

# GNU screen terminal commands

## cheat sheet

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

Command	Description
<b>C-a V</b>	Split windows display vertically. Available starting screen 4.01, i.e. not available on Mac 2020 which still uses screen 4.00.
<b>C-a tab</b>	Jump to the next region in a split window display.
<b>C-a X</b>	Remove the region in focus.
<b>C-a [ or C-a &lt;esc&gt;</b>	Enter buffer navigation mode to scroll output buffer, copy, edit and paste later. Navigation commands as per <b>vim</b> if Vim is set as editor. <b>&lt;esc&gt;</b> to leave the buffer mode.
<b>&lt;space&gt;</b>	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: <b>C-a [ gg &lt;space&gt; G &lt;esc&gt;</b>
<b>C-a ]</b>	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.
<b>C-a h</b>	Screenshot as a text the currently visible terminal window and save the output to <b>hardcopy.&lt;n&gt;</b> , where <b>&lt;n&gt;</b> is auto-incrementing number of your screen session.
<b>C-a a</b>	Send <b>Ctrl-a</b> sequence to the running command, useful to jump to the line start in bash.
<b>C-a M</b>	Monitor window for activity. When enabled, will notify you of any activity while you work in other window.
<b>C-a _</b>	Monitor window for 30 seconds of silence, will notify you in any other window as <b>`Window 0: silence for 30 seconds`</b>
<b>C-a ?</b>	Show all key bindings help.
<b>Save session state</b>	This is not possible. If you use the same layout each session, you can put start up commands to re-create it in <b>.screenrc</b> file in your home directory, but still - you cannot save the current session state, i.e. contents of the windows and their layout.
<b>Sharing session (e.g. for pair programming)</b>	

Command	Description
<p>Original session (say <i>user1</i>):</p> <ol style="list-style-type: none"> <li>1. Set suid root bit on <b>screen</b> binary: <code>sudo chmod +s /usr/bin.screen</code></li> <li>2. Inside session you want to share: <b>C-a :</b> then <b>multiuser on</b> to enable sharing session.</li> <li>3. Add usernames to share the session with: <b>C-a : acladd &lt;username&gt;</b></li> </ol> <p>Connecting user (say <i>user2</i>):</p> <ol style="list-style-type: none"> <li>1. Run in shell: <code>screen -x &lt;sharing username&gt;/</code>, in our example <code>screen -x user1/</code></li> </ol>	<p>Sets up sharing the session. Another user connecting to the session views real-time its output, can enter and run commands himself. Also see <b>aclchg</b>, <b>acldel</b>, <b>aclgrp</b> for controlling what the connecting user can and cannot do. E.g. to remove <i>write</i> permissions from all users on all windows: <code>:aclchg * -w #</code></p>
<b>C-a *</b>	See who is connected to your shared screen session.