

Cheat sheet: File handling

File handling with explicit close

```
fp = open(filename, mode)
```

```
...
```

```
fp.close()
```

File handling via with-block

```
with open(filename, mode) as fp:
```

```
...
```

File opening modes

r reading **w** writing **a** appending

+ updating (read/write)

b binary (otherwise text)

Iterating over lines

```
for line in fp:
```

```
...
```

File reading methods

`fp.readlines()` All lines as list

`fp.readline()` Next line as string

`fp.read()` Entire content as string

File writing methods

`fp.write()` Write str

`fp.writelines()` Write list of strings

Numpy data I/O

`np.loadtxt()` Content into array

`np.savetxt()` Array data to file

Path manipulation

`os.path` Paths as strings

`pathlib` Path as objects

Cheat sheet: Plotting via Matplotlib

Accessing pyplot interface

```
import matplotlib.pyplot as plt
```

Embedding figures in Jupyter-notebook

```
%matplotlib inline
```

Accessing Figures and Axes instances

```
fig, ax = plt.subplots()
```

Some useful Axes methods

```
ax.plot(xvals, yvals, ..)
```

```
ax.set_xlim(), ax.set_ylim()
```

```
ax.set_xticks(), ax.set_yticks()
```

```
ax.annotate(..)
```

```
ax.legend(..)
```

```
ax.xaxis.set_ticks_position(..)
```

```
ax.spines["top"].set_position(..)
```

```
ax.spines["top"].set_color(..)
```

Some useful Figures methods

```
fig.show()
```

```
fig.savefig(..)
```

Some plot options

color "red", "blue", ..

linewidth Width in pixels

linestyle "-", "--", "o"

label Label in legend box

TeX sequences $r"\frac{1}{2}"$

Generating equidistant grid

```
np.linspace(from, to, steps, ..)
```