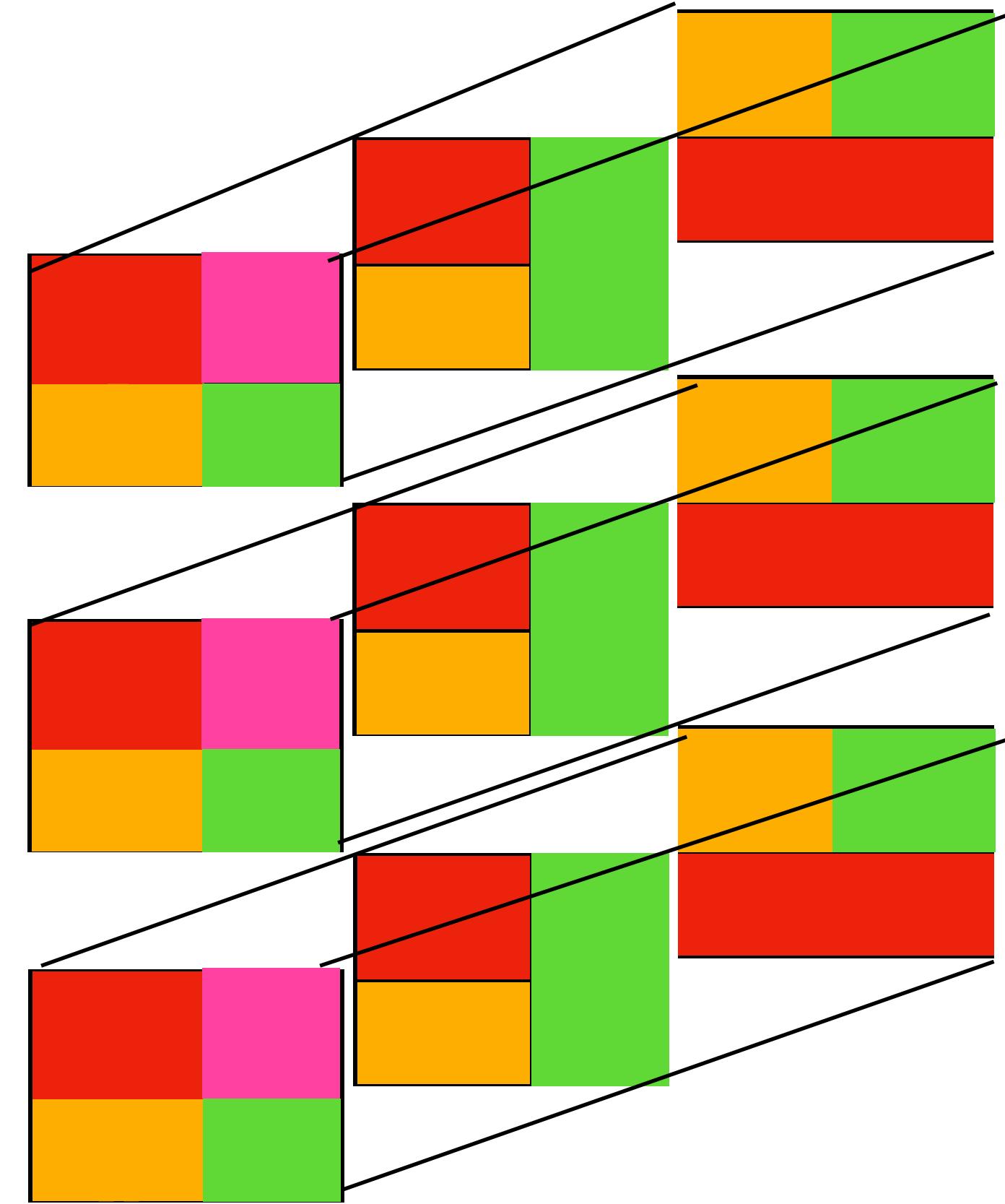


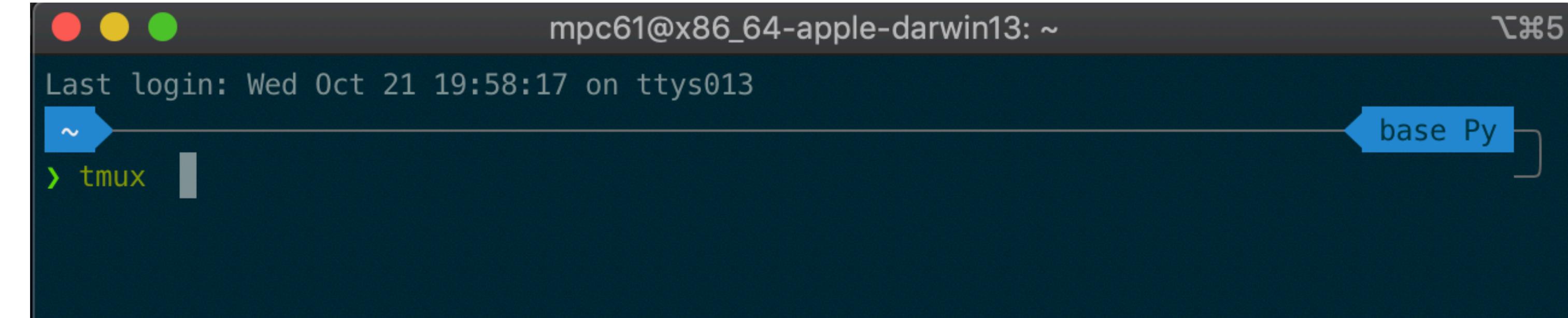
Enter the Multiplex

Not tmux information

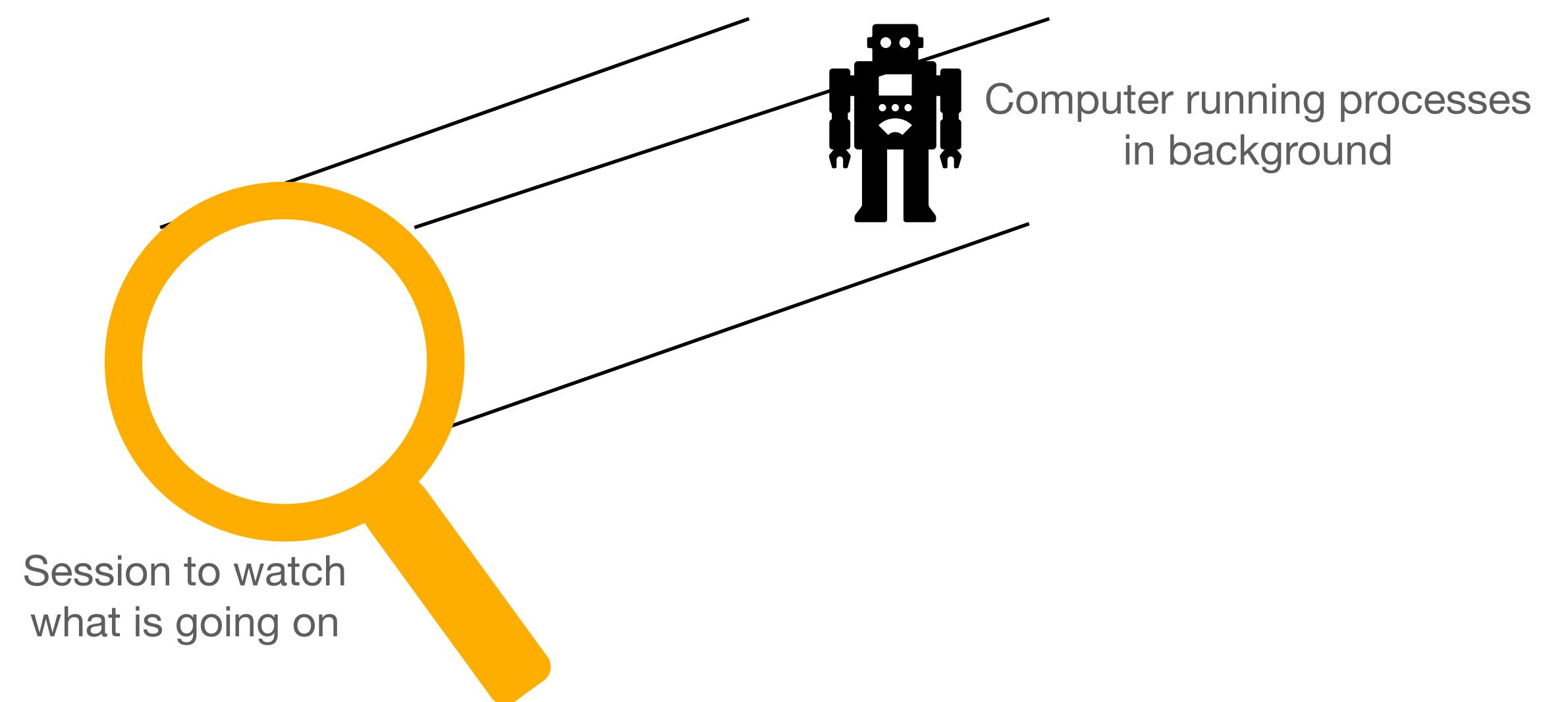
Michael Colaresi



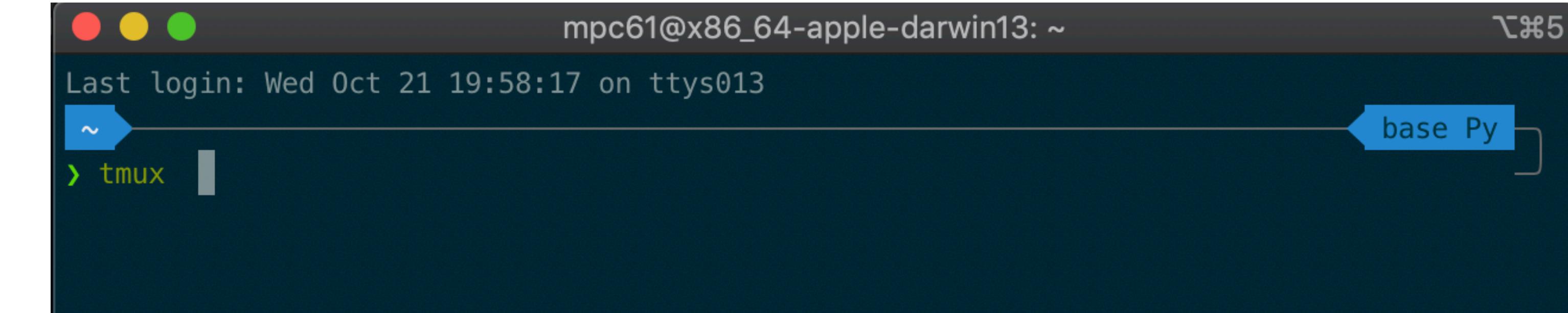
Build tmux-cles



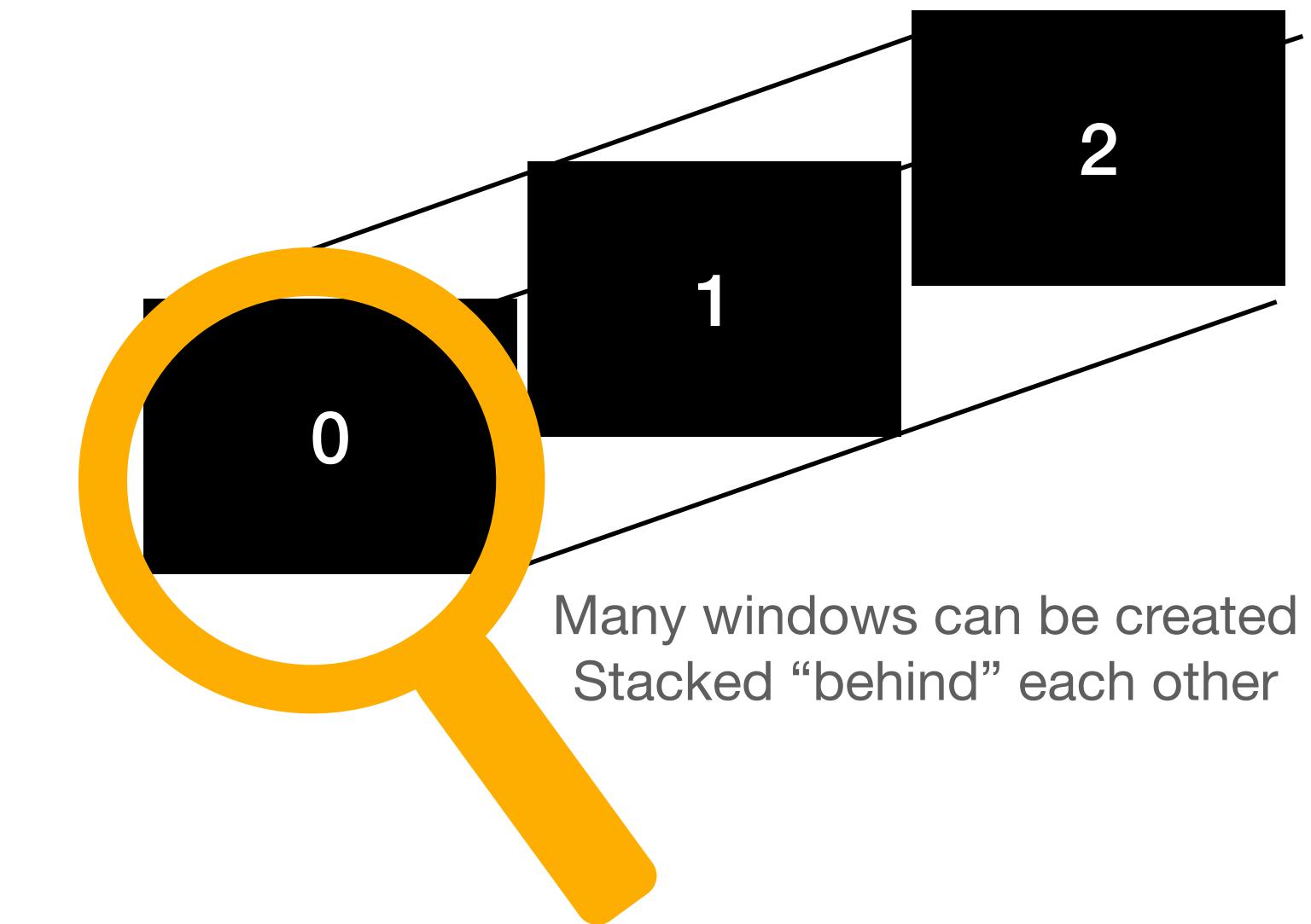
- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions



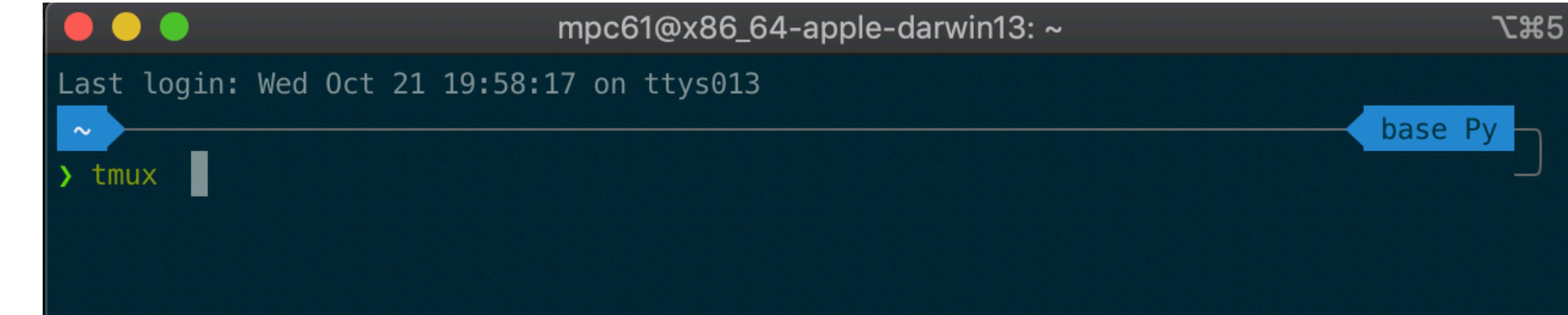
Build tmux-cles



- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions

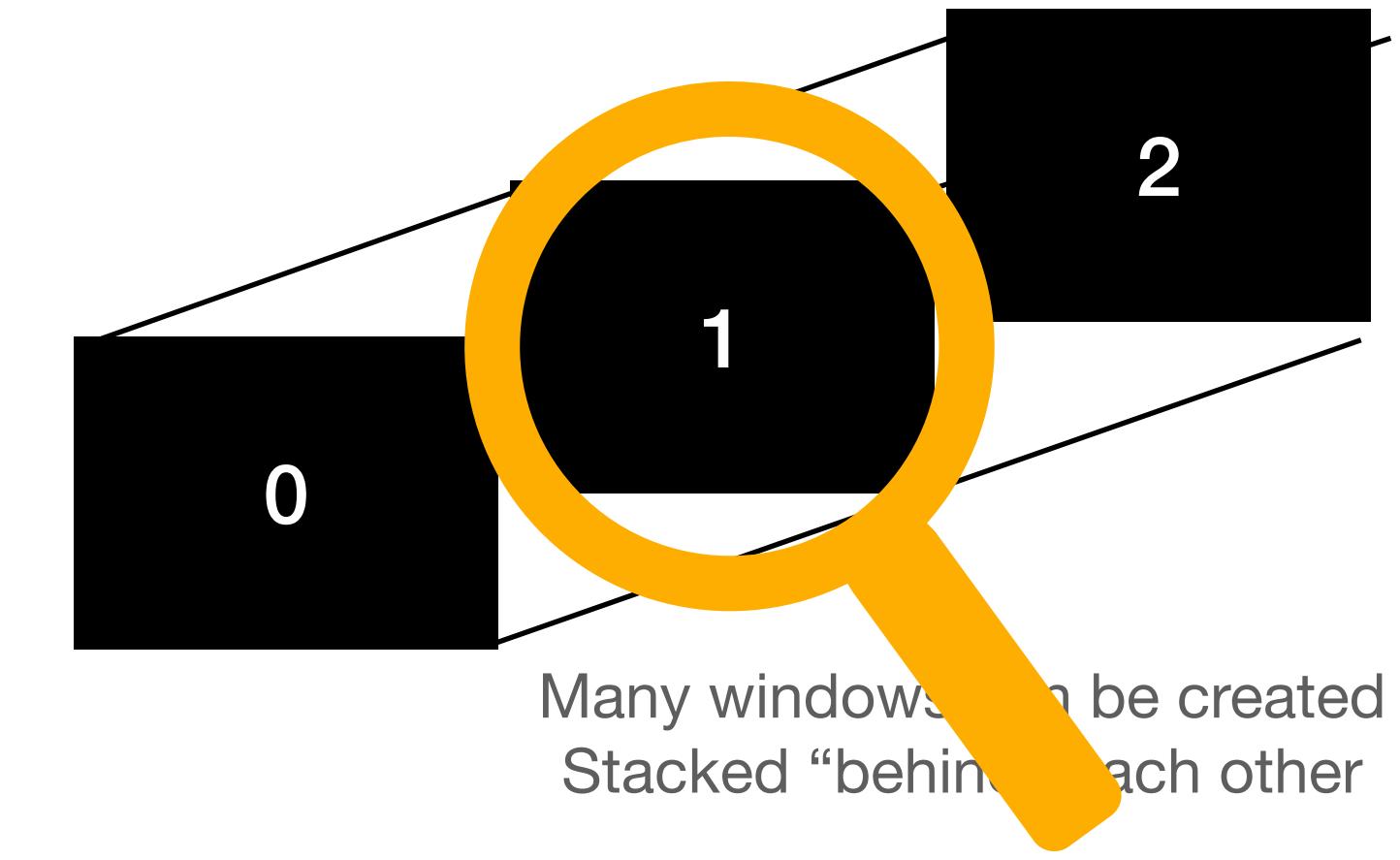


Build tmux-cles

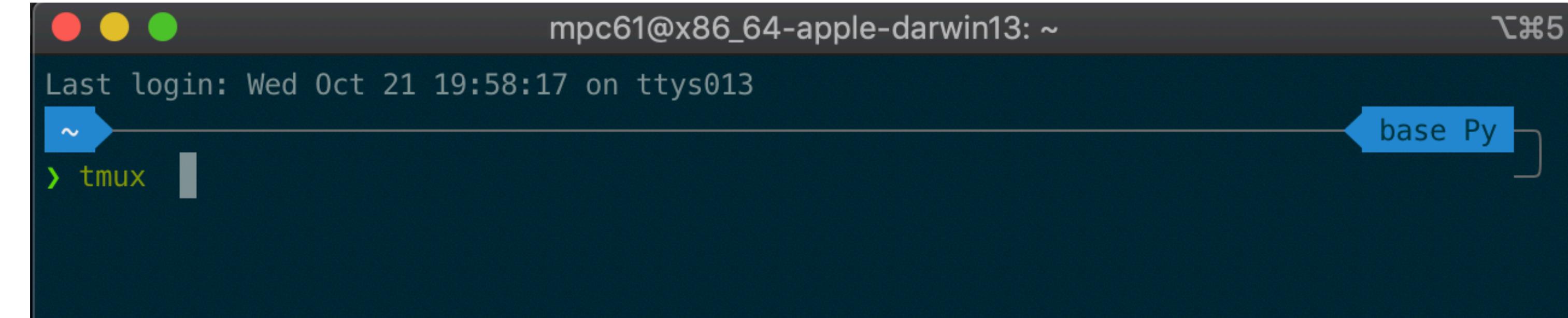


A screenshot of a terminal window titled "base Py". The window shows a tmux session with three panes. The top pane displays the command-line interface with the prompt "mpc61@x86_64-apple-darwin13: ~". The second pane shows the output of "last login" command. The bottom pane shows the "tmux" command being run. The tmux logo is visible in the bottom-left corner of the window.

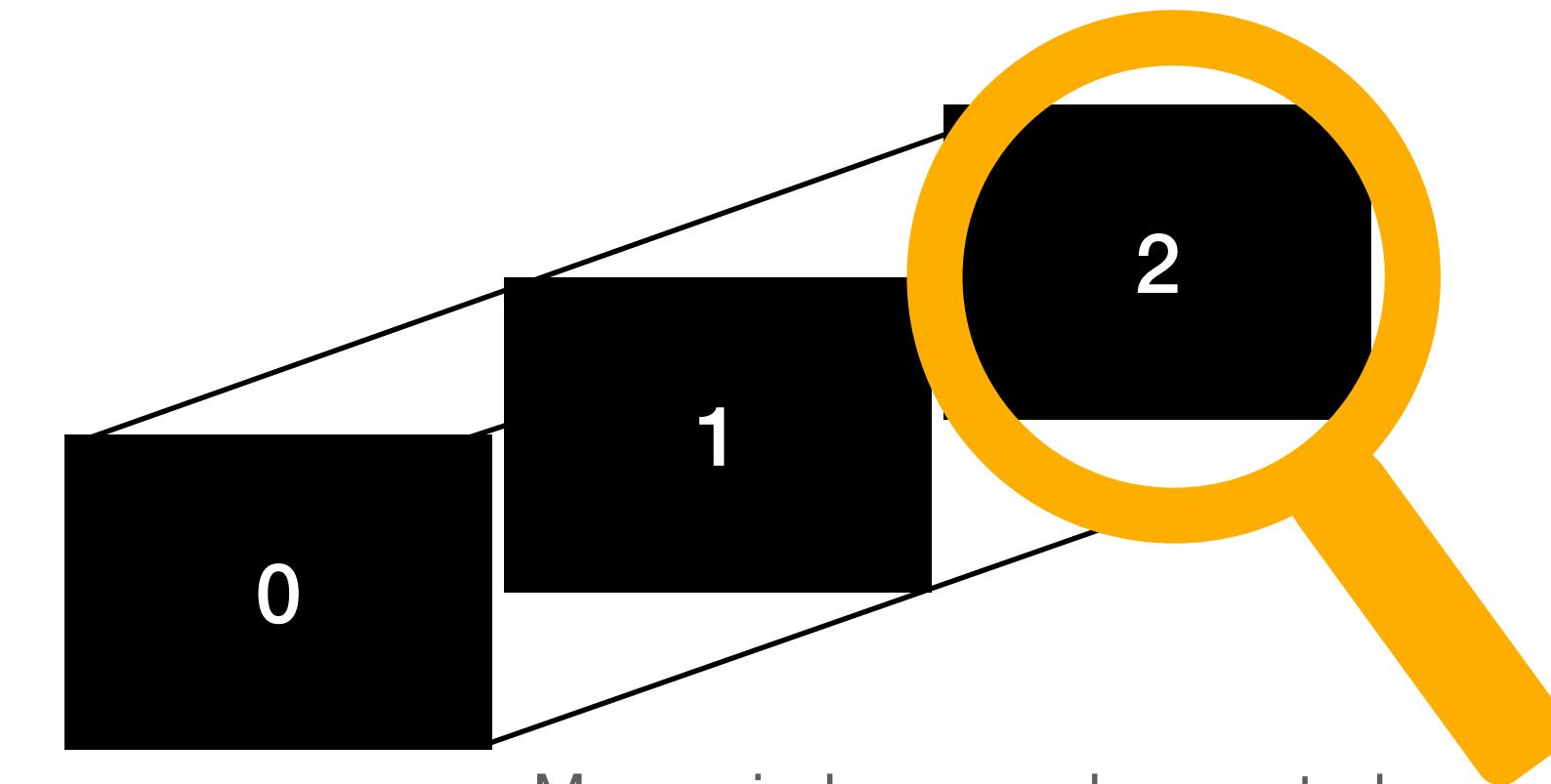
- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions



Build tmux-cles

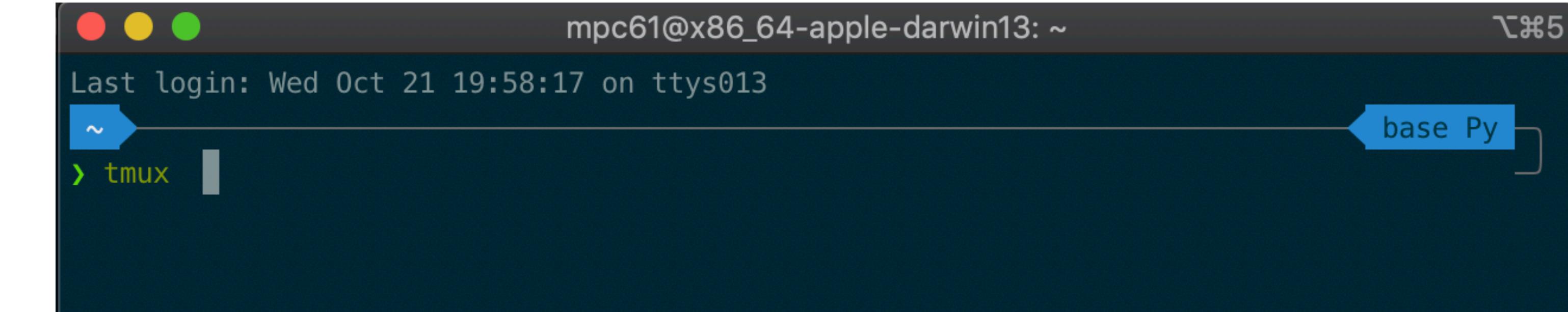


- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions

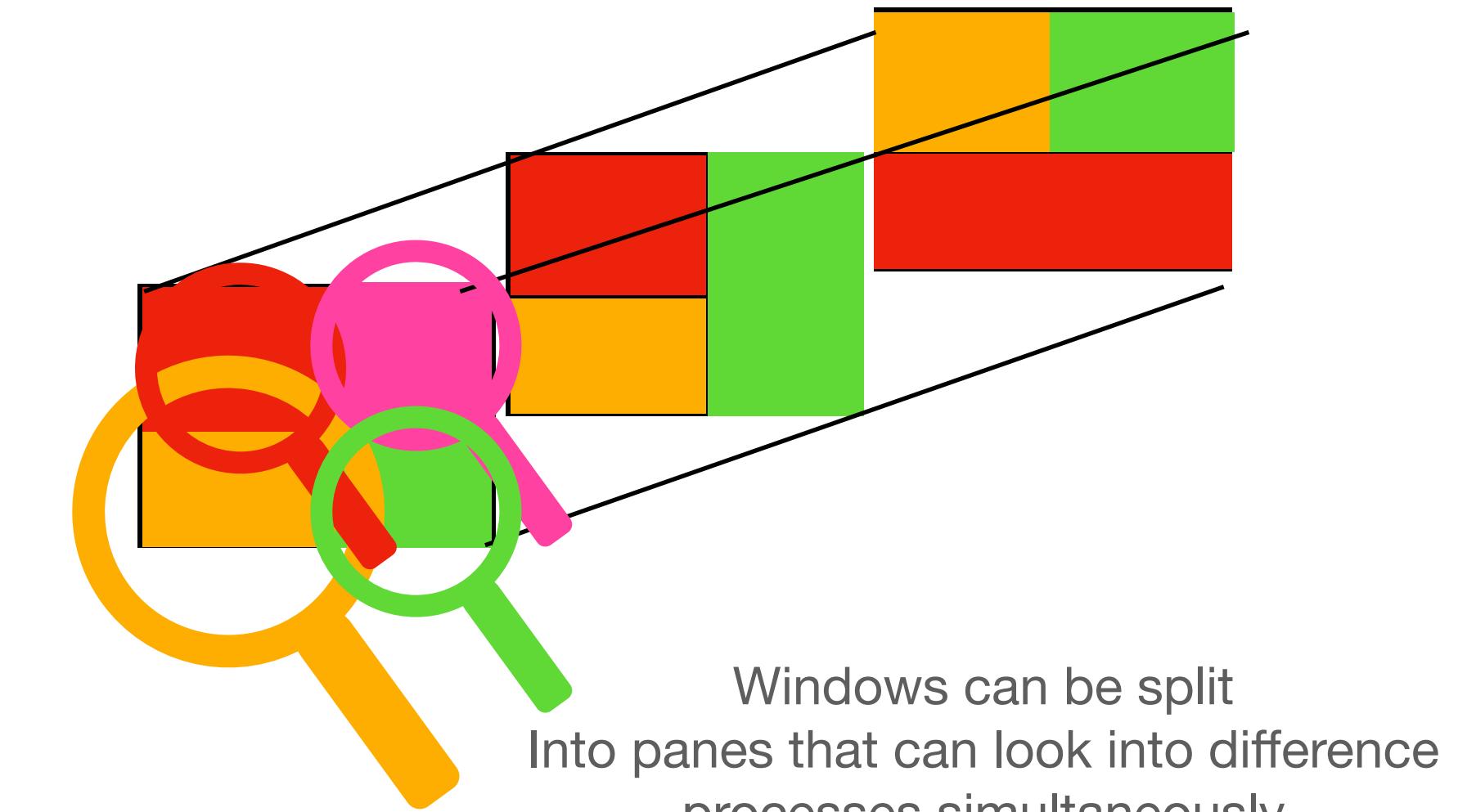


Many windows can be created
Stacked “behind” each other

Build tmux-cles

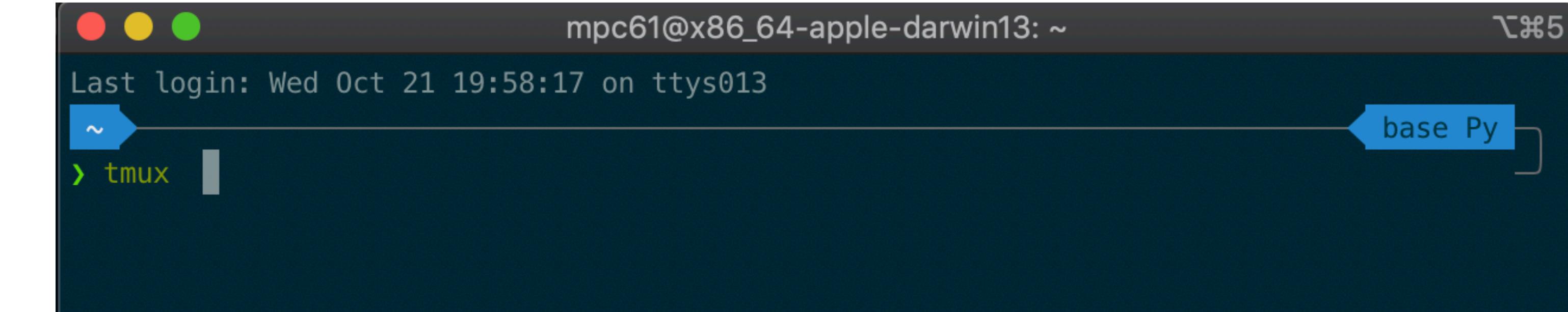


- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions

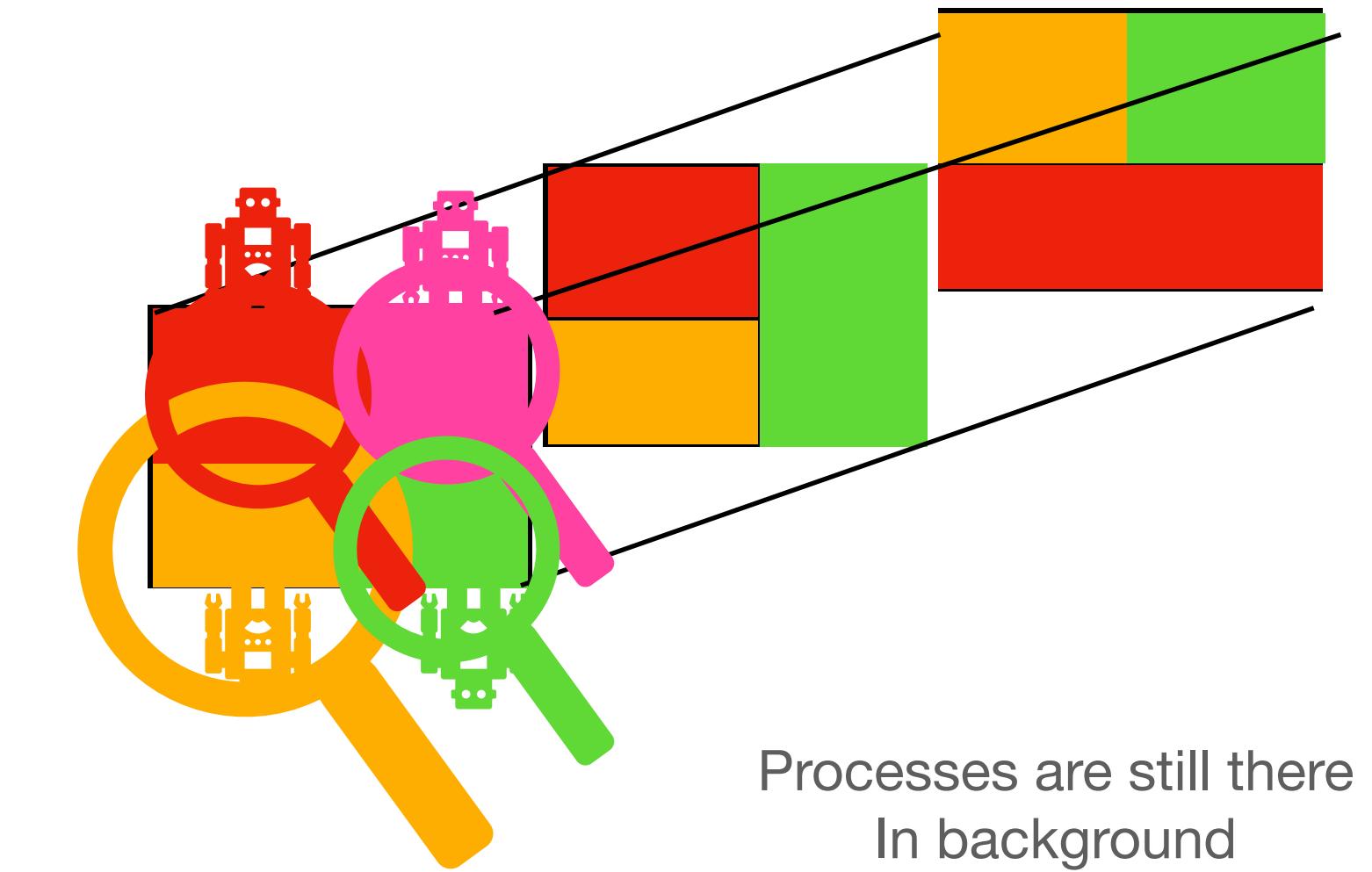


Windows can be split
Into panes that can look into difference
processes simultaneously

Build tmux-cles

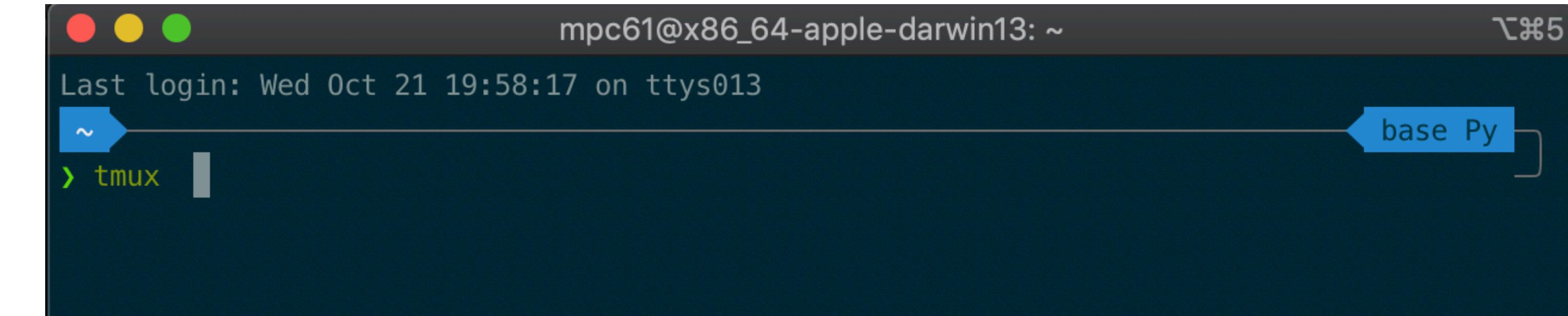


- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions



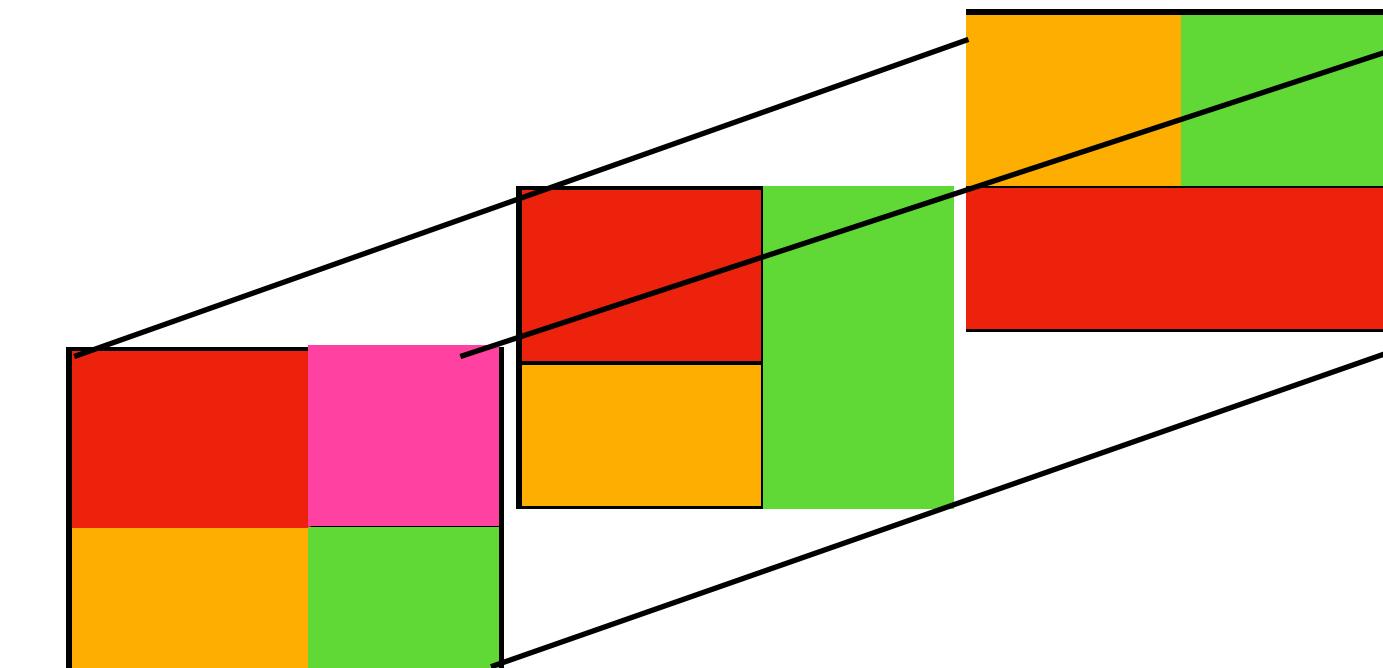
Processes are still there
In background

Build tmux-cles



A screenshot of a terminal window titled "base Py". The window shows a tmux session with a single window containing the command "tmux". The status bar at the bottom indicates "Last login: Wed Oct 21 19:58:17 on ttys013". The window title bar has three colored dots (red, yellow, green).

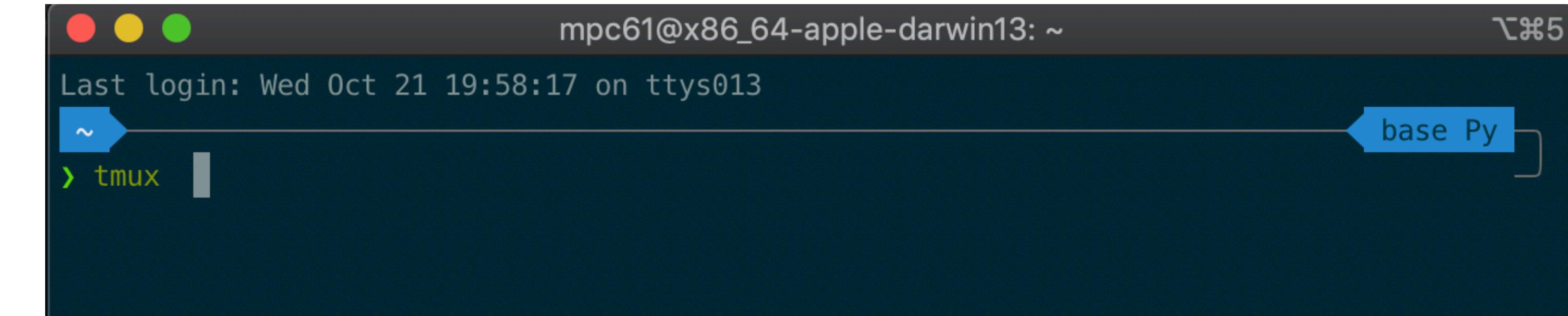
- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow **detaching** and re-attaching sessions



Can “detach” session, but still keep processes running

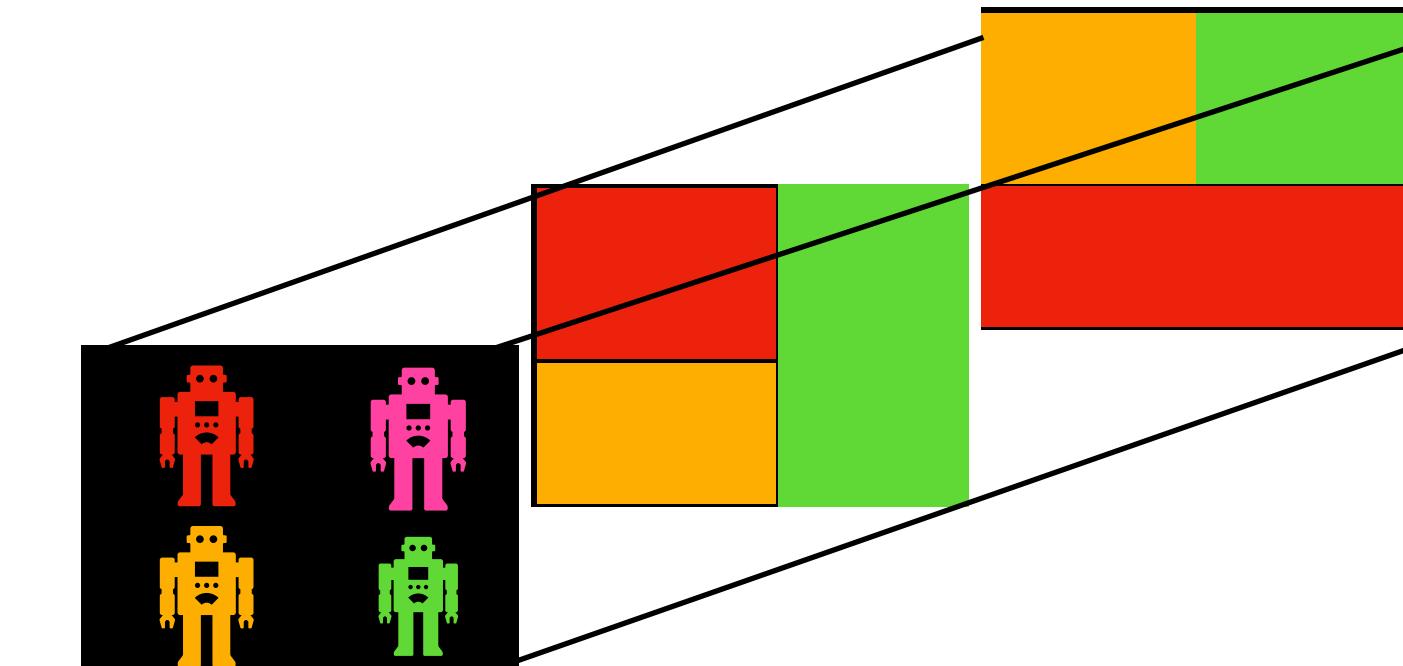
This is unlike your terminal.
when you close item the processes
are killed too

Build tmux-cles



A screenshot of a terminal window titled "base Py". The status bar at the top shows "mpc61@x86_64-apple-darwin13: ~" and "Last login: Wed Oct 21 19:58:17 on ttys013". The main pane displays a tmux session with several colored panes (red, green, yellow) and a command prompt (~). A blue arrow icon is visible in the bottom left corner.

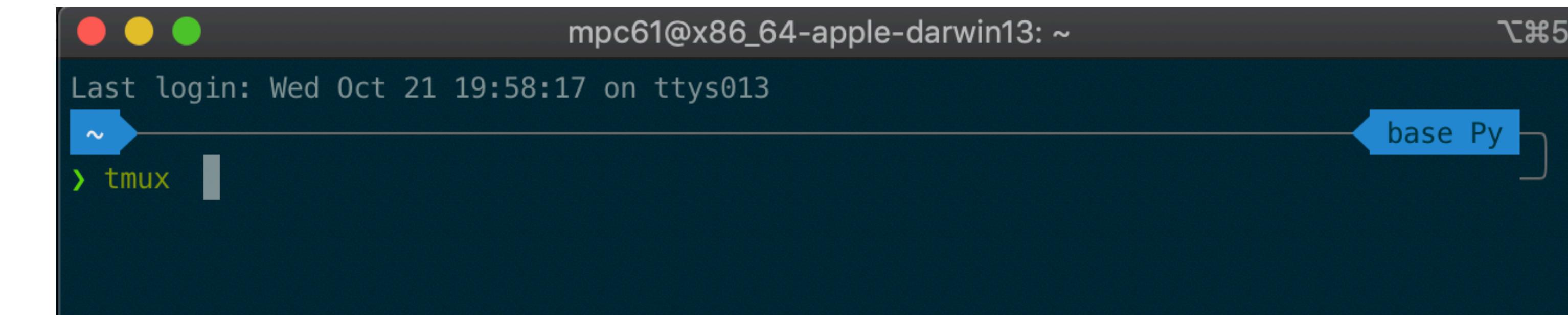
- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow **detaching** and re-attaching sessions



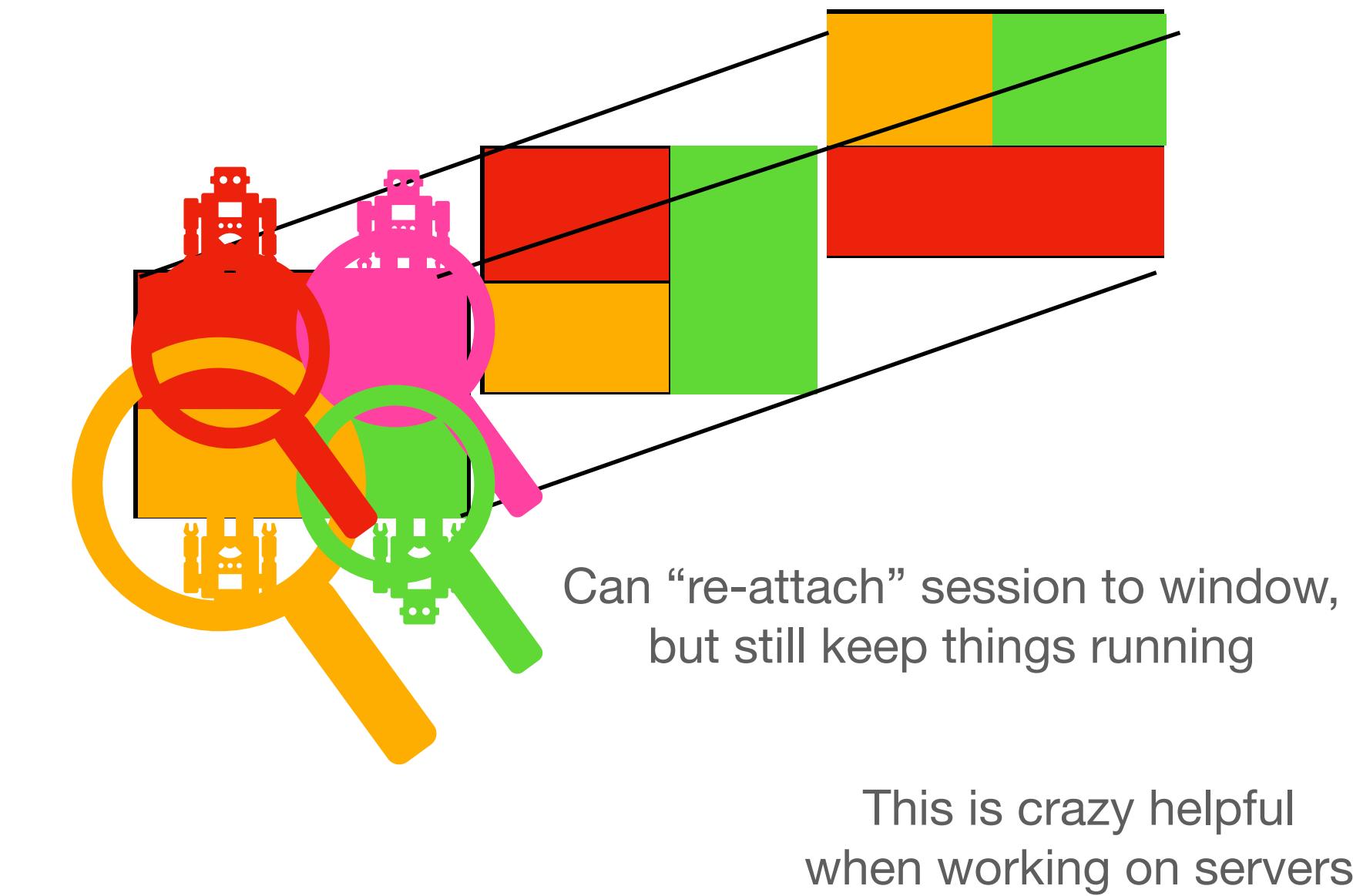
Can “detach” session, but still keep processes running

This is unlike your terminal.
when you close item the processes
are killed too

Build tmux-cles

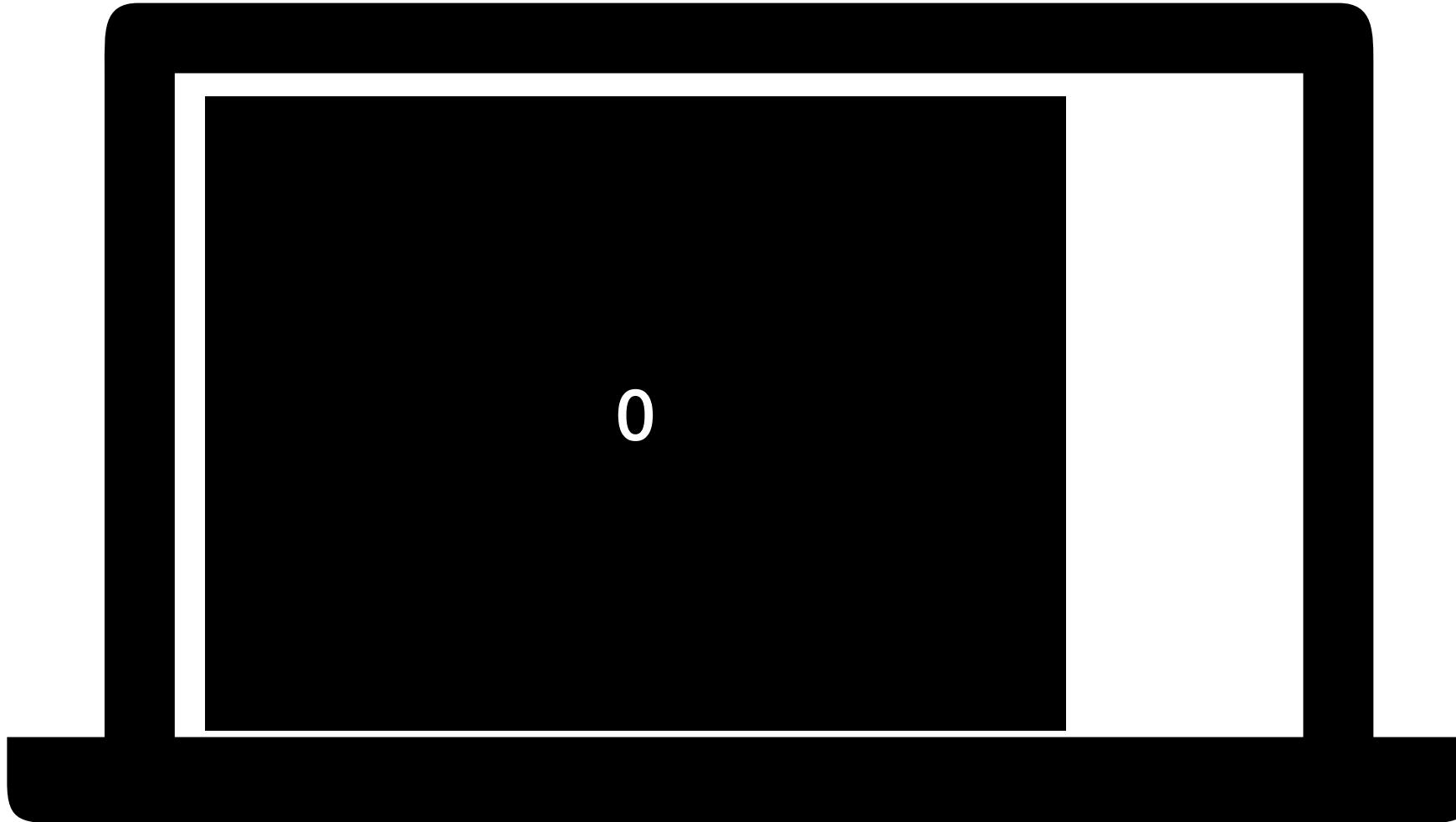


- Terminal Multiplexer (tmux)
- Turn one terminal window into:
 - a Session
 - with Windows
 - each with many panes
 - And allow detaching and re-attaching sessions



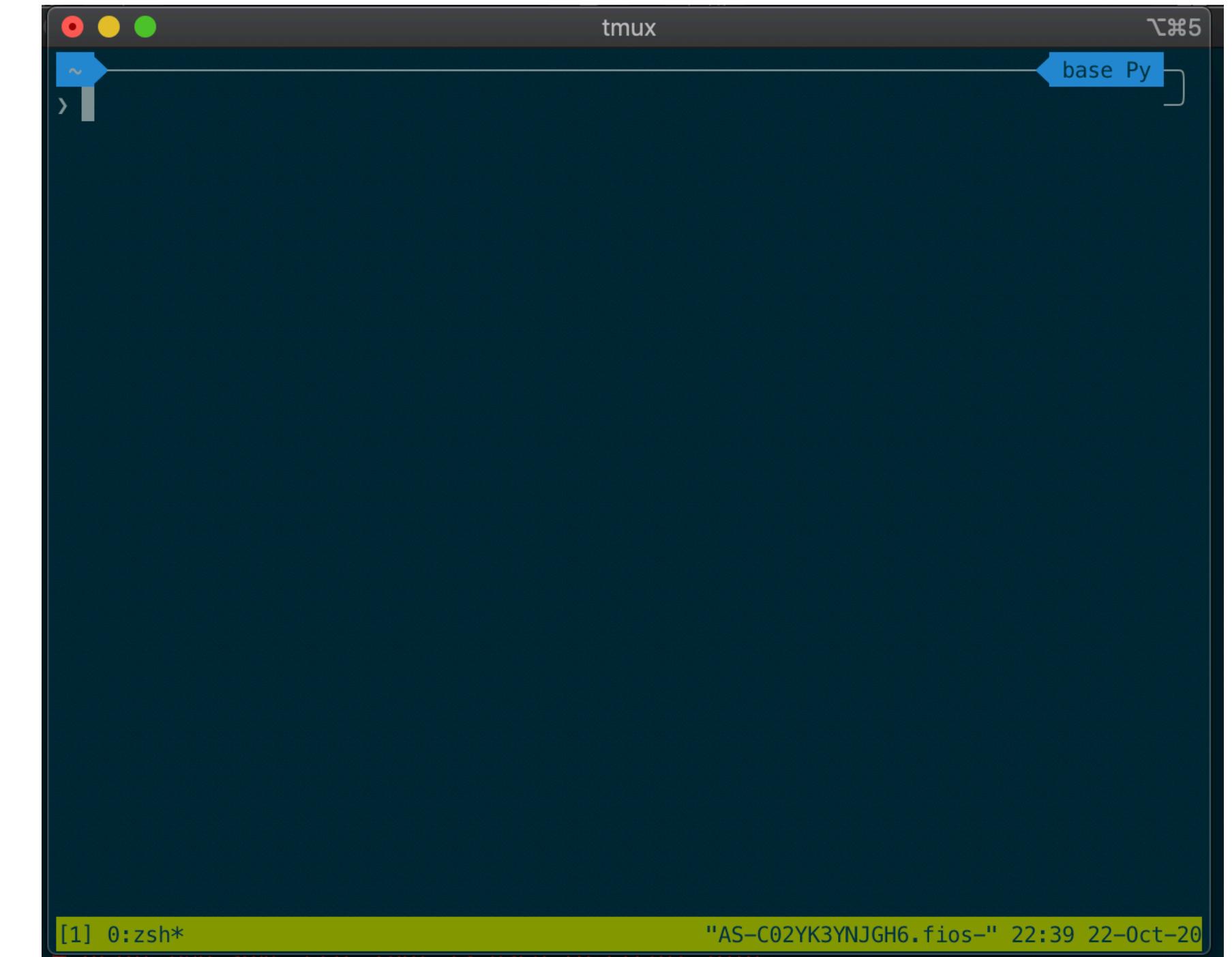
Session

- A session can have multiple windows



Open tmux session

>tmux or >tmux new-session -s NAME



So far, just a terminal window with a prompt

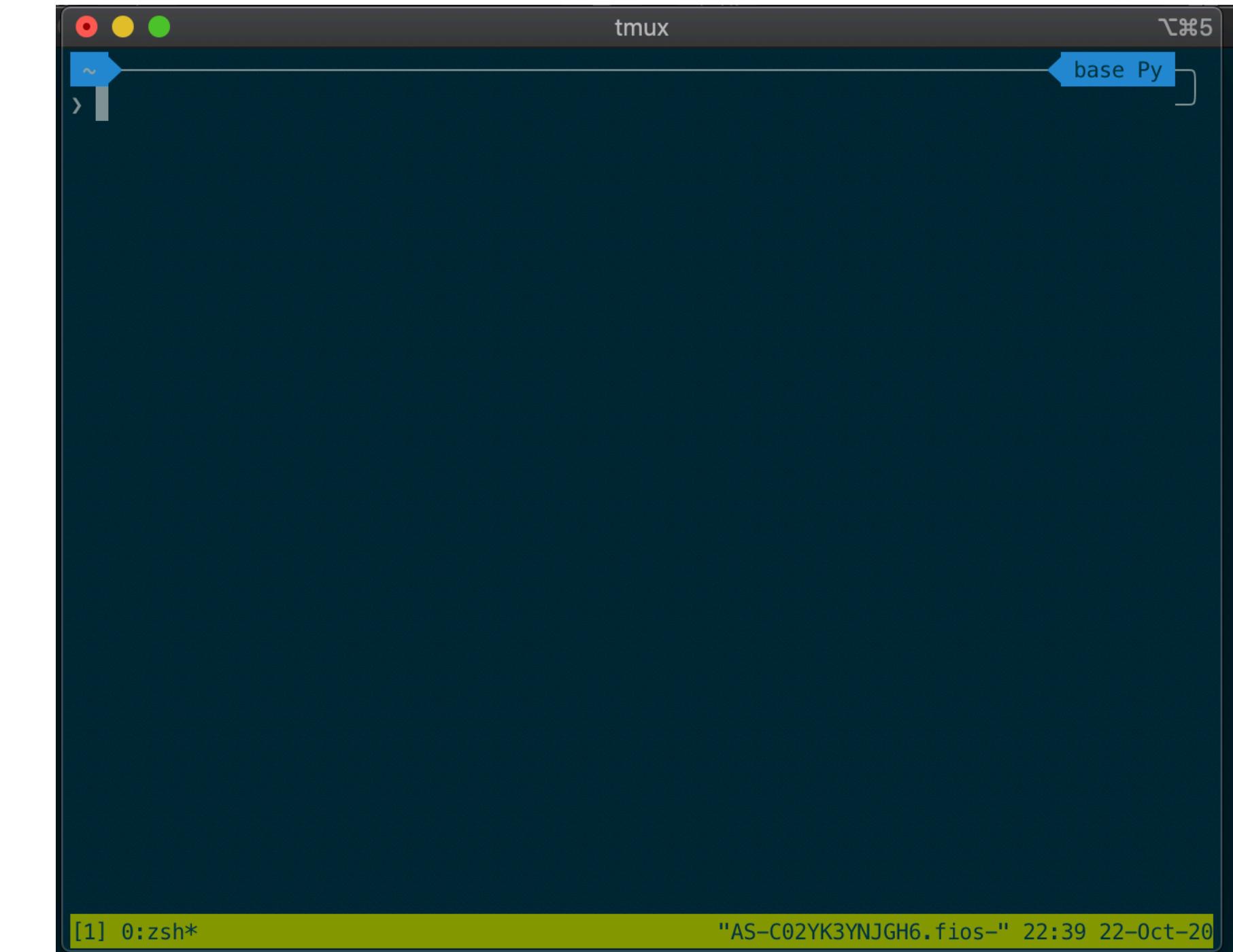
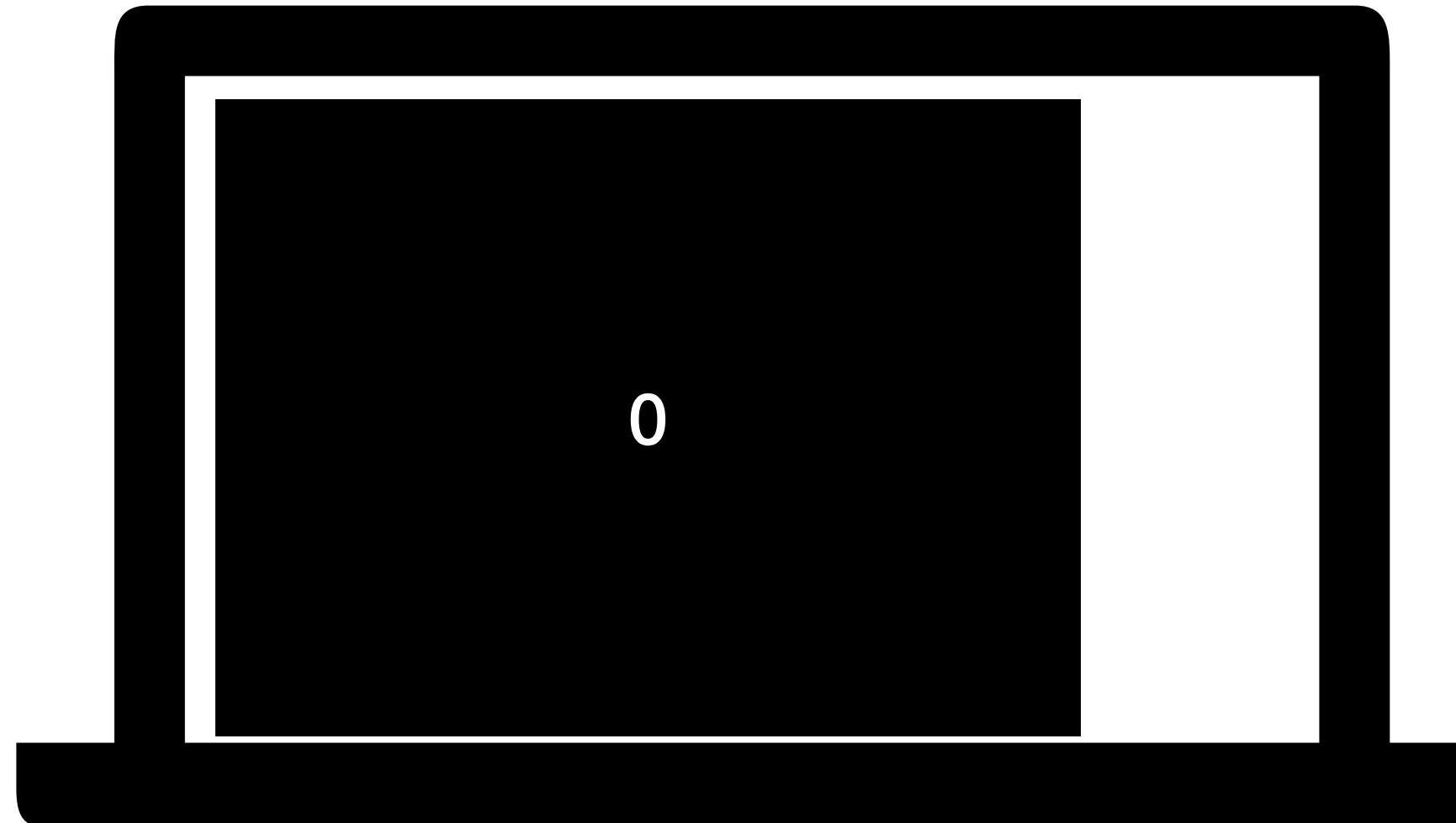
Session



Open tmux session

>tmux or >tmux new-session -s NAME

- A session can have multiple windows



So far, just a terminal window with a prompt

To talk to tmux we must use the key strokes ctrl-b (at the same time)

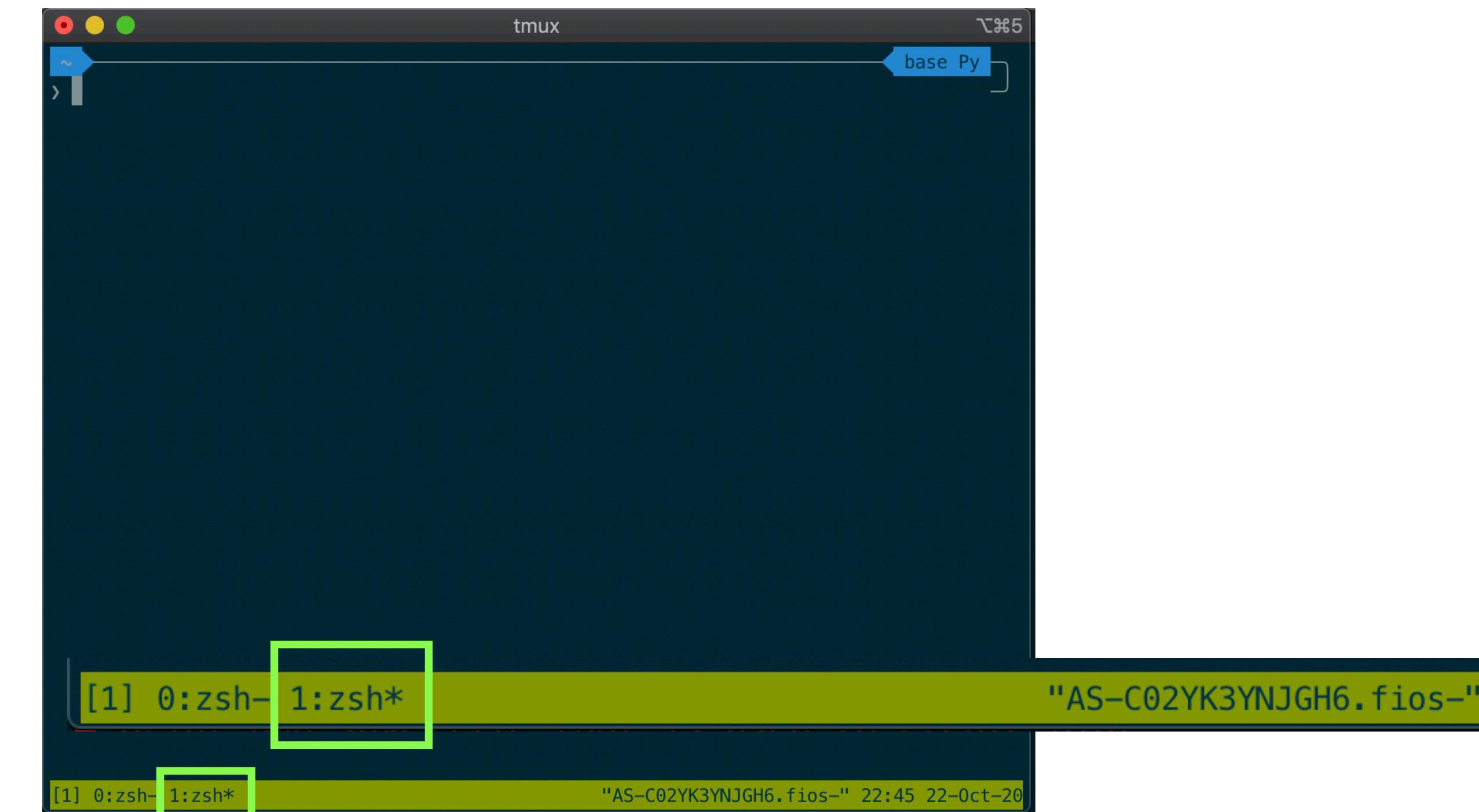
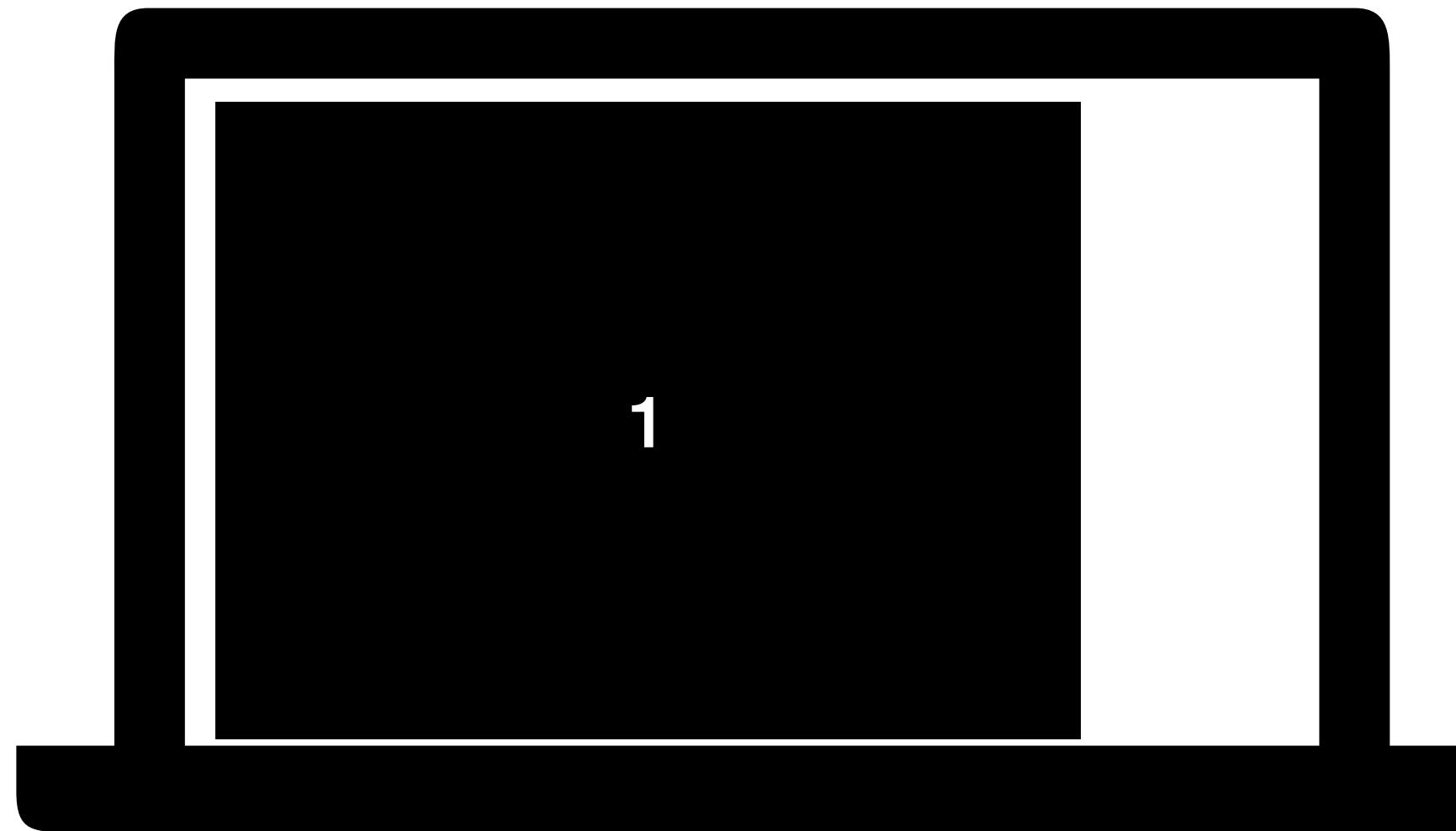
Windows



Open another window
ctrl-b c

c for see a
new window

- A session can have multiple windows



In the same frame

* tells us where we are

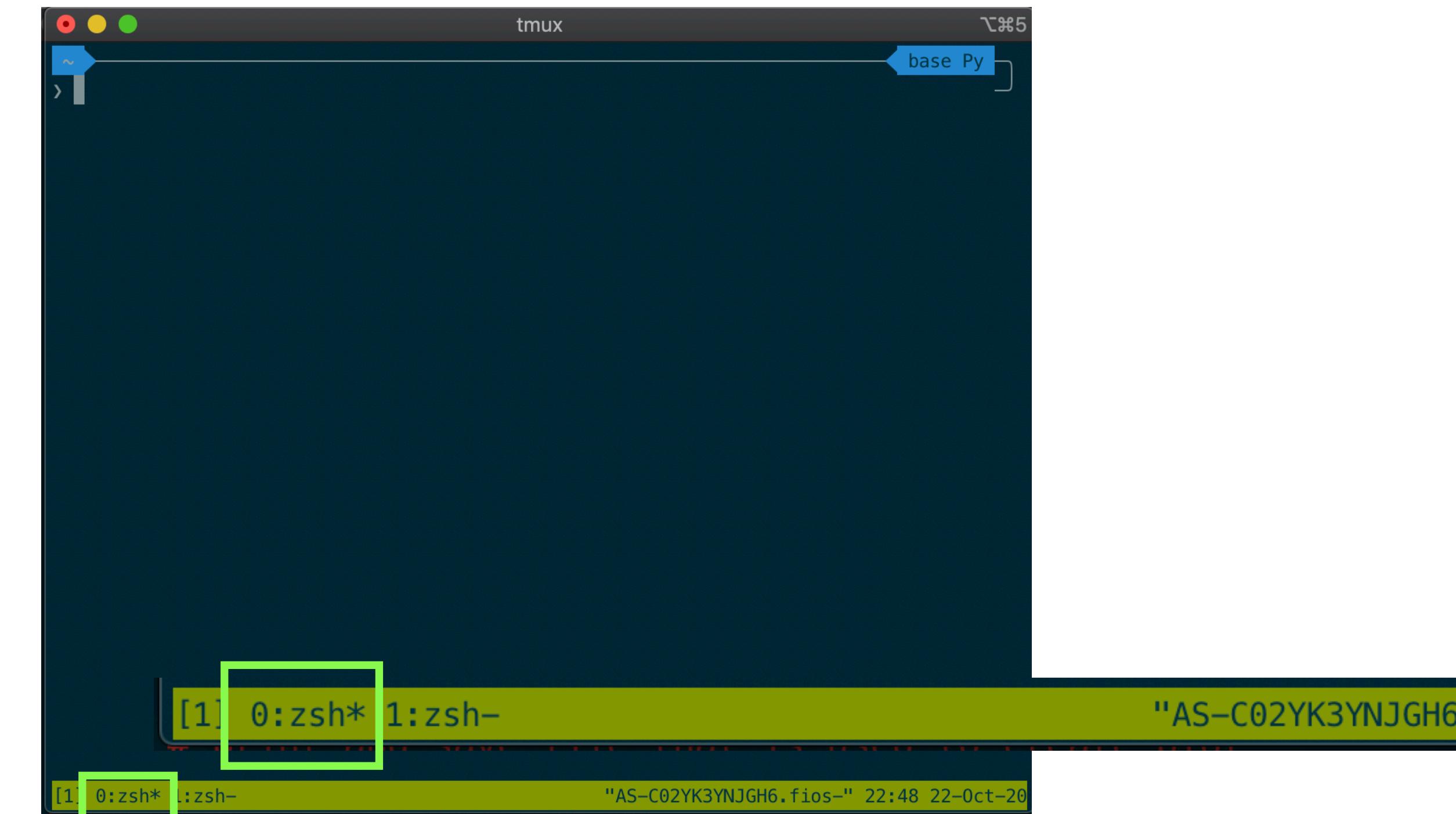
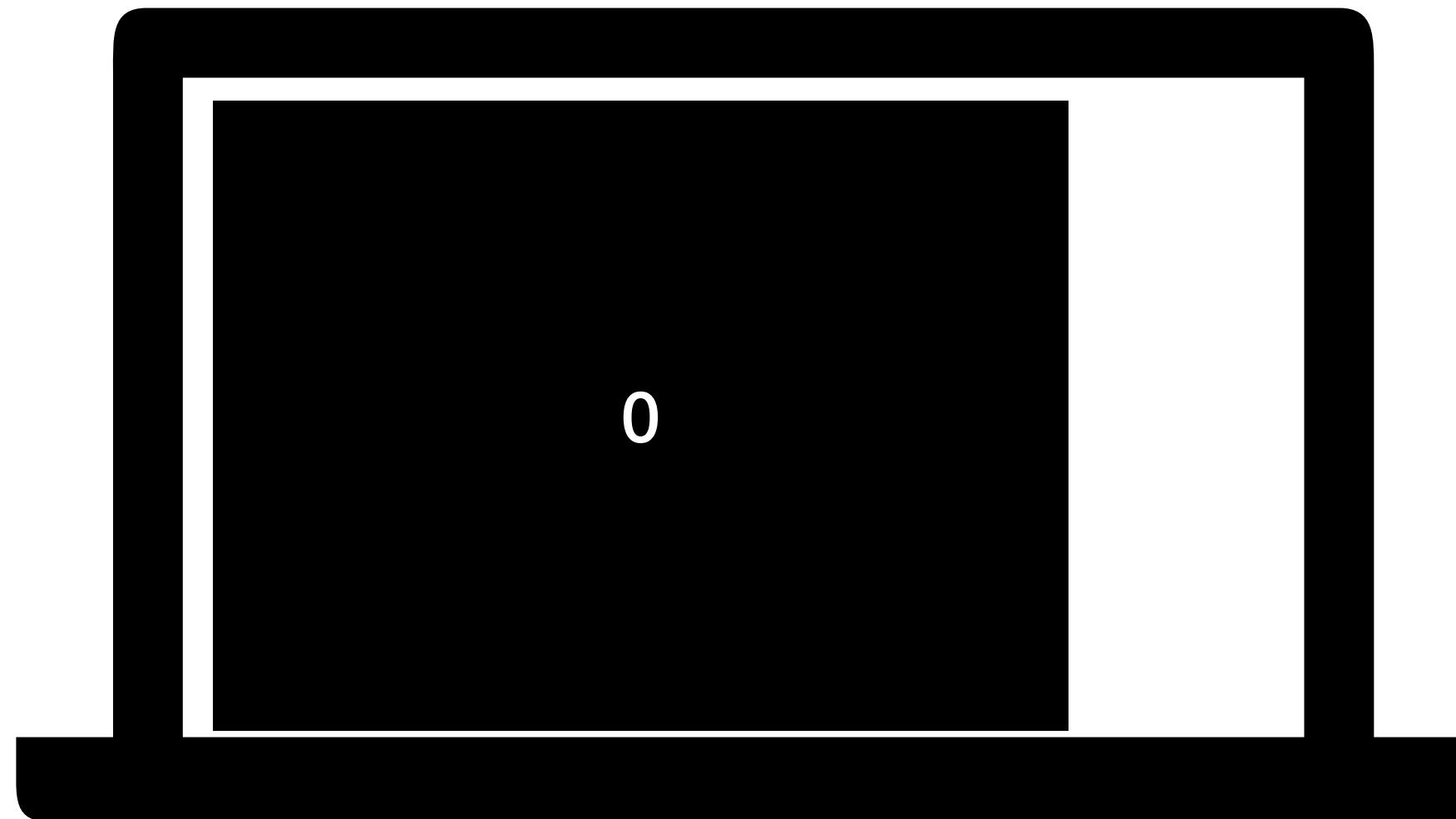
Windows



Back to 0th window
ctrl-b p

Previous
window

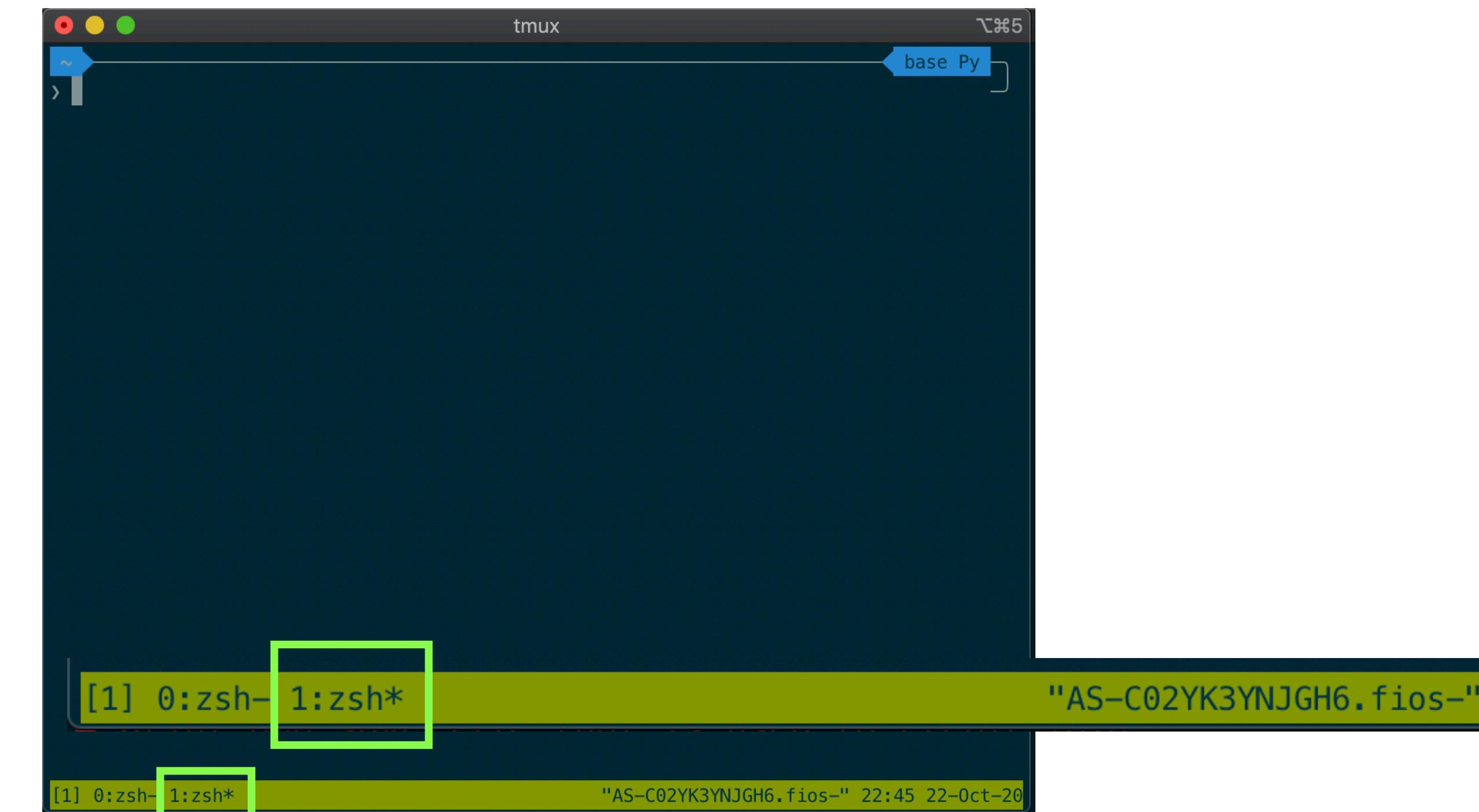
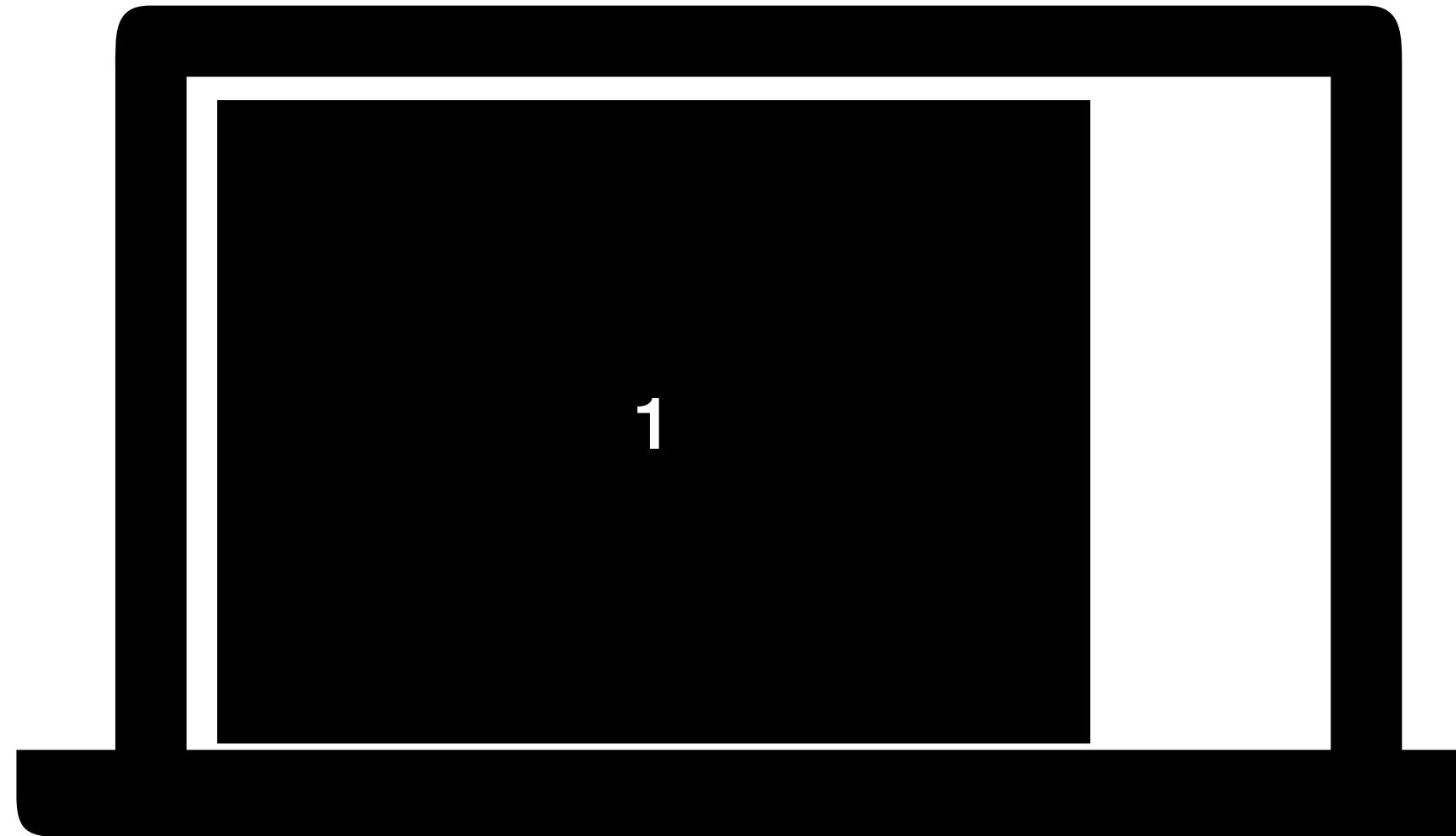
- A session can have multiple windows



Toggle back and forth

Windows

- A session can have multiple windows



Back to 1st window
ctrl-b p

again

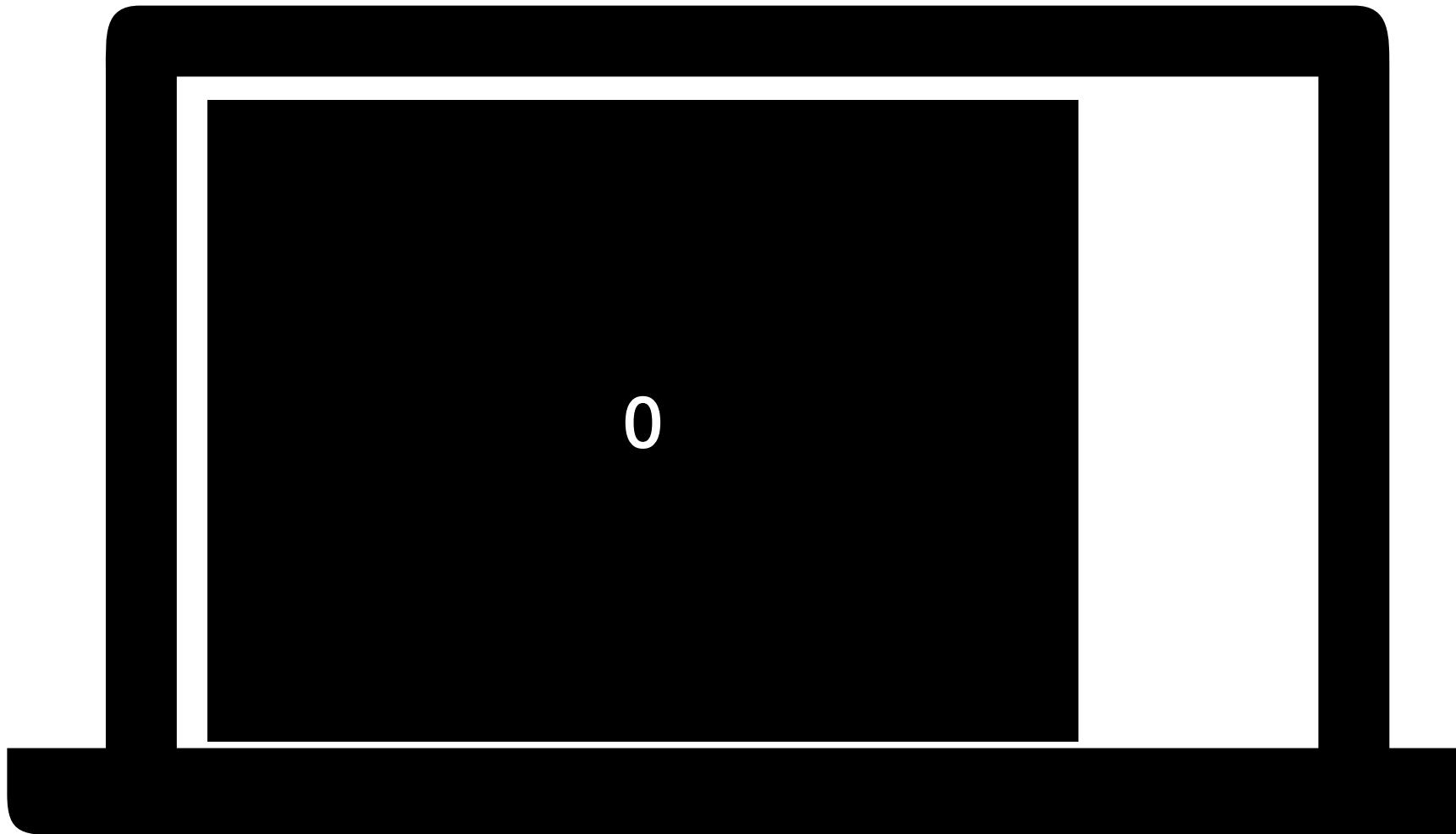
Toggle back and forth

We have two terminal windows, but nothing special yet

Windows

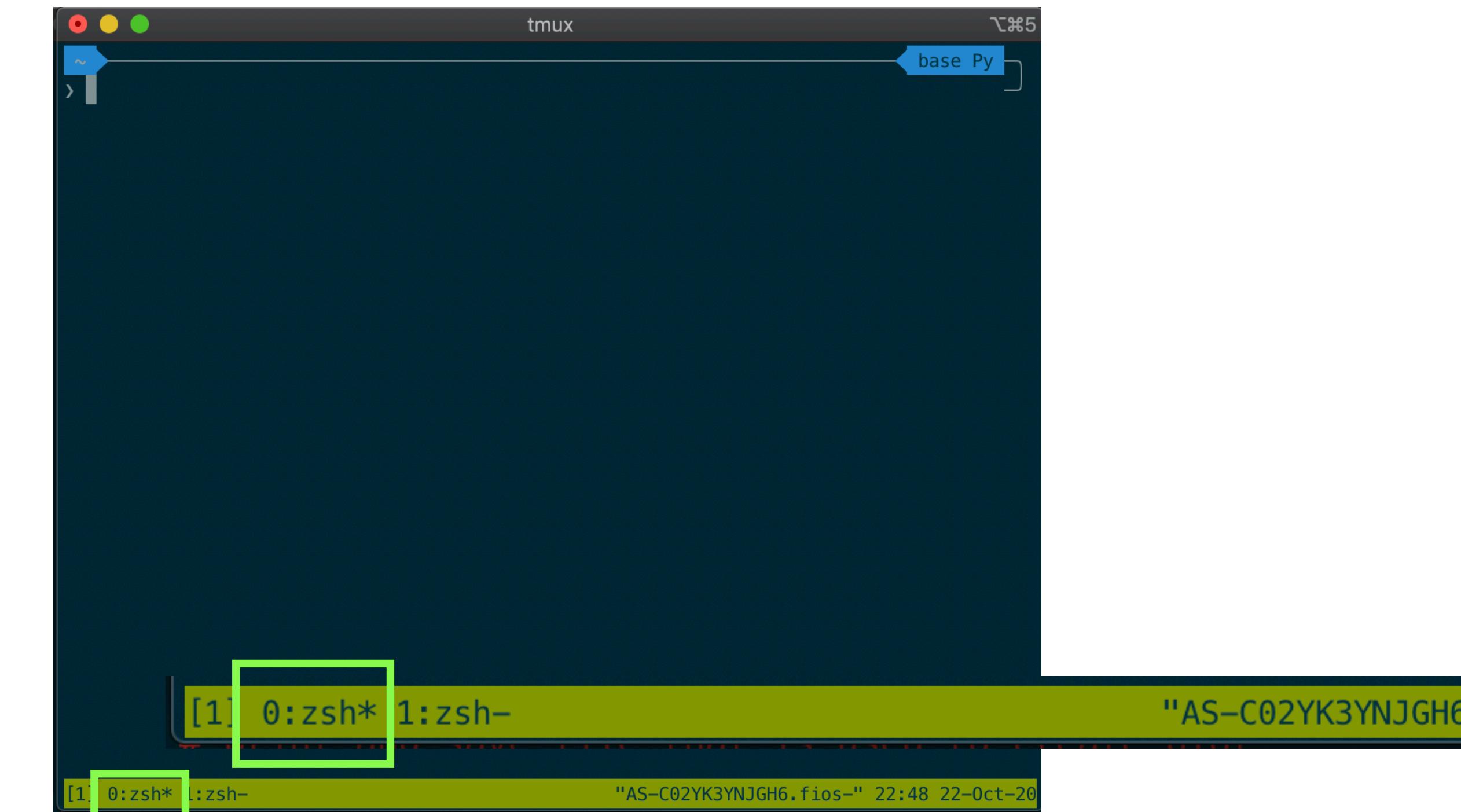


- A session can have multiple windows



Can also navigate with window name
ctrl-b 0

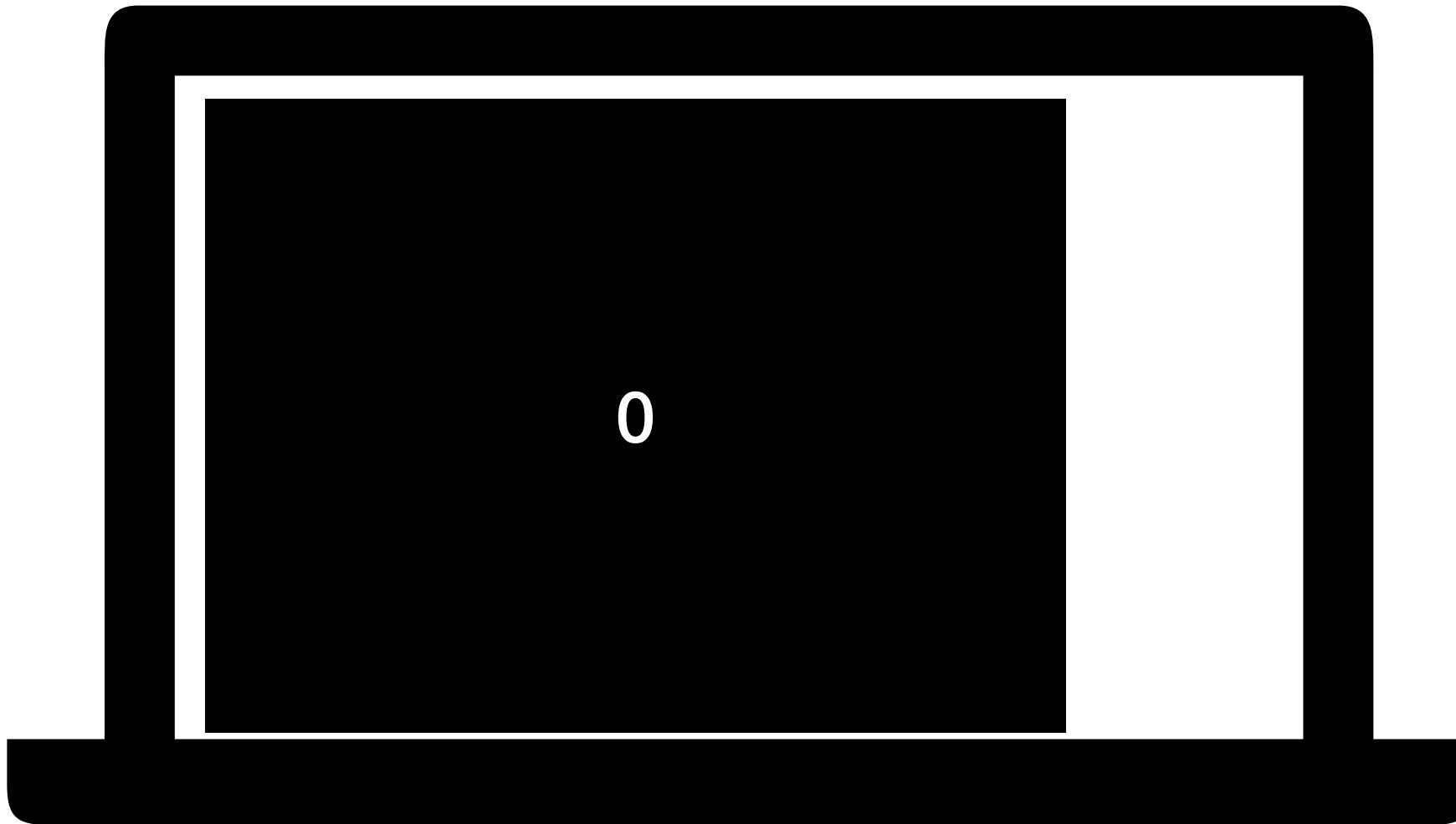
Goes to
window 0



Toggle back and forth

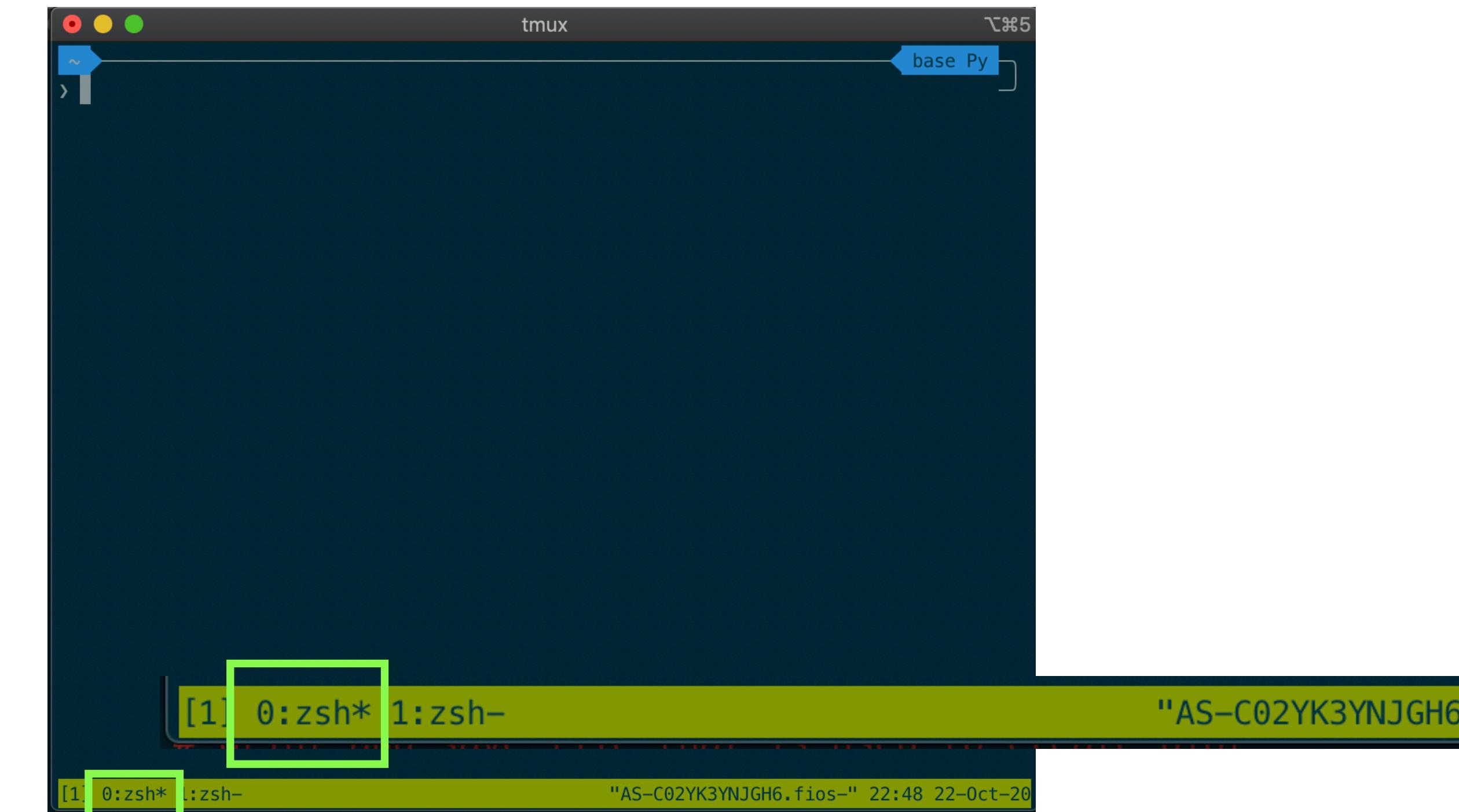
Windows

- A session can have multiple windows

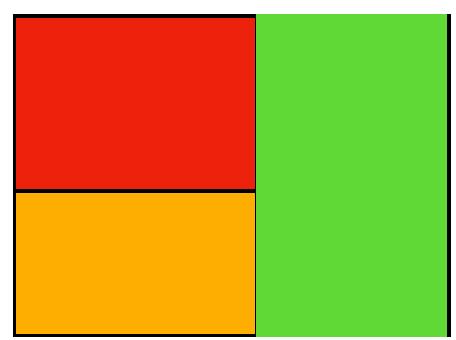


Can also name a window
with ctrl-b ,
then type name

Goes to
window 0



Toggle back and forth

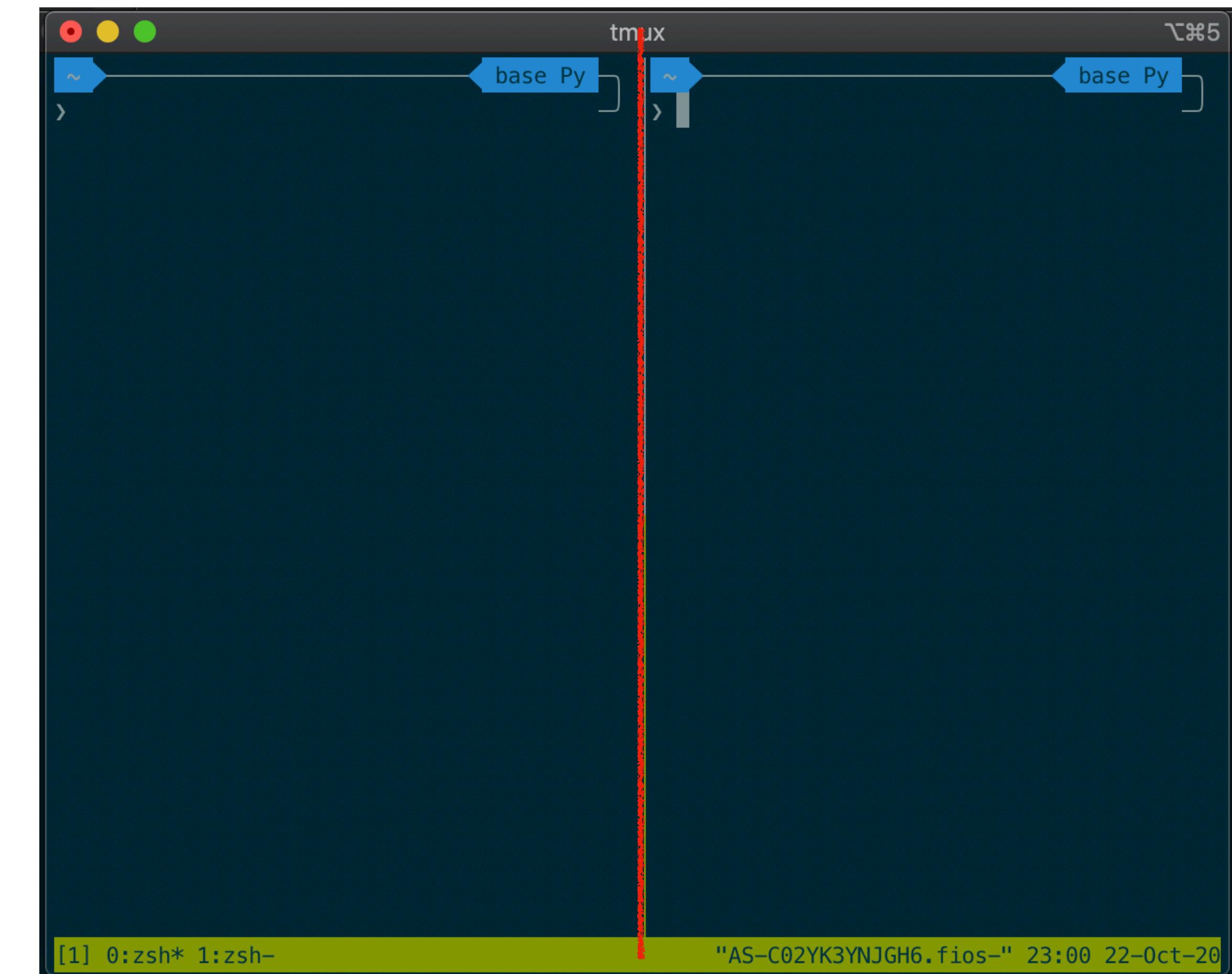
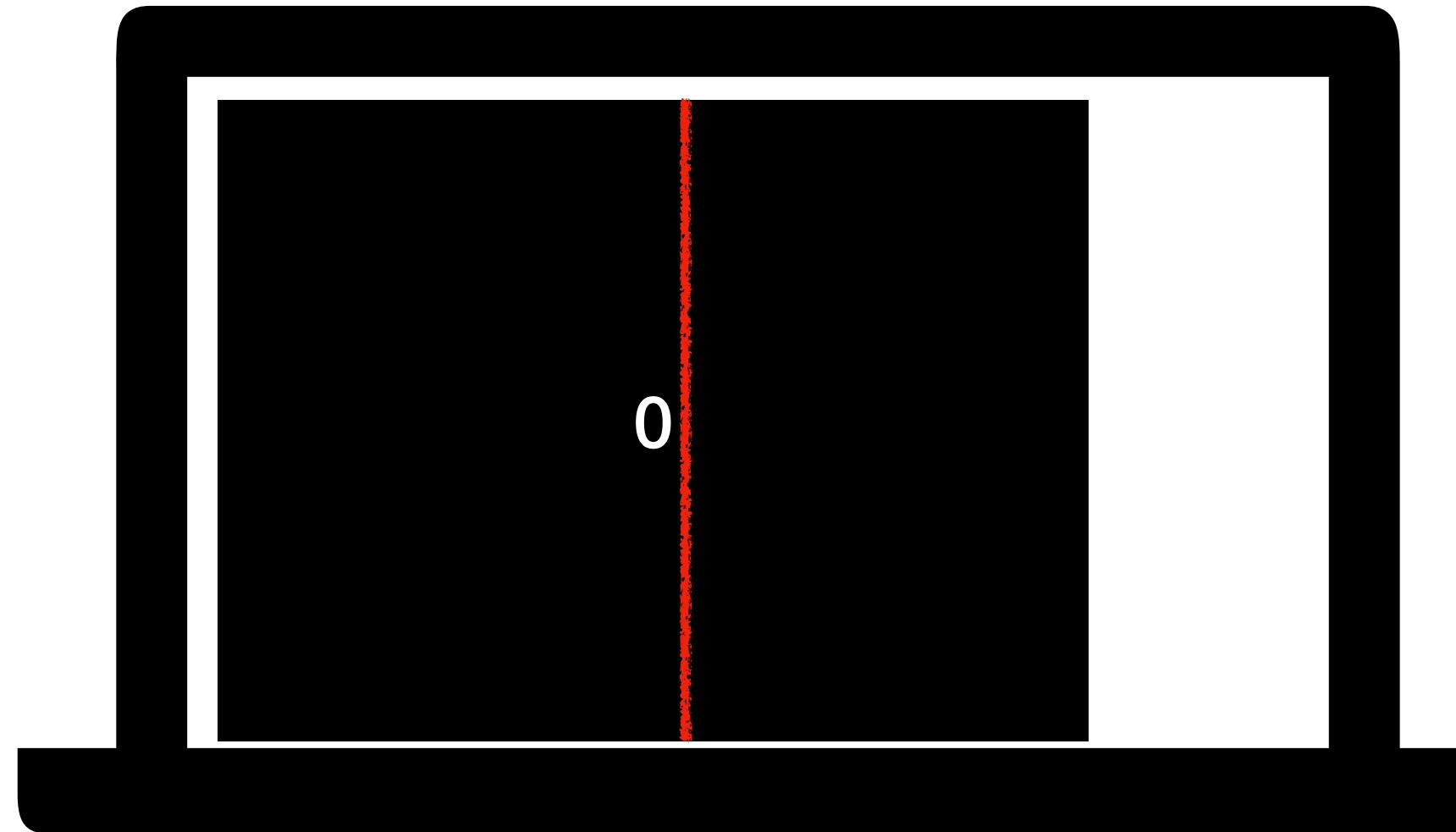


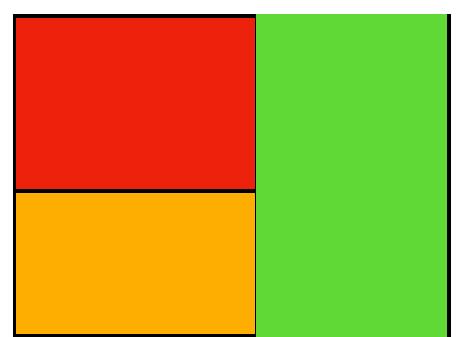
Panes

Panes are splits within windows

Split left/right
ctrl-b %

- Split windows into multiple panes
 - % left | right





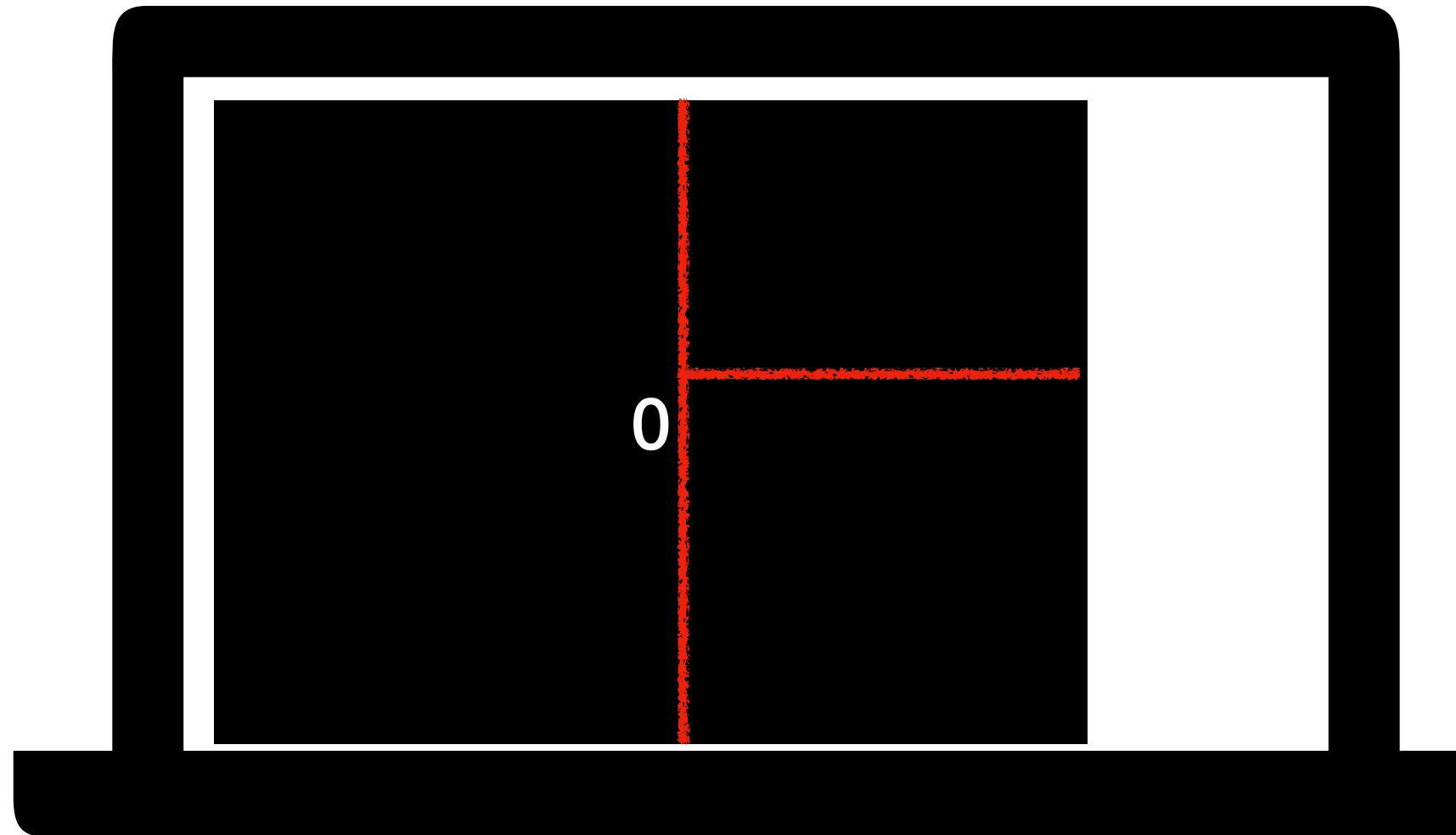
Panes

Panes are splits within windows

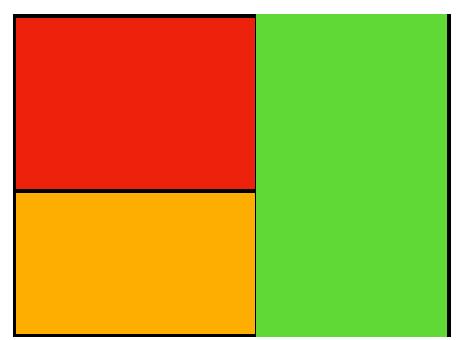
Split pane where cursor is

Split up/down
ctrl-b “

- Split windows into multiple panes
- “ up
down



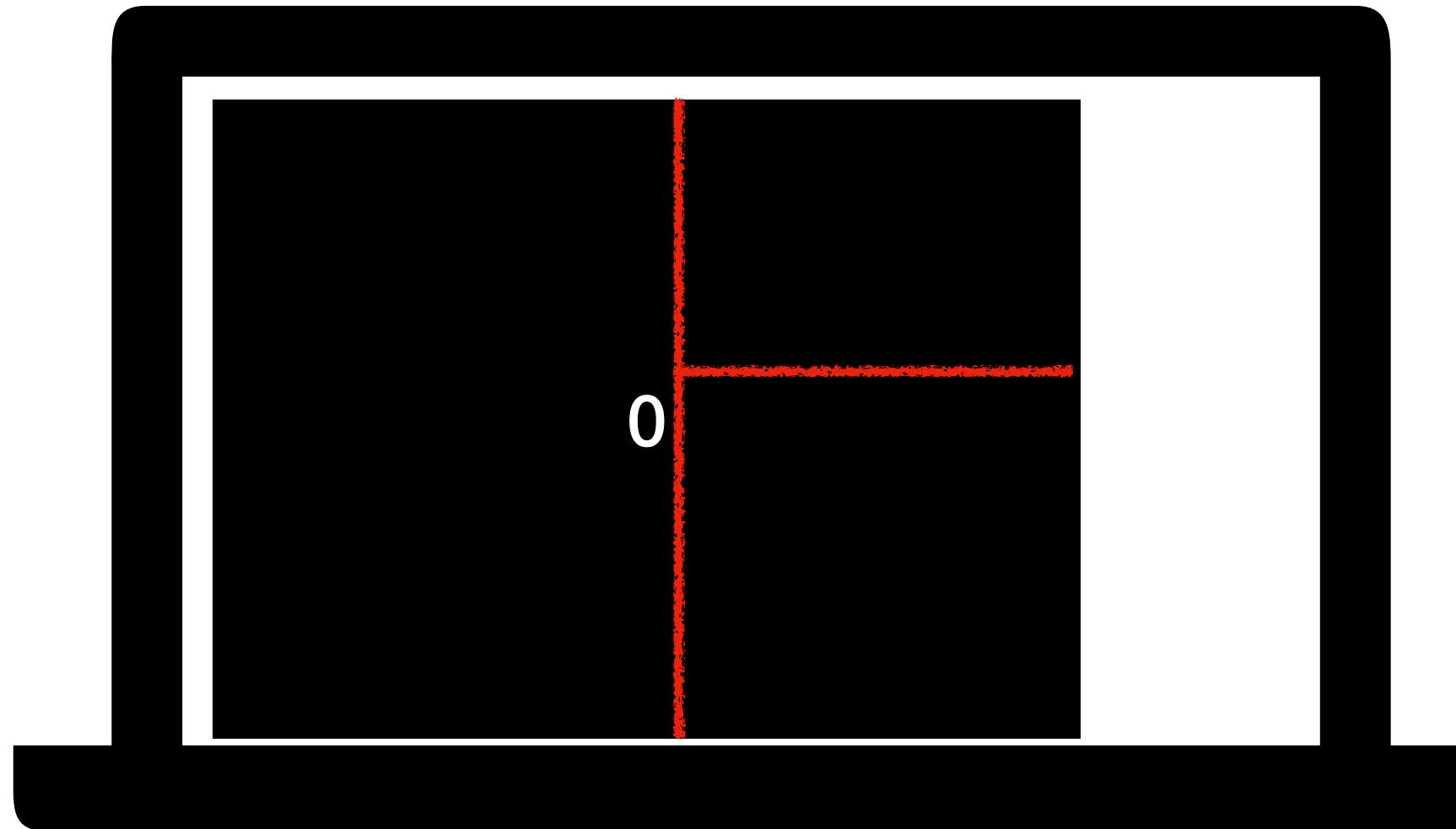
```
[1] 0:zsh* 1:zsh- "AS-C02YK3YNJGH6.fios-" 23:04 22-Oct-20
```

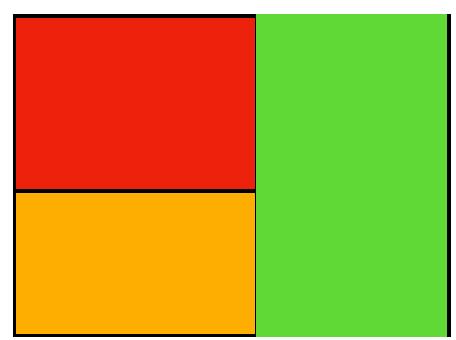


Panes

Move between panes within a window
ctrl-b <arrow keys>

- Split windows into multiple panes
- “up
down”



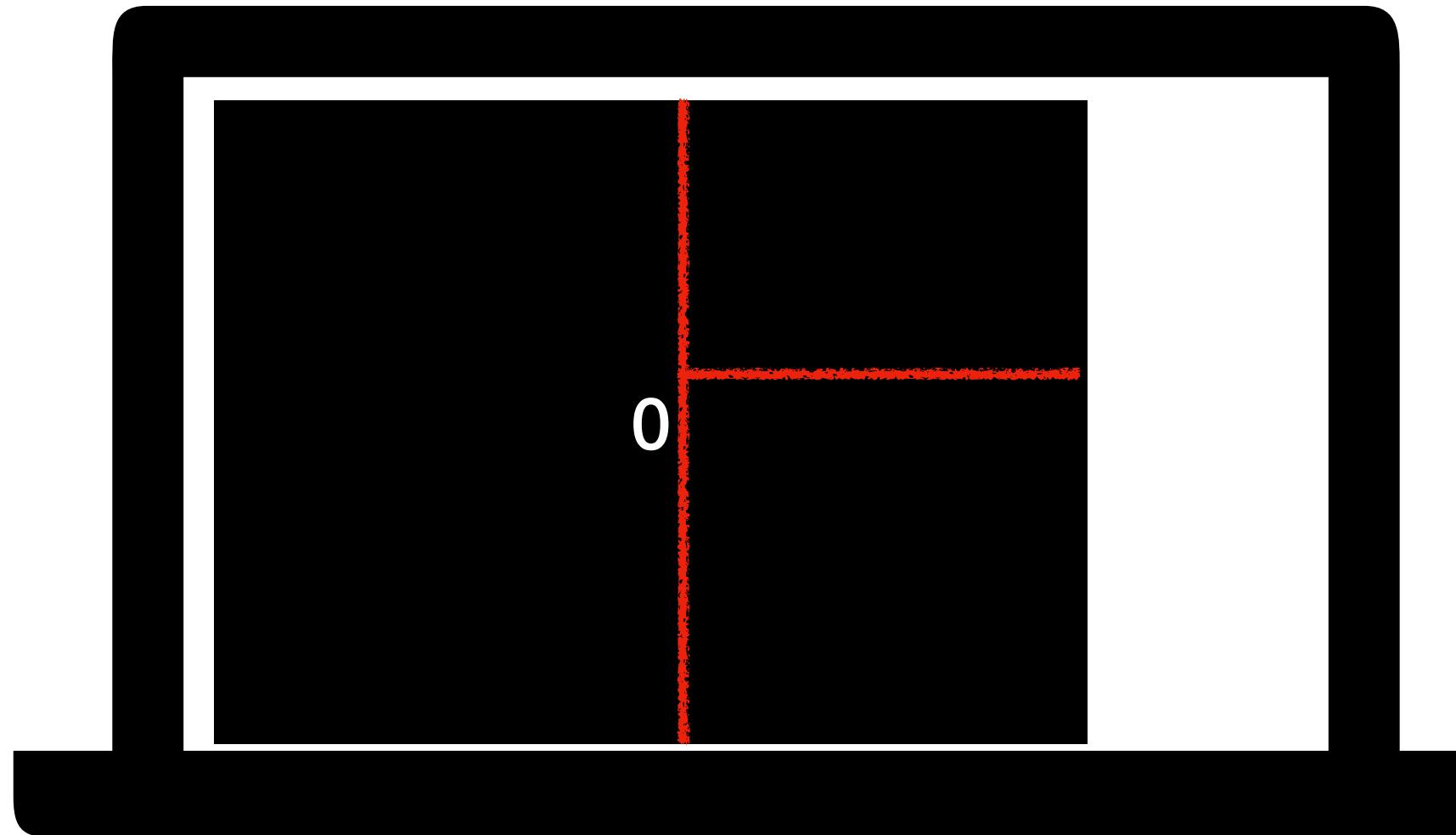


Now window 0 has 3 terminals within it, 1 in each pane

Panes

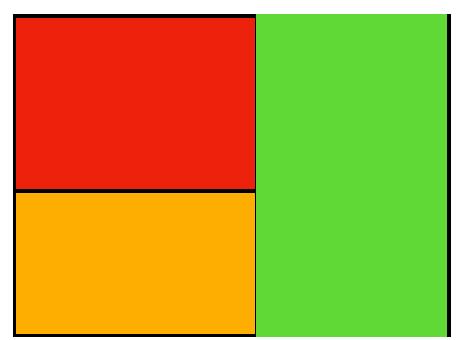
Move between panes within a window
ctrl-b <arrow keys>

- Split windows into multiple panes
- “up
down



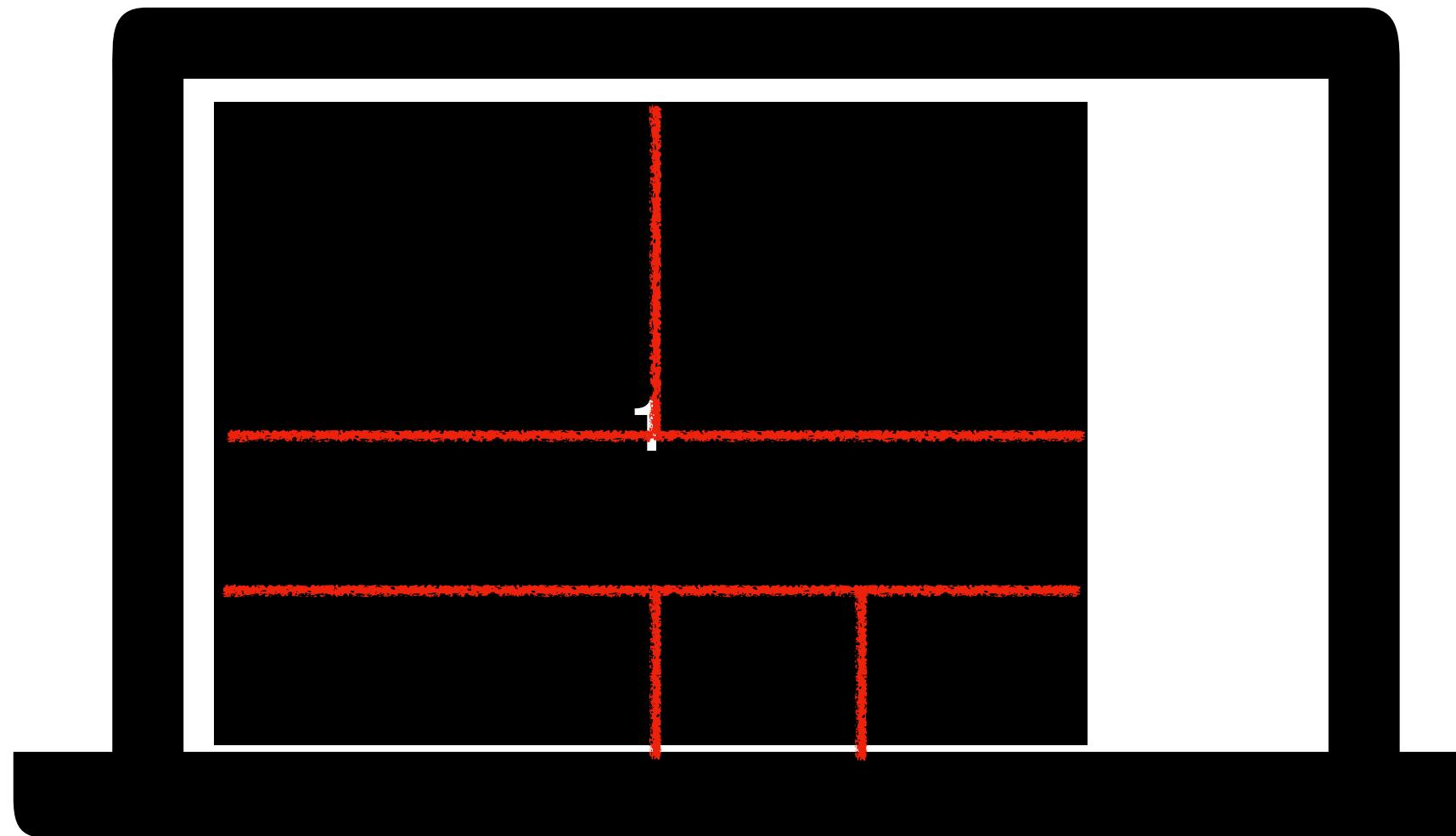
```
[1] 0:zsh* 1:zsh-
```

"AS-C02YK3YNJGH6.fios- 23:04 22-Oct-20"

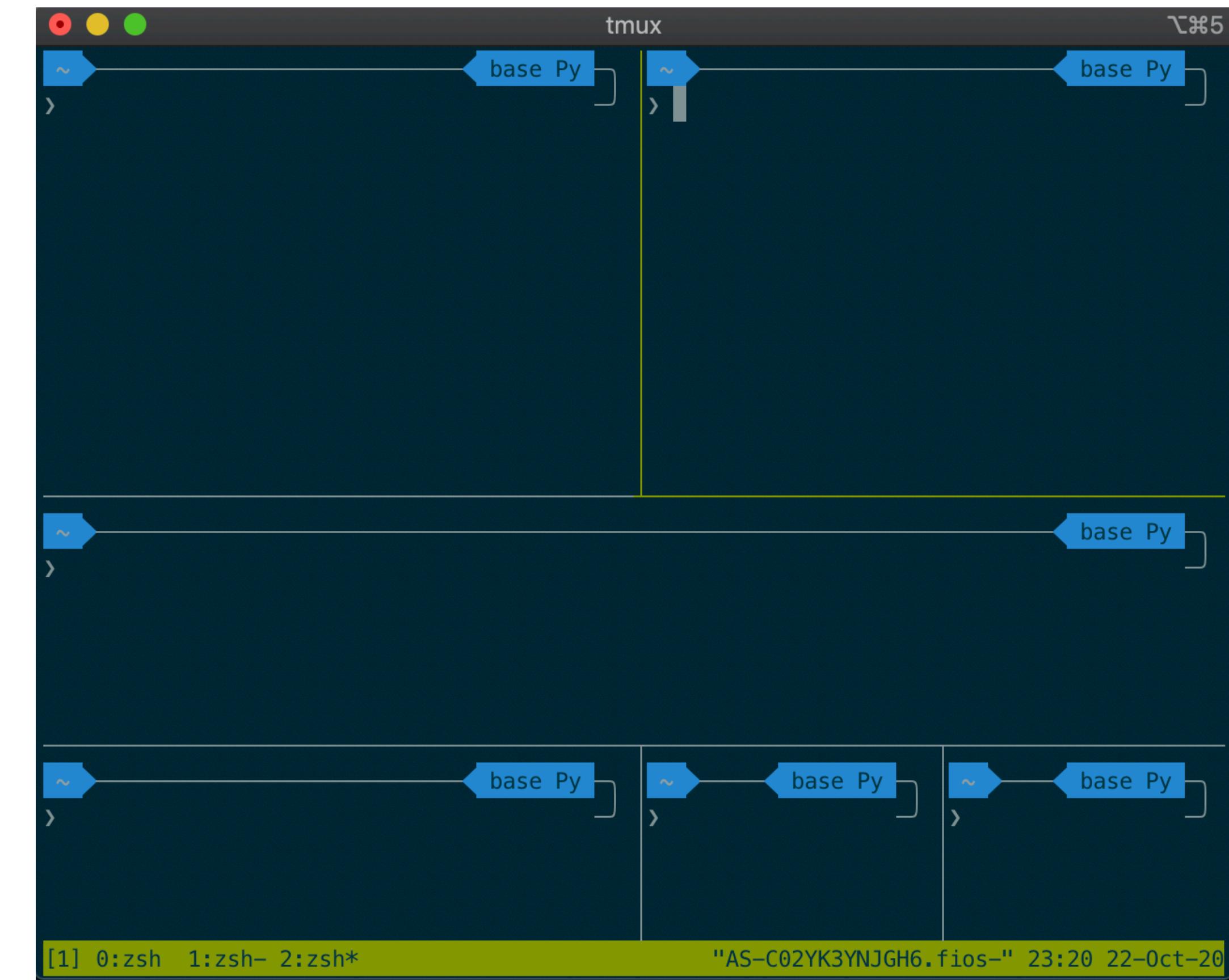


Panes

- Split windows into multiple panes



Can split in just about any pattern



Windows

- Other windows are running in background



1st window is still there

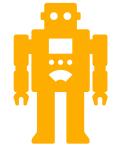
Back to first window
ctrl-b p

Has not changed

```
[1] 0:zsh- 1:zsh* "AS-C02YK3YNJGH6.fios-"  
[1] 0:zsh- 1:zsh* "AS-C02YK3YNJGH6.fios-" 22:45 22-Oct-20
```

Goes to “next” window:
ctrl-b n

* tells us where we are

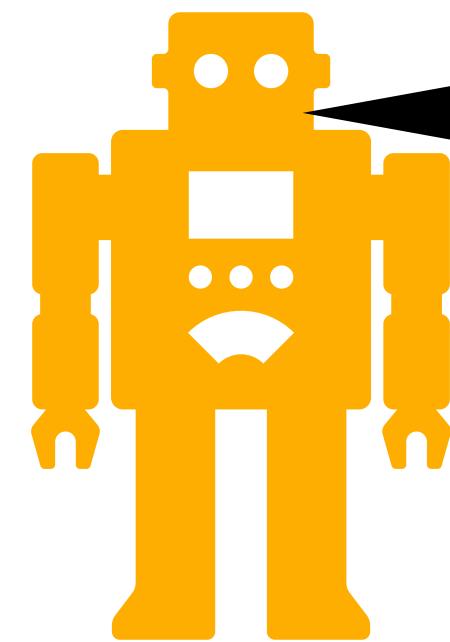


Detach

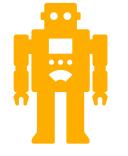
Detach a session
ctrl-b d

d for detach

- Sometimes we want to keep a processes running in the background
- tmux lets us detach the session from the process running behind the scenes.
- Then re-attach later
 - This is instead of closing a terminal window, which ends the process



I am still working



Re-attach

- Pull process back up in a pane
 - tmux ls
 - To ls tmux sessions/windows that are open

Detach a process
tmux attach -t ID

attach
-t attach “to” ID (like 1)
Or NAME (like jLab)

This is my work

Useful tmux commands

Command at Command line	Command with key strokes	What it does	What it operates on
tmux new-session NAME		Creates a new session named NAME	session
tmux ls	ctrl-b s	List out open sessions (with preview and scrollable list)	sessions
	ctrl-b \$	Rename current session	session
tmux detach	ctrl-b d	Detach current session	session
tmux attach -t ID or NAME		Attach ID or NAME to current window	session

Useful tmux commands

Command at Command line	Command with key strokes	What it does	What it operates on
	ctrl-b c	New window	Window in current session
	ctrl-b ,	Rename current window	Current window
	ctrl-b %	Split window or pane horizontally (left, right)	Window or pane
	ctrl-b “	Split window or pane vertically (up, down)	Window or pane
	ctrl-b n	Change to next window	Window
	ctrl-b p	Change to previous window	Window
	ctrl-b 0-9	Change to windows 0, 1, 2,...9	Window
	ctrl-b w	List windows (with previews and scrollable	Windows

Useful tmux commands

Command at Command line	Command with key strokes	What it does	What it operates on
	ctrl-b h	Move left	panes
	ctrl-b l	Move right	panes
	ctrl-b k	Move up	pane
	ctrl-b j	Move down	pane
	ctrl-b q	Show pane numbers	panes
	ctrl-b z	Zoom to current pane (or back out to all)	pane/panes
exit	ctrl-b x	Kill the current pain	pane
	ctrl-b :resize-panel -D or -U or -L or -R ##	Resize pane either -D down, -U up, -L left or -R right, by ## cells	pane

Note: “ctrl-b :” puts tmux in something like “normal” mode in vim

To scroll a pane: ctrl-b [Then use up or down arrows; q to quit scroll mode

tmux tips

- Do not have too many windows or panes. Humans are really bad at multitasking. It does not even look that cool. Just open what you need to be productive.
- Name your sessions and especially your windows (mine is called administrative work, no seriously), switching between 0s and 1s can get confusing.
- Start simple, with 1 window split into vim and a terminal. That is generally really helpful. Note that you can also split vim into two
- Here is how I use tmux:
 - 1 session at a time
 - 1 window per project that I am working on
 - Jupyter notebook/lab usually detached, with browser next to terminal window
 - Vim open in one large pane, taking upper 3/4 of window
 - One terminal below the Vim pane on left for running code
 - One terminal to the right either ssh'ed to a server or monitoring current usage (eg top or htop)
 - This panes are created with:
 - `ctrl-b "`
 - `ctrl-b %`
 - `ctrl-b :resize-pane -D 10`

Vim up here

tmux new-session -s pemdiv

NORMAL [unix] 100% ≡ 0/1 h:1

```
> pwd          zsh here
/Users/mpc61
> ls
AJPS_VizBag.txt      Library        VideoBackUp      permissions.plist
Applications         Movies          VirtualBox VMs   permissions_original.plist
Creative Cloud Files Music           annualReport2Dean2020.txt simpleProgram.sh
Desktop               Pictures        github           stanza_resources
Documents             Public          markdownlint    test.test
Downloads            TADDA.mp4       nltk_data       untitled.txt
Dropbox               Untitled.ipynb   node_modules    package-lock.json
ForNolan.R           Untitled1.ipynb
```

htop running here

PID	USER	PRI	NI	VIRT	RES	S	CPU%	MEM%	TIME+	Command
16416	mpc61	24	0	8346M	63272	?	33.3	0.2	56h00:08	/Applications/Camtasia 2020.app/Contents/Mac
6681	mpc61	24	0	5753M	311M	?	2.8	0.9	40:34.57	/private/var/folders/zy/4_phdmhn0q1c4b0pw0by
35705	mpc61	17	0	4308M	12512	?	1.5	0.0	0:00.48	/usr/sbin/screencapture -pdi -z cmd-shift-4
28760	mpc61	17	0	4605M	19012	?	0.9	0.1	2h36:11	/Users/mpc61/Applications/ArduinoCreateAgent
28743	mpc61	17	0	5216M	167M	?	0.8	0.5	4h41:24	/Applications/RStudio.app/Contents/Framework

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

"AS-C02YK3YNJGH6.fios-" 20:03 24-oct-20

[pemdiv] 0:zsh*

tmux.conf

In home director

- Like .zshrc and .vimrc
- Can change behavior of tmux
 - Example: tmux uses bash and reads from .bash_profile by default.
 - I change this in my tmux.conf with the line:
 - set -g default-command “/bin/zsh”
 - You should input the location of your shell

Vim-itation

Vim also has a language for windows and panes

- It gets complicated though because vim operates through buffers.
 - These are text/data objects in memory. They can be views on files on disk, but they can also not yet be saved, or changed versions.
 - Vim can have multiple buffers open by using `:e PathToFile` in normal mode (command). These are in two windows, so you will not see them at the same time.
 - “:e” is for edit. Can also do :enew for a new buffer not based on a file
 - :bn and :bp toggle to the “buffer next” and “buffer previous”
 - :bd deletes a buffer
 - I find it better to have multiple “panes” open WITHIN a window than to have multiple buffers like this or even instances of vim running in different tmux panes within a given window/project.

Vim-itation

Vim also has a language for windows and panes

- I find it better to have multiple “panes” open WITHIN a window than to have multiple buffers like this or even instances of vim running in different tmux panes within a given window/project.
- `:vsplit PathToFile` creates two panes, left and right
- `:split PathToFile` creates two panes, up and down
- Use `ctrl-w-w` (hold control press w twice) to switch between the panes that are holding buffers in the same window.
- Can use `:w` to write out, as well as `:q`

Vim up here

```

1 ---  

2 title: Intro to Python  

3 author: Michael Colaresi  

4 ---  

5  

6 # This directory includes some example code to learn python  

7  

8 ## Introduction to Python Objects  

9  

10 The files  

11  

12 - pythonObjects1.ipynb -- includes basic types in python  

13  

14 - pythonImmutableObjects.ipynb -- provides examples of tuples and strings and why immutability can be useful  

15  

16  

17 ## Functions and module development  

18  

19 - `Intro2Functions.ipynb` -- a jupyter notebook that walks through functions in python and creates a smoothie function with a docstring, a test, and type hinting  

20  

21 - `smoothie.py` -- this file is an example module file. The function was developed in Intro2Functions.ipynb.  

22 - A few changes were made including matching naming conventions to snake_case.  

23 - In addition a short test for the function is protected by `__name__ == "__main__"`

```

One file on right

Related file on the left

```

1 #! /usr/bin/env python  

1 """a module file to make a string smoothie  

2 functions:  

3     smoothie  

4     main  

5 requirements:  

6     typing  

7 """  

8  

9 import typing  

10  

11  

12 def smoothie(s_1: str, s_2: str) -> str:  

13     """function to blend two strings together  

14     to make a a string smoothie  

15     Args:  

16         s1 (str): first string to blend  

17         s2 (str): second string to blend  

18     Returns:  

19         str: a blended version of s1 and s2  

20     """  

21     assert isinstance(s_1, str), "s1 needs to be a str"  

22     assert isinstance(s_2, str), "s2 needs to be a str"  

23     glass: str = ""  

24     for i, j in zip(s_1, s_2[::-1]):  

25         glass += j + i  

26     return glass

```

NORMAL ➤ master# <dMe.md markdown < utf-8[unix] < 164 words < 3% ≡ 1/32 N:1 W:9(L15)≡ [32]tra... smoothie.py python utf-8[unix] 2% ≡ 1/37 N:1 "readMe.md" 32L, 1098C

ls -a

zsh here

```

. Onward2Victory.pdf      readMe.html
.. data-science-from-scratch    readMe.md
.ipynb_checkpoints        format_lint_typeCheck_test_py.sh smoothie.py
.mypy_cache                python-book           test.py
Intro2Functions.ipynb       pythonImmutableObjects.ipynb
PythOnward2Victory.key     pythonObjects1.ipynb

```

~/Dropbox/ScientificComp4SocSciClass/sandBox/IntroPython ➤ master !17 ?30 ➤ base Py

htop running here

PID	USER	PRI	NI	VIRT	RES	S	CPU%	MEM%	TIME+	Command
16416	mpc61	24	0	8346M	63596	?	29.4	0.2	56h08:33	/Applications/Camtasia 2020.app/Contents/Mac
28704	mpc61	17	0	86.5G	539M	?	10.9	1.6	6:18.29	/System/Library/Frameworks/WebKit.framework/
6681	mpc61	24	0	5755M	313M	?	2.7	1.0	41:31.96	/private/var/folders/zy/4_phdmhn0q1c4b0pw0by
28743	mpc61	17	0	5223M	162M	?	0.8	0.5	4h41:37	/Applications/RStudio.app/Contents/Framework
584	mpc61	17	0	4323M	6728	?	0.6	0.0	34:02.06	/System/Library/PrivateFrameworks/ViewBridge

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

"AS-C02YK3YNJGH6.fios-" 20:33 24-Oct-20

One last tip

Open a file in two panes

- It can be very helpful to open 1 file in two panes (eg left and right panes)
- For example you can look at two (or more) different parts of a file at the same time!
- You make a change in one pane... it shows up in the other pane (they share a view on a buffer)
- Like this...

Same file open on left and right

```

26
25
24 def smoothie(s_1: str, s_2: str) -> str:
23     """function to blend two strings together
22     to make a a string smoothie
21     Args:
20         s1 (str): first string to blend
19         s2 (str): second string to blend
18     Returns:
17         str: a blended version of s1 and s2
16     """
15     assert isinstance(s_1, str), "s1 needs to be a str"
14     assert isinstance(s_2, str), "s2 needs to be a str"
13     glass: str = ""
12     for i, j in zip(s_1, s_2[::-1]):
11         glass += j + i
10     return glass
9
8
7 def main() -> None:
6     """Function to run some tests, not to be called when imported"""
5     assert smoothie("ba", "ab") == "bbaa", "something is wrong"
4     print("module test passed \U0001F44D")
3
2
1 if __name__ == "__main__":
37   main()

```

Same file open on left and right

```

1 #! /usr/bin/env python
1 """a module file to make a string smoothie
2 functions:
3     smoothie
4     main
5 requirements:
6     typing
7 """
8
9 import typing
10
11
12 def smoothie(s_1: str, s_2: str) -> str:
13     """function to blend two strings together
14     to make a a string smoothie
15     Args:
16         s1 (str): first string to blend
17         s2 (str): second string to blend
18     Returns:
19         str: a blended version of s1 and s2
20     """
21     assert isinstance(s_1, str), "s1 needs to be a str"
22     assert isinstance(s_2, str), "s2 needs to be a str"
23     glass: str = ""
24     for i, j in zip(s_1, s_2[::-1]):
25         glass += j + i
26     return glass

```

NORMAL ➤ master# ➤ smoothie.py python utf-8[unix] 100% 37/37 h:1 W:2(L10) ➤ smoothie.py python utf-8[unix] 2% 1/37 h:1

```

> ls -a
.
.. PythOnward2Victory.pdf      readMe.html
.ipynb_checkpoints             data-science-from-scratch    readMe.md
.mypy_cache                      format_lint_typeCheck_test_py.sh smoothie.py
Intro2Functions.ipynb           python-book                  test.py
PythOnward2Victory.key          pythonImmutableObjects.ipynb

```

~/Dropbox/ScientificComp4SocSciClass/sandBox/IntroPython ➤ master !17 ?30 ➤ base Py

```

1 [|||||] 33.8% 4 [|] 0.7% 7 [|||] 16.0% 10 [|] 0.7%
2 [|] 0.7% 5 [|||||] 19.2% 8 [|] 0.7% 11 [|||] 9.4%
3 [|||||] 26.0% 6 [|] 0.7% 9 [|||] 12.6% 12 [ 0.0%
Mem[|||||||||||||||||] 14.8G/32.0G Tasks: 696, 2427 thr; 1 running
Swp[|||||||||||||] 11.1G/12.0G Load average: 2.62 2.36 2.11
Uptime: 188 days(!), 07:30:59

```

PID	USER	PRI	NI	VIRT	RES	S	CPU%	MEM%	TIME+	Command
16416	mpc61	24	0	8346M	63652	?	36.9	0.2	56h12:27	/Applications/Camtasia 2020.app/Contents/Mac
39120	mpc61	17	0	4316M	12436	?	6.2	0.0	0:00.47	/usr/sbin/screencapture -pdi -z cmd-shift-4
6681	mpc61	24	0	5753M	372M	?	4.3	1.1	41:50.90	/private/var/folders/zy/4_phdmhn0q1c4b0pw0by
36011	mpc61	17	0	86.7G	551M	?	3.5	1.7	3:05.11	/System/Library/Frameworks/WebKit.framework/
28743	mpc61	17	0	5215M	163M	?	1.1	0.5	4h41:43	/Applications/RStudio.app/Contents/Framework

F1Help F2Setup F3Search F4Filter F5Tree F6SortByF7Nice - F8Nice + F9Kill F10Quit

[pemdiv] 0:vim*

"AS-C02YK3YNJGH6.fios-" 20:46 24-Oct-20