08 File IO and Error Handling-LECTURE

March 4, 2022

1 Module 8: File I/O and Error Handling

March 4, 2022

Last time we covered data structures (lists, tuples, dictionaries, sets) in Python that allow us to work with more powerful data items than just the individual numbers, strings and Booleans that we had used before. We also discussed the important difference between call by value and call by reference.

Until now the course dealt with the basics of imperative programming in Python, and you have learned about the most important concepts that you need as a programmer. We will now leave the relatively secluded, controlled environment that we were in so far and look at how to read and write data from and to files, access online resources, use external libraries, and connected to that how to make programs more robust against errors that come from "the outside".

Today we will cover how to read and write files in general, how to deal with CSV files in particular, and how to handle runtime errors that can for example be caused by user inputs or file operations.

Next time we will have a look at fetching data and other resources from the internet, and how to interact with web services from within Python programs.

1.1 Reading and Writing Files

Python distinguishes between only two types of files: **text and binary**. Basically, anything that is not a text file is regarded as a binary. Text files are sequences of lines, which are themselves sequences of characters that are terminated with a special end-of-line (EOL) character, often the newline character. The content of text files can be processed with the common string manipulation functionality, while processing binary files requires knowledge about their structure. For the moment we are only concerned with text files.

To open a file, first a file object needs to be created with the `open()` function:

`<file_object> = open(<filename>, <mode>)`

`<filename>` is the name (path) of the file to open, and `<mode>` specifies for which kind of processing the file is opened ("r" for reading content, "w" for writing content, "a" for appending content, or "r+" for a special read and write mode). For example:

[3]:
```python
# create a file object in reading mode
file = open("data/shorttext.txt", "r")
print(file)
file.close()
```

```
<_io.TextIOWrapper name='data/shorttext.txt' mode='r' encoding='UTF-8'>
```

When the file is opened, operations according to the chosen mode can be carried out. When all operations on the file have been performed, the file should be closed again to avoid unintended side effects:

```
file.close()
```

Play around with the following code examples and a small text file if your choice to see what happens. Add printouts to visualize what has been read by the different commands.

For example, when opened in reading mode we can call different functions for reading content from the file:

[7]:
```python
# creating a file object in reading mode
file = open("data/shorttext.txt", "r")

# file.read() to read all characters in the file
# content = file.read()
# print(content)

# file.read(n) to read the first/next n characters of the file
# first_n = file.read(10)
# print(first_n)

# file.readline() to read a (the first/next) line of the file
# first_line = file.readline()
# print(first_line)

# file.readlines to read the content of the files line by line
lines = file.readlines()
print(lines)

# close file
file.close()
```

```
['Invisible Fish\n', 'BY JOY HARJO\n', 'Invisible fish swim this ghost ocean now
described by waves of sand, by water-worn rock.\n', 'Soon the fish will learn to
walk.\n', 'Then humans will come ashore and paint dreams on the dying stone.\n',
'Then later, much later, the ocean floor will be punctuated by Chevy trucks,
carrying the dreamers' decendants, who are going to the store.\n']
```

When opened in writing mode, we can call diffent functions to write text into the file:

[9]:
```python
# creating a file object in writing mode
file = open("data/textdump.txt", "w")

# file.write to write (or append) text to a file
file.write("Hello World!\n")
file.write("It's cold today...\n")
file.writelines(["Another line\n", "and another line\n"])

# close file
file.close()
```

Change this example from writing to appending mode (parameter “a”) and see what the difference is.

With the `with`-statement, Python provides an alternative, elegant way to handle files. It also takes care of closing the file, so it is a good idea to make it a habit to use it for file handling (and never forget closing):

[10]:
```
with open("data/shorttext.txt", "r") as file:
    content = file.read()

with open("data/newtext.txt", "w") as file:
    file.write("Hello World!\n")
    file.write("It's cold today...\n")
    file.writelines(["Another line\n", "and another line\n"])
```

Note that here is also a short and elegant way to iterate over all lines of a file, without explicitly calling `readlines()` before:

```
for line in file:
    <do something with line>
```

As a more complete example, see the following code to read the text from a file, encrypt it using the Caesar cipher, and write it into another file:

[11]:
```
from caesarcipher import CaesarCipher

with open("data/shorttext.txt", "r") as file:
    content = file.read()

content_encrypted = CaesarCipher(content, offset=3)

with open("data/shorttext_encrypted.txt", "w") as file:
    file.write(content_encrypted.encoded)
```

This code produces no output on the command line, but if you try it with a text file yourself, you will see the effect in the new file that is created.

Challenge!

Create a function that receives a filename and counts the number of times each word appears in the text file.

Hints: 1. Use a dictionary to store the word frequencies 2. Don't care for punctuation 3. Use a loop to go over the words 4. Use the `split()` method to get the list of words from a string. E.g.

```
[12]: text = "this is a sample text"
      text.split()
```

```
[12]: ['this', 'is', 'a', 'sample', 'text']
```

```
[13]: def word_frequencies(file):
          frequencies = {}
          with open(file, "r") as file:
              content = file.read()
              for word in content.split():
                  if word in frequencies:
                      frequencies[word] += 1
                  else:
                      frequencies[word] = 1
              return frequencies

      word_frequencies("data/shorttext.txt")
```

```
[13]: {'Invisible': 2,
       'Fish': 1,
       'BY': 1,
       'JOY': 1,
       'HARJO': 1,
       'fish': 2,
       'swim': 1,
       'this': 1,
       'ghost': 1,
       'ocean': 2,
       'now': 1,
       'described': 1,
       'by': 3,
       'waves': 1,
       'of': 1,
       'sand,': 1,
       'water-worn': 1,
       'rock.': 1,
       'Soon': 1,
       'the': 5,
       'will': 3,
       'learn': 1,
       'to': 2,
       'walk.': 1,
       'Then': 2,
       'humans': 1,
       'come': 1,
       'ashore': 1,
       'and': 1,
       'paint': 1,
       'dreams': 1,
       'on': 1,
       'dying': 1,
       'stone.': 1,
       'later,': 2,
       'much': 1,
       'floor': 1,
       'be': 1,
       'punctuated': 1,
       'Chevy': 1,
       'trucks,': 1,
       'carrying': 1,
       'dreamers’': 1,
       'decendants,': 1,
       'who': 1,
       'are': 1,
       'going': 1,
       'store.': 1}
```

1.2 Dealing With CSV Files

Let’s look at another kind of text file, that you will frequently come across when working on data science problems: CSV files. CSV stands for “comma-separated values” and means that commas are used to separate the values in a line from each other. Sometimes also other characters are used as separators, such as the tabulator “” or the semicolon”;”, so don’t be confused if you see that. As such, CSV files are a simple means to represent tabular data. The following example is based on the Dutch municipalities data set from Kaggle (https://www.kaggle.com/justinboon/municipalities-of-the-netherlands/data), stored in the file dutch_municipalities.csv. We can open and read this file as in the examples above:

[14]:
```
with open("data/dutch_municipalities.csv", "r") as csvfile:
    print(csvfile.read())
```

```
municipality    province        latitude        longitude       surface_km2
population      avg_household_income_2012       avg_woz_2014    university
Aa en Hunze     Drenthe 53.010.485      6.749.528       278.9   25243   35500
225000  0
Aalburg Noord-Brabant   51.751.294      5.057.085       53.17   12859   39100
249000  0
Aalsmeer        Noord-Holland   52.262.164      4.761.922       32.29   30792
40900   276000  0
Aalten  Gelderland      51.926.667      6.580.678       96.57   27030   33300
194000  0
Achtkarspelen   Friesland       53.210.357      6.153.565       103.98  28002
```

```
30500   165000  0
Alblasserdam    Zuid-Holland    51.870.337      4.670.202       10.06   19822
35500   195000  0
Albrandswaard   Zuid-Holland    51.858.068      4.423.187       23.75   25044
42700   255000  0
Alkmaar Noord-Holland   52.632.842      4.755.037       31.2    94906   32300
181000  0
Almelo  Overijssel      52.367.027      6.668.492       69.4    72435   31000
156000  0
Almere  Flevoland       52.350.785      5.264.702       248.77  196156  34900
182000  0
Alphen aan den Rijn     Zuid-Holland    52.111.222      4.647.251       132.49
106809  36900   220000  0
Alphen-Chaam    Noord-Brabant   51.509.135      4.861.589       93.51   9712
40000   295000  0
Ameland Friesland       53.440.564      5.658.766       268.5   3565    31100
201000  0
Amersfoort      Utrecht 52.156.111      5.387.827       63.86   150943  36900
222000  0
Amstelveen      Noord-Holland   52.311.421      4.870.087       44.08   85135
40200   257000  0
Amsterdam       Noord-Holland   52.370.216      4.895.168       219.3   853312
31400   231000  2
Apeldoorn       Gelderland      52.211.157      5.969.923       341.15  157535
34800   208000  0
Appingedam      Groningen       53.320.678      6.854.422       24.58   12049
29800   142000  0
Arnhem  Gelderland      51.985.103      5.898.730       101.54  150817  30500
175000  0
Assen   Drenthe 52.992.753      6.564.228       83.45   67209   32200   162000
0
Asten   Noord-Brabant   51.402.994      5.744.078       71.34   16479   35800
275000  0
Baarle-Nassau   Noord-Brabant   51.445.137      4.929.523       76.21   6617
35400   250000  0
Baarn   Utrecht 52.213.182      5.286.410       33.01   24344   38600   290000
0
Barendrecht     Zuid-Holland    51.851.509      4.548.581       21.73   47375
41600   244000  0
Barneveld       Gelderland      52.162.147      5.655.360       176.69  54176
38000   279000  0
Bedum   Groningen       53.301.675      6.599.829       44.96   10475   33900
176000  0
Beek    Limburg 50.939.316      5.795.647       21.03   16268   35300   184000
0
Beemster        Noord-Holland   52.547.559      4.913.332       72.07   8919
41400   304000  0
Beesel  Limburg 51.269.698      6.046.834       29.15   13593   33600   186000
```

```
0
Berg en Dal     Gelderland      51.818.121      5.920.082       44.14   18956
0
Bergeijk        Noord-Brabant   51.319.716      5.357.500       101.73  18250
37700   291000  0
Bergen (L.)     Limburg 51.599.866      6.032.998       108.48  13230   34000
208000  0
Bergen (NH.)    Noord-Holland   52.674.937      4.706.395       119.46  30075
39400   349000  0
Bergen op Zoom  Noord-Brabant   51.494.576      4.287.162       93.13   66423
33600   195000  0
Berkelland      Gelderland      52.115.157      6.561.501       260.53  44650
34700   205000  0
Bernheze        Noord-Brabant   51.660.272      5.501.401       90.41   29703
38500   278000  0
Best    Noord-Brabant   51.507.764      5.397.848       35.1    28594   39100
262000  0
Beuningen       Gelderland      51.859.724      5.769.107       47.09   25254
37400   231000  0
Beverwijk       Noord-Holland   52.486.984      4.657.447       20.09   40052
32500   178000  0
De Bilt Utrecht 52.109.272      5.180.968       67.13   42013   43100   338000
0
Binnenmaas      Zuid-Holland    51.796.188      4.548.157       75.57   28682
39300   231000  0
Bladel  Noord-Brabant   51.362.963      5.213.639       75.62   19825   37200
272000  0
Blaricum        Noord-Holland   52.272.669      5.248.080       15.56   9112
51600   536000  0
Bloemendaal     Noord-Holland   52.404.947      4.620.185       45.18   22077
55600   549000  0
Bodegraven-Reeuwijk     Zuid-Holland    52.082.326      4.746.084       88.64
32986   41200   275000  0
Boekel  Noord-Brabant   51.602.879      5.674.316       34.52   10089   37300
260000  0
Ten Boer        Groningen       53.275.888      6.692.600       45.73   7465
34600   174000  0
Borger-Odoorn   Drenthe 52.889.372      6.888.404       277.89  25633   33100
180000  0
Borne   Overijssel      52.300.237      6.753.726       26.16   21901   36300
207000  0
Borsele Zeeland 51.447.691      3.803.318       194.52  22592   36200   209000
0
Boxmeer Noord-Brabant   51.645.269      5.956.666       113.84  28135   37100
251000  0
Boxtel  Noord-Brabant   51.601.437      5.312.177       64.85   30325   35800
244000  0
Breda   Noord-Brabant   51.571.915      4.768.323       128.68  179999  35200
```

```
221000  0
Brielle Zuid-Holland    51.902.582      4.163.685       31.14   16306   38100
225000  0
Bronckhorst     Gelderland      52.075.595      6.180.892       286.42  36941
36200   257000  0
Brummen Gelderland      52.090.167      6.158.015       85.01   21169   35600
240000  0
Brunssum        Limburg 50.948.896      5.972.166       17.34   28914   29400
133000  0
Bunnik  Utrecht 52.066.528      5.200.776       37.57   14619   42600   271000
0
Bunschoten      Utrecht 52.240.642      5.367.070       34.81   20547   39400
250000  0
Buren   Gelderland      51.910.389      5.334.058       142.92  25995   39500
282000  0
Capelle aan den IJssel  Zuid-Holland    51.938.073      4.591.778       15.4
66177   34700   176000  0
Castricum       Noord-Holland   52.545.259      4.672.735       60.37   34244
40200   259000  0
Coevorden       Drenthe 52.661.357      6.741.062       299.69  35771   33200
191000  0
Cranendonck     Noord-Brabant   51.286.922      5.596.695       78.05   20388
36600   241000  0
Cromstrijen     Zuid-Holland    51.736.089      4.460.849       70.33   12748
39400   238000  0
Cuijk   Noord-Brabant   51.728.927      5.879.209       57.07   24780   34200
212000  0
Culemborg       Gelderland      51.956.108      5.240.045       31.14   27579
36300   220000  0
Dalfsen Overijssel      52.507.755      6.259.667       166.52  27655   37000
255000  0
Dantumadiel     Friesland       53.272.981      5.985.930       87.53   19035
31500   168000  0
Delft   Zuid-Holland    52.011.577      4.357.068       24.06   101493  32100
181000  1
Delfzijl        Groningen       53.331.027      6.924.460       227.5   25686
30600   132000  0
Deurne  Noord-Brabant   51.464.220      5.795.068       118.36  31669   36000
254000  0
Deventer        Overijssel      52.266.075      6.155.217       134.33  98327
32800   191000  0
Diemen  Noord-Holland   52.338.993      4.959.189       14.04   25980   35200
202000  0
Dinkelland      Overijssel      52.407.617      6.897.260       176.82  25937
38300   248000  0
Doesburg        Gelderland      52.014.288      6.139.782       12.96   11449
32100   190000  0
Doetinchem      Gelderland      51.964.699      6.293.774       79.66   56350
```

```
33200   194000  0
Dongen  Noord-Brabant   51.628.710      4.938.239       29.72   25389   35900
226000  0
Dongeradeel     Friesland       53.324.884      5.996.639       266.92  24183
30400   149000  0
Dordrecht       Zuid-Holland    51.813.298      4.690.093       99.47   118716
32600   159000  0
Drechterland    Noord-Holland   52.653.307      5.183.589       80.74   19257
37600   231000  0
Drimmelen       Noord-Brabant   51.708.552      4.803.490       119.43  26705
38000   234000  0
Dronten Flevoland       52.534.682      5.721.809       423.89  40451   34800
195000  0
Druten  Gelderland      51.894.084      5.594.272       42.46   18206   36200
230000  0
Duiven  Gelderland      51.947.458      6.017.950       35.19   25580   36200
210000  0
Echt-Susteren   Limburg 51.093.224      5.907.364       104.53  31940   34100
192000  0
Edam-Volendam   Noord-Holland   52.504.541      5.045.998       24.78   28934
40600   238000  0
Ede     Gelderland      52.040.168      5.664.859       318.62  110657  36400
233000  0
Eemnes  Utrecht 52.253.746      5.261.275       33.7    8773    40100   291000
0
Eemsmond        Groningen       53.400.344      6.648.410       543.35  15910
30700   154000  0
Eersel  Noord-Brabant   51.357.808      5.315.912       83.33   18201   39500
298000  0
Eijsden-Margraten       Limburg 50.819.960      5.821.457       78.32   24980
38200   237000  0
Eindhoven       Noord-Brabant   51.441.642      5.469.722       88.87   220782
32000   209000  1
Elburg  Gelderland      52.449.263      5.834.519       65.91   22658   35200
240000  0
Emmen   Drenthe 52.713.237      6.955.777       346.25  108003  30700   156000
0
Enkhuizen       Noord-Holland   52.707.566      5.274.120       116.25  18395
32700   180000  0
Enschede        Overijssel      52.221.537      6.893.662       142.72  158542
29600   155000  1
Epe     Gelderland      52.345.017      5.983.654       157.37  32352   35600
266000  0
Ermelo  Gelderland      52.298.665      5.629.619       87.33   26080   36300
262000  0
Etten-Leur      Noord-Brabant   51.586.886      4.667.138       55.92   42351
35600   224000  0
Ferwerderadiel  Friesland       53.337.834      5.823.835       133.18  8771
```

```
31800   148000  0
De Fryske Marren        Friesland       52.948.003      5.791.388       559.93
51223                   0
Geertruidenberg Noord-Brabant   51.702.366      4.849.978       29.64   21597
35600   204000  0
Geldermalsen    Gelderland      51.884.365      5.229.359       101.73  26339
39500   278000  0
Geldrop-Mierlo  Noord-Brabant   51.432.447      5.582.970       31.39   38854
35400   233000  0
Gemert-Bakel    Noord-Brabant   51.555.289      5.690.366       123.34  29354
35500   244000  0
Gennep  Limburg 51.697.854      5.972.675       50.42   17285   34500   215000
0
Giessenlanden   Zuid-Holland    51.865.735      4.924.345       65.11   14423
41700   281000  0
Gilze en Rijen  Noord-Brabant   51.559.855      4.909.254       65.66   26013
35700   233000  0
Goeree-Overflakkee      Zuid-Holland    51.759.197      4.116.981       422.34
48233   36500   220000  0
Goes    Zeeland 51.504.646      3.891.130       101.92  36977   33600   198000
0
Goirle  Noord-Brabant   51.523.677      5.064.195       42.34   23060   38100
251000  0
Gooise Meren    Noord-Holland   52.304.762      5.139.699       73.5    56687
0
Gorinchem       Zuid-Holland    51.837.225      4.975.829       21.93   35252
34000   194000  0
Gouda   Zuid-Holland    52.011.521      4.710.463       18.11   70923   35000
182000  0
Grave   Noord-Brabant   51.759.005      5.738.560       28.03   12696   35300
215000  0
's-Gravenhage   Zuid-Holland    52.070.498      4.300.700       98.12   508592
31800   188000  0
Groningen       Groningen       53.219.383      6.566.502       83.75   198108
28500   157000  1
Grootegast      Groningen       53.211.132      6.274.589       87.74   12193
32500   192000  0
Gulpen-Wittem   Limburg 50.800.934      5.897.288       73.35   14492   35200
204000  0
Haaksbergen     Overijssel      52.160.665      6.738.100       105.5   24357
35600   215000  0
Haaren  Noord-Brabant   51.602.133      5.226.712       58.56   13578   40800
332000  0
Haarlem Noord-Holland   52.387.388      4.646.219       32.09   155205  34300
229000  0
Haarlemmerliede en Spaarnwoude  Noord-Holland   52.414.352      4.692.848
21.19   5547    40300   267000  0
Haarlemmermeer  Noord-Holland   52.300.378      4.674.359       185.29  144166
```

```
39700   248000  0
Halderberge     Noord-Brabant   51.586.263      4.513.956       75.21   29379
36200   232000  0
Hardenberg      Overijssel      52.575.408      6.616.695       317.14  59592
34000   197000  0
Harderwijk      Gelderland      52.342.202      5.636.742       48.27   45741
34800   215000  0
Hardinxveld-Giessendam  Zuid-Holland    51.832.430      4.833.641       19.35
17736   37500   225000  0
Haren   Groningen       53.171.826      6.605.243       50.73   18790   41300
261000  0
Harlingen       Friesland       53.174.638      5.425.152       387.67  15810
30100   153000  0
Hattem  Gelderland      52.473.931      6.068.176       24.16   11742   37700
253000  0
Heemskerk       Noord-Holland   52.514.146      4.682.137       31.67   39092
35600   212000  0
Heemstede       Noord-Holland   52.351.063      4.620.300       9.64    26320
47600   370000  0
Heerde  Gelderland      52.390.918      6.049.638       80.42   18515   36200
254000  0
Heerenveen      Friesland       52.960.561      5.920.522       187.76  49388
32600   183000  0
Heerhugowaard   Noord-Holland   52.662.677      4.832.477       39.99   53246
35300   203000  0
Heerlen Limburg 50.888.174      5.979.499       45.53   88202   28200   123000
0
Heeze-Leende    Noord-Brabant   51.360.442      5.598.460       105.04  15376
40200   312000  0
Heiloo  Noord-Holland   52.601.234      4.700.493       19.01   22626   40200
288000  0
Den Helder      Noord-Holland   52.956.281      4.760.797       178.8   56553
30400   147000  0
Hellendoorn     Overijssel      52.376.905      6.459.724       138.99  35697
34700   224000  0
Hellevoetsluis  Zuid-Holland    51.831.863      4.131.810       46.27   38918
36500   195000  0
Helmond Noord-Brabant   51.479.255      5.657.010       54.75   89346   32900
206000  0
Hendrik-Ido-Ambacht     Zuid-Holland    51.842.397      4.639.506       11.9
28952   39100   215000  0
Hengelo Overijssel      52.257.412      6.792.772       61.83   80975   32200
168000  0
's-Hertogenbosch        Noord-Brabant   51.711.742      5.301.631       91.79
143745  34900   221000  0
Heumen  Gelderland      51.764.532      5.843.150       41.54   16342   40000
273000  0
Heusden Noord-Brabant   51.733.004      5.138.279       81.22   43180   37200
```

```
266000  0
Hillegom        Zuid-Holland    52.295.626      4.579.176       13.48   20987
36000   227000  0
Hilvarenbeek    Noord-Brabant   51.465.175      5.151.090       96.49   15082
39400   294000  0
Hilversum       Noord-Holland   52.229.170      5.166.897       46.35   86574
36400   253000  0
Hof van Twente  Overijssel      52.241.427      6.591.555       215.41  34987
36300   229000  0
Hollands Kroon  Noord-Holland   52.811.859      5.001.190       662.2   47501
34900   206000  0
Hoogeveen       Drenthe 52.728.616      6.490.100       129.25  54680   31200
164000  0
Hoorn   Noord-Holland   52.642.365      5.060.212       53.25   71741   33800
186000  0
Horst aan de Maas       Limburg 51.423.226      6.030.317       191.92  41718
36600   237000  0
Houten  Utrecht 52.002.991      5.185.760       58.99   48427   42900   261000
0
Huizen  Noord-Holland   52.299.465      5.243.393       23.32   41239   38900
268000  0
Hulst   Zeeland 51.280.691      4.054.887       251.82  27402   34300   182000
0
IJsselstein     Utrecht 52.017.765      5.040.300       21.68   34268   39300
234000  0
Kaag en Braassem        Zuid-Holland    52.204.135      4.631.315       72.24
25758   39900   267000  0
Kampen  Overijssel      52.557.964      5.914.462       161.79  51069   33200
197000  0
Kapelle Zeeland 51.484.755      3.959.631       49.63   12508   37800   220000
0
Katwijk Zuid-Holland    52.198.020      4.419.943       31.13   62825   36500
249000  0
Kerkrade        Limburg 50.865.946      6.070.549       22.13   46773   28500
130000  0
Koggenland      Noord-Holland   52.652.196      4.942.015       84.13   22473
37200   232000  0
Kollumerland en Nieuwkruisland  Friesland       53.305.888      6.189.109
116.35  12856   31100   167000  0
Korendijk       Zuid-Holland    51.810.961      4.342.386       100.47  10693
39000   236000  0
Krimpen aan den IJssel  Zuid-Holland    51.914.353      4.596.233       8.95
28814   37700   218000  0
Krimpenerwaard  Zuid-Holland    51.982.222      4.781.667       161.3   54287
0
Laarbeek        Noord-Brabant   51.538.447      5.621.273       56.17   21815
36100   253000  0
Landerd Noord-Brabant   51.726.775      5.659.492       70.71   15262   38500
```

```
267000  0
Landgraaf       Limburg 50.892.765      6.022.408       24.66   37530   31400
150000  0
Landsmeer       Noord-Holland   52.440.338      4.920.923       26.5    10457
41700   280000  0
Langedijk       Noord-Holland   52.686.711      4.783.899       27.03   26984
38300   257000  0
Lansingerland   Zuid-Holland    51.998.714      4.516.263       56.37   57188
43900   268000  0
Laren   Noord-Holland   52.256.817      5.224.155       12.41   10846   50800
541000  0
Leek    Groningen       53.161.611      6.390.616       64.28   19607   32800
183000  0
Leerdam Zuid-Holland    51.894.313      5.096.927       34.42   20586   34600
211000  0
Leeuwarden      Friesland       53.201.233      5.799.913       166.99  108113
28700   135000  0
Leiden  Zuid-Holland    52.160.114      4.497.010       23.27   121199  34000
210000  1
Leiderdorp      Zuid-Holland    52.150.985      4.528.173       12.28   26788
39100   239000  0
Leidschendam-Voorburg   Zuid-Holland    52.087.731      4.399.385       35.62
73392   37100   232000  0
Lelystad        Flevoland       52.518.537      5.471.422       765.45  76170
32500   165000  0
Leudal  Limburg 51.261.966      5.890.997       164.89  36213   36500   213000
0
Leusden Utrecht 52.131.793      5.429.469       58.89   28967   41000   267000
0
Lingewaal       Gelderland      51.838.089      5.076.384       54.49   11059
39000   267000  0
Lingewaard      Gelderland      51.903.891      5.935.221       69.14   45814
36000   231000  0
Lisse   Zuid-Holland    52.257.930      4.557.483       16.05   22376   36600
251000  0
Lochem  Gelderland      52.158.665      6.409.816       215.94  33268   38500
282000  0
Loon op Zand    Noord-Brabant   51.627.014      5.072.009       50.71   23104
36600   242000  0
Lopik   Utrecht 51.974.861      4.945.148       78.98   14000   39500   267000
0
Loppersum       Groningen       53.332.175      6.747.987       111.99  10181
32400   153000  0
Losser  Overijssel      52.262.852      7.005.785       99.61   22619   34100
206000  0
Maasdriel       Gelderland      51.768.907      5.323.304       75.46   24197
37600   266000  0
Maasgouw        Limburg 51.167.805      5.884.140       57.99   23888   35100
```

```
202000  0
Maassluis       Zuid-Holland    51.922.607      4.254.566       10.12   32097
34200   187000  0
Maastricht      Limburg 50.851.368      5.690.973       60.03   122331  30400
178000  1
De Marne        Groningen       53.360.934      6.380.521       240.33  10205
30500   145000  0
Marum   Groningen       53.146.252      6.267.326       64.89   10350   34200
215000  0
Medemblik       Noord-Holland   52.767.447      5.106.918       257.56  43331
35600   219000  0
Meerssen        Limburg 50.884.943      5.752.637       27.64   19259   35600
226000  0
Meierijstad     Noord-Brabant   51.613.704      5.548.213       185.52
0
Meppel  Drenthe 52.692.123      6.193.719       57.03   32875   32700   190000
0
Middelburg      Zeeland 51.498.796      3.610.998       53.04   47660   33000
177000  0
Midden-Delfland Zuid-Holland    51.954.201      4.288.310       49.38   18449
42700   280000  0
Midden-Drenthe  Drenthe 52.861.104      6.512.304       345.87  33368   34600
202000  0
Midden-Groningen        Groningen       53.214.825      6.503.424
0
Mill en Sint Hubert     Noord-Brabant   51.679.806      5.755.381       53.17
10843   36300   256000  0
Moerdijk        Noord-Brabant   51.660.156      4.513.072       184.03  36775
36600   215000  0
Molenwaard      Zuid-Holland    51.892.130      4.798.340       126.47  29043
40300   245000  0
Montferland     Gelderland      51.919.617      6.246.311       106.63  34985
34100   208000  0
Montfoort       Utrecht 52.036.213      4.951.859       38.2    13639   41200
264000  0
Mook en Middelaar       Limburg 51.751.230      5.881.630       18.81   7783
40400   276000  0
Neder-Betuwe    Gelderland      51.930.937      5.574.103       68.16   22591
37300   220000  0
Nederweert      Limburg 51.286.563      5.752.701       101.78  16773   36100
237000  0
Neerijnen       Gelderland      51.831.902      5.279.356       72.9    12022
39900   271000  0
Nieuwegein      Utrecht 52.024.821      5.091.819       25.65   61017   35200
206000  0
Nieuwkoop       Zuid-Holland    52.150.056      4.777.359       91.16   27114
38800   271000  0
Nijkerk Gelderland      52.222.484      5.483.563       72.1    40600   37800
```

```
263000  0
Nijmegen        Gelderland      51.812.563      5.837.226       57.6    168499
30900   191000  1
Nissewaard      Zuid-Holland    51.848.034      4.328.748       98.74   85083
0
Noord-Beveland  Zeeland 51.570.283      3.771.484       121.51  7508    32900
199000  0
Noordenveld     Drenthe 53.095.093      6.445.541       205.32  31110   34800
215000  0
Noordoostpolder Flevoland       52.692.622      5.737.842       595.43  46372
32900   176000  0
Noordwijk       Zuid-Holland    52.239.989      4.450.009       51.45   25689
38300   328000  0
Noordwijkerhout Zuid-Holland    52.260.194      4.495.584       23.42   15982
36600   266000  0
Nuenen, Gerwen en Nederwetten   Noord-Brabant   51.478.251      5.556.858
33.94   22596   41000   298000  0
Nunspeet        Gelderland      52.374.772      5.769.895       129.53  26707
36600   261000  0
Nuth    Limburg 50.911.427      5.892.596       33.13   15577   35900   198000
0
Oegstgeest      Zuid-Holland    52.186.226      4.474.810       7.97    22855
45500   320000  0
Oirschot        Noord-Brabant   51.504.448      5.308.463       102.84  17989
39800   312000  0
Oisterwijk      Noord-Brabant   51.565.424      5.203.028       65.13   25835
38200   317000  0
Oldambt Groningen       53.203.691      7.052.970       295.96  38558   29300
133000  0
Oldebroek       Gelderland      52.447.445      5.898.317       98.84   22824
35500   235000  0
Oldenzaal       Overijssel      52.311.655      6.926.828       21.95   32152
34000   201000  0
Olst-Wijhe      Overijssel      52.351.896      6.127.076       118.37  17760
35500   227000  0
Ommen   Overijssel      52.506.652      6.435.289       182.01  17370   35000
242000  0
Onderbanken     Limburg 50.969.852      5.965.396       21.24   7882    32900
168000  0
Oost Gelre      Gelderland      51.990.117      6.567.068       110.12  29648
35100   198000  0
Oosterhout      Noord-Brabant   51.641.020      4.861.690       73.09   53733
35800   225000  0
Ooststellingwerf        Friesland       52.988.319      6.270.427       226.11
25672   31400   180000  0
Oostzaan        Noord-Holland   52.440.497      4.875.722       16.08   9154
39300   265000  0
Opmeer  Noord-Holland   52.705.379      4.942.667       41.94   11374   36700
```

```
233000  0
Opsterland      Friesland       53.039.542      6.115.320       227.64  29894
33800   191000  0
Oss     Noord-Brabant   51.761.180      5.514.048       159.87  85039   34900
213000  0
Oud-Beijerland  Zuid-Holland    51.818.930      4.427.236       19.61   23727
39400   227000  0
Oude IJsselstreek       Gelderland      51.911.046      6.405.564       137.94
39614   32500   198000  0
Ouder-Amstel    Noord-Holland   52.285.929      4.913.383       25.78   13255
43200   294000  0
Oudewater       Utrecht 52.024.163      4.868.417       40.1    9868    39600
283000  0
Overbetuwe      Gelderland      51.932.775      5.781.465       115.08  46653
37300   235000  0
Papendrecht     Zuid-Holland    51.842.481      4.694.183       10.79   32140
36500   190000  0
Peel en Maas    Limburg 51.339.919      6.012.298       161.35  43298   36200
229000  0
Pekela  Groningen       53.085.241      6.977.881       50.2    12733   28400
132000  0
Pijnacker-Nootdorp      Zuid-Holland    52.012.698      4.427.355       38.61
51068   42300   268000  0
Purmerend       Noord-Holland   52.514.381      4.964.061       24.56   79552
34400   189000  0
Putten  Gelderland      52.245.301      5.568.482       87.41   24044   37900
292000  0
Raalte  Overijssel      52.380.903      6.278.318       172.29  36526   36000
238000  0
Reimerswaal     Zeeland 51.410.002      4.154.448       242.42  21915   34900
188000  0
Renkum  Gelderland      51.982.983      5.739.978       47.23   31577   38200
253000  0
Renswoude       Utrecht 52.074.661      5.538.173       18.51   4928    39200
281000  0
Reusel-De Mierden       Noord-Brabant   51.370.650      5.146.555       78.64
12728   37900   270000  0
Rheden  Gelderland      52.005.737      6.027.959       84.35   43563   33600
204000  0
Rhenen  Utrecht 51.962.140      5.571.116       43.76   19123   37100   255000
0
Ridderkerk      Zuid-Holland    51.870.253      4.602.234       25.26   45207
34700   194000  0
Rijssen-Holten  Overijssel      52.292.506      6.439.999       94.38   37696
36200   220000  0
Rijswijk        Zuid-Holland    52.037.698      4.321.974       14.49   47680
33100   183000  0
Roerdalen       Limburg 51.139.166      6.034.269       88.7    20808   35000
```

```
198000  0
Roermond        Limburg 51.191.320      5.987.772       71.1    57030   31500
174000  0
De Ronde Venen  Utrecht 52.206.680      4.886.773       116.98  42648   42700
300000  0
Roosendaal      Noord-Brabant   51.535.849      4.465.321       107.16  76930
33900   204000  0
Rotterdam       Zuid-Holland    51.924.420      4.477.733       325.79  618467
29600   148000  1
Rozendaal       Gelderland      52.009.785      5.966.507       27.92   1500
56800   462000  0
Rucphen Noord-Brabant   51.533.567      4.560.320       64.47   22213   35300
245000  0
Schagen Noord-Holland   52.788.091      4.804.400       187.28  46016   35700
222000  0
Scherpenzeel    Gelderland      52.078.930      5.488.065       13.81   9496
37100   265000  0
Schiedam        Zuid-Holland    51.916.960      4.398.819       19.86   76487
31000   142000  0
Schiermonnikoog Friesland       53.489.374      6.230.911       199.07  938
30000   250000  0
Schinnen        Limburg 50.943.647      5.879.361       24.12   12961   35900
193000  0
Schouwen-Duiveland      Zeeland 51.680.357      3.952.013       488.21  33821
35300   229000  0
Simpelveld      Limburg 50.832.849      5.987.771       16.02   10844   33300
161000  0
Sint Anthonis   Noord-Brabant   51.627.161      5.885.193       99.76   11696
38900   263000  0
Sint-Michielsgestel     Noord-Brabant   51.641.348      5.352.881       59.34
28186   40300   295000  0
Sittard-Geleen  Limburg 51.003.168      5.823.671       80.53   93806   31800
157000  0
Sliedrecht      Zuid-Holland    51.824.868      4.773.162       14.01   24528
34600   173000  0
Sluis   Zeeland 51.308.656      3.387.919       307.16  23795   33200   181000
0
Smallingerland  Friesland       53.127.729      6.079.345       126.17  55496
31100   168000  0
Soest   Utrecht 52.176.352      5.299.197       46.43   45430   38700   273000
0
Someren Noord-Brabant   51.384.967      5.712.367       81.5    18700   35700
265000  0
Son en Breugel  Noord-Brabant   51.511.394      5.498.219       26.51   16241
40700   295000  0
Stadskanaal     Groningen       52.991.985      6.946.222       119.94  32793
29200   146000  0
Staphorst       Overijssel      52.642.914      6.199.494       135.69  16365
```

```
37800   248000  0
Stede Broec     Noord-Holland   52.699.602      5.232.325       16.37   21474
34800   181000  0
Steenbergen     Noord-Brabant   51.581.240      4.315.599       159.14  23409
35500   202000  0
Steenwijkerland Overijssel      52.741.538      6.049.324       321.59  43372
32800   196000  0
Stein   Limburg 50.967.974      5.766.220       22.62   25348   34200   180000
0
Stichtse Vecht  Utrecht 52.144.755      5.033.208       106.82  63823   40800
270000  0
Strijen Zuid-Holland    51.744.684      4.553.920       57.7    8678    37400
222000  0
Súdwest-Fryslân Friesland       53.023.416      5.481.658       841.56  84258
32500   175000  0
Terneuzen       Zeeland 51.332.285      3.832.426       317.76  54711   33000
157000  0
Terschelling    Friesland       53.397.875      5.346.679       673.99  4769
33300   298000  0
Texel   Noord-Holland   53.054.763      4.797.715       463.16  13566   32400
258000  0
Teylingen       Zuid-Holland    52.222.004      4.513.404       33.49   35723
41400   285000  0
Tholen  Zeeland 51.533.271      4.216.222       254     25398   34100   185000
0
Tiel    Gelderland      51.887.618      5.427.877       34.81   41729   33600
195000  0
Tilburg Noord-Brabant   51.560.596      5.091.914       119.18  210382  31600
187000  1
Tubbergen       Overijssel      52.408.907      6.785.489       147.44  21216
37700   240000  0
Twenterand      Overijssel      52.450.068      6.620.745       108.14  33939
33700   203000  0
Tynaarlo        Drenthe 53.101.283      6.576.097       147.7   32506   37800
236000  0
Tytsjerksteradiel       Friesland       53.176.038      5.978.680       161.41
31980   33800   192000  0
Uden    Noord-Brabant   51.663.107      5.623.923       67.53   40934   35100
226000  0
Uitgeest        Noord-Holland   52.531.225      4.712.046       22.29   13242
39100   238000  0
Uithoorn        Noord-Holland   52.244.627      4.831.734       19.42   28407
38000   224000  0
Urk     Flevoland       52.663.053      5.598.947       109.91  19487   37900
194000  0
Utrecht Utrecht 52.090.737      5.121.420       99.21   328577  34300   223000
1
Utrechtse Heuvelrug     Utrecht 52.052.203      5.282.495       134.09  47939
```

```
41400   305000  0
Vaals   Limburg 50.773.218      6.011.274       23.87   9682    29100   143000
0
Valkenburg aan de Geul  Limburg 50.868.477      5.826.448       36.92   16668
34000   202000  0
Valkenswaard    Noord-Brabant   51.355.496      5.453.327       56.49   30353
34500   244000  0
Veendam Groningen       53.106.278      6.875.100       78.68   27795   30500
142000  0
Veenendaal      Utrecht 52.026.301      5.554.431       19.72   63207   34800
210000  0
Veere   Zeeland 51.548.294      3.666.006       206.63  21882   36600   261000
0
Veldhoven       Noord-Brabant   51.417.634      5.406.021       31.93   44136
37300   254000  0
Velsen  Noord-Holland   52.452.059      4.630.587       63.05   67231   35300
222000  0
Venlo   Limburg 51.370.375      6.172.403       128.99  100381  31200   177000
0
Venray  Limburg 51.525.626      5.973.699       165     43120   35100   213000
0
Vianen  Utrecht 51.990.276      5.103.033       42.39   19573   37400   219000
0
Vlaardingen     Zuid-Holland    51.912.067      4.349.437       26.69   71059
31700   164000  0
Vlieland        Friesland       53.250.184      4.951.427       315.8   1097
30600   243000  0
Vlissingen      Zeeland 51.453.667      3.570.912       344.83  44450   30200
153000  0
Voerendaal      Limburg 50.879.153      5.931.160       31.52   12447   36800
206000  0
Voorschoten     Zuid-Holland    52.123.790      4.438.598       11.56   24941
42300   283000  0
Voorst  Gelderland      52.225.271      6.099.050       126.47  23801   37300
261000  0
Vught   Noord-Brabant   51.653.306      5.294.347       34.44   25626   40300
324000  0
Waadhoeke       Friesland       53.184.738      5.531.866
0
Waalre  Noord-Brabant   51.387.833      5.443.202       22.66   16804   43000
302000  0
Waalwijk        Noord-Brabant   51.687.895      5.057.482       67.65   46515
34400   218000  0
Waddinxveen     Zuid-Holland    52.031.297      4.659.132       29.4    25520
37700   207000  0
Wageningen      Gelderland      51.969.187      5.665.395       32.36   37511
33700   244000  1
Wassenaar       Zuid-Holland    52.142.910      4.401.213       62.37   25786
```

```
52600   483000  0
Waterland       Noord-Holland   52.447.050      5.015.475       115.66  17154
41400   269000  0
Weert   Limburg 51.243.941      5.714.222       105.52  48727   34400   211000
0
Weesp   Noord-Holland   52.308.051      5.040.622       21.83   18231   35500
222000  0
Werkendam       Noord-Brabant   51.808.165      4.891.848       121.76  26400
38300   238000  0
West Maas en Waal       Gelderland      51.861.341      5.501.737       85.21
18391   36400   245000  0
Westerveld      Drenthe 52.857.708      6.296.010       282.74  18902   34900
253000  0
Westervoort     Gelderland      51.963.807      5.968.984       7.84    15147
33700   182000  0
Westerwolde     Groningen       53.027.025      7.110.788       280.63
0
Westland        Zuid-Holland    51.999.176      4.217.457       90.58   103335
37100   241000  0
Weststellingwerf        Friesland       52.871.892      6.022.919       228.45
25487   32000   182000  0
Westvoorne      Zuid-Holland    51.887.279      4.084.461       97.48   13977
42800   303000  0
Wierden Overijssel      52.358.260      6.593.873       95.39   23906   38400
235000  0
Wijchen Gelderland      51.813.674      5.752.895       69.56   41010   35800
227000  0
Wijdemeren      Noord-Holland   52.219.688      5.093.666       76.36   23170
41600   331000  0
Wijk bij Duurstede      Utrecht 51.975.600      5.338.450       50.25   23031
39500   261000  0
Winsum  Groningen       53.330.810      6.522.254       102.53  13843   33300
167000  0
Winterswijk     Gelderland      51.971.314      6.720.509       138.81  28873
32000   180000  0
Woensdrecht     Noord-Brabant   51.429.248      4.304.708       91.91   21654
35700   224000  0
Woerden Utrecht 52.079.829      4.862.724       92.92   50607   39900   251000
0
De Wolden       Drenthe 52.681.979      6.366.039       226.35  23592   36300
252000  0
Wormerland      Noord-Holland   52.507.275      4.852.800       45.18   15751
37500   217000  0
Woudenberg      Utrecht 52.082.175      5.426.595       36.82   12386   38800
281000  0
Woudrichem      Noord-Brabant   51.817.007      5.003.751       51.7    14407
38200   251000  0
Zaanstad        Noord-Holland   52.457.966      4.751.043       83.24   150911
```

```
33300   177000  0
Zaltbommel      Gelderland      51.813.554      5.250.773       89.04   27207
38000   233000  0
Zandvoort       Noord-Holland   52.371.149      4.533.355       43.97   16588
33700   255000  0
Zederik Zuid-Holland    51.925.558      5.013.633       76.5    13661   39200
256000  0
Zeewolde        Flevoland       52.331.128      5.540.495       268.86  21543
39200   235000  0
Zeist   Utrecht 52.090.601      5.233.253       48.65   61337   39700   301000
0
Zevenaar        Gelderland      51.929.445      6.076.959       58      32254
33400   186000  0
Zoetermeer      Zuid-Holland    52.060.669      4.494.025       37.05   123614
36000   188000  0
Zoeterwoude     Zuid-Holland    52.120.830      4.516.570       21.96   8089
40900   261000  0
Zuidhorn        Groningen       53.243.148      6.408.104       128.37  18767
35100   185000  0
Zuidplas        Zuid-Holland    52.006.758      4.582.785       62.42   40878
40700   235000  0
Zundert Noord-Brabant   51.470.701      4.662.356       121.17  21374   35800
289000  0
Zutphen Gelderland      52.142.736      6.196.058       42.93   47154   32100
185000  0
Zwartewaterland Overijssel      52.601.512      6.059.625       87.86   22148
35600   197000  0
Zwijndrecht     Zuid-Holland    51.810.598      4.627.272       22.77   44546
34400   169000  0
Zwolle  Overijssel      52.516.775      6.083.022       119.36  123211  33600
200000  0
```

In this form (as one long string) the content of the CSV file is of course not of too much use, as it is difficult to access individual elements from it. Instead of reading the content file completely, we could read it line by line (getting a list of lines), and then split the lines at the separator to create a list or dictionary of the elements in each row of the table, resulting in big list of lists or list of dictionaries. Luckily, however, CSV files are so common that there is a package called csv that provides this and other frequently needed functionality for working with CSV files (please refer to the online documentation at https://docs.python.org/3/library/csv.html for full reference). Here are some examples of what working with the package can look like:

[15]:
```python
# import the csv library
import csv

# csv.reader returns the content of the file as list of lists of strings
with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.reader(csvfile, delimiter='\t')
    for row in csvreader:
        print(row[0])
```

```
municipality
Aa en Hunze
Aalburg
Aalsmeer
Aalten
Achtkarspelen
Alblasserdam
Albrandswaard
Alkmaar
Almelo
Almere
Alphen aan den Rijn
Alphen-Chaam
Ameland
Amersfoort
Amstelveen
Amsterdam
Apeldoorn
Appingedam
Arnhem
Assen
Asten
Baarle-Nassau
Baarn
Barendrecht
Barneveld
Bedum
Beek
Beemster
Beesel
Berg en Dal
Bergeijk
Bergen (L.)
Bergen (NH.)
Bergen op Zoom
Berkelland
Bernheze
Best
Beuningen
Beverwijk
De Bilt
Binnenmaas
Bladel
Blaricum
```

```
Bloemendaal
Bodegraven-Reeuwijk
Boekel
Ten Boer
Borger-Odoorn
Borne
Borsele
Boxmeer
Boxtel
Breda
Brielle
Bronckhorst
Brummen
Brunssum
Bunnik
Bunschoten
Buren
Capelle aan den IJssel
Castricum
Coevorden
Cranendonck
Cromstrijen
Cuijk
Culemborg
Dalfsen
Dantumadiel
Delft
Delfzijl
Deurne
Deventer
Diemen
Dinkelland
Doesburg
Doetinchem
Dongen
Dongeradeel
Dordrecht
Drechterland
Drimmelen
Dronten
Druten
Duiven
Echt-Susteren
Edam-Volendam
Ede
Eemnes
Eemsmond
Eersel
```

```
Eijsden-Margraten
Eindhoven
Elburg
Emmen
Enkhuizen
Enschede
Epe
Ermelo
Etten-Leur
Ferwerderadiel
De Fryske Marren
Geertruidenberg
Geldermalsen
Geldrop-Mierlo
Gemert-Bakel
Gennep
Giessenlanden
Gilze en Rijen
Goeree-Overflakkee
Goes
Goirle
Gooise Meren
Gorinchem
Gouda
Grave
's-Gravenhage
Groningen
Grootegast
Gulpen-Wittem
Haaksbergen
Haaren
Haarlem
Haarlemmerliede en Spaarnwoude
Haarlemmermeer
Halderberge
Hardenberg
Harderwijk
Hardinxveld-Giessendam
Haren
Harlingen
Hattem
Heemskerk
Heemstede
Heerde
Heerenveen
Heerhugowaard
Heerlen
Heeze-Leende
```

```
Heiloo
Den Helder
Hellendoorn
Hellevoetsluis
Helmond
Hendrik-Ido-Ambacht
Hengelo
's-Hertogenbosch
Heumen
Heusden
Hillegom
Hilvarenbeek
Hilversum
Hof van Twente
Hollands Kroon
Hoogeveen
Hoorn
Horst aan de Maas
Houten
Huizen
Hulst
IJsselstein
Kaag en Braassem
Kampen
Kapelle
Katwijk
Kerkrade
Koggenland
Kollumerland en Nieuwkruisland
Korendijk
Krimpen aan den IJssel
Krimpenerwaard
Laarbeek
Landerd
Landgraaf
Landsmeer
Langedijk
Lansingerland
Laren
Leek
Leerdam
Leeuwarden
Leiden
Leiderdorp
Leidschendam-Voorburg
Lelystad
Leudal
Leusden
```

```
Lingewaal
Lingewaard
Lisse
Lochem
Loon op Zand
Lopik
Loppersum
Losser
Maasdriel
Maasgouw
Maassluis
Maastricht
De Marne
Marum
Medemblik
Meerssen
Meierijstad
Meppel
Middelburg
Midden-Delfland
Midden-Drenthe
Midden-Groningen
Mill en Sint Hubert
Moerdijk
Molenwaard
Montferland
Montfoort
Mook en Middelaar
Neder-Betuwe
Nederweert
Neerijnen
Nieuwegein
Nieuwkoop
Nijkerk
Nijmegen
Nissewaard
Noord-Beveland
Noordenveld
Noordoostpolder
Noordwijk
Noordwijkerhout
Nuenen, Gerwen en Nederwetten
Nunspeet
Nuth
Oegstgeest
Oirschot
Oisterwijk
Oldambt
```

```
Oldebroek
Oldenzaal
Olst-Wijhe
Ommen
Onderbanken
Oost Gelre
Oosterhout
Ooststellingwerf
Oostzaan
Opmeer
Opsterland
Oss
Oud-Beijerland
Oude IJsselstreek
Ouder-Amstel
Oudewater
Overbetuwe
Papendrecht
Peel en Maas
Pekela
Pijnacker-Nootdorp
Purmerend
Putten
Raalte
Reimerswaal
Renkum
Renswoude
Reusel-De Mierden
Rheden
Rhenen
Ridderkerk
Rijssen-Holten
Rijswijk
Roerdalen
Roermond
De Ronde Venen
Roosendaal
Rotterdam
Rozendaal
Rucphen
Schagen
Scherpenzeel
Schiedam
Schiermonnikoog
Schinnen
Schouwen-Duiveland
Simpelveld
Sint Anthonis
```

Sint-Michielsgestel
Sittard-Geleen
Sliedrecht
Sluis
Smallingerland
Soest
Someren
Son en Breugel
Stadskanaal
Staphorst
Stede Broec
Steenbergen
Steenwijkerland
Stein
Stichtse Vecht
Strijen
Súdwest-Fryslân
Terneuzen
Terschelling
Texel
Teylingen
Tholen
Tiel
Tilburg
Tubbergen
Twenterand
Tynaarlo
Tytsjerksteradiel
Uden
Uitgeest
Uithoorn
Urk
Utrecht
Utrechtse Heuvelrug
Vaals
Valkenburg aan de Geul
Valkenswaard
Veendam
Veenendaal
Veere
Veldhoven
Velsen
Venlo
Venray
Vianen
Vlaardingen
Vlieland
Vlissingen

```
Voerendaal
Voorschoten
Voorst
Vught
Waadhoeke
Waalre
Waalwijk
Waddinxveen
Wageningen
Wassenaar
Waterland
Weert
Weesp
Werkendam
West Maas en Waal
Westerveld
Westervoort
Westerwolde
Westland
Weststellingwerf
Westvoorne
Wierden
Wijchen
Wijdemeren
Wijk bij Duurstede
Winsum
Winterswijk
Woensdrecht
Woerden
De Wolden
Wormerland
Woudenberg
Woudrichem
Zaanstad
Zaltbommel
Zandvoort
Zederik
Zeewolde
Zeist
Zevenaar
Zoetermeer
Zoeterwoude
Zuidhorn
Zuidplas
Zundert
Zutphen
Zwartewaterland
Zwijndrecht
```

```
Zwolle
```

[16]:
```
# csv.DictReader returns the content of the file as list of dictionaries, using␣
↪the first row of the CSV file as keys
with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile, delimiter='\t')
    for row in csvreader:
        print(f'{row["municipality"]}: {row["university"]}')
```

```
Aa en Hunze: 0
Aalburg: 0
Aalsmeer: 0
Aalten: 0
Achtkarspelen: 0
Alblasserdam: 0
Albrandswaard: 0
Alkmaar: 0
Almelo: 0
Almere: 0
Alphen aan den Rijn: 0
Alphen-Chaam: 0
Ameland: 0
Amersfoort: 0
Amstelveen: 0
Amsterdam: 2
Apeldoorn: 0
Appingedam: 0
Arnhem: 0
Assen: 0
Asten: 0
Baarle-Nassau: 0
Baarn: 0
Barendrecht: 0
Barneveld: 0
Bedum: 0
Beek: 0
Beemster: 0
Beesel: 0
Berg en Dal: 0
Bergeijk: 0
Bergen (L.): 0
Bergen (NH.): 0
Bergen op Zoom: 0
Berkelland: 0
Bernheze: 0
Best: 0
Beuningen: 0
Beverwijk: 0
```

```
De Bilt: 0
Binnenmaas: 0
Bladel: 0
Blaricum: 0
Bloemendaal: 0
Bodegraven-Reeuwijk: 0
Boekel: 0
Ten Boer: 0
Borger-Odoorn: 0
Borne: 0
Borsele: 0
Boxmeer: 0
Boxtel: 0
Breda: 0
Brielle: 0
Bronckhorst: 0
Brummen: 0
Brunssum: 0
Bunnik: 0
Bunschoten: 0
Buren: 0
Capelle aan den IJssel: 0
Castricum: 0
Coevorden: 0
Cranendonck: 0
Cromstrijen: 0
Cuijk: 0
Culemborg: 0
Dalfsen: 0
Dantumadiel: 0
Delft: 1
Delfzijl: 0
Deurne: 0
Deventer: 0
Diemen: 0
Dinkelland: 0
Doesburg: 0
Doetinchem: 0
Dongen: 0
Dongeradeel: 0
Dordrecht: 0
Drechterland: 0
Drimmelen: 0
Dronten: 0
Druten: 0
Duiven: 0
Echt-Susteren: 0
Edam-Volendam: 0
```

```
Ede: 0
Eemnes: 0
Eemsmond: 0
Eersel: 0
Eijsden-Margraten: 0
Eindhoven: 1
Elburg: 0
Emmen: 0
Enkhuizen: 0
Enschede: 1
Epe: 0
Ermelo: 0
Etten-Leur: 0
Ferwerderadiel: 0
De Fryske Marren: 0
Geertruidenberg: 0
Geldermalsen: 0
Geldrop-Mierlo: 0
Gemert-Bakel: 0
Gennep: 0
Giessenlanden: 0
Gilze en Rijen: 0
Goeree-Overflakkee: 0
Goes: 0
Goirle: 0
Gooise Meren: 0
Gorinchem: 0
Gouda: 0
Grave: 0
's-Gravenhage: 0
Groningen: 1
Grootegast: 0
Gulpen-Wittem: 0
Haaksbergen: 0
Haaren: 0
Haarlem: 0
Haarlemmerliede en Spaarnwoude: 0
Haarlemmermeer: 0
Halderberge: 0
Hardenberg: 0
Harderwijk: 0
Hardinxveld-Giessendam: 0
Haren: 0
Harlingen: 0
Hattem: 0
Heemskerk: 0
Heemstede: 0
Heerde: 0
```

```
Heerenveen: 0
Heerhugowaard: 0
Heerlen: 0
Heeze-Leende: 0
Heiloo: 0
Den Helder: 0
Hellendoorn: 0
Hellevoetsluis: 0
Helmond: 0
Hendrik-Ido-Ambacht: 0
Hengelo: 0
's-Hertogenbosch: 0
Heumen: 0
Heusden: 0
Hillegom: 0
Hilvarenbeek: 0
Hilversum: 0
Hof van Twente: 0
Hollands Kroon: 0
Hoogeveen: 0
Hoorn: 0
Horst aan de Maas: 0
Houten: 0
Huizen: 0
Hulst: 0
IJsselstein: 0
Kaag en Braassem: 0
Kampen: 0
Kapelle: 0
Katwijk: 0
Kerkrade: 0
Koggenland: 0
Kollumerland en Nieuwkruisland: 0
Korendijk: 0
Krimpen aan den IJssel: 0
Krimpenerwaard: 0
Laarbeek: 0
Landerd: 0
Landgraaf: 0
Landsmeer: 0
Langedijk: 0
Lansingerland: 0
Laren: 0
Leek: 0
Leerdam: 0
Leeuwarden: 0
Leiden: 1
Leiderdorp: 0
```

```
Leidschendam-Voorburg: 0
Lelystad: 0
Leudal: 0
Leusden: 0
Lingewaal: 0
Lingewaard: 0
Lisse: 0
Lochem: 0
Loon op Zand: 0
Lopik: 0
Loppersum: 0
Losser: 0
Maasdriel: 0
Maasgouw: 0
Maassluis: 0
Maastricht: 1
De Marne: 0
Marum: 0
Medemblik: 0
Meerssen: 0
Meierijstad: 0
Meppel: 0
Middelburg: 0
Midden-Delfland: 0
Midden-Drenthe: 0
Midden-Groningen: 0
Mill en Sint Hubert: 0
Moerdijk: 0
Molenwaard: 0
Montferland: 0
Montfoort: 0
Mook en Middelaar: 0
Neder-Betuwe: 0
Nederweert: 0
Neerijnen: 0
Nieuwegein: 0
Nieuwkoop: 0
Nijkerk: 0
Nijmegen: 1
Nissewaard: 0
Noord-Beveland: 0
Noordenveld: 0
Noordoostpolder: 0
Noordwijk: 0
Noordwijkerhout: 0
Nuenen, Gerwen en Nederwetten: 0
Nunspeet: 0
Nuth: 0
```

```
Oegstgeest: 0
Oirschot: 0
Oisterwijk: 0
Oldambt: 0
Oldebroek: 0
Oldenzaal: 0
Olst-Wijhe: 0
Ommen: 0
Onderbanken: 0
Oost Gelre: 0
Oosterhout: 0
Ooststellingwerf: 0
Oostzaan: 0
Opmeer: 0
Opsterland: 0
Oss: 0
Oud-Beijerland: 0
Oude IJsselstreek: 0
Ouder-Amstel: 0
Oudewater: 0
Overbetuwe: 0
Papendrecht: 0
Peel en Maas: 0
Pekela: 0
Pijnacker-Nootdorp: 0
Purmerend: 0
Putten: 0
Raalte: 0
Reimerswaal: 0
Renkum: 0
Renswoude: 0
Reusel-De Mierden: 0
Rheden: 0
Rhenen: 0
Ridderkerk: 0
Rijssen-Holten: 0
Rijswijk: 0
Roerdalen: 0
Roermond: 0
De Ronde Venen: 0
Roosendaal: 0
Rotterdam: 1
Rozendaal: 0
Rucphen: 0
Schagen: 0
Scherpenzeel: 0
Schiedam: 0
Schiermonnikoog: 0
```

```
Schinnen: 0
Schouwen-Duiveland: 0
Simpelveld: 0
Sint Anthonis: 0
Sint-Michielsgestel: 0
Sittard-Geleen: 0
Sliedrecht: 0
Sluis: 0
Smallingerland: 0
Soest: 0
Someren: 0
Son en Breugel: 0
Stadskanaal: 0
Staphorst: 0
Stede Broec: 0
Steenbergen: 0
Steenwijkerland: 0
Stein: 0
Stichtse Vecht: 0
Strijen: 0
Súdwest-Fryslân: 0
Terneuzen: 0
Terschelling: 0
Texel: 0
Teylingen: 0
Tholen: 0
Tiel: 0
Tilburg: 1
Tubbergen: 0
Twenterand: 0
Tynaarlo: 0
Tytsjerksteradiel: 0
Uden: 0
Uitgeest: 0
Uithoorn: 0
Urk: 0
Utrecht: 1
Utrechtse Heuvelrug: 0
Vaals: 0
Valkenburg aan de Geul: 0
Valkenswaard: 0
Veendam: 0
Veenendaal: 0
Veere: 0
Veldhoven: 0
Velsen: 0
Venlo: 0
Venray: 0
```

```
Vianen: 0
Vlaardingen: 0
Vlieland: 0
Vlissingen: 0
Voerendaal: 0
Voorschoten: 0
Voorst: 0
Vught: 0
Waadhoeke: 0
Waalre: 0
Waalwijk: 0
Waddinxveen: 0
Wageningen: 1
Wassenaar: 0
Waterland: 0
Weert: 0
Weesp: 0
Werkendam: 0
West Maas en Waal: 0
Westerveld: 0
Westervoort: 0
Westerwolde: 0
Westland: 0
Weststellingwerf: 0
Westvoorne: 0
Wierden: 0
Wijchen: 0
Wijdemeren: 0
Wijk bij Duurstede: 0
Winsum: 0
Winterswijk: 0
Woensdrecht: 0
Woerden: 0
De Wolden: 0
Wormerland: 0
Woudenberg: 0
Woudrichem: 0
Zaanstad: 0
Zaltbommel: 0
Zandvoort: 0
Zederik: 0
Zeewolde: 0
Zeist: 0
Zevenaar: 0
Zoetermeer: 0
Zoeterwoude: 0
Zuidhorn: 0
Zuidplas: 0
```

```
Zundert: 0
Zutphen: 0
Zwartewaterland: 0
Zwijndrecht: 0
Zwolle: 0
```

[17]:
```
# same as the previous example, but printing only municipalitiers with at least␣
↪one university
with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile, delimiter='\t')
    for row in csvreader:
        if int(row["university"]) != 0:
            print(f'{row["municipality"]}: {row["university"]}')
```

```
Amsterdam: 2
Delft: 1
Eindhoven: 1
Enschede: 1
Groningen: 1
Leiden: 1
Maastricht: 1
Nijmegen: 1
Rotterdam: 1
Tilburg: 1
Utrecht: 1
Wageningen: 1
```

Challenge!

Write a code to print only the municipalities with an average household income above 40000

[18]:
```
with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile, delimiter='\t')
    for row in csvreader:
        if int(row["avg_household_income_2012"]) > 40000:
            print(f'{row["municipality"]}: {row["province"]}')
```

```
Aalsmeer: Noord-Holland
Albrandswaard: Zuid-Holland
Amstelveen: Noord-Holland
Barendrecht: Zuid-Holland
Beemster: Noord-Holland
```

```
        ␣
↪--------------------------------------------------------------------------

       ValueError                                Traceback (most recent call␣
↪last)

       /tmp/ipykernel_10270/2531272763.py in <module>
         2     csvreader = csv.DictReader(csvfile, delimiter='\t')
         3     for row in csvreader:
   ----> 4         if int(row["avg_household_income_2012"]) > 40000:
         5             print(f'{row["municipality"]}: {row["province"]}')


       ValueError: invalid literal for int() with base 10: ''
```

[19]:
```
with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile, delimiter='\t')
    for row in csvreader:
        income = row["avg_household_income_2012"]
        if income != "" and int(row["avg_household_income_2012"]) > 40000:
            print(f'{row["municipality"]}: {row["province"]}')
```

```
Aalsmeer: Noord-Holland
Albrandswaard: Zuid-Holland
Amstelveen: Noord-Holland
Barendrecht: Zuid-Holland
Beemster: Noord-Holland
De Bilt: Utrecht
Blaricum: Noord-Holland
Bloemendaal: Noord-Holland
Bodegraven-Reeuwijk: Zuid-Holland
Bunnik: Utrecht
Castricum: Noord-Holland
Edam-Volendam: Noord-Holland
Eemnes: Utrecht
Giessenlanden: Zuid-Holland
Haaren: Noord-Brabant
Haarlemmerliede en Spaarnwoude: Noord-Holland
Haren: Groningen
Heemstede: Noord-Holland
Heeze-Leende: Noord-Brabant
Heiloo: Noord-Holland
Houten: Utrecht
Landsmeer: Noord-Holland
Lansingerland: Zuid-Holland
Laren: Noord-Holland
Leusden: Utrecht
```

```
Midden-Delfland: Zuid-Holland
Molenwaard: Zuid-Holland
Montfoort: Utrecht
Mook en Middelaar: Limburg
Nuenen, Gerwen en Nederwetten: Noord-Brabant
Oegstgeest: Zuid-Holland
Ouder-Amstel: Noord-Holland
Pijnacker-Nootdorp: Zuid-Holland
De Ronde Venen: Utrecht
Rozendaal: Gelderland
Sint-Michielsgestel: Noord-Brabant
Son en Breugel: Noord-Brabant
Stichtse Vecht: Utrecht
Teylingen: Zuid-Holland
Utrechtse Heuvelrug: Utrecht
Voorschoten: Zuid-Holland
Vught: Noord-Brabant
Waalre: Noord-Brabant
Wassenaar: Zuid-Holland
Waterland: Noord-Holland
Westvoorne: Zuid-Holland
Wijdemeren: Noord-Holland
Zoeterwoude: Zuid-Holland
Zuidplas: Zuid-Holland
```

1.2.1 Pandas

If you want to do more advanced things with the data from CSV files, like for example merge, join, or concatenate tables from different CSV files, you can absolutely do that with CSV files read in as above and the knowledge about loops, conditions, list, dictionaries etc. that you have, but it can be a bit tricky. This is why when such operations are (likely to be) needed, it is usually recommended to use the pandas library (http://pandas.pydata.org/), which has some specialized functions for this.

Pandas has an own function for reading CSV files, which returns the result as a so-called data frame, as shown in the following example:

[20]:
```
import pandas as pd

df = pd.read_csv('data/dutch_municipalities.csv', sep="\t")
print(df)
```

```
       municipality       province   latitude  longitude  surface_km2  \
0        Aa en Hunze        Drenthe  53.010.485  6.749.528       278.90
1            Aalburg  Noord-Brabant  51.751.294  5.057.085        53.17
2           Aalsmeer  Noord-Holland  52.262.164  4.761.922        32.29
3             Aalten     Gelderland  51.926.667  6.580.678        96.57
4       Achtkarspelen      Friesland  53.210.357  6.153.565       103.98
..               …              …         …         …           …
```

```
375          Zundert  Noord-Brabant  51.470.701  4.662.356       121.17
376          Zutphen     Gelderland  52.142.736  6.196.058        42.93
377  Zwartewaterland      Overijssel  52.601.512  6.059.625        87.86
378      Zwijndrecht   Zuid-Holland  51.810.598  4.627.272        22.77
379           Zwolle      Overijssel  52.516.775  6.083.022       119.36

     population  avg_household_income_2012  avg_woz_2014  university
0       25243.0                    35500.0      225000.0           0
1       12859.0                    39100.0      249000.0           0
2       30792.0                    40900.0      276000.0           0
3       27030.0                    33300.0      194000.0           0
4       28002.0                    30500.0      165000.0           0
..          …                          …             …          …
375     21374.0                    35800.0      289000.0           0
376     47154.0                    32100.0      185000.0           0
377     22148.0                    35600.0      197000.0           0
378     44546.0                    34400.0      169000.0           0
379    123211.0                    33600.0      200000.0           0

[380 rows x 9 columns]
```

Data frames are two-dimensional labeled data structures, very much like tables. The rows are labeled by an index (typically ascending from 0), and the columns are labeled by the column names, corresponding to the kind of data that is contained in them. See https://pandas.pydata.org/pandas-docs/stable/dsintro.html#dataframe for further details.

Souce: https://www.w3resource.com/

The `head()` method return the first `n` rows (default = 5) of a data frame. It is useful for quickly testing if your object has the right type of data in it.

[21]:
```
df.head()
```

[21]:
```
    municipality       province    latitude  longitude  surface_km2  \
0    Aa en Hunze        Drenthe  53.010.485  6.749.528       278.90
1        Aalburg  Noord-Brabant  51.751.294  5.057.085        53.17
2       Aalsmeer  Noord-Holland  52.262.164  4.761.922        32.29
3         Aalten     Gelderland  51.926.667  6.580.678        96.57
4  Achtkarspelen      Friesland  53.210.357  6.153.565       103.98

   population  avg_household_income_2012  avg_woz_2014  university
0     25243.0                    35500.0      225000.0           0
1     12859.0                    39100.0      249000.0           0
2     30792.0                    40900.0      276000.0           0
3     27030.0                    33300.0      194000.0           0
4     28002.0                    30500.0      165000.0           0
```

Data frames have a number of attributes, such as the column labels, the row indices and the types of the data in the columns (see a full list at https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html), that can be accessed as illustrated below:

[22]:
```
print(df.index)
print("----------")
print(df.columns)
print("----------")
print(df.dtypes)
```

```
RangeIndex(start=0, stop=380, step=1)
----------
Index(['municipality', 'province', 'latitude', 'longitude', 'surface_km2',
       'population', 'avg_household_income_2012', 'avg_woz_2014',
       'university'],
      dtype='object')
----------
municipality                  object
province                      object
latitude                      object
longitude                     object
surface_km2                  float64
population                   float64
avg_household_income_2012    float64
avg_woz_2014                 float64
university                     int64
dtype: object
```

The `info()` method prints a concise summary of a DataFrame:

[23]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 380 entries, 0 to 379
Data columns (total 9 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   municipality               380 non-null    object
 1   province                   380 non-null    object
 2   latitude                   380 non-null    object
 3   longitude                  380 non-null    object
 4   surface_km2                378 non-null    float64
 5   population                 376 non-null    float64
 6   avg_household_income_2012  371 non-null    float64
 7   avg_woz_2014               371 non-null    float64
 8   university                 380 non-null    int64
dtypes: float64(4), int64(1), object(4)
memory usage: 26.8+ KB
None
```

Via the `iloc` attribute we can access a row by its index, for example:

[24]:
```
print(df.iloc[39])
print("----------")
print(type(df.iloc[39]))
```

```
municipality                     De Bilt
province                         Utrecht
latitude                      52.109.272
longitude                      5.180.968
surface_km2                        67.13
population                       42013.0
avg_household_income_2012        43100.0
avg_woz_2014                    338000.0
university                             0
Name: 39, dtype: object
----------
<class 'pandas.core.series.Series'>
```

Apparently, such single row of a data frame is of type "Series" (see https://pandas.pydata.org/pandas-docs/stable/reference/series.html for full reference), which basically means a one-dimensional labeled data structure. Series are iterable. You have maybe already noticed that many functions in, e.g., pandas and matplotlib take Series as input, and this is one way to get them.

Slicing works with `iloc`, too, so a range of indices can be used to access several rows at a time. The result is of type "DataFrame" again:

[25]:
```
print(df.iloc[39:42])
print("----------")
print(type(df.iloc[39:42]))
```

```
   municipality       province    latitude  longitude  surface_km2  \
39      De Bilt        Utrecht  52.109.272  5.180.968        67.13
40   Binnenmaas   Zuid-Holland  51.796.188  4.548.157        75.57
41       Bladel  Noord-Brabant  51.362.963  5.213.639        75.62

    population  avg_household_income_2012  avg_woz_2014  university
39     42013.0                    43100.0      338000.0           0
40     28682.0                    39300.0      231000.0           0
41     19825.0                    37200.0      272000.0           0
----------
<class 'pandas.core.frame.DataFrame'>
```

Similarly, a list of indices (not necessarily a range) can be used:

[26]:
```
print(df.iloc[[38,40,42]])
print("----------")
print(type(df.iloc[[38,40,42]]))
```

```
   municipality       province    latitude  longitude  surface_km2  \
38    Beverwijk  Noord-Holland  52.486.984  4.657.447        20.09
40   Binnenmaas   Zuid-Holland  51.796.188  4.548.157        75.57
42     Blaricum  Noord-Holland  52.272.669  5.248.080        15.56

    population  avg_household_income_2012  avg_woz_2014  university
38     40052.0                    32500.0      178000.0           0
40     28682.0                    39300.0      231000.0           0
42      9112.0                    51600.0      536000.0           0
----------
<class 'pandas.core.frame.DataFrame'>
```

The `iloc` access can also be used for indexing at both axes of the data frame, including accessing a single element (note the different resulting data types):

[27]:
```python
print(df.iloc[1:3,1:3])
print("----------")
print(type(df.iloc[1:3,1:3]))
print("----------")
print(df.iloc[3,3])
print("----------")
print(type(df.iloc[3,3]))
```

```
        province    latitude
1  Noord-Brabant  51.751.294
2  Noord-Holland  52.262.164
----------
<class 'pandas.core.frame.DataFrame'>
----------
6.580.678
----------
<class 'str'>
```

Very similar to `iloc`, the `loc` attribute can be used to access (groups of) rows and columns by their labels. For example (note the difference in the interpretation of the range now that the labels of the indexes are used):

[28]:
```python
print(df.loc[1:3,"population"])
print("----------")
print(type(df.loc[1:3,"population"]))
```

```
1    12859.0
2    30792.0
3    27030.0
Name: population, dtype: float64
----------
<class 'pandas.core.series.Series'>
```

Without using any attributes, just in pairs of square brackets, columns in a dataframe can be addressed by their name. For example, to access the "murders_2014" column of our example data frame, it's name can be used as reference:

```
[29]: print(df["population"])
      print("----------")
      print(type(df["population"]))
```

```
0        25243.0
1        12859.0
2        30792.0
3        27030.0
4        28002.0
          …
375      21374.0
376      47154.0
377      22148.0
378      44546.0
379     123211.0
Name: population, Length: 380, dtype: float64
----------
<class 'pandas.core.series.Series'>
```

Again, the output is a Series, so this is another way to get this data structure.

Accessing several columns at once is also possible, the result is a data frame:

Challenge! (small)

What is the diference between `df[39]` and `df.iloc[39]`?

```
[30]: df.iloc[39]
      df[39]
```

```
      ␣
 ↪---------------------------------------------------------------------------


        KeyError                                  Traceback (most recent call␣
 ↪last)

        ~/anaconda3/envs/new/lib/python3.8/site-packages/pandas/core/indexes/
 ↪base.py in get_loc(self, key, method, tolerance)
       3360             try:
    -> 3361                 return self._engine.get_loc(casted_key)
       3362             except KeyError as err:
```

```
      ~/anaconda3/envs/new/lib/python3.8/site-packages/pandas/_libs/index.pyx␣
↪in pandas._libs.index.IndexEngine.get_loc()


      ~/anaconda3/envs/new/lib/python3.8/site-packages/pandas/_libs/index.pyx␣
↪in pandas._libs.index.IndexEngine.get_loc()


      pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()


      pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()


      KeyError: 39


   The above exception was the direct cause of the following exception:


      KeyError                                  Traceback (most recent call␣
↪last)

      /tmp/ipykernel_10270/2124712428.py in <module>
        1 df.iloc[39]
   ----> 2 df[39]


      ~/anaconda3/envs/new/lib/python3.8/site-packages/pandas/core/frame.py in␣
↪__getitem__(self, key)
     3456             if self.columns.nlevels > 1:
     3457                 return self._getitem_multilevel(key)
   -> 3458             indexer = self.columns.get_loc(key)
     3459             if is_integer(indexer):
     3460                 indexer = [indexer]


      ~/anaconda3/envs/new/lib/python3.8/site-packages/pandas/core/indexes/
↪base.py in get_loc(self, key, method, tolerance)
     3361                 return self._engine.get_loc(casted_key)
     3362             except KeyError as err:
   -> 3363                 raise KeyError(key) from err
     3364
     3365         if is_scalar(key) and isna(key) and not self.hasnans:
        KeyError: 39
```

```
[31]: print(df[["municipality","population"]])
      print("----------")
      print(type(df[["municipality","population"]]))
```

```
        municipality  population
0         Aa en Hunze     25243.0
1             Aalburg     12859.0
2            Aalsmeer     30792.0
3              Aalten     27030.0
4        Achtkarspelen    28002.0
..                 …           …
375           Zundert     21374.0
376           Zutphen     47154.0
377   Zwartewaterland     22148.0
378       Zwijndrecht     44546.0
379            Zwolle    123211.0

[380 rows x 2 columns]
----------
<class 'pandas.core.frame.DataFrame'>
```

Another handy feature is to filter data frames based on certain criteria. For example, we might only want to see the data of municipalities with at least 150,000 inhabitants:

```
[34]: print(df[df["population"]>=150000])
```

```
      municipality       province   latitude  longitude  surface_km2  \
9           Almere      Flevoland  52.350.785  5.264.702       248.77
13      Amersfoort        Utrecht  52.156.111  5.387.827        63.86
15       Amsterdam  Noord-Holland  52.370.216  4.895.168       219.30
16       Apeldoorn     Gelderland  52.211.157  5.969.923       341.15
18          Arnhem     Gelderland  51.985.103  5.898.730       101.54
52           Breda  Noord-Brabant  51.571.915  4.768.323       128.68
92       Eindhoven  Noord-Brabant  51.441.642  5.469.722        88.87
96        Enschede     Overijssel  52.221.537  6.893.662       142.72
116  's-Gravenhage   Zuid-Holland  52.070.498  4.300.700        98.12
117      Groningen      Groningen  53.219.383  6.566.502        83.75
122        Haarlem  Noord-Holland  52.387.388  4.646.219        32.09
221       Nijmegen     Gelderland  51.812.563  5.837.226        57.60
272      Rotterdam   Zuid-Holland  51.924.420  4.477.733       325.79
306        Tilburg  Noord-Brabant  51.560.596  5.091.914       119.18
315        Utrecht        Utrecht  52.090.737  5.121.420        99.21
364       Zaanstad  Noord-Holland  52.457.966  4.751.043        83.24

     population  avg_household_income_2012  avg_woz_2014  university
```

```
9       196156.0                        34900.0      182000.0            0
13      150943.0                        36900.0      222000.0            0
15      853312.0                        31400.0      231000.0            2
16      157535.0                        34800.0      208000.0            0
18      150817.0                        30500.0      175000.0            0
52      179999.0                        35200.0      221000.0            0
92      220782.0                        32000.0      209000.0            1
96      158542.0                        29600.0      155000.0            1
116     508592.0                        31800.0      188000.0            0
117     198108.0                        28500.0      157000.0            1
122     155205.0                        34300.0      229000.0            0
221     168499.0                        30900.0      191000.0            1
272     618467.0                        29600.0      148000.0            1
306     210382.0                        31600.0      187000.0            1
315     328577.0                        34300.0      223000.0            1
364     150911.0                        33300.0      177000.0            0
```

Or the data for the province of Utrecht:

[35]:
```python
print(df[df["province"]=="Utrecht"])
```

```
           municipality province   latitude  longitude  surface_km2  \
13           Amersfoort  Utrecht  52.156.111  5.387.827        63.86
22                Baarn  Utrecht  52.213.182  5.286.410        33.01
39             De Bilt  Utrecht  52.109.272  5.180.968        67.13
57               Bunnik  Utrecht  52.066.528  5.200.776        37.57
58           Bunschoten  Utrecht  52.240.642  5.367.070        34.81
88               Eemnes  Utrecht  52.253.746  5.261.275        33.70
157              Houten  Utrecht  52.002.991  5.185.760        58.99
160         IJsselstein  Utrecht  52.017.765  5.040.300        21.68
186             Leusden  Utrecht  52.131.793  5.429.469        58.89
192               Lopik  Utrecht  51.974.861  4.945.148        78.98
213           Montfoort  Utrecht  52.036.213  4.951.859        38.20
218          Nieuwegein  Utrecht  52.024.821  5.091.819        25.65
250           Oudewater  Utrecht  52.024.163  4.868.417        40.10
261           Renswoude  Utrecht  52.074.661  5.538.173        18.51
264              Rhenen  Utrecht  51.962.140  5.571.116        43.76
270      De Ronde Venen  Utrecht  52.206.680  4.886.773       116.98
288               Soest  Utrecht  52.176.352  5.299.197        46.43
297      Stichtse Vecht  Utrecht  52.144.755  5.033.208       106.82
315             Utrecht  Utrecht  52.090.737  5.121.420        99.21
316  Utrechtse Heuvelrug  Utrecht  52.052.203  5.282.495       134.09
321          Veenendaal  Utrecht  52.026.301  5.554.431        19.72
327              Vianen  Utrecht  51.990.276  5.103.033        42.39
355   Wijk bij Duurstede  Utrecht  51.975.600  5.338.450        50.25
359             Woerden  Utrecht  52.079.829  4.862.724        92.92
362          Woudenberg  Utrecht  52.082.175  5.426.595        36.82
369               Zeist  Utrecht  52.090.601  5.233.253        48.65
```

```
     population  avg_household_income_2012  avg_woz_2014  university
13     150943.0                    36900.0      222000.0           0
22      24344.0                    38600.0      290000.0           0
39      42013.0                    43100.0      338000.0           0
57      14619.0                    42600.0      271000.0           0
58      20547.0                    39400.0      250000.0           0
88       8773.0                    40100.0      291000.0           0
157     48427.0                    42900.0      261000.0           0
160     34268.0                    39300.0      234000.0           0
186     28967.0                    41000.0      267000.0           0
192     14000.0                    39500.0      267000.0           0
213     13639.0                    41200.0      264000.0           0
218     61017.0                    35200.0      206000.0           0
250      9868.0                    39600.0      283000.0           0
261      4928.0                    39200.0      281000.0           0
264     19123.0                    37100.0      255000.0           0
270     42648.0                    42700.0      300000.0           0
288     45430.0                    38700.0      273000.0           0
297     63823.0                    40800.0      270000.0           0
315    328577.0                    34300.0      223000.0           1
316     47939.0                    41400.0      305000.0           0
321     63207.0                    34800.0      210000.0           0
327     19573.0                    37400.0      219000.0           0
355     23031.0                    39500.0      261000.0           0
359     50607.0                    39900.0      251000.0           0
362     12386.0                    38800.0      281000.0           0
369     61337.0                    39700.0      301000.0           0
```

Or for the municipalities in the province of Utrecht with at least 150,000 inhabitants:

[36]:
```python
print(df[(df["population"]>=150000) & (df["province"]=="Utrecht")])
```

```
    municipality province    latitude  longitude  surface_km2  population  \
13    Amersfoort  Utrecht  52.156.111  5.387.827        63.86    150943.0
315      Utrecht  Utrecht  52.090.737  5.121.420        99.21    328577.0

     avg_household_income_2012  avg_woz_2014  university
13                     36900.0      222000.0           0
315                    34300.0      223000.0           1
```

Note that are several other clever ways to access (ranges of) values in data frames, but discussing them all would be out of scope of this lecture. We will see some of them in the examples later on, but if you are interested in digging deeper into this, please refer to the official “Indexing and Selecting Data” guide at http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html or ask Google if you are looking for hints how to index best in a specific situation.

In the following we will look at a few methods that pandas data frames provide. This selection is by no means complete, either, but you can find the full list at https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html.

For example, there are methods to easily sum up values, or get basic statistic information like the max, min, mean and median values. Just to show a few:

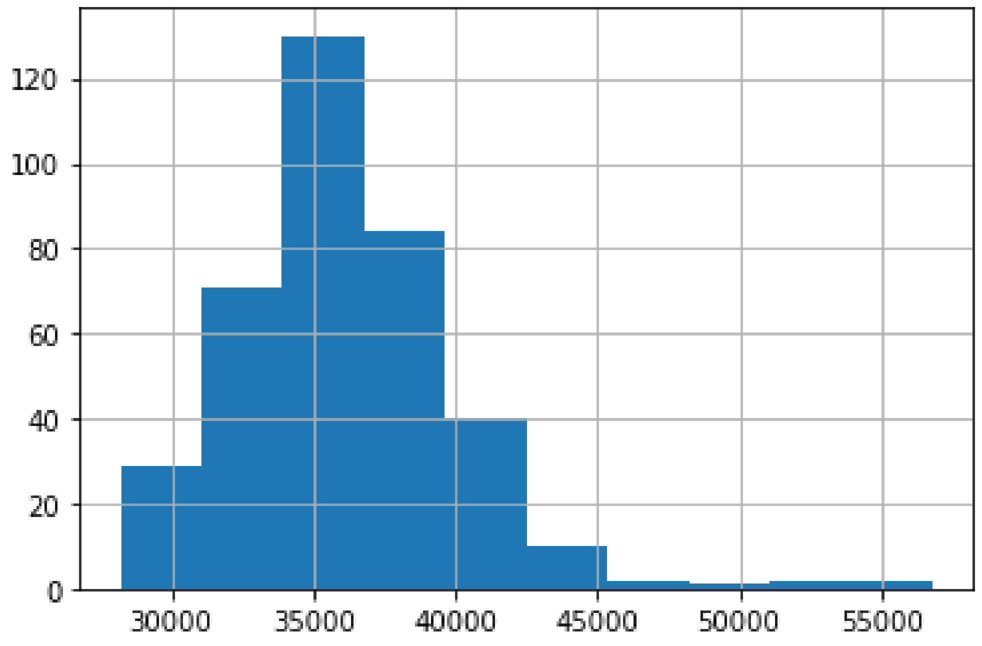
[37]:
```
print(f"Population was {df['population'].sum()} in total.")
print(f"The maximum population in a municipality was "\
      f"{df['population'].max()}.")
print(f"The average population per municipality was "\
      f"{df['population'].mean():.3f}.")
print(f"The average population per municipality with at "\
      f"least 1 university was "\
      f"{df[df['university']>=1]['population'].mean():.3f}.")
```

```
Population was 16589696.0 in total.
The maximum population in a municipality was 853312.0.
The average population per municipality was 44121.532.
The average population per municipality with at least 1 university was
261600.250.
```

The `hist` method can be used to plot simple histograms from data:

[38]:
```
print(df["avg_household_income_2012"].hist())
```

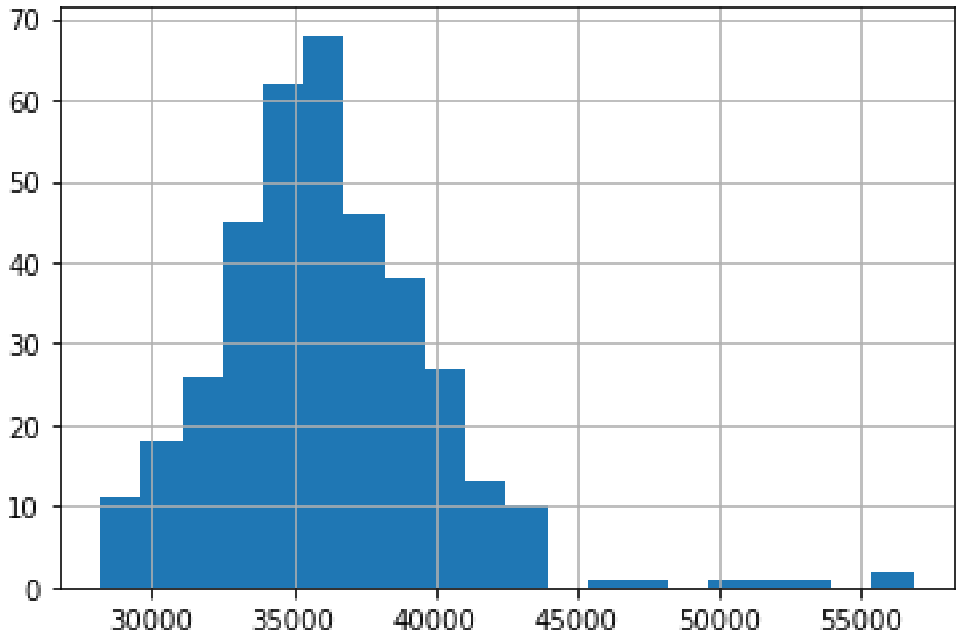
```
AxesSubplot(0.125,0.125;0.775x0.755)
```



Or, with a larger number of bins:

[39]:
```python
print(df["avg_household_income_2012"].hist(bins=20))
```

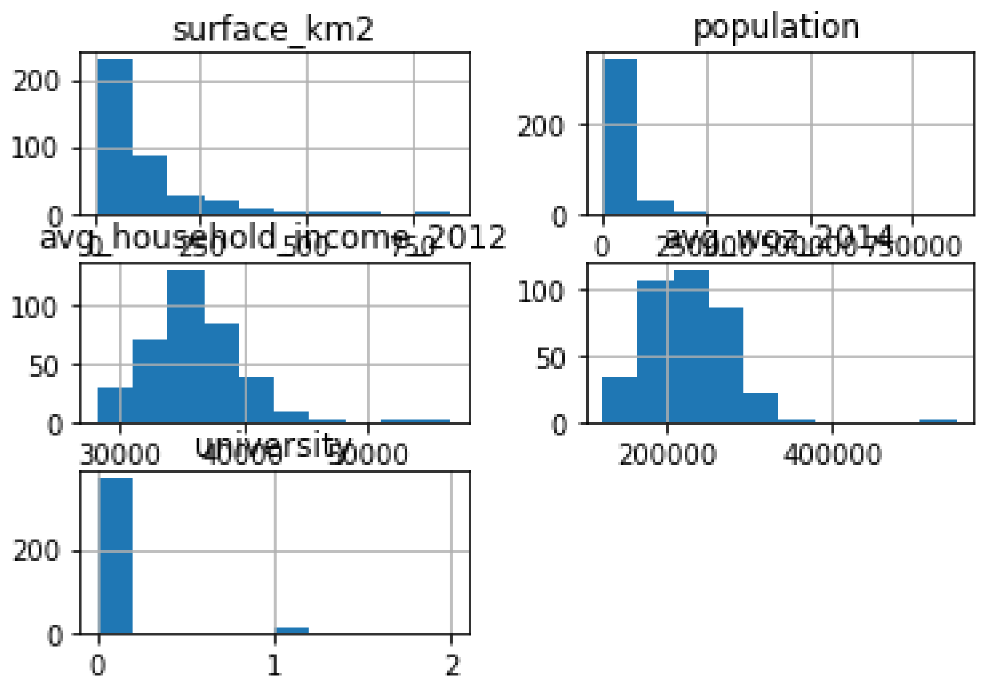
```
AxesSubplot(0.125,0.125;0.775x0.755)
```



If a data frame contains several columns with numeric values, the `hist` method will create histograms for all of them. For example, when called on the whole data frame:

[40]:
```python
print(df.hist())
```

```
[[<AxesSubplot:title={'center':'surface_km2'}>
  <AxesSubplot:title={'center':'population'}>]
 [<AxesSubplot:title={'center':'avg_household_income_2012'}>
  <AxesSubplot:title={'center':'avg_woz_2014'}>]
 [<AxesSubplot:title={'center':'university'}> <AxesSubplot:>]]
```



The possibilities for making histograms with `hist()` more “beautiful” are a bit limited, so other libraries should be used when a better design is wanted. However, for a quick check of the distribution of data in a data frame it is very suitable.

As a last example for today, we want to sort the data in the data frame according to average household income (descending), instead of having them sorted by municipality, like it is now. The `sort_values()` method is what we need:

[41]:
```
sorted_df = df.sort_values("avg_household_income_2012", ascending=False)
print(sorted_df[["municipality", "avg_household_income_2012"]])
```

```
         municipality  avg_household_income_2012
273         Rozendaal                    56800.0
43        Bloemendaal                    55600.0
340         Wassenaar                    52600.0
42           Blaricum                    51600.0
177             Laren                    50800.0
..                ...                        ...
203        Meierijstad                       NaN
208  Midden-Groningen                        NaN
222         Nissewaard                       NaN
335          Waadhoeke                       NaN
348        Westerwolde                       NaN

[380 rows x 2 columns]
```

Note that the index column was sorted with the rest of the data, too. So, if we want to have indices there running up from 0, we need to reset the index:

```
[ ]: sorted_reindexed_df = sorted_df.reset_index(drop=True)
     print(sorted_reindexed_df[["municipality", "avg_household_income_2012"]])
```

Finally, note that data frames can easily be saved as CSV files with the `to_csv()` method. For example:

```
[ ]: sorted_reindexed_df.to_csv('data/dutch_municipalities_sorted.csv')
```

We will see more about data frames in the following lecture(s).

1.3 Error Handling

There are basically two kinds of errors that can be detected by the Python interpreter: syntax (aka parsing) errors and exceptions (aka runtime or execution-time errors). `SyntaxErrors` are caused by syntactically incorrect code (like invalid variable names, forgotten indentations, braces, quotation marks or colons, etc.; Spyder will often already point you to them). They are fixed by correcting the code accordingly. Syntactically correct code can however still cause exceptions during exection. For example, a division by zero will result in a `ZeroDivisonError`, and a type mismatch between str and int will result in a `TypeError`. We say that an exception is "thrown" at runtime when the respective error occurs, and we can add code to "catch" and handle it if that happens (and thus prevent the program from simply crashing). That is done by the try-and-except construct in Python. Simply put, it defines what should be tried, and what happens if that goes wrong:

```
try:
    <do something>
except <error>:
    <do something to react on error>
```

For example, a `ValueError` is thrown when the user's input is not convertible into an integer, so we can catch it and display an error message accordingly:

```
[45]: try:
          x = int(input("Please enter a number: "))
      except ValueError:
          print("That was no valid number.")
```

```
Please enter a number:  isaac

That was no valid number.
```

```
[44]: int('hello')
```

```
        ␣
↪---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call␣
↪last)
```

```
    /tmp/ipykernel_10270/2364024281.py in <module>
----> 1 int('hello')


    ValueError: invalid literal for int() with base 10: 'hello'
```

In this case, it would in practice be handy if the user is asked to try again, until (s)he enters a valid input. Maybe even encapsulated into a function, to have a specific, error-handling reader available for reuse:

[46]:
```
def read_integer(prompt):
    while True:
        try:
            x = int(input(prompt))
            return x
        except ValueError:
            print("That was no valid number. Try again.")

# in main program:
number = read_integer("Please enter a number:" )
```

```
Please enter a number: isaac

That was no valid number. Try again.

Please enter a number: hello

That was no valid number. Try again.

Please enter a number: 78
```

As another example: When handling files, it can easily happen that the path to the file to be opened is not correct, and the file cannot be opened. Then the `FileNotFoundError` can be caught to prevent the program from crashing because of that:

[ ]:
```
filename = input("Enter file name: ")
while True:
    try:
        with open(filename, "r") as file:
            print(file.read())
        break
    except FileNotFoundError:
        print("File not found. Please try again.")
        filename = input("Enter file name: ")
```

There are several built-in exceptions in Python. We cannot go through them all, but you find them listed at https://docs.python.org/3/library/exceptions.html.

Often several things can potentially go wrong, so that it makes sense to catch several exceptions:

```
[ ]: number1 = read_integer("Enter number 1: ")
     number2 = read_integer("Enter number 2: ")
     try:
         print(number1 * number2)
         print(number1 / number2)
     except (FloatingPointError, OverflowError, ZeroDivisionError):
         print("Something went wrong with the calculation.")
```

Or in a more specific variant, distinguishing between division by zero and all other kinds of errors:

```
[ ]: number1 = read_integer("Enter number 1: ")
     number2 = read_integer("Enter number 2: ")
     try:
         print(number1 * number2)
         print(number1 / number2)
     except ZeroDivisionError:
         print("Division by 0!")
     except:
         print("Something went wrong with the calculation.")
```

As you can maybe guess from the previous example, and except clause with no specific error defined will catch all (remaining) errors that happen in the try clause. In such a case, it is often useful to assign a name to the exception that is caught, so that the error-handling code can check its type or get the underlying error message, to deal with the exception accordingly. For example:

```
[ ]: number1 = read_integer("Enter number 1: ")
     number2 = read_integer("Enter number 2: ")
     try:
         print(number1 * number2)
         print(number1 / number2)
     except Exception as err:
         print("Error handling for:", err)
```

Finally, note that with the `raise` statement it is also possible to let your own code throw one of the predefined or also self-defined exceptions:

```
[ ]: temperature = read_integer("Enter temperature: ")
     try:
         if 0 < temperature < 100:
             print("Water is liquid.")
         else:
             raise Exception("incompatible temperature", temperature)
     except Exception as err:
         print(err)
```

In practice it needs a bit of experience to decide how and where to implement error-handling behavior in a software. In the scope of the projects that you are working on in this course, it would not be feasible to surround each individual statement by try-and-except clauses. As a practical rule, error-handling should be implemented at places where things can easily go wrong, such as reading input from the user (even users with a lot of goodwill make typos), handling files (working with file systems is always prone to unexpected behavior) or accessing online resources and services (communication with them can be affected by network problems etc.). Generally, the less control the programmer (or their code) has over what happens, the more error-handling is a good idea.

Challenge!

Write a code to print only the municipalities with an average household income above 40000

Handle the case when the average household income is missing

[47]:
```python
#with open("data/dutch_municipalities.csv", "r") as csvfile:
#    csvreader = csv.DictReader(csvfile, delimiter='\t')
#    for row in csvreader:
#       if int(row["avg_household_income_2012"]) > 40000:
#           print(f'{row["municipality"]}: {row["province"]}')


with open("data/dutch_municipalities.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile, delimiter='\t')
    for row in csvreader:
        try:
            if int(row["avg_household_income_2012"]) > 40000:
                print(f'{row["municipality"]}: {row["province"]}')
        except ValueError:
            print(f'No INCOME for --> {row["municipality"]}: {row["province"]}')
```

```
Aalsmeer: Noord-Holland
Albrandswaard: Zuid-Holland
Amstelveen: Noord-Holland
Barendrecht: Zuid-Holland
Beemster: Noord-Holland
No INCOME for --> Berg en Dal: Gelderland
De Bilt: Utrecht
Blaricum: Noord-Holland
Bloemendaal: Noord-Holland
Bodegraven-Reeuwijk: Zuid-Holland
Bunnik: Utrecht
Castricum: Noord-Holland
Edam-Volendam: Noord-Holland
Eemnes: Utrecht
No INCOME for --> De Fryske Marren: Friesland
Giessenlanden: Zuid-Holland
No INCOME for --> Gooise Meren: Noord-Holland
```

```
Haaren: Noord-Brabant
Haarlemmerliede en Spaarnwoude: Noord-Holland
Haren: Groningen
Heemstede: Noord-Holland
Heeze-Leende: Noord-Brabant
Heiloo: Noord-Holland
Houten: Utrecht
No INCOME for --> Krimpenerwaard: Zuid-Holland
Landsmeer: Noord-Holland
Lansingerland: Zuid-Holland
Laren: Noord-Holland
Leusden: Utrecht
No INCOME for --> Meierijstad: Noord-Brabant
Midden-Delfland: Zuid-Holland
No INCOME for --> Midden-Groningen: Groningen
Molenwaard: Zuid-Holland
Montfoort: Utrecht
Mook en Middelaar: Limburg
No INCOME for --> Nissewaard: Zuid-Holland
Nuenen, Gerwen en Nederwetten: Noord-Brabant
Oegstgeest: Zuid-Holland
Ouder-Amstel: Noord-Holland
Pijnacker-Nootdorp: Zuid-Holland
De Ronde Venen: Utrecht
Rozendaal: Gelderland
Sint-Michielsgestel: Noord-Brabant
Son en Breugel: Noord-Brabant
Stichtse Vecht: Utrecht
Teylingen: Zuid-Holland
Utrechtse Heuvelrug: Utrecht
Voorschoten: Zuid-Holland
Vught: Noord-Brabant
No INCOME for --> Waadhoeke: Friesland
Waalre: Noord-Brabant
Wassenaar: Zuid-Holland
Waterland: Noord-Holland
No INCOME for --> Westerwolde: Groningen
Westvoorne: Zuid-Holland
Wijdemeren: Noord-Holland
Zoeterwoude: Zuid-Holland
Zuidplas: Zuid-Holland
```

1.4 Exercises

Please use Quarterfall to submit and check your answers.

1.4.1 1. Interview Anonymization ( )

Imagine you are a journalist, and you have written a text about an interview with somebody. Because the person wants to remain unrecognized, you have to replace their name through a fictive one everywhere in the text before it gets published. Write a Python program that reads the file containing the interview text, replaces all occurrences of the original name by a new one (the `str.replace()` function can be used here), and saves the changed text in the file. You can use the text file "interview-with-a-syrian-refugee.txt" or create an own one. Do not forget to implement error-handling.

1.4.2 2. Longest Word ( )

Reuse your code from exercise 5.5 (Text Analysis) to create a function that finds the longest word in a text. Apply it to the text file that you used for exercise 1 above. The output should be something like:

```
The longest word in the text is "responsibility".
```

Again, keep in mind to implement error-handling.

1.4.3 3. Randomized Story-Telling ( )

One of the simple pen-and-paper games I remember from my childhood days goes as follows: A paper sheet is divided into four columns for the questions "Who?", "Does what?", "How?" and "Where?". The first player would write down a person in the first column, then fold it away, the second would fill in a verb, fold it away, etc. After the fourth column has been filled, the complete sentence is read out. It could then be something like "My brother is showering excessively at the gas station."

Write a program that creates a user-defined number of such random sentences. The file `"inputs.csv"` contains a list of possible answers to all of the four questions. Take the values from there. Feel free to add further words to the CSV file to create more variation. The output of the program should be something like:

```
How many sentences do you want to create? 3
My granny is dancing massively at the fair.
The butcher is travelling aggressively in bed.
My grandpa is reading nicely in the bathroom.
```

1.4.4 4. Population and Universities per Province ( )

Write a Python program that reads in the CSV file `"dutch_municipalities.csv"` that we already used in the lecture. Sum up the population and universities for each province and write the result into a new CSV file `"dutch_provinces.csv"`, in alphabetical order of the province names. Its content should look like:

```
province,population,universities
Drenthe,488892.0,0
Flevoland,400179.0,0
Friesland,580537.0,0
Gelderland,1993851.0,2
Groningen,495508.0,1
```

```
Limburg,1119751.0,1
Noord-Brabant,2390214.0,2
Noord-Holland,2766854.0,2
Overijssel,1139754.0,1
Utrecht,1254034.0,1
Zeeland,380619.0,0
Zuid-Holland,3579503.0,3
```

1.4.5 5. Error Handling (   )

Add adequate "try and except" error handling to your code for exercises 1.-4. Include it in all code that you write from now on, at least when dealing with user inputs, file reading/writing operations, and accessing resources or services on the web.

1.5 Extras for the Weekend

Exercise 3 was hopefully a bit of fun, but of course we generated a very simple kind of prose text there. The website https://eh.bard.edu/generating-algorithmic-poetry/ describes how to use Python to automatically generate poems in the style of Shakespeare or Dickinson. Have a look if you find that interesting!

[ ]:

```

```