

# FinTech Collaboration

FinTech  
Lesson 1.3



# WELCOME



# Class Objectives

---

By the end of today's class, you will:



Configure the git CLI user credentials from the command line.



Clone a repository using `git clone`.



Modify git repositories by adding, committing, and pushing files.



Create markdown files and implement visual capabilities such as text formatting, images, and links.



Write a FinTech case study in markdown and host it in a shared GitHub repository.

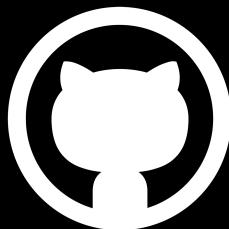
# GitHub Refresher



# GitHub Q&A



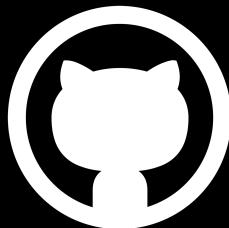
# What is git?



**Answer:** Git is a version-control system for tracking changes in files—often from a coding and software development standpoint. Git is designed for coordinating work among programmers, but it can be used to track changes in any set of files.



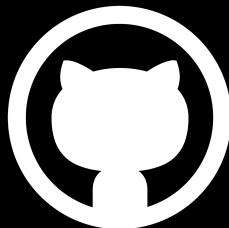
# What is a git repository?



**Answer:** A git repository is a remote or online file repository in which git tracks files and conducts version control as changes are made.



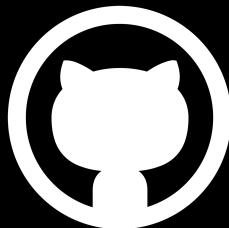
# How is git used?



**Answer:** Git is often used via the command line, but vendors such as GitHub and GitLab provide web and desktop apps that allow users to use git via a GUI.



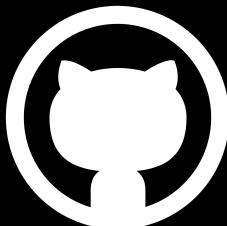
# What is GitHub?



**Answer:** GitHub is a web-based file-hosting service that is one of the many vendors that use git for file version control.



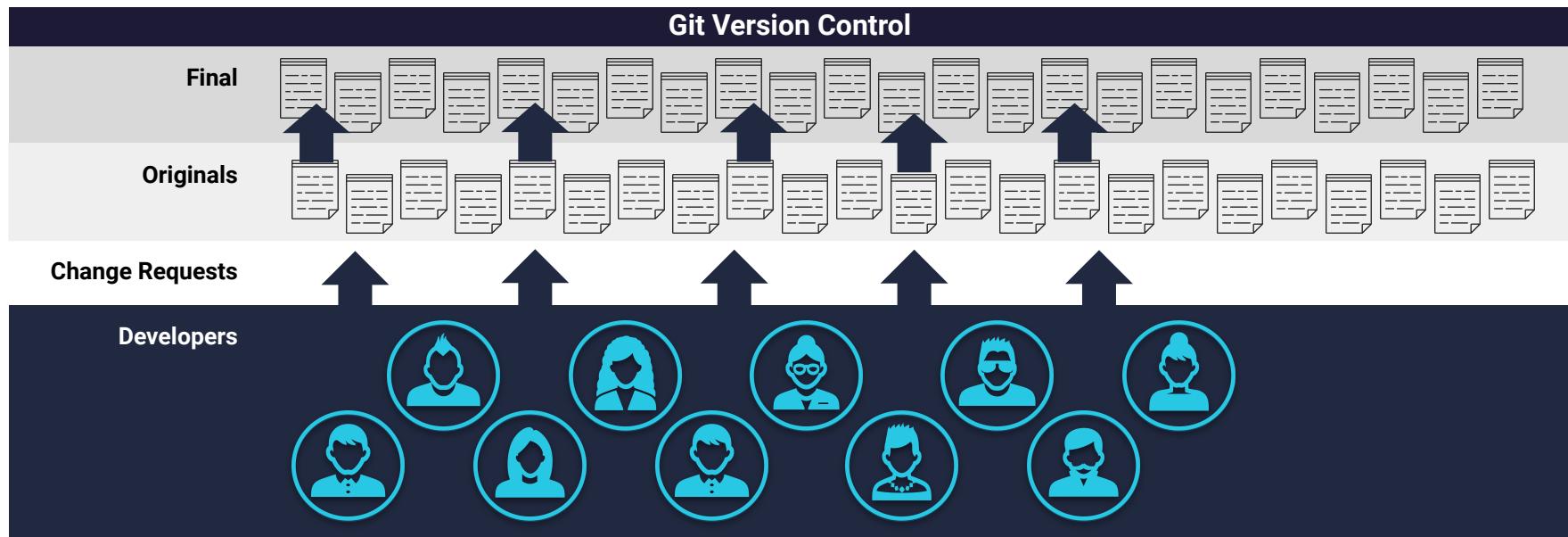
# Why is git important?



**Answer:** Git is an extremely powerful tool for software development. It has become the standard for versioning software and data science tools across industries and is even used to version data and enhance data reproducibility. For these reasons, proficiency in GitHub has become a critical job skill.

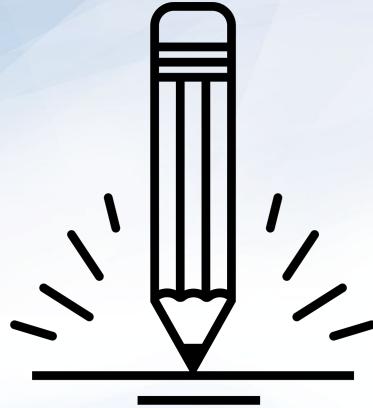
# GitHub Version Control

- Modern web development is highly collaborative.
- Teams are often extremely large and spread out across the country or world.
- Apps are sometimes made up of hundreds or even thousands of files.





## Instructor Demonstration Git Repositories



## **Activity:** Create and Personalize a GitHub Repository

Suggested Time:  
15 Minutes



# Activity: Create and Personalize a GitHub Repository

---

Complete the following steps.

1. Navigate to [GitHub](#) and log in with your credentials.
2. Create a new GitHub repository.
3. Download the ZIP file of the GitHub repository and extract the contents.
4. Modify the initialized README.md file and insert a quick introduction about yourself:
  - Who are you? What's your background?
  - What are your career goals?
  - Why did you choose FinTech as a career path?
5. Update the README.md file in your online GitHub repository.

Suggested Time: 15 Minutes





**Time's Up! Let's Review.**

# Introduction to Git CLI

# Git CLI vs. GitHub Web App

---

Git CLI provides commands to execute git operations.

```
>_
```



The GitHub web app provides a convenient user interface for performing **common git operations**.

>  
\_

**Git CLI** is a command line utility that provides all git operations; it is generally **more robust** than a git-based graphical user interface (GUI).

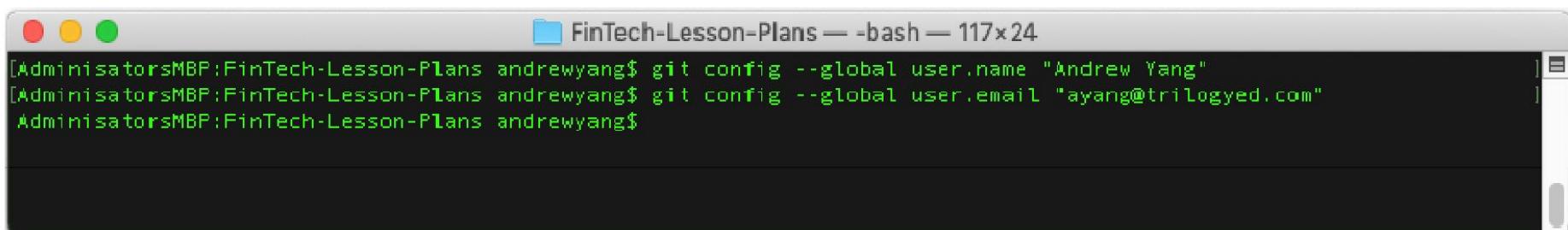
# Git Command Line Interface

# Configuring the Git CLI

---

```
git config --global user.name "Your Name"
```

```
git config --global user.email "Your Email"
```



The screenshot shows a macOS terminal window with a dark theme. The title bar reads "FinTech-Lesson-Plans — bash — 117x24". The terminal content is as follows:

```
[Administrator:MBP:FinTech-Lesson-Plans andrewyang$ git config --global user.name "Andrew Yang"
[Administrator:MBP:FinTech-Lesson-Plans andrewyang$ git config --global user.email "ayang@trilogyed.com"
[Administrator:MBP:FinTech-Lesson-Plans andrewyang$
```

# Popular Git CLI Commands

---

git clone

Clones a git repository to the local file system.

git add

Adds changed files to the queue of tracked files ready to be committed.

git commit

Adds tracked files as a bulk checkpoint ready to be pushed to the remote git repository.

git push

Uploads changed files from the local git repository to the remote git repository and updates the remote files.

git pull

Downloads changed files from the remote git repository to the local git repository and updates the local files.



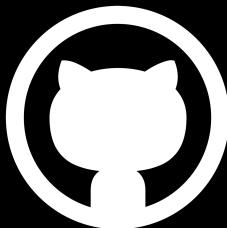
# GitHub Pop Quiz



# What is a git commit?



# What is a git commit?



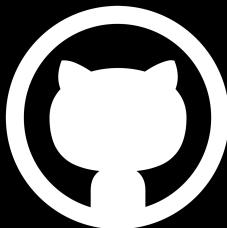
**Answer:** A git commit saves a queue of tracked changed files as a **save** or **checkpoint** for a git repository before the changes are pushed from the local to the remote repository. This way, a repo can be restored to a previous checkpoint in time, thereby undoing any existing changes from that point.



# What is git's Snapshot Model?



# What is git's Snapshot Model?



**Answer:** Git thinks of its data as a series of snapshots of a miniature file system. Every time you commit, or save the state of your project in git, it basically takes a picture of what your files look like at that moment and stores a reference to that snapshot. To be efficient, if files have not changed, git doesn't store the file again; rather, it stores a link to the previous identical file it has already stored.



Instructor Demonstration  
Git CLI



# **Activity:** GitHub Reorganized

(Instructions sent via Slack.)

Suggested Time:  
15 Minutes





**Time's Up! Let's Review.**

# Markdown

# What Is Markdown?

---

Markdown is a lightweight markup language that contains syntax for adding formatting elements to plain-text documents.

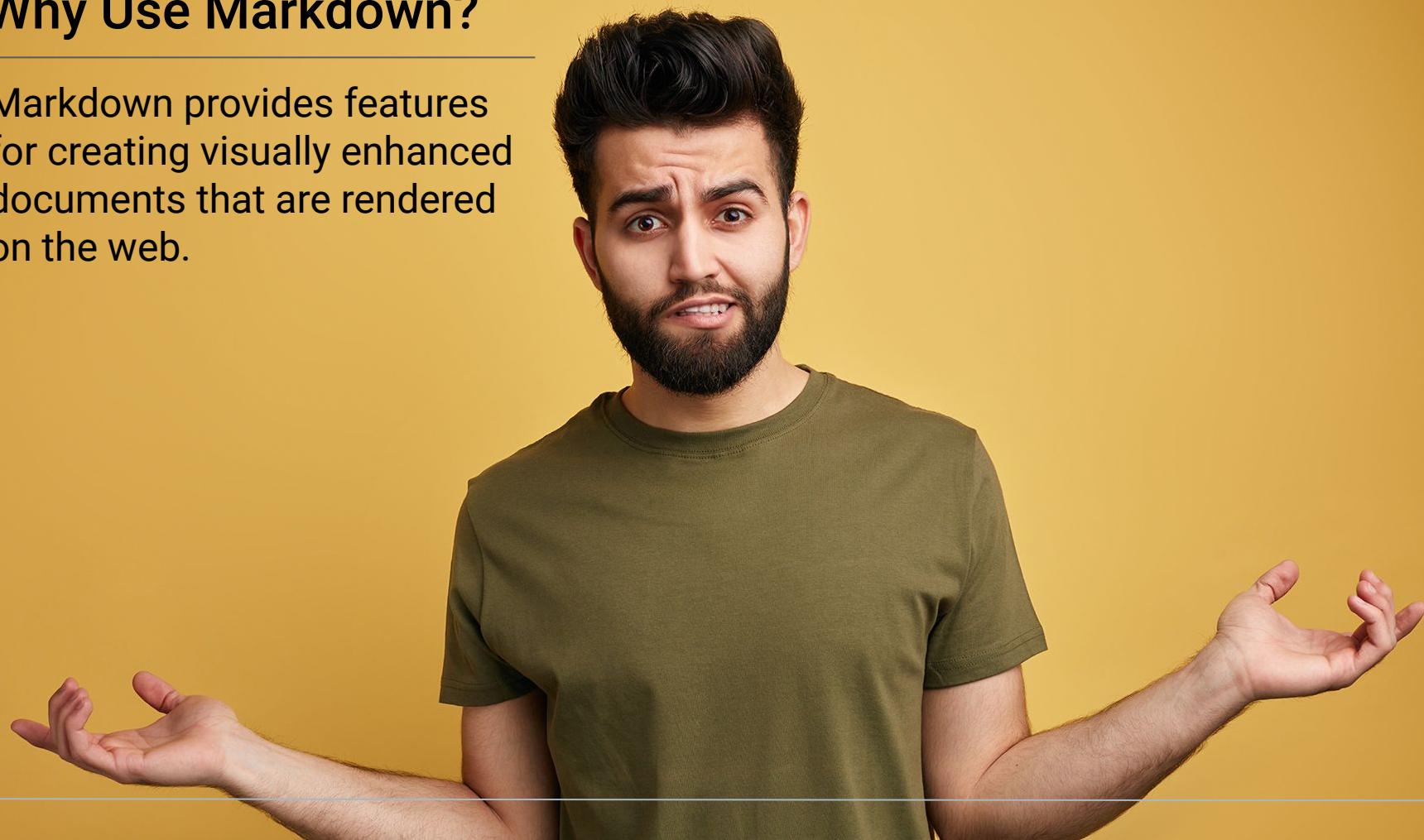


# MARKDOWN

# Why Use Markdown?

---

Markdown provides features for creating visually enhanced documents that are rendered on the web.



# What Are Some Common Markdown Features?

---



Header Formatting: #, ##, ###



Text Formatting: \*\*word\*\*, \*word\*



Line Breaks: ---



Text/Code Snippets: `word`



Block quotes: > “Quote”



Links:

Files/URLs: []()

Images: ![]()



A good **README** file helps people understand the purpose of the repository at a glance, and it shows developers how to navigate, install, and run a project.

# Minimal Structure for README Files

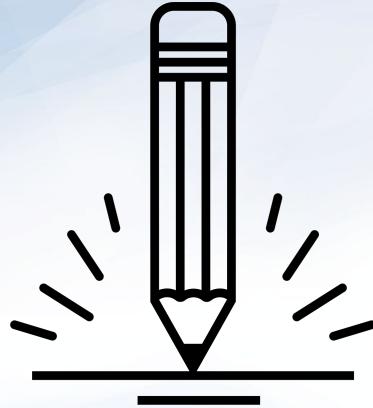
---

A good **README file** could be the difference between getting or not getting your next dream job, so be conscientious while writing these files.

```
# Project Title  
---  
## Technologies  
---  
## Installation Guide  
---  
## Examples  
---  
## Usage  
---  
## Contributors  
---  
## License
```



## Instructor Demonstration Markdown



## Activity: GitHub Welcome

In this activity, you will visually enhance your README files for your GitHub repository by adding additional markdown features.

(Instructions sent via Slack.)

Suggested Time:  
15 Minutes





**Time's Up! Let's Review.**



Countdown timer

40:00

(with alarm)

Break



# Writing Successful Case Studies

# Writing Successful Case Studies

As a FinTech professional, it is critical to learn how FinTech companies start, evolve, fail, and grow, and to be able to speak knowledgeably about the significant players in the FinTech space.

FinTech industry covers so many domains that it's hard to be savvy in every single area, so reading and writing **case studies** could give you the ability to be updated on the most recent advances and success stories in the FinTech realm.



# Writing Successful Case Studies

---

Being able to write a case study may give you an advantage while interviewing for a new job, or proving your mastery for a new position in your current company.

You will have the ability to outline your success or effectiveness in dealing with a client, a tough situation in your team, or a complex business scenario.

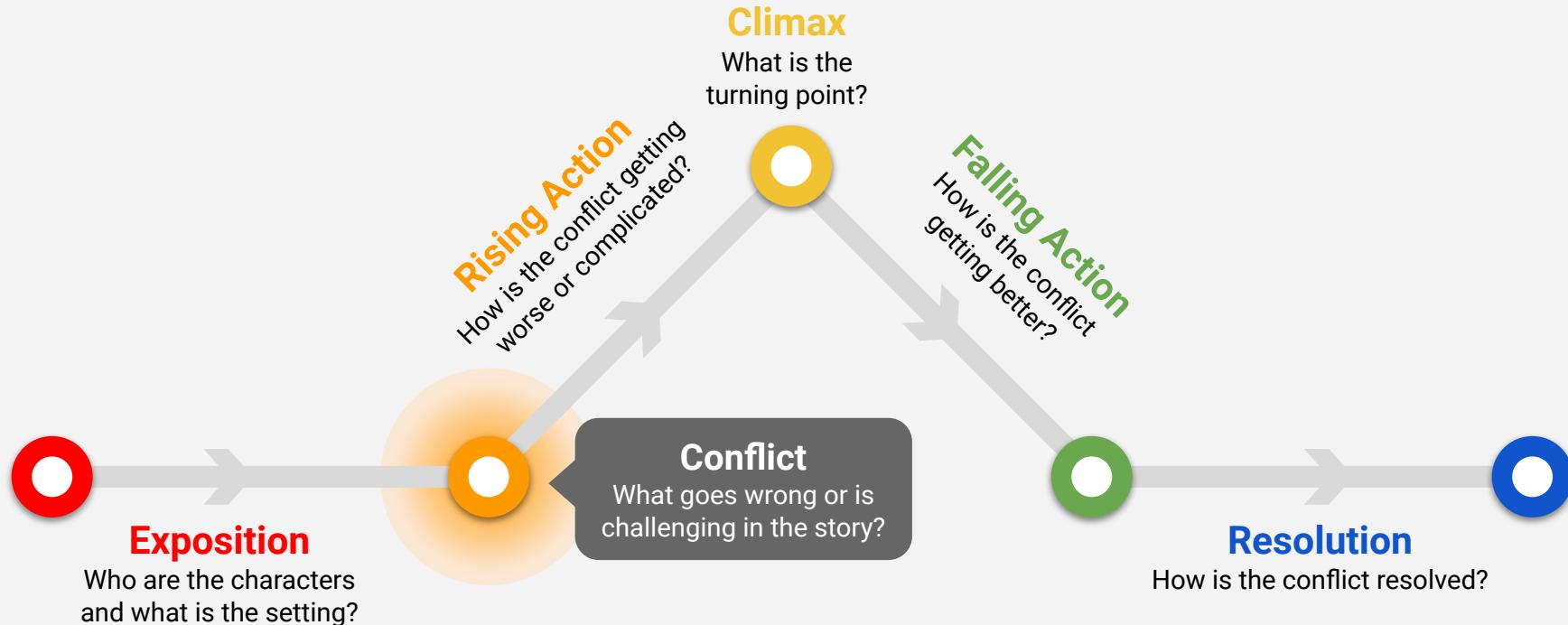


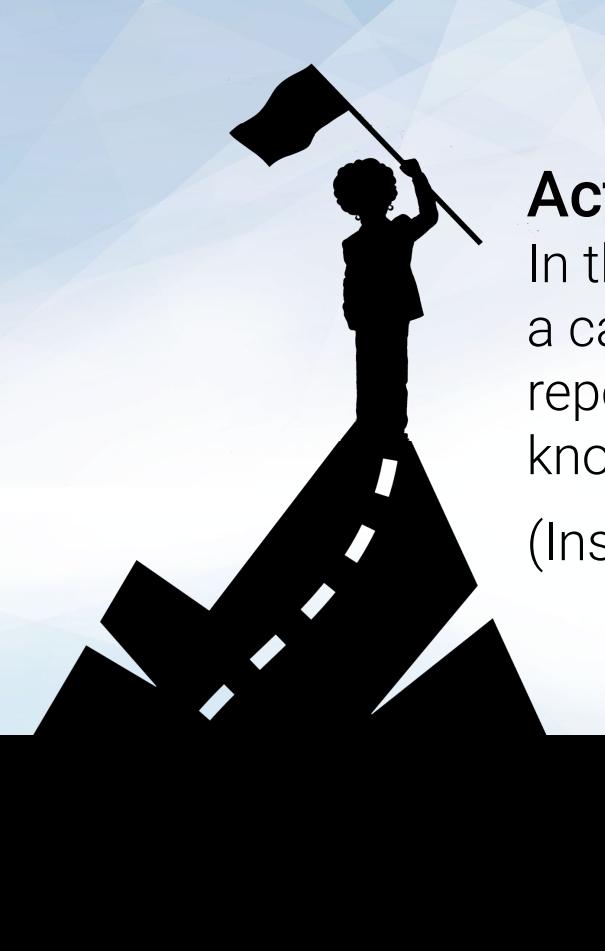


A **successful case study** is a self-contained story about reality. Typically it would be best if you described how a real customer overcame their problems thanks to your products, services, or professional experience.

# The Narrative Arc

As a general guide to writing a case study, you can use the “Narrative Arc” that encompasses the following stages.





## Activity: FinTech Case Study

In this activity, you will work with a partner to write a case study and host it on a shared GitHub repository. This activity will combine your knowledge of the git CLI and markdown syntax.  
(Instructions sent via Slack.)

Suggested Time:  
30 Minutes





**Time's Up! Let's Review.**

# Questions?

*The  
End*