BLACK CODHER

CODING PROGRAMME

CODING BLACK FEMALES

Black Codher Bootcamp

NÍ YÓ

BLACK CODHER

CODING PROGRAMME

UNIT 1
Command Line and Git

CODING BLACK FEMALES

NÍ YÓ

Goals for Unit 1

BLACK CODHER
CODING PROGRAMME

1. Understand the Command Line
2. Understand Git
3. Understand GitHub
4. Understand the difference between them
5. Complete basic tasks with each of these

BLACK CODHER

CODING PROGRAMME

Introduction to Command Lines

What is a command line?

BLACK CODHER
CODING PROGRAMME

An **interface** is a shared boundary e.g. between a computer and a human, hardware and software etc.

A **command line interface (CLI)** or interpreter is found in a computer programme which processes instructions provided as lines of text.

The **graphical user interface (GUI)** is what most people rely on when interacting, as it's more visual with colours, icons, menus, and so on.

Other key terms related to command lines include:

- Terminal
- Console
- Shell

Why use a command line?

BLACK CODHER
CODING PROGRAMME

It’s streamlined, powerful, and used when seeking:

- Key connection to Git - an essential tool
- Certain tools require us to e.g. no GUI
- Greater control over system functions
- Easier to carry out complex operations
- Saves time (automate repetitive tasks)
- More efficient in terms of PC memory

Think about what happens when you install software, i.e Java update?
Discuss pros’ and cons of GUI vs CMD with examples.

How does the command line work?

BLACK CODHER
CODING PROGRAMME

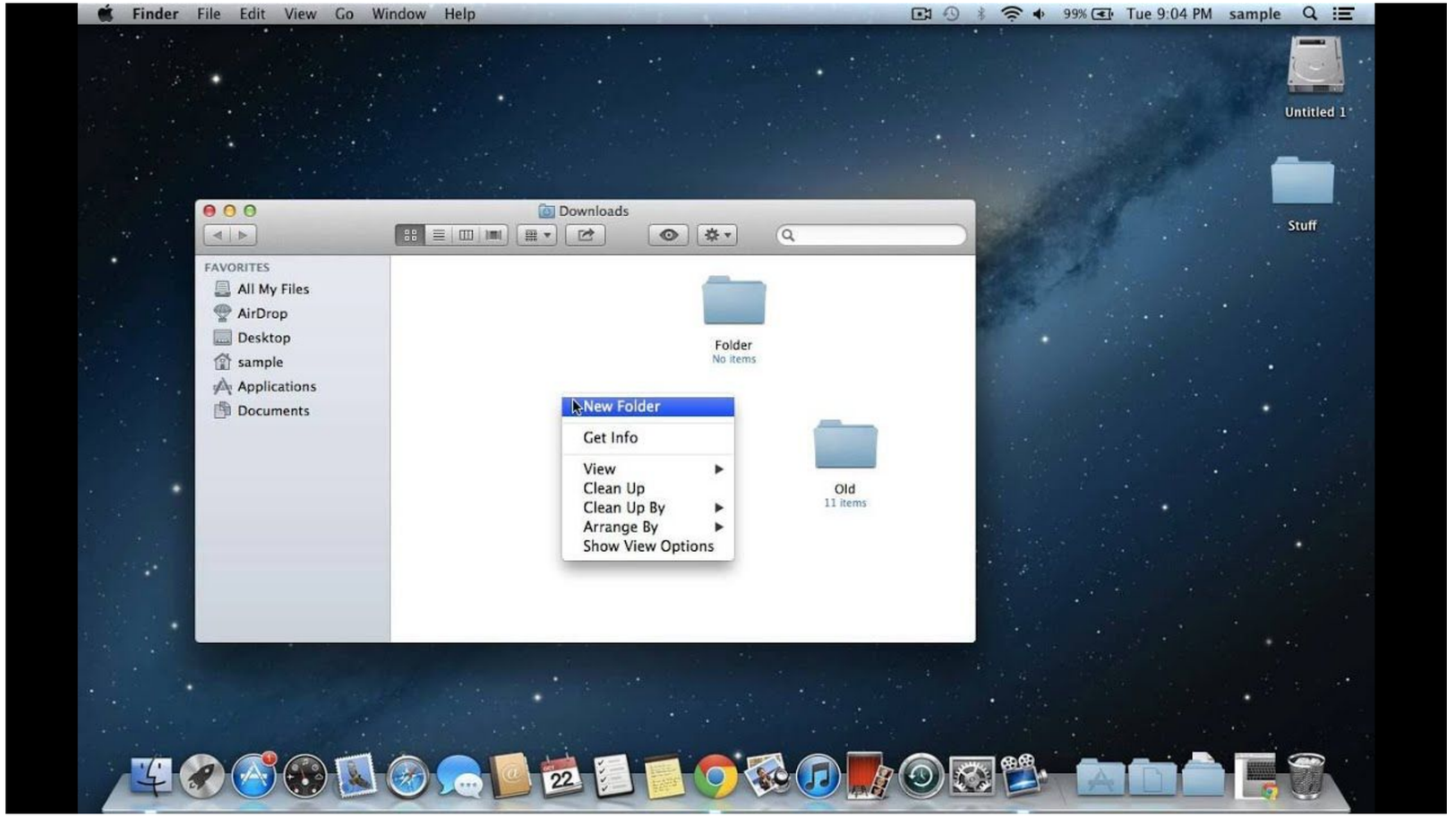
On macOS, the programme is **Terminal**.
On Windows, it's called **Command Prompt**.

You enter **commands** by typing or pasting them in line by line. They tend to look like abbreviations of tasks you want to complete:
e.g. mkdir - make a directory.

This then sends those instructions to the computer. The **commands** for macOS are usually different to the ones for Windows.

How does the command line work?

BLACK CODHER
CODING PROGRAMME



[Downloads] mkdir myNewFolder

directory *command* *[option]*

BLACK CODHER

CODING PROGRAMME

Command Line 101

Learning objectives

BLACK CODHER
CODING PROGRAMME

- Make a directory
- Find the directory
- Change directory
- Create a file
- Move the file

Opening the command line

BLACK CODHER
CODING PROGRAMME

Rules:

- every character matters, including spaces
- you can navigate using the arrow keys
- To interrupt a command that's already running, type Control-C
- Commands are always executed in the current location

[Downloads] mkdir myNewFolder

```
[~]

 >
```

Commenting on our code

We'll be using comments above our code

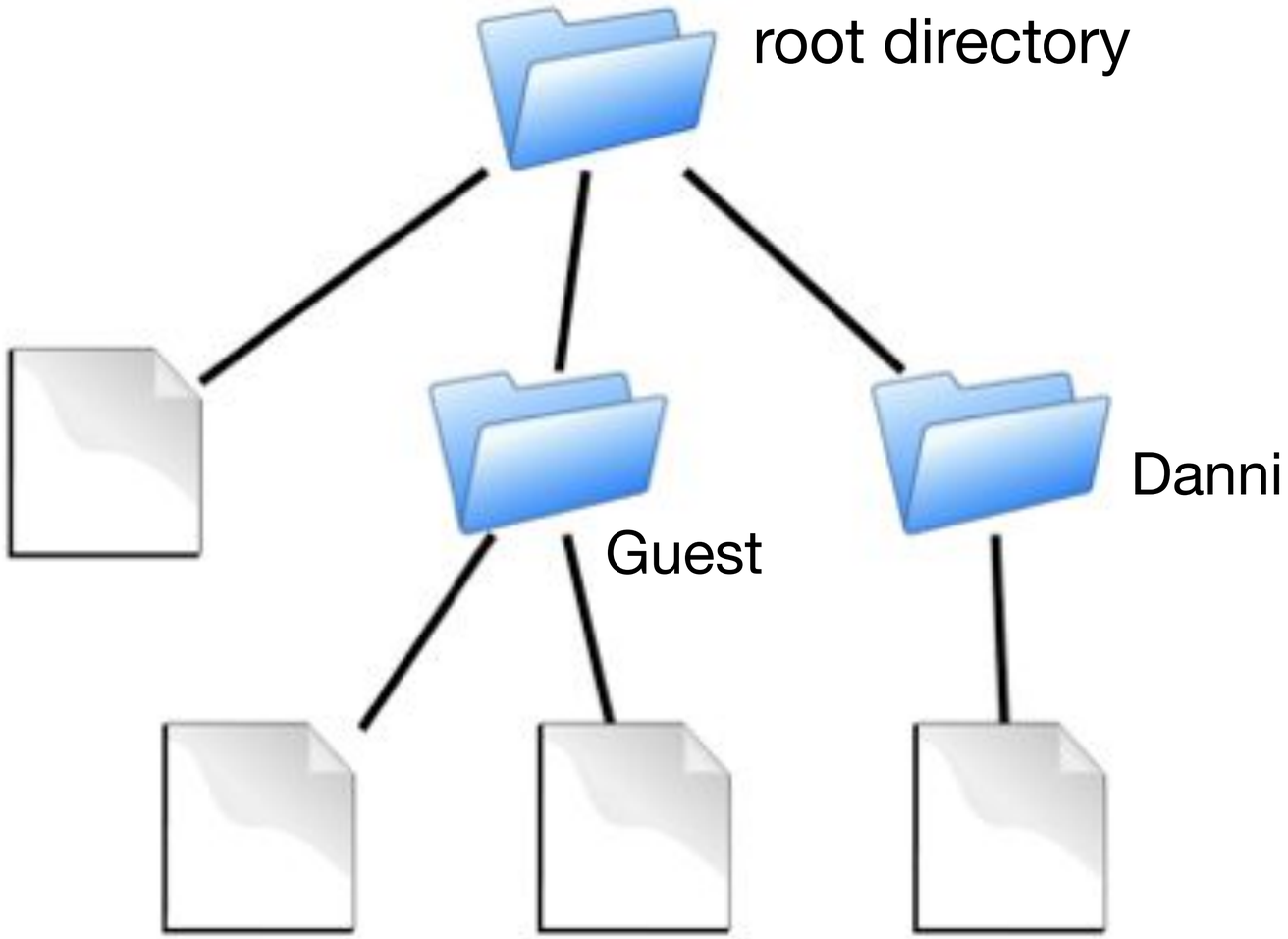
Computers ignore them, they help us humans - note to self and others.

is what we'll be using for macOS. rem is used for Windows, short for remark. It must always come before what you're writing.

```
# macOS
[~]


rem Windows
 >
```

Making a directory



Making a directory

When you first open your command line, you will start at the **root directory**.

This is the same place you land when you open Finder - on macOS and C: on Windows

To make a new directory, you use the **mkdir** *[name]* command, followed by what you want to call it.

NB: Words in italics wrapped in square brackets e.g. *[words]* indicate things which must be replaced.

```
# macOS
rem Windows
[~] mkdir black-codher
```

Finding the directory

On macOS, the **pwd** command prints the directory where you are located in the file system.
pwd - print working directory.

On Windows, the **cd** command does the same.

Here, print refers to the output shown on your command line.

```
# macOS
[~] pwd

rem Windows
> cd
```

Changing directory

BLACK CODHER
CODING PROGRAMME

The **cd** *<directoryname>* command changes your location to where you state.
cd - change directory.

cd .. you back the next level up, leading to the root.

These apply to both macOS and Windows.

```
# macOS
rem Windows
[~] cd black-codher
> cd ..
```

Creating a file

BLACK CODHER
CODING PROGRAMME

For macOS, the **touch** *[filetitle]* command creates a new blank file.

For Windows, **fsutil file createnew** *[filetitle] [length]* does the same.

File system utility is shortened to **fsutil**. The file extension/suffix **.txt** shows that the type is a text file.

Length here is expressed in characters. 1000 = 1 kilobyte (KB). Feel free to create a few more files.

```
# macOS
[~] touch example.txt

rem Windows
> fsutil file createnew example.txt 1000
```

Moving the file

For macOS, the command **mv** *[filename] [pathname]* moves a file to the destination.

For Windows, it's similar **move** *[filename] [pathname].*

Pathname refers to exactly where you want to place it, the file system "journey".

```
# macOS
[~] mv example.txt ~/Documents

rem Windows
> move example.txt C:\Documents
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Pop Quiz

BLACK CODHER
CODING PROGRAMME

How do we:

- make a directory?
- find the directory?
- change directory?
- create a file?
- move the file?

You can use the slides/notes/Google!
Coding is not about a memory test...

BLACK CODHER

CODING PROGRAMME

Time for a break - you deserve it!

Learning objectives

BLACK CODHER
CODING PROGRAMME

- Open a file
- Show the content of a file
- List the content of a directory
- Delete a file
- Delete a directory

Opening a file

BLACK CODHER
CODING PROGRAMME

You can open the file in your VSCode with the command we installed earlier.

The full stop means that you want to open the file in the directory you are currently in.

This should launch your VSCode. Once it's open, write something inside your file, and click save.

```
# macOS
rem Windows
[~] code .
```

Show the content of a file

BLACK CODHER
CODING PROGRAMME

On macOS, the **cat** *[filename]* command prints the content of the file you specify on the command line.

On Windows, **type** *[filename]* does the same.

So this should show what you wrote in the example file.

```
# macOS
[~] cat example.txt

rem Windows
[~] type example.txt
```

List the content of a directory

BLACK CODHER
CODING PROGRAMME

On macOS, the **ls** *[directoryname]* command prints the content of the directory.

On Windows, **dir** *[directoryname]* does the same.

So this should show you all the files inside your directory.

```
# macOS
[~] ls black-codher

rem Windows
[~] dir black-codher
```

Delete a file

BLACK CODHER
CODING PROGRAMME

On macOS, the **rm** *[filename]* command removes a file.

On Windows, **del** *[filename]* also deletes it.

This is more permanent than the recycle bin or trash. So be sure! A common use case is problems files which can't be removed or deleted as normal.

```
# macOS
[~] rm example.txt

rem Windows
[~] del example.txt
```

Delete a directory

BLACK CODHER
CODING PROGRAMME

On macOS, **rm -R** *[directoryname]* removes a directory.

On Windows, **deltree** *[directoryname]* does the same. The tree metaphor links to the the branch concept that we talked about earlier for Git.

These commands work for directories with files in them. You can use **rmdir** on both OS's, but only to remove empty directories.

```
# macOS
[~] rm -R black-codher

rem Windows
[~] del black-codher
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Pop Quiz

BLACK CODHER
CODING PROGRAMME

How do we:

- open a file?
- show the content of a file?
- list the content of a directory?
- delete a file?
- delete a directory?

Try this sequence

BLACK CODHER
CODING PROGRAMME

Task One: Here's a list of commands! Do them in order! What happens?

```
$ mkdir test
$ mv example.txt test
$ cd test
$ ls
$ rm example.txt
$ cd ..
$ rmdir -i test
```

```
$ mkdir test
$ move example.txt test
$ cd test
$ dir
$ del example.txt
$ cd ..
$ rmdir /s test
```

Try this sequence

BLACK CODHER
CODING PROGRAMME

Task Two: Try and find out how to do the following

1. Write a command that prints out the string “hello, world”
2. What command ‘cleans’ the screen?
3. Create a .txt file in your Documents folder called helloWorld. Copy the folder and put the copy on your desktop. Get a list of all the folders on your computer called helloWorld.

BLACK CODHER

CODING PROGRAMME

Introduction to Git

What is Git?

BLACK CODHER
CODING PROGRAMME

Git is a **version control system** that lets you store different variations of a **codebase** on your own computer. It's accessed via the **command line or terminal**.

A **codebase** is a collection of source code to build a particular software:

- system
- component
- application

Why use Git?

BLACK CODHER
CODING PROGRAMME

With it you can:

- Track changes to your code easily which also scales well for larger projects
- Go back to previous versions with a '*revert*' option e.g. if there is a bug to fix
- Try things with separate '*branches*' without affecting the main '*tree*' source
- Have integrity via hash ID secure '*cryptographic*' keys generated each time

How does Git work?

BLACK CODHER
CODING PROGRAMME

Git is very different from saving things in word.

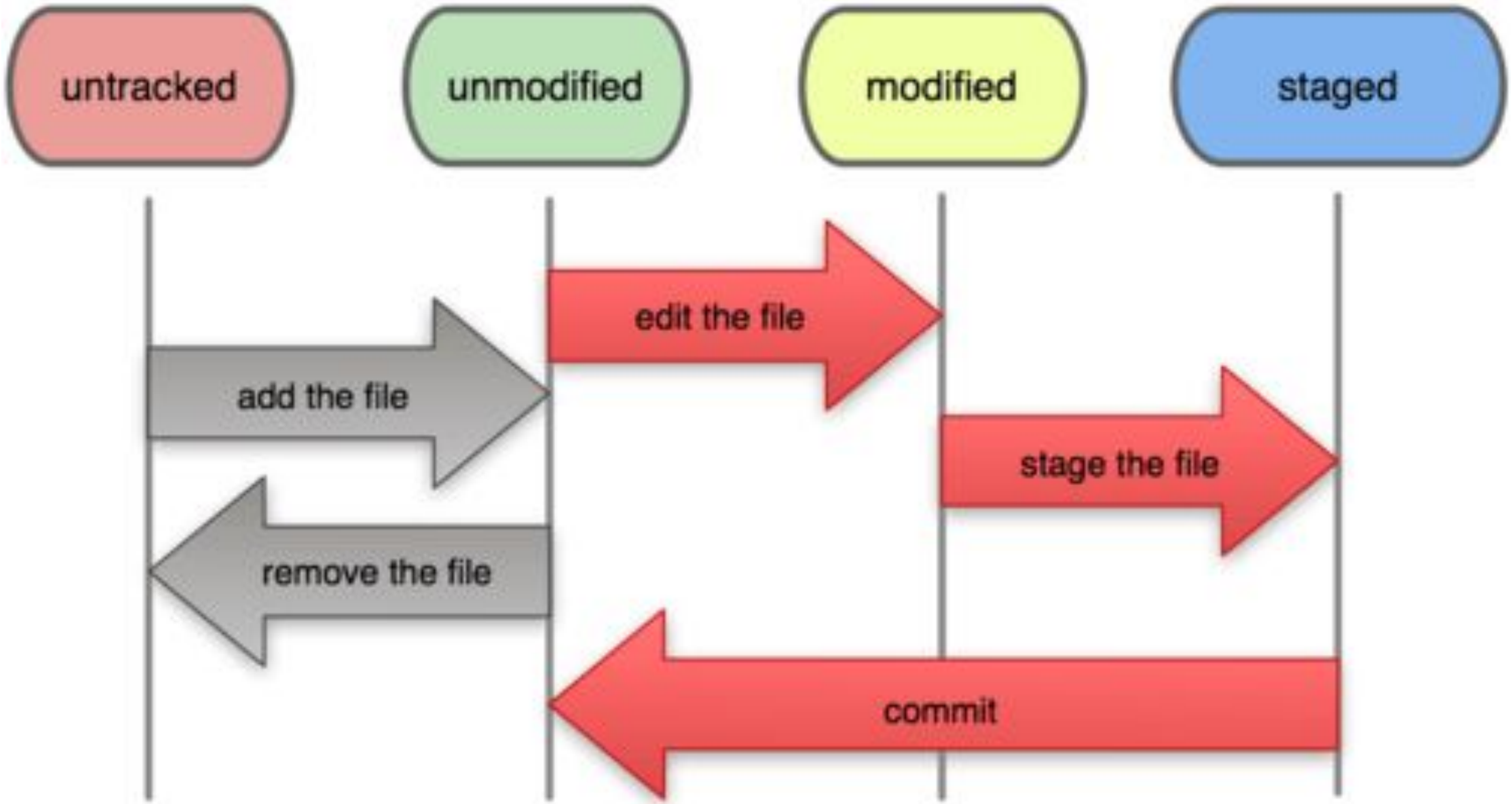
If you create a new empty folder to store your code and then initialize Git inside the main folder. This creates a **git repository.**

Every time you want to record a snapshot of your changes you can a Git command, like git add or git commit.

Git will save the snapshots inside your repository. Each one is like a checkpoint or node in an ever growing tree. Each commit includes all line/filechanges, a **unique label** and **comment** about what you did.

How does Git work?

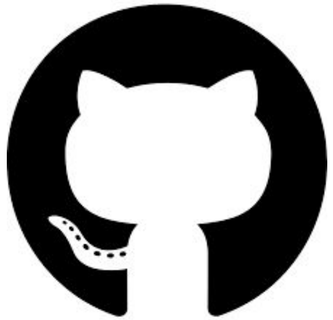
BLACK CODHER
CODING PROGRAMME



BLACK CODHER

CODING PROGRAMME

Introduction to GitHub

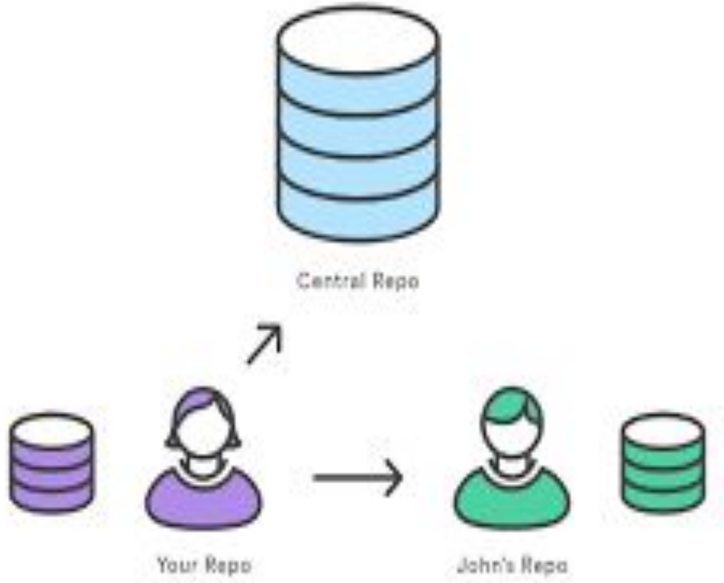


What is GitHub?

GitHub is a **distributed version control system (DVCS)** and **source code management (SCM)** functionality of **Git** using the cloud. It's sometimes described as a *'graphical web interface'* for it.

Additionally, it serves as a **social network** for developers, plus companies have accounts on there too where they share code publicly.

You can follows others, favourite, and make use of resources. As well as the private uses, a big part of of its online community is the **open source** aspect enabling collaboration.



Why use GitHub?

BLACK CODHER
CODING PROGRAMME

GitHub has many added functionalities to enhance **Git**'s capabilities:

- task management to assign issues
- access control
- wikis
- bug tracking
- feature requests

...and many more

Who uses GitHub?

- 50 million users
- 2.9 million companies
- 100 million codebases

Most commonly used service.

[graph to be inserted showing range of company types]

BLACK CODHER

CODING PROGRAMME

Time for a break - you deserve it!

BLACK CODHER

CODING PROGRAMME

Git 101

Learning objectives

BLACK CODHER
CODING PROGRAMME

- Know how to get help
- Introduce yourself to Git
- Initialise a Git repository
- Add files
- Make a commit

Connect to Git in VSCode Terminal

BLACK CODHER
CODING PROGRAMME

Open Visual Studio Code to access its built-in terminal.

You can do so via the keyboard shortcut **CTRL + `** (2 keys) on macOS, Windows, or Linux.

Once you're there, make a new directory. Then change location to move to this new folder.

```
mkdir git_test
cd git_test
```

Getting help

BLACK CODHER
CODING PROGRAMME

You can use the **git help** command to get a list of the most common commands used in **Git**.

If you put the name of a command at the end, you can view an on screen manual for how to use it.

There are other suffixes which can show different lists - we encourage you to look that up on Google if you'd like to learn some more.

```
git help
```

Greeting Git

BLACK CODHER
CODING PROGRAMME

A useful part of configuring Git before carrying out further operations is using the 2 main **git config** commands to identify your work with your name and public email address. This creates **.gitconfig** text files.

```
git config --global user.name [Name]
git config --global user.email [Email]

git config --global user.name Melz Owusu
git config --global user.email melz@gmail.com
```

Feel free to use a nickname as your global user.name and or create a new email address for your global user.email if you prefer. As this is visible in every commit you make, GitHub has 2 privacy settings.

Initialise a Git repository

BLACK CODHER
CODING PROGRAMME

git init loads / boots version control. This turns the directory into a **Git** repository by adding a **.git** subdirectory.

A software **repository** is where code is stored. It is also known as a repo.

It contains a **HEAD** file, which refers to the **main branch** where everything branches from in the tree structure we explained. This code must always be **deployable** - ready for users to use.

```
git init

#Alternatively
git init black-codher
```

Add a file

BLACK CODHER
CODING PROGRAMME

The **git add** command is mainly used for adding files and adding changes.

git add *[filename]* adds that file
git add . - adds all files

A key term here is **staging** - it's what we're doing by adding files as we get them ready to be committed to the codebase. We need to make sure they can perform before that - ready for the big stage!

```
git add example.txt

git add.
```

Make a commit

BLACK CODHER
CODING PROGRAMME

The **git commit** command takes a snapshot of the repository's currently staged changes at that point in time.

It prompts you to note what you did in a commit message. After you do so, save and close the editor to finish.

Commits are the building block units of a Git timeline. They can also be considered as milestones. They are always saved to the local repository.

```
git commit

# All changes
git commit -a

# This shortcut skips a step
git commit -m "commit message"

# What do you think this does?
git commit -am "commit message"
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Pop Quiz

How do we:

- git help *wink wink*?
- initialise a repository
- add a file
- add all files?

Let’s discuss

What is staging?

Why do we need version control?

How would you make sure the code in the main branch is always deployable?

Let’s try

BLACK CODHER
CODING PROGRAMME

- Initialise a Git repository
- Add files
- Make a commit
- Delete files
- Make a commit

BLACK CODHER

CODING PROGRAMME

Time for a break - you deserve it!

Learning objectives

BLACK CODHER
CODING PROGRAMME

- Learn how to use git branch
- Learn how to use git checkout
- Learn how to use git merge

Branching out with git branch

BLACK CODHER
CODING PROGRAMME

The **git branch** *[name]* command is used to depart from the main branch e.g. to test out ideas, fix bugs, experiment with features etc.

A branch represents a new line of development. As such, you get another working directory, staging area, and project history.

```
git branch codingblackf

# What do you think this does?
git branch -d codingblackf

# And what about the capital D?
git branch -D codingblackf
```

Checking out with git checkout

BLACK CODHER
CODING PROGRAMME

git checkout *[name]* is how you switch to the branch that you want to work on, adding, editing files etc.

It can also be used to undo things, view past commits, and as a shortcut to create.

```
git checkout codingblackf

# What do you think this does?
git checkout -b codingblackf
```

Merging branches with git merge

git merge brings together separate lines of development made using **git branch** and integrates them into one branch. In most cases, this is two branches - sets of commits.

As always, let’s try it out.

```
git merge codingblackf
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

True or False

git branch leads to a new line of development with a new directory, staging area, and project history.

git checkout can also be used to create a branch via a shortcut, besides switching branches etc..

git merge can only merge two branches together each time.

BLACK CODHER

CODING PROGRAMME

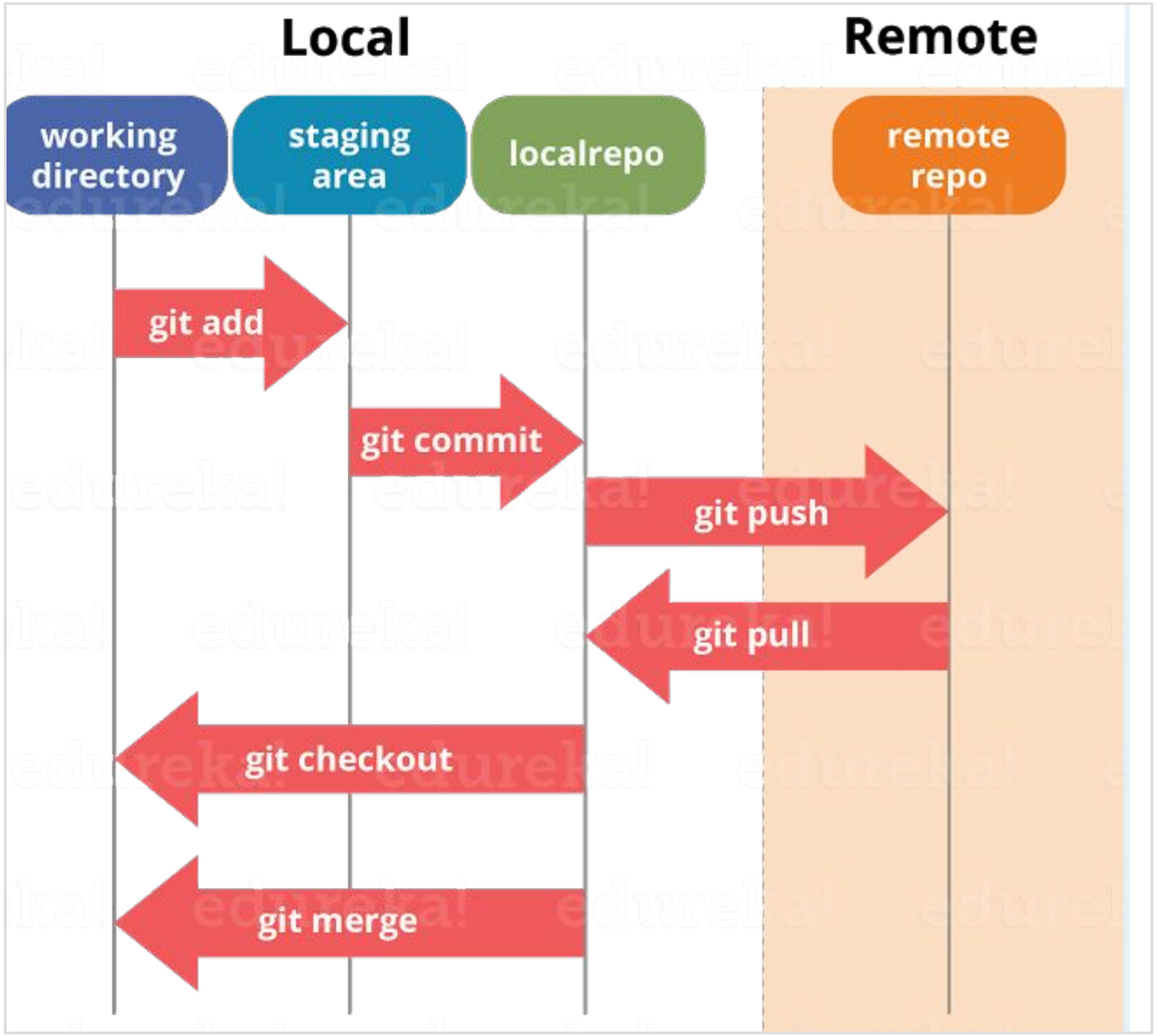
GitHub 101

Learning objectives

BLACK CODHER
CODING PROGRAMME

- Understand how to set up remotes
- Understand how to push changes
- Understand how to pull changes

Setting up a remote repository



Setting up a remote repository

BLACK CODHER
CODING PROGRAMME

The **git remote command** lets you create, view, and delete connections to other repositories.

git remote lists the remote connections you have added.

git remote add *[name] [URL]* is how you link Git on your command line to the web page of a repository you made on GitHub. Let's start there...

```
git remote


git remote add origin github.com/user/repo.git
```

Pushing changes with git push

BLACK CODHER
CODING PROGRAMME

The **git push** command is used to upload the content of a local repository to a remote one. In essence, you're sending over your commits.

git push *[remote] [branch]* calls upon the names you assigned.

Let's...

```
git push origin codingblackf
```

Pulling changes with git pull

BLACK CODHER
CODING PROGRAMME

git pull fetches and downloads the content of a remote repository and updates the local one to sync them.

It combines **git fetch + git merge,** getting all the changes from the point where the local repo and main branch became different.

git fetch does the same in terms of downloading, but it does not make any changes with **git merge.**

```
git pull
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said

Carry out these steps

Task Five:

1. Go to your github account and create a new remote repo
2. Do no add a ReadMe or License
3. Copy the SSH link and use it to clone the repo to your local workspace
4. Create a new file in your local repo and commit the change
5. Push the change to your remote repo
6. Refresh your GitHub account, has the change appeared?

BLACK CODHER

CODING PROGRAMME

1. Bonus: Clone the following rep
2. Change a file and submit a merge request

See you next time!