GNU screen terminal commands cheat sheet

Command	Description
~/.screenrc & **/etc/screenrc**	Commands the screen runs on start up.
screen -ls	List active screen sessions
screen -Q windows	List windows' names inside screen session
screen -S <session name>	Create a new screen session with the name <session name>
screen -x **screen -r <session name>**	Attach to the running session, also by its name
screen -dRR	Attach to the screen session, detach on other display if attached. If no session exists, will create a new one.
C-a d	Detach from the session, session keeps running. Here, and further **C** means Ctrl.
C-a c	Create new window in the session.
C-a C-a	Switch to the previous window.
C-a <number>	Switch to the window number `number`.
C-a '	Switch to the window by its name.
C-a n	Switch to the next window.
C-a p	Switch to the previous window.
C-a "	List all windows with option to highlight and enter any of them.
exit	Exit and close current window. If it was the last window in a session, exits `screen` terminating the session.
C-a k	Kill the current window forcefully (not recommended).
C-a : quit	Quit screen session compeltely terminating it. Alternatively - exit all screen windows.
C-a A	Rename current window.
C-a S	Split windows display horizontally. Use **C-a c** to create a new window inside the new split.

Command	Description
C-a V	Split windows display vertically. Available starting screen 4.01, i.e. not available on Mac 2020 which still uses screen 4.00.
C-a tab	Jump to the next region in a split window display.
C-a X	Remove the region in focus.
C-a [or **C-a <esc>**	Enter buffer navigation mode to scroll output buffer, copy, edit and paste later. Navigation commands as per `vim` if Vim is set as editor.**<esc>** to leave the buffer mode.
<space>	Start/stop selection while in the buffer mode to select the text. All selected text is being copied to the clipboard automatically. E.g. to select/copy the whole buffer: `C-a [ gg <space> G <esc>`
C-a]	Paste the selected text at the cursor of the terminal, or create a new window and say start Vim there and paste into it while in Insert mode.
C-a h	Screenshot as a text the currently visible terminal window and save the output to `hardcopy.<n>`, where <n> is auto-incrementing number of your window.
C-a H	Start/end logging all output of the curent window into a file `screenlog.N` where `N` is the window number. The data is appended, not overwritten if the file exists. Contents displayed until enabling the logging is not logged.
C-a a	Send `Ctrl-a` sequence to the running command, useful to jump to the line start in bash.
C-a M	Monitor window for activity. When enabled, will notify you of any acitvity while you work in other window.
C-a _	Monitor window for 30 seconds of silence, will notify you in any other window as `Window 0: silence for 30 seconds `
C-a ?	Show all key bindings help.
Save session state	This is not possible. If you use the same layout each session, you can put start up commands to re-create it in `.screenrc` file in your home directory, but still - you cannot save the current session state, i.e. contents of the windows and their layout.

Command	Description
Sharing session (e.g. for pair programming)	
Original session (say *user1*): 1. Set suid root bit on `screen` binary: `sudo chmod +s /usr/bin/screen` 2. Inside session you want to share: `C-a :` then `multiuser on` to enable sharing session. 3. Add usernames to share the session with: `C-a : acladd <username>` Connecting user (say *user2*): 1. Run in shell: `screen -x <sharing username>/`, in our example `screen -x user1/`	Sets up sharing the session. Another user connecting to the session views real-time its output, can enter and run commands himself. Also see **aclchg**, **acldel**, **aclgrp** for controlling what the connecting user can and cannot do. E.g. to remove *write* permissions from all users on all windows: `:aclchg * -w #`
C-a *	See who is connected to your shared screen session.