

# find

**H**opefully you are discovering the various ways you sabotage yourself even before you begin to work. Maybe it's not that dramatic, but you should at least be identifying things you can improve in your environment that are making it difficult for you to start working. These little exercises are a good way for you to focus on the beginning since they are not that important and fit into a time scale that you can analyze. If these projects were hours long, you'd get bored reviewing what you've done and making improvements. A short 45-minute project is something you can take notes about (or record) and review very quickly.

This is a pattern I use throughout my studies. I'll identify something that I need to improve on, such as how I get started, or how I handle a tool. Then I'll devise an exercise that simply focuses on that. When I was learning to paint I struggled with going outside to paint trees. I sat down and looked at the problems, and the first thing I identified was I simply dragged too much stuff with me. I also kept all my things in random places around my apartment. I purchased a specific bag just for my painting supplies and kept that bag ready to go. When I wanted to paint outside I grabbed this bag and walked to one of a few places, rather than planning elaborate painting hikes. I practiced just grabbing my bag, walking to one of two places, setting up, doing a painting, then walking home until the process was smooth as silk. After that I watched Bob Ross to figure out how to paint trees because that guy can crank out some trees.

This is what you should be doing. One place many people waste time and effort is in their work area. Do you have a dedicated place to work that never changes? I ditched my laptop and now just use a desktop machine so that I can have a consistent place to do my work. This also saved my back and neck from hauling around that chunk of metal and gave me a bigger screen to work with, all improving my ability to work. In this exercise, I want you to focus on your work area and make sure that it's ready to go before you begin:

1. Do you have enough light? Do you need less light?
2. How's your chair? Do you need a better keyboard?
3. What other tools are getting in the way? Are you trying to do Unix-like things on a Windows machine? Trying to do Mac things on Linux? Don't go buy a new computer, but consider it for your next big purchase if you find there's just too much friction for what you want to do.
4. How's your desk? Do you even have one? Do you hack in cafés all day with terrible chairs and too much coffee?
5. How about music? Do you listen to music with words? I find that if I listen to music without words it's easier for me to focus on the voice in my head that helps me write or code.
6. Do you work in an open plan office and your coworkers are annoying? Go buy yourself a pair of big over-the-ear headphones. When you wear them it's obvious you're not paying attention,

so people will leave you alone and they'll feel it's less rude than if you're plugged in and they can't see. This will also block out distractions and help you focus.

Spend this exercise thinking about topics like this and trying to simplify and enhance your environment. One thing, though: Don't go buying crazy contraptions and spending tons of money. Just identify problems, and then try to find ways to fix them.

## Exercise Challenge

In this challenge you are implementing a basic version of the `find` tool for finding files. You run `find` like this:

```
find . -name "*.txt" -print
```

That will search the current directory for every file ending in `.txt` and print it out. `find` has an insane number of command line arguments, so you are not expected to implement them all in one 45-minute session. The general format of `find` is the following:

1. The directory to start searching in: `.` or `/usr/local/`
2. A filter argument like `-name` or `-type d` (files of type directory)
3. An action to do with each found file: `-print`

You can do useful things like execute a command on every found file. If you want to delete every Ruby file in your home directory you can do this:

```
find . -name "*.rb" -exec rm {} \;
```

Please don't run this without realizing it will delete all the files that end in `.rb`. The `-exec` argument takes a command, replaces any instance of `{}` with the name of the file, and then stops reading the command when it hits a `;` (semicolon). We use `\;` in the preceding command because `bash` and many other shells use `;` as part of their language, so we have to escape it.

This exercise will really test your ability to use either `argparse` or `sys.argv`. I recommend you run `man find` to get a list of arguments, and then try using `find` to figure out exactly what arguments you'll implement. You only have 45 minutes, so you probably can't get too many, but `-name` and `-type` are both essential as well as `-print` and `-exec`. The `-exec` argument will be a challenge though, so save it for last.

When you implement this, try to find libraries that can do the work for you. You'll definitely want to look at the `subprocess` module and also the `glob` module. You will definitely want to look at `os` more carefully as well.

## Study Drills

1. How much of `find` did you get implemented?
2. What are the libraries you found to improve this implementation?
3. Did you count finding libraries as part of your 45 minutes? You could say that research before you start hacking doesn't count, and I'd be alright with that. If you want the extra challenge, then include your research in the 45 minutes.

## Further Study

How much of `find` can you implement in more 45 minute hacks? Maybe make this your hacking warmup challenge for the next week to see what you can get done. Remember that you should be trying to slap together the best ugly hack you can. Don't worry, I won't tell the Agile people you are just having fun.