

---

# Yutuber Documentation (more detailed)

---

## 1. A Quick Command Reference

---

- These all play off of the ctrl-b shortcut.

### 1. Basics

**?** Get help

### 2. Session management

**s** list sessions

**\$** rename the current session

**d** detach from the current session

### 3. Windows

**c** create a new window

**,** rename the current window

**w** list windows

**%** split horizontally

**"** split vertically

**n** change to the next window

**p** change to the previous window

**0 to 9** select windows 0 through 9

### 4. Panes

**%** create a horizontal pane

**"** create a vertical pane

**h** move to the left pane \*

**j** move to the pane below \*

**l** move to the right pane \*

**k** move to the pane above \*

**q** show pane numbers

**o** toggle between panes

- } swap with next pane
- { swap with previous pane
- ! break the pane out of the window
- x kill the current pane

## 2. TMUX

---

Tmux lets you keep things running persistently on servers, so you can disconnect and connect as needed without interrupting tasks that are in progress.

### 1. Installation

- macOS:

```
brew install tmux
```

- Ubuntu Linux

```
apt install tmux
```

### 2. Creating a session

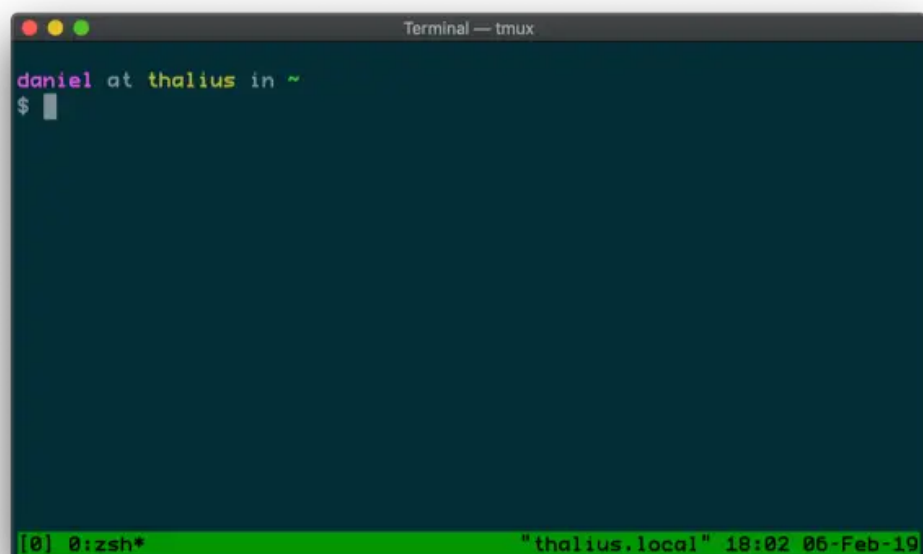
- If you invoke tmux by itself, you'll get dropped into a new session. And if you exit this session, you drop right back into your normal shell.
- The ctrl-b shortcut is good to memorize.

```
tmux
```

- Creating named sessions lets you reattach more easily.

```
tmux new -s session-name
```

- Fig: A Raw TMUX Session



### 3. Detaching from a session

- You want a pause between the shortcut and the following command.  
To detach from a session you invoke the shortcut (ctrl-b)—followed by d, for detach, or by typing detach explicitly.

```
ctrl+b-d
```

```
tmux detach
```

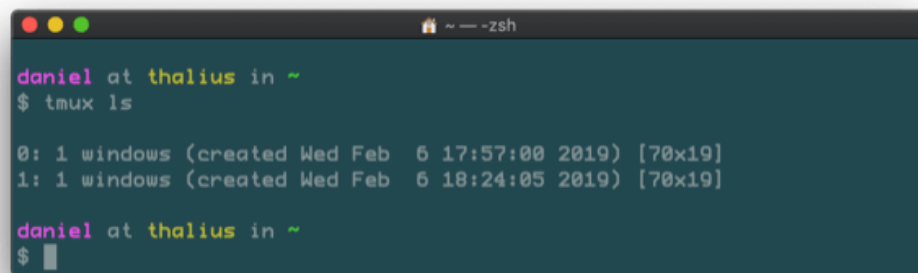
### 4. Show existing sessions

- You can and often will have multiple tmux sessions on a single system, so you want to be able to see what they are. You can also show sessions using the shortcut:

```
ctrl+b-s
```

```
tmux ls
```

- Fig: A view of running sessions



### 5. Attaching to an existing session

- Now that we can see those sessions, you can either connect to one by session name, or by number. The session names start at 0 and increment upwards.

```
tmux attach -t 0
```

- tmux a will connect you to the first available session.

```
tmux attach -t session-name
```

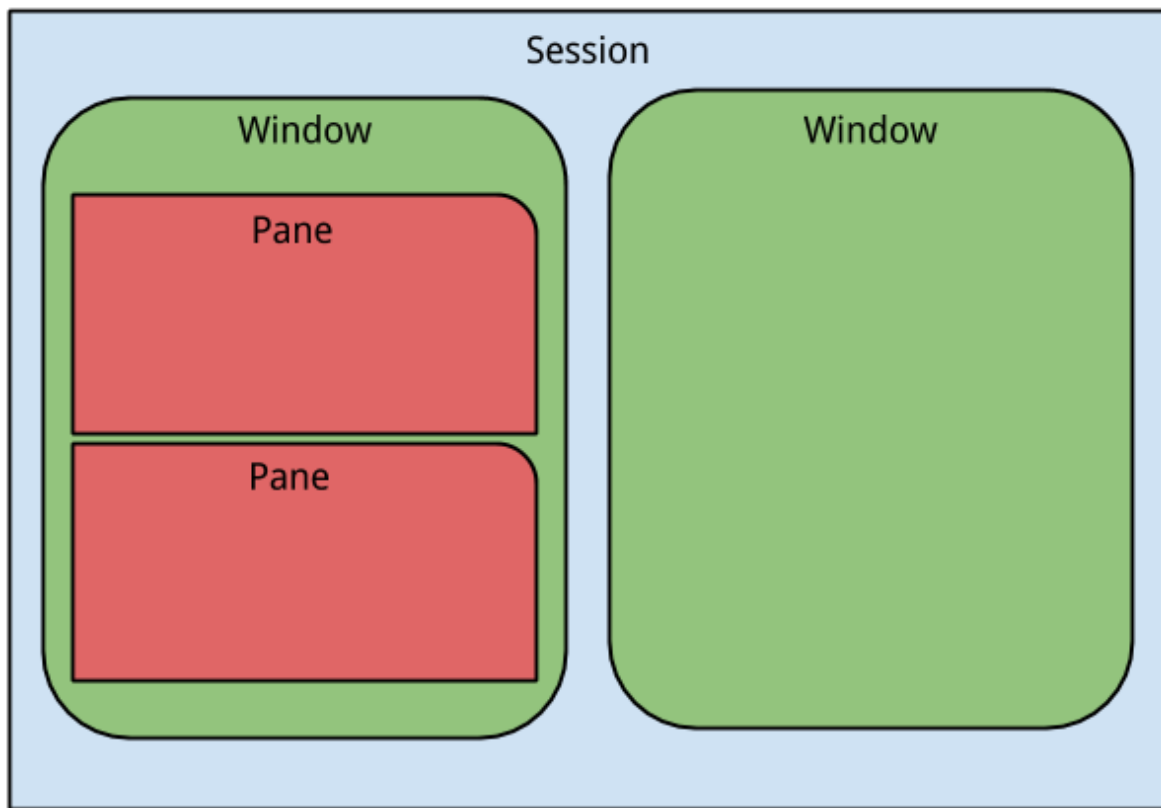
### 6. Killing a session

- There are times when you'll want to destroy a session outright, and that can be done similar to attaching to one.

```
$ tmux kill-session -t session-name
```

- You can also kill tmux altogether with:

## 7. Windows and Panes (tmux nesting)



I don't use this functionality myself.

- Another feature of tmux is the ability to break your session into more discreet components, called windows and panes. These are good for organizing multiple activities in a logical way.
- Basically, tmux sessions have windows, and windows have panes. Here's how I conceptualize the structure.
- **Sessions** are for an overall theme, such as work, or experimentation, or sysadmin.
- **Windows** are for projects within that theme. So perhaps within your experimentation session you have a window titled noderestapi, and one titled lua sample.
- **Panes** are for views within your current project. So within your sysadmin session, which has a logs window, you may have a few panes for access logs, error logs, and system logs.

## 8. Parting thoughts

Here are a few tips I've picked up over the years using tmux.

- Consider using *as few sessions and windows as possible*. Humans aren't as good at multitasking as we think we are, and while it feels powerful to have 47 panes open it's usually not as functional as you'd imagine.
- When you do use windows and panes, take the time to *name them*. They are indeed useful, but switching between sessions and windows is supremely annoying when they're all labeled 0, 1, and 2.
- *Start with a basic config and get used to it* before you get silly with it. I've seen multiple people spend hours configuring vim or tmux only to confuse themselves and abandon the project altogether. Start simple.

- tmux is a lot **like screen, only better**. The short answer for how it's better is that tmux is:
  1. Tmux is built to be truly *client/server*; screen emulates this behavior
  2. Tmux supports both *emacs and vim shortcuts*
  3. Tmux supports *auto-renaming windows*
  4. Tmux is *highly scriptable*
  5. *Window splitting* is more *advanced* in tmux

---

# My Documentation

---

## 1. TMUX Window Operations

---

1. To start **TMUX**:

`tmux`

2. To create a new window:

hit **Ctrl+B**, then press **c**

3. To close a pane or window:

hit **Ctrl+D** or input **exit** and hit **Enter**

4. To rename a window:

hit **Ctrl+B**, then press **comma(,)**, Enter the **name** of the window and press **Enter**

5. To switch window:

hit **Ctrl+B**, then press **p** for previous and **n** for next window

6. To list and select window:

hit **Ctrl+B**, then press **w** to list windows  
select desired window by scrolling **up** or **down** and press **Enter**

---

## 2. TMUX Pane Operations

---

1. To split windows vertically:

hit **Ctrl+B**, then press **%**

2. To split windows horizontally:

hit **Ctrl+B**, then press **Colon(:)**, then enter **split-window** and press **Enter**

3. To close all windows:

Either enter **exit** or close the window by **cross button**

---

## 3. Sessions

---

1. To start a new session:

```
tmux new -s <session-name>
```

2. To detach from a session:

hit **Ctrl+B**, then press **D**

3. To list running sessions, enter command:

```
tmux list-sessions
```

4. To reattach to a running session, enter command:

```
tmux attach -t <session-name>
```