

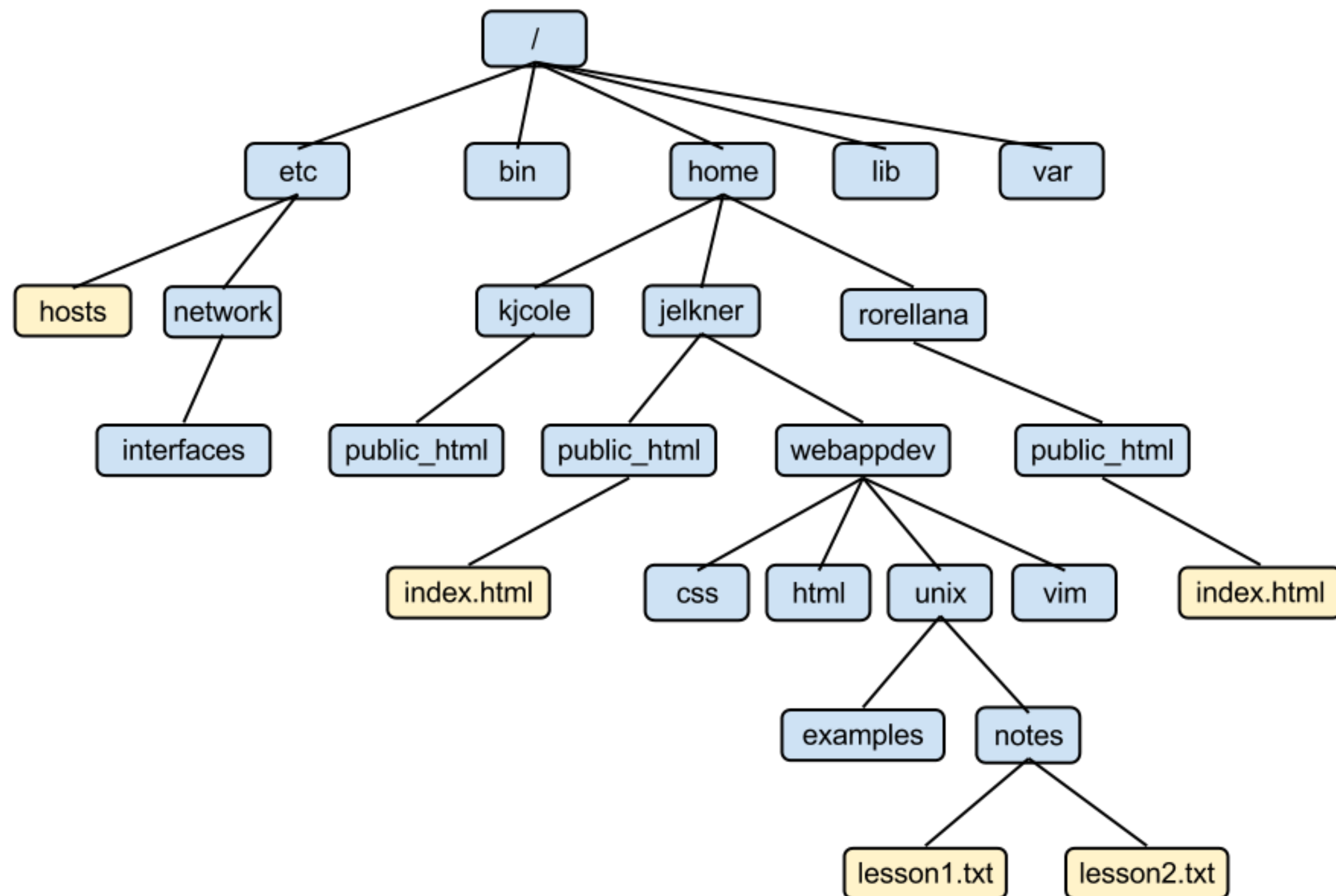
Python for Astronomers

Week 1- Basic Python

UNIX

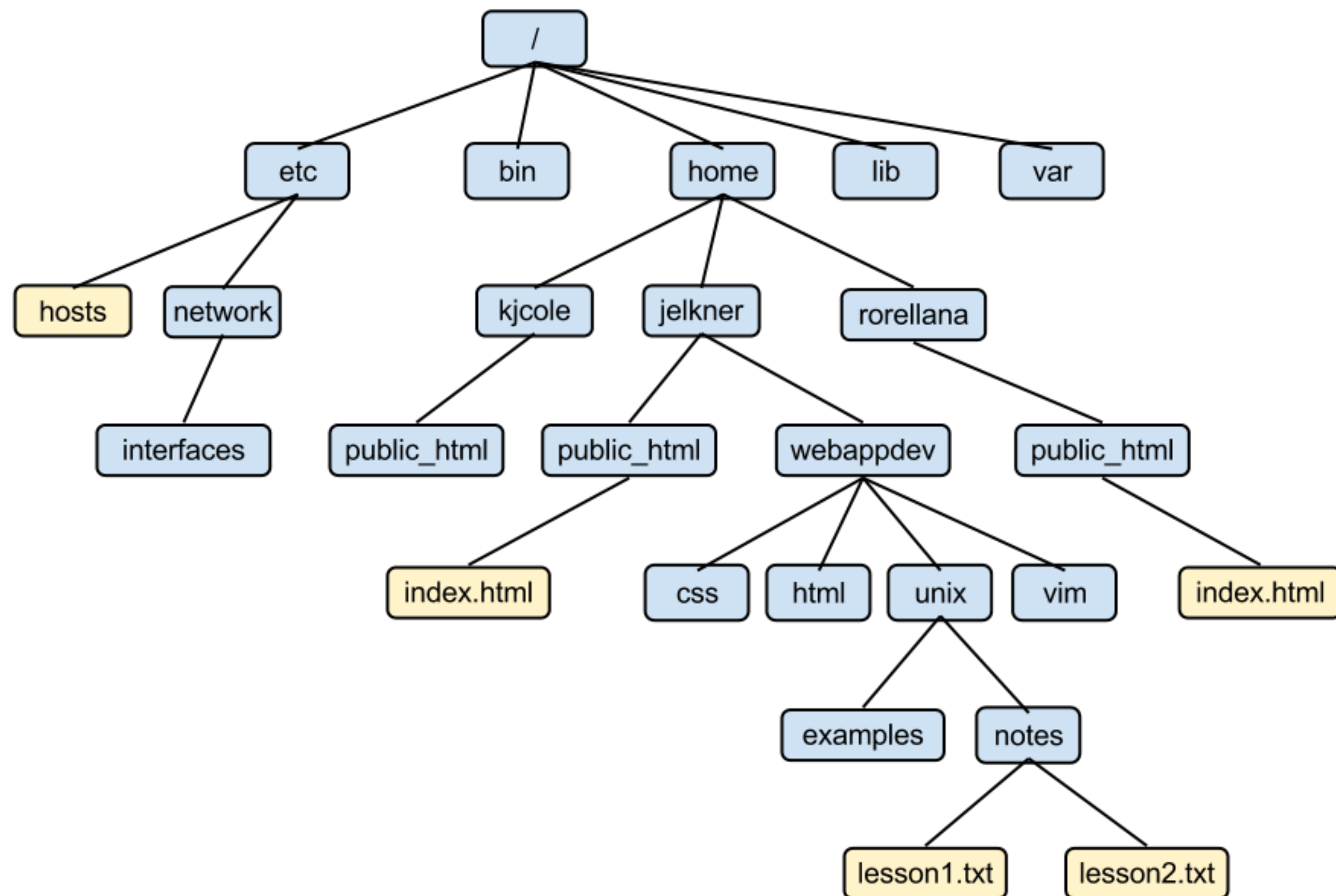
- UNIX is the operating system of Linux (and in fact Mac). It comprises primarily of a certain type of file-system which you can interact with via the terminal (command line).
- Navigating a file-system via command line take some getting used to, but ultimately it provides much higher efficiency than a GUI (where you would open multiple folder windows)

An Example Directory Tree



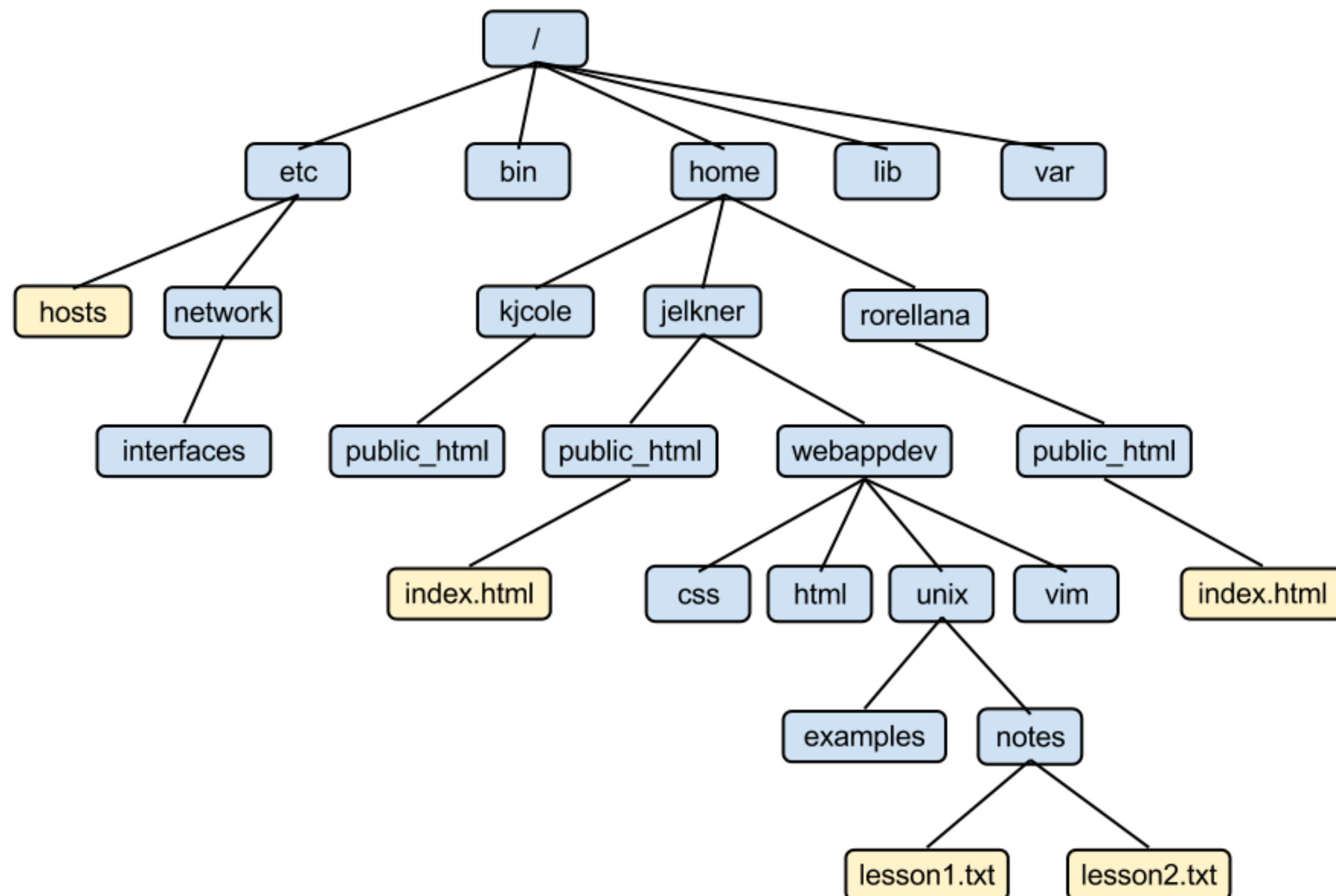
- A server/computer running over UNIX has a directory tree like this, with main directories, and various subdirectories within.
- To access any specific directory, you can use the UNIX command “cd” and the full path location of your directory
- i.e: `cd /home/jelkner/webappdev/unix/notes`

The cd Command



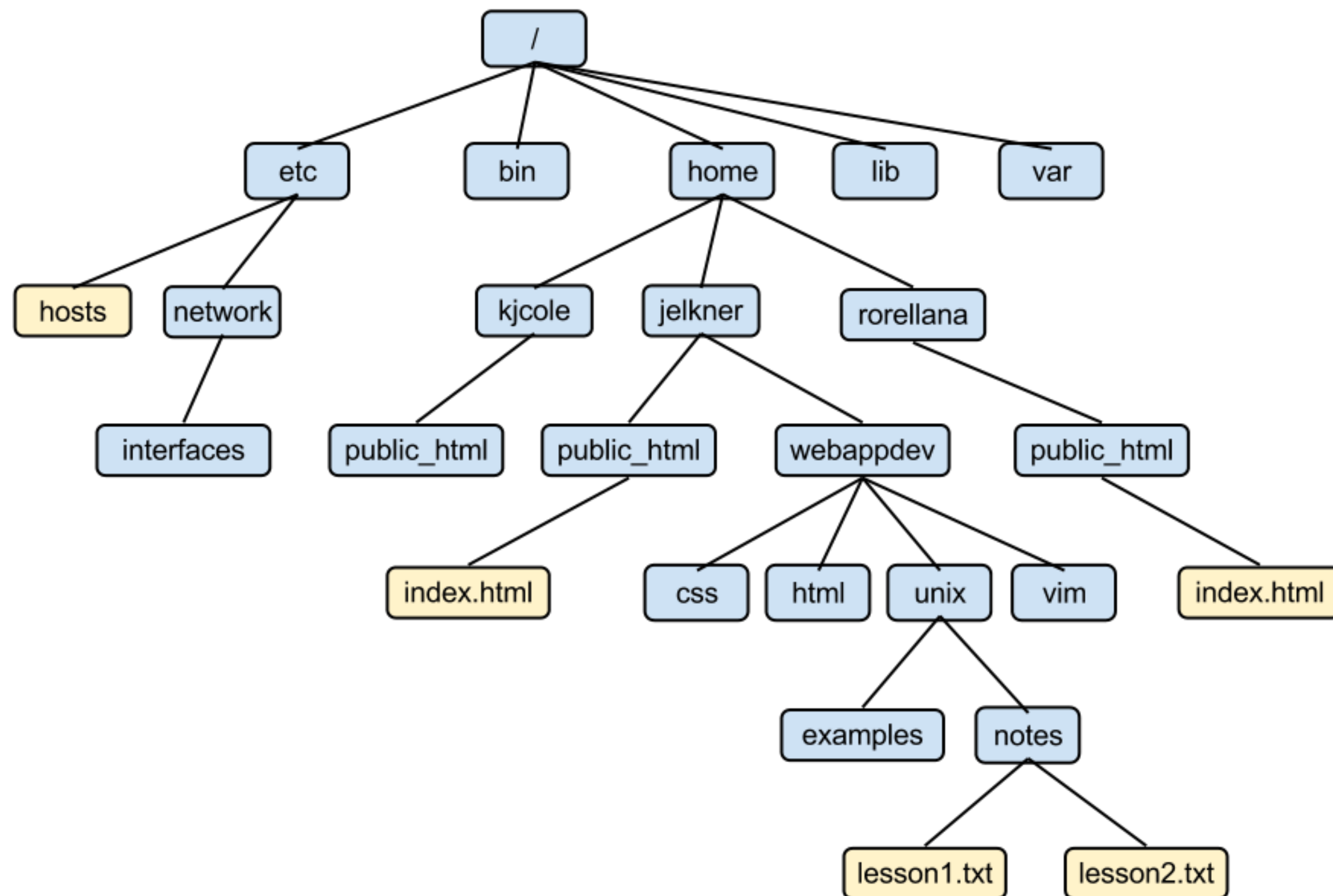
- Additionally, you can use the command `cd` without any arguments to return to the home directory of your user.
- This is different than `/home`. Each person with access to a server/ computer has an account name, and YOUR home directory is typically the one in the filesystem with your name on it- so if i were `kjcole`, typing `cd` <enter> would take me to `/home/kjcole`.

Some shortcuts with cd



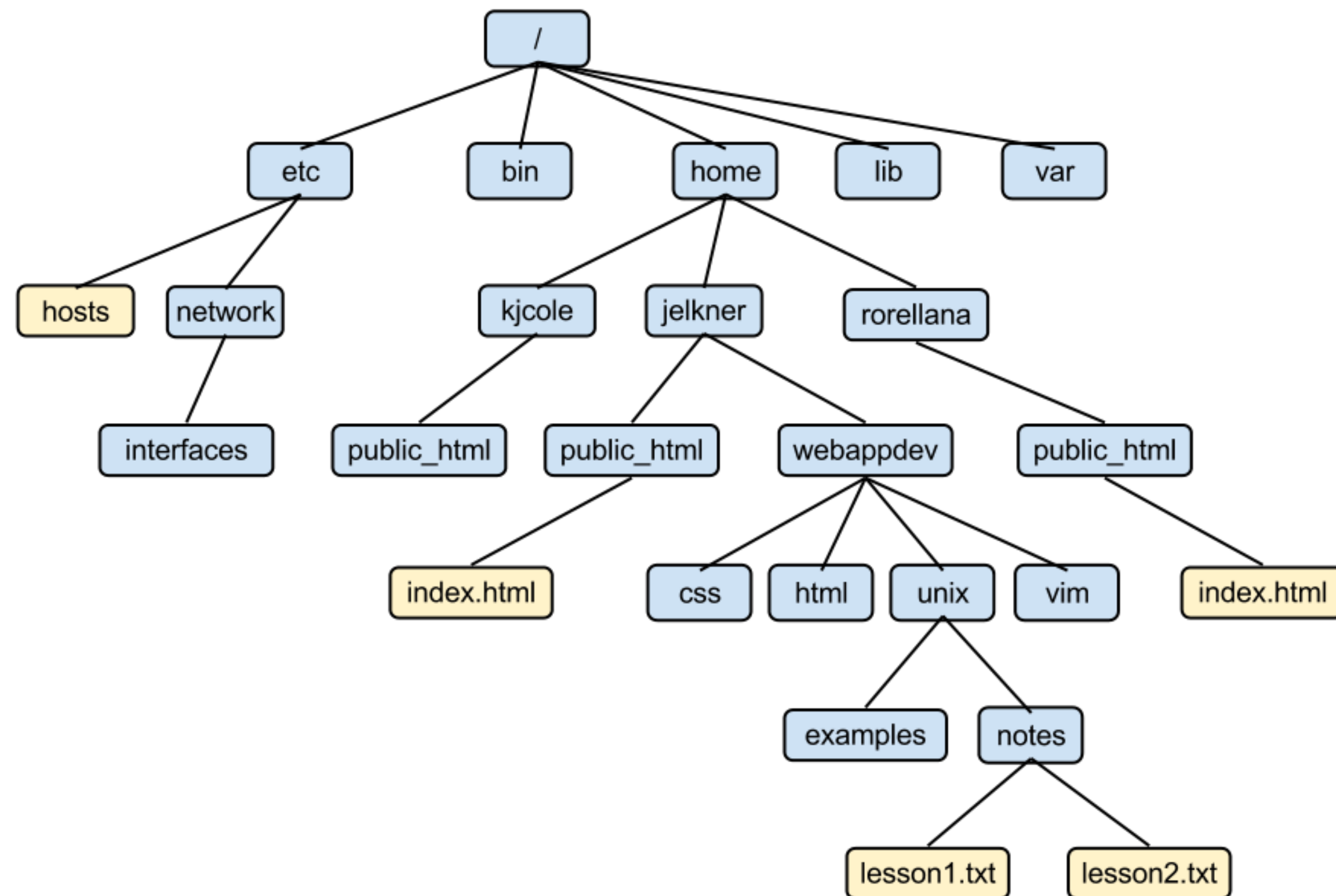
- To move only one/two levels in the tree there are some shortcuts
- >> `cd ..` will take you out one directory (i.e., from `unix` to `webappdev`)
- If you are in `/unix`, typing `cd notes` will take you into notes*
- Aside: You can use “`pwd`” to see exactly where you are in a tree at any time.

Some shortcuts with cd



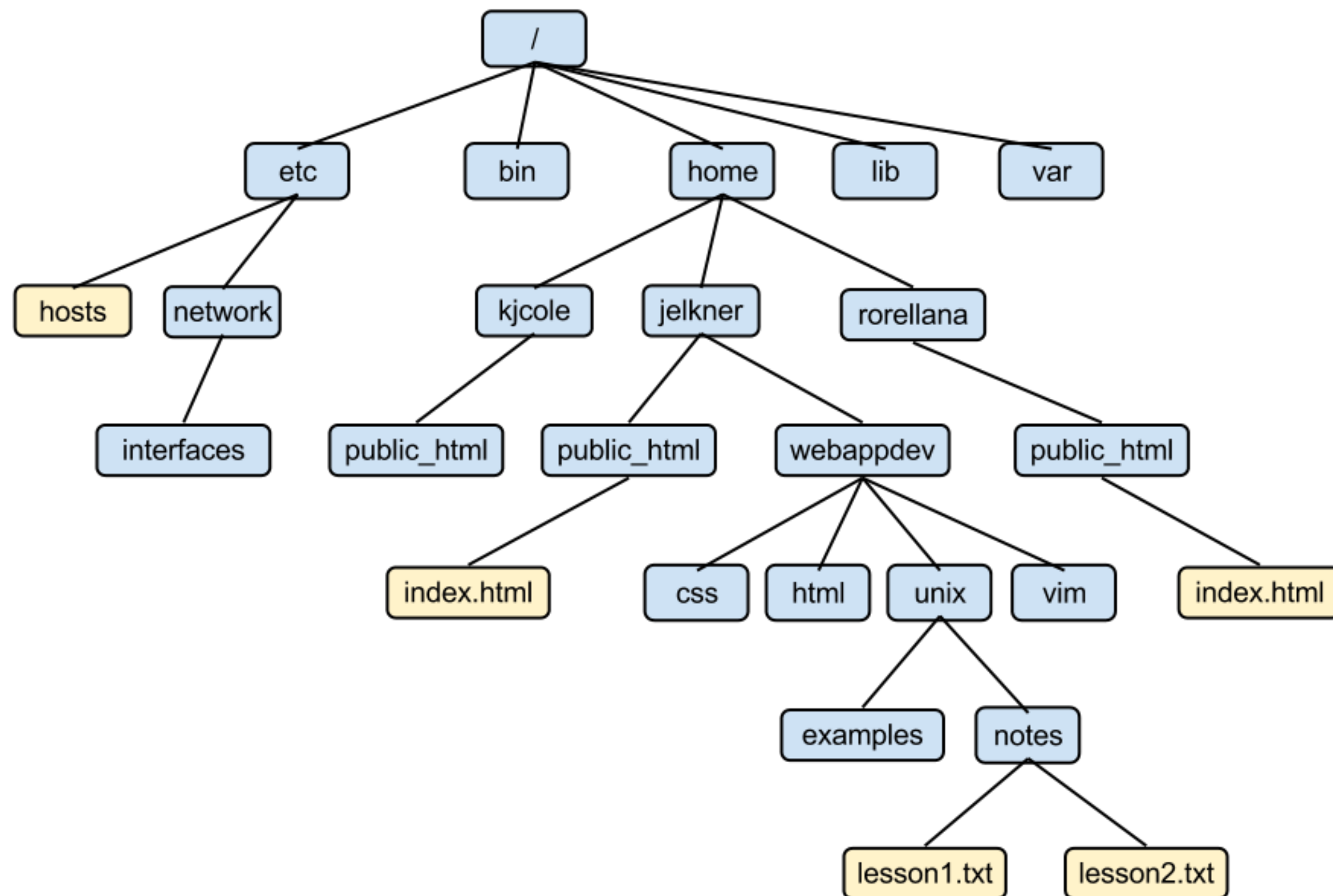
- You can string these shortcuts together from your current directory.
- If you were in unix, `cd ../../` would take you to `/jelkner`
- If you were in `/webappdev`, `cd unix/notes` would take you to `notes`.

Moving and Copying



- Use the 'mv' command to move a file. The syntax is (if you were in /notes):
- `>>mv lesson1.txt /home/kjcole/public_html`
- i.e: mv filename new-name new location (in the example I chose not to rename it along the way)
- mv also used to rename files
- cp uses the same syntax.

Deleting files and Directories



- To delete a file, use the “rm” command: `rm filename`
- To delete an empty directory you can use `rmdir directory name`
- The recursive lets you delete a directory and all files within:
- `>>rm -r /home/jeljner/webappdev/unix/notes`

Other useful commands

- ls: list all files/folders in the current directory
- pwd: print the current directory
- * : wildcard (shown best with examples)
- ssh: allows you to log in to a remote server
- scp: allows you to copy files to/from a remote server

Python

- Python is a high level programming language that is useful particularly for data analysis and plotting
- Python is one of the easier languages to learn; being “high level” means there’s less archaic syntactical expressions for things-it reads closely to plain english
- Programming in python (or any language) comes down to a combination of data-types being saved into variables, control-flow/ conditionals and loops handling the data, and plotting the outputs (or saving them to/from files).

iPython interpreter

- We just demonstrated how python can be used as a calculator, performing basic arithmetic (+, -, /, *, ^).
- We saw that 1/2 produced a result of 0. This comes down to a question of data-types, so we should try to understand the different ones available to us

Datatypes

- Integers: Counting numbers (1,2,3,...). Math performed with integers has to produce an integer (hence the rounding error).
- Floats: Any number with a decimal (1., 2., 3.1456, ...). Floats take up more memory, but $1./2$ will produce the right response. If any one number in a calculation is a float, any other integers will be converted to floats.
- strings: Anything contained in quotes. Usually used to introduce words like 'python', and are often used in the syntax for loading data files.

Datatypes

- Lists: lists of any data-type, contained in brackets:
 - `>> list1 = [1,2,3.,[4,5,6], 'cat', 'dog', 8]`
- Arrays: a numpy datatype, like a list but where all values have to be numbers. Arrays can be easily mathematically manipulated- multiplication by an array will multiply every value in the array by that number element wise.
 - `>>array_1 = np.array([1., 5., 8.])`
- Booleans: either true or false.

Variables

- Variables allow us to store objects of the data-types mentioned for use in calculations- for example we already declared the variables list1 and array_1. Variables give us the capacity to work with a 1000 entry list as a single letter, without having to type out all the entries each time.
- To declare a variable, you pick a name for it, and set it equal to its value

Recasting

- Within certain parameters, you can recast a variable of one type into another, using the name of the datatype. For example, ints can be cast as floats, floats can be cast as ints, floats/ints can be cast as strings, strings containing only numbers can be cast as floats/ints, etc.
- This is useful if you want to change the datatype of a variable after you have declared it (even if you no longer know what its value is since it has been calculated upon).