter in (7, 11, 13, 15, 12):
3     summe = summe + alter
4 print(summe)

```

58

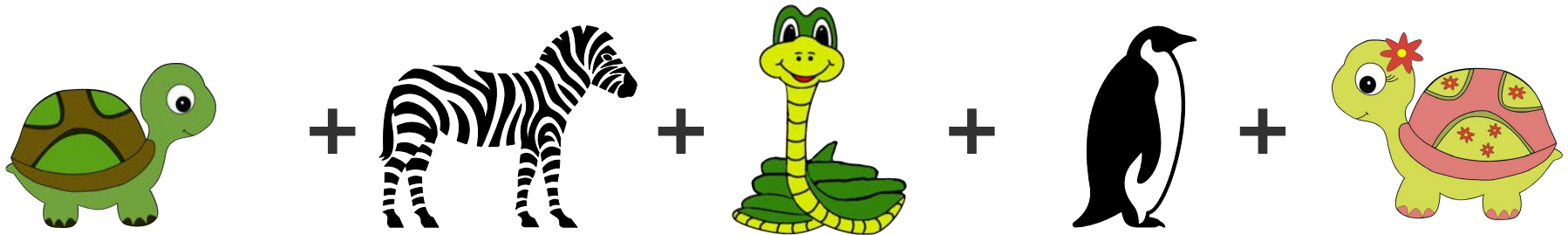
alter = 7  
summe = 7

alter = 11  
summe = 18

alter = 13  
summe = 31

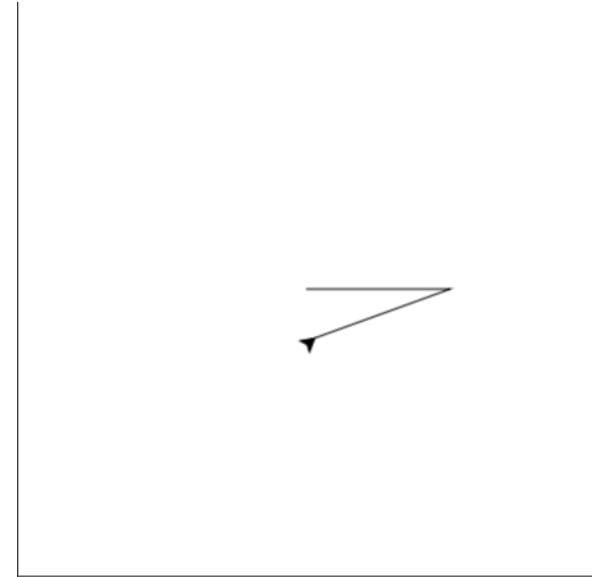
alter = 15  
summe = 46

alter = 12  
summe = 58



# Schleifen in Turtle

```
1 from turtle import *
2 forward(100)
3 right(160)
4 forward(100)
5 right(160)
```



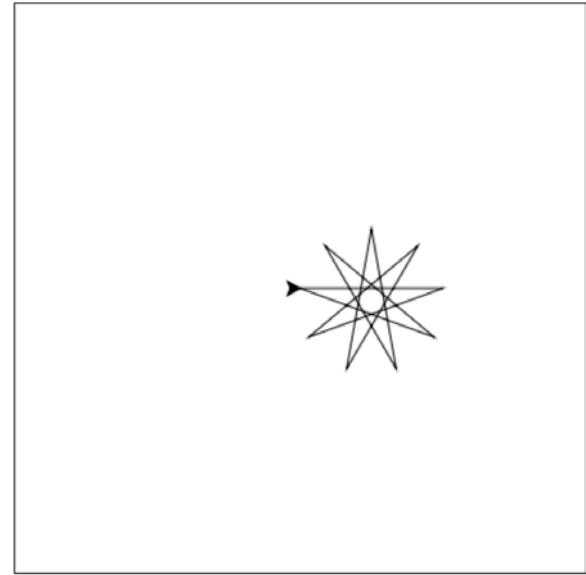
## Stern zeichnen

- Die Schleife soll 9 mal ausgeführt werden, um wieder am Anfangspunkt zu landen
- Wir erzeugen dazu eine Aufzählung mit 9 Elementen (0 bis 8)
- Die Variable a nimmt diese Werte an, wird aber nicht verwendet



# Schleifen in Turtle

```
1 from turtle import *
2 for a in range(9):
3     forward(100)
4     right(160)
```




## Stern zeichnen

- Die Schleife soll 9 mal ausgeführt werden, um wieder am Anfangspunkt zu landen
- Wir erzeugen dazu eine Aufzählung mit 9 Elementen (0 bis 8)
- Die Variable a nimmt diese Werte an, wird aber nicht verwendet

# Programmablauf

```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.")  
3 print("Wir haben keine Zeit mehr.")
```



Haben wir  
noch Zeit zum  
Spielen?

JA 😊

# Programmablauf

---


```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.") ←  
3     print("Wir haben keine Zeit mehr.")
```

Lasst uns noch eine Runde spielen.

# Programmablauf

```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.")  
3 print("Wir haben keine Zeit mehr.")
```

Lasst uns noch eine Runde spielen.



Haben wir  
noch Zeit zum  
Spielen?

JA 😊

# Programmablauf

---


```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.") ←  
3 print("Wir haben keine Zeit mehr.")
```

```
Lasst uns noch eine Runde spielen.  
Lasst uns noch eine Runde spielen.
```

# Programmablauf

```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.")  
3 print("Wir haben keine Zeit mehr.")
```

Lasst uns noch eine Runde spielen.  
Lasst uns noch eine Runde spielen.




Haben wir  
noch Zeit zum  
Spielen?

NEIN 😞

# Programmablauf

```
1 #TODO: Anweisung, die eingerückte Code Zeilen so lange  
   wiederholt wie eine Bedingung wahr ist  
2     print("Lasst uns noch eine Runde spielen.")  
3 print("Wir haben keine Zeit mehr.")
```



```
Lasst uns noch eine Runde spielen.  
Lasst uns noch eine Runde spielen.  
Wir haben keine Zeit mehr.
```

## While-Schleife

- `while` Bedingung:
- Zu Beginn der Schleife wird die Bedingung geprüft
- Wenn diese wahr ist, wird die Aktion in der Schleife ausgeführt
- Danach wird die Bedingung erneut geprüft

# While-Schleifen

```
1 verbleibende_zeit = 40
2 while verbleibende_zeit >= 15:
3     print("Lasst uns noch eine Runde spielen")
4     verbleibende_zeit = verbleibende_zeit - 15
5     print("Verbleibende Zeit:", verbleibende_zeit)
6 print("Jetzt gibt es Essen")
```

## 1. Schleifendurchlauf

```
Lasst uns noch eine Runde spielen
Verbleibende Zeit: 25
```



# While-Schleifen

```
1 verbleibende_zeit = 40
2 while verbleibende_zeit >= 15:
3     print("Lasst uns noch eine Runde spielen")
4     verbleibende_zeit = verbleibende_zeit - 15
5     print("Verbleibende Zeit:", verbleibende_zeit)
6 print("Jetzt gibt es Essen")
```

## 2. Schleifendurchlauf

```
Lasst uns noch eine Runde spielen
Verbleibende Zeit: 25
Lasst uns noch eine Runde spielen
Verbleibende Zeit: 10
```

# While-Schleifen

---

```
1 verbleibende_zeit = 40
2 while verbleibende_zeit >= 15:
3     print("Lasst uns noch eine Runde spielen")
4     verbleibende_zeit = verbleibende_zeit - 15
5     print("Verbleibende Zeit:", verbleibende_zeit)
6 print("Jetzt gibt es Essen")
```

```
Lasst uns noch eine Runde spielen
Verbleibende Zeit: 25
Lasst uns noch eine Runde spielen
Verbleibende Zeit: 10
Jetzt gibt es Essen
```

# Zusammenfassung

- Mit Schleifen kann man Aktionen mehrfach ausführen
- for-Schleifen laufen für jedes Element der übergebenen Aufzählung die Schleife einmal durch
  - `for variable in Aufzählung:`
- `range(n)` gibt eine Aufzählung von 0 bis n-1 zurück
- while-Schleifen wiederholen Aktionen so lange die übergebene Bedingung erfüllt ist
  - `while Bedingung:`

