**A close up of a logo

Description automatically generated**

**Git for Windows**

**Function: Track project history and work together.**

[**https://git-scm.com/**](https://git-scm.com/)

A distributed (rather than centralised) version control system that records changes made to code over time and saves this to a special database repository. Can see changes everyone makes and can easily rewind it to a certain place in time.

*What is Git?*

<https://www.youtube.com/watch?v=2ReR1YJrNOM>

*A one-hour tutorial on Git:*

<https://www.youtube.com/watch?v=8JJ101D3knE>

*GUI for using Git:*

* GitKraken Git GUI
* Sourcetree (windows and mac only)

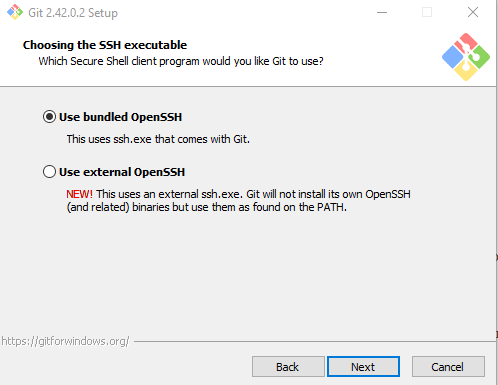
Learning to use the ***command line*** is an important tool to use Git, because sometimes a GUI tool is limited. Or you can use GitCMD which comes installed with GitHub

**How to Install Git on Windows**

1. Go to git and download[**https://git-scm.com/download/win**](https://git-scm.com/download/win)

When installing let the defaults apply.

**A screenshot of a computer

Description automatically generated  A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

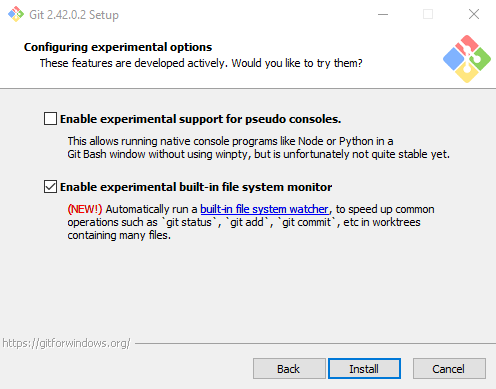
Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated **

**Notice on the last option I have selected the NEW feature, to stay up with the latest advancements.**

1. Click search icon and type **cmd**

**A screenshot of a computer

Description automatically generated**

1. Type this into the command line and hit enter: winget install --id Git.Git -e --source winget
2. Then you need to update the global settings using the command line to add your personal details with the below commands. You also need to download VS Studio Code first. <https://code.visualstudio.com/>

**A screenshot of a computer program

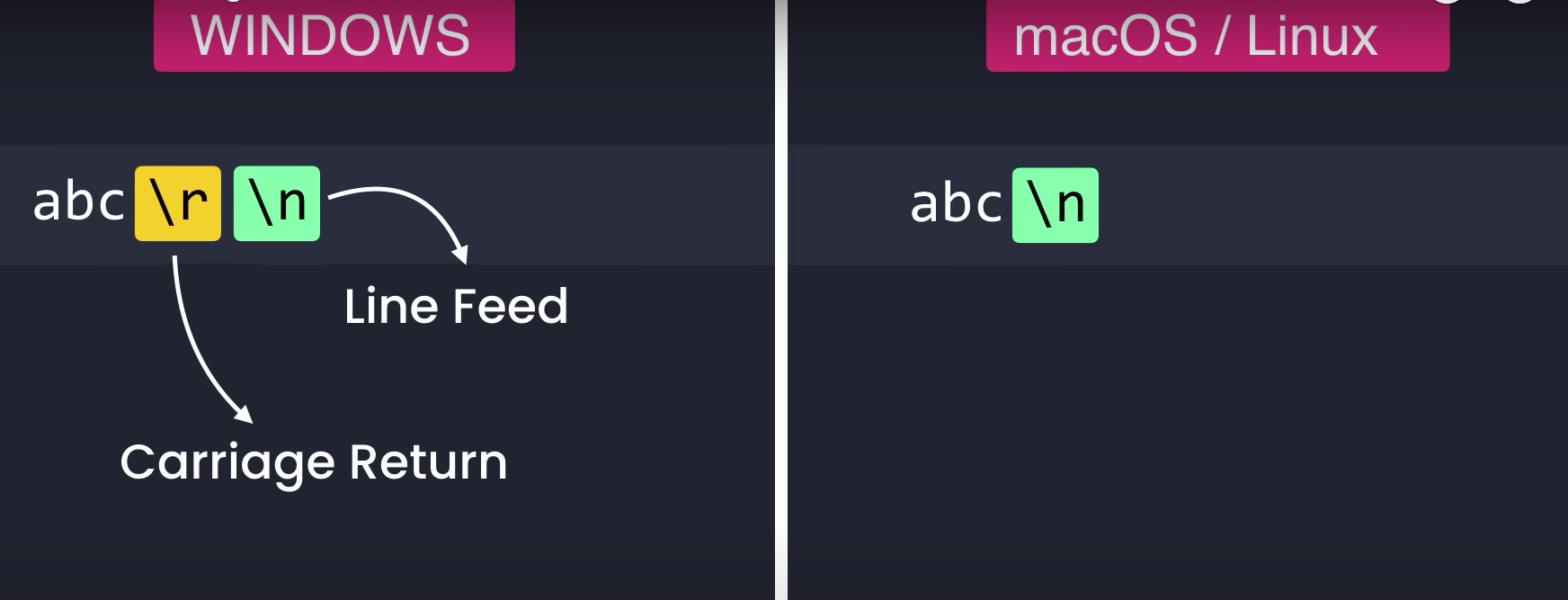
Description automatically generated**

1. The last prompt should open a new window of VS Code:

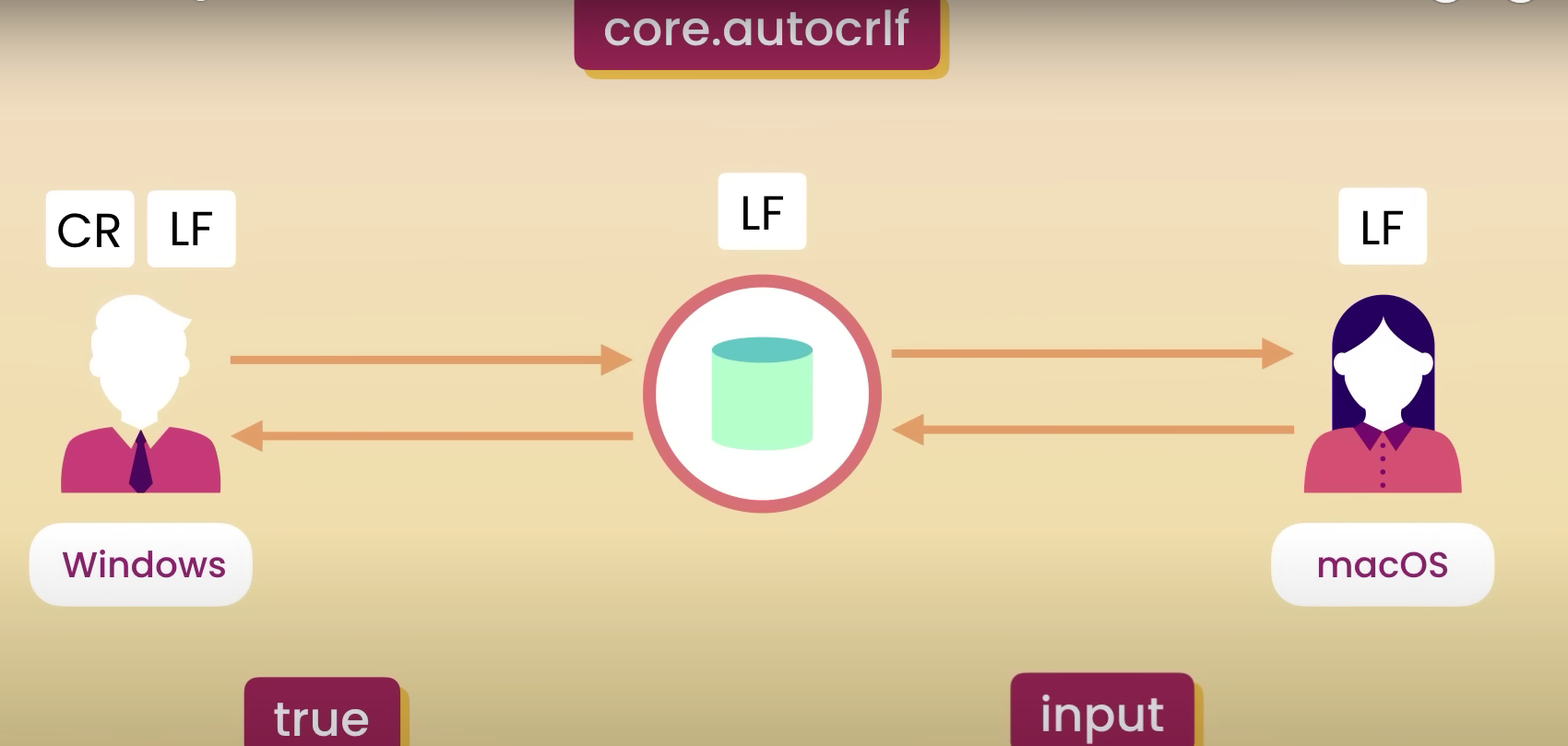
A screenshot of a computer

Description automatically generated

1. Close VS code window and now we need to configure how windows handle end of line, this is important because when different users are using mac and window this will prevent problems.



We need to configure a property called: core.autocrlf Short for carriage return line feed



Type this into command line for Windows: git config --global core.autocrlf true

Type this into command line for Mac: git config --global core.autocrlf input

**How to Create SNAPSHOTS of Your Workflow**

1. Initialising a new empty repository in a folder. Open the folder by typing cd into the command line followed by the filepath where you want to create your new project
2. Create a directory for the project (in this example we are making a folder “Moon”): mkdir Moon
3. Open the directory moon: cd Moon
4. Initialise an empty git repository: git init

A screenshot of a computer

Description automatically generated

1. You can add files and check on status:

A computer screen with white text

Description automatically generated

1. You can commit them to the staging area

A screenshot of a computer program

Description automatically generated

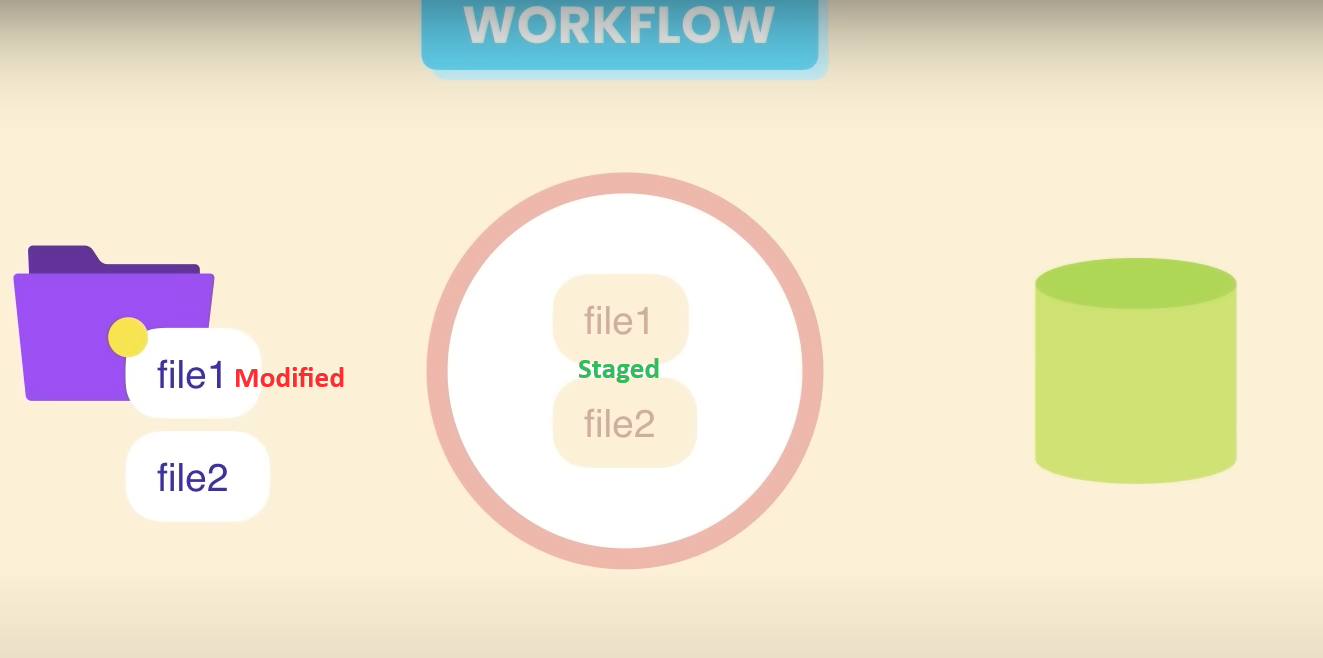
1. Now the files are on the staging area. But we can modify the original file, example: echo world >> file1.txt

Then check on: git status

A screenshot of a computer program

Description automatically generated

1. Or this can be represented like this:



1. You can then add your modified file to the staging area and check status again.

A screenshot of a computer

Description automatically generated

**Committing Changes**

**How to Commit Snapshot to Permanently Store in Repository**

1. Start with this command and a message about the snapshot: git commit -m “Initial commit”

Or you might want to write more details about the snapshot so you can use the VS Code editor by typing this into the command line and it will open since we configured this at the start: git commit



A new window of VS Code should open where you can type your message.

A screenshot of a computer

Description automatically generated

1. Write title of message in first line under 80 characters and then a line space and you can write a longer more detailed message from there. Hit “ctrl + S” to save and close window, you should now see this message (or similar) in the GitCMD console.

A screen shot of a computer code

Description automatically generated

These are basic statistics of what was changed

1. Or if confident you can skip the staging area: Making change and straight to Commit with message:

A black screen with white text

Description automatically generated

**Removing files**

A black screen with white text

Description automatically generated

**FAQS:**

**What is Git BASH?**

It gives users Unix based shell utilities and experience for Git command line commands.

<https://www.gitkraken.com/blog/what-is-git-bash>

**Is it advisable to get familiar with GitKrakken: Yes**

<https://www.gitkraken.com/>