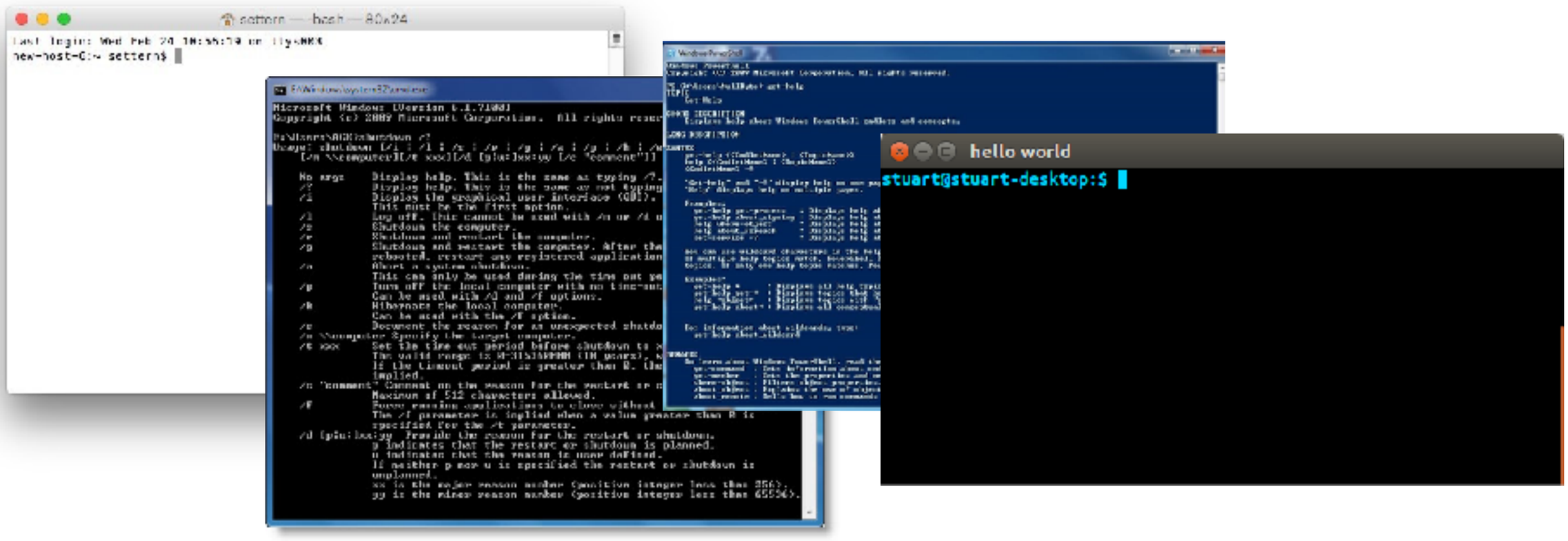
[301] The Terminal

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)

https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

History: the Original Terminals

Mainframe
(powerful computer)

How to share it?

https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

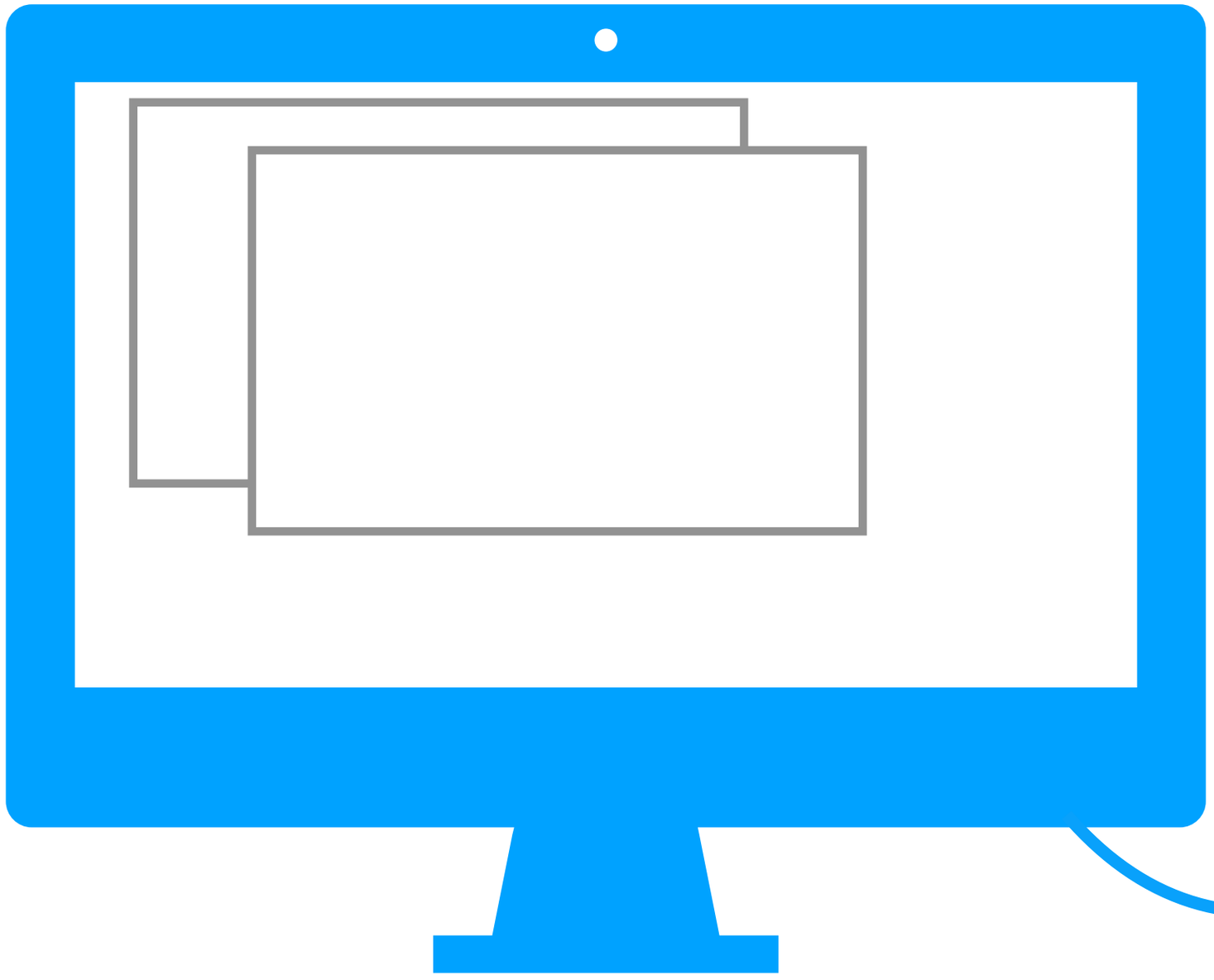
History: the Original Terminals

Mainframe
(powerful computer)

dumb terminals
(text based)

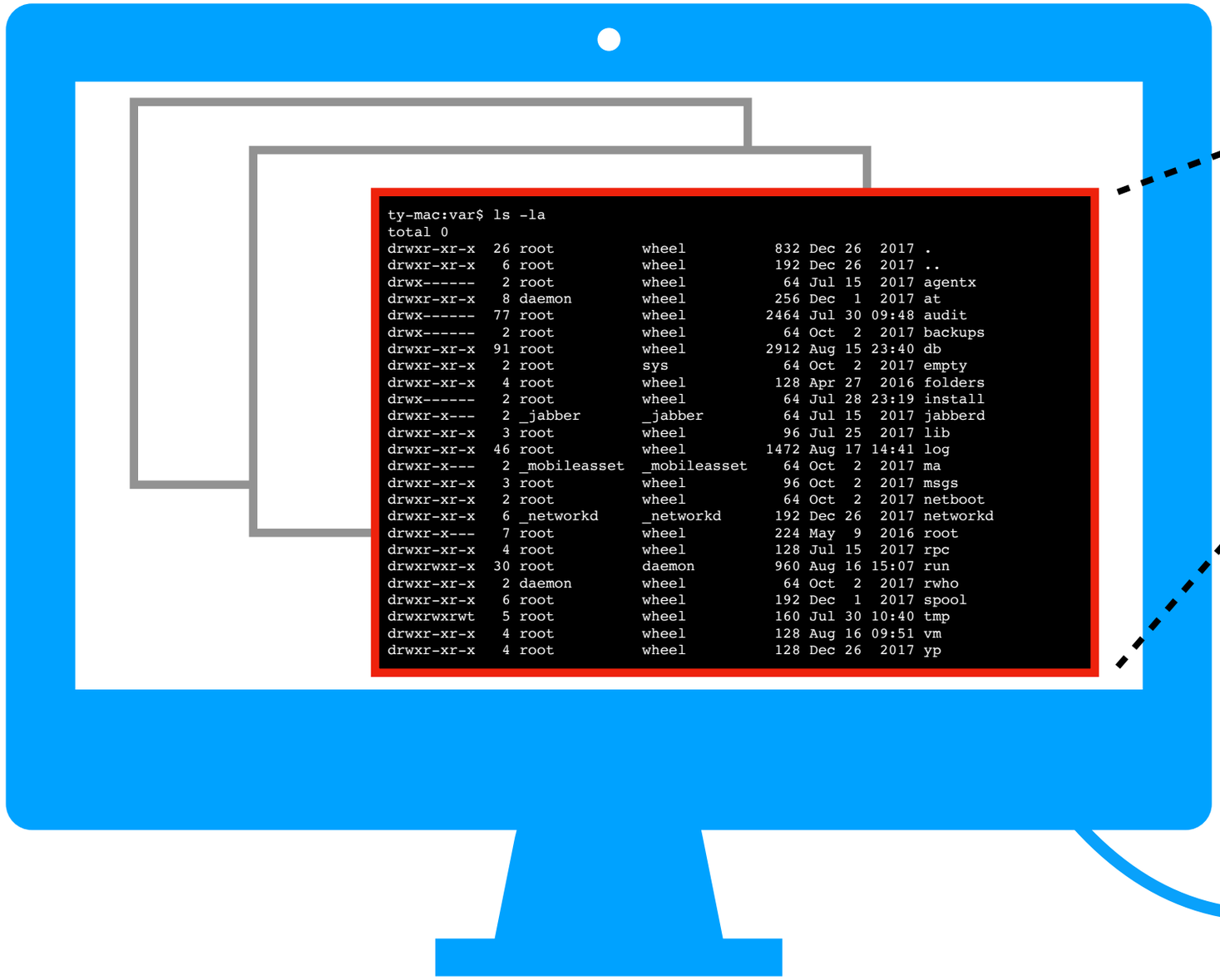
https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

Terminal Emulators



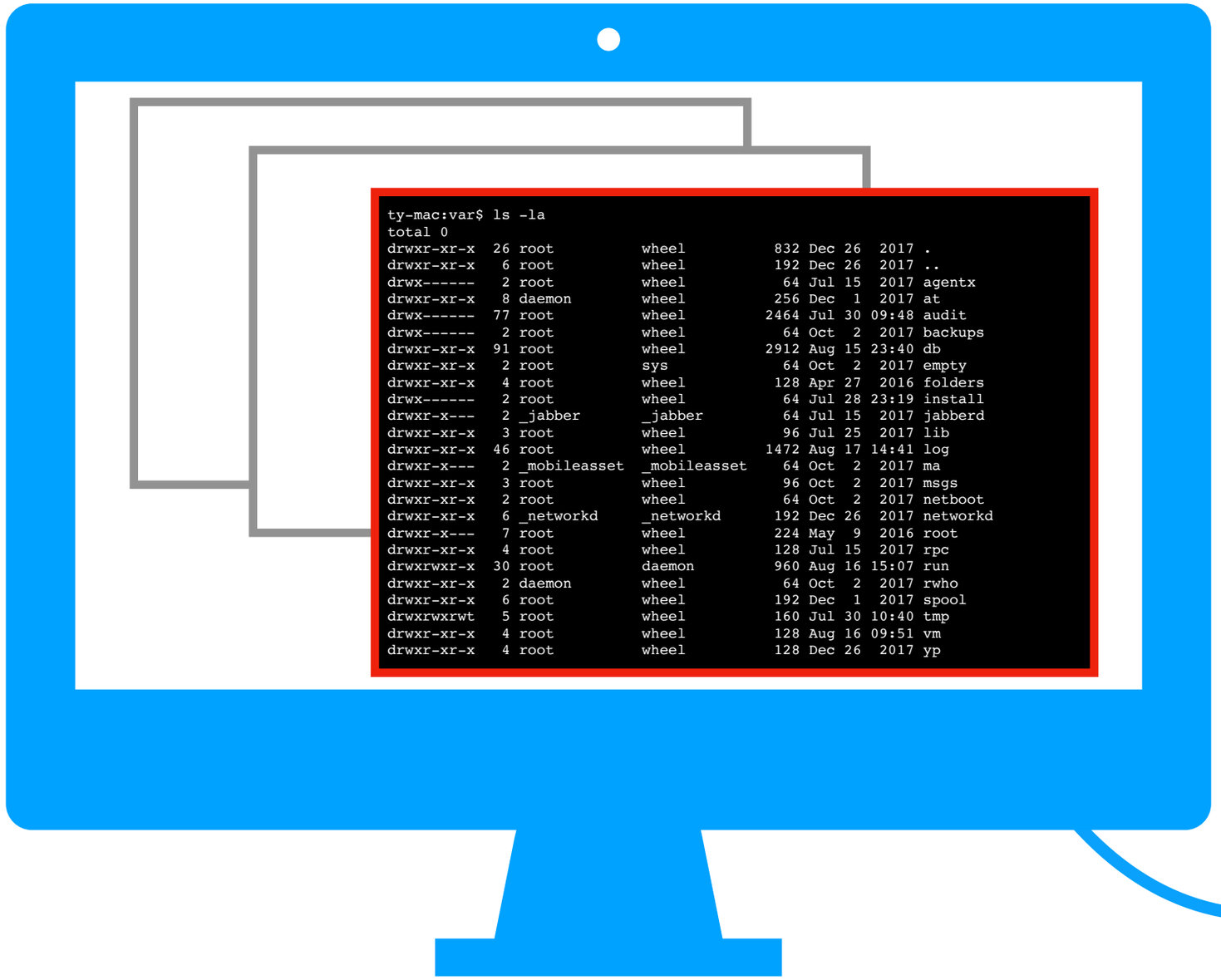
https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

Terminal Emulators



https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

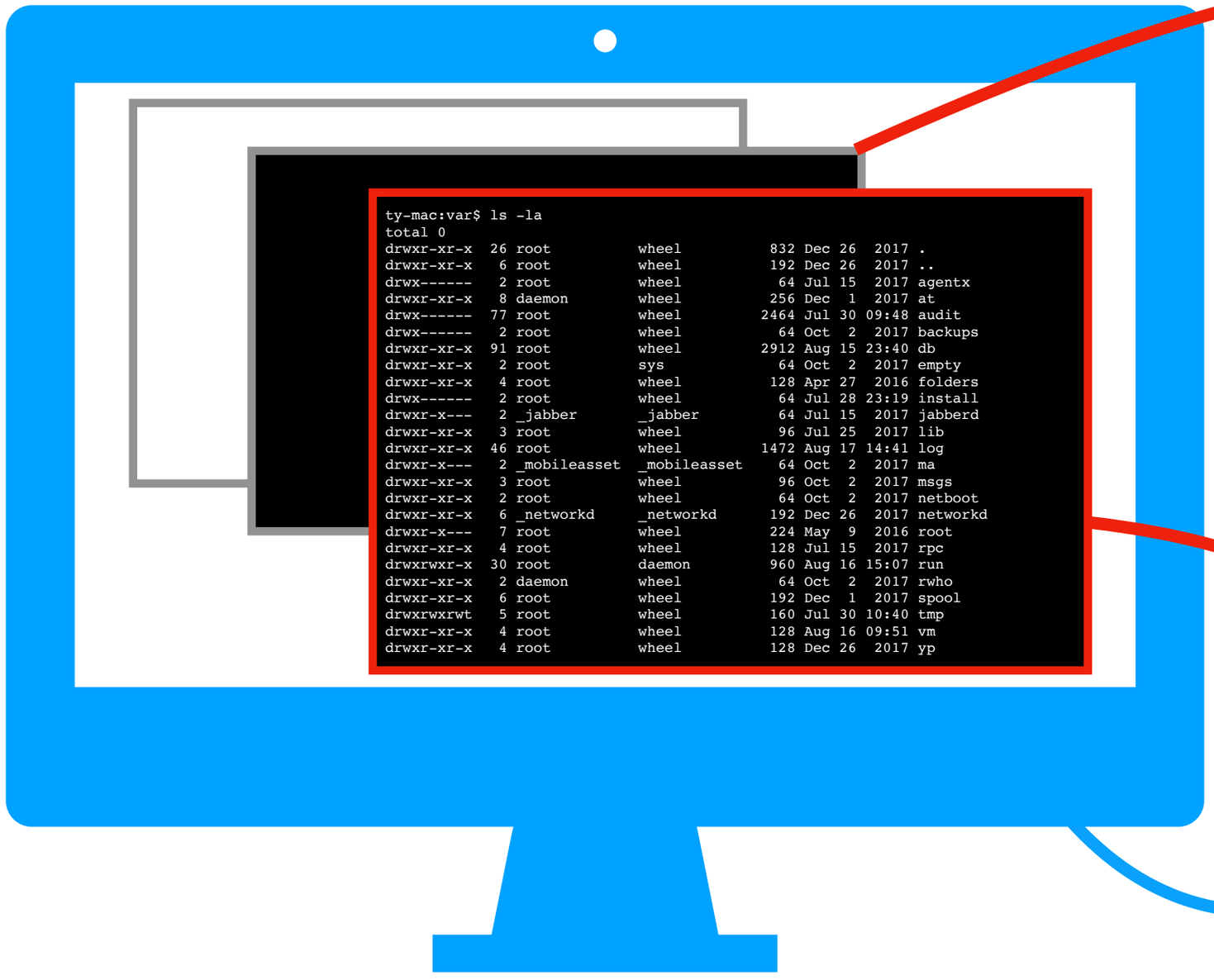
Terminal Emulators



local computer
(e.g., personal)

https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

Terminal Emulators



remote computer
(e.g., CS lab)

OR

local computer
(e.g., personal)

https://techcrunch.com/2015/08/16/ibm-teams-with-canonical-on-linux-mainframe/
https://en.wikipedia.org/wiki/Computer_terminal#Dumb_terminals

Today's Topics

Terminal Emulators and Shells

- Terminal history
- Shells
- Running programs from a shell

Navigation

Running Programs and Commands

Demos

Shells

Inside a terminal, a program called a “shell” runs

- The shell lets users type commands, then carries out the appropriate actions
- Exploring files and running programs are common activities
- You will be running Python programs from a shell in a terminal!

Shells

Inside a terminal, a program called a “shell” runs

- The shell lets users type commands, then carries out the appropriate actions
- Exploring files and running programs are common activities
- You will be running Python programs from a shell in a terminal!
- Different shells have minor (or major) variations

Windows Shells

- cmd
- PowerShell

type “dir” to view files

type “ls” (for list) to view files

Shells

Inside a terminal, a program called a “shell” runs

- The shell lets users type commands, then carries out the appropriate actions
- Exploring files and running programs are common activities
- You will be running Python programs from a shell in a terminal!
- Different shells have minor (or major) variations

Windows Shells

- cmd
- PowerShell

UNIX Shells

- bash
- csh
- zsh
- many more

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

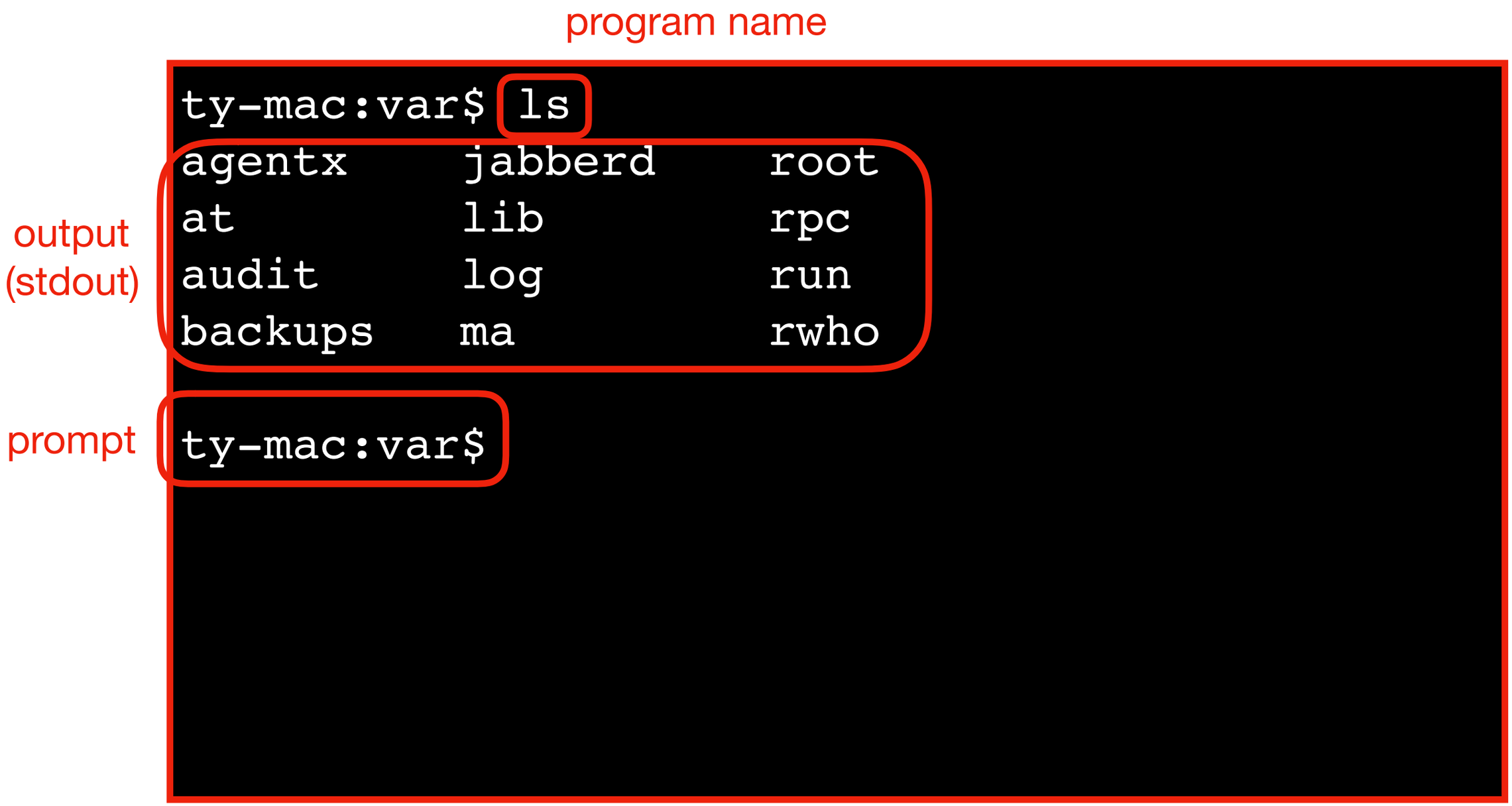
Running programs is easy, just type name of the program and hit enter:

```
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:



Today's Topics

Terminal Emulators and Shells

Navigation

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

Running Programs and Commands

Demos

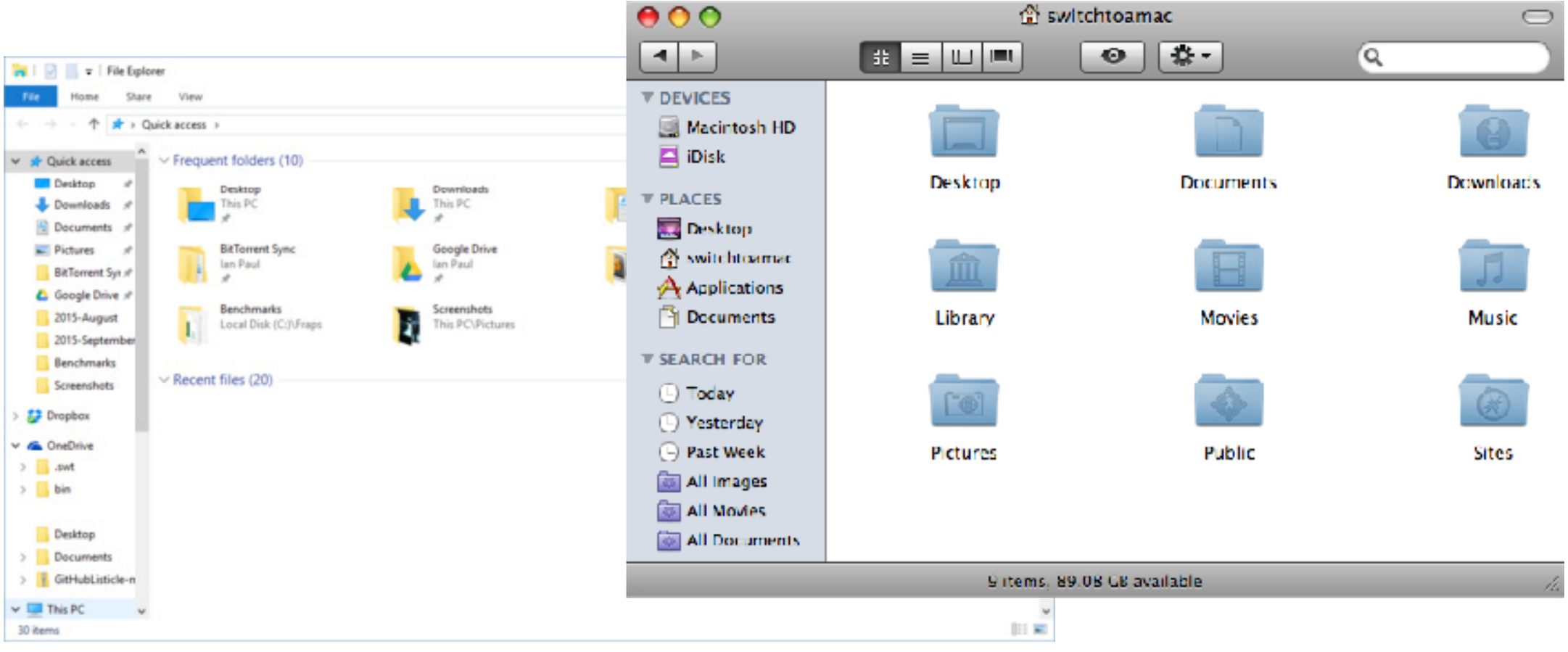
What is navigation?

Navigation is looking around for files/folders you want

- Enter a folder, go up, search, etc

Common navigation programs

- File Explorer (Windows)
- Finder (Mac)



What is navigation?

Navigation is looking around for files/folders you want

- Enter a folder, go up, search, etc

Common navigation programs

- File Explorer (Windows)
- Finder (Mac)

In the shell, you navigate by typing various commands

Today's Topics

Terminal Emulators and Shells

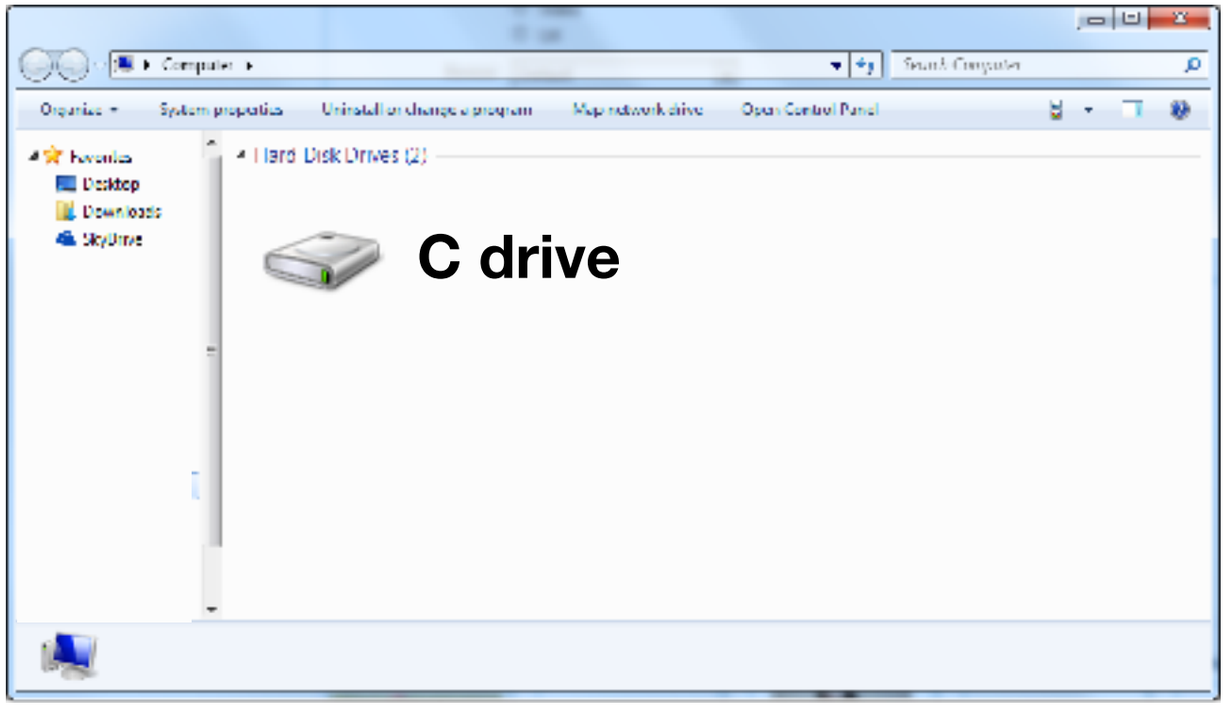
Navigation

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

Running Programs and Commands

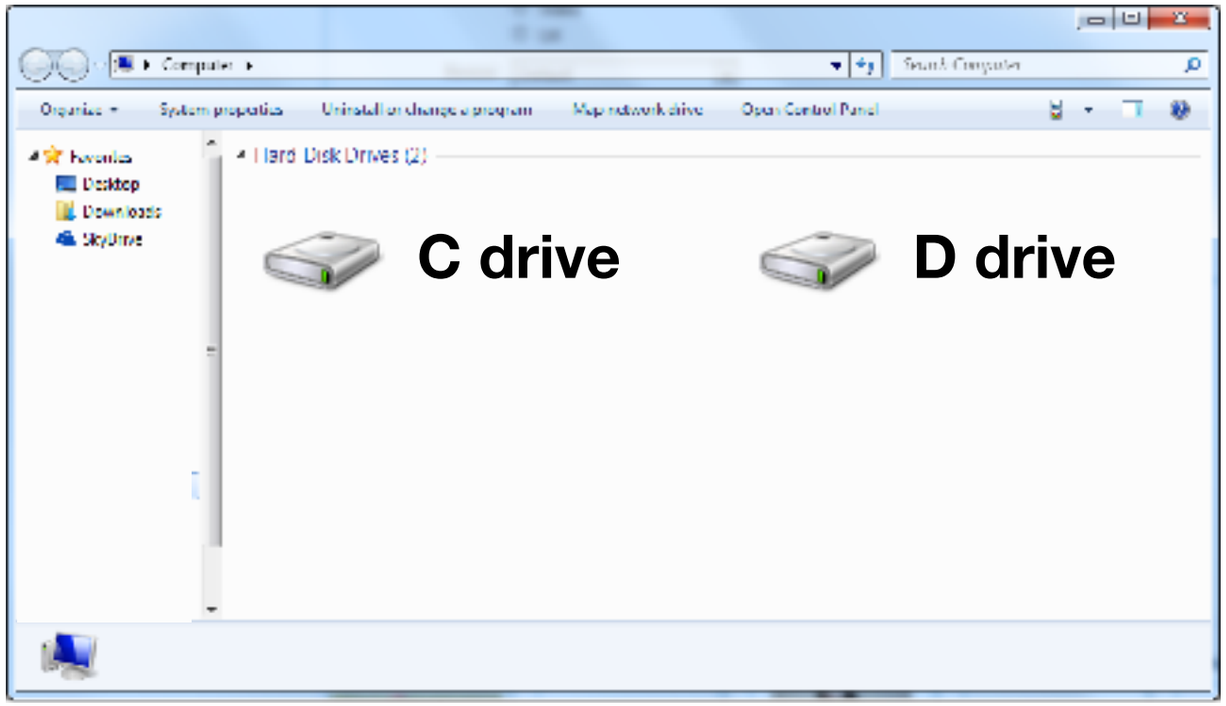
Demos

Windows Storage Drives



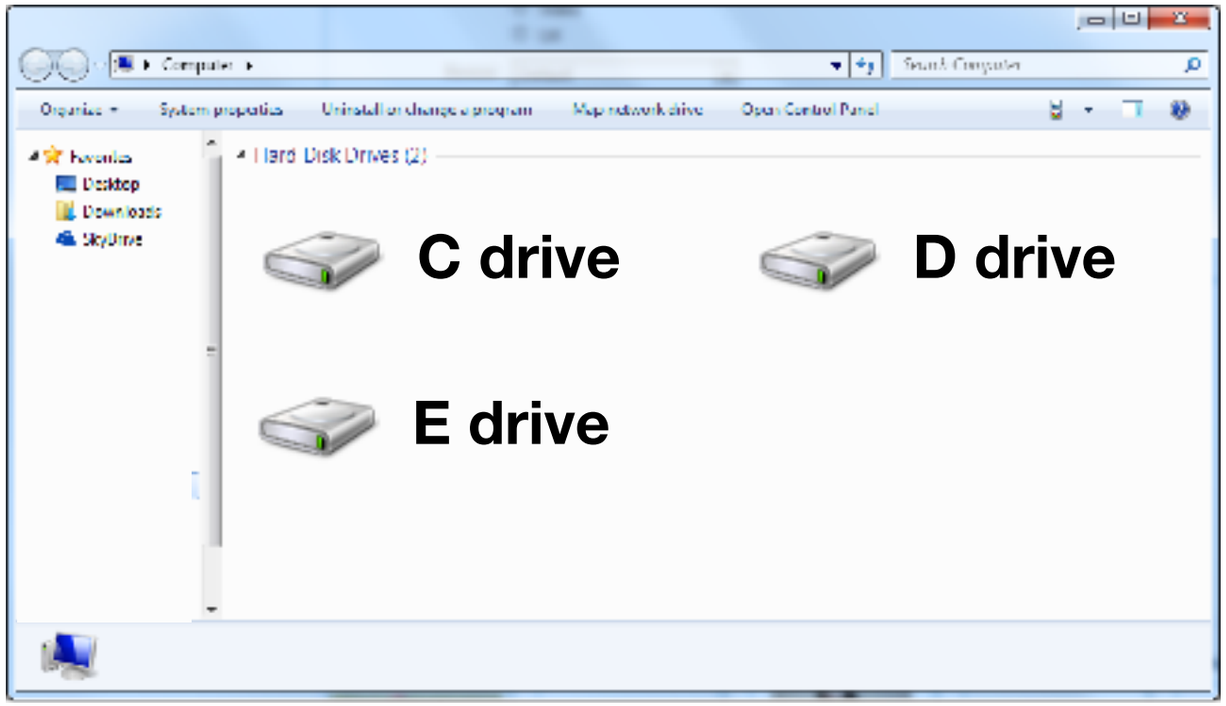
Each added drive is given its own drive letter

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. UNIX

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”

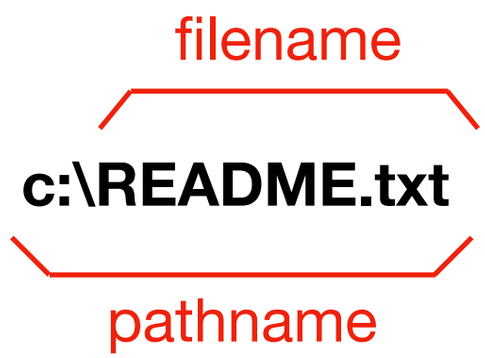
c:\hw.docx

d:\page.html

e:\main.py

Files

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



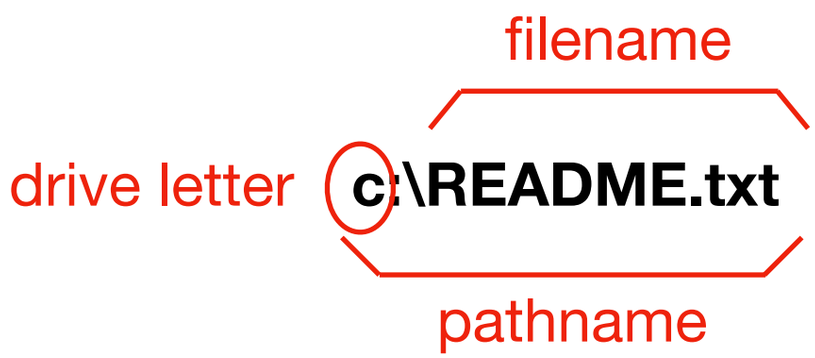
c:\hw.docx

d:\page.html

e:\main.py

Files

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



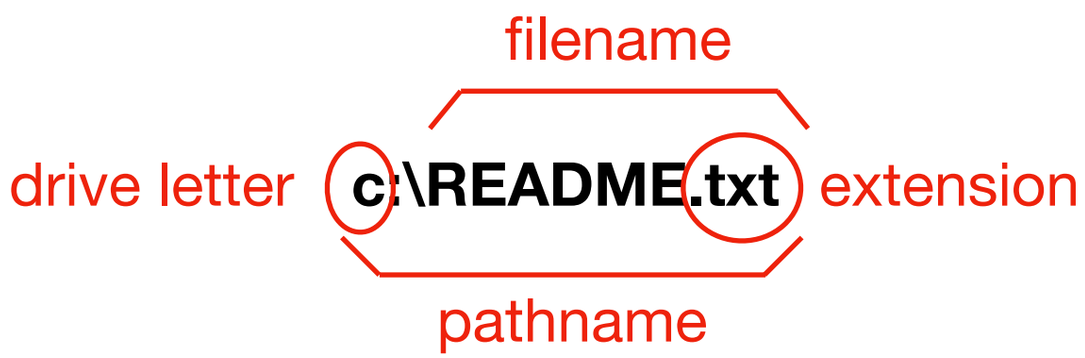
c:\hw.docx

d:\page.html

e:\main.py

Files

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



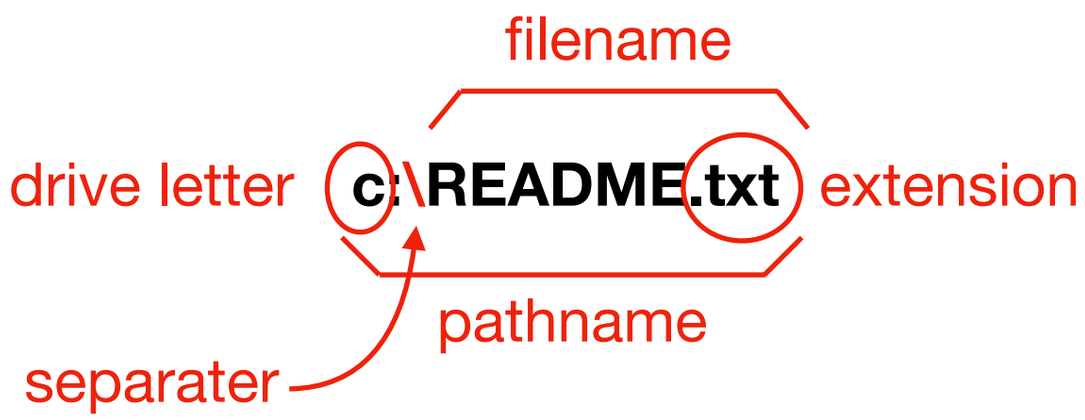
c:\hw.docx

d:\page.html

e:\main.py

Files

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



c:\hw.docx

d:\page.html

e:\main.py

Files

Files are sources of input and destinations for output for processes.

Files are managed by a part of the operating system called the “file system”

Today's Topics

Terminal Emulators and Shells

Navigation

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

Running Programs and Commands

Demos

Directories

Directories are used to organize files

- Also called “folders”
- A directory also has pathname
- Each directory may contain other directories and files

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

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

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
- Current location/directory is called "working directory" or "current working 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	
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:\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:\x\y\z	c:\x	.\.\test.txt
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:\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:\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 tutorial later…

Today's Topics

Terminal Emulators and Shells

Navigation

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

Running Programs and Commands

Demos

Multiple Drives in Linux

Windows

- Generally, every absolute pathname starts with “c:\” or “d:\” or similar
- Name indicates which drive stores the file

Multiple Drives in Linux

Windows

- Generally, every absolute pathname starts with “c:\” or “d:\” or similar
- Name indicates which drive stores the file

UNIX

- Every absolute pathname starts with “/”
- For example, /home/tyler/my-file.docx (note forward slash)
- Name does not indicate on which drive a file lives

Multiple Drives in Linux

Windows

- Generally, every absolute pathname starts with “c:\” or “d:\” or similar
- Name indicates which drive stores the file

UNIX

- Every absolute pathname starts with “/”
- For example, /home/tyler/my-file.docx (note forward slash)
- Name does not indicate on which drive a file lives

How can we use multiple drives if every file paths starts the same, with “/” ???

Multiple Drives in Linux

Windows

- Generally, every absolute pathname starts with “c:\” or “d:\” or similar
- Name indicates which drive stores the file

UNIX

- Every absolute pathname starts with “/”
- For example, /home/tyler/my-file.docx (note forward slash)
- Name does not indicate on which drive a file lives

How can we use multiple drives if every file paths starts the same, with “/” ???

Answer: different drives feel like different directories in UNIX systems

Comparison

Windows	Mac	Linux	Drives
c:\Users\tyler\file.txt	/Users/tyler	/home/tyler	
c:\Program Files	/usr/local/bin	/usr/local/bin	
c:\Windows\…\Logs	/var/log	/var/log	
d:\	/Volumes	/mnt/backup	
d:\aug	/Volumes/backup/aug	/mnt/backup/aug	
e:\movies	/Volumes/movies	/home/tyler/movies	

Comparison

Windows	Mac	Linux	Drives
c:\Users\tyler\file.txt	/Users/tyler	/home/tyler	
c:\Program Files	/usr/local/bin	/usr/local/bin	
c:\Windows\…\Logs	/var/log	/var/log	
d:\	/Volumes	/mnt/backup	
d:\aug	/Volumes/backup/aug	/mnt/backup/aug	
e:\movies	/Volumes/movies	/home/tyler/movies	

On Mac, extra drives often appear under /Volumes.
On Linux, extra drives often appear under /mnt (for mount).

Comparison

Windows	Mac	Linux	Drives
c:\Users\tyler\file.txt	/Users/tyler	/home/tyler	
c:\Program Files	/usr/local/bin	/usr/local/bin	
c:			
d:			
d:\aug	/Volumes/backup/aug	/mnt/backup/aug	
e:\movies	/Volumes/movies	/home/tyler/movies	

Unlike in Windows, on UNIX systems, you can’t tell what drive a file is on, just by looking at the path name

On Mac, extra drives often appear under /Volumes.
On Linux, extra drives often appear under /mnt (for mount).

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…

Today's Topics

Terminal Emulators and Shells

Navigation

Running Programs and Commands

- Navigational commands
- Arguments
- Saving output

Demos

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

Command: **`pwd`**

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

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

Command: **`pwd`**

"print working directory"

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

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

Command: **`pwd`**

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

Path
----
/Users/trh/scratch


PS /Users/trh/scratch>
```

this is the current directory

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 /`**

```
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

```
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

```
PS /Users/trh> cd scratch
PS /Users/trh/scratch> cd /
PS /> ls
Applications        etc
Library             home
                    rerequests
Users               README.txt
PS /> cat README.txt
The file says Hello!

PS />
```

program name (cat)

an argument (README.txt)

echo Example

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

echo Example

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

echo Example

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
```

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