

Linux Terminal: Where Am I? A Folder Treasure Hunt

Ages 11-18 · about 45 minutes, in one sitting

Your child opens a real Linux-style terminal with a treasure hunt already hidden in its folders. Each clue says where the next one is and how to get there by typing, down through folders, back up, from the very top, and home to a hidden file. Along the way they learn the question a command line keeps asking, where am I, and the commands that answer it.

What you need

- A computer or Chromebook with a keyboard and a current browser.
- The lesson link: mytekdev.com/try/terminal?lesson=where-am-i&ref=fam-terminal-1. No account, no email and nothing to install. It opens a guest terminal, and nothing is kept after the tab is closed.
- Optional: the activity sheet at the end of this plan, printed, and a pencil.

Before you start: try it once yourself

Do the hunt yourself once before you start. It takes about ten minutes, and you will know exactly what your child will see.

1. **Open the lesson.** Go to mytekdev.com/try/terminal?lesson=where-am-i&ref=fam-terminal-1. The terminal prints Lesson ready: Where Am I?
2. **Look around.** Type `ls`: the hunt's games/ and school/ sit beside the folders every account has (apps/, projects/, public_html/, shared/), with start-here.txt. Type `cat start-here.txt` and read it.
3. **Clue 1.** `cd school/science/labs`, then `pwd`: it prints /home/ and a guest name, then /school/science/labs. `cat clue.txt`.
4. **Clue 2.** `cd ../../art`, then `cat clue.txt`.
5. **Clue 3.** Type `cd /home/`, your own guest name from pwd, then `/games/saves`. `cat clue.txt`.
6. **The treasure.** `cd ~`, then `ls -a`: .treasure.txt appears. `cat .treasure.txt` gives the code word LANTERN.
7. **The map.** `apt install tree`, wait a few seconds, then `tree`. The last line says 8 directories, 8 files.

```

guest21@mytekos:~$ tree
guest21
├── games/
│   ├── maps/
│   │   └── level1.txt
│   └── saves/
│       └── clue.txt
├── school/
│   ├── art/
│   │   └── clue.txt
│   ├── math/
│   │   └── homework.txt
│   └── science/
│       ├── labs/
│       │   └── clue.txt
│       └── notes.txt
├── .treasure.txt
└── start-here.txt

8 directories, 8 files
guest21@mytekos:~$

```

What the tree shows at the end.

If you see something else: a typo prints No such file or directory, so check the spelling and the slashes. A command typed while tree is still installing can lose letters; wait for the prompt. If the files get into a mess, typing lesson start where-am-i puts them back. Reloading the page starts a new, empty guest terminal; the lesson link sets the files up again.

The ideas, in plain words

A terminal is a way to use a computer by typing commands instead of clicking. Each command is a word, sometimes followed by options that start with a dash and by the names of files. The computer runs it and prints an answer, or prints nothing when it worked. The terminal in this lesson is a real Linux-style shell running on our server; each visitor gets a fresh guest one, and nothing they type can reach anyone else's files.

Files live in a tree of folders. The top of the tree is / (called root). Each guest has a home folder, /home/ and their guest name, which ~ stands for. A path is the directions down the tree: school/science/labs means labs, inside science, inside school. A relative path starts from the folder you are in; an absolute path starts at / and works from anywhere. .. means the folder above. A file whose name starts with a dot is hidden: ls skips it and ls -a shows it.

YOUR CHILD MIGHT ASK	A GOOD ANSWER
Why did ls not show the treasure?	Its name starts with a dot, which hides it from a plain ls. ls -a shows everything.
What is the difference between / and ~?	/ is the top of the whole tree. ~ is your own home folder, partway down.
Why do we need cd ../../art and not cd art?	art is not inside labs. .. goes up one folder at a time: labs, then science, then school, which has art in it.
Is this how real computers work?	Yes. Every Mac, Linux and Windows computer stores files in a tree like this; the folders you click through are the same paths.

Do it together

1. Start with a question: your computer has thousands of files, so when you double-click a folder, how does the computer know which one you mean? Then open mytekdev.com/try/terminal?lesson=where-am-i&ref=fam-terminal-1 together.

You'll know it worked when: The terminal prints Lesson ready: Where Am I?. The hunt is already set up in the folders.

2. Type `pwd` and press Enter: it prints the folder you are in, your home folder. Then type `ls`: it lists what is in this folder, with a / after each folder's name.

You'll know it worked when: `pwd` prints /home/ and a guest name ending in a number, which is different every time. `ls` shows games/ and school/ beside apps/, projects/, public_html/ and shared/, and a file called start-here.txt.

3. Type `cat start-here.txt` and read it together. Learn the three moves: `cd` and a folder name goes down into it, `cd ..` goes up one, and `cd ~` (or just `cd`) goes home from anywhere. A command that worked often prints nothing at all.

You'll know it worked when: Your child can say the three moves, or point to them at the top of the activity sheet.

4. Clue 1: type `cd school/science/labs`, then `pwd`, then `cat clue.txt`. Write down what `pwd` says. That path goes three folders down from home: labs, inside science, inside school.

If you get stuck: No such file or directory after a `cd` means a typo or a missing slash. `ls` shows the folder names exactly as they are spelled.

You'll know it worked when: `pwd` ends in /school/science/labs, and the clue is on the screen.

5. The clue on the screen says to go up two folders and into art in one step: `cd ../../art`, then `pwd` and `cat clue.txt`. Ask why it takes two .. and not one.

You'll know it worked when: Your child explains that art is not inside labs: each .. goes up one folder, labs to science, science to school, and school has art in it.

6. The next clue needs an absolute path, the whole path from the top: type `cd /home/`, then the guest name from `pwd`, then `/games/saves`, all as one command. Then `cat clue.txt`.

You'll know it worked when: `pwd` says /home/, the guest name, then /games/saves. Your child can say the difference: an absolute path starts at / and works from anywhere.

7. The treasure is hidden at home. Type `cd ~`, then `ls`: nothing new. Then `ls -a`, and `cat` the file that appears.

If you get stuck: A plain `ls` never shows it: the file's name starts with a dot, which hides it, so only `ls -a` lists it.

You'll know it worked when: `ls -a` shows `.treasure.txt`, and `cat .treasure.txt` gives the code word LANTERN.

8. Draw the map: type `apt install tree`, wait for the prompt to come back, then type `tree`. Ask where the tree command came from: the terminal itself is system software, and apt added a small application on top of it.

If you get stuck: A command typed while tree is still installing can lose letters, so wait for the prompt. If the files get into a mess, typing `lesson start where-am-i` puts them back. Reloading the page starts a new, empty terminal; the lesson link sets the files up again.

You'll know it worked when: `tree` draws every folder of the hunt and ends with 8 directories, 8 files.

9. Talk it over: why does a computer keep files in a tree of folders instead of one long list?

You'll know it worked when: Your child names a reason or two: it is easier to find things, related files stay together, and two files with the same name can live in different folders.

10. The real question, without trying it first: you are in school/science/labs. What one command takes you to school/math? Then try it.

You'll know it worked when: Your child answers `cd ../../math`, reasoning up to school and back down, without a clue to copy. That means they can picture the tree.

Talk about it

Ask these while you work together. A child who has understood the lesson can answer them without help:

- Where are you right now? How do you know?
- Why does `cd ../../art` need two `..` and not one?
- What would happen if you typed `cd ../../art` from a different folder? Why does the clue in art ask for an absolute path?
- Why didn't plain `ls` show the treasure?
- Without trying it: you are in school/science/labs. What one command takes you to school/math?

The last one is the real test. A child who answers `cd ../../math`, reasoning up to school and back down, can picture the tree and use a relative path without a clue to copy.

For younger and older kids

Younger: For ages 9 and 10, do the first two clues side by side, then write the command for each remaining clue on paper and let your child type it and read the result.

Older: For older teens and adults, look at the special entries `.` and `..` that `ls -a` shows, compare `cd -` with `cd ~`, and talk about why scripts use absolute paths.

Keep going

- Without `cd`, read clue 2 from your home folder in one command: `cat school/art/clue.txt`. Why does that work?
- Go to `school/math` using only an absolute path, then back home using only `..`
- Type `ls /` to see the very top of the tree. What other folders are up there?

Bring it into other subjects

- Math: a folder tree is a tree diagram, like a probability tree or a family tree. Draw the hunt's tree by hand, then compare with tree.
- Social studies: a library shelves books by a path too (floor, section, shelf). Write directions to a book as a path.
- Language arts: write the instructions for a treasure hunt in your home as a set of relative directions.
- Computer science: the same terminal can make, move and delete files.

For your records

Ready to copy into an hour log, a portfolio or a transcript.

LESSON	Linux Terminal: Where Am I? A Folder Treasure Hunt
SUBJECTS	Computer science
TIME	45 minutes
WHAT WE DID	Used a Linux-style terminal to follow a treasure hunt through a tree of folders with <code>pwd</code> , <code>ls</code> , <code>cd</code> and <code>cat</code> . Navigated with relative paths, an absolute path and <code>..</code> , found a hidden file with <code>ls -a</code> , and installed a tool that drew the whole folder tree.
SKILLS PRACTICED	file systems as a tree of folders; relative and absolute paths; typing commands in a terminal; hidden files; installing a tool with <code>apt</code>

Activity sheet (optional)

Name: _____ Date: _____



Open the terminal: scan the QR code, or go to mytekdev.com/try/terminal?lesson=where-am-i

Essential question: How does a computer know where a file is?

pwd = where am I. ls = what is here (ls -a shows hidden files too). cd name = go into a folder. cd .. = up one. cd ~ = home. cat name = print a file.

STOP	COMMAND YOU TYPED TO GET THERE	PWD SAYS
Start (home)	(none)	
Clue 1		
Clue 2		
Clue 3		
Treasure		

The code word is: _____

1. **Read.** `cat start-here.txt`, then follow each clue exactly.

2. **Map it.** At the end: `apt install tree`, then `tree`.

1. What is the difference between a relative path and an absolute path? Give one of each from the hunt.

Quick check

Name: _____

1. You are in `/home/sam/school/art`. Where does `cd ..` take you? (A) `/home/sam`, (B) `/home/sam/school`, (C) `/`.

2. Which command shows hidden files? (A) `ls`, (B) `ls -a`, (C) `cat`.

3. Is `/home/sam/games` an absolute or a relative path? How can you tell, and why would you use it?

Answers

From the guest terminal with the hunt set up. Every guest's home folder is /home/ plus their guest username (for example /home/guest10), so the pwd column shows that guest's own name.

STOP	COMMAND	PWD
Clue 1	cd school/science/labs	/home/(username)/school/science/labs
Clue 2	cd ../../art	/home/(username)/school/art
Clue 3	cd /home/(username)/games/saves	/home/(username)/games/saves
Treasure	cd ~, ls -a, cat .treasure.txt	/home/(username)

The code word is LANTERN. tree then shows 8 directories and 8 files. Relative example: school/science/labs. Absolute example: /home/(username)/games/saves.

QUICK CHECK	ANSWER	WHY
1	B	.. is the folder one level up: art is inside school.
2	B	-a means all, including names that start with a dot.
3	Absolute	It starts with /, so it begins at the top and works from any folder.

Want the whole subject taught?

This is one lesson to do together. If you would rather have the whole subject taught, our Computer Science 1 and Information Technology 2 courses teach the command line as part of a full course, with an instructor, live and recorded classes, graded work and credit.

If your state or umbrella school asks, this lesson covers CSTA 2017 3A-DA-10, CSTA 2017 3A-CS-02, Common Core Literacy RST.6-8.3, Common Core Literacy RST.6-8.4.

Terms of use

This lesson was made by mytekDev. You may print and copy it for the children you teach at home, and change it for them. You may not sell it, share it with other families, or post it publicly.

The app this lesson uses is free and runs in the browser, with nothing to install and no account needed. It is at mytekdev.com.

© 2026 mytekDev. All rights reserved.