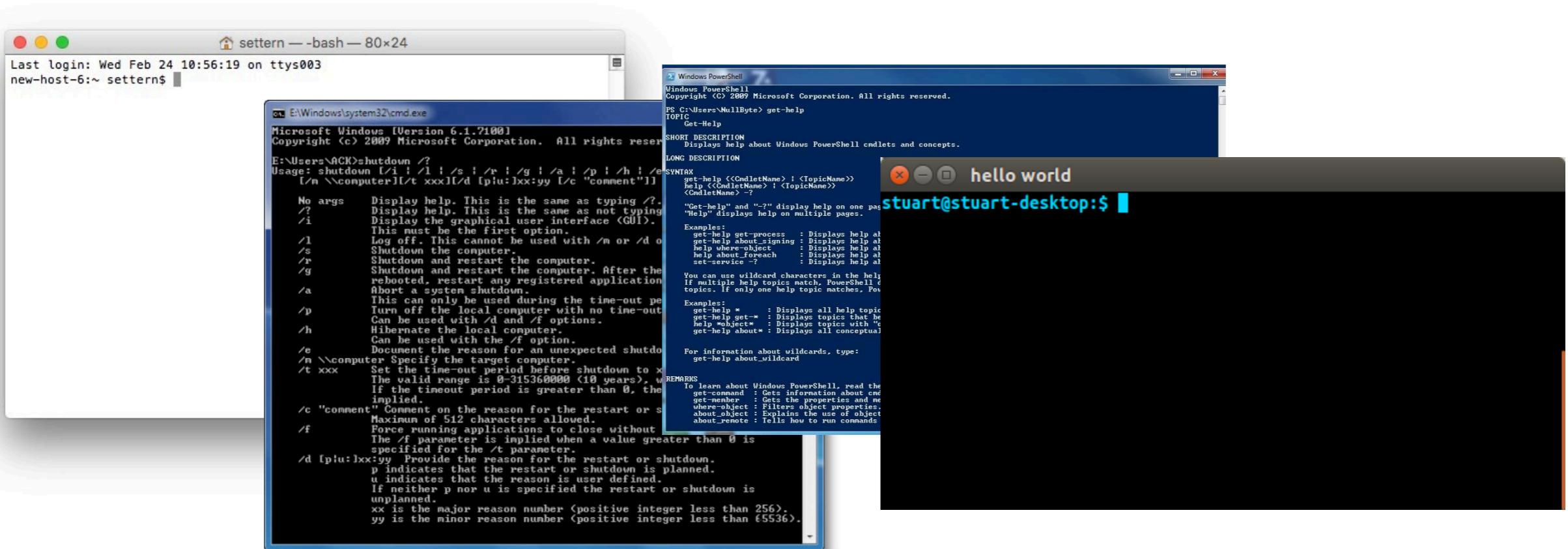


[30 |] The Terminal

Tyler Caraza-Harter



Today's Topics

Terminal Emulators and Shells

- Terminal history
- Shells
- Running programs from a shell

Navigation

Running Programs and Commands

Demos

History: the original terminals



**Mainframe
(powerful computer)**

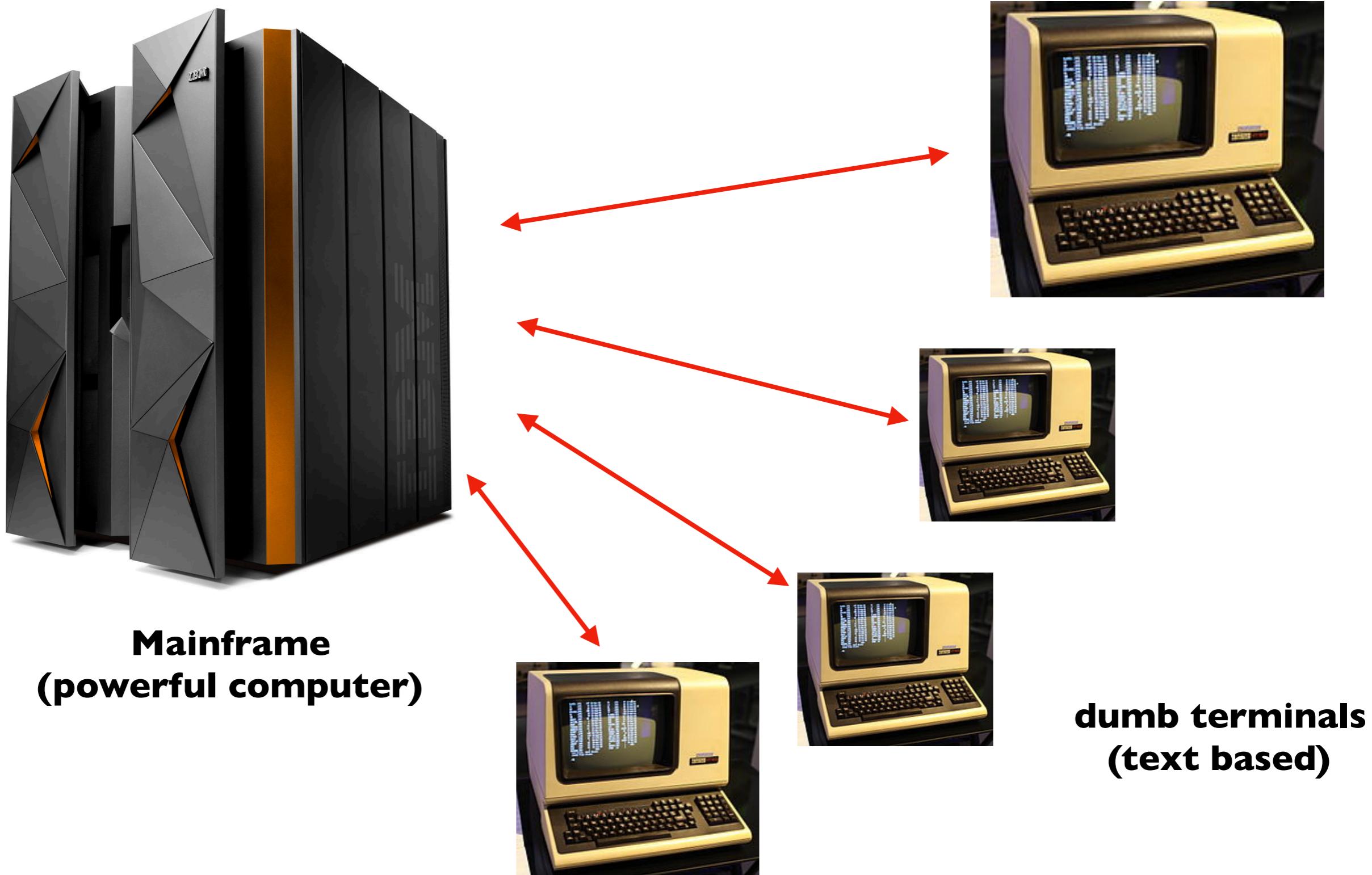
History: the original terminals



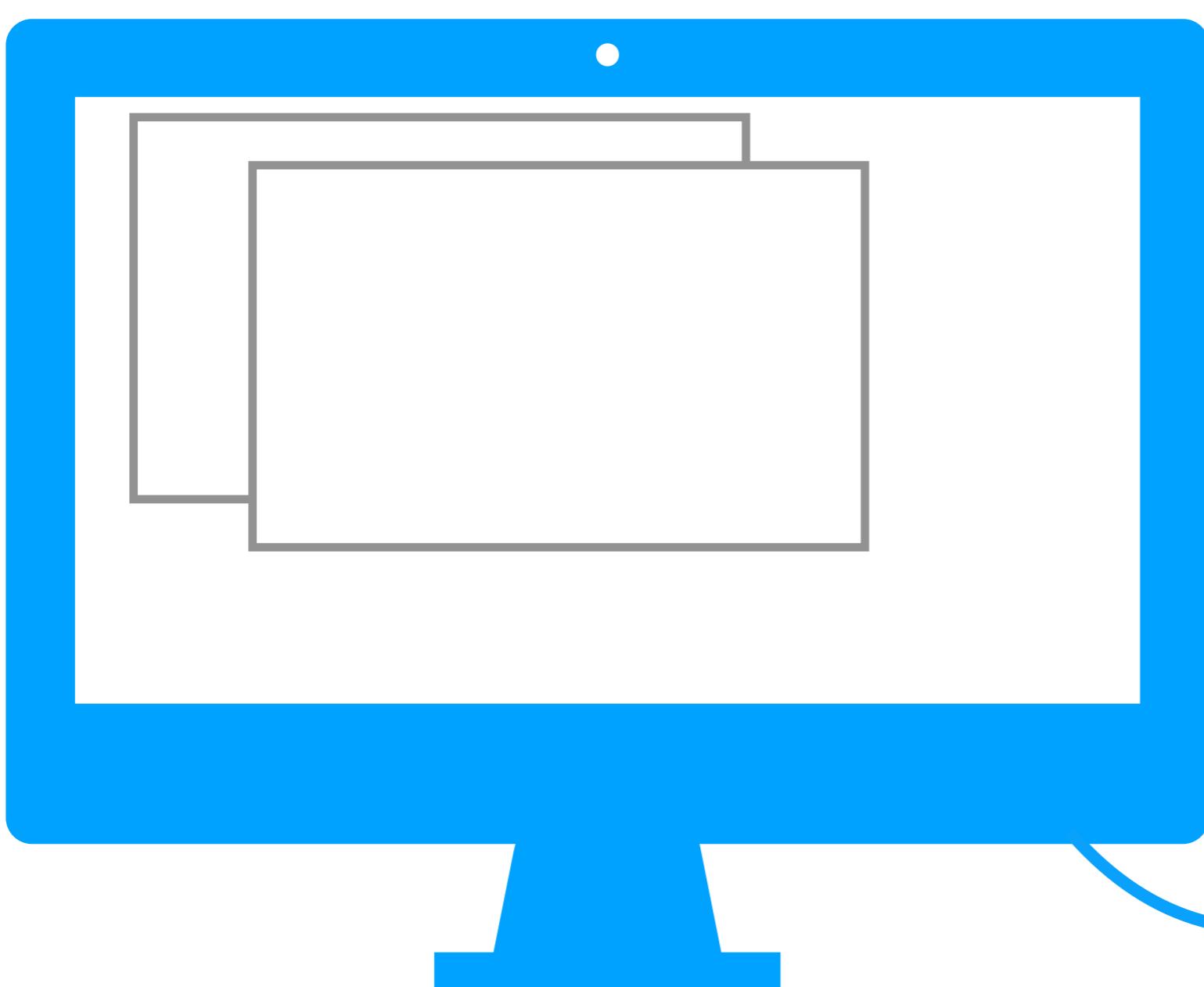
**Mainframe
(powerful computer)**

How to share it?

History: the original terminals

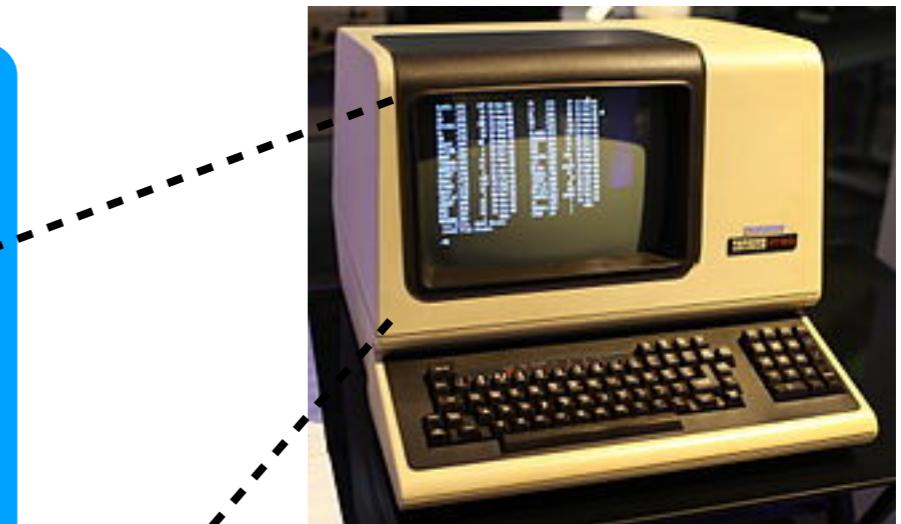
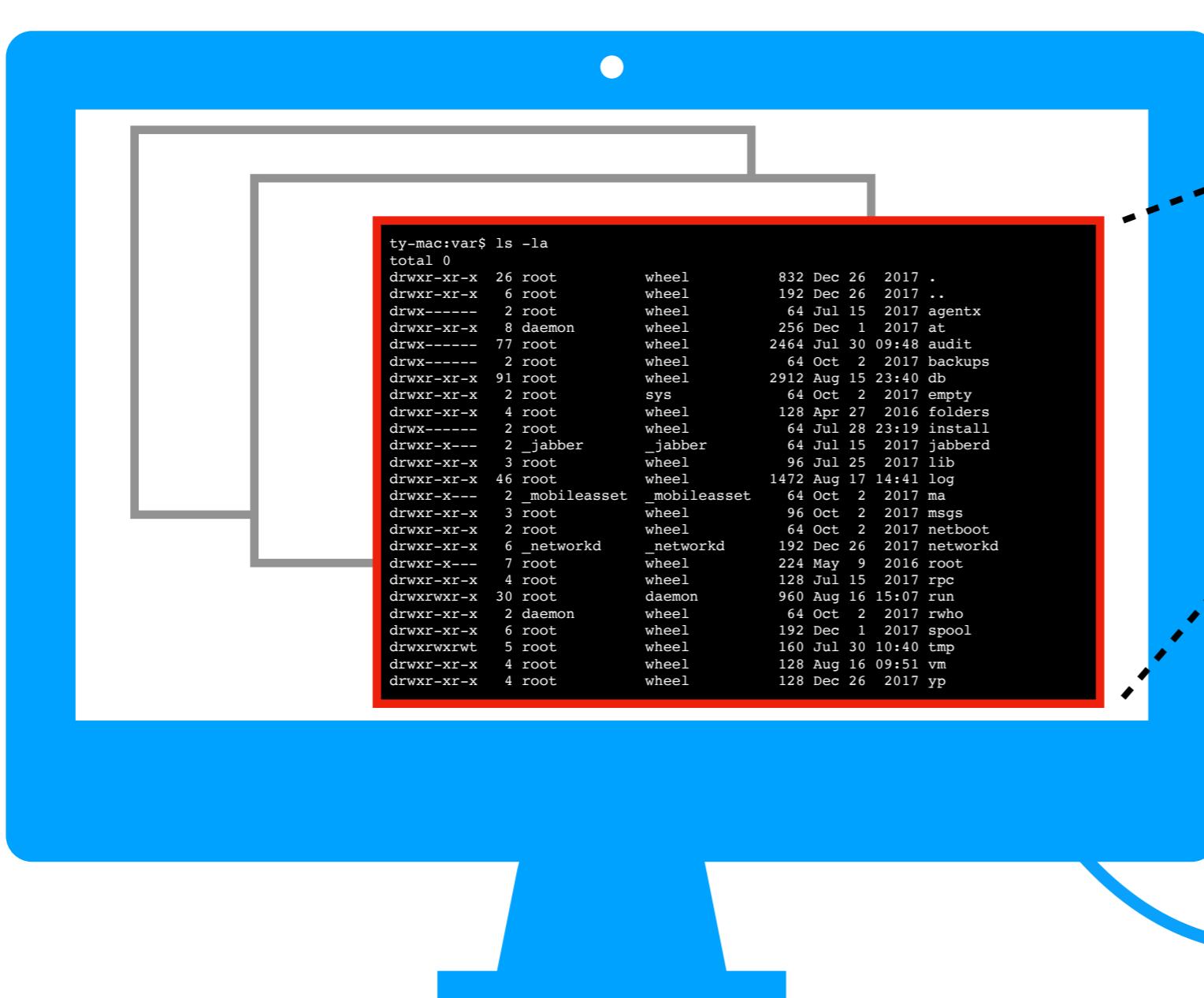


Terminal emulators



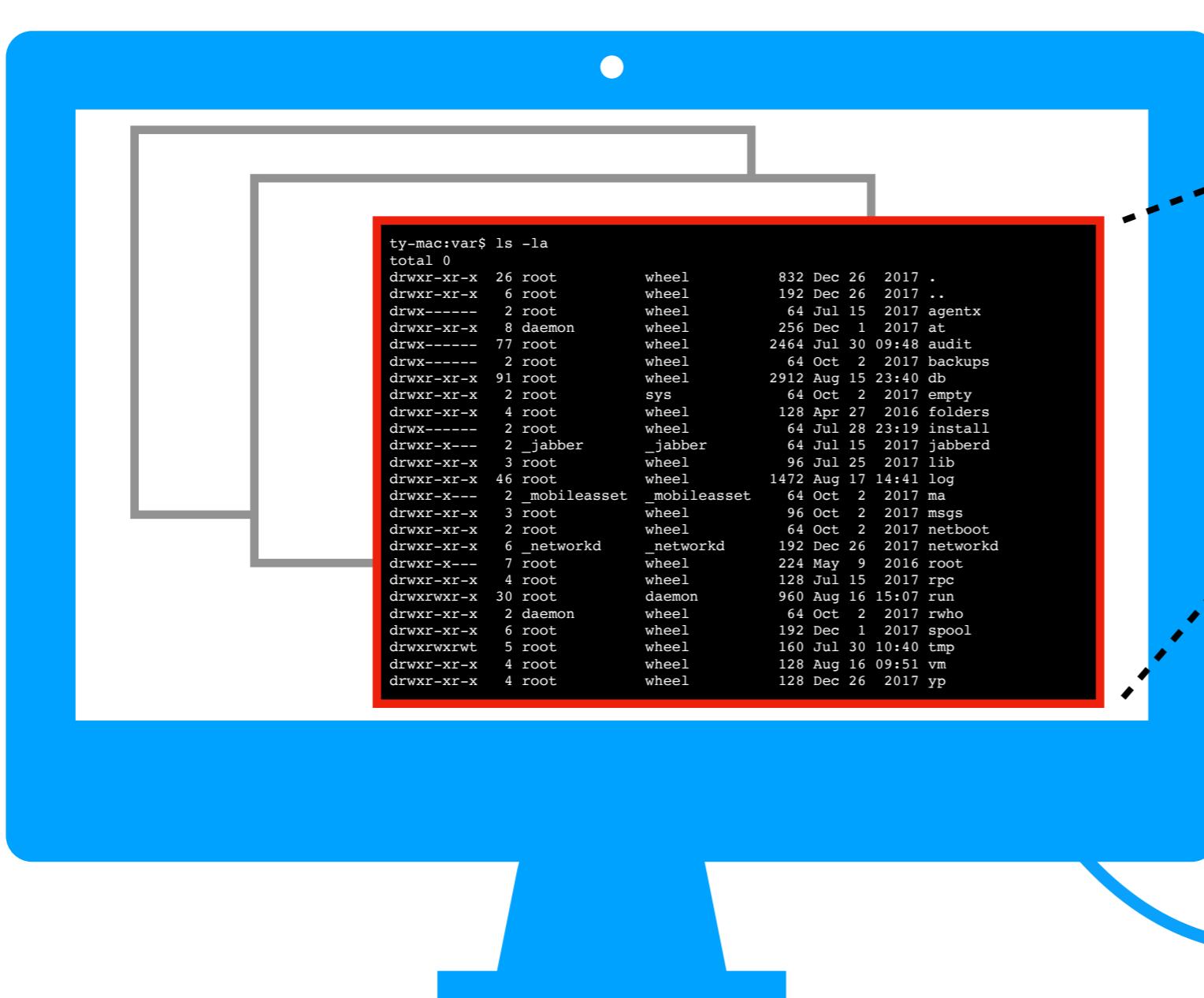
**local computer
(e.g., personal)**

Terminal emulators



**local computer
(e.g., personal)**

Terminal emulators

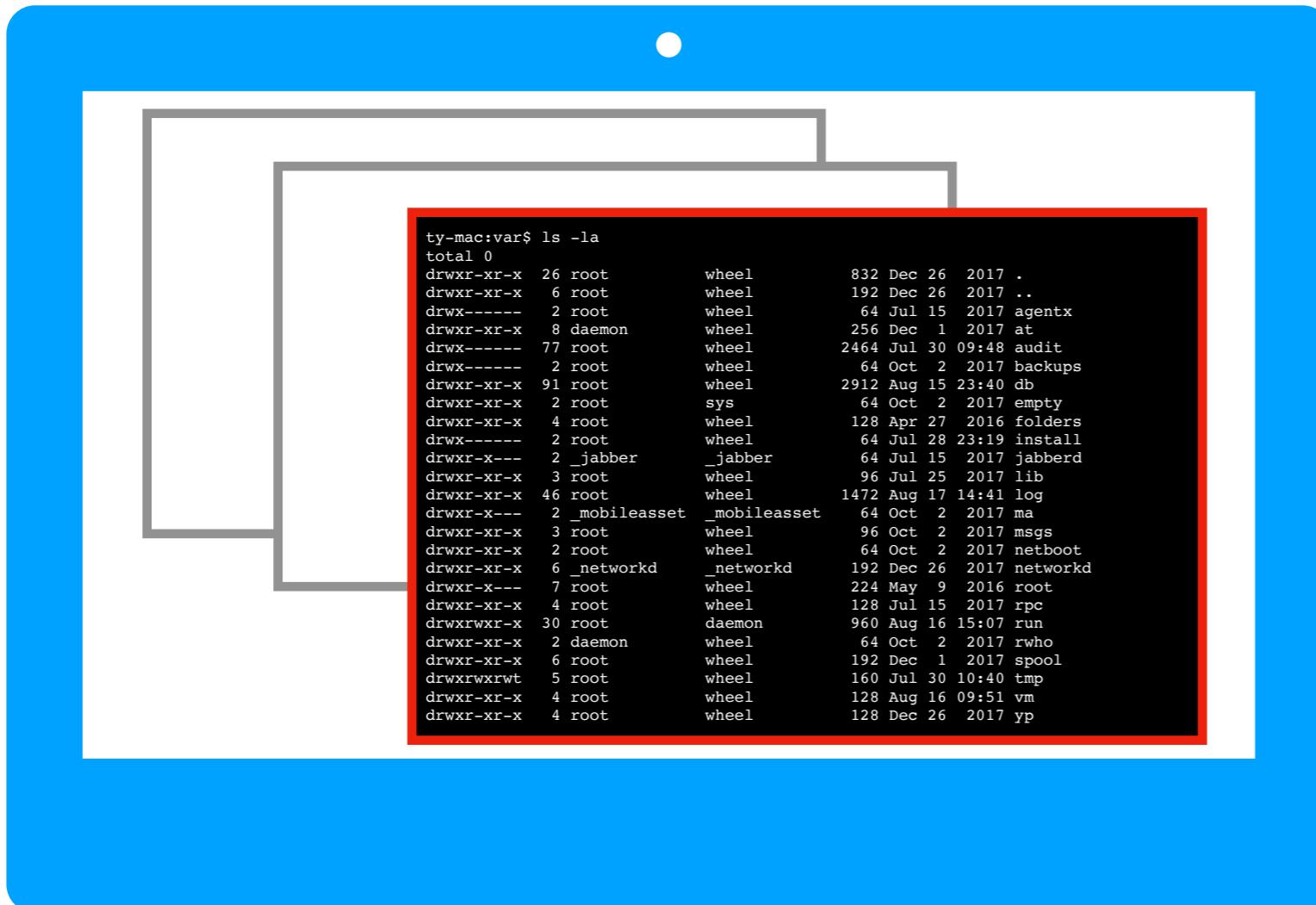


why???



local computer
(e.g., personal)

Terminal emulators



fast

**local computer
(e.g., personal)**



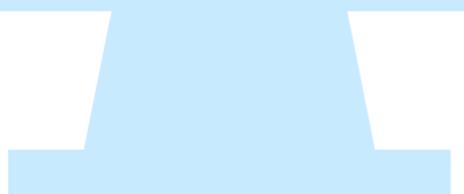
Terminal emulators

Career Tip I: know the difference between **familiar** tools and **good** tools

Practice using good tools that are unfamiliar

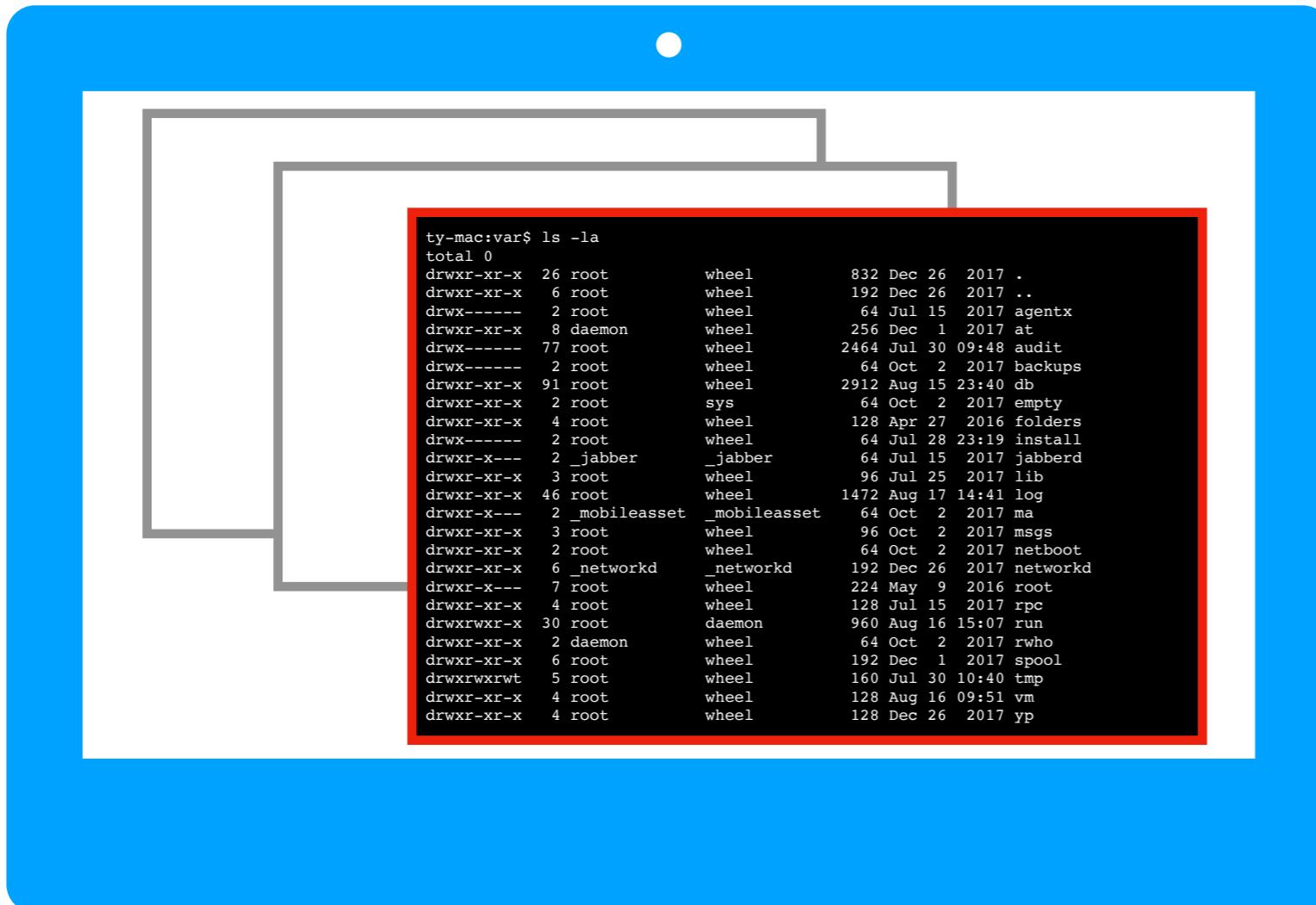
Investment is more important than working hard

```
drwxr-xr-x  2 daemon    wheel        64 Oct  2  2017 /var  
drwxr-xr-x  6 root      wheel     192 Dec  1  2017 spool  
drwxrwxrwt  5 root      wheel    160 Jul 30 10:40 tmp  
drwxr-xr-x  4 root      wheel    128 Aug 16 09:51 vm  
drwxr-xr-x  4 root      wheel    128 Dec 26  2017 yp
```



local computer
(e.g., personal)

Terminal emulators



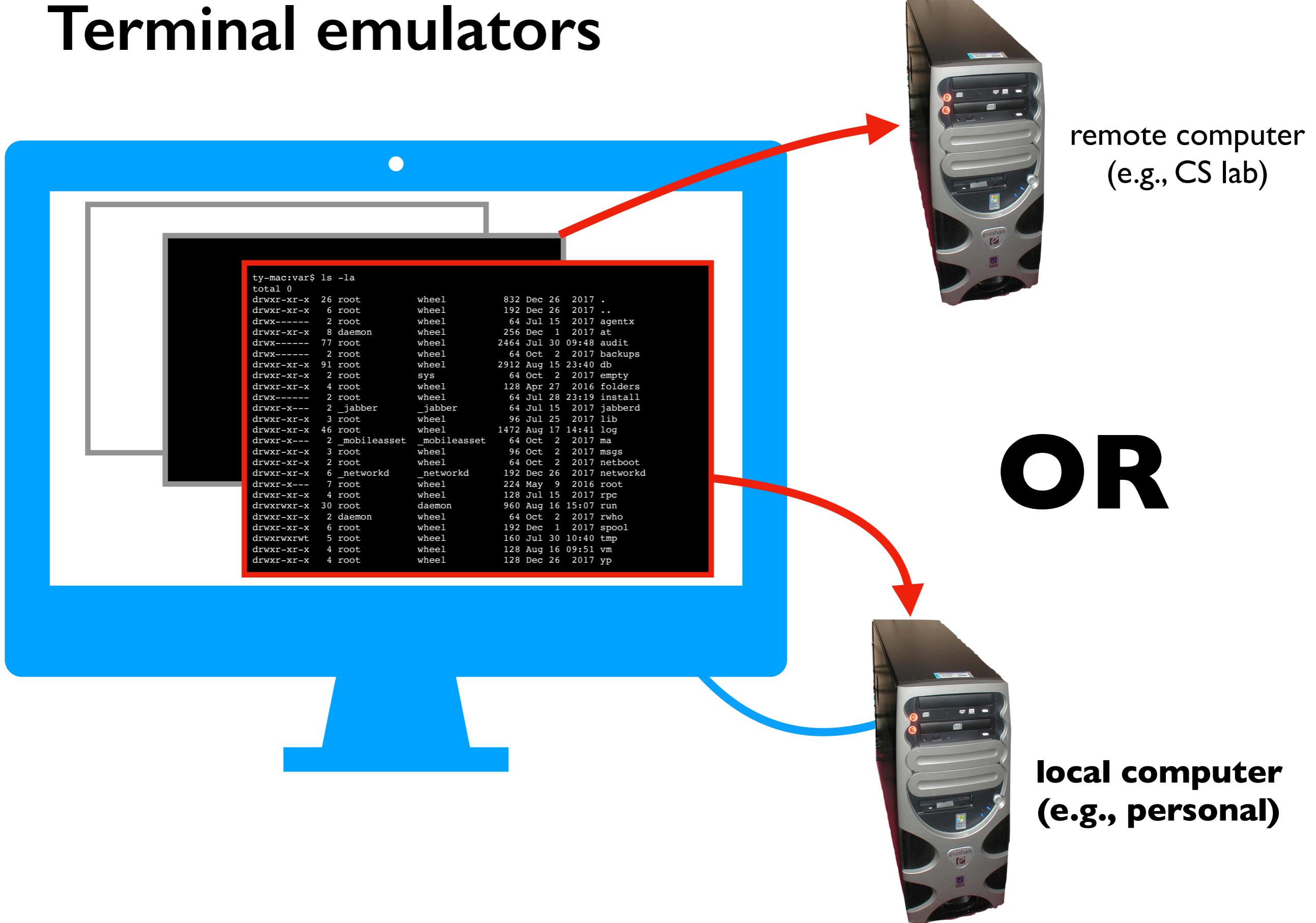
fast

slow



**local computer
(e.g., personal)**

Terminal emulators



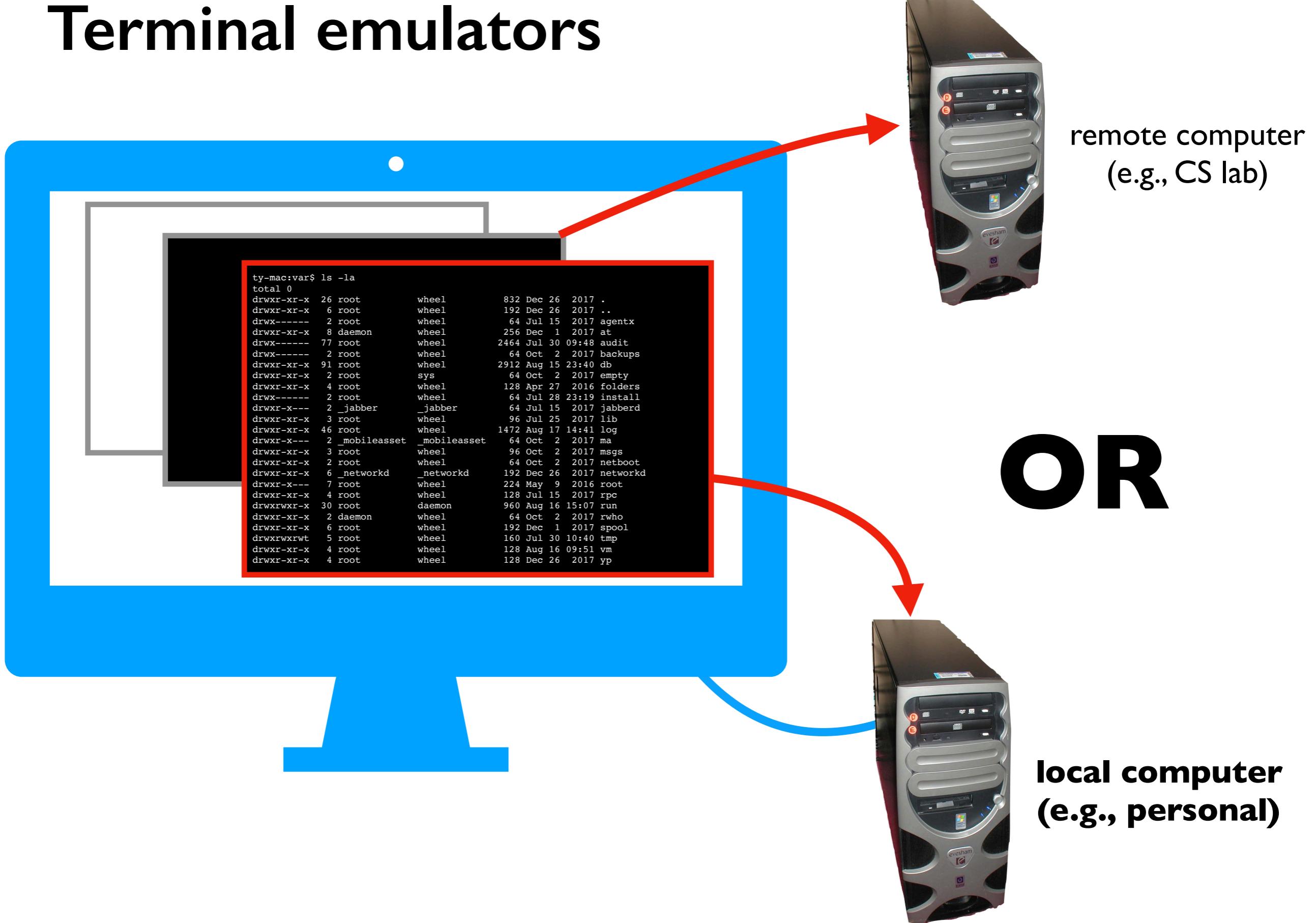
Terminal emulators

Career Tip 2: master the tools that let you work from anywhere

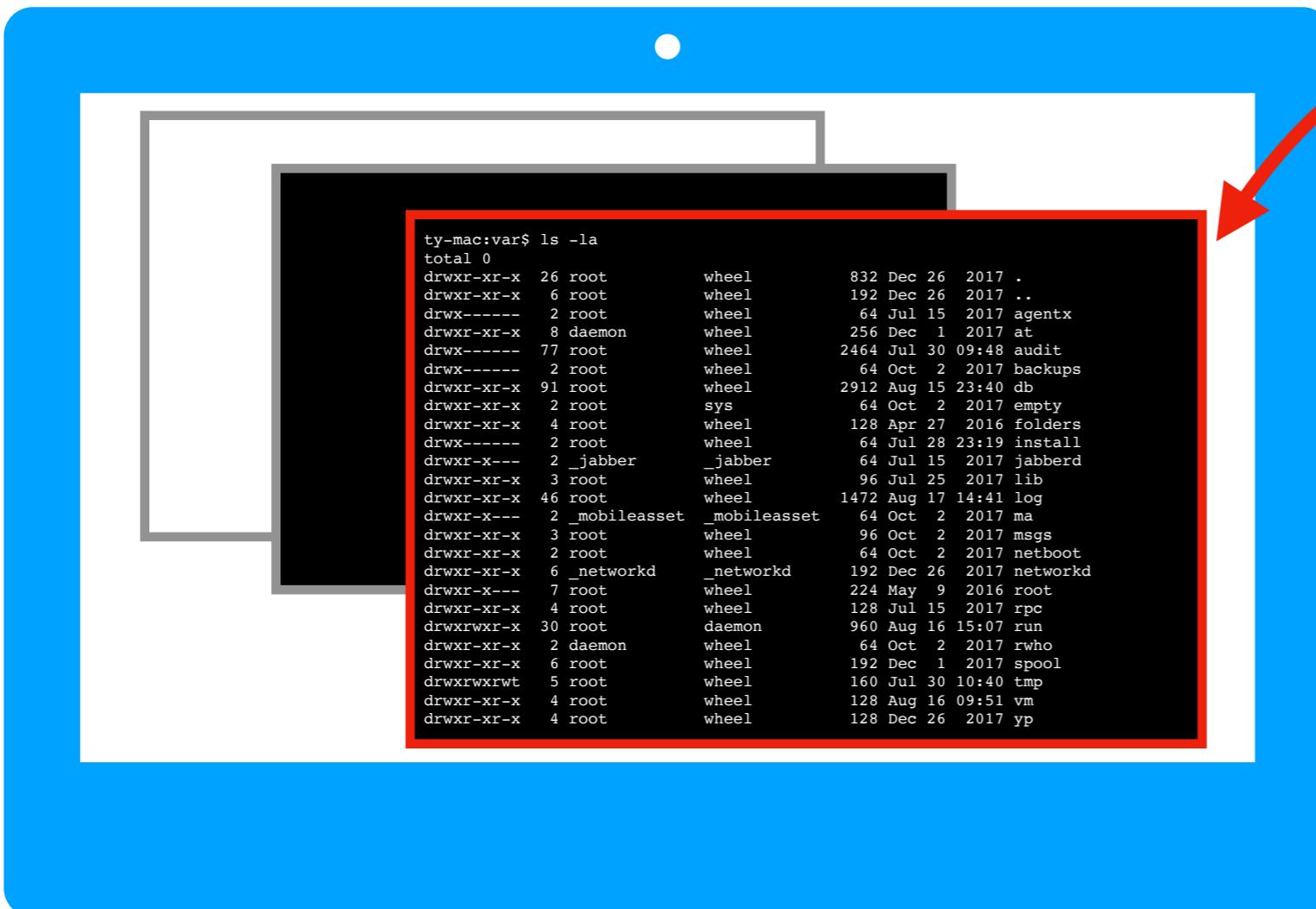
Work for the highest-paying place from the most enjoyable place (home? beach?)



Terminal emulators



Terminal emulators



**programming running in
the terminal emulator
is called a "shell"**

Today's Topics

Terminal Emulators and Shells

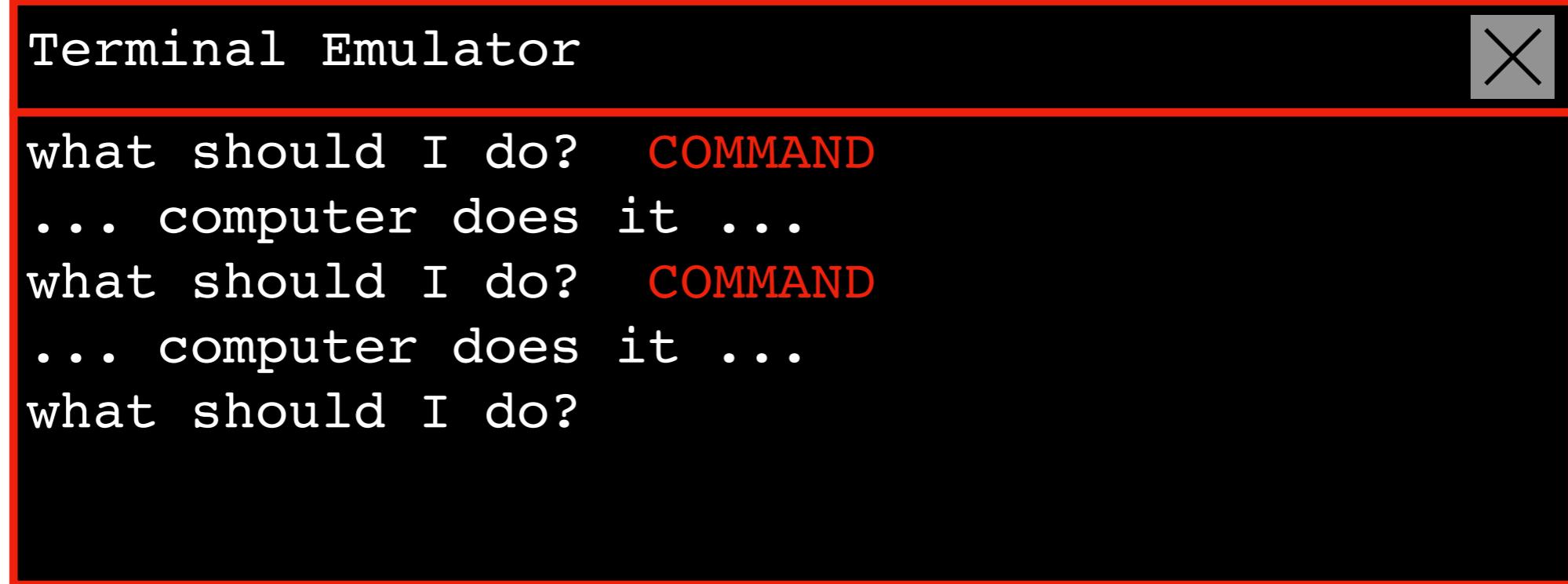
- Terminal history
- Shells
- Running programs from a shell

Navigation

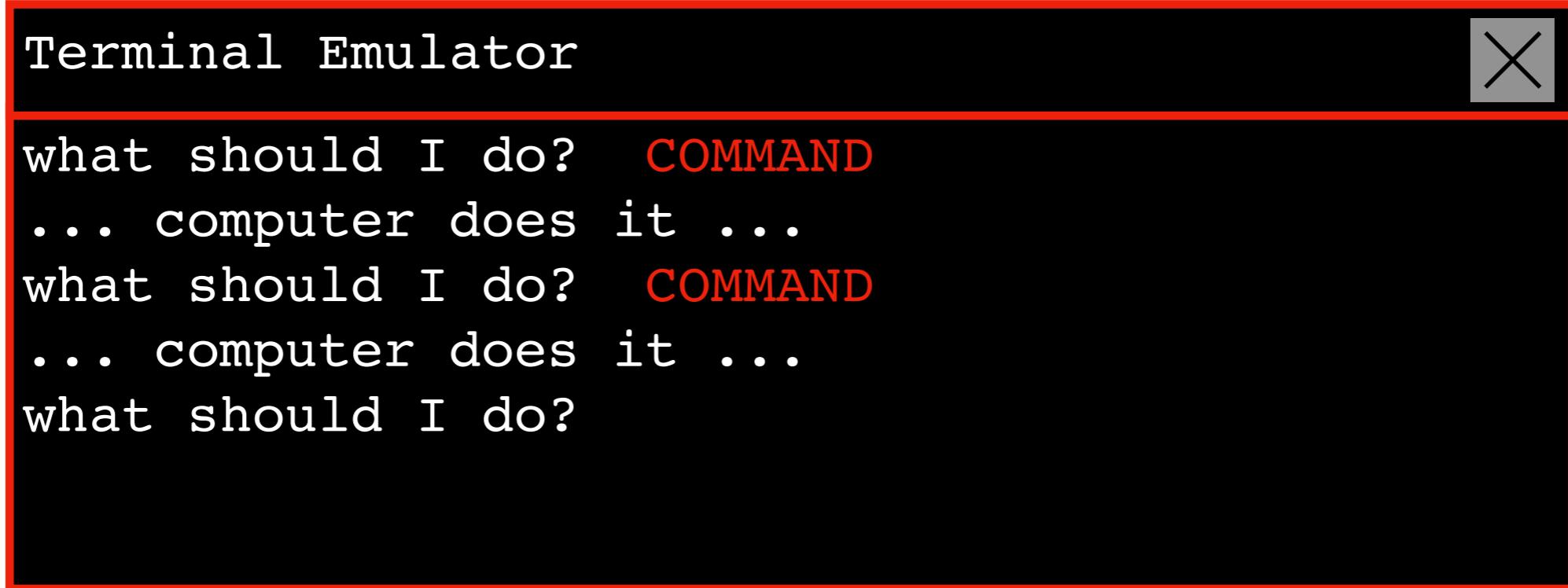
Running Programs and Commands

Demos

Shell: the most helpful program

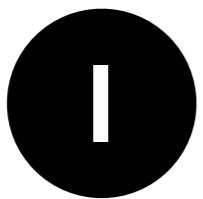


Shell: the most helpful program



A screenshot of a terminal emulator window titled "Terminal Emulator". The window has a red border and a close button in the top right corner. Inside, the text shows a loop of interaction:

```
what should I do? COMMAND  
... computer does it ...  
what should I do? COMMAND  
... computer does it ...  
what should I do?
```

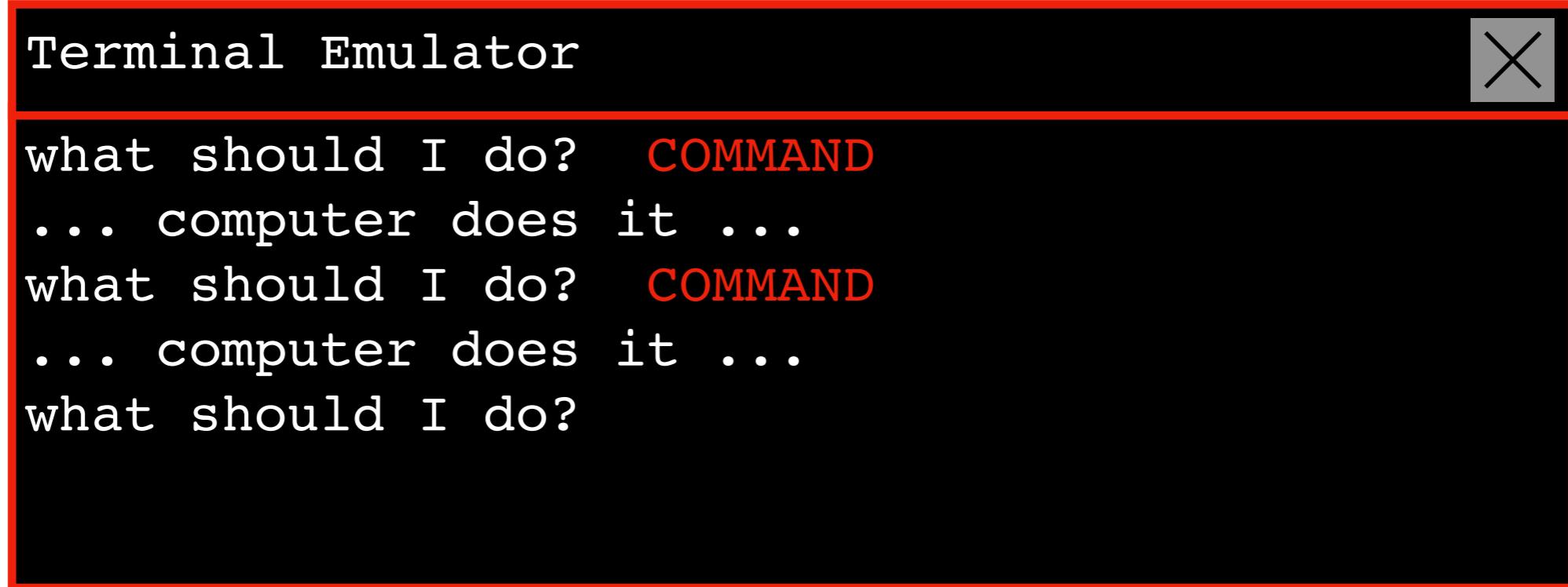


1 **navigate:** dig through folders and files



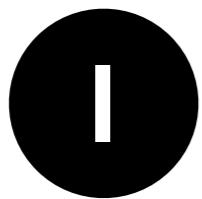
2 **run programs**

Shell: the most helpful program



A screenshot of a terminal emulator window titled "Terminal Emulator". The window has a red border and a close button in the top right corner. Inside, the text shows a loop of interaction:

```
what should I do? COMMAND
... computer does it ...
what should I do? COMMAND
... computer does it ...
what should I do?
```



1 **navigate:** dig through ~~folders~~ directories and files



2 **run programs**

You have a few options when it comes to shells...



<https://en.wikipedia.org/wiki/Seashell>

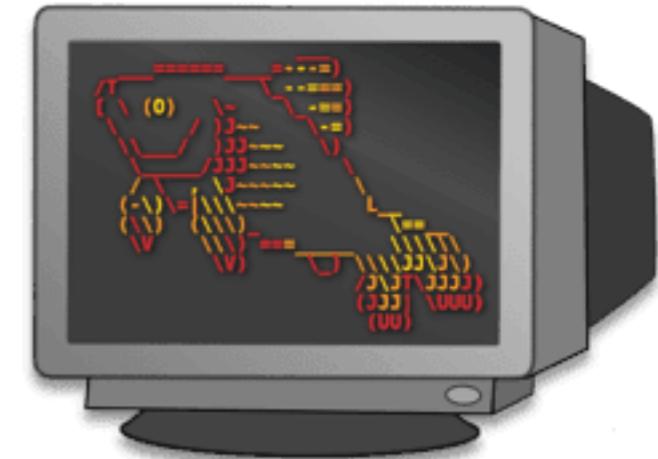
You have a few options when it comes to shells...



cmd



PowerShell



fish



ksh

csh

zsh

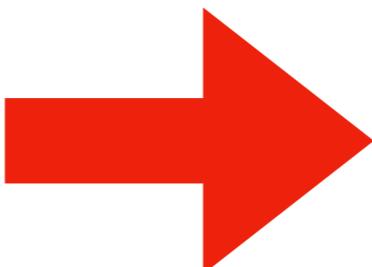
today



/bin/sh
Bourne Shell

1979

Stephen Bourne



BASH
THE BOURNE-AGAIN SHELL

Today's Topics

Terminal Emulators and Shells

- Terminal history
- Shells
- **Running programs from a shell**

Navigation

Running Programs and Commands

Demos

Running Programs

Running programs is easy, just type name of the program and hit enter:

```
ty-mac:var$
```

Running Programs

Running programs is easy, just type name of the program and hit enter:

```
ty-mac:var$ ls
```

Running Programs

Running programs is easy, just type name of the program and hit enter:

```
ty-mac:var$ ls
agentx      jabberd      root
at           lib          rpc
audit        log          run
backups     ma          rwho
```

```
ty-mac:var$
```

Running Programs

Running programs is easy, just type name of the program and hit enter:

program name

```
prompt ty-mac:var$ ls
agentx      jabberd      root
at           lib          rpc
audit        log          run
backups     ma           rwho
prompt ty-mac:var$
```

a "prompt" is the question, *what should I do?*

Today's Topics

Terminal Emulators and Shells

Navigation

- Storage Drives (Windows)
- Files
- Directories (aka Folders)
- Windows vs. Mac

Running Programs and Commands

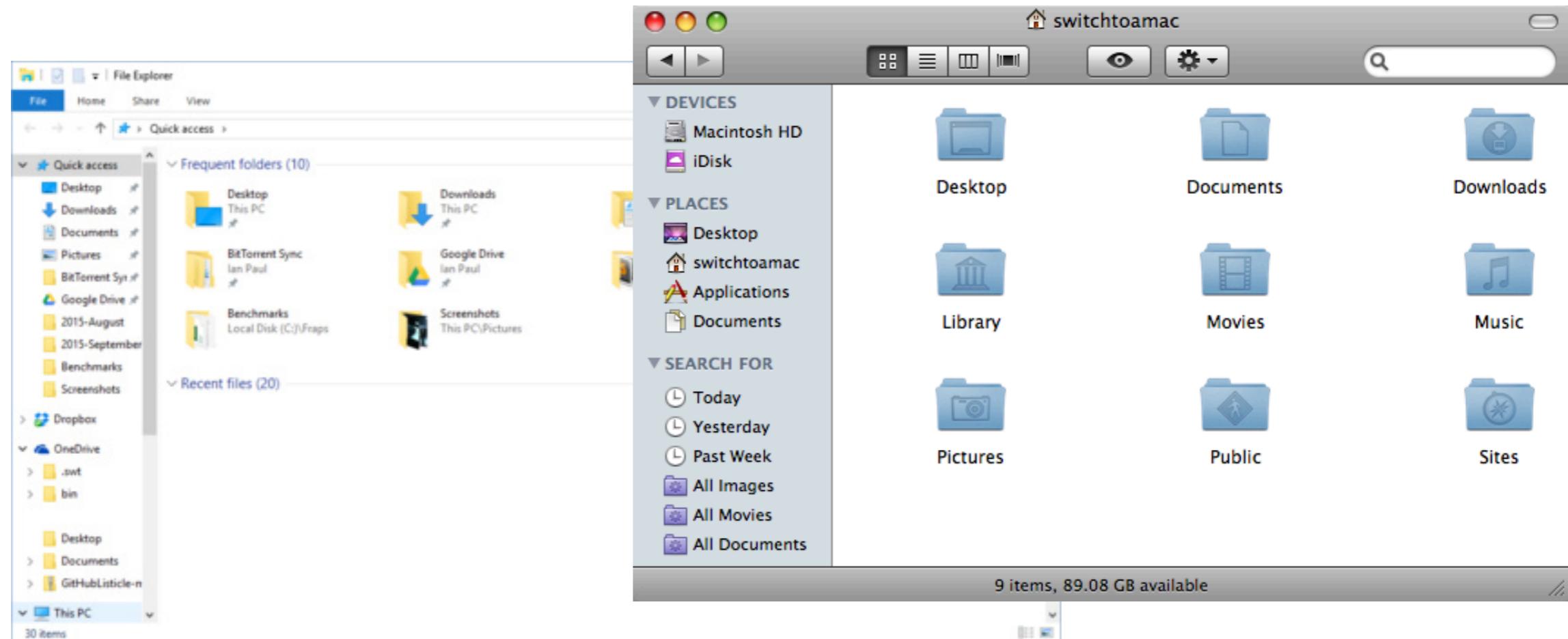
Demos

What is navigation?

Navigation is looking around for files/folders you want

Navigation programs

- File Explorer (Windows)
- Finder (Mac)



What is navigation?

Navigation is looking around for files/folders you want

Navigation programs

- File Explorer (Windows)
- Finder (Mac)

With shell, navigate w/ various commands...

ls

pwd

cd

cat

mkdir

• • •

Today's Topics

Terminal Emulators and Shells

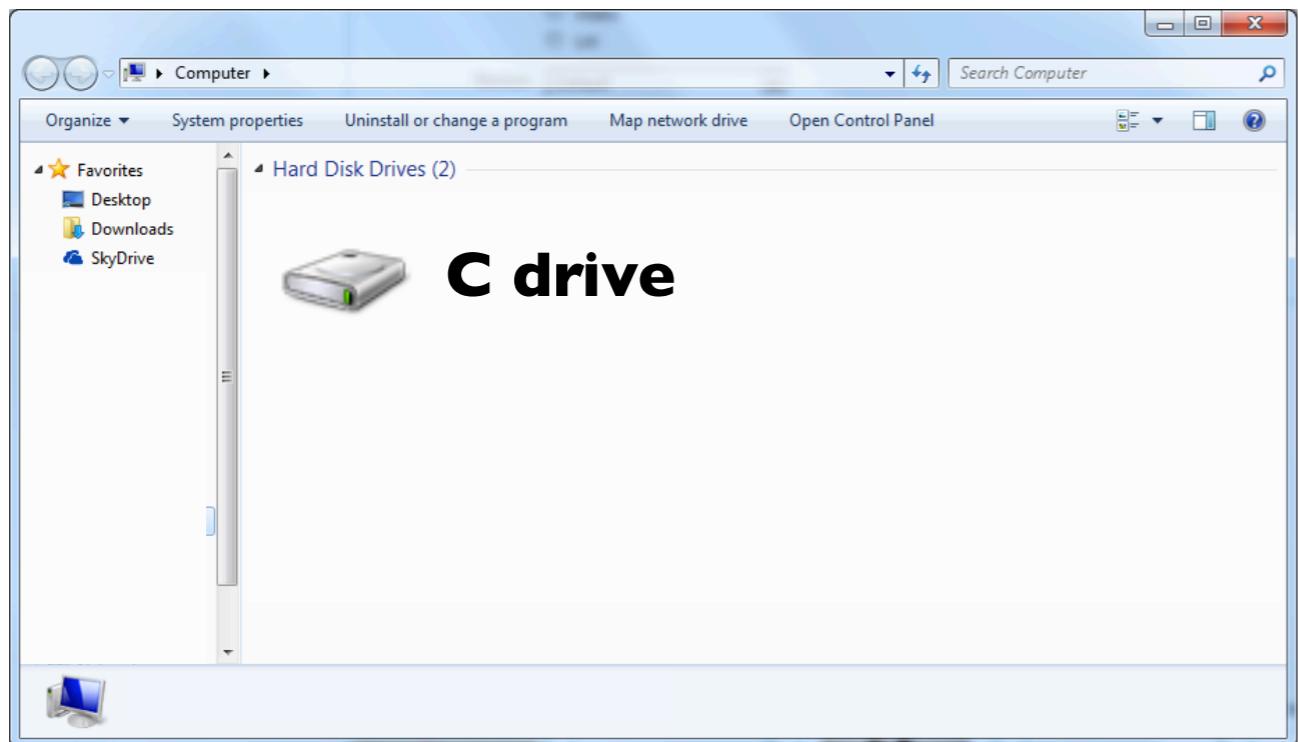
Navigation

- Storage Drives (Windows)
- Files
- Directories (aka Folders)
- Windows vs. Mac

Running Programs and Commands

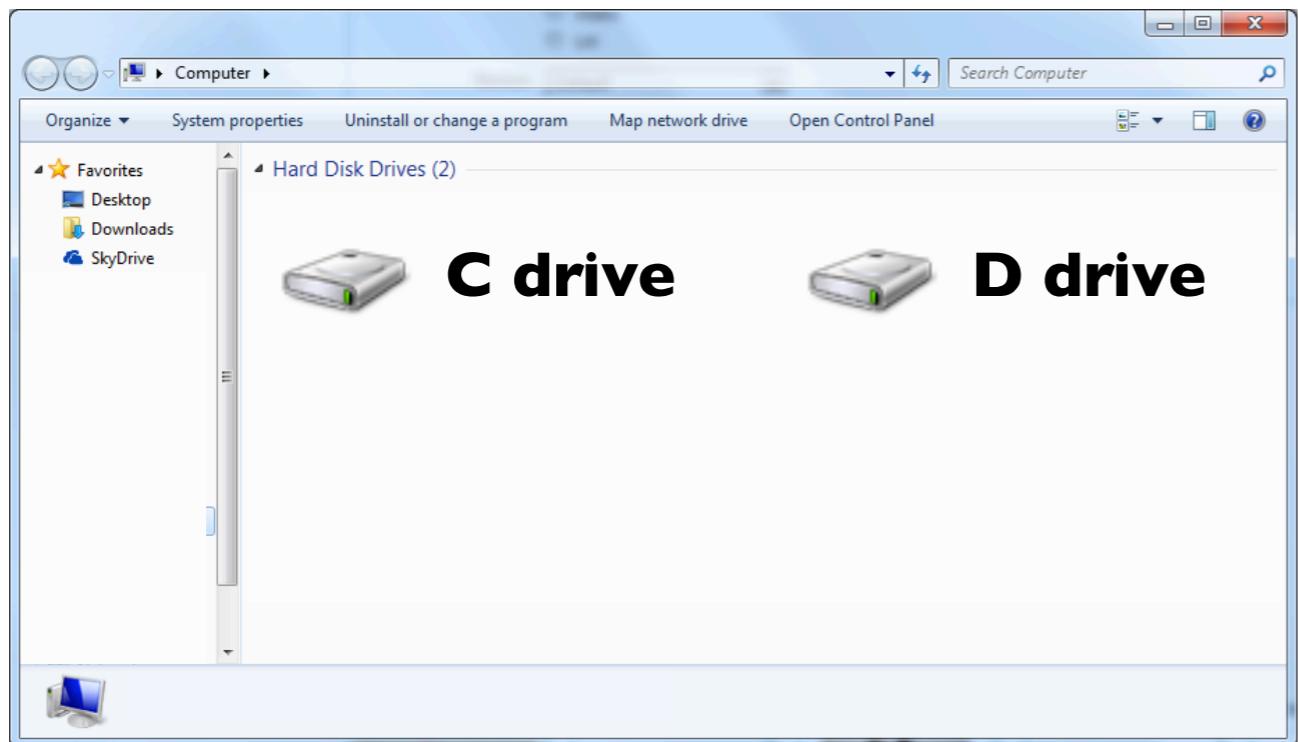
Demos

Windows Storage Drives



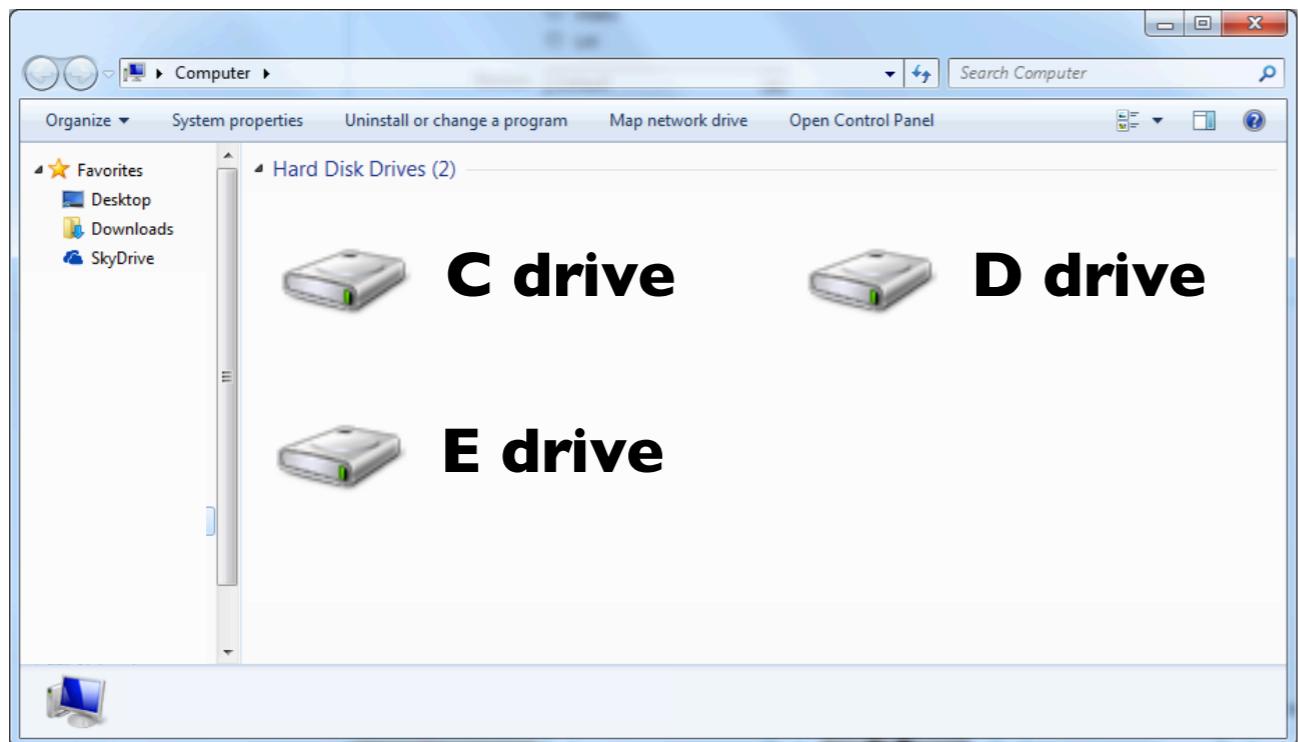
**Each added drive is given
its own drive letter**

Windows Storage Drives



**Each added drive is given
its own drive letter**

Windows Storage Drives



**Each added drive is given
its own drive letter**



Today's Topics

Terminal Emulators and Shells

Navigation

- Storage Drives (Windows)
- **Files**
- Directories (aka Folders)
- Windows vs. Mac

Running Programs and Commands

Demos

Files

Each file has a name, called a “path name”

c:\README.txt

c:\hw.docx

d:\page.html

e:\main.py

Files

Each file has a name, called a “path name”

c:\README.txt

filename



c:\hw.docx

d:\page.html

e:\main.py

Files

Each file has a name, called a “path name”

The diagram illustrates the structure of a file path. It shows the text "c:\README.txt" in black. A red bracket above the ".txt" suffix is labeled "filename" in red. Another red bracket below the "c:\" prefix is labeled "pathname" in red.

c:\README.txt

filename

pathname

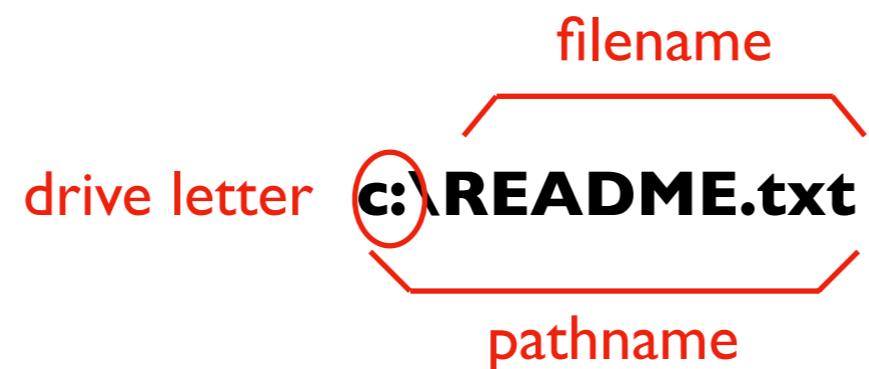
c:\hw.docx

d:\page.html

e:\main.py

Files

Each file has a name, called a “path name”



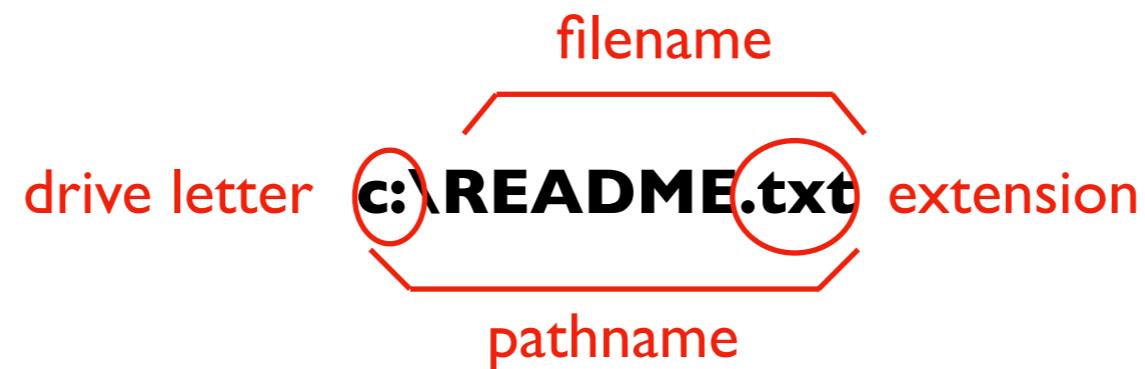
c:\hw.docx

d:\page.html

e:\main.py

Files

Each file has a name, called a “path name”



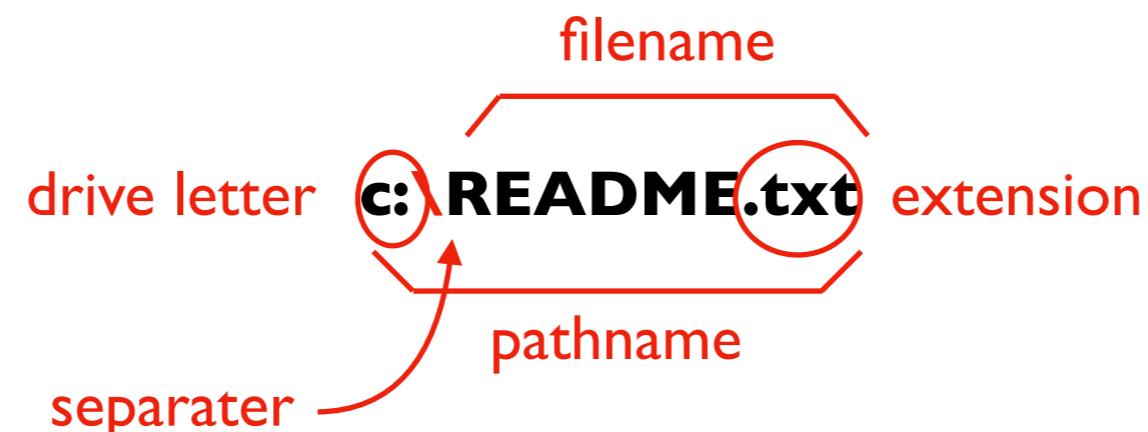
c:\hw.docx

d:\page.html

e:\main.py

Files

Each file has a name, called a “path name”



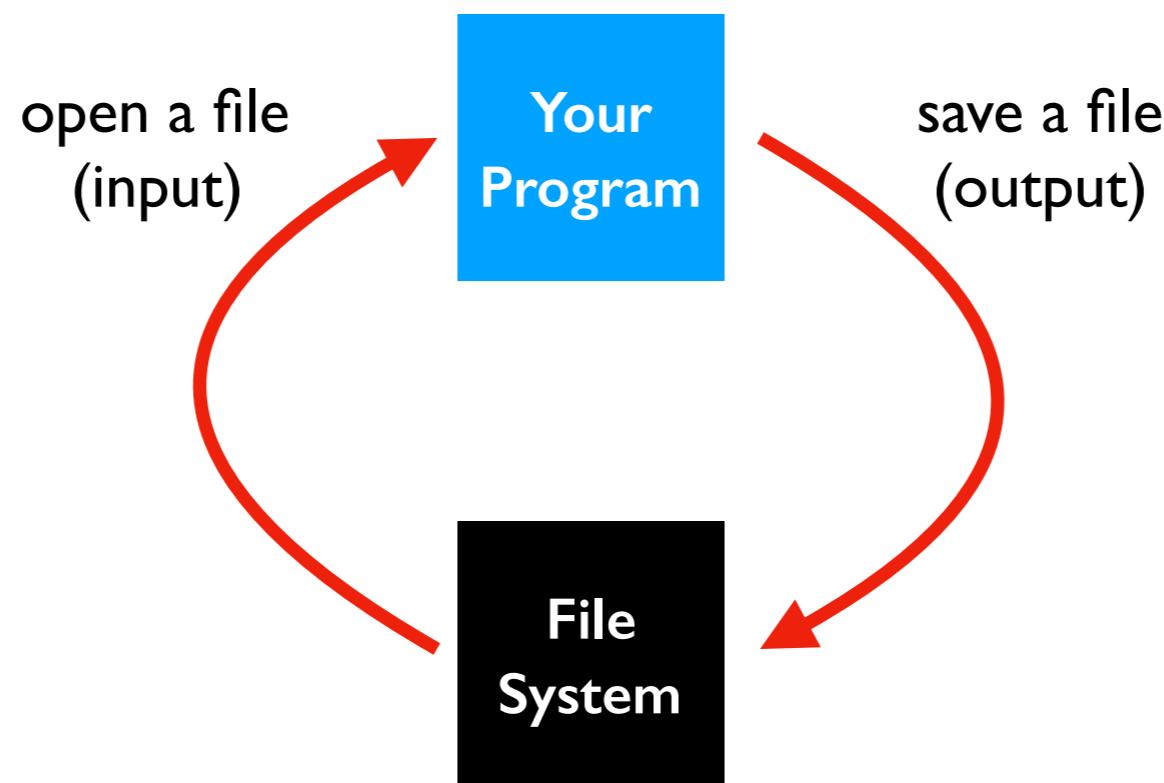
c:\hw.docx

d:\page.html

e:\main.py

Files

Files might be either **input** or **output** for your programs



Today's Topics

Terminal Emulators and Shells

Navigation

- Storage Drives (Windows)
- Files
- Directories (aka Folders)
- Windows vs. Mac

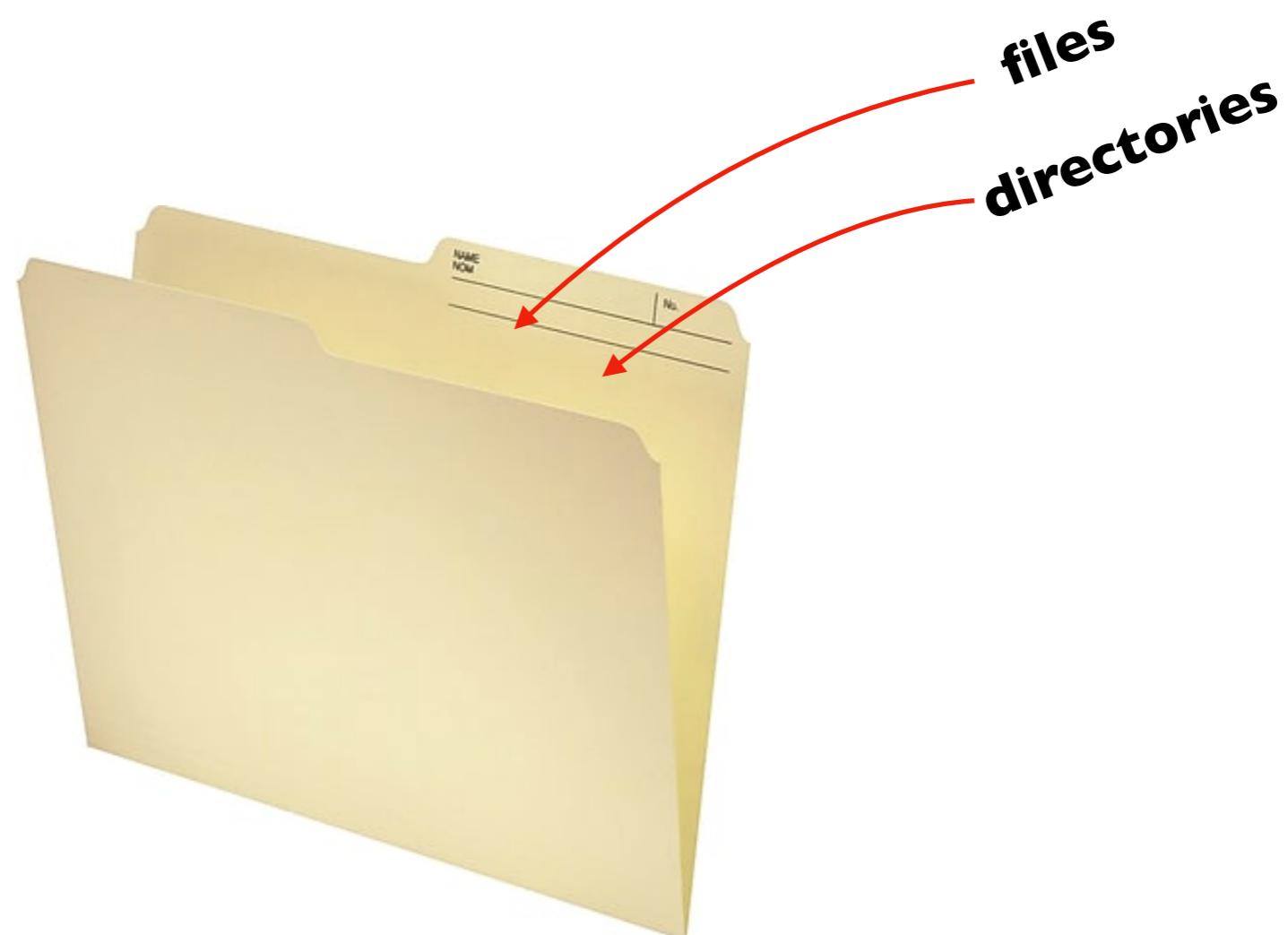
Running Programs and Commands

Demos

Directories

Directories are used to organize files and sub directories

- Also called “folders”

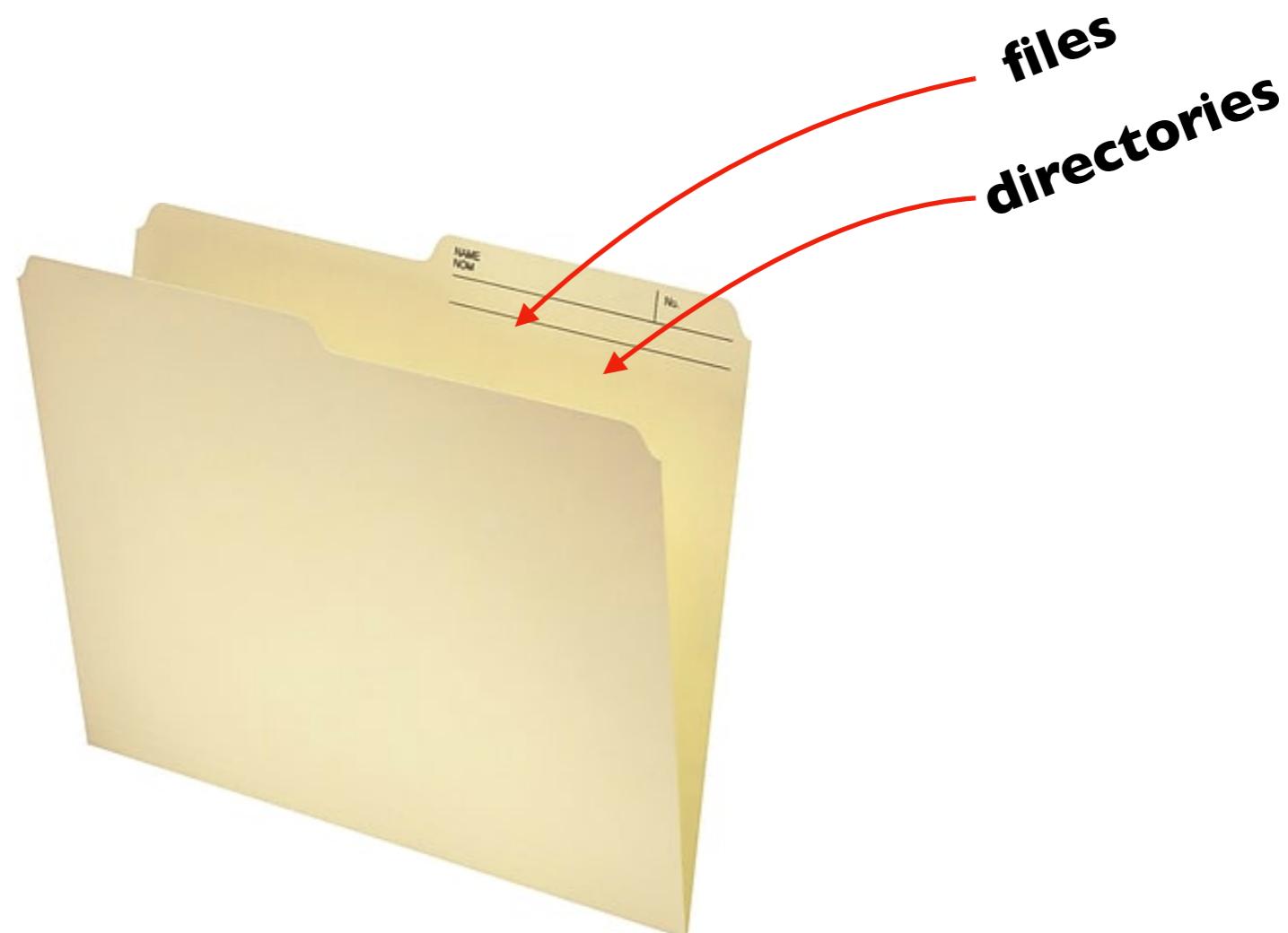


https://www.staples.ca/en/Staples-Recycled-File-Folder-1-2-Cut-Letter-Size-11-pt-Manila-100-Pack/product_13579_1-CA_1_20001

Directories

Directories are used to organize files and sub directories

- Also called “folders”
- A directory also has pathname



https://www.staples.ca/en/Staples-Recycled-File-Folder-1-2-Cut-Letter-Size-11-pt-Manila-100-Pack/product_13579_1-CA_1_20001

Directories

Directories are used to organize files and sub directories

- Also called “folders”
- A directory also has pathname

Example paths:

- c:\my-directory\file1.docx
- c:\my-directory\file2.docx
- c:\my-directory\file3.docx



in my-directory

Directories

Directories are used to organize files and sub directories

- Also called “folders”
- A directory also has pathname

Example paths:

- c:\my-directory\file1.docx
- c:\my-directory\file2.docx
- c:\my-directory\file3.docx
- c:\directory1\directory2\file1.docx
- c:\same-dir\same-dir\readme.txt

Directories

Directories are used to organize files and sub directories

- Also called “folders”
- A directory also has pathname

Example paths:

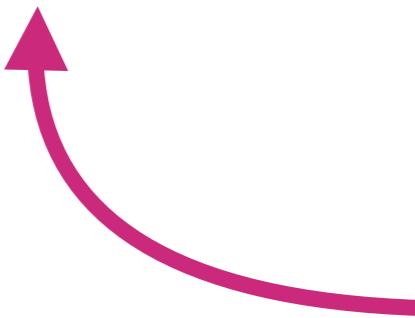
- c:\my-directory\file1.docx
- c:\my-directory\file2.docx
- c:\my-directory\file3.docx
- c:\directory1\directory2\file1.docx
- c:\same-dir\same-dir\readme.txt

two types of paths: **relative** or **absolute**

Relative Paths

Where is the Computer Science building?

- Answer 1: 1210 W Dayton St, Madison, WI 53706
- Answer 2: on the other side of Johnson street



When is Answer 2 appropriate?

Relative Paths

Where is the Computer Science building?

- Answer 1: 1210 W Dayton St, Madison, WI 53706
- Answer 2: on the other side of Johnson street



When is Answer 2 appropriate?

- When you're in the psychology building
- It may be more convenient

Relative Paths

Where is the Computer Science building?

- Answer 1: 1210 W Dayton St, Madison, WI 53706
- Answer 2: on the other side of Johnson street



When is Answer 2 appropriate?

- When you're in the psychology building
- It may be more convenient

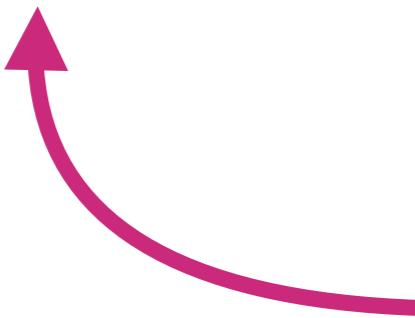
Pathnames are absolute (answer 1) or relative (answer 2)

- Absolute paths: always possible
- Relative paths: if current location is known

Relative Paths

Where is the Computer Science building?

- Answer 1: 1210 W Dayton St, Madison, WI 53706
- Answer 2: on the other side of Johnson street



When is Answer 2 appropriate?

- When you're in the psychology building
- It may be more convenient

Pathnames are absolute (answer 1) or relative (answer 2)

- Absolute paths: always possible
- Relative paths: if current location is known
- Working Directory (our current location)

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	
c:\x\y\z\my.docx	c:\x\y	
c:\x\y\z	c:\x	

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	
c:\x\y\z	c:\x	

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\test.txt
c:\test.txt	c:\	
c:\x\y\z	c:\x	
c:\x	c:\x\y\z	

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	..\\..\\test.txt
c:\x\y\z	c:\x	
c:\x	c:\x\y\z	

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\\test.txt	c:\\	..\\..\\test.txt
c:\\x\\y\\z	c:\\x	.\\y\\z
c:\\x	c:\\x\\y\\z	

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	

Two special directory names

- “..” means up a directory
- “.” means current directory

Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory

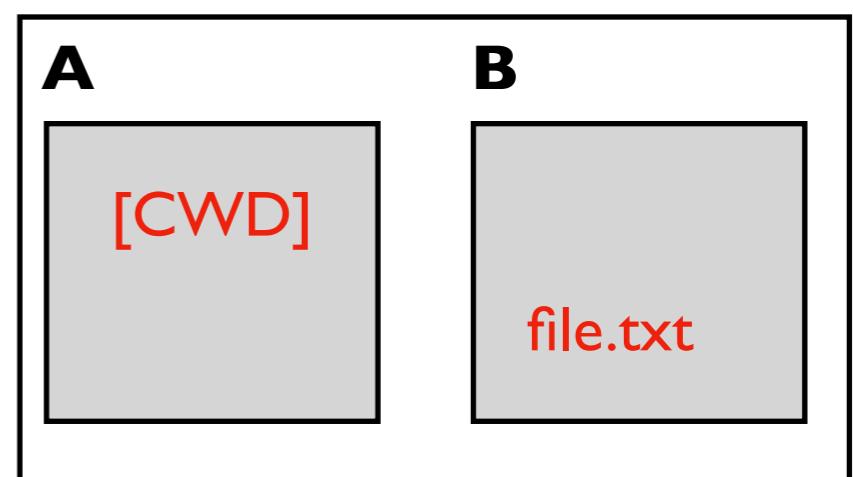
Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	..\\..\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory

c:\

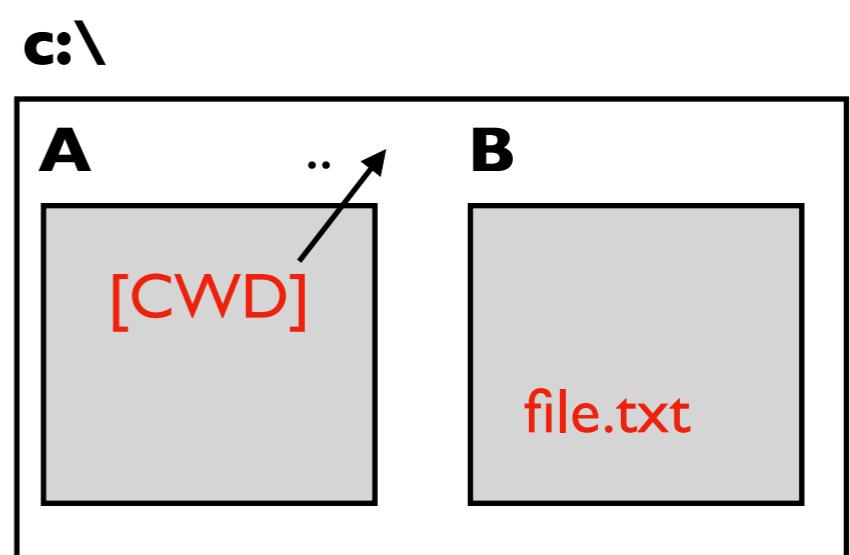


Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	..\\..\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory

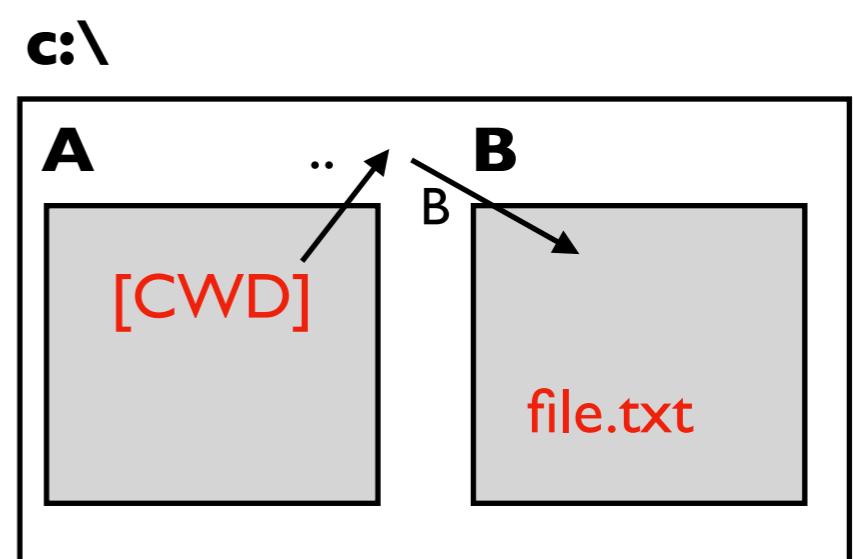


Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory

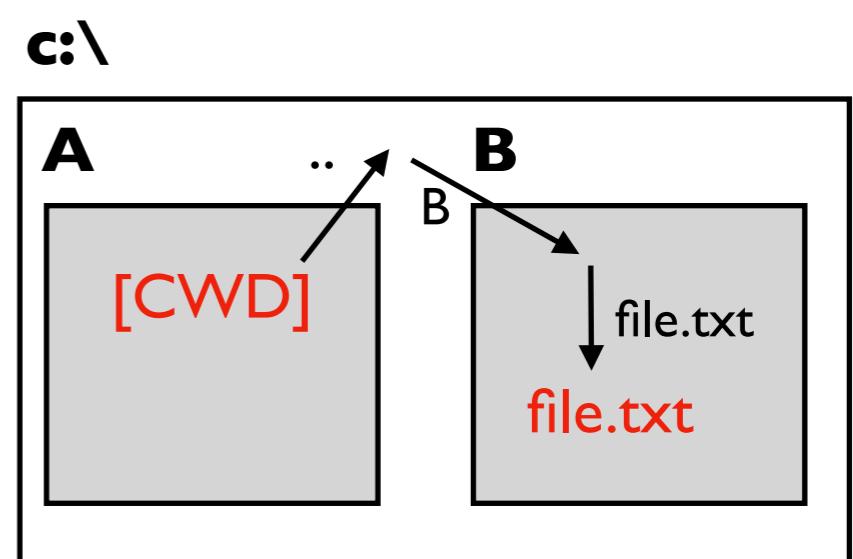


Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory



Absolute vs. Relative

Absolute Path	Working Directory	Relative Path
c:\test.txt	c:\	test.txt
c:\x\y\z\my.docx	c:\x\y\z	my.docx
c:\x\y\z\my.docx	c:\x\y	z\my.docx
c:\x\y\z	c:\x	y\z
c:\test.txt	c:\	.\\test.txt
c:\test.txt	c:\	.\\.\\test.txt
c:\x\y\z	c:\x	.\\y\\z
c:\x	c:\x\y\z	..\\..
c:\B\file.txt	c:\A	..\\B\\file.txt

Two special directory names

- “..” means up a directory
- “.” means current directory

more examples in demo later...

Today's Topics

Terminal Emulators and Shells

Navigation

- Storage Drives (Windows)
- Files
- Directories (aka Folders)
- Windows vs. Mac

Running Programs and Commands

Demos

Multiple Drives in Mac

Windows

- Absolute paths start with `c:\` or `d:\`
- Indicates which drive

Mac

- Absolute paths start with `/`
- Example: `/Users/tyler/my-file.docx`
- Don't know which drive

How can we use multiple drives if every file paths starts the same???

`/.....`

Multiple Drives in Mac

Windows

- Absolute paths start with `c:\` or `d:\`
- Indicates which drive

Mac

- Absolute paths start with `/`
- Example: `/Users/tyler/my-file.docx`
- Don't know which drive

How can we use multiple drives if every file paths starts the same???

`/.....`

Answer: different drives feel like different directories

Comparison

Windows	Mac	Drives
c:\Users\tyler\file.txt	/Users/tyler/file.txt	
c:\Program Files	/usr/local/bin	
c:\Windows\...\Logs	/var/log	
d:\	/Volumes/backup	
d:\A	/Volumes/backup/A	
e:\movies	/Volumes/movies	
e:\movies\demo1.mov	/Volumes/movies/demo1.mov	

Comparison

on a Mac, a path doesn't tell you what drive you're on

Windows

c:\Users\tyler\file.txt

c:\Program Files

c:\Windows\...\Logs

Mac

/Users/tyler/file.txt

/usr/local/bin

/var/log

Drives



d:\

d:\A

/Volumes/backup

/Volumes/backup/A



e:\movies

e:\movies\demo1.mov

/Volumes/movies

/Volumes/movies/demo1.mov



Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

- Navigational commands
- Arguments
- Saving output

Demos

We'll cover a few simple examples for reference in the slides, then go into more detail in the demo...

Most of these examples work in both **PowerShell** (Windows) and **bash** (Mac)

Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

- Navigational commands
- Arguments
- Saving output

Demos

Where am I? (What directory am I in?)

Command: **pwd**

```
PS /Users/trh/scratch>
```

Where am I? (What directory am I in?)

Command: **pwd**

“print working directory”

```
PS /Users/trh/scratch> pwd
```

Where am I? (What directory am I in?)

Command: **pwd**

```
PS /Users/trh/scratch> pwd
```

Path

```
/Users/trh/scratch
```

this is the current directory

```
PS /Users/trh/scratch>
```

Go up a directory

Command: **cd ..**

```
PS /Users/trh/scratch> pwd
```

```
Path
```

```
----
```

```
/Users/trh/scratch
```

```
PS /Users/trh/scratch>
```

Go up a directory

Command: **cd ..**

```
PS /Users/trh/scratch> pwd
```

```
Path
```

```
----
```

```
/Users/trh/scratch
```

```
PS /Users/trh/scratch> cd ..
```

Go up a directory

Command: **cd ..**

```
PS /Users/trh/scratch> pwd
```

```
Path
```

```
----
```

```
/Users/trh/scratch
```

```
PS /Users/trh/scratch> cd ..
```

```
PS /Users/trh>
```

Clear the screen

Command: **clear**

```
PS /Users/trh/scratch> pwd
```

```
Path
```

```
----
```

```
/Users/trh/scratch
```

```
PS /Users/trh/scratch> cd ..
```

```
PS /Users/trh> clear
```

Clear the screen

Command: **clear**

```
PS /Users/trh>
```

Go inside a directory

Command: **cd directory-name**

```
PS /Users/trh>
```

Go inside a directory

Command: **cd directory-name**

name of directory we started in

```
PS /Users/trh> cd scratch
```

Go inside a directory

Command: **cd directory-name**

```
PS /Users/trh> cd scratch  
PS /Users/trh/scratch>
```

Go to top directory

Command: **cd /**

is this Windows or Mac?

```
PS /Users/trh> cd scratch  
PS /Users/trh/scratch> cd /
```

Go to top directory

Command: **cd /**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS />
```

View contents of current directory

Command: **ls**

```
PS /Users/trh> cd scratch  
PS /Users/trh/scratch> cd /  
PS />
```

View contents of current directory

Command: **ls**

```
PS /Users/trh> cd scratch  
PS /Users/trh/scratch> cd /  
PS /> ls
```

View contents of current directory

Command: **ls**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS />
```

View contents of a file

Command: **cat file-name**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS />
```

View contents of a file

Command: **cat file-name**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS /> cat README.txt
```

View contents of a file

Command: **cat file-name**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS /> cat README.txt
The file says Hello!
PS />
```

View contents of a file

Command: **cat file-name**

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS /> cat README.txt
The file says Hello!
PS />
```

data saved in README.txt

Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

- Navigational commands
- Arguments
- Saving output

Demos

Arguments (program input)

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications          etc
Library               home
Network               installer.failurerequests
System                net
Users                 README.txt
PS /> cat README.txt
The file says Hello!

PS />
```

Arguments (program input)

```
PS /Users/trh> cd scratch  
PS /Users/trh/scratch> cd /
```

```
PS /> ls
```

```
Applications  
Library
```

```
etc
```

```
home
```

```
Notesbooks
```

```
Desktop Shared Folders Requests
```

program name (cat)

Users

```
PS /> cat README.txt
```

```
The file says Hello!
```

an argument (README.txt)

README.txt

```
PS />
```

echo Example

```
PS /Users/trh>
```

echo Example

```
PS /Users/trh> echo hello
```

echo Example

program is “echo”

argument is “hello”

```
PS /Users/trh> echo hello
```

echo Example

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh>
```

echo Example

```
PS /Users/trh> echo hello  
hello  
PS /User
```

the echo program prints
whatever it's argument is

Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

- Navigational commands
- Arguments
- Saving output

Demos

Saving output

Format: **program > file-name**

```
PS /Users/trh>
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh>
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt
```

“redirect” operator, sends output to a file

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt  
PS /Users/trh>
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt  
PS /Users/trh>
```

without redirect, output
was printed to the screen

with redirect, output was
saved in the output.txt file

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt  
PS /Users/trh>
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt  
PS /Users/trh> cat output.txt
```

Saving output

Format: **program > file-name**

```
PS /Users/trh> echo hello  
hello  
PS /Users/trh> echo hello > output.txt  
PS /Users/trh> cat output.txt  
hello  
PS /Users/trh>
```

Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

Demos

Conclusion

Today we covered

- What a terminal and shell is
- What it looks like to have multiple storage drives attached to your computer
- How to navigate between directories/folders
- How to run programs in the terminal