



# Persistent Remote Shells with screen

Serge Y. Stroobandt

Copyright 2014–2022, licensed under [Creative Commons BY-NC-SA](#)

## Connecting with ssh

Connect to the remote host computer using `ssh`:

```
$ ssh username@hostname.url:port
```

## Evoking screen

Screen sockets and screen sessions are one and the same thing. By default, screen sessions are named according to the scheme `pid.tty.hostname`. Here are a number of commands to evoke `screen` from a normal shell.

**Table 1: Commands to evoke screen from a terminal**

command	function
\$ <code>screen -S sessionname</code>	Start a new screen session with a given name and attach to it.
\$ <code>screen -ls</code>	Lists all existing screen sessions.
\$ <code>screen -r</code>	Reattach* this terminal to the only existing screen session.
\$ <code>screen -r -S sessionname</code>	Reattach* this terminal to an existing screen session by name.
\$ <code>screen -r ttynumber</code>	Reattach* this terminal to an existing screen session by tty number.
\$ <code>screen -S oldsessionname -X newsessionname</code>	Rename a screen session.
\$ <code>screen -S sessionname -X quit</code>	Kill the specified screen session by executing <code>quit</code> .

**Note:**

\* If there exists only one screen session, it is not necessary to specify any name; just type `screen -r` at the command line.

# Within a session

All commands that can be issued within a screen session start by hitting the **Ctrl** + **A** key combination, followed by another keystroke.

**Table 2: Keystroke commands for use within a screen session**

keystroke sequence	function
<b>Ctrl</b> + <b>A</b> ?	Show the <b>help</b> screen.
<b>Ctrl</b> + <b>A</b> d	<b>Detach</b> * the current screen session from this terminal. The screen session and its processes remain in existence.
<b>Ctrl</b> + <b>A</b> c	<b>Create</b> a new window and switch to it.
<b>Ctrl</b> + <b>A</b> C	<b>Clear</b> the screen.
<b>Ctrl</b> + <b>A</b> K	<b>Kill</b> the current window.
<b>Ctrl</b> + <b>A</b> p	Go to the <b>previous</b> screen.
<b>Ctrl</b> + <b>A</b> n	Go to the <b>next</b> screen.
<b>Ctrl</b> + <b>A</b> #	Go to a specific screen <b>number</b> .
<b>Ctrl</b> + <b>A</b> "	<b>Select</b> a screen from a list using the arrow keys.
<b>Ctrl</b> + <b>A</b> S	<b>Split</b> the current window in half horizontally.
<b>Ctrl</b> + <b>A</b>	<b>Split</b> the current window in half vertically.
<b>Ctrl</b> + <b>A</b> <b>Tab</b>	<b>Cycle</b> through window regions.
<b>Ctrl</b> + <b>A</b> X	<b>Eliminate</b> a window split.
<b>Ctrl</b> + <b>A</b> H	<b>Log</b> the current screen sessions in logfiles called <code>screenlog.n</code> where n is the number of the screen session.
<b>Ctrl</b> + <b>A</b> h	Create a <b>screenshot</b> of the current window in a file called <code>hardcopy.n</code> .
<b>Ctrl</b> + <b>A</b> x	<b>Lock</b> the screen session with the password of the current user.
<b>Ctrl</b> + <b>A</b> <b>Esc</b>	<b>Enter</b> the copy and <b>scrollback</b> mode.
<b>Esc</b>	<b>Leave</b> the copy and <b>scrollback</b> mode.

**Note:**

\* Closing the ssh connection also results in a detachment of the screen session.

Typing `exit` at the command line of a screen session will destroy that screen session.

# Where am I?

Experiencing an *Inception* moment? Of course, you can always hit `Ctrl`+`A` ? and see if you get the screen help information. Another way is to let the environment variable `$TERM` tell you what terminal type you are currently employing.

Inside a screen session:

```
$ echo $TERM
screen.xterm-256color
```

Inside a normal terminal:

```
$ echo $TERM
xterm-256color
```



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Other licensing available on request.

Unattended [CSS](https://www.howtotip.com/css-typesetting-with-prince/) typesetting with . The Prince logo features the word "Prince" in a stylized purple font with "Print with CSS" written in smaller text above it.

This work is published at <https://hamwaves.com/screen/en/>.

Last update: Sunday, August 21, 2022.