

Block 3: Datentypen

IPython-REPL II

- Windows-Startmenu aufrufen
- unter 'A': Anaconda auswählen
- IPython anklicken

Navigieren im Dateiensystem

```
In [1]: pwd
Out[1]: 'C:/Users/Martin/Documents'
In [2]: cd ..
C:/Users/Martin
In [3]: cd Documents
C:/Users/Martin/Documents
```

Magische Befehle

```
%magic
%cpaste
```

Zeichenketten / Strings

```
print("Hallo Welt!")
```

Konkatenieren

```
print("Hallo " + "Welt!")
print("Historikertag " + "2018")
print("Hallo " * 2 + "Historikertag")
```

Variablen

```
event = "Historikertag"
year = 2018
```

Interpolieren

```
print(f"Der {event} findet {year} statt.")  
  
print("Der {} findet {} statt.".format(event, year))
```

Funktionen und Methoden bei Strings

```
len(event)  
  
len("Historikertag")  
  
event.lower()  
  
"Historikertag".upper()
```

Listen / Lists

```
list_1 = [1, 2, 3, 4]  
  
list_2 = ["Banane", "Apfel", "Birne"]  
  
list_3 = [17, "Obst", 23, "Ball"]  
  
list_4 = [[1, 2, 3,], ["a", "b", "c"], ["Hallo", "Welt"]]
```

Zuordnungen / Dictionaries

```
dict_1 = {"Banane": 5, "Apfel": 7, "Birne": 9}  
  
dict_2 = {"Vorname": "Jan", "Nachname": "Meier", "Alter": 48}
```

Mengen / Sets

```
menge = {1,2,3,4,4,5,5,5}  
  
print(menge)  
  
type(menge)
```

Tuple

```
word_freq = ("Wort A", 42)

type(word_freq)
```

Indexieren von Strings

```
"Historikertag" [1]
```

```
"Historikertag" [0]
```

```
"Historikertag" [8]
```

Slicing

```
"Historikertag" [0:10]
```

```
"Historikertag" [3:6]
```

Indexieren von Listen

```
list_2 = ["Banane", "Apfel", "Birne"]
```

```
list_2[2]
```

```
list_2[0]
```

Funktionen und Methoden bei Listen

```
len(list_2)
```

```
list_2.append("Ananas")
```

```
list_2.pop()
```

Zugriff auf Dictionaries

```
dict_1 = {"Banane": 5, "Apfel": 7, "Birne": 9}
```

```
dict_1["Banane"]
```

```
dict_1.keys()

dict_1.values()

sum(dict_1.values())
```

Logische Ausdrücke

```
5 > 3

5 > 7

"Mauer" == "Haus"

"Mauer" != "Haus"
```

Logische Ausdrücke: and, or

```
5 > 3 and "Mauer" != "Haus"

5 > 3 and "Mauer" == "Haus"

5 > 3 or "Mauer" == "Haus"

5 > 7 or "Mauer" == "Haus"
```

Block 4: kleine Skripte

VS Code einrichten

- Windows-Startmenu aufrufen
- unter 'A': Anaconda auswählen
- Anaconda Navigator anklicken
- VS Code installieren

VS Code starten

- Windows-Startmenu aufrufen
- unter 'A': Anaconda auswählen
- Anaconda-Prompt anklicken

- ‘code’ eingeben

Python Extension einbinden

- ganz links viertes Symbol von oben klicken
- im Suchfeld ‘Python’ eingeben
- ersten Treffer auswählen und installieren
- Code neu starten

Python-Datei anlegen

- speichern unter: Dateiname eingeben
- Dateiendung: .py anhängen

User-Input Skript

```
# Eingabe Name
name = input("Wie heißt du? >>>")
# Eingabe Alter
alter = input("Wie alt bist du? >>>")
# Eingabe Wohnort
ort = input("Wo wohnst du? >>>")
jahr = 2018 - int(alter)
# Ausgabe
print(f"""
\nHallo {name}, schön, dass du da bist!\n
Du bist {jahr} geboren.\n
{ort} ist der beste Ort auf dem Planeten.""")
```

Skript aufrufen

- STRG + ö
- ins Verzeichnis navigieren
- python DATEINAME.py

Anweisungsaufbau und Blöcke

ANWEISUNGSKOPF:DOPPELPUNKT

EINRÜCKUNG ANWEISUNG

Fehlerbehandlung / error handling

```
for x in range
```

```
File "<ipython-input-1-a89b35aad551>", line 1
    for x in range
           ^
```

SyntaxError: invalid syntax

help() Function

```
help(len)
```

Help on built-in function len in module builtins:

```
len(obj, /)
```

Return the number of items in a container.

Jupyter Notebooks einrichten

- Windows-Startmenu aufrufen
- unter 'A': Anaconda auswählen
- Anaconda Navigator anklicken
- Jupyter Lab installieren

Jupyter Lab starten

- Windows-Startmenu aufrufen
- unter 'A': Anaconda auswählen
- Anaconda-Prompt anklicken
- 'jupyter lab' eingeben

Jupyter Notebook starten

- ganz links 'Files' klicken
- in das Arbeitsverzeichnis navigieren
- im Launcher Notebook Python 3 auswählen

Jupyter Notebook Short Cuts

- STRG + Eingabe: = Zelle ausführen
- B = neue Zelle unten einfügen

Kontrollstrukturen

For-Schleife

```
for x in range(10):  
    print(x)
```

While-Schleife

```
x = 10  
while x > 0:  
    print(x)  
    x -= 1
```

If-Anweisung

```
for x in range(1, 11):  
    if x % 2 == 0:  
        print(f"{x} ist eine gerade Zahl.")  
    else:  
        print(f"{x} ist eine ungerade Zahl.")
```

Dateien lesen 1

```
f = open("DATEI.txt", "r", encoding="utf-8")  
grundgesetz = f.read()  
f.close()
```

Dateien lesen 2

```
with open("DATEI.txt", "w", encoding="utf-8") as f:  
    f.read()
```

I/O Parameter

- “r” = read / lesen
- “w” = write / schreiben
- “a” = append / anhängen