Cheat sheet: Functions & modules

Function definition

```
def fname(arg, kwarg=defval, ..):
    """Documentation ..."""
    ..
    return result
```

Function call

```
fname(arg, .. kwarg=kwvalue, ..)
```

None (singleton)

```
item is None, item is not None
```

Comparsion

```
==, /=        checking on value
is, is not    checking on identity
              (same object in mem.?)
```

Module definition

```
module name = name of file
                containing the module
```

Module import

```
import modulename
import modulename as newname
from modulename import entity
```

Cheat sheet: Numpy arrays

Creating numpy array

```
np.array([1, 2, 3], dtype=float)
np.array([[1.0, 2.0], [3.0 ,4.0]])
np.array(otherarray)   (copy)
np.empty((3, 3), dtype=float)
np.zeros(..), np.ones(..)
```

rank number of indices (axes)
size number of all elements

Accessing elements *arr*[*i1*, *i2*, …]

Array slices

at least one index is a range
arr[1, 2:3], *arr*[1, :]
Slices = **array views** (mutable!)

Array shape *arr*.shape()

Elementwise array operators

+, -, *, /, **, <, >, <=, >=, !=
ufuncs (np.sin, np.sqrt, ..)

Matrix multiplication

A @ *B*, np.dot(*A*, *B*)

Reduction

np.sum(..), np.product(..)

Broadcasting

Array repeated along missing axes
Axes inserted **before** existing ones
np.**newaxis** Inserting axis at arbitrary position

Iteration over array

Behaves as iteration over nested lists