Files & Directories
Overview

We’ve covered how to interact with Files for both input and output. But how are files stored/organized on our computer?
Learning Objectives

- Be able to navigate directories and use them to organize files
- Describe how a computer may search through a directory
- Gain some understanding for the terminal, and what it does
Poll:

Do you know what a “folder” is on a computer?
Poll:

Do you know what a “directory” is on a computer?
**Definitions pt. 1**

What is a file:

- A “resource” on a computer that holds data.
  - This data is held even when the computer is turned off.
- Files have names to identify them e.g. `Hello.txt`
- Files can be opened, read, written to, saved, deleted, etc.
- A file can store image data, programs, text, etc.
  - Anything that can be represented by a sequence of bits
Definitions pt.2

What is a Directory (Folder):

- A directory is a special type of file that contains a list of other files (and directories)
- A directory is still named
- For most cases, we can use the word Directory and Folder interchangeably
Is README.md a regular file or is it a directory?
Organizing Files and Directories

Files on a computer are structured as a Hierarchical File System built of directories can contain other directories.

- **Subdirectory** is used to describe a directory contained in another
- **Parent** and **Child** are often used to describe the relationship between a subdirectory and the directory it is in.
- a few directories are the “overall root” or “overall parent”
Poll:

Can a file system have two files that are the same name?
**Working Directory**

You can have two files that are the same name, but they must be in *different directories*.

To help with managing files when in the terminal (or file explorer), there is a **working directory**

- Working Directory is where we start when searching for files
- “The directory we are currently in”
  - `pwd` prints this out from a terminal
  - on Finder or File Explorer, it's listed at the top of the window.

Demo: navigating files & duplicate files
**Paths**

To specify a specific file, we have the concept of a path. A **path** describes the location of a file inside of a file system structure.

- "Starting from a directory, how would I get to your file through other directories?"
- Relative path: starting from the current directory, e.g. `examples/Hi.java`
- Absolute path: starting from the absolute directory, e.g. `/home1/s/sharry/directory_demo/Hi.java`

A path comprises a list of directories separated by either a `/` or a `\` character, depending on operating system (Windows, Mac, Linux).
Java: File

Java has a File object we can use for navigating Directories

Construct with `File f = new File(path);`

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listFiles()</td>
<td>Returns a File[] containing all entries in the directory</td>
</tr>
<tr>
<td>getName()</td>
<td>Returns the name of the file as a String</td>
</tr>
<tr>
<td>isFile()</td>
<td>Returns True if it is a normal file</td>
</tr>
<tr>
<td>isDirectory()</td>
<td>Returns True if it is a directory</td>
</tr>
</tbody>
</table>
**Demo: Ls.java**

Given a specified directory name, list all of the contents of that directory. This mirrors the `ls` command in the terminal.
Demo: Search.java

Given a specified directory name, and a word print all of the files in that directory (or any subdirectories) that contain that word.

```
grep -rlw "path/to/directory/" -e "search string"
```