

# WOCHE 5

## DATENSTRUKTUREN I

### STRINGS & LISTEN



# LERNZIELE

- Inhalte der Wochen 1-4 vertiefen
- Sinn und Zweck von Datenstrukturen verstehen
- Wichtige Begriffe:



Strings



Listen



Slicing

# WIEDERHOLUNG WOCHE 1-4

# print()-FUNKTION

- Dinge ausgeben

```
print("Hello World!")
```

# input()-FUNKTION

- Benutzereingaben entgegennehmen
- Eingabe immer **str**





```
name = input("Wie lautet dein Name?")  
print(name)
```

# VARIABLEN

- Informationen unter einem Namen (Bezeichner) speichern
- **Name = Wert**
- Sinnvoll benennen!

```
vorname = "Jan"  
alter = 17
```

# DATENTYPEN

- Daten besitzen einen Typ -> Datentyp
- **type()** gibt Datentyp zurück
- 4 Grunddatentypen:
  -  Strings(**str**)
  -  Integer(**int**)
  -  Float(**float**)
  -  Boolean(**bool**)

# ÜBERSICHT: DATENTYPEN

Strings      Zeichenketten

`"Hallo Welt"`

`'Gustav Gans'`

`"""Text`

`mit mehreren  
Zeilen"""`

---

Float      Fließkommazahlen

`5.14`

---

Integer      Ganzzahlen

`5, 99, -23`

---

Boolean      Wahrheitswerte

`True, False`



# BEDINGUNGEN

- "Wenn X wahr ist, dann mach ..."
- **if** / **elif** / **else**

```
hour = 14
if hour < 12:
    print("guten Morgen")
elif hour < 17:
    print("guten Tag")
elif hour < 20:
    print("guten Abend")
else:
    print("gute Nacht")
```

# SCHLEIFEN

- 2 Schleifen - **while** / **for**-Schleife
- **while** = "Solange <Bedingung> .."
- **for** = "Für jede/s/n .."

# while-SCHLEIFE

```
i = 0
while i < 10:
    print("Hello World")
    i = i+1
```

# for-SCHLEIFE

```
for i in range(10):  
    print("Hello World!")
```

# DATENSTRUKTUREN

- Datenstruktur -> Daten strukturiert speichern
- 4 Datenstrukturen in Python:



Strings



Listen



Tupel



Dictionaries

# MOTIVATION:

# DATENSTRUKTUREN

- Bisher: Jede Information -> 1 Variable
- Vokabeltrainer: 1 Vokabel = 1 Variable?
- Beispiele aus dem Alltag:
  - Schaubilder / Mindmaps
  - Tabellen
  - Ordner

# STRINGS

# STRINGS

- "Liste" von darstellbaren Zeichen ~ Text
- Beispiele:
  - `"xy3c!"`
  - `"Das ist ein Satz!"`
  - `"\n\r\tHallo\n\r\t"`



# STRINGS ANLEGEN

## EINZEILIG MIT EINFACHEN ' '

```
a = 'Lorem ipsum dolor sit amet, consectetur.'
```

## EINZEILIG MIT " "

```
b = "Lorem ipsum dolor sit amet, consectetur."
```

## MEHRZEILIG MIT "" "" "" ""

```
c = """Lorem ipsum dolor sit amet, consectetur  
adipisicing elit. Sunt molestiae corrupti ipsa  
placeat sit ea maiores quam repellat corporis  
asperiores expedita vero rem eveniet  
impedit nulla adipisci, nihil quod, quas?"""
```

# STRING-FUNKTIONEN

- Strings bringen Vielzahl an Funktionen mit
- Strings manipulieren/bearbeiten

`str.replace()`   `str.format()`

`str.upper()`   `str.lower()`

- Strings auf Eigenschaften prüfen

`isX()`   `startswith()`   `endswith()`

- (Teil)strings suchen/finden

[Python Dokumentation \(3.7.1\): String-Methoden](#)

[Python Dokumentation \(3.7.1\): String-Operationen](#)

//CODING.SCHULE

# SLICING

- Teile aus Datenstruktur entnehmen
  - `[start:stop:step]`
- `start` inklusive, `stop` exklusive
- Jedes Zeichen hat einen eindeutigen *Index*

# INDEX

- Index = Stelle des jeweiligen Zeichens
- beginnt bei **0**

-6	-5	-4	-3	-2	-1
<hr/>					
0	1	2	3	4	5
<hr/>					
P	Y	T	H	O	N



# LIVE-CODING

# AUFGABENBLATT 1



# LISTEN

```
fruits = ["apple", "banana", "mango", "ananas"]
```

# DEFINITION: LISTEN

- ermöglicht Listen von Daten zu speichern
- Kann Daten unterschiedlichen Typs enthalten
- `my_list = ["anna", 5.0, 3, True]`
- Zugriff auf Elemente über Indizes
  - `my_list[0] # "anna"`



# UMGANG MIT LISTEN

## LISTEN ANLEGEN

```
names = ["Anna", "Anton", "Judith"]
```

## ELEMENTE HINZUFÜGEN

```
names.append("Peter")  
names.extend(["Peter", "Ingo", "Andreas"])
```

## ELEMENTE LÖSCHEN

```
del names[2]
```

## LISTEN SORTIERTEN

```
names.sort()
```

# LISTENFUNKTIONEN

[Dokumentation: Listen-Funktionen](#)

# DATENSTRUKTUREN & SCHLEIFEN

- Datenstrukturen lassen sich mit **for**-Schleife *iterieren*
- *Iteration* = Elemente schrittweise verarbeiten
- Strings = Zeichen für Zeichen
- Listen = Element für Element

# BEISPIELE: ITERATION

```
satz = "Lorem ipsum dolor sit amet"  
for buchstabe in satz:  
    print(buchstabe)
```

```
fruits = ["Ananas", "Mango", "Kiwi", "Apple"]  
for fruit in fruits:  
    if fruit.startswith("A"):  
        print(fruit)
```



# LIVE-CODING

# AUFGABENBLATT 2



**VIELEN DANK!**