Linux Path part 2

First, find where `ls` is, and see if there are any other files named `ls`.

```
I need to list files in my home/
To check on project logos
But what I see with ls there,
Are quotes from desert hobos...

which piece of my command does fail?
I surely cannot find it.
Make straight my path and locate that-
I'll praise your skill and sharp wit!

Get a listing (ls) of your current directory.
elf@edcede7a980d:~$ find / -name ls 2>/dev/null
/usr/local/bin/ls
/bin/ls
elf@edcede7a980d:~$
```

The find command is handy for this. We told it to start looking at the root of the file system (/) and to look for files named ls (`-name ls`). The `2>/dev/null` says to send any error messages (`2>`) to the trash (`/dev/null`) so we don't have to look at them. When you use find from the root (/) you will get many permission errors.

```
find / -name ls 2>/dev/null
```

There are two files named `ls`. Which one is executed when you type `ls`? One way to discover that is to look at the PATH variable using

```
echo $PATH.
```

```
elf@edcede7a980d:~$ find / -name ls 2>/dev/null
/usr/local/bin/ls
/bin/ls
elf@edcede7a980d:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
elf@edcede7a980d:~$
```

This tells us that the `ls` that executes is the one in `/usr/local/bin` because it is the first on in the PATH.

```
elf@edcede7a980d:~$ ls
This isn't the ls you're looking for
elf@edcede7a980d:~$ /usr/local/bin/ls
This isn't the ls you're looking for
elf@edcede7a980d:~$
```

That is not the `ls` we are looking for.

Another way to tell which `ls` executes is to use the `which` command.

```
elf@edcede7a980d:~$ which ls
/usr/local/bin/ls
elf@edcede7a980d:~$
```

When you execute the other `ls`, `/bin/ls`, you win the challenge.