Reading and Writing Files with Pathlib

Create Paths

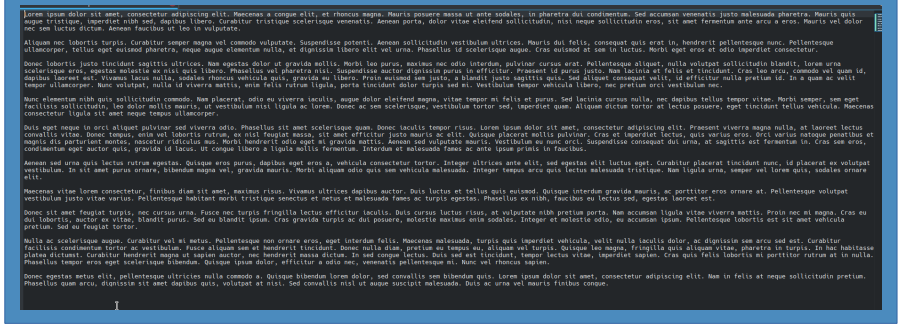
```
from pathlib import Path

relative_path = Path("folder/file.txt")
# On Linux
absolute_path = Path("/folder/subfolder/file.txt")
# On Windows
absolute_path = Path(r"C:\User\folder\file.txt")
absolute_path = Path("/User/folder/file.txt")

home = Path.home()
workingDir = Path.cwd()

some_path = home / "folder/subfolder" / "file.txt"
```

Extract Information

```
>>>
>>> from pathlib import Path
>>> fizzbuzz = Path("easyExercise_05_fizzbuzz.py")
>>> fizzbuzz
PosixPath('easyExercise_05_fizzbuzz.py')
>>> fizzbuzz = fizzbuzz.resolve()
>>> fizzbuzz
PosixPath('/home/jonas/Dokumente/pythonBootcamp2021/code/easyExercise_05_fizzbuzz.py')
>>> str(fizzbuzz)
'/home/jonas/Dokumente/pythonBootcamp2021/code/easyExercise_05_fizzbuzz.py'
>>> fizzbuzz.name
'easyExercise_05_fizzbuzz.py'
>>> fizzbuzz.parent
PosixPath('/home/jonas/Dokumente/pythonBootcamp2021/code')
>>> fizzbuzz.stem
'easyExercise_05_fizzbuzz'
>>> fizzbuzz.suffix
'.py'
>>> fizzbuzz.anchor
'/'
>>>
```

Reading Files

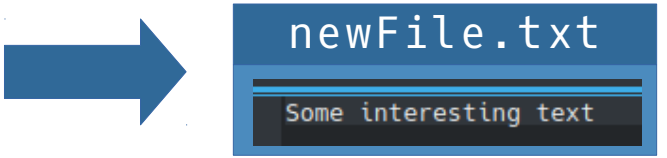
loremIpsum.txt



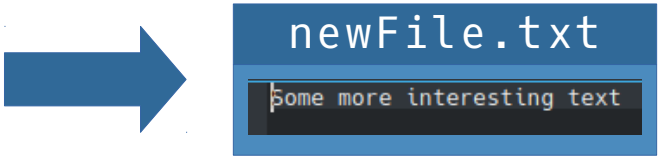

```
>>> from pathlib import Path
>>> file = Path("loremIpsum.txt")
>>> content = file.read_text()
>>> content[0:20]
'Lorem ipsum dolor si'
>>>
```

Writing Files

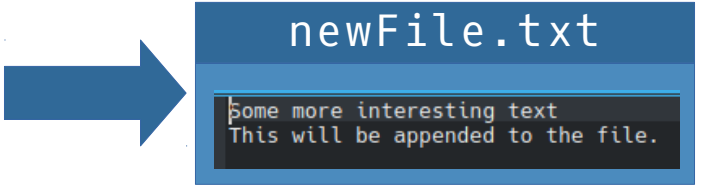
```
>>>
>>> from pathlib import Path
>>> file = Path("newFile.txt")
>>> file.write_text("Some interesting text\n")
22
```




```
>>> file.write_text("Some more interesting text\n")
27
>>>
```




```
>>> with file.open("a") as f:
...     f.write("This will be appended to the file.\n")
...
35
```



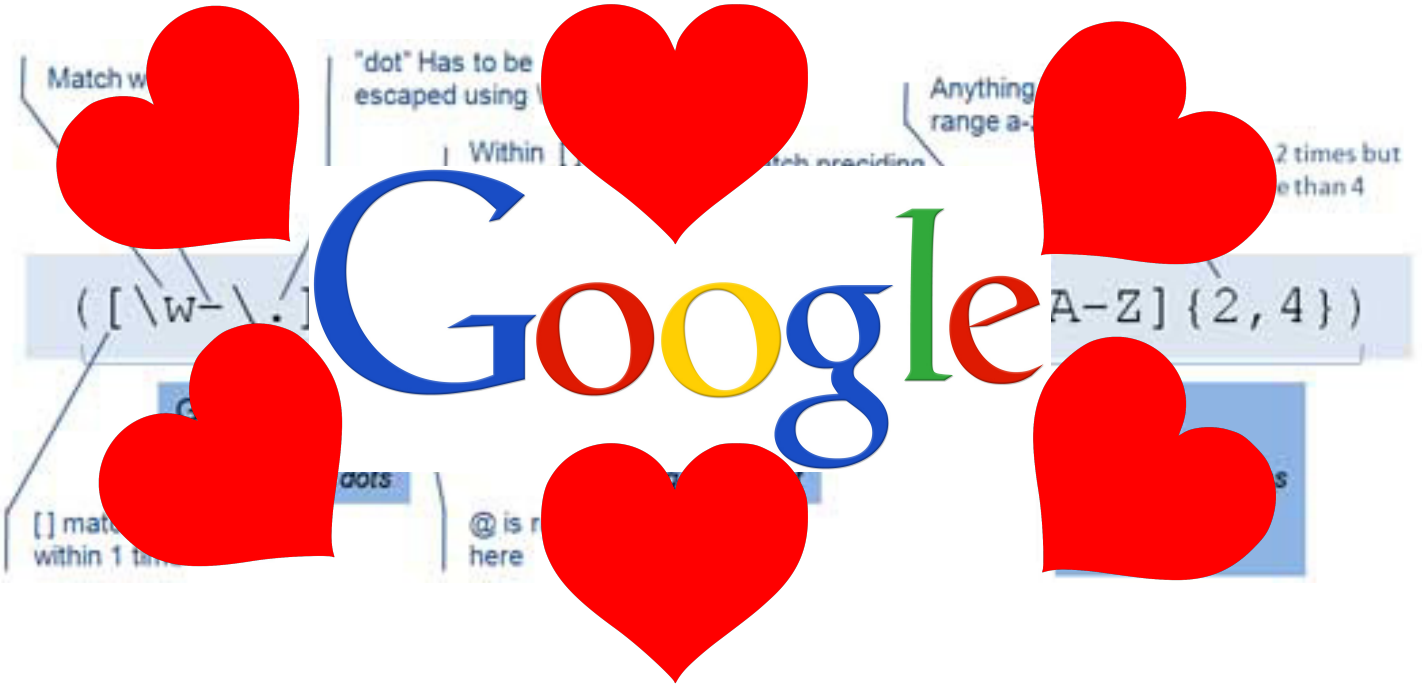
Handling Mutliple Files

```
>>>
>>> folder = Path("dummyFiles")
>>> print(folder.resolve())
/home/jonas/Dokumente/pythonBootcamp2021/code/dummyFiles
>>> for file in folder.iterdir():
...     content = file.read_text()
...     print(f"{str(file)} : {content[0:20]}")
...
dummyFiles/loremIpsum.txt : Lorem ipsum dolor si
dummyFiles/file2.txt : EFGH

dummyFiles/newFile.txt : Some more interestin
dummyFiles/file1.txt : ABCD

dummyFiles/file3.txt : IJKL

dummyFiles/file4.txt : MNOP
```

```
>>>
>>> for file in folder.glob("file*"):
...     print(file.name)
...
file2.txt
file1.txt
file3.txt
file4.txt
>>> unsorted_file_list = folder.glob("file*")
>>> sorted_file_list = sorted(unsorted_file_list)
>>> for file in sorted_file_list:
...     print(file.name)
...
file1.txt
file2.txt
file3.txt
file4.txt
>>>
```

Regex / Regular Expressions

```
import re

re.findall(pattern, string)
```



Exercise 6: Reading Files

You should have a folder with a bunch of files named alphabet and numbers.

Read the content of the files and print the content of the files in the right order to the screen.

Your result should be the alphabet and the numbers from 0 to 9.

And more …

Move, delete, copy files.

Create, delete, move directories.

Do anything you can do with files and directories.

File Types

CSV

```
"1.0.0.0","1.0.0.255","16777216","16777471","AU","Australia"
"1.0.1.0","1.0.3.255","16777472","16778239","CN","China"
"1.0.4.0","1.0.7.255","16778240","16779263","AU","Australia"
"1.0.8.0","1.0.15.255","16779264","16781311","CN","China"
"1.0.16.0","1.0.31.255","16781312","16785407","JP","Japan"
"1.0.32.0","1.0.63.255","16785408","16793599","CN","China"
"1.0.64.0","1.0.127.255","16793600","16809983","JP","Japan"
"1.0.128.0","1.0.255.255","16809984","16842751","TH","Thailand"
```

```
import csv
```

JSON

```
{
  "orderID": 12345,
  "shopperName": "Ivan Ivanov",
  "shopperEmail": "ivanov@example.com",
  "contents": [
    {
      "productID": 34,
      "productName": "Super product",
      "quantity": 1
    },
    {
      "productID": 56,
      "productName": "Wonderful product",
      "quantity": 3
    }
  ],
  "orderCompleted": true
}
```

```
import json
```

YAML

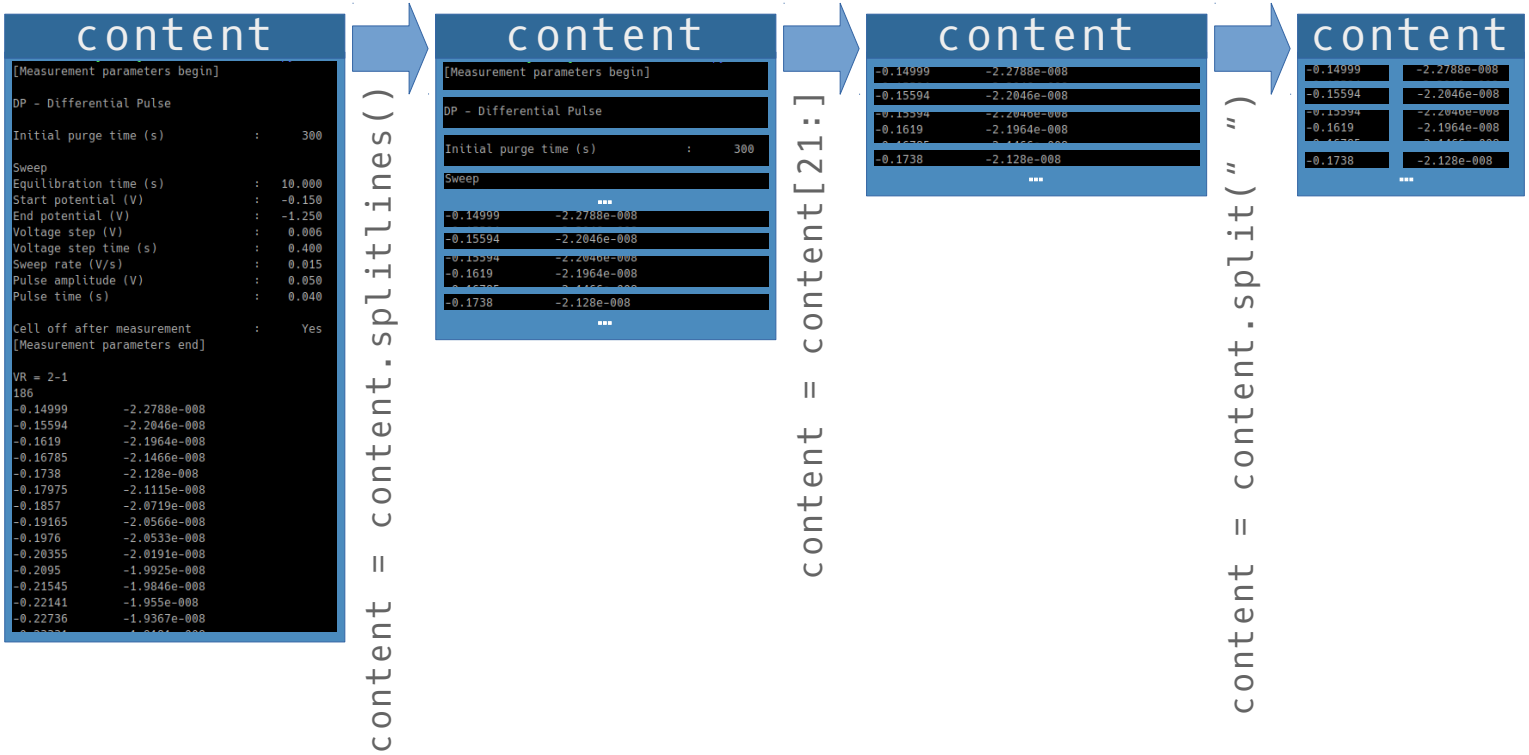
```
---
key: value
map:
    key1: "foo:bar"
    key2: value2
list:
  - element1
  - element2
# This is a comment
listOfMaps:
  - key1: value1a
    key2: value1b
  - key1: value2a
    key2: value2b
...
```

```
import yaml
```

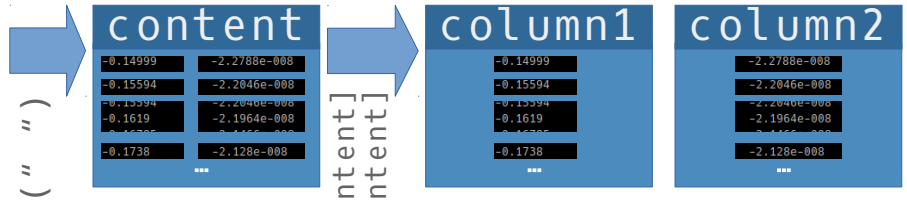
TXT

```
350,    0,250
351,    0,260
352,    0,291
353,    0,284
354,    0,299
355,    0,290
356,    0,265
357,    0,274
358,    0,278
359,    0,261
360,    0,294
361,    0,312
362,    0,279
363,    0,264
364,    0,272
365,    0,285
366,    0,291
367,    0,262
368,    0,265
```

Parsing Text



Parsing Text




```
content = content.split(" ")
column1 = [row[0] for row in content]
column2 = [row[1] for row in content]
```

convert German number notation to English notation with string.replace(“,”, “.”)
“2,34”.replace(“,”, “.”) → “2.34”
convert string to numbers with float()

Exercise 7: Parsing Files

You should have a file named measuredData.txt.
Read the file and convert the data in it into an array.
Print the resulting array to the screen.