Merkblatt Dateien

1 Dateien: Daten einlesen und schreiben

Dateien sind Daten, die auf einem Datenträger gespeichert wurden. Zum Schreiben, Lesen und Bearbeiten einer Datei muss diese göffnet werden. Dabei wird ein Referenzobjekt erstellt, das in Python *stream* genannt wird.

Nach der Operation ist die Datei wieder zu schließen.

1.1 Dateimodus

Eine Datei kann in verschiedenen Modi geöffnet werden. Nur mit dem richtigen Dateimodus sind bestimmte Operationen möglich.

Modus	Beschreibung
r	Die Datei wird nur zum Lesen (read) geöffnet. Dies ist der Standardmodus. Ohne Angabe eines Modus wird diseer verwendet.
w	Die Datei wird zum Schreiben geöffnet. Wenn die Datei noch nicht vorhanden ist, dann wird sie neu angelegt. Wenn die Datei bereits vorhanden ist, dann wird sie geleert. Achtung! Leeren heißt, dass der vorhandene Inhalt gelöscht wird.
x	Dieser Modus erzeugt zwingend eine neue Datei oder bricht mit einem Fehler ab. Dieser Modus ist sicherer als `w`, da so nicht versehentlich Daten gelöscht werden.
a	Neu geschriebene Daten werden ans Ende der Datei angehängt (append).

1.2 open()

Geöffnet wird eine Datei, wie folgt:

```
f = open("test.txt", encoding = "utf-8")
```

`f` enthält dann den *stream*.

Der erste Parameter gibt den Namen der Datei samt Pfad an.

Der Parameter `encoding` gibt an, wie die Daten auf dem Datenträger kodiert sind. Hierzu wird gibt es in Abschnitt 05 ein Merkblatt *Kodierungen*

Mit dem `with`-Statement kann eine Datei geöffnet werden, so dass im nachfolgenden Block gearbetiet werden kann. Das sieht beispielsweise so aus:

```
with open("text.txt") as f:
    content = f.read()
```

1.3 close()

Die Datei mit dem *stream* `f` wird so wieder geschlossen:

```
f.close()
```

1.4 read()

Der Inhalt einer Datei wird mit `read()` gelesen.

```
content = f.read()
```

Anschließend enthält die Variable `content` den Inhalt der Datei zum *stream* `f`.

*Hinweis: Mit ´readline()´ existiert eine weitere Funktion, mit der Daten aus einem Stream eingelesen werden können. Sie liest jedes mal genau eine Zeile ein.

1.5 os.path.exists()

Diese Funktion wird erst in Abschnitt 5 erklärt.

Nach einem 'import os' steht auch eine Funktion zur Verfügung, mit der die Existenz einer Datei oder eines Verzeichnisses geprüft werden kann.

[3]:
```
import os
if os.path.exists("text.txt"):
    print("Die Datei text.txt existiert.")
else:
    print("Die Datei text.txt existiert nicht.")
```

```
Die Datei text.txt existiert nicht.
```