

mytekDev

Linux Terminal: Where Am I? A Folder Treasure Hunt

Grades 6-12 · 45 minutes

Computer science

```
guest21@mytekos:~$ tree
guest21
├── games/
│   ├── maps/
│   │   └── level11.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:~$
```

Slides, rubric and answer key

Print and go

No student accounts

A real command line

What's included

- A one-page overview: objectives, essential question, vocabulary, standards and assessment
- A ten-minute walkthrough to rehearse before class, and background in plain words for the teacher
- A 45-minute lesson sequence: warm-up, direct instruction, guided practice, independent practice, closure
- Slides for the board: a PowerPoint file that also opens in Google Slides
- A student worksheet with a QR code
- A 3-question exit ticket, two to a page, with an answer key
- A rubric: 3 criteria, 4 levels, scored out of 12
- Differentiation for English learners, students with IEPs or 504 plans, and students who need more support or more challenge
- Cross-curricular extensions, and notes for teaching it in grades 4-5 and beyond grade 12
- Standards: CSTA 2017 and Common Core literacy

Linux Terminal: Where Am I? A Folder Treasure Hunt

Grades 6-12 · 45 minutes · Computer science

The worksheet's link opens a terminal with a treasure hunt already laid out in its folders. Each clue file says where the next one is and exactly how to get there: a path down through three folders, then up two with `..`, then a full path from the very top, and finally home with `~` to a hidden file that a plain `ls` does not show. Along the way students learn the one question a command line keeps asking, where am I, and the commands that answer it. They finish by installing a tool that draws the whole map of where they have been.

At a glance

ESSENTIAL QUESTION

How does a computer know where a file is?

OBJECTIVES

Students will be able to:

- say where they are in a folder tree with `pwd`, and list what is there with `ls`
- move through folders with `cd`, using a relative path, `..` and `~`
- explain the difference between a relative path and an absolute path, and use each
- find hidden files with `ls -a`

VOCABULARY

terminal, folder (directory), path, relative path, absolute path, home (`~`), hidden file

STANDARDS

3A-DA-10, 3A-CS-02, RST.6-8.3, RST.6-8.4

TIME AND MATERIALS

45 minutes, grades 6-12.

- One computer or Chromebook per student or pair
- A projector
- The worksheet and exit ticket, one copy each per student
- The rubric, one copy per student if you grade the worksheet

PREP

Make copies of the worksheet and exit ticket. Before class, open mytekdev.com/try/terminal?lesson=where-am-i&ref=tpt-terminal-1 once on a student computer: it starts a guest terminal and sets up the hunt by itself. Check that your school's filter allows it. No account is needed.

ASSESSMENT

- Formative: the questions under What to look for, asked while students work, and the exit ticket (answer key included).
- Summative: the completed worksheet, with every stop of the hunt and the code word, scored with the rubric out of 12.

TECHNOLOGY AND PRIVACY

Runs in any browser, Chromebooks included, with nothing to install. Each student gets a guest terminal for the lesson, with no account and no email; nothing is kept after the tab is closed, and nothing a guest makes is ever public. Up to 40 students on one school network can have a guest terminal at once. If your school blocks new sites, ask IT to allow mytekdev.com.

Before you teach: do it once yourself

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

1. **Open the lesson.** Go to mytekdev.com/try/terminal?lesson=where-am-i&ref=tpt-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 every student's 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 a student's 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.

Background 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 student 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 student 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.

STUDENTS 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.

Lesson sequence

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
0 to 6	Warm-up Hook	Show the title, essential question and Today I can slides (about 1 minute). Ask: "Your computer has thousands of files. When you double-click a folder, how does the computer know which one you mean?" Take a few answers. Then students open the link or scan the QR code on the worksheet. A terminal opens and prints Lesson ready: Where Am I?. The hunt is already set up in their folders.	Answer the question, then open the terminal.
6 to 14	Direct instruction (I do) Model it	On the projector, type pwd: it prints the folder you are in, a path like /home/guest10, which is your home folder (yours will have its own number). Type ls: it lists what is in this folder, folders with a / after their name. Explain the tree: folders inside folders, and a path is the directions down the tree, names separated by /. Then type cat start-here.txt and read it together. Show the three moves: cd folder goes down into a folder, cd .. goes up one, cd ~ (or just cd) goes home from anywhere. Point out that typing a command and pressing Enter is all there is: the terminal answers, or says nothing when it worked. It's working when: pwd prints /home/ and a guest name ending in a number. Every student's number is different, and that is fine.	Type pwd, ls and cat start-here.txt, and copy the three moves into the worksheet.

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
14 to 22	Guided practice (We do) The first two clues	Together, students follow the first clue: <code>cd school/science/labs</code> (a relative path, three folders down from home), <code>pwd</code>, then <code>cat clue.txt</code>. They write the <code>pwd</code> on the worksheet. Clue 1 says to go up two folders and into <code>art</code> in one step: <code>cd ../../art</code>. Ask why two <code>..</code> and not one: <code>labs</code> is inside <code>science</code>, which is inside <code>school</code>, so up twice reaches <code>school</code>. <code>pwd</code> again, and <code>cat clue.txt</code> for clue 2. Watch for: No such file or directory after a <code>cd</code> means a typo or a missing slash. <code>ls</code> shows the folder names exactly as they are spelled. If they ask: "Why <code>cd ../../art</code> and not <code>cd art</code>?" <code>art</code> is not inside <code>labs</code>. Each <code>..</code> goes up one folder: <code>labs</code> to <code>science</code>, <code>science</code> to <code>school</code>, and <code>school</code> has <code>art</code> in it.	Follow clues 1 and 2, writing the <code>pwd</code> at each stop.
22 to 36	Independent practice (You do) Finish the hunt	Clue 2 asks for an absolute path: the whole path from the top, <code>cd /home/</code> then their own username then <code>/games/saves</code>. Students read their username from <code>pwd</code> (it is the name after <code>/home/</code>). An absolute path works from anywhere, which is the difference from a relative one. Clue 3 sends them home with <code>cd ~</code> and says the treasure is hidden: a file whose name starts with a dot. <code>ls</code> shows nothing new; <code>ls -a</code> shows <code>.treasure.txt</code>. <code>cat .treasure.txt</code> gives the code word, <code>LANTERN</code>. Last, <code>apt install tree</code> and then <code>tree</code> draw the whole map. Walk the room and ask the questions under What to look for. Students who finish try the ready-for-more challenges. Watch for: <code>ls</code> shows no treasure: the file's name starts with a dot, so only <code>ls -a</code> lists it. A command typed while <code>apt install tree</code> is still running can lose letters; wait for the prompt. It's working when: <code>tree</code> ends with 8 directories, 8 files, and the code word is <code>LANTERN</code>.	Finish the hunt, find the code word, and draw the map.
36 to 45	Closure Say it back, and exit ticket	Ask the essential question again. From the class: every file has a path, the directions down the tree to it; a relative path starts where you are, an absolute path starts at the top with <code>/</code> ; and <code>pwd</code> , <code>ls</code> and <code>cd</code> answer where am I, what is here and go there. Then one more question to discuss: why do computers keep files in a tree of folders instead of one long list? (Finding, grouping and keeping two files with the same name apart.) And where did <code>tree</code> come from? The shell is system software; <code>apt</code> added a small application on top of it. Then hand out the exit ticket.	Answer the essential question out loud, then the exit ticket on their own.

What to look for

Ask these while students work. A student 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 the clue 2 command from a different folder? Why does clue 2 use 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 student 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.

Cross-curricular extensions

- 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 school building as a set of relative directions.
- Computer science: Lesson 2 makes, moves and deletes files in a messy folder.

Vocabulary

terminal	A window where you type commands to a computer instead of clicking.
folder (directory)	A container for files and other folders. Folders inside folders make a tree.
path	Directions to a file or folder, with the names separated by <code>/</code> .
relative path	Directions from where you are now, like <code>school/art</code> .
absolute path	Directions from the very top, starting with <code>/</code> , like <code>/home/guest10/games</code> . They work from anywhere.
home (~)	Your own folder. <code>cd ~</code> (or just <code>cd</code>) always takes you there.
hidden file	A file whose name starts with a dot. <code>ls</code> skips it; <code>ls -a</code> shows it.

Differentiation

ENGLISH LEARNERS

- Post the vocabulary with a picture for each word: a folder inside a folder for tree, a map with a you-are-here dot for pwd.
- Read each clue aloud with the class before students type it.
- Let students draw the path as boxes and arrows before typing it.

STUDENTS WITH IEPs OR 504 PLANS

- Where the IEP allows a modified assignment, give the exact command for each clue and ask only for the pwd at each stop.
- Allow students to press the up arrow to bring back the last command instead of retyping.
- Pair students: one types, the other reads the clue and records.
- Read the exit ticket aloud.

STUDENTS WHO NEED MORE SUPPORT

- Draw the folder tree from the tree map on the board before the hunt starts.
- Remind students that names must be typed exactly: lowercase, no spaces.
- Check in at minute 26: does the student know their own username for the absolute path?

STUDENTS READY FOR MORE

- 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?

Adapting for other grades

Younger students: For grades 4 and 5, do the first two clues together on the projector and let students finish in pairs, with the commands for each clue on the board. Score the first and third rubric rows only (out of 8; 6 or more meets the objectives).

Older students: For students beyond grade 12 or in a college course, add the special entries `.` and `..` that `ls -la` shows, compare `cd -` and `cd ~`, and explain why scripts use absolute paths.

Student worksheet

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.

Exit ticket

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?

Exit ticket

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?

Answer key: worksheet and exit ticket

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

EXIT TICKET	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.

Rubric

Score each row from 1 to 4.

Name: _____

CRITERIA	4 EXCEEDS	3 MEETS	2 APPROACHING	1 BEGINNING
Navigates the tree	Every stop is reached and recorded, and the student also finds a file without the clue's command.	Every stop is reached and its pwd recorded.	Some stops are reached, with help.	The hunt is not started.
Explains paths	Explains relative and absolute paths, and when each is the better choice.	Explains the difference between a relative and an absolute path.	Uses paths but cannot explain the difference.	No explanation.
Finds hidden files	Finds the treasure and explains what makes a file hidden and why ls skips it.	Finds the treasure with ls -a and records the code word.	Finds the treasure with help.	Does not find the treasure.

Total: ____ / 12. A score of 9 or more meets the objectives.

Standards

CSTA's grade 6 to 8 standards do not name file systems or the command line, so this lesson is listed against the closest grade 9 to 10 standards. Middle school teachers can use it as an early introduction to them.

FRAMEWORK	CODE	GRADES	STANDARD	HOW THE LESSON MEETS IT
CSTA 2017	3A-DA-10	9-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.	Students navigate a tree of folders by path and discuss why a tree beats one flat list.
CSTA 2017	3A-CS-02	9-10	Compare levels of abstraction and interactions between application software, system software, and hardware layers.	Students work in the shell, which is system software, and install an application, tree, on top of it.
Common Core Literacy	RST.6-8.3	6-8	Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.	Students follow each clue file's directions exactly, cd down three folders, up two with .., then a full path, checking each stop with pwd.
Common Core Literacy	RST.6-8.4	6-8	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 6-8 texts and topics.	Students work out what the symbols /, .., ~ and a leading dot mean in a path and use each one to reach the next clue.

Checked October 2026 against: CSTA 2017 progression chart, Common Core State Standards, Literacy in Science and Technical Subjects, grades 6-8.

Optional: keep work in a free class

The whole lesson runs at the link on the worksheet, with no accounts and no sign-up. A guest terminal is wiped when the tab is closed.

To give students a terminal of their own that keeps their files, and to see their work, make a free class. It takes about ten minutes, once, and covers every lesson in this series: mytekdev.com/tools/class-setup

Students in a class open mytekdev.com/tools/linux-terminal/lessons/where-am-i/start instead of the worksheet link. It opens their own terminal with the lesson set up.

Terms of use

Thank you for downloading this lesson. The plan, the handouts and the art were made by mytekDev.

You may:

- print and copy it for your students
- change it for your own classes
- post it on your class website or learning management system
- share this file with other teachers, whole and unchanged, with this page included

You may not:

- sell it, or put it in a product you sell
- share a changed copy
- claim it as your own

The app this lesson uses is free and runs in the browser. Students need no account. It is at mytekdev.com.

Questions, or found a mistake? Ask on the Teachers Pay Teachers Q&A for this resource, or write to us at mytekdev.com/contact.

© 2026 mytekDev. All rights reserved.