PYTHON 101 — Recap

Reading Files

```python
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"
```

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

Reading Multiple 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
>>>
```

Parsing Files

Ready to use parsers

...

Slicing

```
>>>
>>> array = [0,1,2,3,4,5,6,7,8,9]
>>> array[0]
0
>>> array[-1]
9
>>> array[0:2]
[0, 1]
>>> array[2:6]
[2, 3, 4, 5]
>>> array[2:6:2]
[2, 4]
>>> array[::-1]
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
>>> array[:]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

array[start:end:step]

Regex

```
import re

re.findall(pattern, string)
```

Regressions

```
# Import scikit-learn
from sklearn.linear_model import LinearRegression

# Create a linear regression model
linearModel = LinearRegression()
# Fit testing data
linearFit   = linearModel.fit(xvalues, yvalues)
# Extract the fitting parameters
coefficient = linearFit.coef_
intercept   = linearFit.intercept_
```

```
xvalues = [x0, …, xn]
```

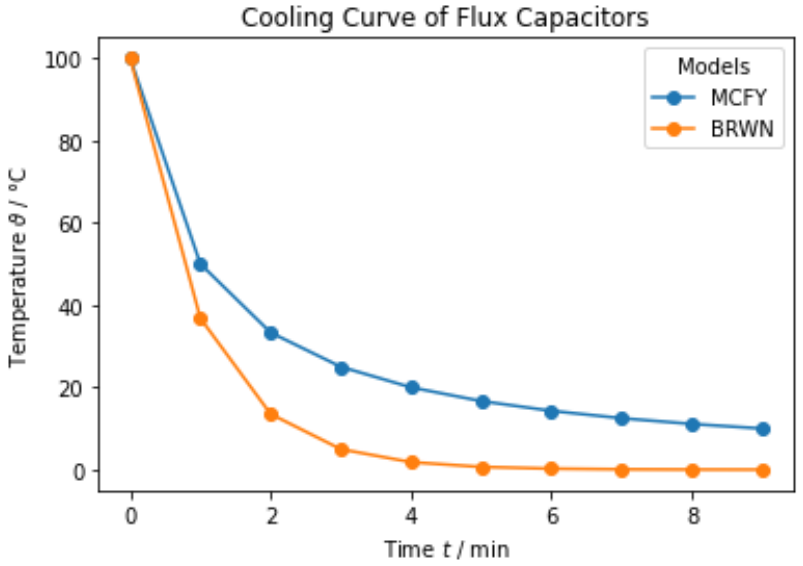
xvalues.reshape(-1,1)

```
xvalues = [ [x0],
            …
            [xn] ]
```

```
# Import modules
from scipy.optimize import curve_fit
import numpy as np

# Define a regression formula
model = lambda x, A, b : A * np.exp(b*x)
# Fit testing data
fit = curve_fit(    f     = model,
                    xdata = xvalues,
                    ydata = yvalues,
                    p0    = [initial_A, intial_b] )
# Extract fitting parameters
parameters = fit[0] # → [ A, b ]
```

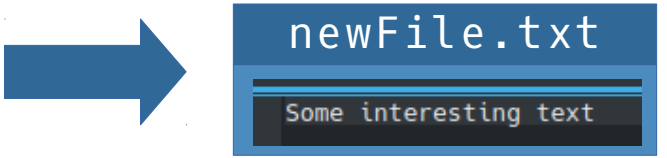
Plotting

```python
# Create some dummy data
experimentOne = {"time": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
                 "temperature": [100.00, 50.00, 33.33, 25.00, 20.00, 16.67, 14.29, 12.50, 11.11, 10.00] }
experimentTwo = {"time": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
                 "temperature": [100.00, 36.79, 13.53,  4.98,  1.83,  0.67,  0.25,  0.09,  0.03,  0.01] }

# Import module that creates plots
from matplotlib import pyplot as plt

# Create a plot with the dummy data
# The parameter "-o" creates a lineplots with dot marks.
# The label parameter assures that the curves will be colored and added to the legend
#
# Add first dictionary
plt.plot("time", "temperature", "-o", data = experimentOne, label = "MCFY")
# Add second dictionary
plt.plot("time", "temperature", "-o", data = experimentTwo, label = "BRWN")
# Add titles and axis labels
plt.title("Cooling Curve of Flux Capacitors")
plt.xlabel("Time $t$ / min")
plt.ylabel(r"Temperature $\vartheta$ / °C")
# Add a legend. Give the legend a descriptive title.
plt.legend(title = "Models")
```

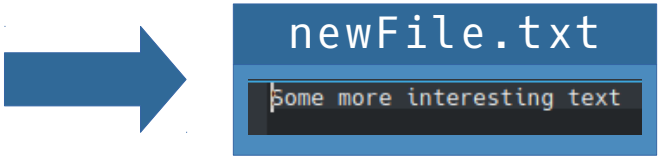


Writing Files

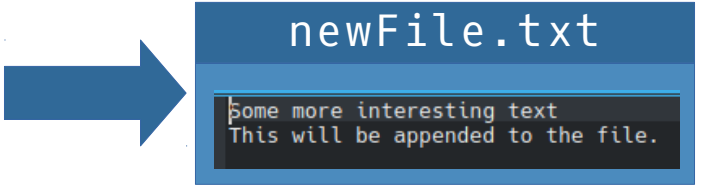
```
>>>
>>> from pathlib import Path
>>> file = Path("newFile.txt")
>>> file.write_text("Some interesting text\n")
22
```




```
>>> 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
```



Writing Files

Create a string with a table

```
from tabulate import tabulate

table = tabulate(2D_array_with_data, headers = 1D_array_column_names)

Path("table.txt").write_text(table)
```

Ready to use parsers

...