Notebook Application

Objects

**Notes**

Attributes

* Memo -> Content
* Tags
* Creation\_date

Method

Match -> Take a str and tells us whether a note matches the string without accessing the attribute directly. This way if we want to modify the search parameter to add search tags instead of note contents or to make the search case insensitive we only have to do it in 1 place

Notebook

Attribute

Notes -> A list containing note objects

Method

Add -> Takes an object of note and add it to the notebook

Modify -> Need to modify an existing note by id

Search -> Takes a str and outputs a list of notes.

Menu

* WE need ability to interact with note and notebook. This is a command line app. Means we can run program with different options to add ort edit commands. Menu is the list of commands that allow us to do different things to notebook.
* We should also design interface to support future changes such as changing interfaced to GUI toolkit to a web-based interface

Folder strucuture

Menu interface is its own module as it is a script. This is to allow other scripts ability to access notebook in future and they can be in the same folder.

Notebook and Note objects live together in 1 module.

Both modules exists in the same top-level directory without putting them in a package.

Empty command\_options.py moduile can help remind us in the future if we plan to add a new user interface

Notes

Remember if you have too many break statements you probably need to break up the function to have 1 method explicitly to just find what you are looking for.