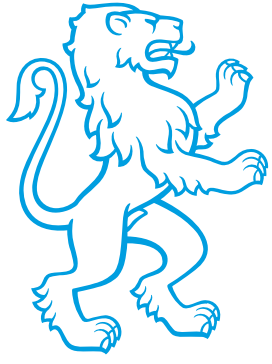


Kantonsschule Im Lee

Informatik: Programmieren

👨‍🍳 Kapitel 04: **Neue Funktionen definieren**

Wiederholung letztes Mal

```
import turtle as t

for _ in range(6):
    t.fd(100)
    t.rt(360/6)
```

- Wie nennt man die Ausdrücke auf den Zeilen 4 und 5?

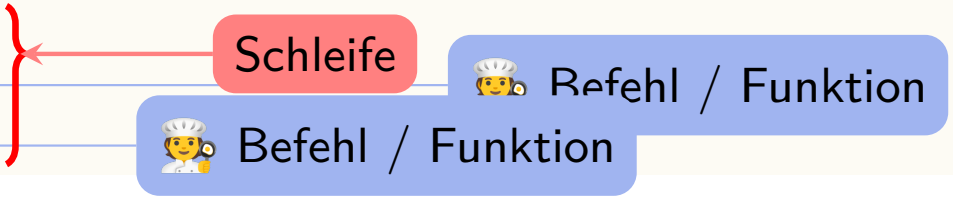
Wiederholung letztes Mal

```
import turtle as t

for _ in range(6):
    t.fd(100)
    t.rt(360/6)
```

- Wie nennt man die Ausdrücke auf den Zeilen 4 und 5?
- Wie nennt man Zeilen 3 bis 5?

Wiederholung letztes Mal

```
import turtle as t

for _ in range(6):
    t.fd(100)
    t.rt(360/6)
```



- Wie nennt man die Ausdrücke auf den Zeilen 4 und 5?
- Wie nennt man Zeilen 3 bis 5?
- Wie nennt man Zeilen 4-5?

Wiederholung letztes Mal

```
import turtle as t

for _ in range(6):
    t.fd(100)
    t.rt(360/6)
```

- Wie nennt man die Ausdrücke auf den Zeilen 4 und 5?
- Wie nennt man Zeilen 3 bis 5?
- Wie nennt man Zeilen 4-5?

Wiederholung letztes Mal

```
import turtle as t

for _ in range(6):
    t.fd(100)
    t.rt(360/6)
```

- Wie nennt man die Ausdrücke auf den Zeilen 4 und 5?
- Wie nennt man Zeilen 3 bis 5?
- Wie nennt man Zeilen 4-5?

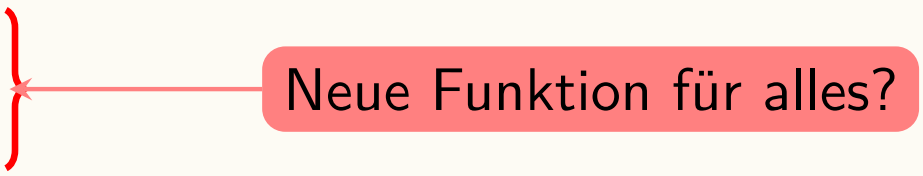
- Mit **Funktionen** können wir den Computer eine abstrakte Tätigkeit ausführen lassen
- **Parameter** dienen dazu, die Funktionen *parametrisieren* (Analogie: Kochrezept für 2 oder 4 Personen)
- **Schleifen** erlauben es, repetitive Tätigkeiten kompakt zu beschreiben
- Unterschiedliche **Datentypen** für unterschiedliche Zwecke

Neue Funktion definieren

```
import turtle as t

# zeichne ein Quadrat
for _ in range(4):
    t.fd(100)  # Funktion
    t.rt(90)   # Funktion

t.done()
```

Neue Funktion definieren

```
import turtle as t

# zeichne ein Quadrat
for _ in range(4):
    t.fd(100)   # Funktion
    t.rt(90)    # Funktion

t.done()
```



Neue Funktionen definieren

```
import turtle as t

def quadrat():
    for _ in range(4):
        t.fd(100)
        t.rt(90)

quadrat()
t.rt(45)
quadrat()

t.done()
```

- Definition = `def` = Funktion
- Ähnlich einem Kochrezept 📄
  - Beschreibung der auszuführenden Schritte
  - Wird erst ausgeführt, sobald aufgerufen! 👨‍🍳

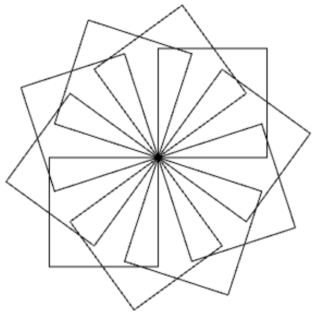
Neue Funktionen definieren

```
import turtle as t

def quadrat():
    for _ in range(4):
        t.fd(100)
        t.rt(90)

quadrat()
t.rt(45)
quadrat()

t.done()
```

- Definition = `def` = Funktion
- Ähnlich einem Kochrezept
  - Beschreibung der auszuführenden Schritte
  - Wird erst ausgeführt, sobald aufgerufen!

Viereck-Blume

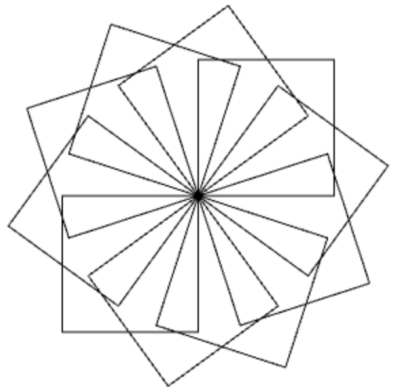
```
import turtle as t

for _ in range(10):
    for _ in range(4):
        t.fd(100)
        t.rt(360 / 4)
    t.rt(360 / 10)

t.done()
```



```
import turtle as t

def viereck():
    for _ in range(4):
        t.fd(100)
        t.rt(360 / 4)

def viereck_blume():
    for _ in range(10):
        viereck()
        t.rt(360/10)

viereck_blume()

t.done()
```

Neue Funktionen definieren

```
import turtle as t

def viereck():
    for _ in range(4):
        t.fd(100)
        t.rt(360/4)

def viereck_blume():
    for _ in range(10):
        viereck()
        t.rt(360/10)

viereck_blume()
```

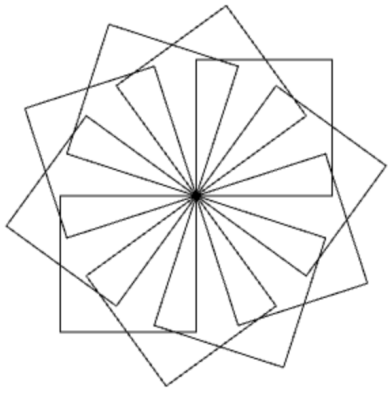


- **Modularität**: Zerlegung des Programms in kleinere Stücke
- **Übersichtlicher**!

Neue Funktionen definieren

```
import turtle as t

def viereck():
    for _ in range(4):
        t.fd(100)
        t.rt(360/4)

def viereck_blume():
    for _ in range(10):
        viereck()
        t.rt(3
```


```
viereck_blume()
```



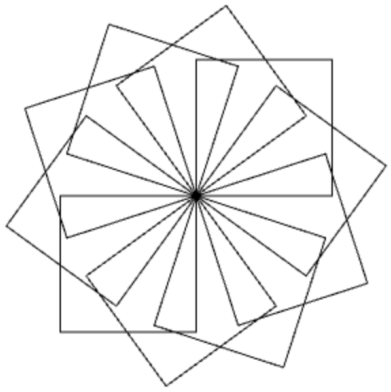
- **Modularität**: Zerlegung des Programms in kleinere Stücke
- **Übersichtlicher**!

Neue Funktionen definieren

```
import turtle as t

def viereck():
    for _ in range(4):
        t.fd(100)
        t.rt(3

def viereck_blume():
    for _ in range(10):
        viereck()
        t.rt(3

viereck_blume()
```

Aufruf

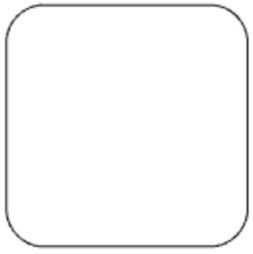
Aufruf



- **Modularität**: Zerlegung des Programms in kleinere Stücke
- **Übersichtlicher**!

Funktionen & Schleifen

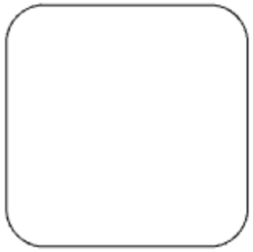
```
import turtle as t

for _ in range(4):
    for _ in range(9):
        t.fd(4)
        t.lt(10)
    t.fd(100)

t.done()
```



```
import turtle as t

def viertelkreis():
    for _ in range(9):
        t.fd(4)
        t.lt(10)

def abgerundetes_quadrat():
    for _ in range(4):
        viertelkreis()
        t.fd(100)

abgerundetes_quadrat()

t.done()
```

Funktionen & Schleifen

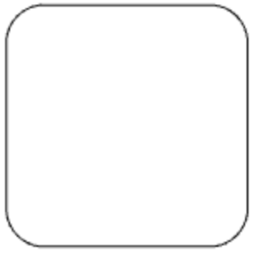
```
import turtle as t

for _ in range(4):
    for _ in range(9):
        t.fd(4)
        t.lt(10)
    t.fd(100)

t.done()
```



```
import turtle as t

def viertelkreis():
    for _ in range(9):
        t.fd(4)
        t.lt(10)

def abgerundetes_quadrat():
    for _ in range(4):
        viertelkreis()
        t.fd(100)

abgerundetes_quadrat()

t.done()
```

Funktionen & Schleifen

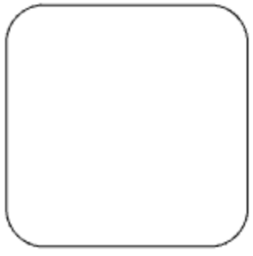
```
import turtle as t

for _ in range(4):
    for _ in range(9):
        t.fd(4)
        t.lt(10)
    t.fd(100)

t.done()
```



```
import turtle as t

def viertelkreis():
    for _ in range(9):
        t.fd(4)
        t.lt(10)

def abgerundetes_quadrat():
    for _ in range(4):
        viertelkreis()
        t.fd(100)

abgerundetes_quadrat()

t.done()
```

Funktionen & Schleifen

```
import turtle as t

for _ in range(4):
    for _ in range(9):
        t.fd(4)
        t.lt(10)
    t.fd(100)

t.done()
```



```
import turtle as t

def viertelkreis():
    for _ in range(9):
        t.fd(4)
        t.lt(10)

def abgerundetes_quadrat():
    for _ in range(4):
        viertelkreis()
        t.fd(100)

abgerundetes_quadrat()

t.done()
```

Übersichtlichkeit

- Innere Schleifen → Funktionen
- Sprechende Definitions-Namen

Zusammenfassung

Nutzen von **Funktionen**

- Problem in **Teilprobleme** gliedern

Zusammenfassung

Nutzen von **Funktionen**

- Problem in **Teilprobleme** gliedern
- **Übersichtlichkeit** erhöhen

Zusammenfassung

Nutzen von **Funktionen**

- Problem in **Teilprobleme** gliedern
- **Übersichtlichkeit** erhöhen
- **Sinnstiftender** Name für jede Definition!

Zusammenfassung

Nutzen von **Funktionen**

- Problem in **Teilprobleme** gliedern
- **Übersichtlichkeit** erhöhen
- **Sinnstiftender** Name für jede Definition!
- Im „Hauptprogramm“ Funktionen zusammensetzen

Auftrag: Skript, Kapitel 4

- Skript immer zuerst durchlesen 😉

  „ *A small step for students but a giant leap for the teacher...* “

- Übungen in VS Code und auf Moodle lösen
- 🖉 Aufgaben 4.1, 4.2
- ↑ Moodle-Test „Kapitel 4 (Teil 1)“