

# File system exploration

- It's not unusual for a bash script to search a directory tree
- To do so, it needs to be able to check that a directory is actually accessible, look at its contents, distinguish between files and directories, call itself recursively, etc
- We'll look at a simple exploration function, that expects a directory name as its parameter then searches that directory, looking for git repositories

# Basic design

- First the function checks it was actually passed a parameter
- Then it checks that actually is the name of an accessible directory (it exists, is readable, is executable)
- Then it checks if the directory is the root of a git repository, printing the directory name if so
- If the directory isn't the root of a repo, the function calls itself on each subdirectory

# Code part 1: error checks

```
function search() {  
    # bail out if they didn't pass the right number of parameters  
    if [ $# -ne 1 ] ; then  
        return 0  
    end  
  
    # get the parameter, bail out if it's an empty string  
    local dir=$1  
    if [ -z $dir ] ; then  
        return 0  
    end
```

# Code part 2: is it a git repo?

```
elif [ -d $dir ] ; then # yes, it is a directory
    # if it's a git repo, it must have a .git subdirectory, let's check
    local gitname="$dir/.git"
    if [ -d $gitname ] ; then
        echo "git repository found: $dir"
    else # not a repo, so need to check each subdirectory
        for file in "$dir"/*; do
            if [ -d $file ] ; then
                search $file
            fi
        done
    fi
fi
}
```