Commanding the Command Line

“You’re never a loser until you quit trying"
-Mike Ditka

Cybersecurity
Terminal 101 Day 2

Class Objectives

By the end of class today, students will be able to:

- ☐ Identify and explain the elements of basic Unix command
- ☐ Use the `find` command to locate files based on various search parameters
- ☐ Use the `exec` command to perform bulk operations on files.
- ☐ Use the `grep` command to search within the contents of files
- ☐ Use the `wc` command to count words and lines
- ☐ Devise strategies for combining multiple commands in sequence to accomplish intermediate IT tasks.

Warm-Up Activity

In this activity, you will review the lessons of last class by working on the command line.

Instructions sent via Slack

Suggested Time:
7 Minutes

Your Turn: Warm-Up

Instructions:

You've just been given a series of server logs. Your task is to use the command line to:

a. Create a folder called Archive.
b. Combine all the logs contained in the TodaysLog folder into a text file called 09_15_18.txt.
c. Move the file you created to the Archive folder.
d. Preview the file contents.

7 Minutes

Time’s Up! Let’s Review.

Warm-Up Activity

Command Line Structure

Command Line Structure

Large commands may seem intimidating to newcomers...

```
find . -exec grep -l -e 'myregex' {} \; >> outfile.txt

find "$DIR" -type f -atime +5 -exec rm {} \;

find .  -type f -name "* *" | while read file
```

But every command follows a relatively consistent structure:

```
command [-options][arguments]
```

Command Line Structure - command

`command [-options][arguments]`

- Programs that tell Unix system to do something
- Case sensitive
- Examples: `cd`, `mkdir`, `find`, `exec`, `tar`

Command Line Structure - options

`command [-options][arguments]`

- Options modify what the command will do.
- Not always a requirement.
- Preceded by a hyped.
- Examples: `-f, -size, -cmin`

Command Line Structure - arguments

`command [-options][arguments]`

- Usually indicates on what file or folder the command will act
- Can also be used to specify a parameter needed by the option.
- Appear immediately after the command if they are inputs for the command or after an option if they are an input for the option.
- Examples: `My_Folder, 10b, *.docx`

Command Line Structure Example #1

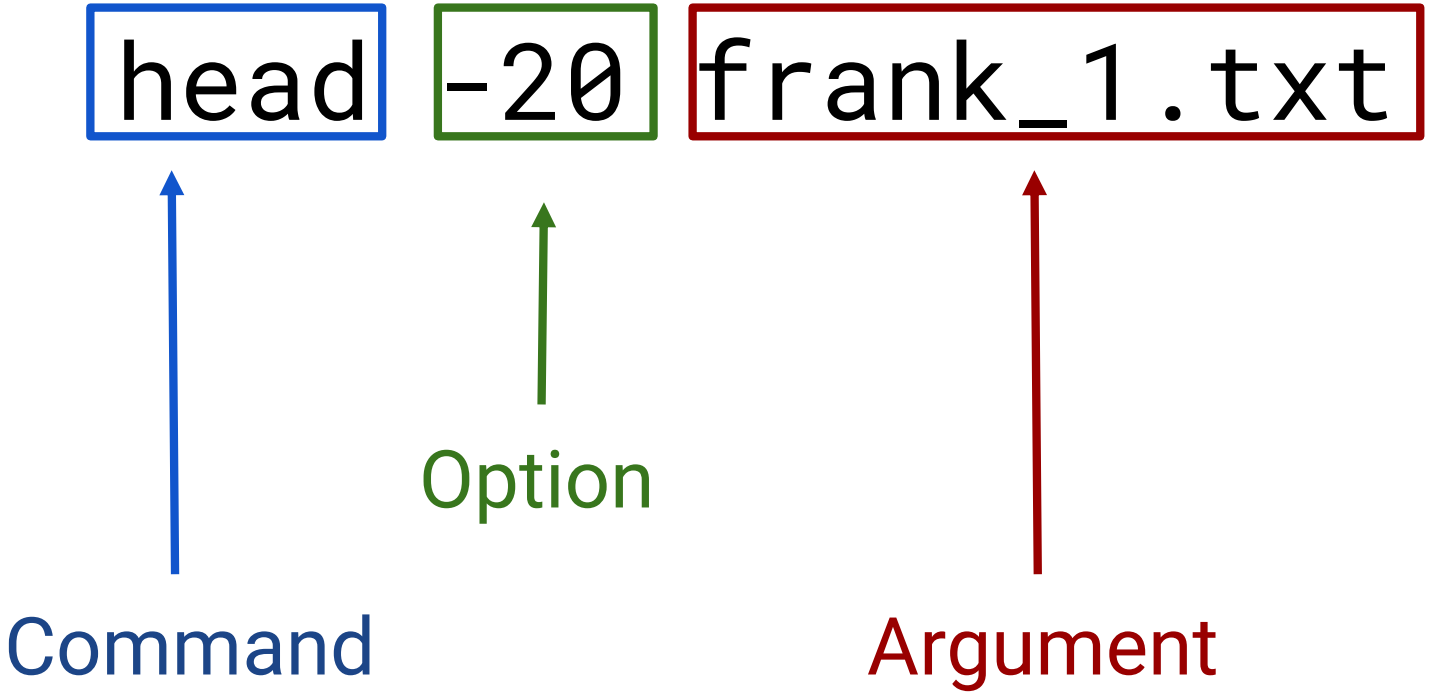
```
head -20 frank_1.txt
```

What is the command?

What is the argument?

What is the option?

Command Line Structure Example #1



Command Line Structure Example #1

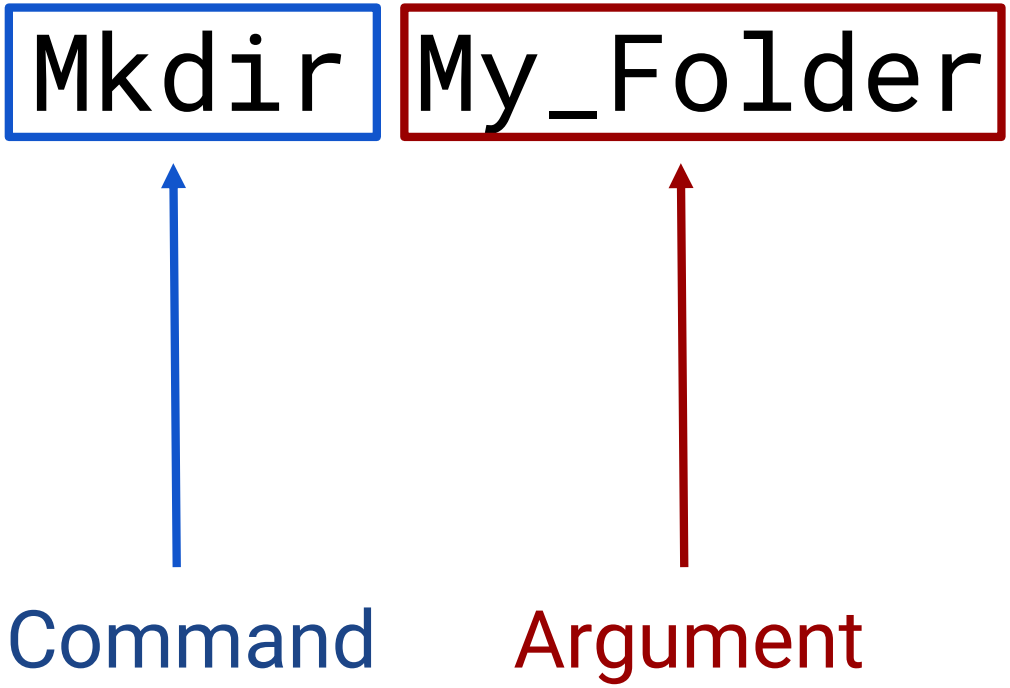
```
Mkdir My_Folder
```

What is the command?

What is the argument?

What is the option?

Command Line Structure Example #2



Command Line Structure Example #1

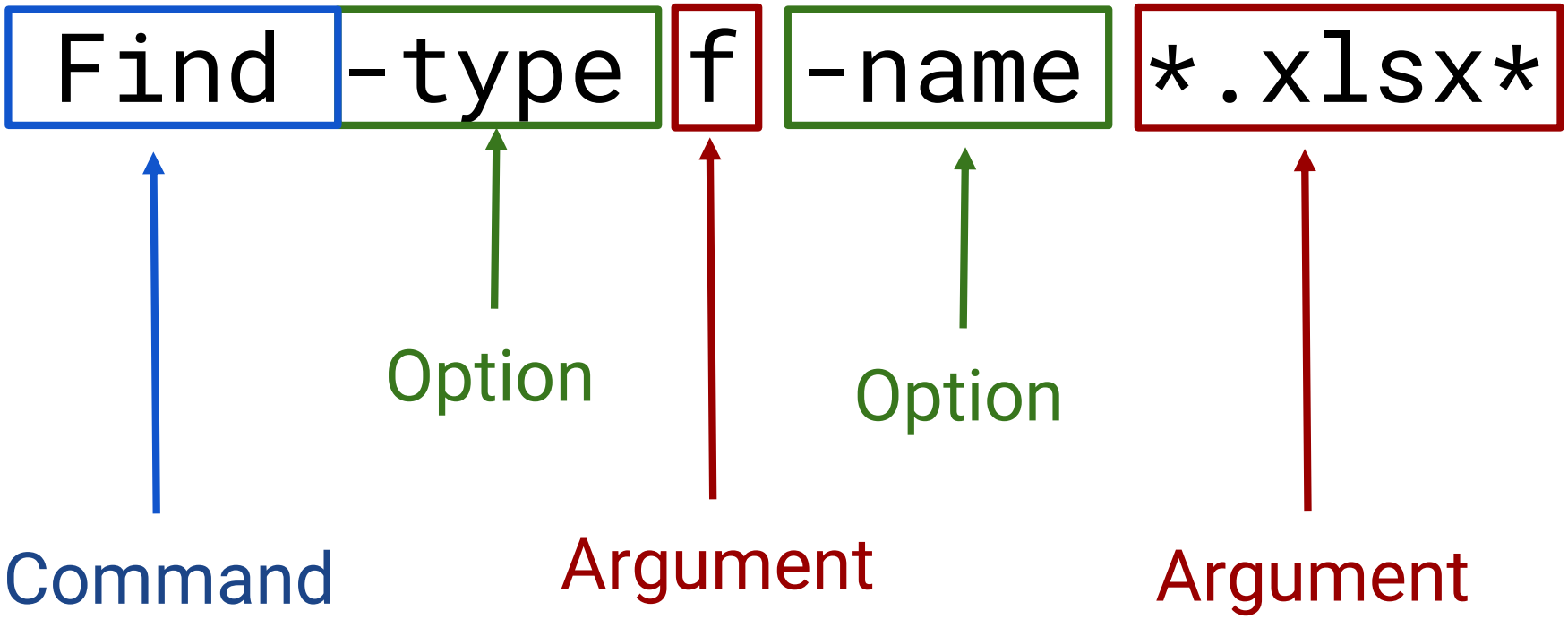
```
Find -type f -name *.xlsx*
```

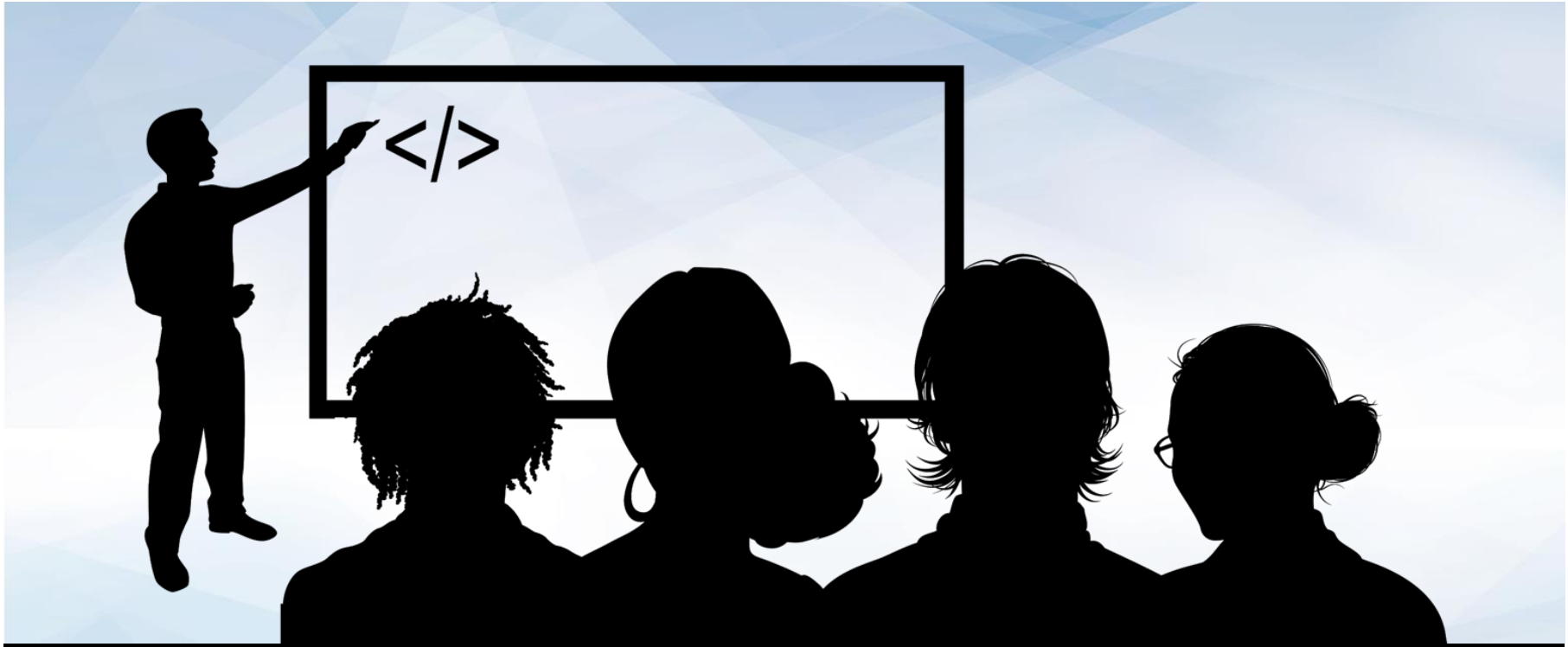
What is the command?

What is the argument?

What is the option?

Command Line Structure Example #3





Instructor Demonstration

find Command

Activity: PathFinder

In this activity, you'll revisit an updated version of the Terminal Maze. This time you'll use the find command to expedite the process.

Instructions sent via Slack

Suggested Time:
12 Minutes

Your Turn: PathFinder

Instructions:

1. Use the find command from within the Maze_1, Maze_2, and Maze_3 folders to identify the location of the start.txt and Bonus.txt files.
2. Use the find command once more to find the End directory folder that is hidden deep within the folder mazes.
3. Use the provided path results to help you copy each of the start.txt and Bonus.txt files into their respective Endfolder.

Hints:

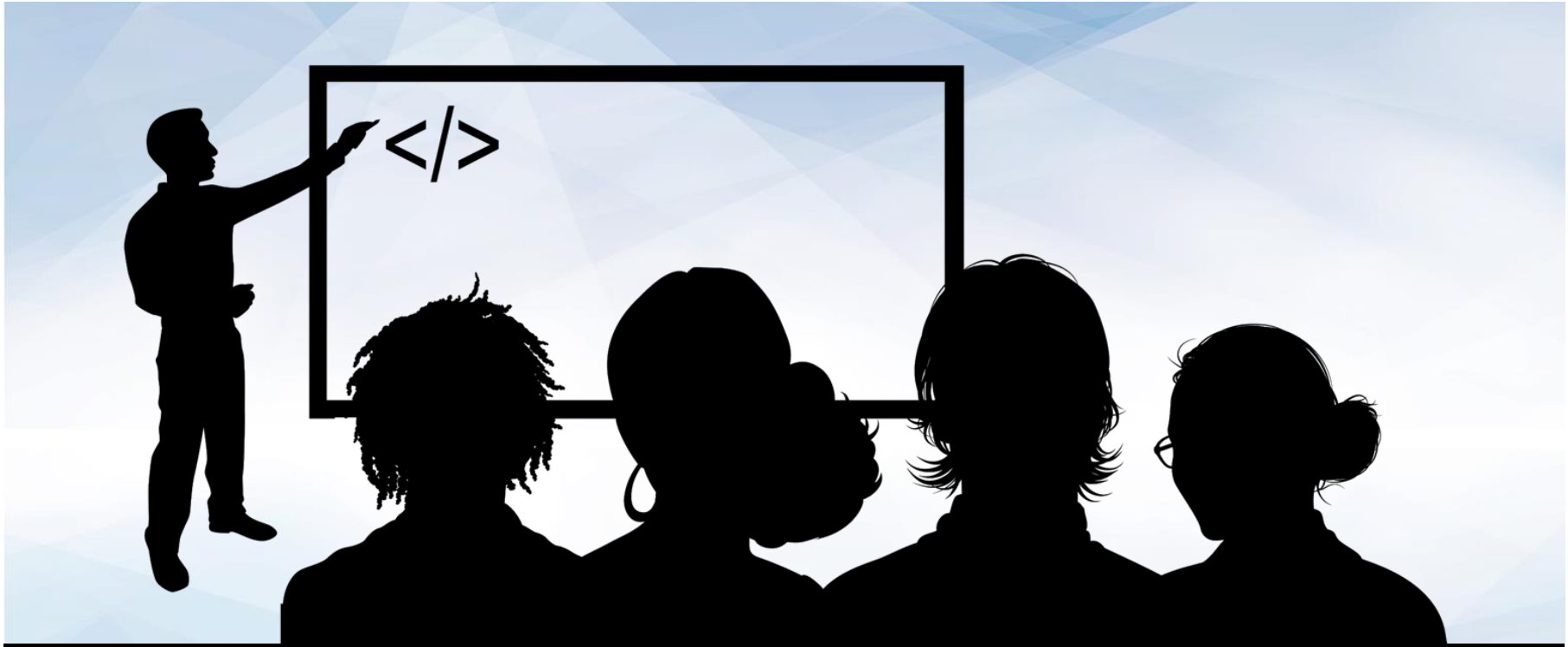
You should be able to complete this task in 9–10 commands (including the Bonus.txt files).

When typing in the paths, try hitting the tab key. What happens? This should speed up the coding for this activity just a little bit.

15 Minutes

Time’s Up! Let’s Review.

Pathfinder



Instructor Demonstration

find Command Options

find Command Options

Using the find command, we can

- Use `find {path}` construct to specify specific folders
- Use the `-iname` option to create case-insensitive file or folder searches
- Use the `-o (or)` and `-a (and)` options to combine search parameters
- Use the `* (wildcard)` to search for all files of a certain type
- Use the `-ctime` option to specify creation date before, after or between times.
- Use the `-size` option to specify file sizes

Activity: Gibberish Finder

In this activity, you will use find options to retrieve files based on specific search parameters.

Instructions sent via Slack

Suggested Time:
15 Minutes

Your Turn: Terminal Maze

Instructions:

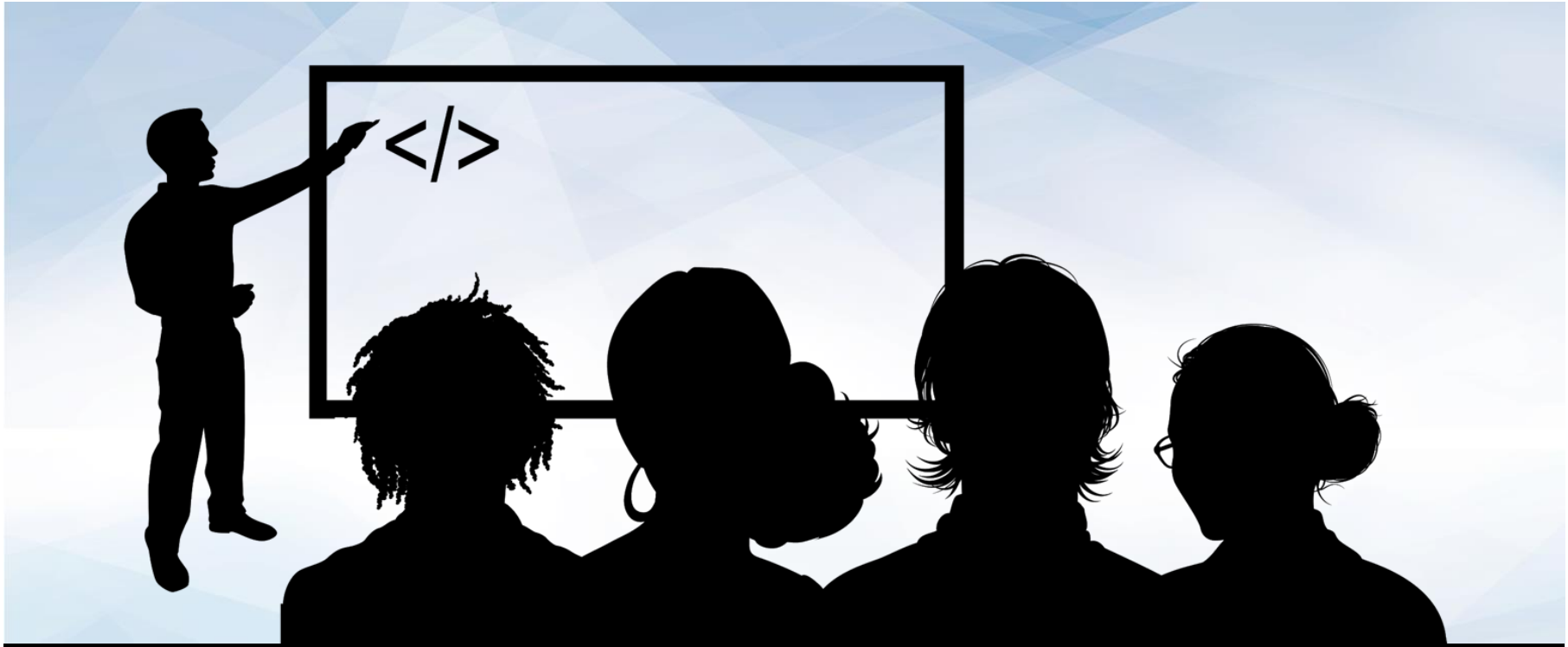
Part 1: Simple Search	Part II: Title Search	Part III: File Type Search	Part IV: Combo Search	Part V: Challenge Search	Part VI: Partner Search
Find a listing of all files. Find a listing of all folders.	Find all files containing the letter “z”. Find all files containing the the word “rice”.	Find all Excel files. Find all Word files.	Find all text files containing the number 2. Find all word files at least 200 KB in size.	Find all Excel files, at most 15 KB in size, and max of 3 levels of depth from the surface. Find all files that are not writable in **Gibberish_Folder/folder3**	If you finish early, Create a file and modify the contents of a different file deep in the Gibberish folder. Then see if your partner can find them using the find command.

15 Minutes

Time’s Up! Let’s Review.

Gibberish Finder

Take a Break!



Instructor Demonstration

exec Command

Activity: Executive Cleaning

In this activity, you will use the find and exec to sort the Gibberish_Folder.

Instructions sent via Slack

Suggested Time:
10 Minutes

Your Turn: Terminal Maze

Instructions:

You've been tasked with sorting the `Gibberish_Folder`.

1. Create a new folder called `Sorted_Gibberish`.
2. Crete three new subfolders within `Sorted_Gibberish` called `Docs`, `Data`, and `Text`.
3. Copy all the word documents into the `Docs` folder, all the Excel files into the `Data` folder, and all the text files into the `Text` folder.

Bonus:

- Create a subfolder called `LargeFiles` in the `Sorted_Gibberish` folder.
- Move all files that are larger than 200 KB into the `LargeFiles` folder.

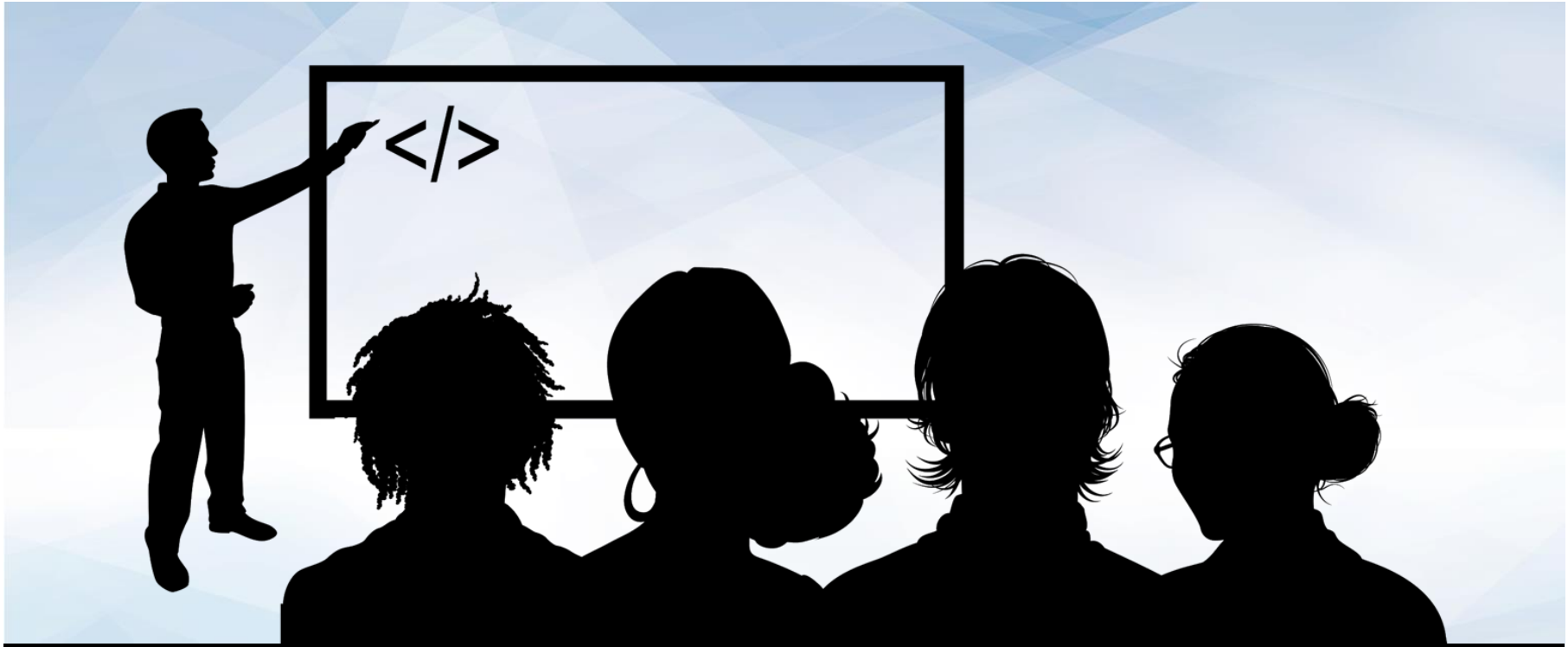
15 Minutes

Time’s Up! Let’s Review.

Preview Practice

-exec Command

- `find -type f -iname *{file type}*` signals that we are searching for one file type.
- `-exec cp {} {destination}` signifies that we would like to move these files to our new destination
- `\;` signifies that we are concluding out statement



Instructor Demonstration

grep Command

`grep` Command Options

Using the `grep` command, we can:

- Use `grep` command to search within the body of file text.
- The basic construction is: `grep` (command) `{text}{location}`.
- Use the `-i (or)` option to signify a case-insensitive search.
- Use the `-iv` to signify a search for all files not including the specified term.
- Use the `grep -rl {text}{location}` to search for a list of files that include the specified text.
- `grep {text}` outputs the lines of text for which the text appears.

Activity: grep Detective

In this activity, you will use the grep command to identify the power users of a provide chat log, as well as when these users logged on and off.

Instructions sent via Slack

Suggested Time:
10 Minutes

Your Turn: grep Detective

Instructions:

You've just been given a series of chat logs from April 2014. Use the grep command to identify the following:

1. The days for which users power2all, glanzmann, gansbrest, and E1ven were active in the channel.
2. The log-on and log-off times for these users.

Hint: When determining log-on time, look for a string pattern that captures specifically what you're looking for.

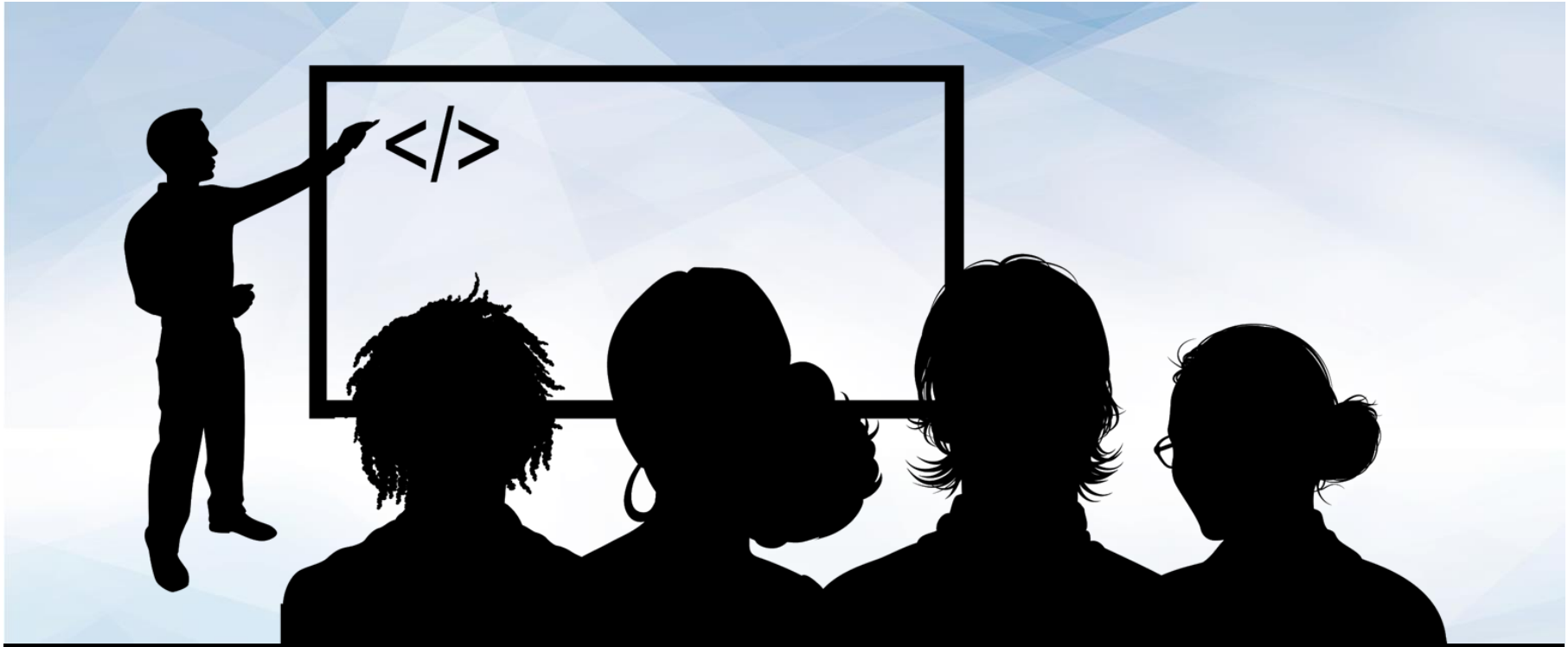
Bonus:

- Create a folder for each of the users of interest.
- Create a combination of `find`, `exec`, and `grep` that allows you to retrieve all logs for which these users were active and immediately copy these files into the respecif folder.

10 Minutes

Time’s Up! Let’s Review.

grep Detective



Instructor Demonstration

wc Command

wc Command

- `|` conveys that we are piping the results from our find and grep commands into the next command.
- `wc -l` conveys that we are looking to count the number of lines retrieved.
- `|wc -l` in conjunction with `find` and `grep` retrieves the record count.

Activity: Log Counter Activity

In this activity, you will expand on their investigative work from the previous IRc example to count the relative activity levels of various users.

Instructions sent via Slack

Suggested Time:
10 Minutes

Your Turn: Log Counter

Instructions:

Part 1: Basic Counts

1. How many log files are included in the IRC_Logs folder?

2. How many log files exceed 100KB in size.

Part 2: Login Counter

3. How many times did the user glanzmann log on?

4. How many times did user E1ven log on?

Part 3: Chat Counter

5. How many times did the user glanzmann speak?

6. How many times did the user E1ven speak?

10 Minutes

Time’s Up! Let’s Review.

Log Counter

Lesson Recap

What are the basic components of a command-line statement?

What are the basic components of a command-line statement?

```
command [-options][arguments]
```

Which command would we use to identify all files in the current directory?

Which command would we use to identify all files in the current directory?

```
find . -type f
```

Which command would we use to identify all directories in the current directory?

Which command would we use to identify all directories in the current directory?

```
find . -type d
```

Which command would we use to find all text files in a folder named manuals?

Which command would we use to find all text files in a folder named manuals?

```
find *.txt -type f manuals
```

Which command would we use to find all .docx files and .pdf files?

Which command would we use to find all .docx files and .pdf files?

```
find *.docx -o *.pdf -type f .
```

What does each element of the following statement signify?

```
find . -type f -exec cp {} My_Files \;
```

What does each element of the following statement signify?

```
find . -type f -exec cp {} My_Files \;
```

- `find . -type f` — Specifies the search condition
- `-exec` — Run subcommand
- `cp {}` — Copy file list
- `My_Files` — Destination
- `\;` — Command close

Which command would we use to search within files for instances of a certain text?

Which command would we use to search within files for instances of a certain text?

```
grep {text} .
```

Which command would we use to search within files for files containing a certain text?

Which command would we use to search within files for files containing a certain text?

```
grep -rli {text} .
```

Which command would we use to count the number of search results?

Which command would we use to count the number of search results?

```
{query} | wc -l
```

Today's Summary

`Find {condition} {location}` finds all files based on provided conditions

- `-type f / -type d`: Searches for files or directories.
- `-name / -iname`: searches file titles
- `-size`: searches file sizes
- `-cmin / min:` searches creation or modified date.

`exec {condition}{}{destination} \;` performs a bulk operation on multiple files

`grep` searches within the body of files for instances of text

- grep -i {text}{location}: Searches for instances of text.
- grep -rli {text}: Searches for files containing text

`| wc -l` counts lines. Useful in counting the number of records

Class Objectives

By the end of class today, students will be able to:

- Identify and explain the elements of basic Unix command
- Use the `find` command to locate files based on various search parameters
- Use the `exec` command to perform bulk operations on files.
- Use the `grep` command to search within the contents of files
- Use the `wc` command to count words and lines
- Devise strategies for combining multiple commands in sequence to accomplish intermediate IT tasks.