Basic Bash

Hamid Semiyari

- Bash Reference Manual
- **Note: make sure to download the file `git_hist.txt` into your working directory. We will use it later.**

BASH

Basic Bash (Bourne Again SHel)

- `GNU` read "Gnoo" is an open source operating system.
- Bash is the GNU Project's shell ## The Command Line
- The **command line** is like the R command prompt: you insert code, hit enter, and then the computer executes your command.
- However, instead of inserting R code, you insert Shell Script.
- In this class, we will use the command line primarily for two things:
  - Moving around your file system.
  - Running git commands.

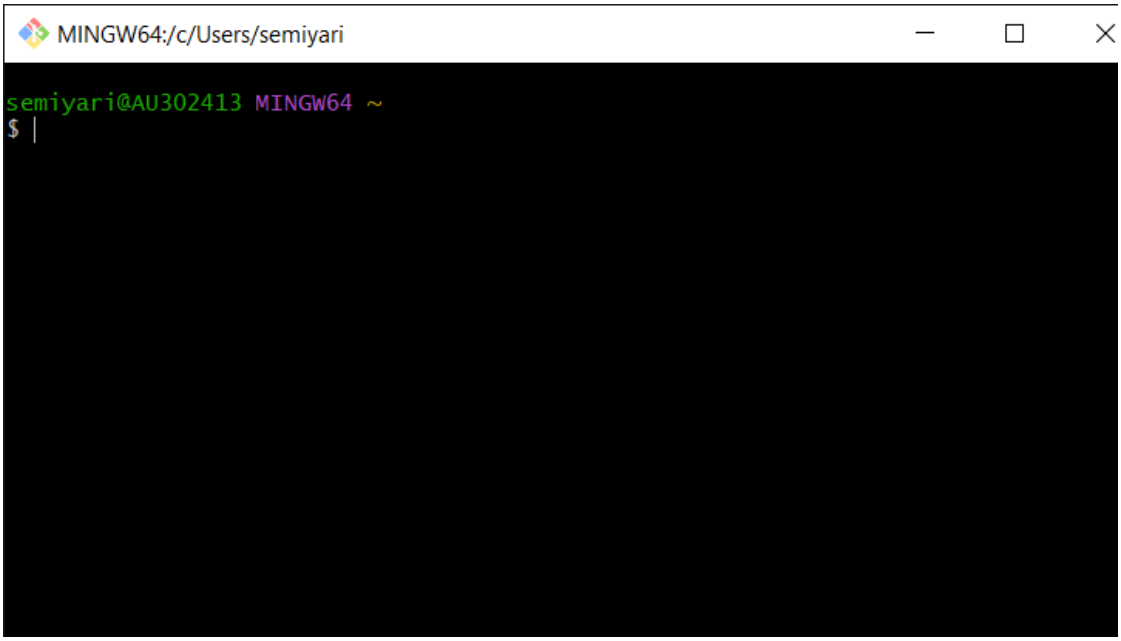
Other words for command line:

- Shell,
- terminal,
- command line interface (cli), and
- console.
  - These terms are technically slightly different.

- There are many types of shells, each with their own scripting language. We will use the bash scripting language for this class.
- A huge difference between R and bash is how commands/functions are called.
  - R: `f(x, y = 1)`
  - Bash: `f x --y=1`
  - Arguments that are "flags" use only one dash, like `f x -g` would incorporate the `g` flag.
- If you are using Linux or Mac, then you can keep going. If you are using Windows, you need to first download and install git (and thus git bash) from here: http://git-scm.com/download/win. You might need to restart R Studio if you are already running it.

Open up the terminal

- Windows: Open up the Git Bash app. It should look like this:



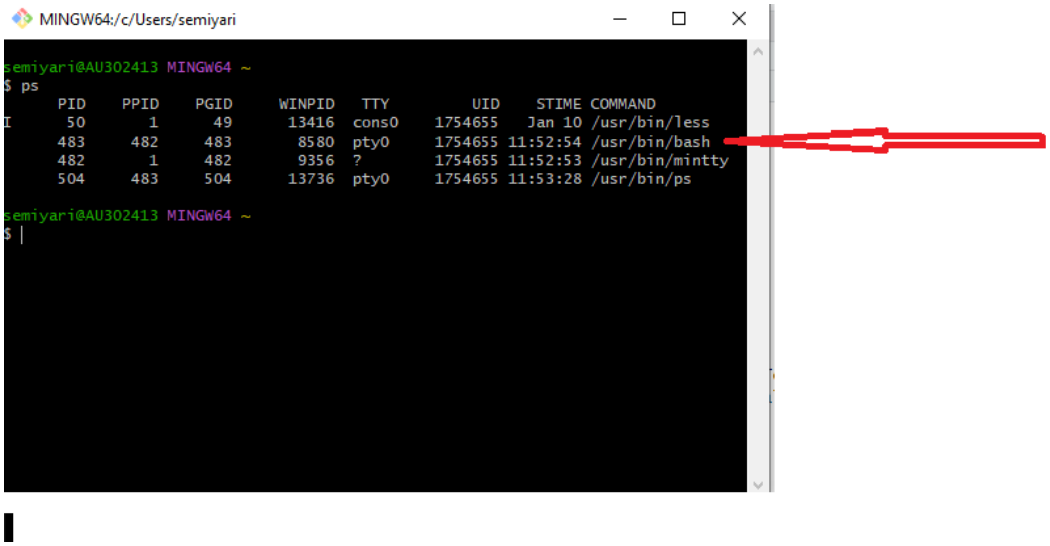
- Mac: On your Mac, do one of the following:

- Click the Launchpad icon in the Dock, type Terminal in the search field, then click Terminal. - In the Finder  , open the /Applications/Utilities folder, then double-click Terminal.
- Ubuntu: Do one of the following

Open the dash and search for "terminal". Open up the terminal.

Use keyboard shortcut: Ctrl+Alt+T

- **The easiest way to open a terminal is within R Studio with "Tools > Terminal > New Terminal"**
- All commands get placed after the dollar sign.
  - Bash is type of shell. There are different type of shells available, such as Korn shell (ksh), C shell (csh), and Z shell (zsh).
  - You can determine your shell type using the `ps` command:



- The path before the dollar sign is the working directory of the terminal, **not** R's working directory. It's where the shell will reference all files from.
- The tilde "~" is shorthand for the "home directory". Each computer has a home directory that is the "default directory".

Useful Commands:

- `date`: Displays the current date

```
date
```

```
Tue Feb 13 06:16:42 PM EST 2024
```

- `pwd`: Print working directory. Show the current working directory. This is like `getwd()` in R.

```
  pwd
```

```
/mnt/c/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Git
```

- `ls`: List the current files and folders in a directory.

```
  ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
```

```
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
```

```
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

- `echo`: Prints a string of text, or value of a variable to the terminal.

```
echo "Hello World"
```

```
Hello World
```

- `man` command in Linux is used to display the user manual of any command that we can run on the terminal.

```
man ls
```

```
/bin/bash: line 1: man: command not found
```

- If you are using Windows with a Bash emulator like Git Bash, the man command may not be available. You can use online resources or alternative commands like `--help`. You can scroll through this page using the up and down arrows.

```
ls --help
```

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing them;
                             e.g., '--block-size=M'; see SIZE format below
```

```
-B, --ignore-backups       do not list implied entries ending with ~
-c                         with -lt: sort by, and show, ctime (time of last
                           modification of file status information);
                           with -l: show ctime and sort by name;
                           otherwise: sort by ctime, newest first

-C                         list entries by columns
    --color[=WHEN]         color the output WHEN; more info below
-d, --directory            list directories themselves, not their contents
-D, --dired                generate output designed for Emacs' dired mode
-f                         list all entries in directory order
-F, --classify[=WHEN]      append indicator (one of */=>@|) to entries WHEN
    --file-type            likewise, except do not append '*'
    --format=WORD          across -x, commas -m, horizontal -x, long -l,
                           single-column -1, verbose -l, vertical -C

    --full-time            like -l --time-style=full-iso
-g                         like -l, but do not list owner
    --group-directories-first
                           group directories before files;
                           can be augmented with a --sort option, but any
                           use of --sort=none (-U) disables grouping

-G, --no-group             in a long listing, don't print group names
-h, --human-readable       with -l and -s, print sizes like 1K 234M 2G etc.
    --si                   likewise, but use powers of 1000 not 1024
-H, --dereference-command-line
                           follow symbolic links listed on the command line
    --dereference-command-line-symlink-to-dir
                           follow each command line symbolic link
                           that points to a directory

    --hide=PATTERN         do not list implied entries matching shell PATTERN
                           (overridden by -a or -A)

    --hyperlink[=WHEN]     hyperlink file names WHEN
    --indicator-style=WORD
                           append indicator with style WORD to entry names:
                           none (default), slash (-p),
                           file-type (--file-type), classify (-F)

-i, --inode                print the index number of each file
```

```
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN
  -k, --kibibytes            default to 1024-byte blocks for file system usage;
                             used only with -s and per directory totals

  -l                         use a long listing format
  -L, --dereference          when showing file information for a symbolic
                             link, show information for the file the link
                             references rather than for the link itself

  -m                         fill width with a comma separated list of entries
  -n, --numeric-uid-gid      like -l, but list numeric user and group IDs
  -N, --literal              print entry names without quoting
  -o                         like -l, but do not list group information
  -p, --indicator-style=slash
                             append / indicator to directories
  -q, --hide-control-chars   print ? instead of nongraphic characters
      --show-control-chars   show nongraphic characters as-is (the default,
                             unless program is 'ls
```

- Comments in bash starts with a `#` in bash scripting. This means that any line that begins with a `#`is a comment and will be ignored by the interpreter.

```
echo "Comments in bash starts with a # in bash scripting." # This line that begins with a
```

```
Comments in bash starts with a # in bash scripting.
```

- `cd`: Change directories. This is like `setwd()` in R. As when we specified paths in R, using two periods mean "move back a folder".

```
pwd
```

```
/mnt/c/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Git
```

- The `cd../` tells move back a folder

```
cd ../
pwd
```

```
/mnt/c/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24
```

- If you use `cd` without specifying a folder to move to, it will move the working directory to the home directory.

```
cd
pwd
```

```
/home/semiyari
```

- OK, I'm going to move us back to the "Week1" directory.

```
cd c:Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Week1
pwd
```

```
/bin/bash: line 1: cd: c:Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Week1: No such f
/mnt/c/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Git
```

Other commands.

```
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
```

```
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
```

```
git.txt.7
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

- cp: Copy a file.

```
cp Exercise-Bash.qmd copy_exercise_bash.qmd
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
```

```
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
copy_exercise_bash.qmd
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
init_repo_files
```

```
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

- mv: Move/rename a file.

```
mv copy_exercise_bash.qmd rename_exercise_bash.qmd
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
```

```
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
```

```
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
rename_exercise_bash.qmd
ssh_pat_2fa.qmd
```

- `rm`: Remove a file.

```
rm rename_exercise_bash.qmd copy_exercise_bash.qmd
ls
```

```
rm: cannot remove 'copy_exercise_bash.qmd': No such file or directory
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
```

```
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
```

```
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

- `mkdir`: Make a directory/folder.

```
mkdir tempdir
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
```

```
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
tempdir
```

- `rmdir`: Remove a directory/folder.

```
rmdir tempdir
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
```

```
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
init_repo_files
 is used to append to a file.
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

- Play Terminus for more practice.

some other useful terminal/bash tools.

- grep
- wget
- reverse-i-search

GREEP

String search with grep

- grep is a powerful command-line tool in Unix/Linux environments for searching through text using regular expressions.
  - -r recursive
  - -n line number
  - -w whole word only
  - -e pattern
  - pdfgrep (need to install separately) for searching text in PDFs.

Here's a quick guide to help you get started with grep. I have a file name called "git_hist.txt"

Search for a specific word in a file:

```
pwd
```

```
/mnt/c/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Git
```

```
cat git_hist.txt
```

```
A Short History of Git
As with many great things in life, Git began with a bit of creative destruction and fiery co
The Linux kernel is an open source software project of fairly large scope. During the early
2002), changes to the software were passed around as patches and archived files. In 2002, th
In 2005, the relationship between the community that developed the Linux kernel and the comm
•   Speed
•   Simple design
•   Strong support for non-linear development (thousands of parallel branches)
```

- Fully distributed
- Able to handle large projects like the Linux kernel efficiently (speed and data size)

```
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
_file
```

```
grep "Git" git_hist.txt
```

```
A Short History of Git
As with many great things in life, Git began with a bit of creative destruction and fiery co
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
```

If Terminal doesn't return any result

- Double check the command. Make sure file exists in your directory and also the file contains the word that you are searching.

```
grep "tig" git_hist.txt   # There will be no match
```

Also, by default, `grep` is case sensitive. If your search contains lowercase or uppercase letters you want to use `-i` option

```
grep "git" git_hist.txt   # There will be no match
```

```
grep -i "git" git_hist.txt   #not-case sensative
```

```
A Short History of Git
As with many great things in life, Git began with a bit of creative destruction and fiery co
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
```

- Sometimes, there might be non-printable characters in the file that are not visible. You can try using tools like `cat -v` to display non-printable characters:

```
cat -v git_hist.txt
```

```
A Short History of Git^M
As with many great things in life, Git began with a bit of creative destruction and fiery co
The Linux kernel is an open source software project of fairly large scope. During the early
In 2005, the relationship between the community that developed the Linux kernel and the comm
M-bM-^@M-"  Speed^M
```

```
M-bM-^@M-"  Simple design^M
M-bM-^@M-"  Strong support for non-linear development (thousands of parallel branches)^M
M-bM-^@M-"  Fully distributed^M
M-bM-^@M-"  Able to handle large projects like the Linux kernel efficiently (speed and data s
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
_file^M
```

Search recursively in directories:

- This command restricts the search to files with the `.txt` extension within the documents directory and its subdirectories.

```
grep -r "Git" --include="*.txt"
```

```
git.txt:Git is a fast, scalable, distributed revision control system with an
git.txt:commands.  The link:user-manual.html[Git User's Manual] has a more
git.txt:page to learn what commands Git offers.  You can learn more about
git.txt:individual Git commands with "git help command".  linkgit:gitcli[7]
git.txt:A formatted and hyperlinked copy of the latest Git documentation
git.txt:    Prints the Git suite version that the 'git' program came from.
git.txt:    available commands are printed. If a Git command is named this
git.txt:    Path to wherever your core Git programs are installed.
git.txt:    Print the path, without trailing slash, where Git's HTML
git.txt:    this version of Git and exit.
git.txt:    version of Git are installed and exit.
git.txt:    Do not pipe Git output into a pager.
git.txt:top-level of the working tree are discovered), and tells Git
git.txt:should tell Git where the top-level of the working tree is,
git.txt:    Set the Git namespace.  See linkgit:gitnamespaces[7] for more
git.txt:    Do not use replacement refs to replace Git objects. See
git.txt:We divide Git into high level ("porcelain") commands and low level
git.txt:Although Git includes its
git.txt:The following documentation pages are guides about Git concepts.
git.txt:Git uses a simple text format to store customizations that are per
git.txt:Any Git command accepting any <object> can also use the following
git.txt:Various Git commands pay attention to environment variables and change
git.txt:The Git Repository
git.txt:These environment variables apply to 'all' core Git commands. Nb: it
git.txt:Git so take care if using a foreign front-end.
git.txt:    Due to the immutable nature of Git objects, old objects can be
git.txt:    of Git object directories which can be used to search for Git
```

```
git.txt:    Set the Git namespace; see linkgit:gitnamespaces[7] for details.
git.txt:    set, it is a list of directories that Git should not chdir up
git.txt:    command line or in the environment.  Normally, Git has to read
git.txt:    can add an empty entry to the list to tell Git that the
git.txt:    directory, Git tries to find such a directory in the parent
git.txt:    can be set to true to tell Git not to stop at filesystem
git.txt:Git Commits
git.txt:Git Diffs
git.txt:    value passed on the Git diff command line.
git.txt:    program named by it is called to generate diffs, and Git
git.txt:    to an empty string or to the value "cat", Git will not launch
git.txt:    It is used by several Git commands when, on interactive mode,
git.txt:    This environment variable overrides the configured Git editor
git.txt:    If this environment variable is set, it overrides Git's autodetection
git.txt:    tells Git not to verify the SSL certificate when fetching or
git.txt:    If this environment variable is set, then Git commands which need to
git.txt:    not set, Git will choose buffered or record-oriented flushing
git.txt:and lower than 10 (strictly) then Git will interpret this
git.txt:(starting with a '/' character), Git will interpret this
git.txt:    time of each Git command.
git.txt:    working directory after Git has completed its setup phase.
git.txt:and lower than 10 (strictly) then Git will interpret this
git.txt:(starting with a '/' character), Git will interpret this
git.txt:`af_unix:[<socket_type>:]<absolute-pathname>`, Git will try
git.txt:    By default, when tracing is activated, Git redacts the values of
git.txt:    Setting this Boolean environment variable to true will cause Git to treat all
git.txt:    literal paths to Git (e.g., paths previously given to you by
git.txt:    Setting this Boolean environment variable to true will cause Git to treat all
git.txt:    Setting this Boolean environment variable to true will cause Git to treat all
git.txt:    Setting this Boolean environment variable to true will cause Git to treat all
git.txt:    over lists of refs. Normally Git will try to include any such
git.txt:    When loading a commit object from the commit-graph, Git performs an
git.txt:    If this Boolean environment variable is set to false, Git will complete any requ
git.txt:    inherit them, possibly blocking regular Git operat
```

- There is no restriction

```
grep -r "Semiyari"
```

```
1_intro.qmd:subtitle: "Hamid Semiyari"
2_update_r_rstudio.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.qmd:subtitle: "Hamid Semiyari"
```

```
3_basic_bash.qmd:grep -r "Semiyari"
3_basic_bash.qmd:grep -ri "Semiyari"    # Not a case sensetive
3_basic_bash.rmarkdown:subtitle: "Hamid Semiyari"
3_basic_bash.rmarkdown:grep -r "Semiyari"
3_basic_bash.rmarkdown:grep -ri "Semiyari"    # Not a case sensetive
3_basic_bash.tex:\subtitle{Hamid Semiyari}
3_basic_bash.tex:\FunctionTok{grep} \AttributeTok{{-}r} \StringTok{"Semiyari"}
3_basic_bash.tex:1_intro.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:2_update_r_rstudio.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.qmd:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.qmd:grep -ri "Semiyari"    # Not a case sensetive
3_basic_bash.tex:3_basic_bash.rmarkdown:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.rmarkdown:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.rmarkdown:grep -ri "Semiyari"    # Not a case sensetive
3_basic_bash.tex:3_basic_bash.tex:\subtitle{Hamid Semiyari}
3_basic_bash.tex:3_basic_bash.tex:\FunctionTok{grep} \AttributeTok{{-}r} \StringTok{"Semiyari
3_basic_bash.tex:3_basic_bash.tex:1_intro.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.tex:2_update_r_rstudio.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -ri "Semiyari"    # Not a case senset
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:grep -ri "Semiyari"    # Not a case
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\subtitle{Hamid Semiyari}
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\FunctionTok{grep} \AttributeTok{{-}r} \St
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:1_intro.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:2_update_r_rstudio.qmd:subtitle: "Hamid Se
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:subtitle: "Hamid Semiyari
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -ri "Semiyari"    #
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:subtitle: "Hamid Se
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:grep -r "Semiyari"
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:grep -ri "Semiyari
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\subtitle{Hamid Semiyari
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\FunctionTok{grep} \Attri
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:1_intro.qmd:subtitle: "Ha
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:2_update_r_rstudio.qmd:su
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:subtitle
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -r
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.qmd:grep -ri
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:su
```

```
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:g
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.rmarkdown:g
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\subtit
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:\Functi
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:1_intro
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:2_updat
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:4_vcs_g
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:5_git_s
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:5_git_s
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6a_auth
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6b_2fa.
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6c_pat.
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6_githu
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6_githu
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:6_githu
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:7_git_b
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:7_git_b
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:authent
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:authent
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:authent
3_basic_bash.tex:3_basic_bash.tex:3_basic_bash.tex:3_ba
```

```
grep -r "semiyari"   # lower case 's'
```

```
.Rhistory:setwd("C:/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Week1")
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:))) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphic
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-scr
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-scri
```

```
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/am
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/amst
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/amsg
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/a
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/am
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsfonts/a
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsfonts/ams
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/iftex/i
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-ma
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3backend/l3
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-math
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3packages/x
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3packages
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/f
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/fon
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/fontenc
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/f
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/ts1enc
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-math
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/upquote/up
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/textcom
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/etoolbox/eto
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype/mi
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype/mi
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roamin
```

```
grep -ri "Semiyari"    # Not a case sensetive
```

```
.Rhistory:setwd("C:/Users/semiyari/Desktop/Personal/AU/DATA613/Spring24/Week1")
1_intro.qmd:subtitle: "Hamid Semiyari"
2_update_r_rstudio.qmd:subtitle: "Hamid Semiyari"
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
3_basic_bash.log:))) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-scri
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-script/
```

```
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/koma-scrip
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/am
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/amst
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/amsg
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/a
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsmath/am
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsfonts/a
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/amsfonts/ams
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/iftex/i
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-ma
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3backend/l3
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-matl
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3packages/i
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/l3packages
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/i
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/fon
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/fontenc
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fontspec/i
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/ts1enc
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/unicode-matl
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/upquote/up
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/base/textcom
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/etoolbox/etc
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype/mi
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/microtype/mi
3_basic_bash.log:)) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/xcolor/xc
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics-cfg
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics-def
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics/m
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics/c
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/xcolor/svg
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/xcolor/x11
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/fancyvrb/f
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/framed/fra
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/tools/long
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/booktabs/boo
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/tools/arra
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/tools/calc
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/mdwtools/m
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics/g
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics/gra
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics/tri
```

```
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/graphics-
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/caption/ca
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/caption/capt
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/caption/capt
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/caption/ltca
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/float/floa
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/caption/subc
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/tcolorbox/
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/pgf/basiclay
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/util
3_basic_bash.log:) (c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/util
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/latex/pgf/basiclay
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/utilit
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/utilit
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/systen
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/systen
3_basic_bash.log:(c:/Users/semiyari/AppData/Roaming/TinyTeX/texmf-dist/tex/generic/pgf/systen
```

Show line numbers with matching lines:

```
grep -n "Git" git_hist.txt
```

```
1:A Short History of Git
2:As with many great things in life, Git began with a bit of creative destruction and fiery 
10:Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain thes
```

Search for lines matching multiple patterns (logical OR):

```
grep -e "Speed" -e "Fully distributed" git_hist.txt
```

- Speed
- Fully distributed

```
grep -e "speed" -e "Fully distributed" -e "since" git_hist.txt  # both "s" are lowercase
```

- Fully distributed
- Able to handle large projects like the Linux kernel efficiently (speed and data size)
  - Does not matter were you put `-i` for the case sensitivity

```
grep -ie "speed" -e "Fully distributed" -ie "since" git_hist.txt  # both "s" are lowercase
```

```
•   Speed
•   Fully distributed
•   Able to handle large projects like the Linux kernel efficiently (speed and data size)
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
```

```
grep -e "speed" -e "Fully distributed" -ie "since" git_hist.txt  # both "s" are lowercase
```

```
•   Speed
•   Fully distributed
•   Able to handle large projects like the Linux kernel efficiently (speed and data size)
Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these
```

Search for lines matching all patterns (logical AND):

```
grep "develope" git_hist.txt | grep "linus"
```

```
grep "develope" git_hist.txt | grep -i "linus"
```

```
In 2005, the relationship between the community that developed the Linux kernel and the comm
```

WGET

- Download data with `wget`
- It is a free command-line tool that is used to download files from internet.
  - it provides "Recursive Downloads", meaning, `wget` loads requested document, then document linked fromthat document, and then the next, etc.
  - It follows lnks and directory structure.
  - It lets you "Overwrite" the links with correct domain, helping you create mirrors of website. (Replicas of website with different URL but host, or almost, identical content.)

Download `wget`

- Check if the `wget` is installed. Type

```
wget --version
```

If it is installed, it will return the version. Otherwise you need to install it.

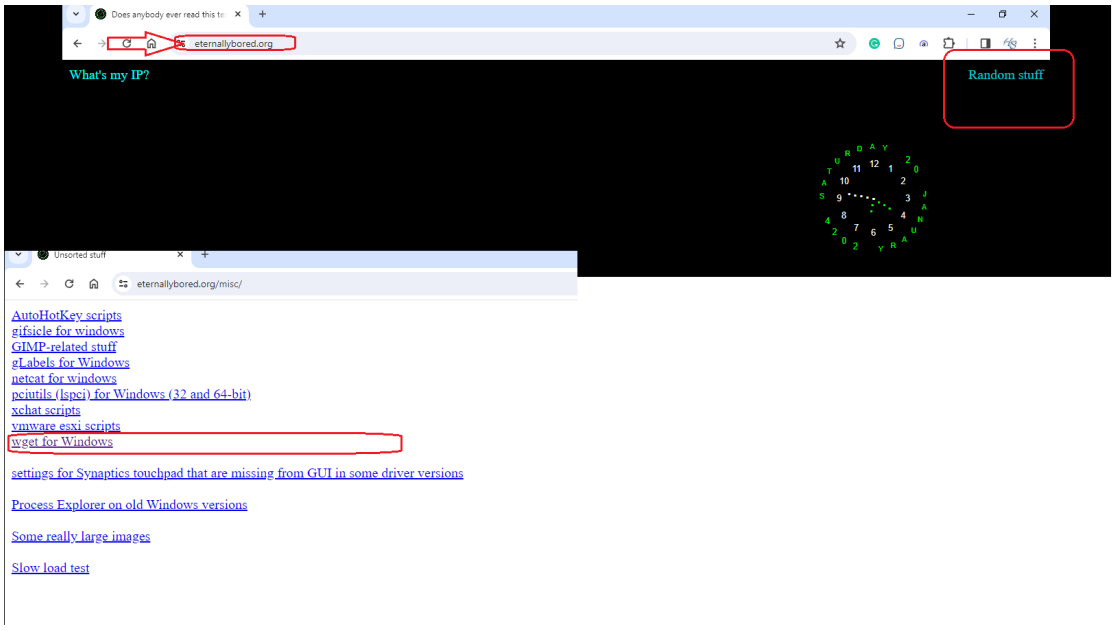
- Mac
  - Install with "Homebrew": On terminal type

    ```
    brew install wget
    ```

  - Install with "Macports": On terminal type

    ```
    sudo port install wget
    ```

- Windows:
  - Go to https://eternallybored.org/
  - Click "Random stuff" on top right, then select "wget for Windows"



+ Use the `EXE` (executable file) to download the file and save it. Simply copy the file to folder `C:\Windows\System32`

- **Question: What is the difference between "Arm64", "x86", and "x64"?**

  - Arm64 is used in SmartPhone, Apple Computers and some Servers.
  - X86 released in 1978 and its creator was "Intel". 32-Bit Architecture and a 32 bit system can handle a maximum of 2^{32} of RAM, which comes out to be 4GB of RAM.
  - X64 released in 2000 and its creator was "AMED". 64-Bit Architecture and 64 bit system can handle a maximum of 2^{64} of RAM, which comes out to be 16 Exabytes (EB) of RAM which is two levels up from Terabyte (TB).

Therefore x64 allows the CPU to store more data and access it faster.

- Check if `wget` is installed

```
wget -V   #Uppercase V
```

or

```
wget --verson
```

- The command `wget -h` or `wget --help` gives you all possible flags you need with `wget`

Non-interactive downloading of data.

- `-nc` Don't download new copies if already there.
- `-nd` Put all files in current working directory.
- `-P` Tell where to download the files. Default is current working directory (`.`)
- `-r` Recursive downloading. Download all files in the directory up to a certain level.
- `-l` Determine the level for recursive downloading.
- Let us download "git.txt" from "https://github.com/git/git/blob/master/Documentation/git.txt". But, that GitHub serves raw content through a different URL. On GITHUB click on the `Raw` bottom on the top right corner to get "https://github.com/git/git/blob/master/Documentation/git.txt

```
wget https://raw.githubusercontent.com/git/git/master/Documentation/git.txt
```

```
--2024-02-13 18:16:51--  https://raw.githubusercontent.com/git/git/master/Documentation/git.t
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... c
HTTP request sent, awaiting response... 200 OK
Length: 41884 (41K) [text/plain]
Saving to: 'git.txt.9'

     0K ......... ......... ......... .........           100% 2.80M=0.01s
```

```
2024-02-13 18:16:51 (2.80 MB/s) - ‘git.txt.9’ saved [41884/41884]
```

- You can use the `cat` command to display the contents of the downloaded “git.txt” file.

```
cat git.txt
```

- The above command will download the “git.txt” file from the specified URL. It will be saved in your workingdirectory
- If you want to save it with a different name, you can use the -O option:

```
wget -O MYGIT.txt https://raw.githubusercontent.com/git/git/master/Documentation/git.txt
```

```
--2024-02-13 18:16:51--  https://raw.githubusercontent.com/git/git/master/Documentation/git.t
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.1
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... c
HTTP request sent, awaiting response... 200 OK
Length: 41884 (41K) [text/plain]
Saving to: ‘MYGIT.txt’

     0K ......... ......... ......... .........           100% 2.58M=0.02s

2024-02-13 18:16:52 (2.58 MB/s) - ‘MYGIT.txt’ saved [41884/41884]
```

```
cat MYGIT.txt
```

- `-P`: Save files to the specified directory.

```
wget -P "$(dirname "$(pwd)")/f1" https://raw.githubusercontent.com/git/git/master/Document
```

This command uses `$(dirname "$(pwd)")` to get the parent directory of the current working directory and appends “/wg_files” to it as the target directory. This way, it avoids issues with relative paths. Note: if there is not such a folder “wg_files” then it will create and save the file “git.txt” in it.

Now, let’s say you want to download all files from a website recursively. For this example, I’ll use the GNU Wget test site:

```
# Recursive download
wget -r <Enter URL>
```

- In this command:
- `-r` is for recursive download.
- `--no-parent` ensures that `Wget` doesn't ascend to the parent directory when downloading.
- This command will download the entire content of the specified URL, including all files and subdirectories, into the current working directory. The structure of the website will be replicated locally.
- Please note that downloading an entire website may take a significant amount of time and bandwidth, and you should ensure that you have permission to do so. It's also good practice to check the website's `robots.txt` file to make sure you're not violating any access policies.
- The `robots.txt` file must be located at the root of the site host to which it applies. For instance, to control crawling on all URLs below https://www.example.com/ , the `robots.txt` file must be located at https://www.example.com/robots.txt .
- E.g. to download the HTML file that contains the Wikipedia list of theological demons, we can go

```
wget -nc -nd https://en.wikipedia.org/wiki/List_of_theological_demons
```

```
--2024-02-13 18:16:52--  https://en.wikipedia.org/wiki/List_of_theological_demons
Resolving en.wikipedia.org (en.wikipedia.org)... 208.80.154.224, 2620:0:861:ed1a::1
Connecting to en.wikipedia.org (en.wikipedia.org)|208.80.154.224|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 92641 (90K) [text/html]
Saving to: 'List_of_theological_demons'

     0K .......... .......... .......... .......... .......... 55% 1.65M 0s
    50K .......... .......... .......... ..........           100% 3.61M=0.04s

2024-02-13 18:16:52 (2.18 MB/s) - 'List_of_theological_demons' saved [92641/92641]
```

```
ls
```

```
1_intro.qmd
2_update_r-rstudio_files
2_update_r_rstudio.pdf
2_update_r_rstudio.qmd
```

```
3_basic_bash.aux
3_basic_bash_files
3_basic_bash.log
3_basic_bash.pdf
3_basic_bash.qmd
3_basic_bash.rmarkdown
3_basic_bash.tex
4_vcs_git.pdf
4_vcs_git.qmd
5_git_setup.pdf
5_git_setup.qmd
6a_authentication_github_ssh.pdf
6a_authentication_github_ssh.qmd
6b_2fa.pdf
6b_2fa.qmd
6c_pat.pdf
6c_pat.qmd
6_github_setu_files
6_github_setup.pdf
6_github_setup.qmd
7_git_basic_files
7_git_basic.html
7_git_basic.qmd
8a1_ssh_pat.html
8a1_ssh_pat.pdf
8a1_ssh_pat.qmd
8a2_init_repo.qmd
8a_ssh_pat_files
8b_push.qmd
8c_pull_fork.qmd
8d_move_remove.qmd
8_git_github.qmd
9_practice.qmd
Authentication_github_PAT.qmd
authentication-pat_files
authentication-pat.html
BasicBash.qmd
basic-usage.png
Exercise-Bash_files
Exercise-Bash.html
Exercise-Bash.qmd
file1.txt
Git_1.jpg
```

```
Git_2.jpg
Git_3.jpg
gitbasic_commit_push_pull_files
gitbasic_commit_push_pull.html
gitbasic_commit_push_pull.qmd
git_github_basicGit_files
git_github_basicGit.html
git_github_basicGit.qmd
git_hist.txt
git_setup.qmd
git.txt
git.txt.1
git.txt.2
git.txt.3
git.txt.4
git.txt.5
git.txt.6
git.txt.7
git.txt.8
git.txt.9
init_repo_files
 is used to append to a file.
List_of_theological_demons
MYGIT.txt
NotePractice.R
nul
pat_clone.qmd
practice_open_data.R
raw.githubusercontent.com
ssh_pat_2fa.qmd
```

I'll remove that file now

```
rm List_of_theological_demons
```

RREVERSE-I-SEARCH

Search your command history with `reverse-i-search`

- Search your command history

1. `ctrl+r` to get search prompt
2. Type a search term
3. `ctrl+r` to navigate through matches

Here's an example:

On your Bash

Let's say you want to find a "git" command you used recently. You would:

Press `Ctrl + R` Type "git" (or any part of the command) Press `Ctrl + R` repeatedly to cycle through matches Once you find the correct command, press Enter to execute it.

A few shortcut:

1. Type `explorer` in Bash/Terminal to open "File Explorer". Or on keyboard hold window button and click `E`
2. Type a few letters on Terminal/Bash and then click on `tab` button to get the possible match. (Once for terminal-Twice for Bash)
3. To interrupt the current command press `CTRL+C` .
4. Type `clear` to clear the Terminal/Bash page
5. Type `exit` and press ENTER
6. Use `alt+shift+R` to open terminal-pane

- **What is difference between >> and > in Bash/Terminal?**

This symbol > overwrites the file if it exists or creates it if it doesn't exist.

```
echo "The `>` overwrites the file if it exists or creates it if it doesn't exist." > file1
```

```
/bin/bash: command substitution: line 1: syntax error near unexpected token `newline'
/bin/bash: command substitution: line 1: `>'
```

```
cat file1.txt
```

```
The  overwrites the file if it exists or creates it if it doesn't exist.
```

```
echo "This will overwrite the file and what we had before has been replaced with this line
```

```
cat file1.txt
```

```
This will overwrite the file and what we had before has been replaced with this line
```

This `>>` is used to append to a file.

```
echo "This symbol \">>\" is used to append to a file." >> file1.txt
```

if you run `cat file1.txt` you will see both lines are in file1.txt

```
cat file1.txt
```

```
This will overwrite the file and what we had before has been replaced with this line
This symbol ">>" is used to append to a file.
```