Instructions for the “Raspberry Pi” module

Martin L. Purschke

purschke@bnl.gov

What we'll do

1. First, I need you to prepare your shell environment with a few simple things (we did that on purpose to make it more interesting ☺)
2. We'll play with the shell a little bit (if you know all that in your sleep, move on)
3. We'll play with the rcdaq data acquisition for a while
4. If we get that far, we'll look at actual beam data

Don't feel bad if you don't get to do everything! It is more important that you get a good understanding, rather than rush through the material

And you can buy your own RPi and do it at home when you have time!

About your Raspberry Pi

You will see a sticker on your Pi such as “pi19”

I will at times refer to a “piXX” - replace the XX with your number

So for now, log into “your” pi:

```
ssh  -l pi piXX
```

or

```
ssh  pi@piXX
```

Both versions are equivalent.

ssh is the “secure shell”

-l pi means “log in as the user pi”. The password is “CapeTown”

We need to prepare our environment. (Do not skip this!)

First check that our environment in not prepared yet. **The previous group may have done it already!**

Type:

```
echo $DISPLAY
```

If you see the answer "localhost:0.0" or so, that's good. But we are not done yet:

Type:

```
cat .ssh/config
```

If you get some output like

```
ForwardAgent yes
ForwardX11 yes
ForwardX11Trusted yes
NoHostAuthenticationForLocalhost yes
```

then you are in luck! Move on. Skip the next page.

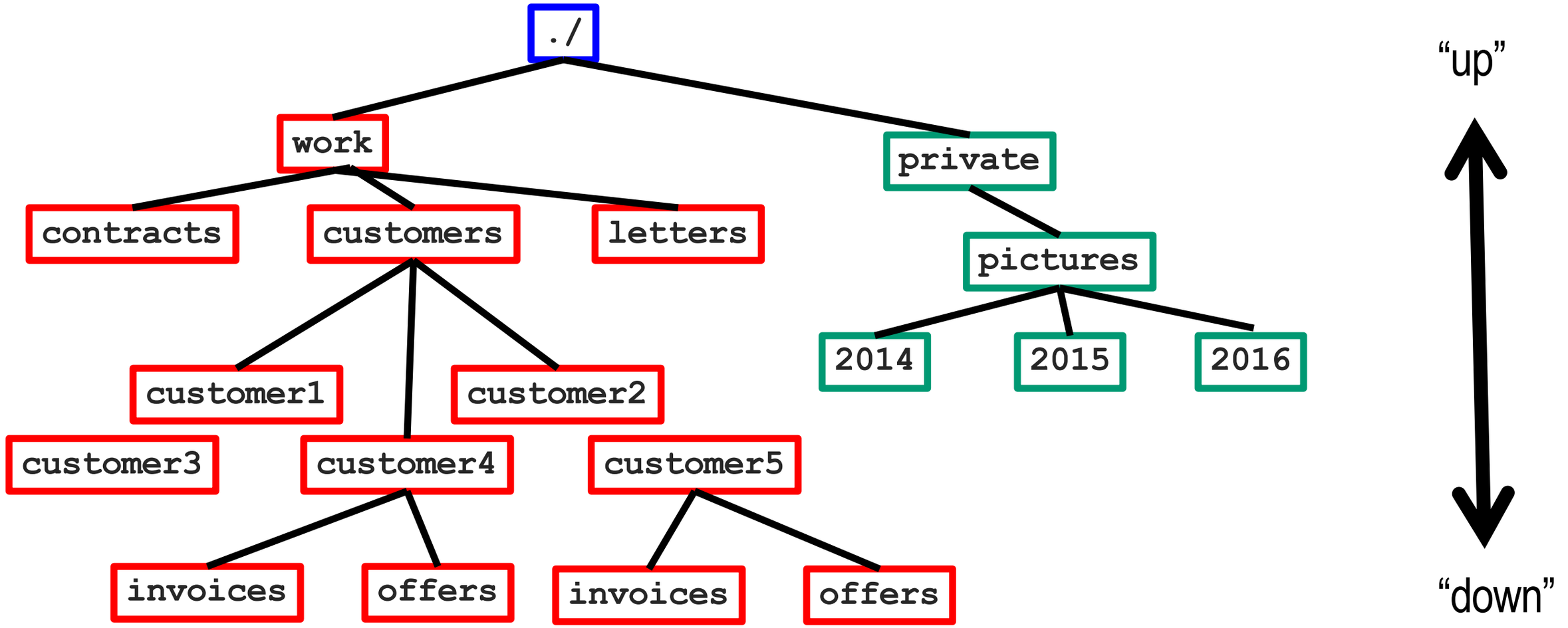
If your environment was not prepared yet, do this:

Make sure you are in the home directory.

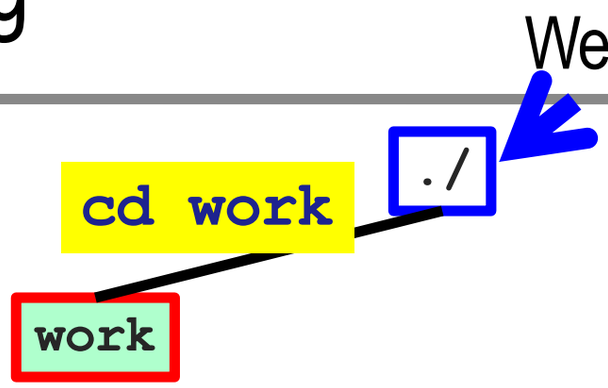
Type (you did pay attention in my lecture, didn't you?):

```
cd
wget -q -O - http://www.phenix.bnl.gov/~purschke/addons.tar.gz | tar xvz
```

Now follow the instruction from my lecture… navigate the tree like I showed you

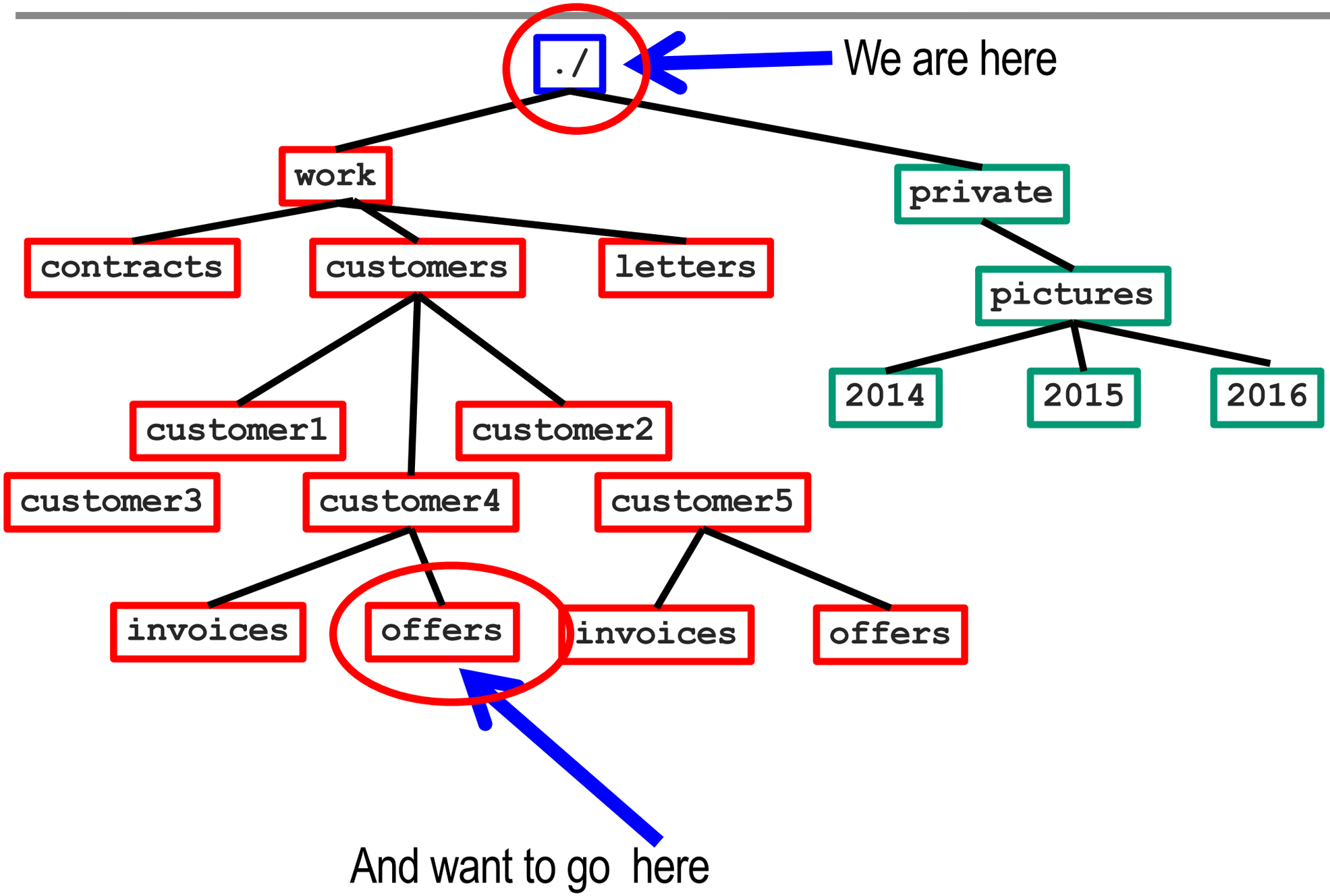


Navigating

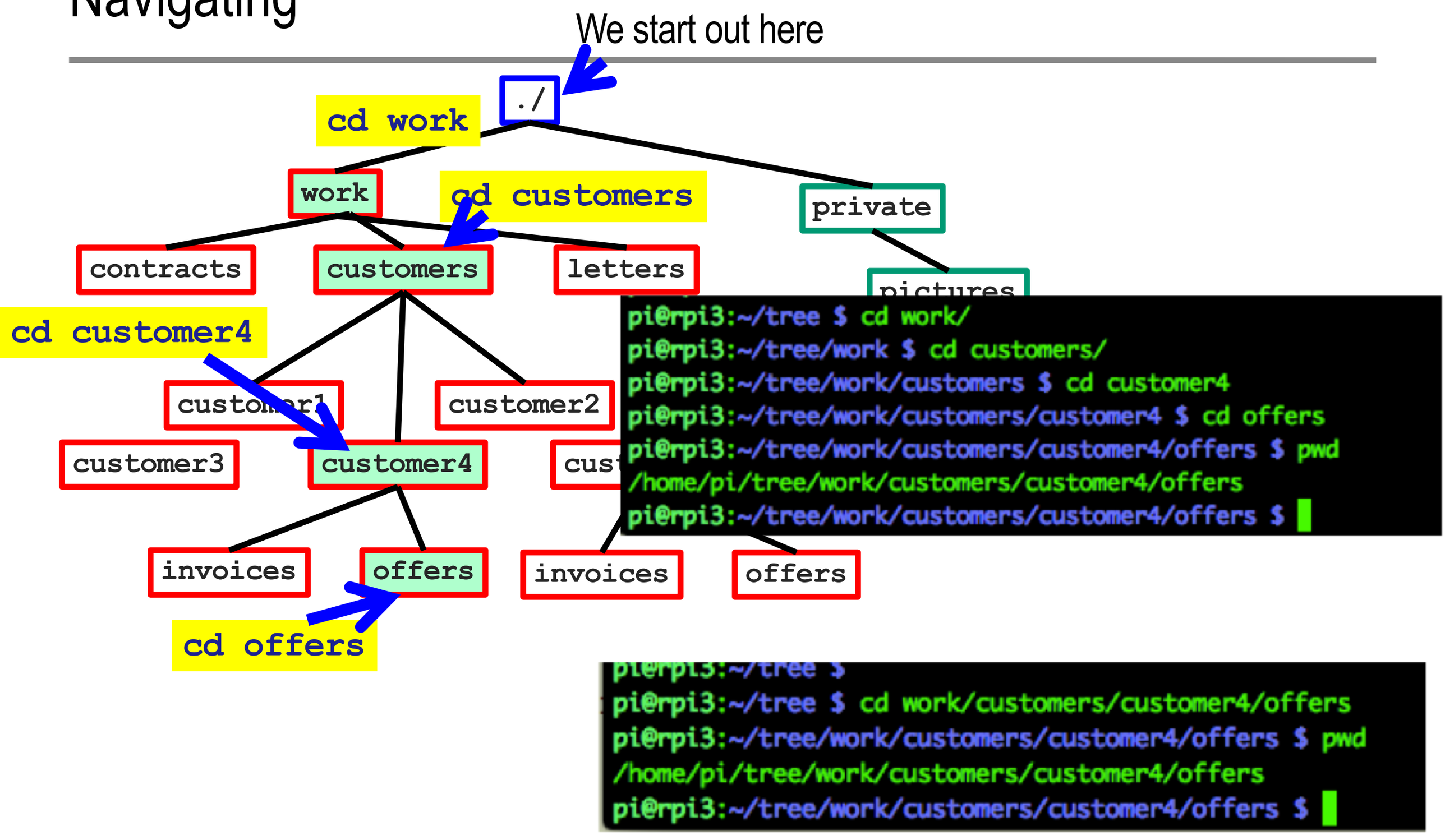



```
pi@rpi3:~/tree $ pwd
/home/pi/tree
pi@rpi3:~/tree $ cd work
pi@rpi3:~/tree/work $ pwd
/home/pi/tree/work
pi@rpi3:~/tree/work $
```

Navigating a directory tree



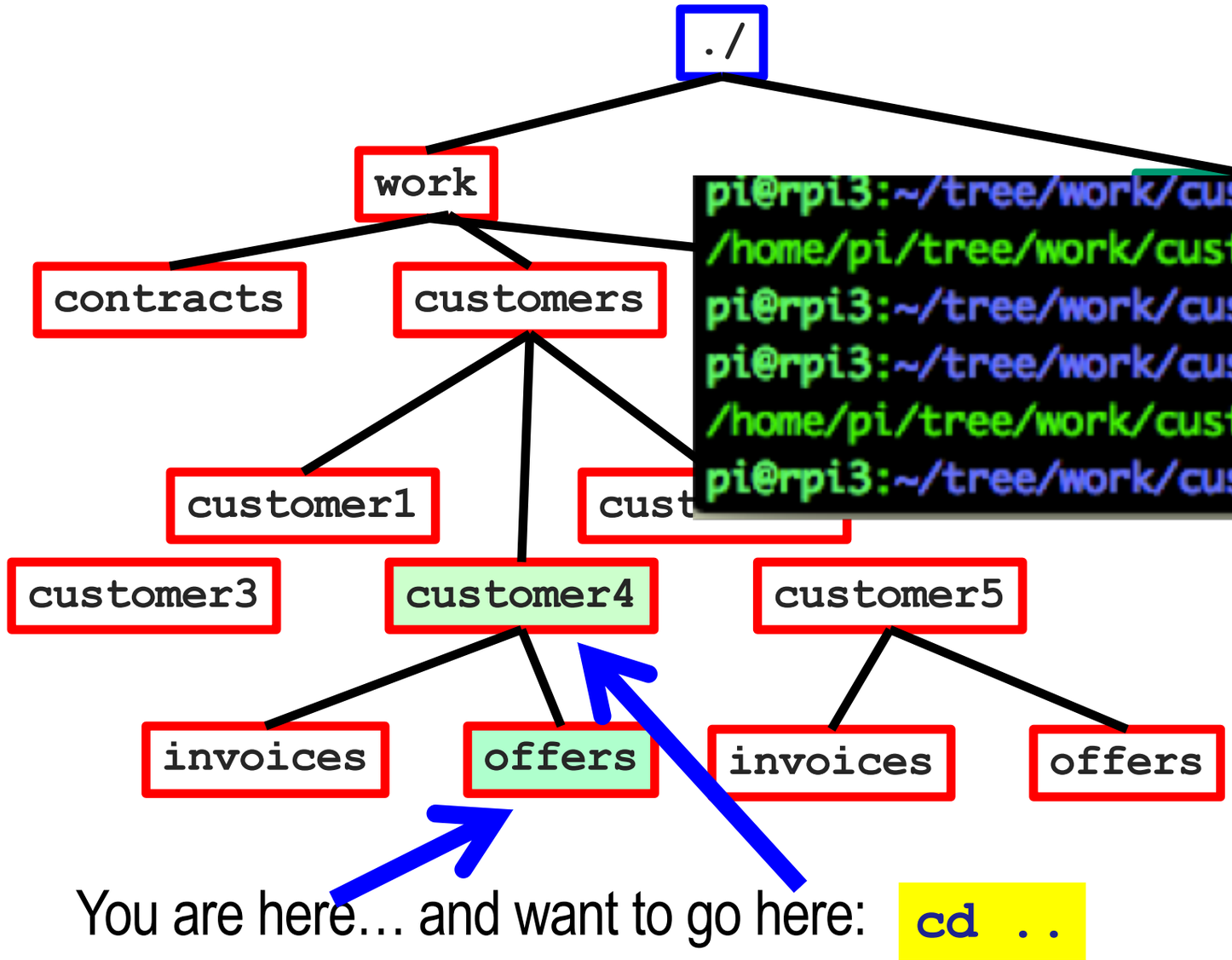
Navigating


We start out here
./
cd work
work
cd customers
private
contracts
customers
letters
pictures
cd customer4
customer1
customer2
customer3
customer4
cust
invoices
offers
invoices
offers
cd offers
pi@rpi3:~/tree $ cd work/
pi@rpi3:~/tree/work $ cd customers/
pi@rpi3:~/tree/work/customers $ cd customer4
pi@rpi3:~/tree/work/customers/customer4 $ cd offers
pi@rpi3:~/tree/work/customers/customer4/offers $ pwd
/home/pi/tree/work/customers/customer4/offers
pi@rpi3:~/tree/work/customers/customer4/offers $
pi@rpi3:~/tree $
pi@rpi3:~/tree $ cd work/customers/customer4/offers
pi@rpi3:~/tree/work/customers/customer4/offers $ pwd
/home/pi/tree/work/customers/customer4/offers
pi@rpi3:~/tree/work/customers/customer4/offers $

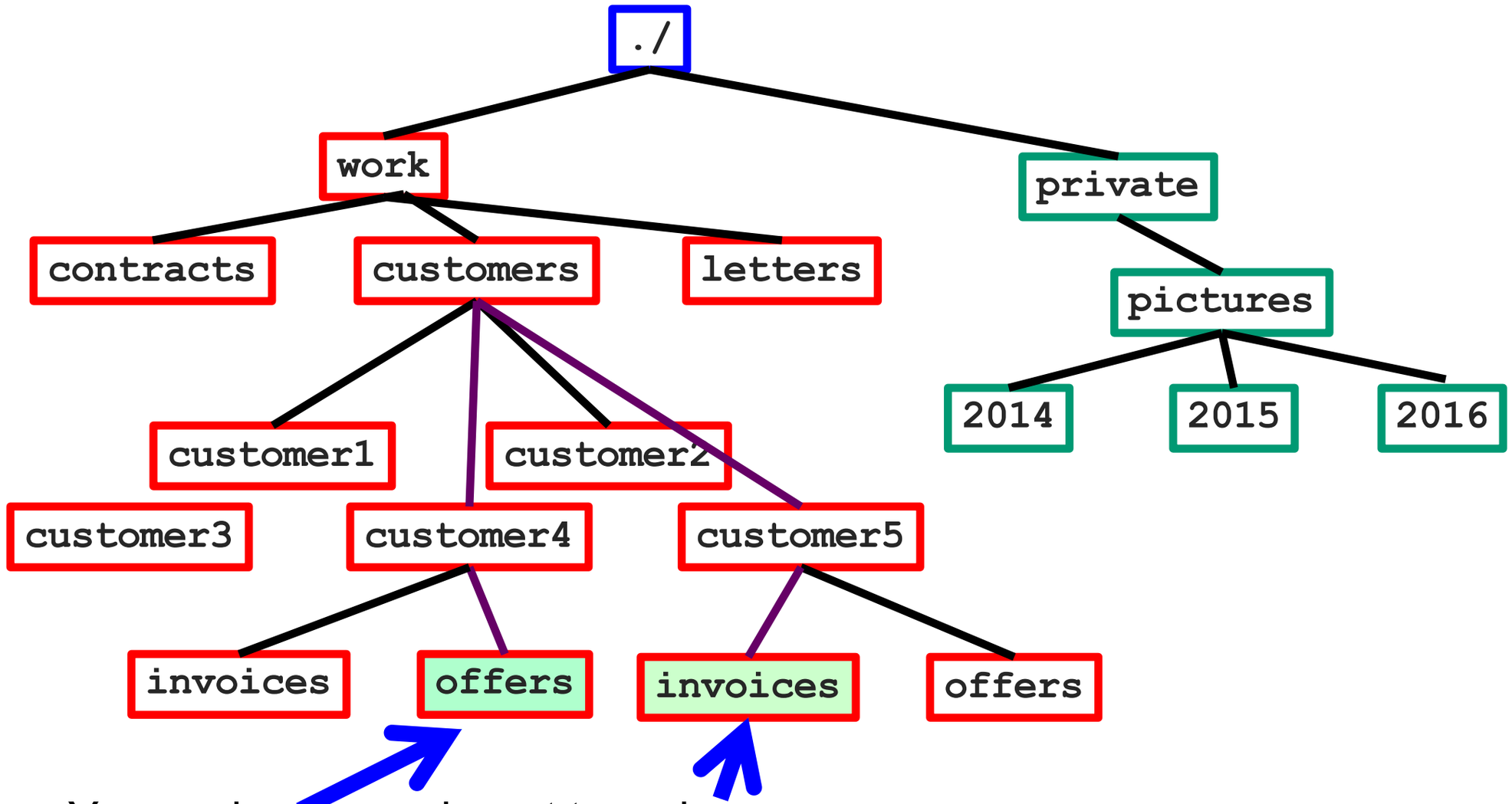
Navigating: . and ..

"." is a shorthand for "here"

".." is "one level up"



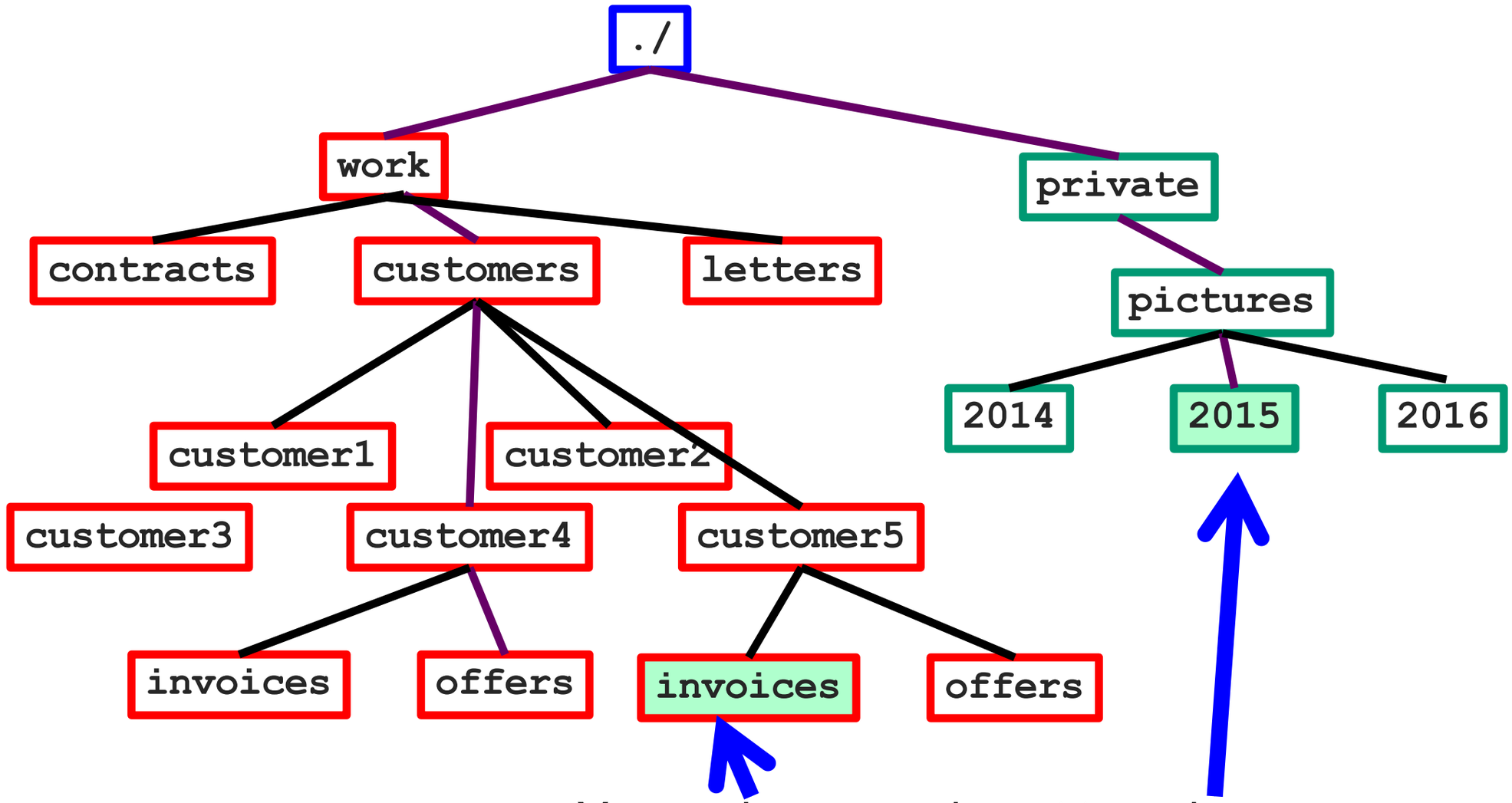
Navigating: combining “up” and “down”



You are here… and want to go here:

```
pi@rpi3:~/tree/work/customers/customer4/offers $ pwd
/home/pi/tree/work/customers/customer4/offers
pi@rpi3:~/tree/work/customers/customer4/offers $ cd ../../customer5/invoices/
pi@rpi3:~/tree/work/customers/customer5/invoices $ pwd
/home/pi/tree/work/customers/customer5/invoices
pi@rpi3:~/tree/work/customers/customer5/invoices $
```

Navigating: combining “up” and “down”



You are here… and want to go here:

```
pi@rpi3:~/tree/work/customers/customer5/invoices $ pwd
/home/pi/tree/work/customers/customer5/invoices
pi@rpi3:~/tree/work/customers/customer5/invoices $ cd ../../../../private/pictures/2015
pi@rpi3:~/tree/private/pictures/2015 $ pwd
/home/pi/tree/private/pictures/2015
pi@rpi3:~/tree/private/pictures/2015 $
```

Use the tab key!

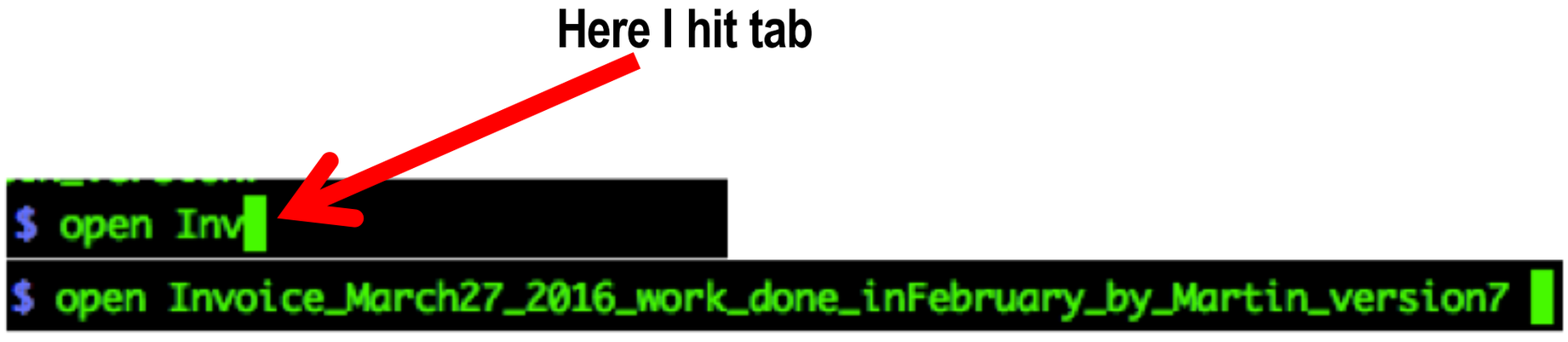
When you type something, the tab key will expand this as much as possible

It saves you tons of typing! And saves mistakes!

You will see seasoned shell users hit tab all the time

Let's say you have a file called

`Invoice_March27_2016_work_done_inFebruary_by_Martin_version7`



Use ls –l

“ls” lists the files in a directory

ls –l adds more information “ls dash ell”

Try “man ls” (get out with “q”)

Try “ls –ltr” (and find out what that does…)

Try “ ls –lSr”

Check the man pages for other cool options. Play around some.

“man <command>” gives you a brief manual of a given command. E.g.

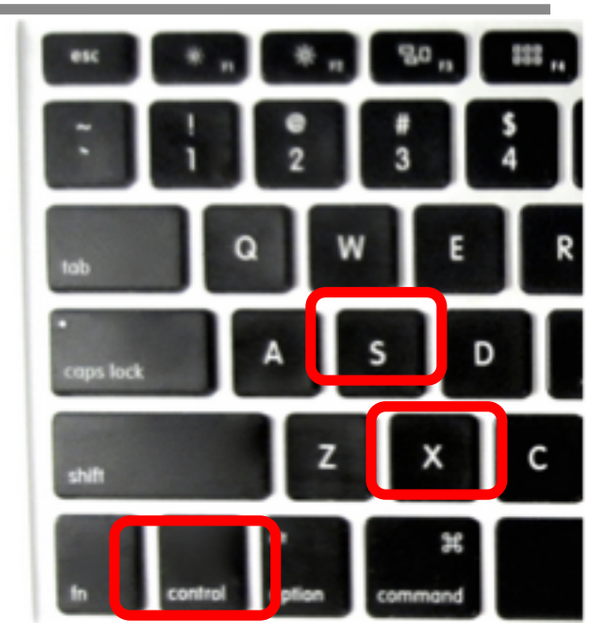
“man find”

Use “emacs –nw my_script.sh” to generate the script

Type the commands as you see them here

Then use CTRL-x-s (hold the control key and hit x, then s) to save the file

(made a mistake with the controls? Hit CTRL-g to clear)




```
pwd
cd work
pwd
cd ..
pwd
```

```
pi@rpi3:~/tree $ sh my_script.sh
/home/pi/tree
/home/pi/tree/work
/home/pi/tree
pi@rpi3:~/tree $
```

```
pi@rpi3:~/tree $ chmod +x my_script.sh
pi@rpi3:~/tree $ ./my_script.sh
/home/pi/tree
/home/pi/tree/work
/home/pi/tree
pi@rpi3:~/tree $
```

We add one more line

```
#! /bin/sh

pwd
cd work
pwd
cd ..
pwd
```

Use “wc” – “word count”

wc counts lines, words, and characters in a file

```
wc my_script.sh
wc -l my_script.sh

ls -l | wc -l

find . -type f | wc -l
```

Shell variables

You can store values in “environmental variables”

```
VARIABLE=value
```

You can then retrieve the stored value by $VARIABLE

Let’s see what this gives:

```
echo $DISPLAY

echo $SSH_AUTH_SOCK

echo $LANG

printenv
```

Try this one:

"sed" is the **"streamline editor"**

It is an enormously powerful editor that takes the input, does something to it, outputs – that's an actual filter! (you would not use it for editing some big thing….)

Remember echo? It takes the argument and prints it to stdout:

```
$ echo Martin
Martin
```

Here we go – we tell sed to substitute "a" for an "u":

```
$ echo Martin | sed 's/a/u/'
Murtin
```

But sed again prints its output to stdout, so we can go on:

```
$ echo Martin | sed 's/a/u/' | sed 's/i/e/'
Murten
```

And we can go on like this – "tr" translates one group or characters to another one (here: make everything uppercase)

```
$ echo Martin | sed 's/a/u/' | sed 's/i/e/' | tr a-z A-Z
MURTEN
```

A super-useful program: “awk” – what does that even mean?

Use “date” and pipe into awk to extract the “2018” field (or any other)

```
pi@rpi3:~$ date
Tue 10 Jul 09:52:23 EDT 2018

pi@rpi3:~$ date | awk '{print ?????????? ....
```

sort

Try

```
du  *

du  * | sort -n

du  * | sort -rn
```

bc - arbitrary precision (try 100, 500, 5000 precision…)

Arctan(1) = pi/4

```
$ bc -lq
4*a(1)
3.14159265358979323844

scale=1000
4*a(1)
3.14159265358979323846264338327950288419716939937510582097494459230\
78164062862089986280348253421170679821480865132823066470938446095505\
82231725359408128481117450284102701938521105559644622948954930381964\
42881097566593344612847564823378678316527120190914564856692346034861\
04543266482133936072602491412737245870066063155881748815209209628292\
54091715364367892590360011330530548820466521384146951941511609433057\
27036575959195309218611738193261179310511854807446237996274956735188\
57527248912279381830119491298336733624406566430860213949463952247371\
90702179860943702770539217176293176752384674818467669405132000568127\
14526356082778577134275778960917363717872146844090122495343014654958\
53710507922796892589235420199561121290219608640344181598136297747713\
09960518707211349999983729780499510597317328160963185950244594553469\
08302642522308253344685035261931188171010003137838752886587533208381\
42061717766914730359825349042875546873115956286388235378759375195778\
18577805321712268066130019278766111959092164201988
```

Install “xclock”

Let’s use a graphics program called “xclock” to test our “X display”

I use this as a convenient tool to check that the graphics display works

```
which xclock      [ should give nothing, not there ]

sudo apt-get install xclock
```

Still with me? Good.

Let’s run rcdaq now for a bit

Download the RCDAQ manual at http://www.phenix.bnl.gov/~purschke/rcdaq/rcdaq_doc.pdf

Or from our indico site

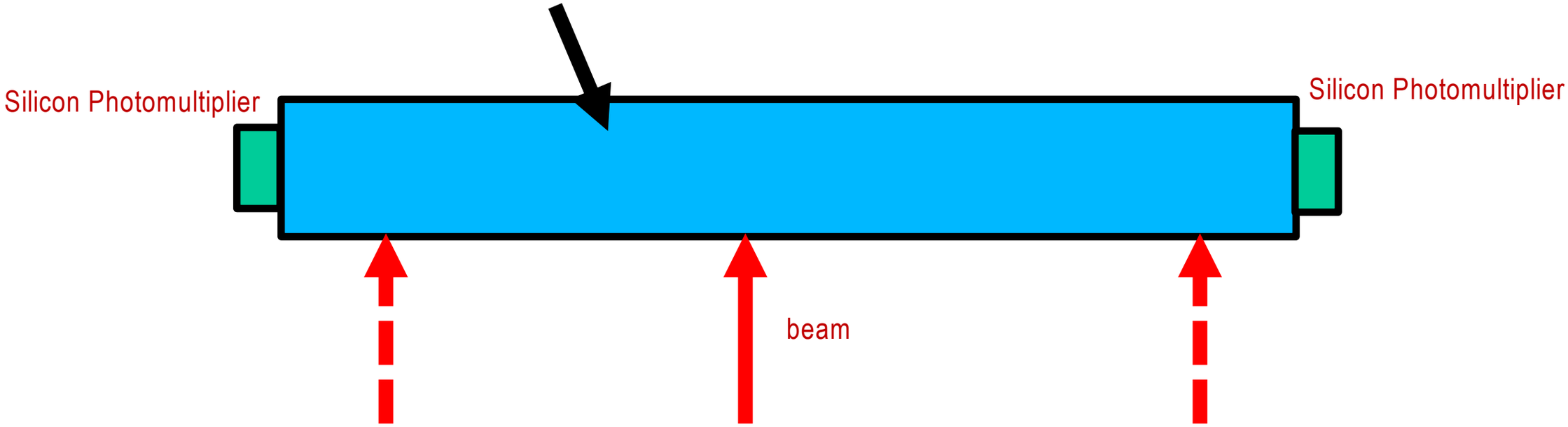
Goto page 8 chapter 5 and follow the steps, one by one

Want to go on?

Let’s analyze some data

I took data at the Fermilab test beam facility earlier this year

We had a tungsten-scintillator block in the beam to test what we call the “dual sided readout”



Can we use either the signal height “left vs right” or the time difference to get the position along the block?

Download data

Let’s analyze some data

Visit https://www.phenix.bnl.gov/WWW/publish/purschke/SASchool/

Use wget to download the “onebuffer_00109-0000.evt file

Run

```
ddump -p 1001  /media/psf/Home/data/PWO_2018/onebuffer_00109-0000.evt
     | sed 's/|//' | tail -n +3 > r109.csv
```

Then import this, for now, into libreoffice (calc) and plot ”ch0” and “ch1”

We will take it from there.