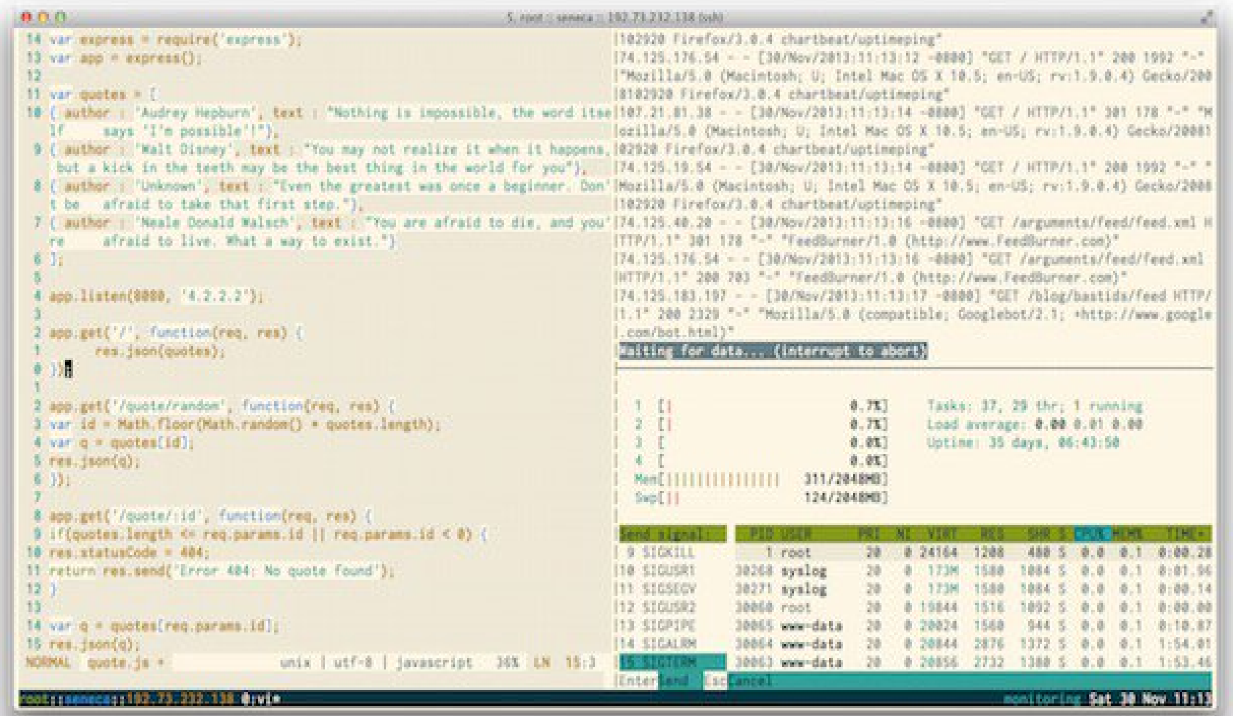
A tmux Primer

Home » Study » A tmux Primer

Site Sponsor: Netsparker — find vulnerabilities in your web applications before someone else does it for you.



```
var express = require('express');
var app = express();

var quotes = [
{ author : 'Audrey Hepburn', text : "Nothing is impossible, the word itse
lf    says 'I'm possible'!"},
{ author : 'Walt Disney', text : "You may not realize it when it happens,
 but a kick in the teeth may be the best thing in the world for you"},
{ author : 'Unknown', text : "Even the greatest was once a beginner. Don'
t be   afraid to take that first step."},
{ author : 'Neale Donald Walsch', text : "You are afraid to die, and you'
re    afraid to live. What a way to exist."}
];

app.listen(8080, '4.2.2.2');

app.get('/', function(req, res) {
      res.json(quotes);
});

app.get('/quote/random', function(req, res) {
var id = Math.floor(Math.random() * quotes.length);
var q = quotes[id];
res.json(q);
});

app.get('/quote/:id', function(req, res) {
if(quotes.length <= req.params.id || req.params.id < 0) {
res.statusCode = 404;
return res.send('Error 404: No quote found');
}

var q = quotes[req.params.id];
res.json(q);
NORMAL  quote.js +                 unix | utf-8 | javascript  36%  LN  15:3
```

```
102920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.176.54 - - [30/Nov/2013:11:13:12 -0800] "GET / HTTP/1.1" 200 1992 "-"
"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/200
8102920 Firefox/3.0.4 chartbeat/uptimeping"
107.21.81.38 - - [30/Nov/2013:11:13:14 -0800] "GET / HTTP/1.1" 301 178 "-" "M
ozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/20081
02920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.19.54 - - [30/Nov/2013:11:13:14 -0800] "GET / HTTP/1.1" 200 1992 "-" "
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/2008
102920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.40.20 - - [30/Nov/2013:11:13:16 -0800] "GET /arguments/feed/feed.xml H
TTP/1.1" 301 178 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
74.125.176.54 - - [30/Nov/2013:11:13:16 -0800] "GET /arguments/feed/feed.xml
HTTP/1.1" 200 703 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
74.125.183.197 - - [30/Nov/2013:11:13:17 -0800] "GET /blog/bastida/feed HTTP/
1.1" 200 2329 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google
.com/bot.html)"
Waiting for data... (interrupt to abort)
```

```
  1  [|                        0.7%]     Tasks: 37, 29 thr; 1 running
  2  [|                        0.7%]     Load average: 0.00 0.01 0.00
  3  [                         0.0%]     Uptime: 35 days, 06:43:50
  4  [                         0.0%]
  Mem[||||||||||||||||    311/2048MB]
  Swp[||                  124/2048MB]

Send signal:     PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+
 9 SIGKILL         1 root       20   0 24164  1200   480 S  0.0  0.1  0:00.28
10 SIGUSR1     30268 syslog     20   0  173M  1580  1084 S  0.0  0.1  0:01.56
11 SIGSEGV     30271 syslog     20   0  173M  1580  1084 S  0.0  0.1  0:00.14
12 SIGUSR2     30060 root       20   0 19844  1516  1092 S  0.0  0.1  0:00.00
13 SIGPIPE     30065 www-data   20   0 20024  1560   944 S  0.0  0.1  0:10.87
14 SIGALRM     30064 www-data   20   0 20844  2876  1372 S  0.0  0.1  1:54.01
15 SIGTERM     30063 www-data   20   0 20856  2732  1380 S  0.0  0.1  1:53.46
Enter Send   Esc Cancel
```

```
root::seneca::192.73.232.138 0:vi*                         monitoring Sat 30 Nov 11:13
```

[Every Sunday I put out a curated list of the most interesting stories in infosec, technology, and humans. I do the research, you get the benefits.Subscribe.]

There are 4,257 tutorials on tmux. That’s a rough number that I just made up. This one is designed to take you from “wtf tmux” to “omg tmux” with extreme haste.

Let’s get started.

WHY TMUX

`tmux` is useful to people in different ways. To me, it’s most useful as a way to maintain persistent working states on remote servers—allowing you to detach and re-attach at will.

You could, for example, have a session on your server for hacking on a node REST API (my current project), and call it "nodeapi". And let us say that you are compiling something for it that will take two hours (work with me), but you're currently working at a coffee shop and you have to leave. `tmux` lets you simply detach from that session and come back to it later.

That's handy.

Others like to focus on how you can use tmux to have multiple panes within multiple windows within multiple tabs within multiple sessions. I don't do that. I like fewer of those—as few as possible, actually—and this guide will be focused on a simple *persistent-remote-sessions* model.

A remote computing lifestyle

Mobility is a central theme for tmux users. Many developers do all of their work from the server, and simply connect in from `$wherever` to do it. `tmux` (and similar tools) allow you to work from a coffee shop in SF, start something building on the server, disconnect to take a flight, and then pick up that same task on the ground in NYC when you land.

A related advantage to this mobile approach is the fact that your client machine is not too terribly important. You can upgrade your laptop, clone a repo with your vim/tmux dotfiles in it, and you're back to your optimum computing environment in minutes rather than days.

Anyway, those are some reasons that people love tmux, but you don't have to make this lifestyle change in order to see its benefits.

WHAT ABOUT SCREEN?

Good question. `tmux` is a lot like screen, only better. The short answer for *how* it's better is that tmux is better designed to perform the same functions. Screen gets you there (kind of) but does so precariously.

Here are a few of the key advantages of tmux over screen:

- Tmux is built to be truly client/server; screen emulates this behavior
- Tmux supports both emacs and vim shortcuts
- Tmux supports auto-renaming windows
- Tmux is highly scriptable
- Window splitting is more advanced in tmux

Enough about that. Use tmux.

BASICS

Now is a good time to mention that there is a universal tmux shortcut that lets you quickly perform many tasks.

THE `TMUX` SHORTCUT

By default, tmux uses `Ctrl-b` as its shortcut activation chord, which enables you perform a number of functions quickly. Here are a few of the basics:

First you hit:

```
$ Ctrl-b
```

…followed by a number of options that we'll talk about below. But get ready to use that `Ctrl-b` combo. Also consider remapping CAPSLOCK to CONTROL within your operating system; it makes the pinky walk for `Ctrl-b` quite nice.

INVOCATION

Right then. Let's start by running `tmux`. You want to do this from the system that you want to detach and re-attach to—which for me is usually a remote server.

```
$ tmux
```

Simple enough. You now have a tmux session open that you can disconnect from and come back to later.

SHOW SESSIONS

Since the idea of tmux is having multiple sessions open, and being able to disconnect and reconnect to them as desired, we need to be able to see them quickly.

```
# Via shortcut (by default Ctrl-b)

$ Ctrl-b s

# Via tmux command

$ tmux ls
```

Either way you get the same thing:

```
0: 1 windows (created Thu Nov 28 06:12:52 2013) [80x24] (attached)
```

CREATE A NEW SESSION

Now we're going to create a new session. You can do this with just the *new* command, or by providing an argument to it that serves as the session name. I recommend providing a session name, since organization is rather the point of tmux.

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

```
# Without naming the new session (not recommended)
```

```
$ tmux new
```

ATTACHING TO AN EXISTING SESSION

Since we’re going to be creating sessions with names, and we may have more than one, we want to be able to attach to them properly. There are a couple ways of doing this.

You can simply type `tmux a` and it’ll connect you to the first available session.

```
$ tmux a
```

Or you can attach to a specific session by providing an argument.

```
$ tmux a -t session-name
```

DETACHING FROM A SESSION

You can detach from an existing session (so you can come back to it later) by sending the *detach* command.

```
$ tmux detach
```

Or you can use the shortcut.

```
$ Ctrl-b d
```

KILLING A SESSION

There are times when you'll want to destroy a session. This can be done using the following syntax, which is much the same as attachment:

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

[**NOTE**: You can kill windows the same way, but using `kill-window` instead. You can also kill tmux altogether with `killall tmux`.]

CONFIGURATION

As with most things in tech, you can get pretty silly with your tmux config. The common things to tinker with are:

- The primary `tmux` shortcut
- Your status bar
- Your various keyboard shortcuts

I went pretty Spartan with mine.

```
# Set a Ctrl-b shortcut for reloading your tmux config
bind r source-file ~/.tmux.conf



# Rename your terminals
set -g set-titles on
set -g set-titles-string '#(whoami)::#h::#(curl ipecho.net/plain;echo)'



# Status bar customization
set -g status-utf8 on
set -g status-bg black
set -g status-fg white
set -g status-interval 5
set -g status-left-length 90
set -g status-right-length 60
set -g status-justify left
set -g status-right '#[fg=Cyan]#S #[fg=white]%a %d %b %R'
```

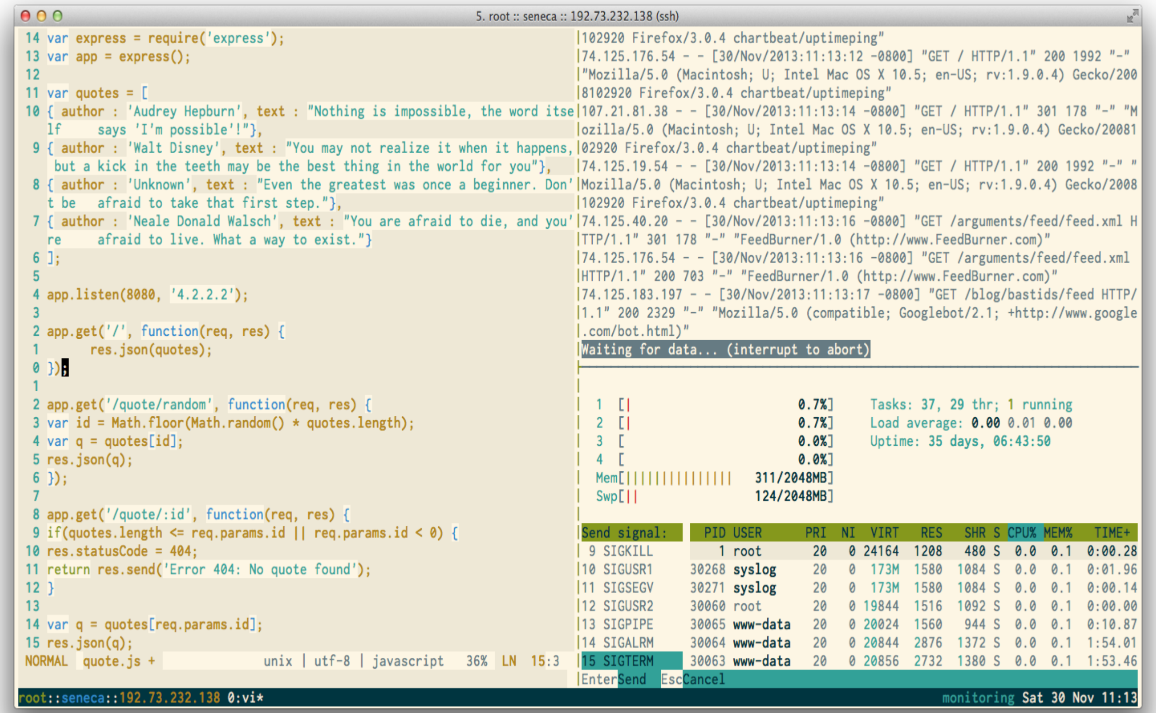
One thing worth noting here is that I use ipecho.net to get my current WAN IP4 WAN address instead of icanhazip as most other tutorials have. It’s just faster and less prone to error, from my experience.

[My current, updated configuration can be found here if you’re interested.]

ADVANCED

That covers how I usually use `tmux`, but I do often make use of some of the more powerful features.

WINDOWS AND PANES



```
5. root :: seneca :: 192.73.232.138 (ssh)
```

```
var express = require('express');
var app = express();

var quotes = [
{ author : 'Audrey Hepburn', text : "Nothing is impossible, the word itse
lf     says 'I'm possible'!"},
{ author : 'Walt Disney', text : "You may not realize it when it happens,
 but a kick in the teeth may be the best thing in the world for you"},
{ author : 'Unknown', text : "Even the greatest was once a beginner. Don'
t be   afraid to take that first step."},
{ author : 'Neale Donald Walsch', text : "You are afraid to die, and you'
re     afraid to live. What a way to exist."}
];

app.listen(8080, '4.2.2.2');

app.get('/', function(req, res) {
      res.json(quotes);
});

app.get('/quote/random', function(req, res) {
var id = Math.floor(Math.random() * quotes.length);
var q = quotes[id];
res.json(q);
});

app.get('/quote/:id', function(req, res) {
if(quotes.length <= req.params.id || req.params.id < 0) {
res.statusCode = 404;
return res.send('Error 404: No quote found');
}

var q = quotes[req.params.id];
res.json(q);
NORMAL  quote.js +              unix | utf-8 | javascript   36%  LN  15:3
```

```
102920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.176.54 - - [30/Nov/2013:11:13:12 -0800] "GET / HTTP/1.1" 200 1992 "-"
"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/200
8102920 Firefox/3.0.4 chartbeat/uptimeping"
107.21.81.38 - - [30/Nov/2013:11:13:14 -0800] "GET / HTTP/1.1" 301 178 "-" "M
ozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/20081
02920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.19.54 - - [30/Nov/2013:11:13:14 -0800] "GET / HTTP/1.1" 200 1992 "-" "
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.0.4) Gecko/2008
102920 Firefox/3.0.4 chartbeat/uptimeping"
74.125.40.20 - - [30/Nov/2013:11:13:16 -0800] "GET /arguments/feed/feed.xml H
TTP/1.1" 301 178 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
74.125.176.54 - - [30/Nov/2013:11:13:16 -0800] "GET /arguments/feed/feed.xml
HTTP/1.1" 200 703 "-" "FeedBurner/1.0 (http://www.FeedBurner.com)"
74.125.183.197 - - [30/Nov/2013:11:13:17 -0800] "GET /blog/bastids/feed HTTP/
1.1" 200 2329 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google
.com/bot.html)"
Waiting for data... (interrupt to abort)
```

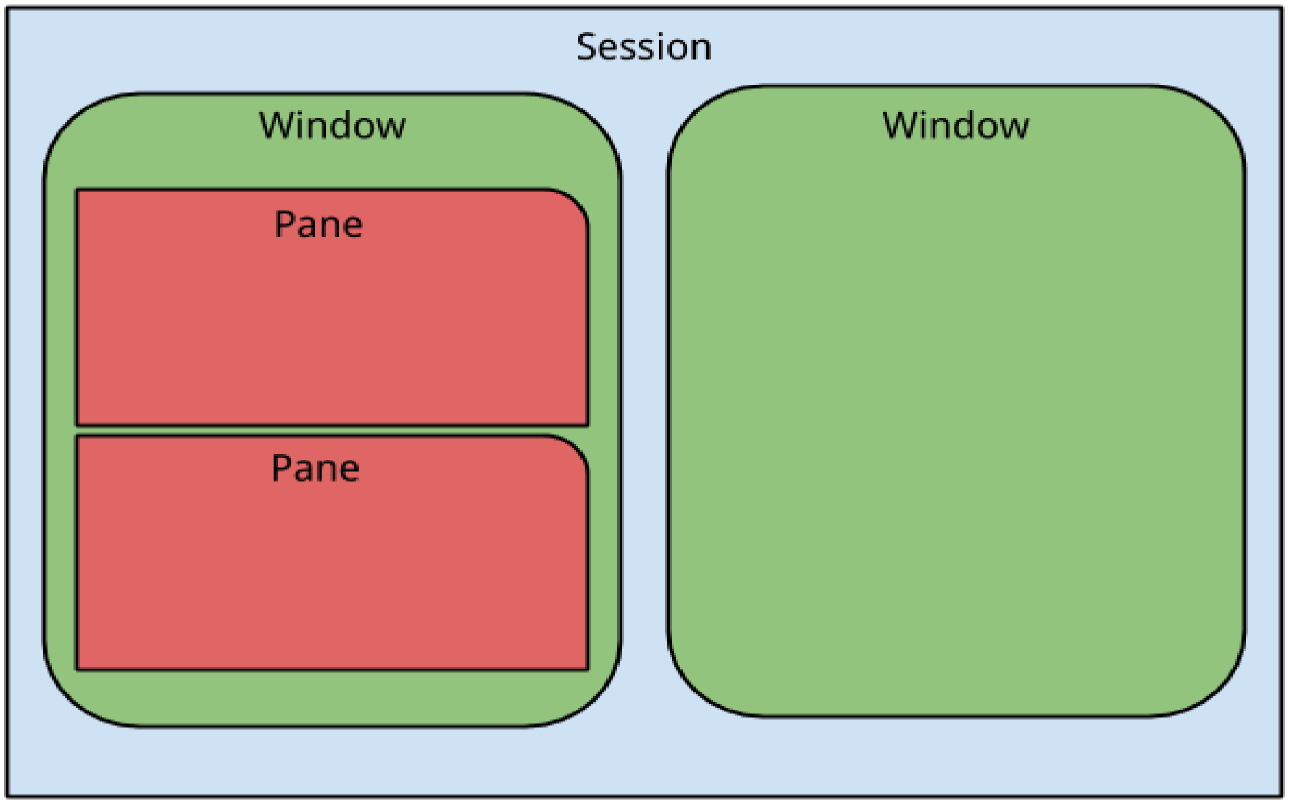
```
  1  [|                         0.7%]     Tasks: 37, 29 thr; 1 running
  2  [|                         0.7%]     Load average: 0.00 0.01 0.00
  3  [                          0.0%]     Uptime: 35 days, 06:43:50
  4  [                          0.0%]
  Mem[||||||||||||||      311/2048MB]
  Swp[||                  124/2048MB]

Send signal:     PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+
 9 SIGKILL         1 root       20   0 24164  1208   480 S  0.0  0.1  0:00.28
10 SIGUSR1     30268 syslog     20   0  173M  1580  1084 S  0.0  0.1  0:01.96
11 SIGSEGV     30271 syslog     20   0  173M  1580  1084 S  0.0  0.1  0:00.14
12 SIGUSR2     30060 root       20   0 19844  1516  1092 S  0.0  0.1  0:00.00
13 SIGPIPE     30065 www-data   20   0 20024  1560   944 S  0.0  0.1  0:10.87
14 SIGALRM     30064 www-data   20   0 20844  2876  1372 S  0.0  0.1  1:54.01
15 SIGTERM     30063 www-data   20   0 20856  2732  1380 S  0.0  0.1  1:53.46
EnterSend  EscCancel
```

```
root::seneca::192.73.232.138 0:vi*                         monitoring Sat 30 Nov 11:13
```

One of these features is the ability to break your session into more discreet components, called windows and panes. These are good for organizing multiple varied activities in a logical way.

Let’s look at how they relate to each other.

Nesting



`tmux` sessions have windows, and windows have panes. Below you can see how how I conceptualize them—although if anyone has a more authoritative or useful hierarchy I'll happily embrace it.

- **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.

It's also possible to create panes within a session without first creating a separate window. I do this sometimes. Hopefully it isn't as horrible as it sounds right after reading about nesting. As I said in the beginning, I incline towards simplicity with my use of `tmux`.

Navigating with panes

There's a default way to navigate between panes, but I don't know what it is. I'm a vim guy, so I navigate within my panes using the *h*, *j*, *k*, and *l* keys like so:

```
# Remap window navigation to vim
unbind-key j
bind-key j select-pane -D
unbind-key k
bind-key k select-pane -U
unbind-key h
bind-key h select-pane -L
unbind-key l
bind-key l select-pane -R
```

RECOMMENDATIONS

A few thoughts that may help you in your `tmux` travels:

1. 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.
2. 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.

3. Start with a basic config and get used to it before you get silly. I've seen multiple people spend hours configuring vim or tmux only to confuse themselves and abandon the project altogether. Start simple.

SHORTCUT REFERENCE

Now a `Ctrl-b` options reference:

Basics

? get help

Session management

s list sessions

$ rename the current session

d detach from the current session

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

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 *

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

Miscellaneous

t show the time in current pane

I hope this has been helpful.

[If you liked this, check out my other technical primers here.]

[CREATED: March 2013, UPDATED: January 2015]

NOTES

1. The man page.
2. A thousand other great tutorials.

RECOMMENDED

A vim Tutorial and Primer

9 Enhancements to Shell and Vim Productivity

Windows File Sharing: Facing the Mystery

Web Application Security Testing Resources

8 Powerful Features of Safari That Few People Know About

The First 10 Things I Do on a New Mac

NEWSLETTER

Every Sunday I put out a list of the most interesting stories in infosec, technology, and humans.

I curate for 5 hours, you read in 5 minutes. Over 7K subscribers.

email address

SUBSCRIBE