



# Dateien

# Dateien



## Verschiedene Modi im Detail

- obligatorisch: `r, w, a, x`
- optional: `+, b, t`

# Dateien



Modus "r"

- nur Leseoperationen
- Zeiger am Anfang der Datei
- FileNotFoundError, wenn Datei nicht vorhanden

# Dateien



Modus "r"

```
1 with open( "neueDatei.txt", "r" ) as datei:  
2     print( 'Zeiger auf: ', datei.tell() )  
3     b = datei.read()
```

```
Zeiger auf: 0
```

# Dateien



Modus "r+"

- Schreib- und Leseoperationen
- Zeiger am Anfang der Datei
- FileNotFoundError, wenn Datei nicht vorhanden

# Dateien



Modus "r+"

```
1 with open( "neueDatei.txt", "r+" ) as datei:  
2     print( 'Zeiger auf: ', datei.tell() )  
3     datei.write("neue Zeile a\n")  
4     b = datei.read()  
5     datei.write("neue Zeile b\n")  
6     print( 'Zeiger auf: ', datei.tell() )
```

Zeiger auf: 0

Zeiger auf: 53

# Dateien



Modus "w"

- Schreiboperationen
- Zeiger am Anfang der Datei
- Datei wird erzeugt oder überschrieben

# Dateien



Modus "w+"

- Lese- und Schreiboperationen
- Zeiger am Anfang der Datei
- Datei wird erzeugt oder überschrieben



# Dateien



Modus "a"

- Schreiboperationen
- Zeiger am Ende der Datei
- Datei wird erzeugt wenn nicht vorhanden

# Dateien



Modus "a+"

- Lese- und Schreiboperationen
- Zeiger am Ende der Datei
- Datei wird erzeugt wenn nicht vorhanden

# Dateien



Modus "x"

- keine Lese- und Schreiboperationen
- Zeiger am Anfang = Ende der Datei
- `FileExistsError`, wenn Datei vorhanden

# Dateien



Modus "x+"

- Lese- und Schreiboperationen
- Zeiger am Anfang=Ende der Datei
- FileNotFoundError, wenn Datei vorhanden

# Dateien



Modus "rb"

liest

bytestring

```
1 with open( "neueDatei.txt", "rb" ) as datei:  
2     zeile = datei.readline()  
3     print( zeile )
```

```
b'Zeile 1\n'
```

```
4
```

```
5
```

# Dateien



Modus "wb"

schreibt nur  
bytestring

```
1 with open( "neueDatei.txt", "wb" ) as datei:  
2     datei.write('abcde')
```

```
TypeError: a bytes-like object is required,  
not 'str'
```

# Dateien



Fehler vermeiden statt abfangen

z.B. mit `isfile()`

```
1 from os.path import isfile
2
3 print( isfile('neueDatei.txt') )
True
```

# Dateien



## Kopieren einer Datei

```
1 with open("in.txt", "rb") as in_file, open(  
2     "out.txt", "wb") as out_file:  
3  
4     out_file.write( in_file.read() )
```



# Dateien



The following code snippet when run will (select all that apply):

```
1 f = open("file.txt", "w")  
2 f.close()
```

- a) open the file file.txt in write mode
- b) delete the file contents if the file file.txt already exists
- c) leave the file contents unchanged if the file file.txt already exists
- d) create the file file.txt if it does not exist
- e) raise the FileNotFoundError exception if the file does not exist

# Dateien



What is the result of the following code?

```
1 with open('txt.txt', 'r+') as file:  
2     line = file.read()  
3     file.write(line)
```

given txt.txt

12345

- a) txt.txt will still contain 12345
- b) txt.txt will now contain 123451
- c) txt.txt will now contain 1234512345
- d) NameError: name 'line' is not defined
- e) ValueError: I/O operation on closed file.

# Dateien



## Aufgabe

Schreiben Sie ein Programm, das

- von der Tastatur einen Dateinamen `dn` als String einliest
- den Inhalt der Datei `dn.txt` einliest
- eine neue Datei `dn10.txt` erzeugt, die
- von jeder Originalzeile die maximal 10 ersten Zeichen enthält.