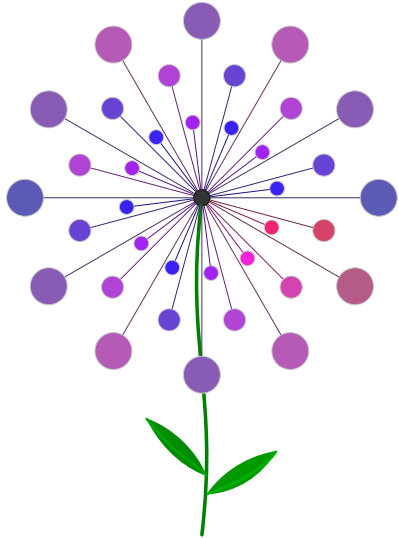


FreeFlower

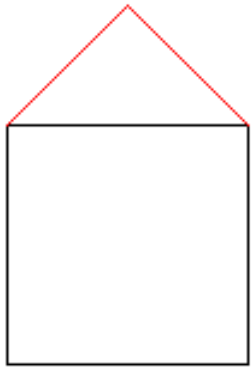
Informatik: Programmieren

Kapitel 01: **Einführung**

Was ist Programmieren?

Beispiel: `haus_guesscode.py`

Schreiben Sie einen Code, um dieses Haus zu zeichnen!



Mit den Befehlen `forward(pixelanzahl)`, `left(winkel)`, `right(winkel)` etc. können Sie die Turtle steuern, um Linien zu zeichnen. Wie könnte der Code aussehen?

Weshalb Programmieren?



Programmieren = “Super-Power”, um eigene Ideen umzusetzen und Probleme zu lösen. Es ermöglicht uns, die Welt um uns herum zu gestalten und zu automatisieren.

Weshalb Programmieren?



Programmieren = “Super-Power”, um eigene Ideen umzusetzen und Probleme zu lösen. Es ermöglicht uns, die Welt um uns herum zu gestalten und zu automatisieren.

- **Kreativität**: Eigene Ideen umsetzen, Probleme lösen

Weshalb Programmieren?



Programmieren = “Super-Power”, um eigene Ideen umzusetzen und Probleme zu lösen. Es ermöglicht uns, die Welt um uns herum zu gestalten und zu automatisieren.

- **Kreativität**: Eigene Ideen umsetzen, Probleme lösen
- **Automatisierung**: Wiederkehrende Aufgaben vereinfachen

Weshalb Programmieren?



Programmieren = “Super-Power”, um eigene Ideen umzusetzen und Probleme zu lösen. Es ermöglicht uns, die Welt um uns herum zu gestalten und zu automatisieren.

- **Kreativität**: Eigene Ideen umsetzen, Probleme lösen
- **Automatisierung**: Wiederkehrende Aufgaben vereinfachen
- **Zukunftssicherheit**: Digitale Kompetenzen sind gefragt

Einführung

Was erwartet Sie?

- **Programmieren**: Grundkonzepte (Schleifen, Verzweigungen, Variablen)
- **Robotik**: Steuerung von Technik (Roboter, Sensoren, Aktoren)

Lehrmittel: Skript



Arten von Aufgaben im Skript

- 1.1

  → Prüfungsrelevant

- Challenge: 1.1

  → Weiterführende Aufgaben

Lehrmittel: Skript



Arten von Aufgaben im Skript

- 1.1

  → Prüfungsrelevant

- Challenge: 1.1

  → Weiterführende Aufgaben

Lehrmittel: Skript



Arten von Aufgaben im Skript

- 1.1

  → Prüfungsrelevant

- Challenge: 1.1

  → Weiterführende Aufgaben

Lehrmittel: Skript



Arten von Aufgaben im Skript

- 1.1

  → Prüfungsrelevant

- Challenge: 1.1

  → Weiterführende Aufgaben

Was ist Programmieren?

- 💡 Seine Gedanken logisch ausdrücken

Was ist Programmieren?

- 💡 Seine Gedanken logisch ausdrücken
  „The truth is you don't really understand something until you've taught it to a computer, until you've been able to program it.“ — Donald Knuth, PhD

Was ist Programmieren?

- 💡 Seine Gedanken logisch ausdrücken
  „The truth is you don't really understand something until you've taught it to a computer, until you've been able to program it.“ — Donald Knuth, PhD
- 👉 Dem Computer **Befehle** / Aufträge geben

Was ist Programmieren?

- 💡 Seine Gedanken logisch ausdrücken
  „The truth is you don't really understand something until you've taught it to a computer, until you've been able to program it.“ — Donald Knuth, PhD
- 👉 Dem Computer **Befehle** / Aufträge geben
- 🦦🏝️ Automatisieren & das Leben vereinfachen

Programmieren: Demo

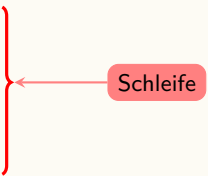
Programmieren: Erste **Befehle**

Befehl	**Abkürzung**	**Bedeutung**
`import turtle as t`		Alle `turtle`-Funktionen "laden
`t.forward(100)`	`t.fd(100)`	Gehe 100 Pixel nach vorne
`t.left(90)`	`t.lt(90)`	Um 90 Grad nach links drehen
`t.right(90)`	`t.rt(90)`	Um 90 Grad nach rechts drehen
`t.color("green")`		Stiftfarbe auf grün setzen
`t.width(2)`		Stift-Dicke auf 2 setzen

Allgemein: `befehlsname(p1, p2, ...)`

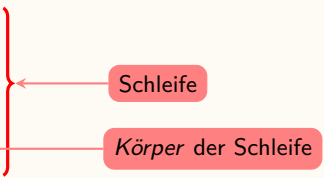
Programmieren: Erste **Schleifen**

```
for _ in range(15):
    t.fd(15)
    t.lt(90)
    t.fd(25)
    t.rt(90)
```



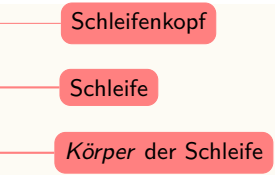
Programmieren: Erste **Schleifen**

```
for _ in range(15):
    t.fd(15)
    t.lt(90)
    t.fd(25)
    t.rt(90)
```



Programmieren: Erste **Schleifen**

```
for _ in range(15):
    t.fd(15)
    t.lt(90)
    t.fd(25)
    t.rt(90)
```



Zusammenfassung

- Mit **Befehlen** können wir den Computer eine abstrakte Tätigkeit ausführen lassen

Zusammenfassung

- Mit **Befehlen** können wir den Computer eine abstrakte Tätigkeit ausführen lassen
- **Parameter** dienen dazu, die Befehle *parametrisieren* (Analogie: Kochrezept für 2 oder 4 Personen)

Zusammenfassung

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

Zusammenfassung

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

Auftrag

Skript Programmieren

- Kapitel 1 (Installation Python und VS Code)
- Kapitel 2 (Einführung in Python), Aufgaben 2.1-2.5
- Challenge: Moodle, Schachbrett-Aufgabe

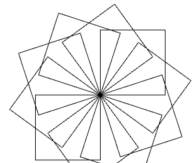
Schleifen in Schleifen

blume.py

```
import turtle as t

for _ in range(10):
    # zeichne ein Quadrat
    for _ in range(4):
        t.fd(100)
        t.rt(360 / 4)
    # Rechtsdrehung um 36 Grad
    t.rt(360 / 10)

# Turtle-Zeichnung nicht schliessen
t.done()
```



Auftrag

Skript

- **Kapitel 1**: Installation von Python
- **Kapitel 2**: Einführung in Python, Aufgaben 2.1-2.5
- Challenge: Weitere Aufgaben im Kapitel 2

Achtung

Immer zuerst Beispiele und Text lesen, danach Aufgaben lösen!

Datentypen

Beispiel	**Datentyp (englisch, Abk.)**	**Datentyp (Deutsch)**
`"Hallo Welt", "abc", "42"`	string (str)	Zeichenkette
`15, 45, -5, 0`	integer (int)	Ganzzahl
`12.23, -5.33, 23.0`	float (float)	Kommazahl

Analogie:

- Name (string)
- Alter in Jahren (integer)
- Grösse in Metern (float)

Auftrag

Skript

- Kapitel 3
- Tipp: Backslash (\)
  - auf MacOS: ⇧ + ⌥ + 7
  - auf Windows: Alt Gr + \