Python cheat sheet

Basics

- `import libraryname` imports a library into a program
- `variable = value` assigns a value to a variable
- `print(something)` displays the value of `something`.
- `# some kind of explanation`: begin a line with `#` to make it a comment.

Arrays

- `numpy` is a library to work with arrays in Python.
- `array.shape` gives the shape of an array.
- `array[x, y]` selects a single element from a 2D array.
- Array indices start at 0, not 1.
- `low:high` specifies a `slice` that includes the indices from `low` to `high*1`.
- All the indexing and slicing that works on arrays also works on strings.
- Use `numpy.mean(array)`, `numpy.max(array)`, and `numpy.min(array)` to calculate simple statistics.
- Use `numpy.mean(array, axis=0)` or `numpy.mean(array, axis=1)` to calculate statistics across the specified axis.
- Use the `pyplot` library from `matplotlib` for creating simple visualizations.

Loops

- Use `for variable in sequence` to process the elements of a sequence one at a time.
- Use `len(thing)` to determine the length of something that contains other values.

Lists

- `[value1, value2, value3, ...]` creates a list.
- Lists are indexed and sliced in the same way as strings and arrays.
- Lists are mutable (i.e., their values can be changed in place).
- Strings are immutable (i.e., the characters in them cannot be changed).

Analyzing data from multiple files

- Use `glob.glob(pattern)` to create a list of files whose names match a pattern.
- Use `*` in a pattern to match zero or more characters, and `?` to match any single character.

Conditionals

- Use `if condition` to start a conditional statement, `elif condition` to provide additional tests, and `else` to provide a default.
- The bodies of the branches of conditional statements must be indented.
- Use == to test for equality.
- `X and Y` is only true if both `X` and `Y` are true.
- `X or Y` is true if either `X` or `Y`, or both, are true.
- Zero, the empty string, and the empty list are considered false; all other numbers, strings, and lists are considered true.

Functions

- Define a function using `def name(...params...)`.
- Call a function using `name(...values...)`.
- Use `help(thing)` to view help for something.
- Specify default values for parameters when defining a function using `name=value` in the parameter list.

Errors and exceptions

- `SyntaxError`: an error having to do with the 'grammar' or syntax of the program.
- `IndentationError`: an issue has to do with how the code is indented.
- `NameError`: occurs if you use a variable that has not been defined, either because you meant to use quotes around a string, you forgot to define the variable, or you just made a typo.
- `IndexError`: containers like lists and strings will generate this error if you try to access items in them that do not exist.
- `FileNotFoundError`: occurs when trying to read a file that does not exist.
- `IOError`: occurs when trying to read a file that is open for writing, or writing to a file that is open for reading, will give you an.