# Reference

## Preamble: The purpose of this document

Each section will describe a file we have and its high-level purpose. We will not go into a lot of detail in this document.

## Get\_links.py – Deprecated

Given a Wikipedia article name, create a page object and search through each paragraph for hyperlinks. Filter out those housing references (e.g. [1]).

Two outputs:

1. A file for users to view: contains list of <Name\_That\_Was\_Used\_To\_Invoke\_Link, link>
2. File for the program to use: contains list of links

Problem: Output number one will be difficult to sort. This relies on Bash to sort and streamline output number two.

## Get\_links2.py

Given a Wikipedia article name, create a page object and search through each paragraph for hyperlinks. Filter out references, e.g. [1].

Outputs a list of links that are on the Wikipedia article to standard output. Sorted in lexicographical order, and with duplicates removed. Right now, there is an array of strings in which each link (as a string) is stored. Each string is in the format of Name\_Of\_Article.

This is done recursively, so each Name\_Of\_Article has the function run on it again. The recursive function is called get\_links(name, depth, DEPTH\_LIMIT), where it will recurse until hitting depth==DEPTH\_LIMIT. The DEPTH\_LIMIT can be changed to have deeper or shallower searches. As of now, the default DEPTH\_LIMIT is 2. This recursive call mimics depth-first search of the Wikipedia links, sorted by alphabetical order.

This also contains a function (and corresponding call) for a BFS version of the search.

## Main.sh – Deprecated

Control the file. Run this file to set everything in motion.

## Streamline.sh – Deprecated

Sort and remove duplicates from the second output of get\_links.py

## Makefile

Used for file control (removing rubbish files)

Run `make` to run main.sh