



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences

# FS24 CAS PML - Python

Niklaus Johner

[niklausbernhard.johner@bfh.ch](mailto:niklausbernhard.johner@bfh.ch)

# FS24 CAS PML - Python

## 14. File input/output

# File input/output

- ▶ Eine Datei wird einfach mit `open(filename, mode)` geöffnet

```
f = open("path/to/file", "r") #read mode  
f = open("path/to/file", "w") #write mode  
f = open("path/to/file", "a") #append mode
```

- ▶ Mit der `close` Methode geschlossen

```
f.close() # close file
```

- ▶ Mit `with` wird das file automatisch geschlossen (context manager)

```
with open("path/to/file", "r") as f:  
    #do something with file
```

# File input/output

## ► Von einem file lesen:

```
f.read() #read the whole file -> str  
f.readline() #read one line -> str  
f.readlines() #read all lines -> list
```

## ► Über das file direkt iterieren

```
for line in file:  
    #do something
```

## ► In ein file schreiben

```
f.write("0,1\n1,2") #write into the file  
f.writelines(["0,1\n", "1,2"]) #write list to file
```

# CSV lesen und schreiben

- ▶ Das *csv* Modul ist in den Standard libraries enthalten.
- ▶ Das *pandas* Modul ist ein Datenanalyse Modul
  - ▶ *DataFrame* Klasse
- ▶ Hat eine bessere Funktion (*read\_csv*) um csv Dateien zu lesen:

```
import pandas
with open("test.csv", "r") as f:
    data_frame = pandas.read_csv(f)
```

- ▶ Ein *DataFrame* kann direkt als csv geschrieben werden

```
with open("test_out.csv", "w") as f:
    data_frame.to_csv(f)
```

# Das *pickle* Modul

- ▶ *pickle* kann python Objekte speichern und lesen
  - ▶ *pickle.dump(object, file)*
  - ▶ *object = pickle.load(file)*

```
#Pickle
import pickle
d = {1:2, "k":[1, 2, 3], "fun":print}
with open("dict.pkl", "wb") as fout:
    pickle.dump(d, fout)

with open("dict.pkl", "rb") as fin:
    d2 = pickle.load(fin)
```