# Cheat sheet: File I/O

## 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 file content as string |

## File writing methods

| | |
|---|---|
| *fp*.**write()** | Write str to file |
| *fp*.**writelines()** | Write list of strings to file |

## Numpy data I/O

| | |
|---|---|
| np.**loadtxt()** | Content into array |
| np.**savetxt()** | Array data to file |

## Path manipulation

**os.path** Module with functions

**os.path.join(***path1***,** *path2***)**
Join path components

# Cheat sheet: Plotting

## Simplified interface

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

## Embedding figures in Jupyter

```
%matplotlib inline
```

## Some useful functions

```
np.linspace(from, to, steps, ..)
plt.plot(xvals, yvals, ..)
plt.subplot(nrow, ncol, iplot)
plt.xlim(), plt.ylim()
plt.xticks(), plt.yticks()
plt.annotate(..)
plt.legend(..)
plt.show()
plt.savefig(..)
```

## Axis objects

```
ax = plt.gca()      Current axis
ax.xaxis.set_ticks_position(..)
ax.spines["top"].set_position(..)
ax.spines["top"].set_color(..)
```

## Some plot options

```
color          "red", "blue", ..
linewidth      Width in pixels
linestyle      "-", "--", "o"
label          Label in legend box
```

## TeX sequences

```
r"$\frac{1}{2}$"
```