Programming Basics

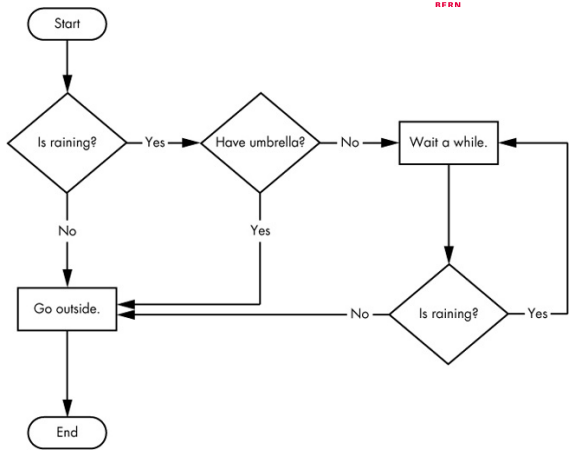
What you will/won’t learn

- Version control (Git and Github)
- Documenting with “Separation of Concerns” (Markdown)
- Structuring and Describing Problems (UML)
- Programming Basics (Python)

- Hacking your neighbour’s bank account
- Programming and solving complex math problems
- Programming an ego-shooter
- Solving the complete data analysis of project X
- ... but after the course, you should have all ingredients for it

Programming Basics

- “Computational Thinking” → Übersetzen von Ideen zu Algorithmen
- Algorithmen implementieren
  - mit graphischen Hilfsmitteln
  - mit der Programmiersprache Python
- Workflow beim Programmieren
  - Versionierte Projekte git
  - iterative Umsetzung von Algorithmen
  - Arbeiten im Team
- Modulares, strukturiertes Design von Programmen




```python
def left():
    for _i in range(3):
        right()

def forward_n(steps):
    for _i in range(steps):
        forward()

def downstairs(steps):
    for _i in range(steps):
        step_down()

forward()
left()
forward_n(6)
right()
forward_n(2)
right()
downstairs(3)
found()
```

Programming Basics - Resources

- Kein ILIAS
- Github Repositories
- „Microsoft Learn“ learning paths

Heute

- Einstieg
  - Video
  - Partnerarbeit
  - Erstes Programm

PAUSE

- Computational Thinking
  - Strukturieren von Algorithmen
  - Übung

PAUSE

- Erkenntnisse festhalten
  - Im Code
  - Zusätzliche Dokumentation
  - Übung

PAUSE

- Arbeiten mit Python

TEDx Talk Programming

https://www.youtube.com/watch?v=xfBWk4nw440

Zu zweit...

Anleitung zum Auffinden eines Schöggelis anfertigen. Es dürfen auschliesslich folgende Anweisungen verwendet werden:

```
forward()    # 1 Schritt vorwärts
right()      # 90° Drehung
step_up()    # Treppenstufe hoch
step_down()  # Treppenstufe runter
found()      # Schoggi gefunden
```

Zeit: 5’

Anleitung tauschen und suchen gehen: 5’

Algorithmus

Ein **Algorithmus** ist eine eindeutige **Handlungsvorschrift** zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus **endlich vielen, wohldefinierten Einzelschritten**. Damit können sie zur Ausführung in ein Computerprogramm implementiert, aber auch in menschlicher Sprache formuliert werden.

Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.

https://de.wikipedia.org/wiki/Algorithmus

```
forward()
right()
right()
right()
forward()
forward()
forward()
forward()
forward()
forward()
right()
forward()
forward()
right()
step_down()
step_down()
step_down()
found()
```

Programm

Ein **Computerprogramm** oder kurz **Programm** ist eine den Regeln einer bestimmten Programmiersprache genügende **Folge von Anweisungen** (bestehend aus Deklarationen und Instruktionen), um bestimmte Funktionen bzw. Aufgaben oder Probleme mithilfe eines Computers zu bearbeiten oder zu lösen.

https://de.wikipedia.org/wiki/Computerprogramm

```
forward()
right()
right()
right()
forward()
forward()
forward()
forward()
forward()
forward()
right()
forward()
forward()
right()
step_down()
step_down()
step_down()
found()
```

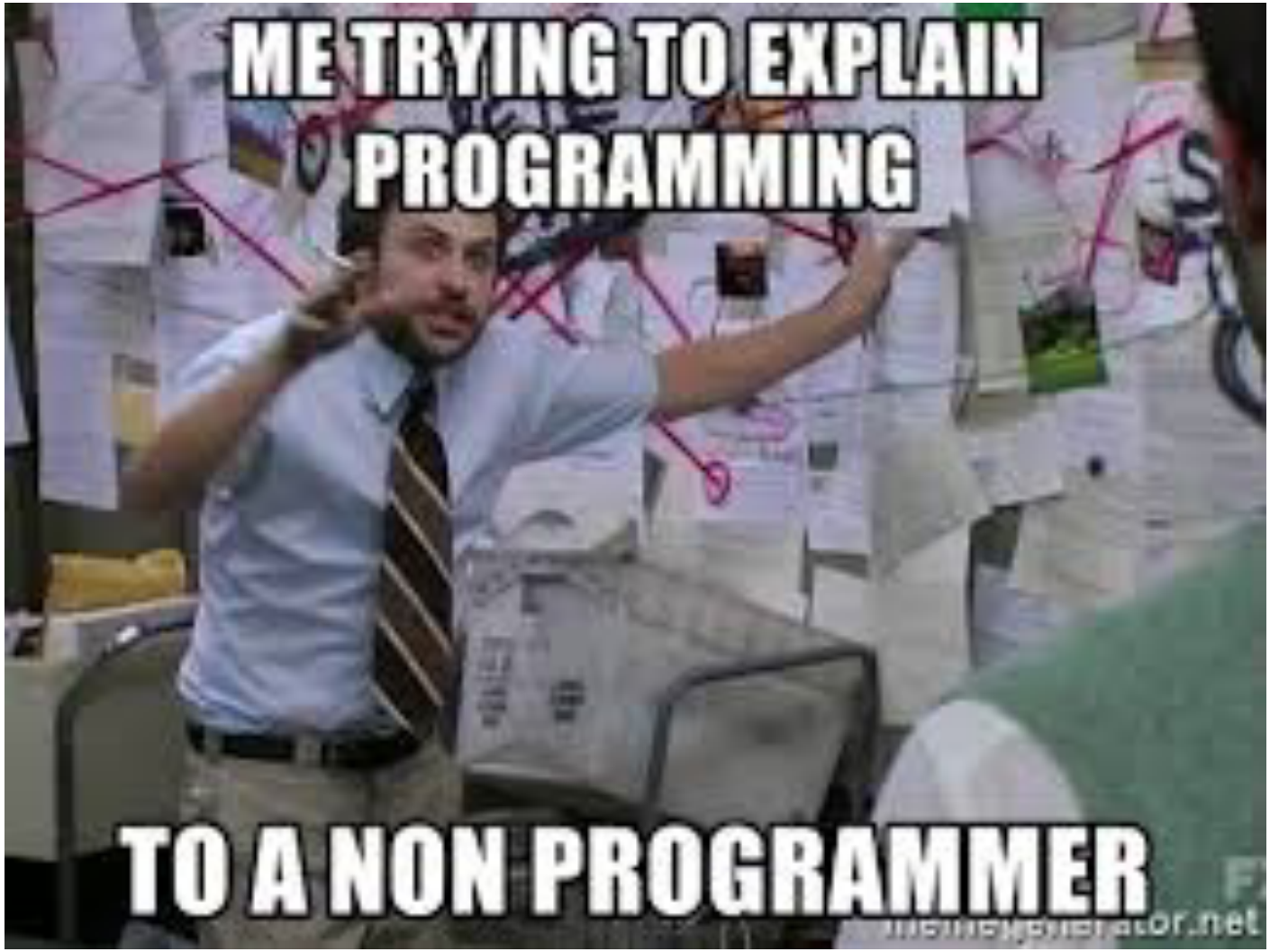
In 2er Gruppen, nur 1. Person darf schauen…

… und anschliessend dem Gegenüber nur mit Worten eine Anleitung zum zeichnen der Figur geben (3’):

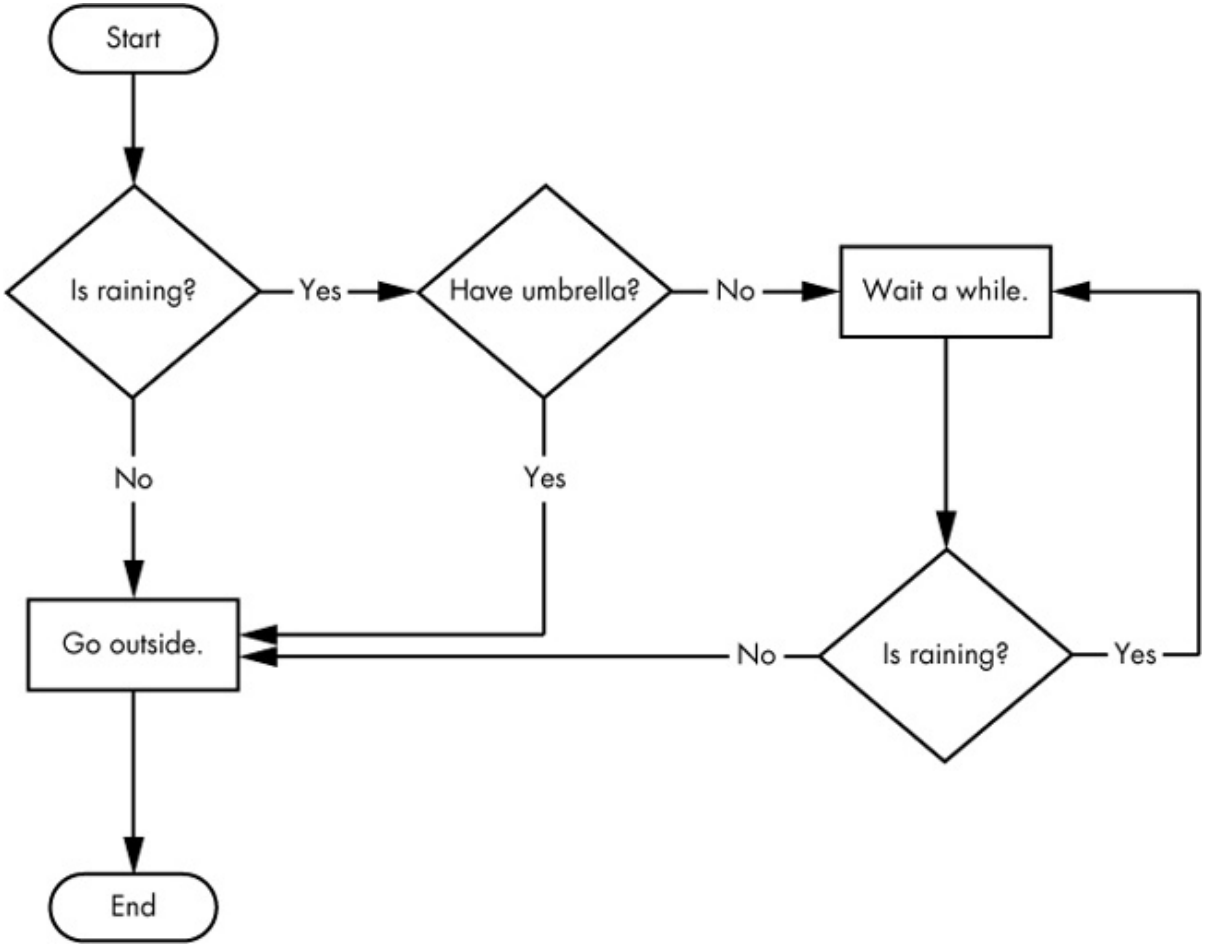
Folgende Begriffe dürfen NICHT verwendet werden:

- **Dreieck**
- **Diagonale**

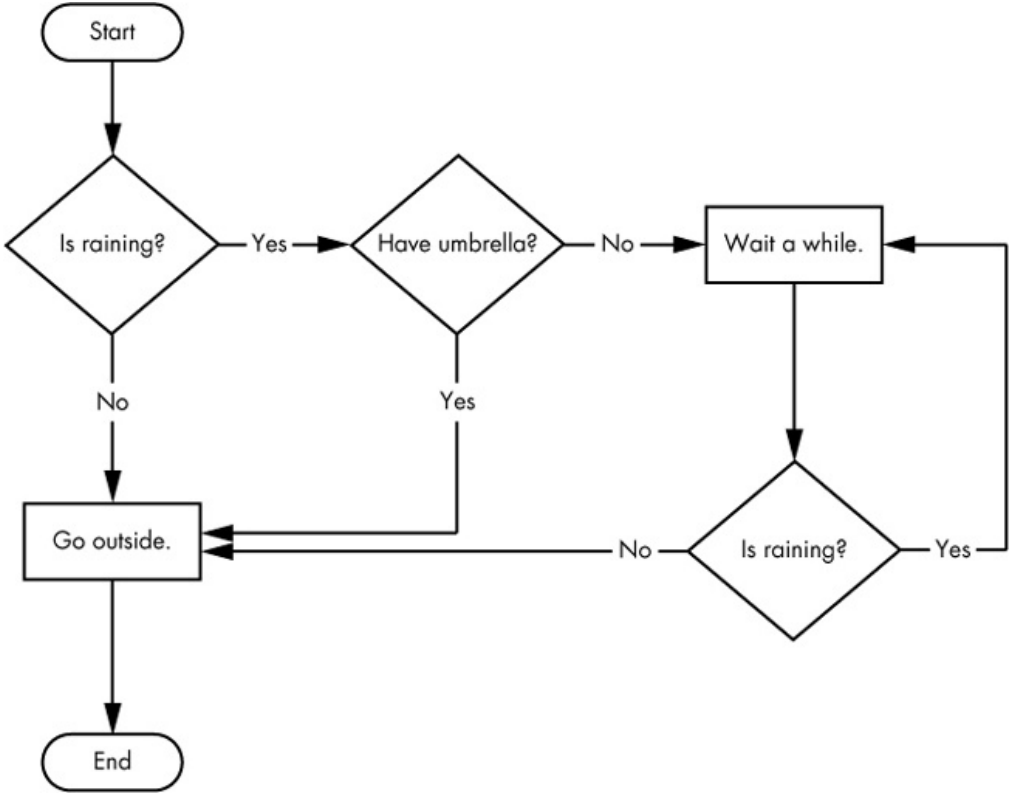
```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
```

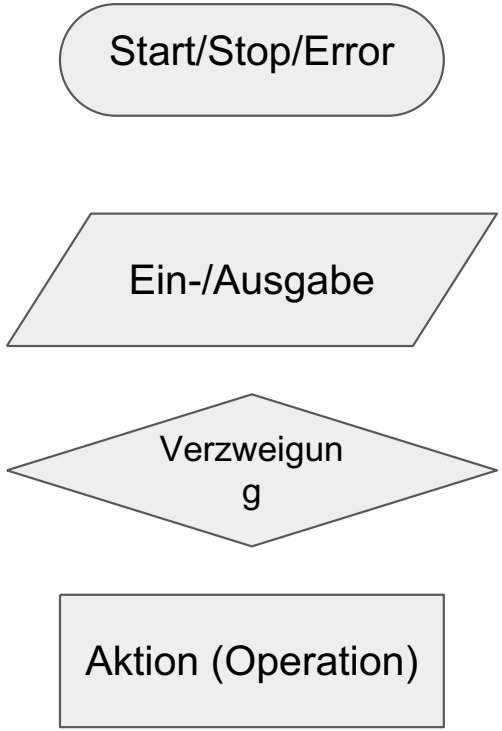

ME TRYING TO EXPLAIN
PROGRAMMING
TO A NON PROGRAMMER

Beispiel

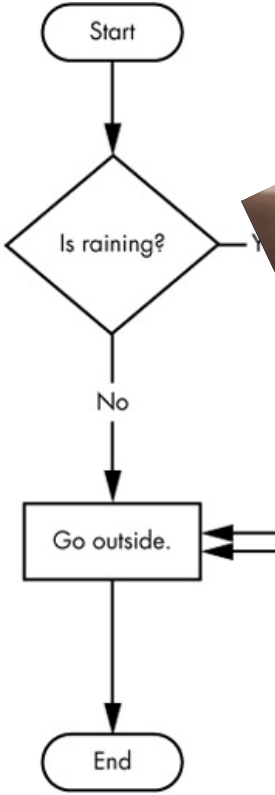

Start
Is raining?
Yes
Have umbrella?
No
Wait a while.
No
Yes
Go outside.
No
Is raining?
Yes
End

Programmablaufplan - UML





Programmablaufplan - Li



Start/Stop/Error

-/Ausgabe

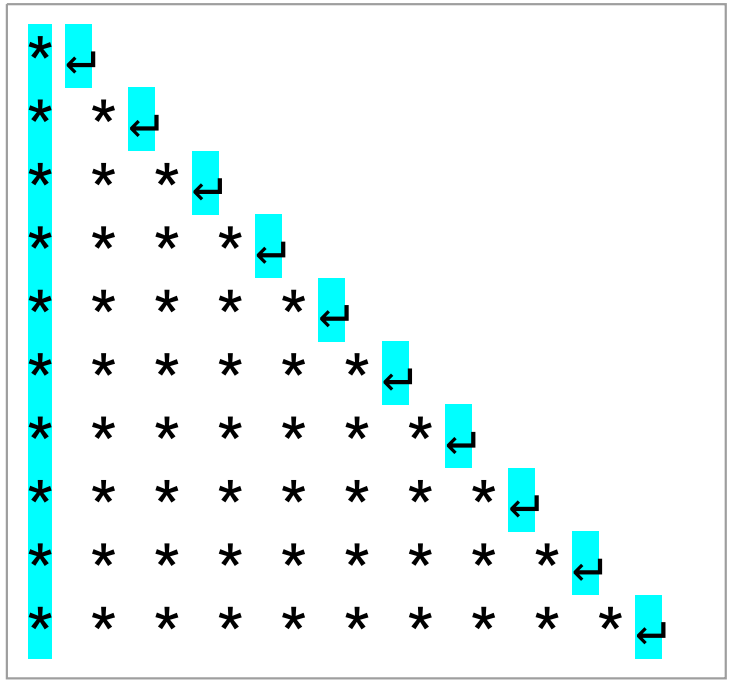
Verzweigun
g

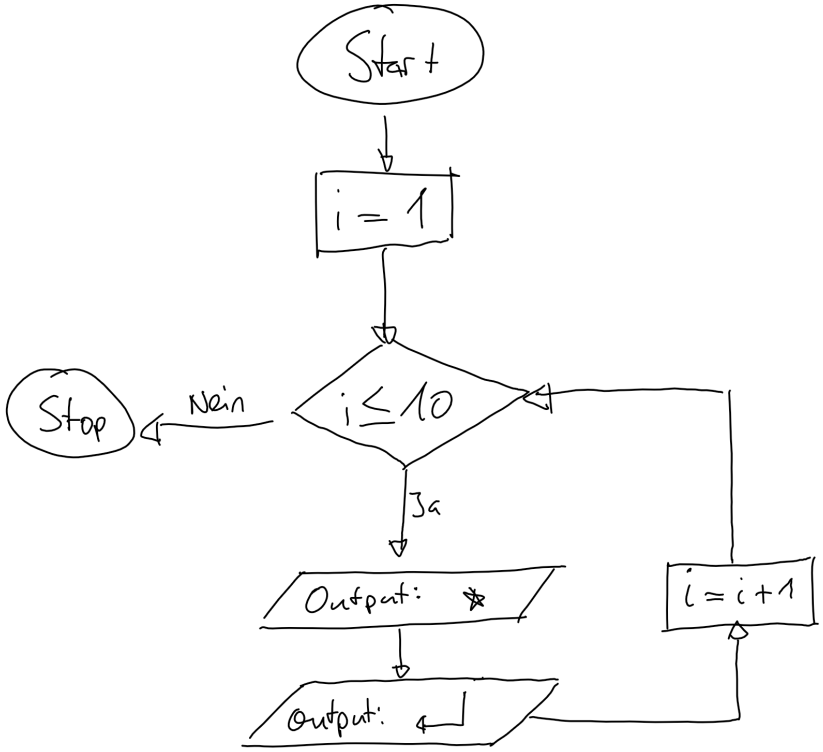
Aktion (Operation)

Formalisierung eines Problems mit UML

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
```

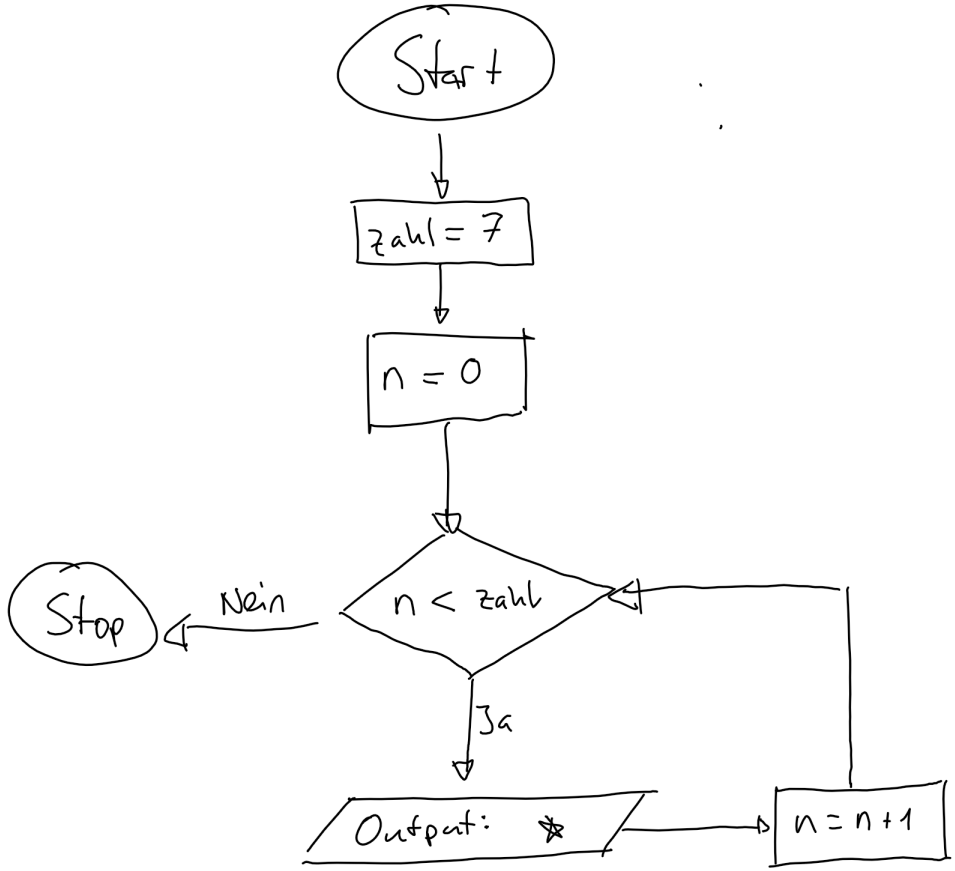
Schritt 1: 1 Stern & Neue Zeile ausgeben



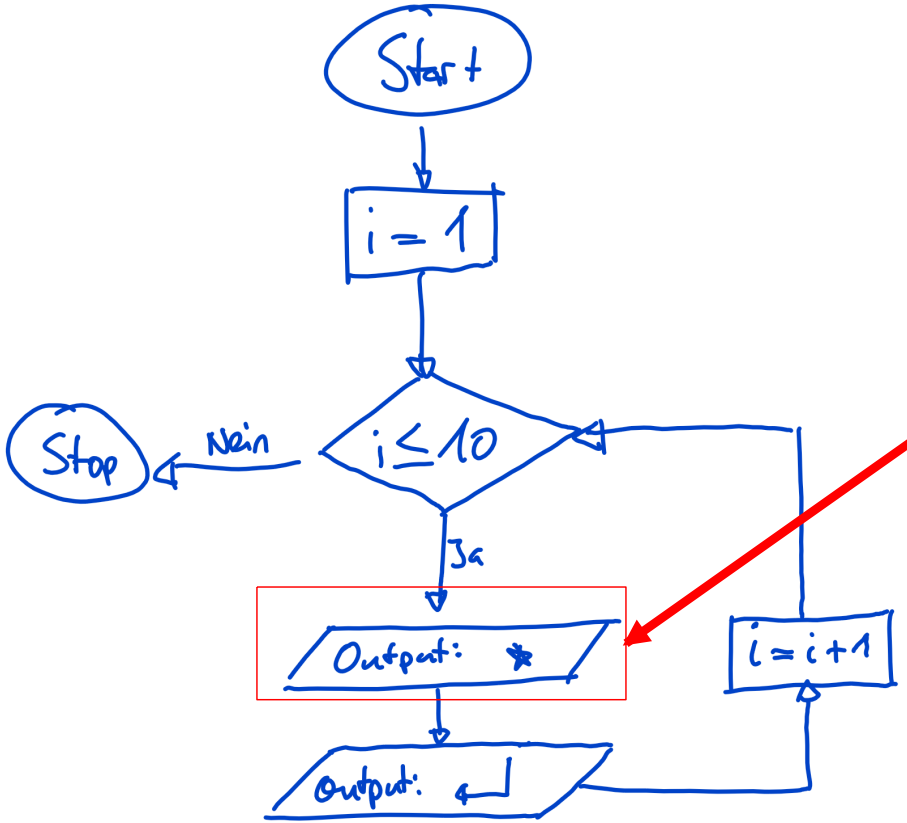


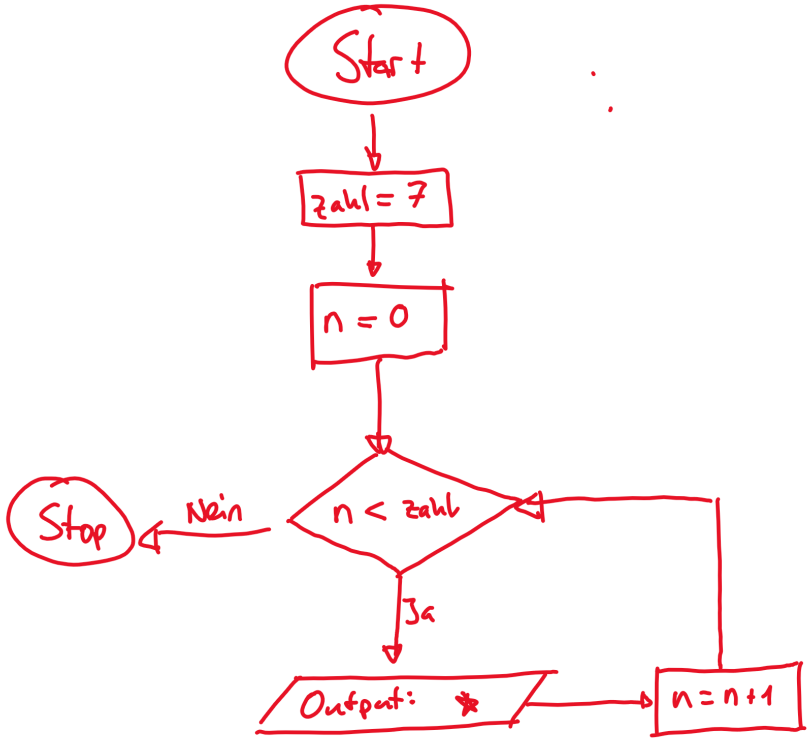
Schritt 2: Für eine gegebene Zahl ebenso viele Sterne ausgeben

```
zahl
 1 ?  *
 2 ?    *  *
 3 ?    *  *  *
 4 ?    *  *  *  *
 5 ?    *  *  *  *  *
 6 ?    *  *  *  *  *  *
 7 ?    *  *  *  *  *  *  *
 8 ?    *  *  *  *  *  *  *  *
 9 ?    *  *  *  *  *  *  *  *  *
10 ?    *  *  *  *  *  *  *  *  *  *
```



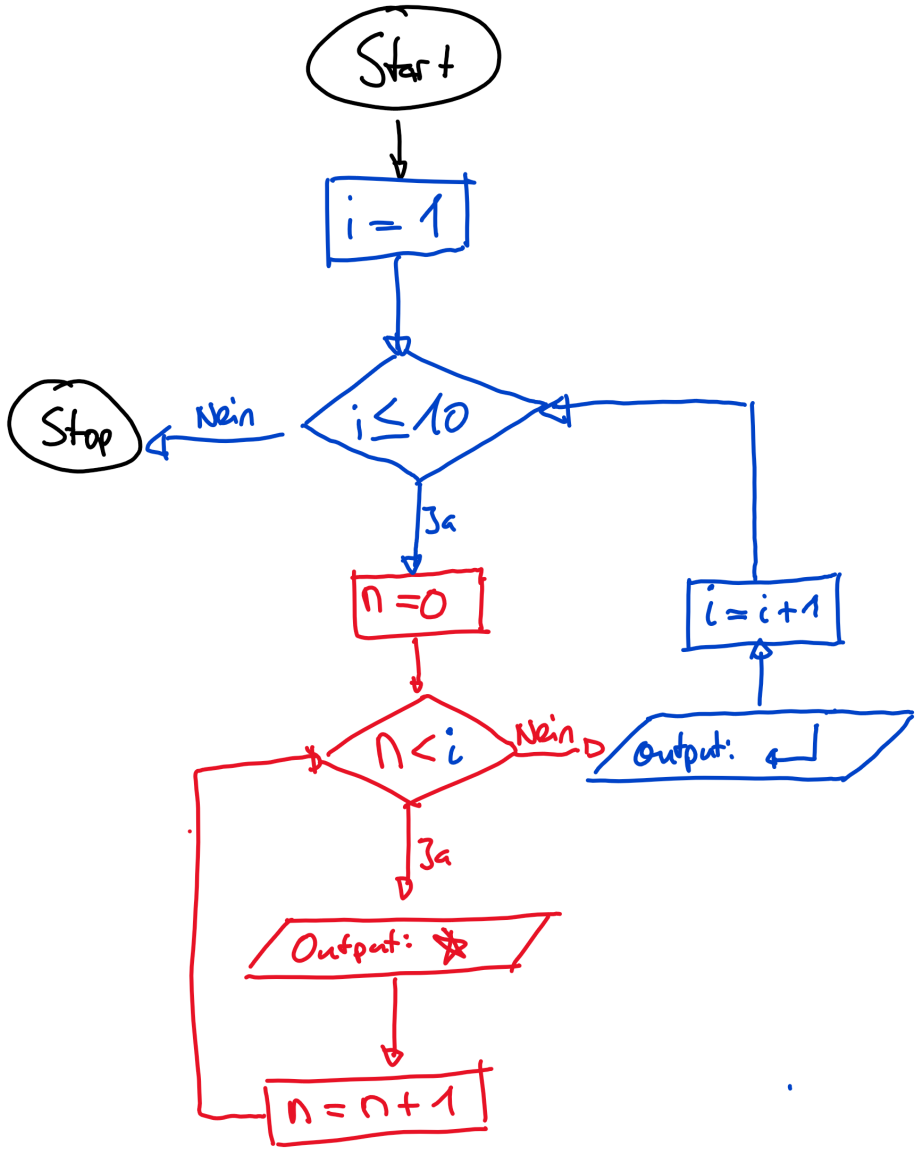
Schritt 3: Zusammensetzen





Schritt 3: Zusammensetzen

```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *  *  *
*  *  *  *  *  *  *
*  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
```



Passe den Programmablauf so an, dass die folgende Ausgabe erzeugt wird

```
*
   *
      *
         *
            *
               *
                  *
                     *
                        *
                           *
```

Hilfestellung

- Identifiziere zuerst das neue Teilproblem und entwirf dafür das UML

```
zahl
 1 [?] *
 2 [?]        *
 3 [?]         *
 4 [?]                        *
 5 [?]              *
 6 [?]                 *
 7 [?]                    *
 8 [?]                       *
 9 [?]                          *
10 [?]                            *
```

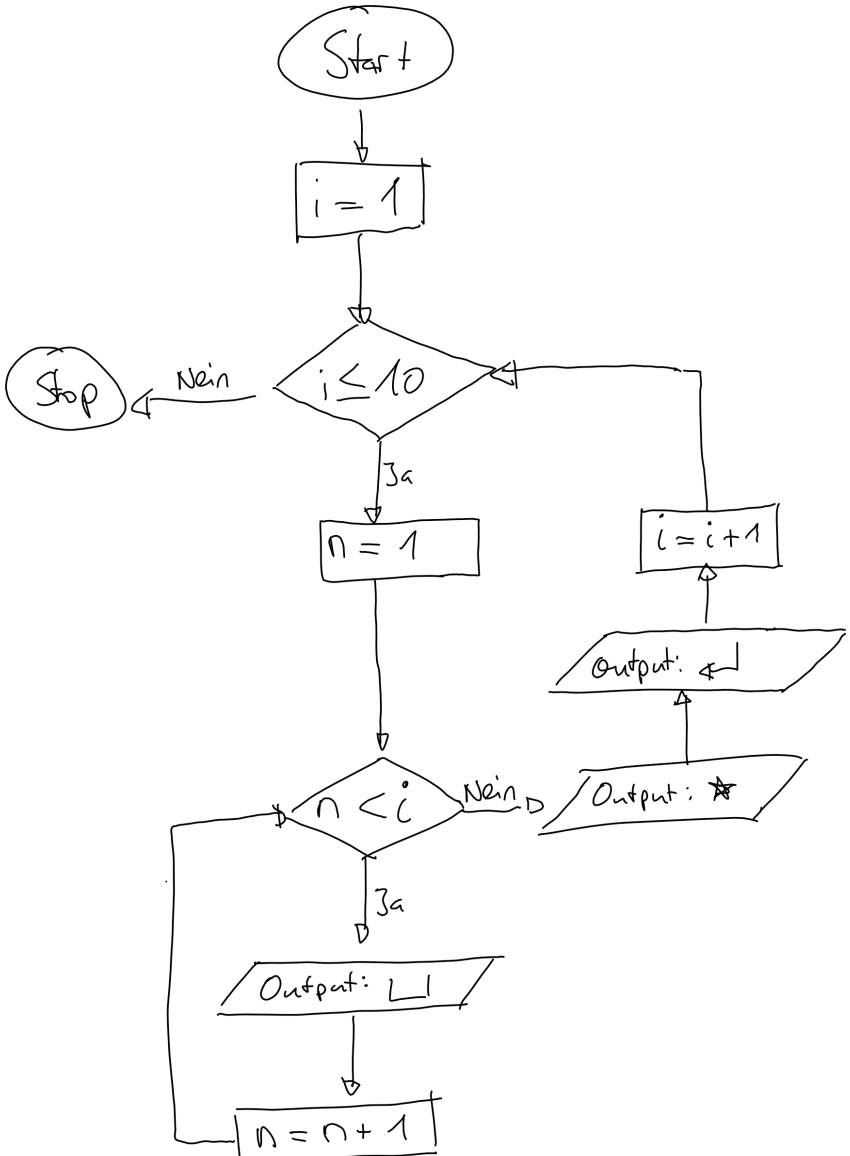
Passe den Programmablauf so an, dass die folgende Ausgabe erzeugt wird

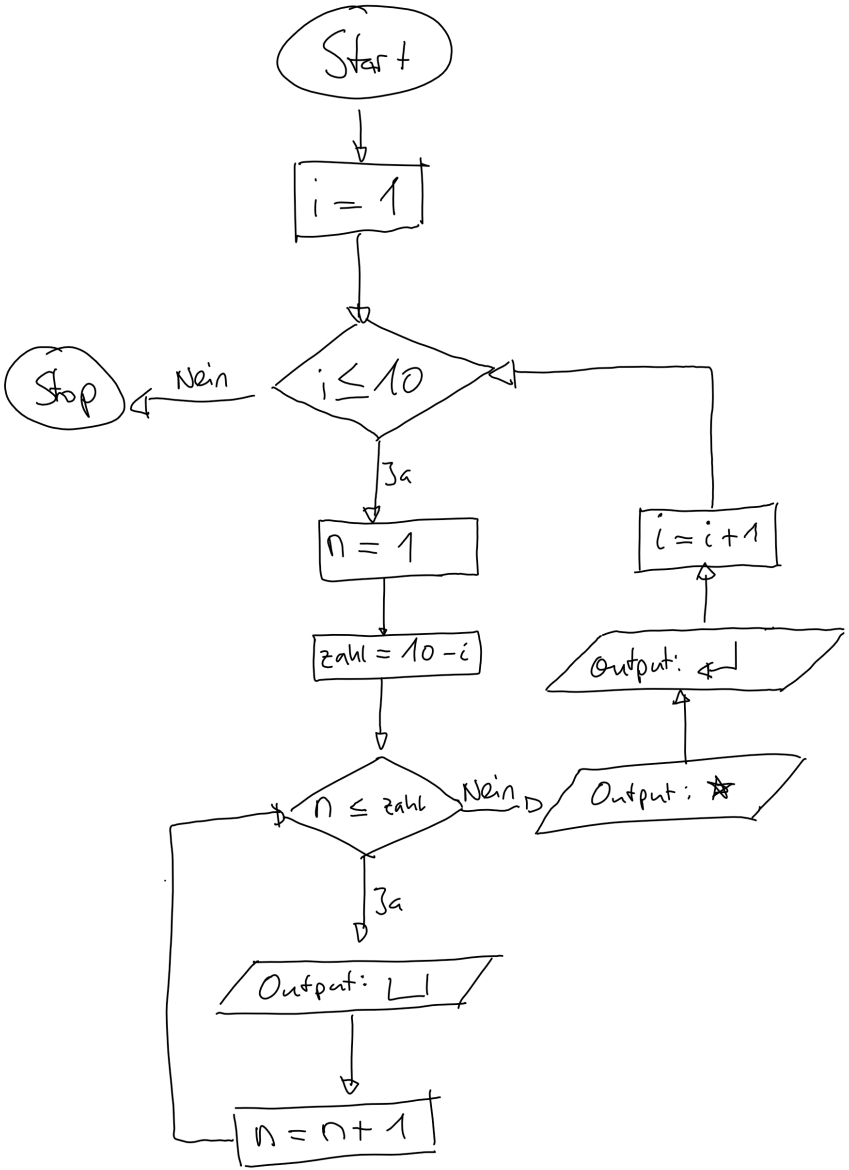
```
                    *
                  *
                *
              *
            *
          *
        *
      *
    *
  *
```

Hilfestellung

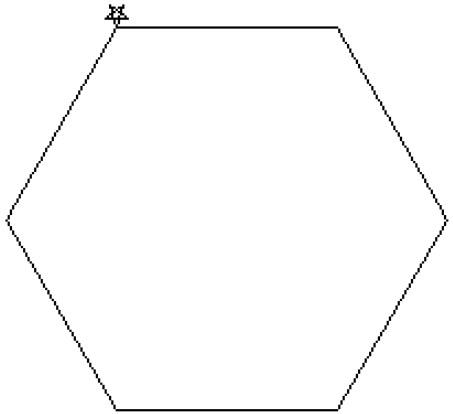
- Identifiziere zuerst das neue Teilproblem und entwirf dafür das UML

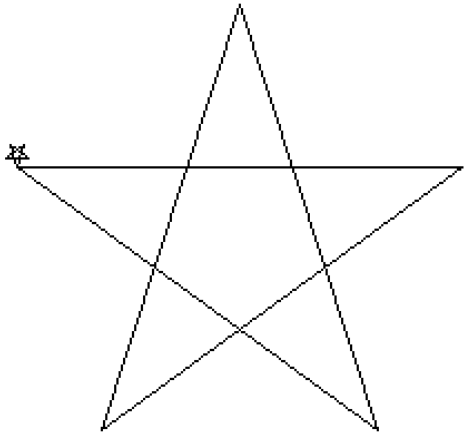
```
zahl
 1 ?                    *
 2 ?                   *
 3 ?                *
 4 ?                  *
 5 ?             *
 6 ?          *
 7 ?       *
 8 ?    *
 9 ?  *
10 ? *
```

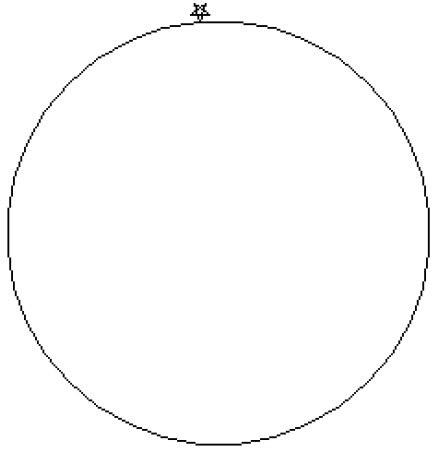

Start
i = 1
i ≤ 10
Nein
Stop
Ja
n = 1
n < i
Nein
Ja
Output: ↵
n = n + 1
Output: ★
Output: ↵
i = i + 1


Start
i = 1
i ≤ 10
Nein
Stop
Ja
n = 1
zahl = 10 − i
n ≤ zahl
Nein
Ja
Output: ↵
n = n + 1
Output: ★
Output: ↵
i = i + 1

Zeichne ein UML für eines der folgenden Objekte





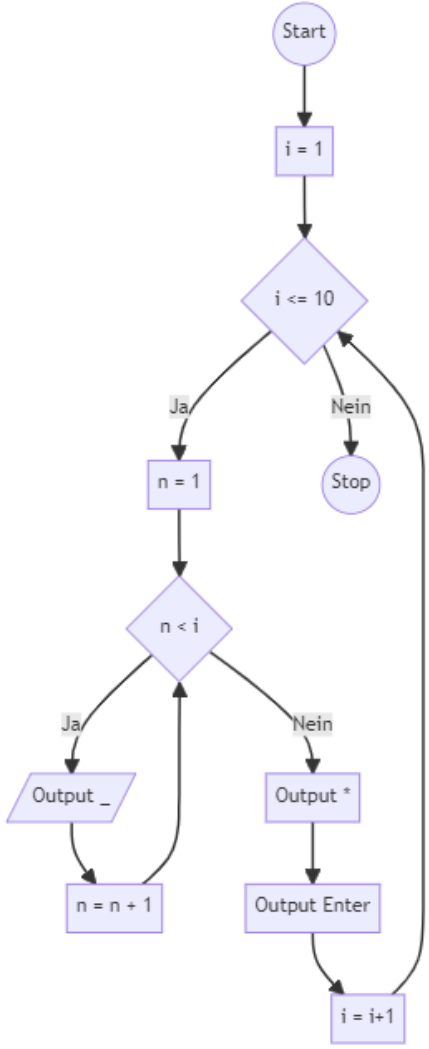


Mermaid.live

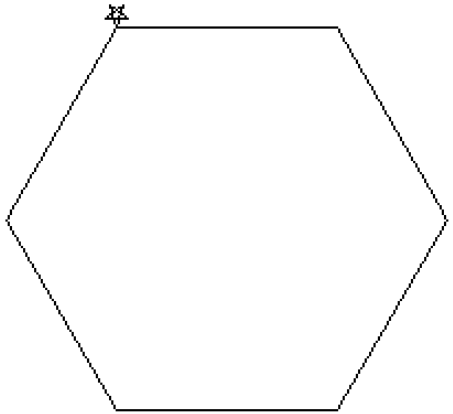
- https://mermaid.live/

```
flowchart TD
    A((Start)) --> B[i = 1]
    B --> C{i <= 10}
    C -->|Ja| D[n = 1]
    C -->|Nein| E((Stop))
    D --> F{n < i}
    F-->|Ja| G[/Output _/]
    G --> J[n = n + 1] --> F
    F -->|Nein| H[Output *] --> I[Output
Enter] --> K[i = i+1] -->C
```

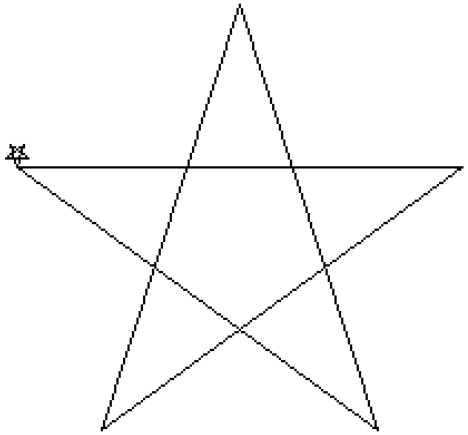
- https://mermaid.js.org/syntax/flowchart.html



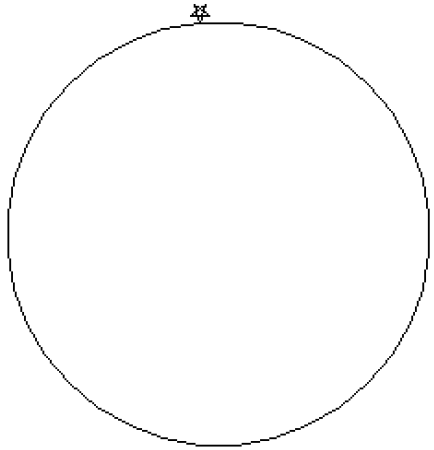
…setze das UML in Python um



hexagon.py



star.py



circle.py

`forward(10)` und `right(90)` können mit beliebigen Zahlen aufgerufen werden.

Pause

VS Code

- Fürs Schreiben von Textdokumenten wird oft Word verwendet…
- Fürs Schreiben von Programmiercode gibt es eine Vielzahl an geeigneten Applikationen, sogenannte Editoren.
  Wir verwenden **VS Code**, weil
    - Open Source (daher kostenlose Nutzung)
    - Aktuell eines der beliebtesten Tools zum Schreiben von Code
    - Viele Erweiterungen u.a. für Python (und auch für R…)
    - Benötigt „wenig“ Speicherplatz

https://www.software.com/src/ranking-the-top-5-code-editors-2019

VS Code

Code File Edit Selection View Go Run Terminal Window Help
100% Mo. 12:40 Balthasar Hofer
Welcome
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.
Clone Repository
OUTLINE
NPM SCRIPTS
LOCAL HISTORY
Welcome
Visual Studio Code
Editing evolved
Start
New file
Open folder...
Add workspace folder...
Recent
intro ~/Downloads/pylab-master-intro
pylab ~/Documents/git_code/gitlab/progr...
pylab-tutors ~/Documents/git_code/gitla...
ofi-24p ~/Documents/git_code/lebalz
ofi-24o ~/Documents/git_code/lebalz
More... (^R)
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Join our Newsletter
Customize
Tools and languages
Install support for JavaScript, Python, ...
Settings and keybindings
Install the settings and keyboard short...
Color theme
Make the editor and your code look the...
Learn
Find and run all commands
Rapidly access and search commands ...
Interface overview
Get a visual overlay highlighting the ma...
Interactive playground
Try out essential editor features in a sh...
Show welcome page on startup
pylab-master-intro
Favoriten
AirDrop
Zuletzt benutzt
Programme
Downloads
Bilder
Dokumente
git_code
Musik
Filme
Schreibtisch
OneDrive - Erziehungsdir...
Erziehungsdirektion des...
OneDrive
iCloud
Name
intro
Heute, 12:15
Macintosh HD

VS Code & Python Installation

- https://learn.microsoft.com/de-de/training/paths/beginner-python/

- Login mit Student Account
- Kapitel 1 bearbeiten

Pause

Python Grundlagen / Arbeitsaufträge

(Nach-)Arbeiten im Lernpfad und (mit Markdown) Erkenntnisse festhalten.
https://learn.microsoft.com/de-de/training/paths/beginner-python/

- Kapitel 1 + 2 (gerne auch mehr, wenn ihr Lust habt)

Arbeiten im Buch (Kapitel 1 und 2) und (mit Markdown) Erkenntnisse festhalten.
https://ssr-2024.github.io/programming_basics

- Variablen
- Datentypen
- Operatoren
- Fehler