<https://math2001.github.io/post/bashs-find-command/>

find: output all path of current directory and file

find filename.extension

* find the diredctory or file.ext

find -type f

* f : regular file
* d: directory

find -name \*.txt

* takes global file
* iname : case insensitive

find -path

-same as name, but not just filename, also path

- portable version of “wholename”

find -and, find -or, find -not

* or can be use to conjuct two different name -name \*.txt -or -name \*.cp
* -and is default
* -not can be use to filter file not contain that argument

grouping ()

* we can group argument together
  + make sure to use \ to escape from bash

comma ,

* Separates 2 expressions: it evaluates both of them, but only returns the value of the second one

alias

* we can assign operator to some other name alias
  + -o = -or

-print:

* the default action of find is printz

-delete:

- use it to delete file to the bin directory

- find -name "\*~" -delete

delete temperary file created by vim

-exec: exectute the command

- find -name "\*~" -exec rm {} \;

- exectue rm to the temporary file

1. Find all files in /bin, /sbin, /usr/bin, and /usr/sbin that are setuid and owned by root. Why are these files potential security risks?
   1. find /sbin /bin /usr/bin /usr/sbin -user root -perm -4000
2. Find all files across the entire system that have setuid or setgid enabled (regardless of owner).
   1. find / -perm -4000 -or -perm -2000
3. Find all files in /var that have changed in the last 20 minutes.
   1. find /var -cmin -20
4. Find all files in /var that are regular files of zero length.
   1. find /var -type f -size 0
5. Find all files in /dev that are not regular files and also not directories. The same command should print a listing that includes permissions and modification times (at a minimum) for these files.
   1. find /dev -not -type f -and -not -type d -ls -printf %a
6. Find all directories in /home that are not owned by root. In the same command, change their permissions to ensure they have 711 (-rwx--x--x) permissions.
   1. find /home -type d -not -user root -exec chmod 711 {} \;
7. Find all regular files in /home that are not owned by root. In the same command, change their permissions to ensure they have 755 (-rwx-wx-wx) permissions.
   1. find /home -type f -not -user root -exec chmod 755 {} \;
8. Find all files (of all types) in /etc that have changed in the last 5 days.
   1. find /etc -ctime -5