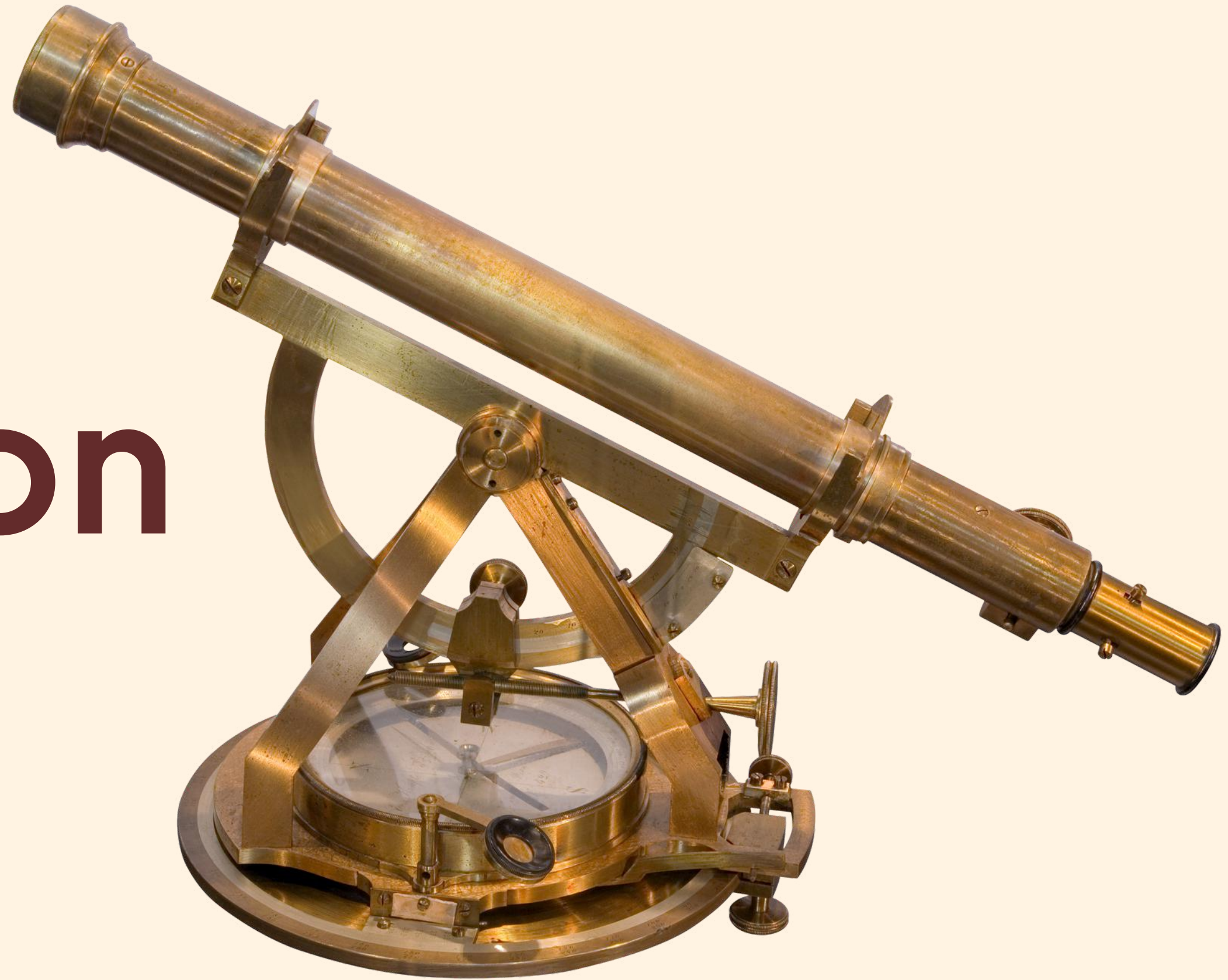


# Navigation



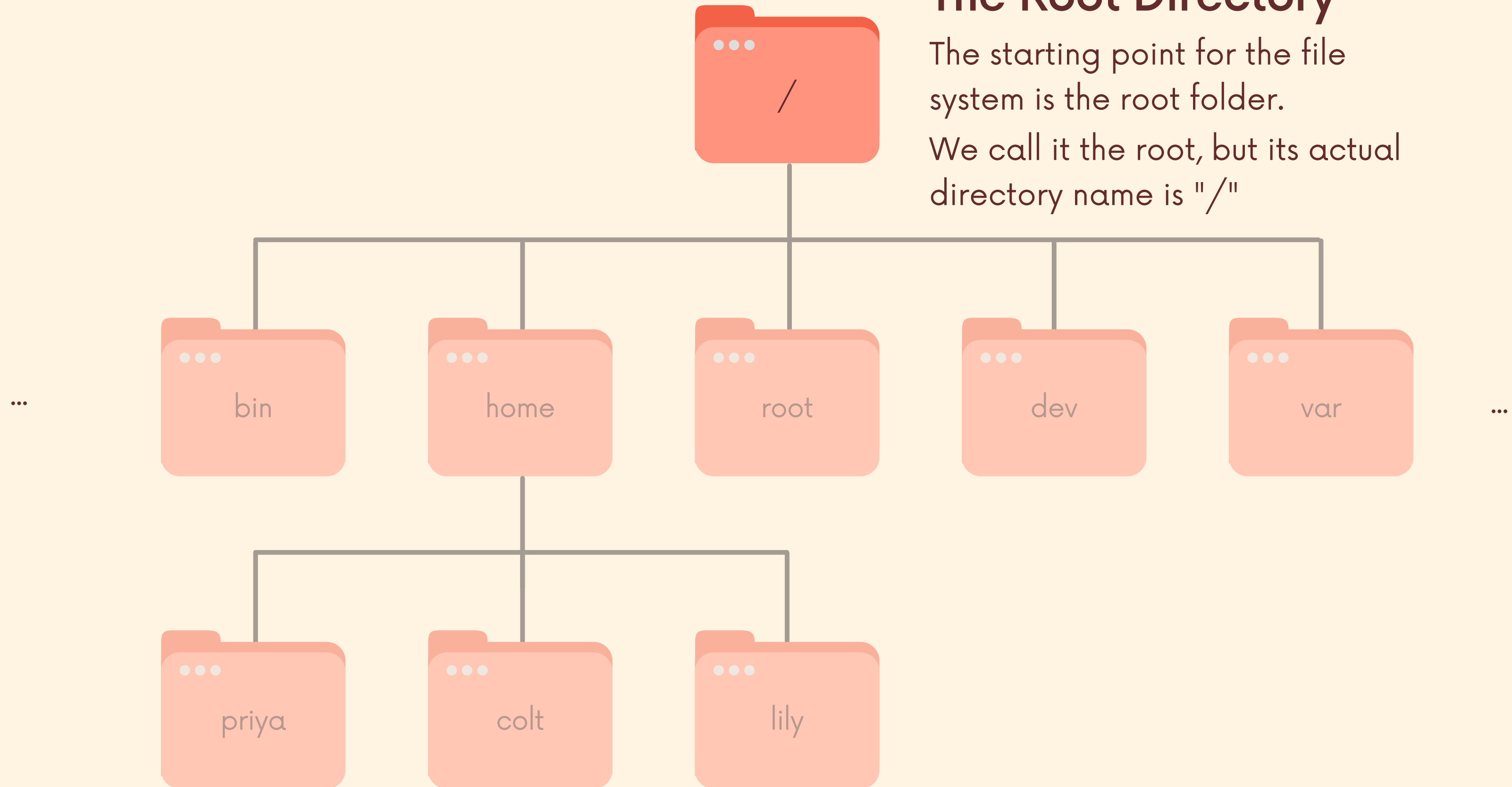




# The Root Directory

The starting point for the file system is the root folder.

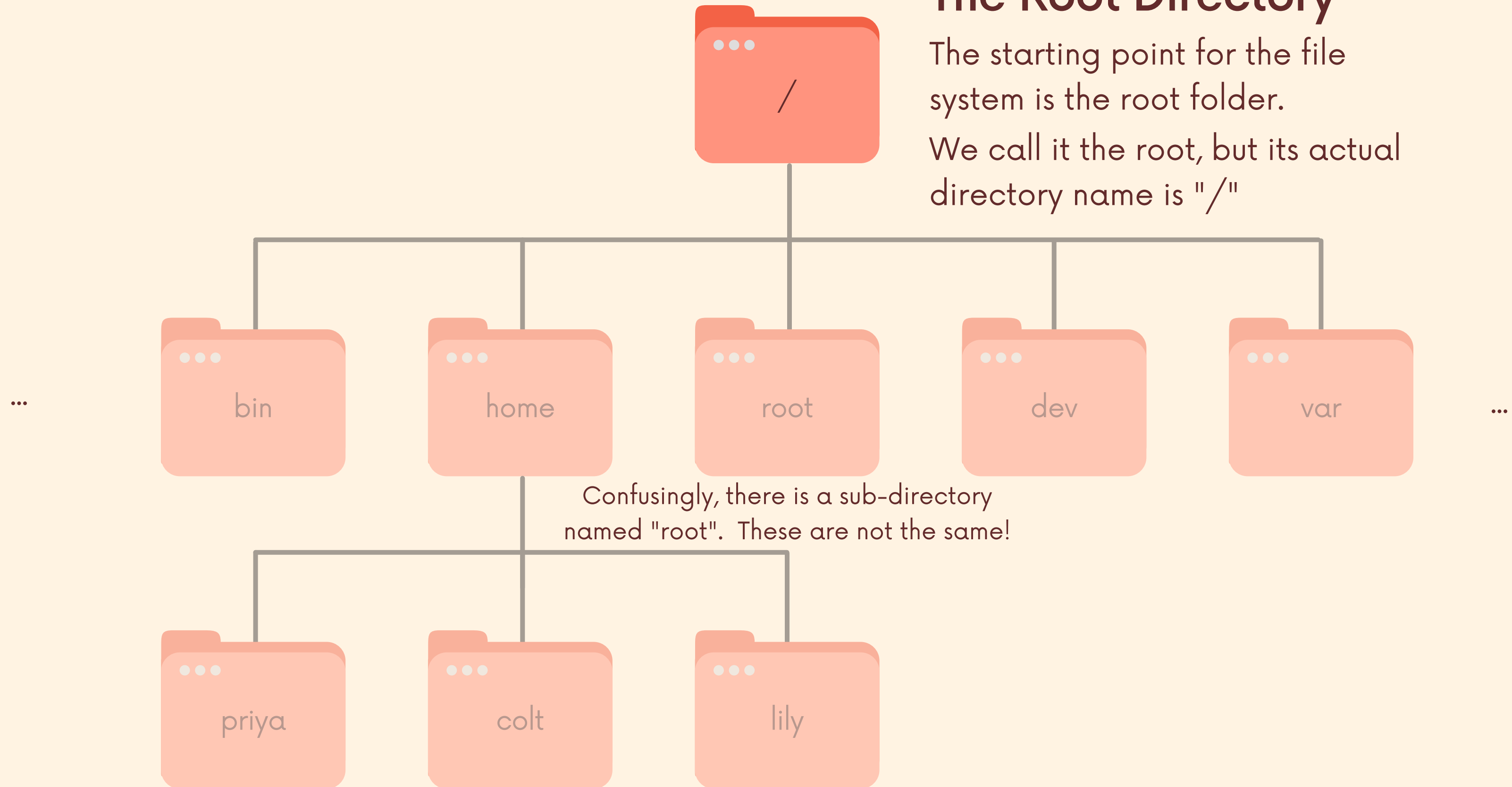
We call it the root, but its actual directory name is `"/"`

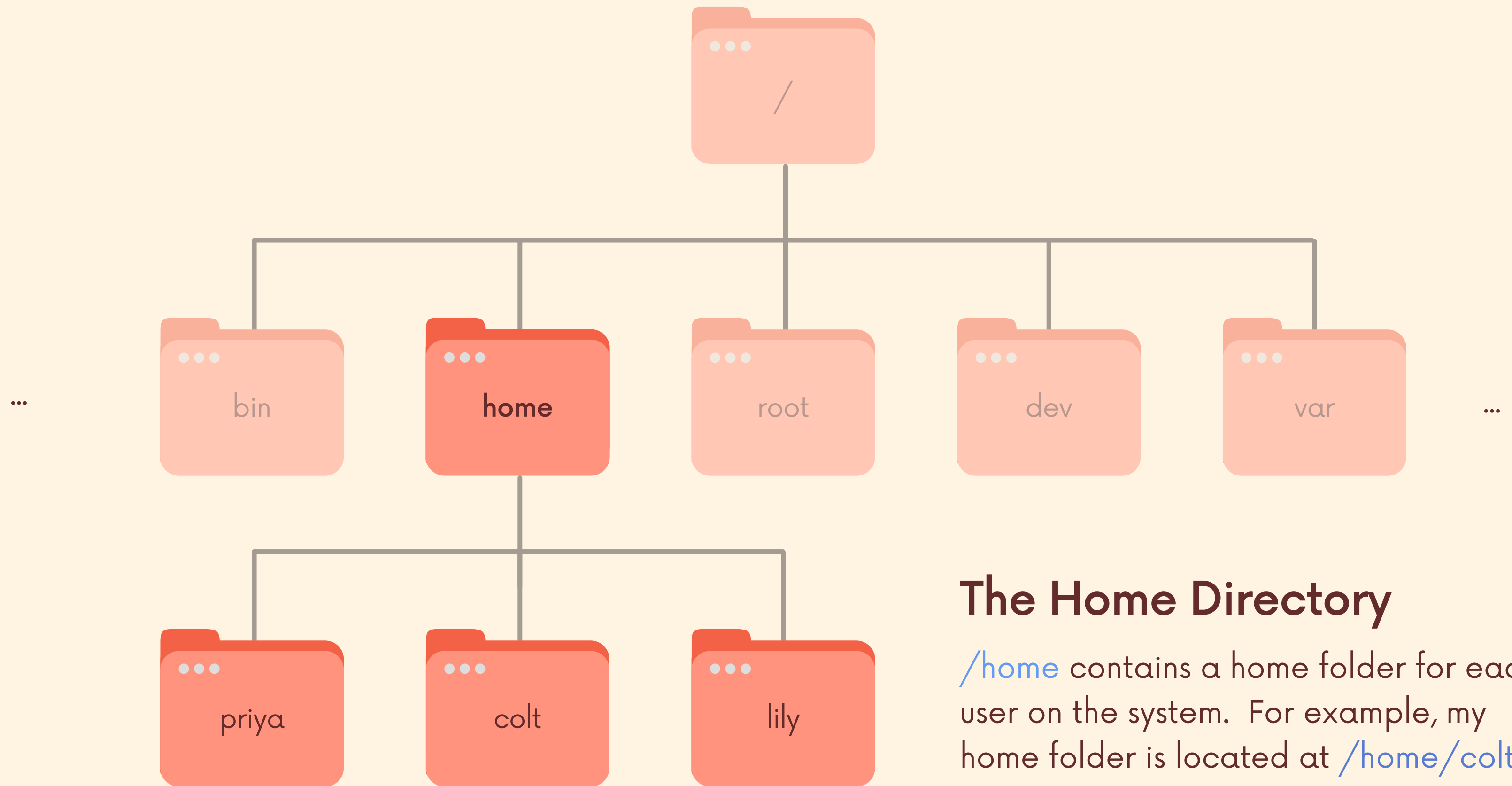


# The Root Directory

The starting point for the file system is the root folder.

We call it the root, but its actual directory name is `"/"`





/ - root

~ - home

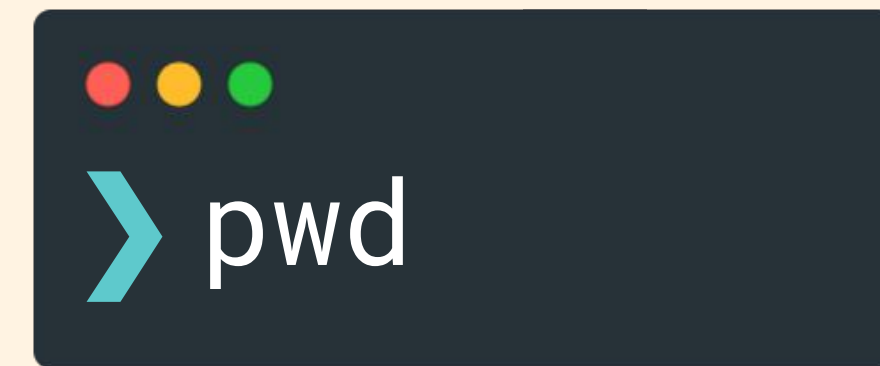


# pwd

The **print working directory** command is super simple but very useful. Think of it as a "where am I" command.

It will print the path of your current working directory, starting from the root /

For example, if I were on my desktop and I ran pwd, I would see `/home/colt/Desktop`



"where am I?"







# ls

The list command will list the contents of a directory.

When used with no options or arguments, it prints a list of the files and folders located in the current directory.

We can also list the contents of a specific directory using **ls path**. For example **ls /bin** will print the contents of the /bin directory

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. A light blue prompt character is followed by the command 'ls' in white text.

list contents of current directory

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. A light blue prompt character is followed by the command 'ls /home/colt' in white text.

list contents of /home/colt







# ls options

The ls command accepts a ton of options.  
Two of the more commonly used are `-l` and `-a`

A dark blue terminal window with three colored window control buttons (red, yellow, green) in the top left corner.

```
> ls -l
```

`-l` (lowercase L) prints in long listing format. It shows far more information about each file/folder.

A dark blue terminal window with three colored window control buttons (red, yellow, green) in the top left corner.

```
> ls -a
```

the `-a` option will also list any hidden files that begin with `.` These are normally not listed.

A dark blue terminal window with three colored window control buttons (red, yellow, green) in the top left corner.

```
> ls -la
```

We can combine options! This example prints detailed information for all files, including hidden files.





# cd

The **cd** command is used to change the current working directory, "moving" into another directory.

For example... **cd chickens** would change into the chickens directory (assuming it exists)

**cd /home/colt** would take me my home directory

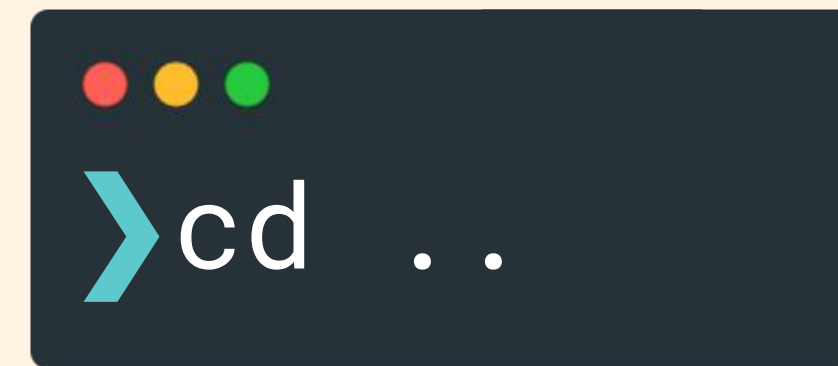




# backing up

In Unix-like operating systems, a single dot (.) is used to represent the current directory. Two dots (..) represent the parent directory.

So we can use `cd ..` to move up one level, from our current directory into the parent directory.



"back up" one level  
into the parent folder



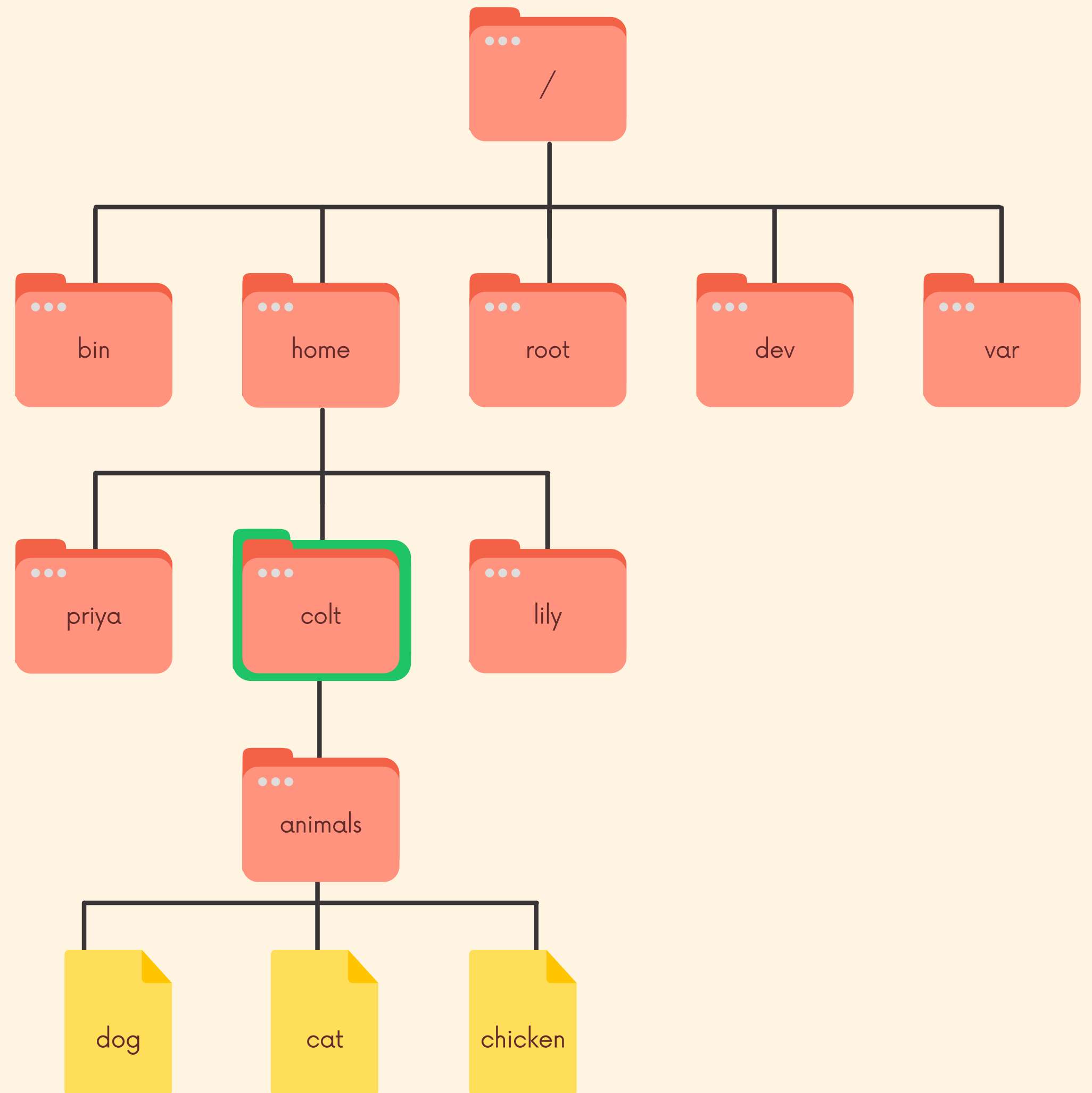
# Relative Paths

When providing paths to commands like `cd` or `ls`, we have the option of using relative or absolute paths.

Relative paths are paths that specify a directory/file relative to the current directory.

For example, if our current directory is `/home/colt` and we want to `cd` into `animals`, we can simply run `cd animals`.

However, `cd animals` does NOT work if we are located in another directory like `/bin`. The relative path from `/bin` is `../home/colt/animals`



# Absolute Paths

Absolute paths are paths that start from the root directory (they start with a `/`)

The absolute path to the animals directory is `/home/colt/animals`. We can use absolute paths to specify a location no matter our current location.

For example, from the `/bin` directory I could use `cd /home/colt/animals` to change into the animals directory.

