

1

# File Input/Output







### File IO in Python

Often you want to put calculated or generated data in a file so that the results are still available later.

Or, you may already have a file with data that you want to analyze.



In Python, storing data in files and reading date from files is very easy.





### Write a list of numbers to a file

Write a list of numbers to a file

```
liste = [3, 4, 3.5, 'a']
path="text.txt"
with open(path, 'w') as file:
    for n in range(0, len(liste)):
        elem = liste[n]
        file.write(str(elem)+"\n" )
```

Open file under the path in *write* mode

Convert elem to text -> only texts can be written to files, end each line with "n" -> new line

The file is automatically closed at the end







### Write some text into a file

#### Write personal data to a file

Open the file under the path in *append* mode Separate individual items by ";" End each line with a *newline* -> "\n" Append line to the file

The content of the person.txt file:

54;Urs;Muster;Bahnhofstr. 6; Bern 57;Hans;Muster;Hauptstr. 3; Bern 60;Anna;Muster;Hauptstr. 7; Bern 65;Ida;Muster;Bahnhofstr. 5; Bern 74;Karin;Muster;Hauptstr. 1; Bern



### Read numbers from a file

```
with open(path, 'r') as reader:
    numbers = list()
    for line in reader:
        try:
            numbers.append(float(line))
        except:
            pass
print(numbers)
```



Convert read lines in numbers (if possible). Ignore exceptions

Print list of read numbers







### Read text from a file

```
fileContent = list()
path = "people.txt"
with open (path, 'r') as reader:
    for line in reader:
        try:
            line = line.rstrip() #remove \n
            words = line.split(";") #split into words
            fileContent.append(words)
            except:
            pass
for elem in fileContent:
            print(elem)
```



54;Urs;Muster;Bahnhofstr. 6; Bern
 57;Hans;Muster;Hauptstr. 3; Bern
 60;Anna;Muster;Hauptstr. 7; Bern
 63;Peter;Muster;Hauptstr. 8; Bern
 65;Ida;Muster;Bahnhofstr. 5; Bern
 74;Karin;Muster;Hauptstr. 1; Bern

people.txt

Open file at path for reading
Remove newline with rstrip
Words are separated by ";" Append words to list.



print fileContent



### Append to an existing file with 'a'







## SQLite Database sqlite3







### **Database Access in Python**

Often you want to put calculated or generated data in a

Or, you may already have a database with data that you want to analyze.



With python and sqlite storing data and reading data is very easy.





### **Create new Database**

import sqlite3

dbName = "database.db" dropTable\_statement = "DROP TABLE IF EXISTS person" # The name of the database file# This will drop the table if it exists# Create person table statement

createTable\_statement = "CREATE TABLE IF NOT EXISTS person (id INTEGER PRIMARY KEY, firstName TEXT, lastName TEXT);"

#### def createDatabase():

# This function creates the database and the person table

#### try:

with sqlite3.connect(dbName) as connection: cursor = connection.cursor() cursor.execute(dropTable\_statement) cursor.execute(createTable\_statement) connection.commit() except sqlite3.Error as e: print("Error: " + str(e))

# Open a connection to the database
# Drop the table if it exists
# Create the new table
# Commit the changes to the database

*# Print any error that occurs* 



### Insert values into DB

import sqlite3

dbName = "database.db,,

*# SQL statement to insert values into the person table* insert statement = "INSERT INTO person (firstName, lastName) VALUES (?, ?);"

#### def insertIntoDatabase(firstName, lastName):

*#* This function inserts one new person into the database try:

```
with sqlite3.connect(dbName) as connection:
    cursor = connection.cursor()
                                                        # Open a connection to the database
    cursor.execute(insert statement, (firstName, lastName))
    connection.commit()
                                                        # Commit the changes to the database
except sqlite3.Error as e:
                                                        # Print any error that occurs
  print("Error: " + str(e))
```





### Read values from DB

import sqlite3
dbName = "database.db"

# SQL statement to read all values from the person table read\_statement = "SELECT \* FROM person"

#### def readFromDatabase():

*# This function reads all values from the person table and prints them* **try**:

#### with sqlite3.connect(dbName) as connection:

cursor = connection.cursor()
cursor.execute(read\_statement)
rows = cursor.fetchall()
for row in rows:
 print(row)
except sqlite3.Error as e:
 print("Error: " + str(e))

# Open a connection to the database# Execute the read statement# Fetch all rows from the result set

*# Print each row* 

*# Print any error that occurs* 



### Update values in DB

import sqlite3 dbName = "database.db"

*# SQL statements to update values in the person table* update statement = "UPDATE person SET firstName = ?, lastName = ? WHERE id = ?;"

#### def updateDatabase(id, firstName, lastName):

# This function updates a person's first and last name in the database try:

```
with sqlite3.connect(dbName) as connection:
    cursor = connection.cursor()
                                                        # Open a connection to the database
    cursor.execute(update statement, (firstName, lastName, id))
    connection.commit()
                                                        # Commit the changes to the database
except sqlite3.Error as e:
  print("Error: " + str(e))
                                                        # Print any error that occurs
```





### **Test main Function**

# Initial values to be inserted into the database values = [["Peter","Muster"], [ "Paul","Muster"], [ "Mary", "Muster"]]

#### def main():

print("Creating database and inserting initial values...")

createDatabase()

for firstName, lastName in values:

insertIntoDatabase(firstName, lastName)

readFromDatabase()

*# Create the database and table, and insert initial values* 

# Insert initial values into the database *#* To verify initial values

### print("\nUpdate database and set new value...")

updateDatabase(2, "John", "Smith") readFromDatabase()

*# Update the person with id 2 # To verify the update* 

1	Peter	Muster
2	John	Smith
3	Mary	Muster

