

Introduction to Linux*

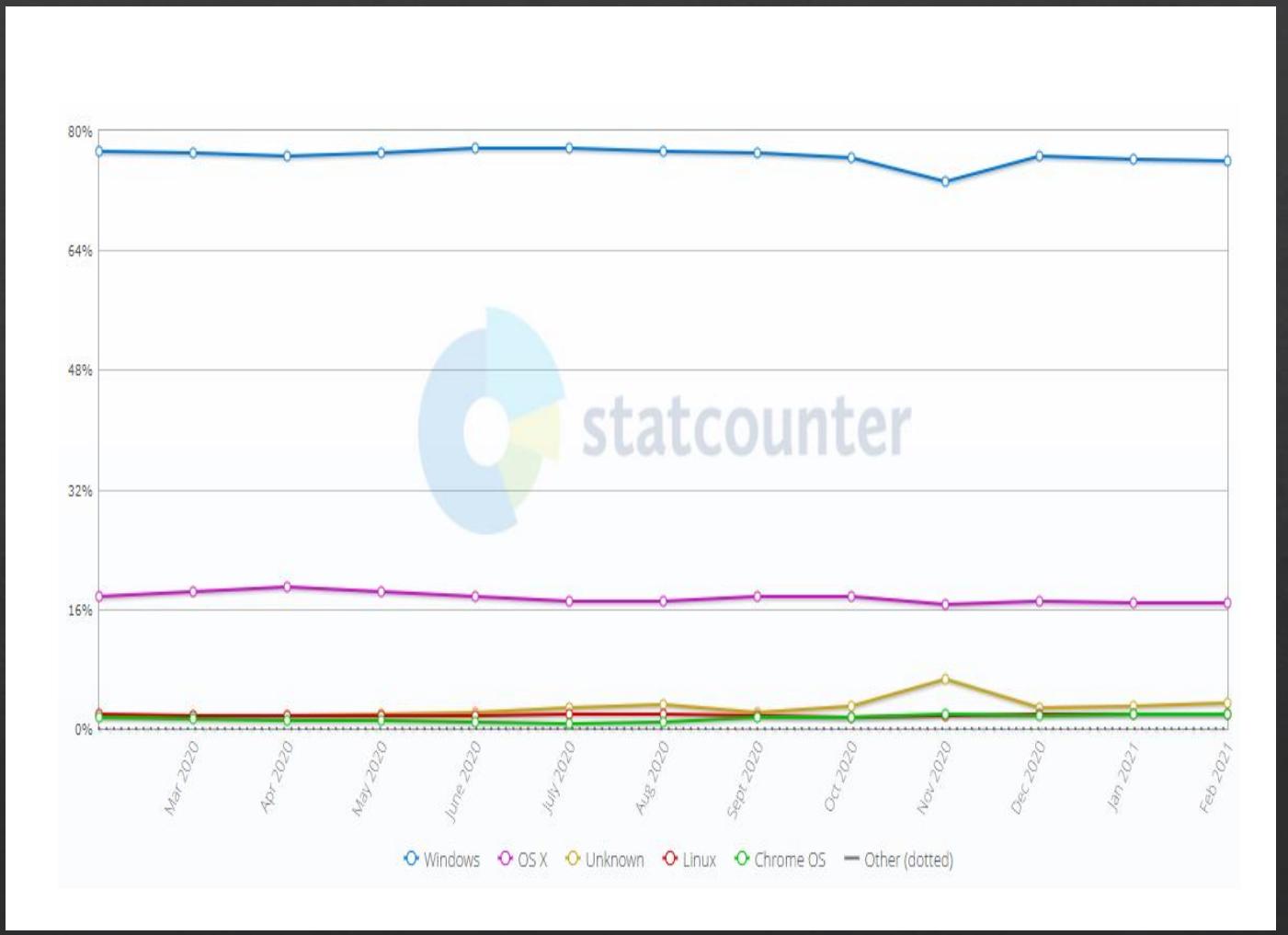
*And some Git, too!

Why Linux?

- ❖ First off... We actually call it *NIX (as in, UNIX-Like). But, for all intents and purposes, “Linux” will do.
 - ❖ The “Why” is complicated, and like many things, wrapped in legalities.
- ❖ When you think of a “computer” what comes to mind?
 - ❖ There’s a lot of computers out there. Not all are intended to do things that we traditionally use them for.
 - ❖ Client / Server model
 - ❖ GUI

Look Familiar?

- ◊ Is it exactly this? Probably not.
 - ◊ But it is probably very close.
 - ◊ It's extremely misleading
 - ◊ It's a very small slice of “the pie”
-
- ◊ Credit:
<https://gs.statcounter.com/os-market-share/desktop/worldwide>



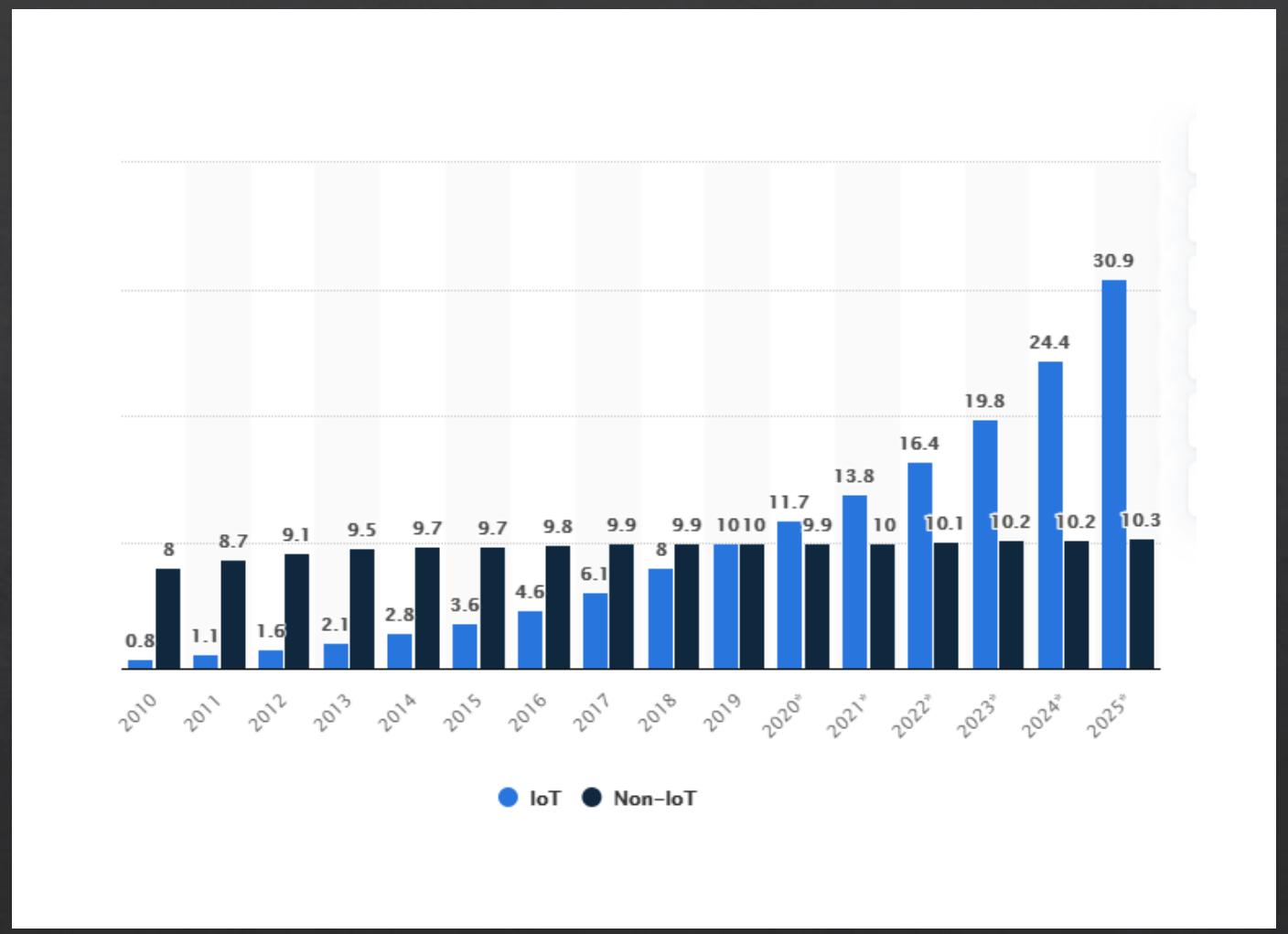
What's a “Computer” again?

- ❖ No, not the commercial from apple
 - ❖ Although, it was very on point.
 - ❖ It was also kind of universally hated.
 - ❖ Seriously... They took it off of YouTube
- ❖ How many “things” in your house do you think have a processor?
 - ❖ Do the terms “IoT” or “Smart” mean anything to you?

Embedded Systems

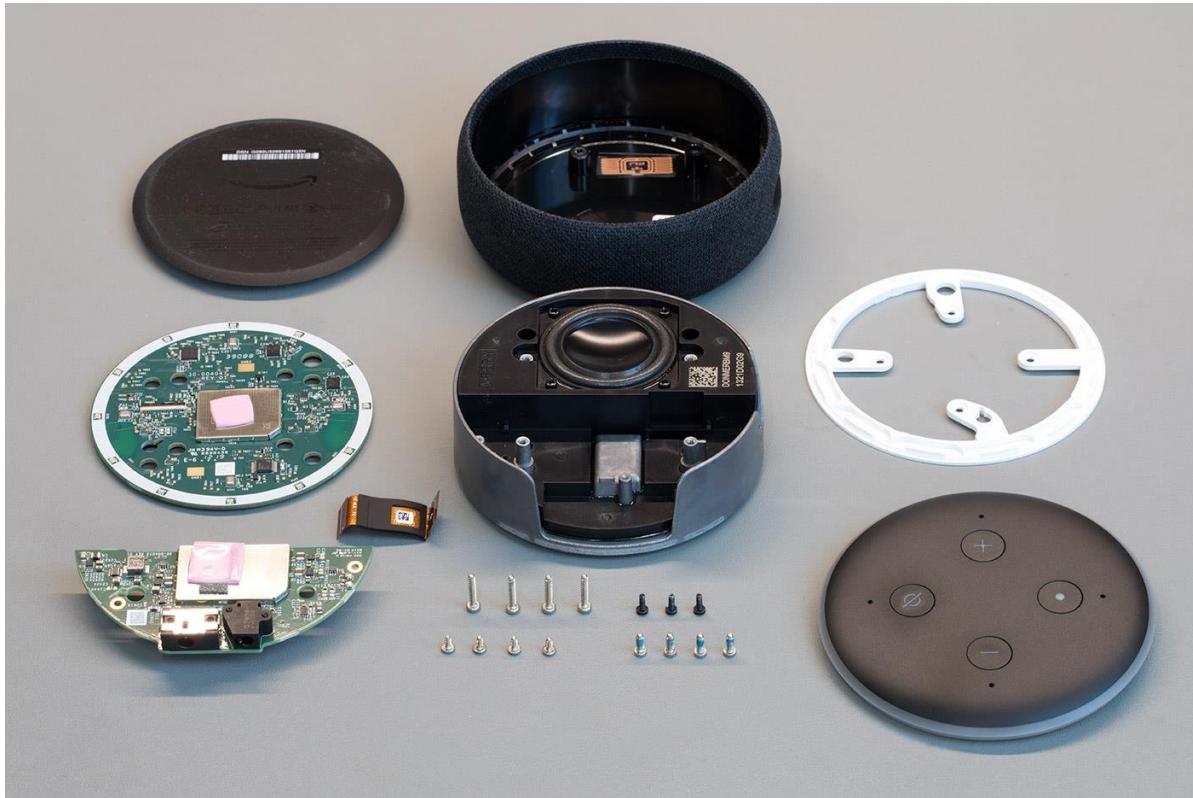
- ❖ You've probably heard of the "Internet of Things" or IoT
 - ❖ It's more of a concept than a specific kind of device.
 - ❖ Think: A giant web of connected devices that send data between each other
 - ❖ Observe: Fancy graph!

Credit:
<https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>



Embedded Systems

- ◊ Remember when we said the Desktop graph was a little misleading?
- ◊ I have some bad news
 - ◊ IoT graph is a little misleading, too.
 - ◊ IoT is primarily made of “Embedded Systems”
- ◊ Credit: <https://www.briandorey.com/post/echo-dot-3rd-gen-smart-speaker-teardown>

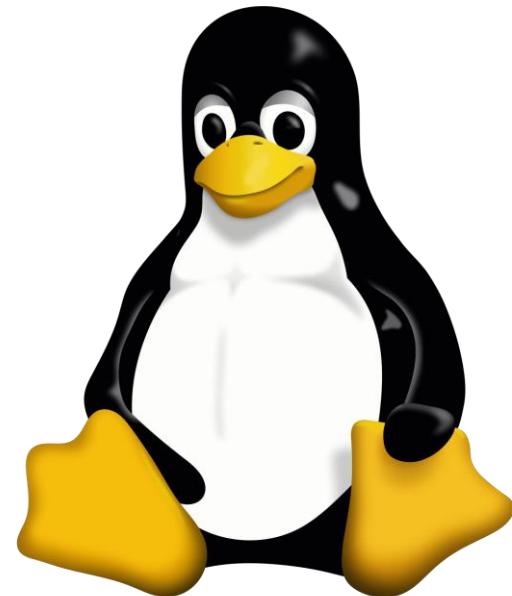


Okay...

- ❖ So what exactly are “Embedded Systems,” then?
 - ❖ It’s a type of computer system!
 - ❖ Has a processor
 - ❖ Has memory
 - ❖ Handles Input/Output (IO)
 - ❖ Usually does a *thing* or specific *things*
- ❖ Hopefully you are starting to think of a lot of things that fit that description
 - ❖ Because it is a lot of things..
 - ❖ Spoiler: It’s most things.

THAT is why we are talking about Linux*

- ❖ Some 98% of computing devices are embedded systems. That's a lot.
- ❖ And what do these systems run? Is it Windows?
 - ❖ Spoiler: NO
 - ❖ They run good, old, trusty, Linux*



*And by Linux, we mean: *NIX or UNIX-Like

Therefore

- ❖ You need to be prepared and comfortable using the Linux environment
 - ❖ Headless Mode
 - ❖ Command Line Interface (CLI)
 - ❖ Graphical User Interface (GUI)
 - ❖ More like what you're used to.
 - ❖ Think: macOS (gasp), Ubuntu, PopOS, etc.
- ❖ You must be able to configure these systems. They run the world.
 - ❖ Websites, e-mail, social media, you name it.
 - ❖ If you write software: Version Control
 - ❖ More on this in a bit.

The Basics

- ❖ The Command Line Interface (CLI)
 - ❖ Super-hacky, type stuff and do things.
 - ❖ In some cases, it does feel better. In other cases, the GUI was a much-needed change.
- ❖ Think back to last time: You type commands and things happen.
 - ❖ But how do you find what you can do?
 - ❖ How do you add new commands?
 - ❖ What's happening in the background when you do these things?
- ❖ There's a lot of circular problems here. You can dig as deep as you'd like.
 - ❖ We don't need to over-complicate this

CLI

- ❖ The CLI that you use provides you a shell.
 - ❖ There are many kinds of shells that people use.
 - ❖ Most common: BASH, sh, zsh
 - ❖ /bin/bash, /bin/sh, /bin/zsh
- ❖ A shell is just an interface to a *NIX system.
 - ❖ It takes input, executes programs, provides output.
 - ❖ Even Windows them, but I won't bore you with it! (PowerShell, DOS, CMD)
- ❖ Your shell maintains what's available to you.
- ❖ It keeps track of where you are in the File System.
 - ❖ It does lots of other things, too.

Let's take a look

- ❖ env
 - ❖ Displays all of your environment variables
 - ❖ Observe the SHELL
 - ❖ Observe the working directory (PWD)
 - ❖ Observe the HOME directory
 - ❖ Observe the USER
- ❖ Your shell must keep track of all kinds of things.

```
kali㉿kali:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/kali:@/tmp/.ICE-unix/1006,unix/kali:/tmp/.ICE-unix/
WINDOWID=0
QT_ACCESSIBILITY=1
XDG_CONFIG_DIRS=/etc/xdg
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_MENU_PREFIX=xfce-
LANGUAGE=
SSH_AUTH_SOCK=/tmp/ssh-T0Rt622ttE0G/agent.1006
DESKTOP_SESSION=lightdm-xsession
SSH_AGENT_PID=1035
XDG_SEAT=seat0
PWD=/home/kali
XDG_SESSION_DESKTOP=lightdm-xsession
LOGNAME=kali
QT_QPA_PLATFORMTHEME=qt5ct
XDG_SESSION_TYPE=x11
PANEL_GDK_CORE_DEVICE_EVENTS=0
XAUTHORITY=/home/kali/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm/data/kali
HOME=/home/kali
LANG=en_US.utf8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33
31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31
:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tar
war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.z
pg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35
:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35
01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.ASF=01;35:*.rm=01;35:*.rmvb=01
35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36
36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
XDG_CURRENT_DESKTOP=Xfce
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=kali
COLORFGBG=15;0
DISPLAY=:0.0
SHLVL=1
```

Permissions

- ❖ Let's keep it simple.
 - ❖ Just like Windows:
 - ❖ There are things your user account can do
 - ❖ There are things your user account cannot do
 - ❖ In order to do those things, you need to have more privileges
 - ❖ The highest level of privileges in Linux is the super user, “root”, user.
 - ❖ If your user has permission to do administrative things, you must elevate your privileges
 - ❖ This is done via sudo (super user do)
 - ❖ sudo <command>
 - ❖ Literally, super user do <command>

An Illustration

- ❖ Lots of new commands here:
 - ❖ Whoami
 - ❖ Id
 - ❖ Groups
 - ❖ Sudo
 - ❖ Exit
- ❖ Notice the changes in information between users

```
kali㉿kali:$ whoami
kali
kali㉿kali:$ id
uid=1000(kali) gid=1000(kali) groups=1000(kali),
kali㉿kali:$ groups
kali cdrom floppy sudo audio dip video plugdev r
kali㉿kali:$ sudo su
root@kali:~# whoami
root
root@kali:~# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:~# groups
root
root@kali:~# exit
exit
kali㉿kali:$ whoami
kali
kali㉿kali:$ █
```

Commands

- ❖ Hey, those were a lot of commands!
 - ❖ Yep! We used them last time, too.
 - ❖ A lot of the time, their name implies what they do. Other times it isn't as clear right away.
- ❖ Scenario: You need to find out all of the files in your working folder (directory)
 - ❖ Remember, folders and directories are the same thing. Don't let the words confuse you.
- ❖ The great Google has informed you that the 'ls' command is the way to go.
 - ❖ How do we find out what it does?

Manual Pages (Man Pages)

- ◊ On our system, if we type ‘man ls’ without quotes
- ◊ we are greeted with a manual page entry for the ls command.
- ◊ It provides an overview for the command, and other options you may provide to do other things with it.
- ◊ You can follow this process for nearly any command you come across in Linux
- ◊ An alternative is –h or –help e.g., ls --help

LS(1)

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION] ... [FILE] ...

DESCRIPTION

List information about the FILEs (the current directory by default).

Mandatory arguments to long options are mandatory for short options

-a, --all
do not ignore entries starting with .

-A, --almost-all
do not list implied . and ..

--author
with -l, print the author of each file

-b, --escape
print C-style escapes for nongraphic characters

--block-size=SIZE
with -l, scale sizes by SIZE when printing them; e.g., '--blo

-B, --ignore-backups
do not list implied entries ending with ~

-c
with -lt: sort by, and show, ctime (time of last modification first)

-C
list entries by columns

--color[=WHEN]

Scenario Continued

- ❖ What if that command didn't exist on your system?
- ❖ All Linux distributions keep track of all the packages installed on the system.
- ❖ You can install the package.
- ❖ The process to do this depends on your Package Manager
 - ❖ It also depends on your Operating System
 - ❖ It also usually requires elevated privileges
 - ❖ Therefore, no package installation in our Jupyter lab
 - ❖ Everything you need should be there

What are some common commands?

- ◊ ls
 - ◊ List contents of directory
- ◊ find
 - ◊ Used to find files, folders
- ◊ cd
 - ◊ change directory (move to a different directory)
- ◊ pwd
 - ◊ Print the working directory (print out where you are in the file system right now)
- ◊ man
 - ◊ Manual pages for different commands
- ◊ mv
 - ◊ Move a file to a new place with the name you specify. Be careful!
- ◊ cp
 - ◊ Copy a file to a place with a name you specify.
- ◊ cat
 - ◊ Take whatever is inside of a file and send it somewhere (like standard out so we can read it)
- ◊ grep
 - ◊ Find a pattern in some data
- ◊ echo
 - ◊ Give a string to echo and it will repeat it for you. Seems weird, but it's useful.
- ◊ mkdir
 - ◊ Make a new directory
- ◊ less
 - ◊ Print a file to standard out, but in smaller chunks
- ◊ rm
 - ◊ Remove a file (delete it). Be careful! There's no going back!!
- ◊ ssh
 - ◊ Make a connection to a remote computer
- ◊ ps
 - ◊ Look at the running processes on the computer system
- ◊ ifconfig
 - ◊ Configure interfaces. Most of the time it's to look at your IP address
- ◊ vim
 - ◊ Open a text editor
- ◊ touch
 - ◊ Create a file

Input and Output

- ❖ Frequently, you'll need to use a command and keep the output somewhere.
 - ❖ Some commands can write directly to a file
 - ❖ Most of the time, you need to utilize ‘redirection’
- ❖ Redirection operators: >,<, >>, <<, and the Pipe: |
- ❖ Let's walk through it

```
kali㉿kali:~$ pwd  
/home/kali  
kali㉿kali:~$ touch watchMeMakeAFile.txt  
kali㉿kali:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos watchMe  
kali㉿kali:~$ echo "I'm going to write to my file now!" > watchMeMakeAFile.txt  
kali㉿kali:~$ cat watchMeMakeAFile.txt  
I'm going to write to my file now!  
kali㉿kali:~$ echo "I want to add more text to my file" > watchMeMakeAFile.txt  
kali㉿kali:~$ cat watchMeMakeAFile.txt  
I want to add more text to my file  
kali㉿kali:~$ echo "oops, I overwrote it. I need another redirector to append to my  
file" > watchMeMakeAFile.txt  
kali㉿kali:~$ cat watchMeMakeAFile.txt  
I want to add more text to my file  
oops, I overwrote it. I need another redirector to append to my file  
kali㉿kali:~$ echo "i'll add ANOTHER line just for fun, so I can show what piping and  
grep can do" > watchMeMakeAFile.txt  
kali㉿kali:~$ cat watchMeMakeAFile.txt | grep "piping"  
i'll add ANOTHER line just for fun, so I can show what piping and grep can do  
kali㉿kali:~$ cat watchMeMakeAFile.txt  
I want to add more text to my file  
oops, I overwrote it. I need another redirector to append to my file  
i'll add ANOTHER line just for fun, so I can show what piping and grep can do  
kali㉿kali:~$
```

Introduction to Git

- ❖ What's Git?
 - ❖ Git is a way to track changes on a file.
 - ❖ Frequently used for “Source Control”
 - ❖ Software Development term
- ❖ What if you have 10, 100, 1000 programmers working on one project at once?
 - ❖ Need to be able to track who has done what
 - ❖ Need a way to deal with conflicts
 - ❖ Need a way to be able to collaborate separately, but maintain software in one place.

You may have heard of it

- ❖ It's used in a variety of popular products. The most common are Gitlab and Github.
 - ❖ Gitlab – Owned by Atlassian
 - ❖ Github – Now owned by Microsoft
- ❖ Git is primarily a command line utility
 - ❖ So, that's how we are going to practice it.

Common git commands

- ❖ git pull
 - ❖ Pulls all of the latest changes to the repository you are working in.
- ❖ git commit
 - ❖ Captures a snapshot of the current state of the project and the staged changes.
- ❖ git push
 - ❖ Send the files that you've added and changed and push them up to the branch that you're working on. Others can pull the changes down.
- ❖ git rm
 - ❖ Remove files from the repository (no longer tracked)
- ❖ git status
 - ❖ Shows the status of the repository.
- ❖ git clone
 - ❖ Grab a repository online and clone it locally to your system so you can build it or work on it.
- ❖ git add
 - ❖ Add your local files to the branch that you are working on
- ❖ git stash
 - ❖ Make a snapshot of your repository and store it away so you can go back to a clean repository.

Let's clone a repository

sean-wagner / **SeansGreatRepository** Private

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[main](#) [1 branch](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

 sean-wagner	Initial commit	2630444	now	 1 commit
	LICENSE	Initial commit	now	
	README.md	Initial commit	now	

README.md 

SeansGreatRepository

A Repository to show how cool Git is

git clone

- ❖ Lets talk through it
 - ❖ We need to git clone <link to repo>
 - ❖ Asks us for credentials
 - ❖ You won't see the password as you type it

```
kali:kali:~/Desktop/GitStuff$ git clone https://github.com/sean-wagner/SeansGreatRepository.git
Cloning into 'SeansGreatRepository' ...
Username for 'https://github.com': sean-wagner
Password for 'https://sean-wagner@github.com':
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Looking around

- ❖ An ls command shows a new directory where we are, the repo!
 - ❖ Let's cd into it and see what's in there.
 - ❖ ls command with the -la options shows all files with more information
- ❖ There's my README file and my license that I specified for my repo.

```
kali㉿kali:~/Desktop/GitStuff$ ls
SeansGreatRepository
kali㉿kali:~/Desktop/GitStuff$ cd SeansGreatRepository/
kali㉿kali:~/Desktop/GitStuff/SeansGreatRepository$ ls -la
total 20
drwxr-xr-x 3 kali kali 4096 Mar 13 02:25 .
drwxr-xr-x 3 kali kali 4096 Mar 13 02:25 ..
drwxr-xr-x 8 kali kali 4096 Mar 13 02:25 .git
-rw-r--r-- 1 kali kali 1068 Mar 13 02:25 LICENSE
-rw-r--r-- 1 kali kali   60 Mar 13 02:25 README.md
kali㉿kali:~/Desktop/GitStuff/SeansGreatRepository$ █
```

Let's add a file

```
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ touch myNewFile.txt
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ echo "Hello World!!!" > myNewFile.txt
echo "Hello World" > myNewFile.txt!
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ cat myNewFile.txt
Hello World
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ git add myNewFile.txt
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ git commit -m "Add a new file to the repository, to make it even more awesome"

*** Please tell me who you are.
```

Run

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'kali@kali.(none)')
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ git config --global user.email "sean@seanis.net"
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ git config --global user.name "Sean W"
kali@kali:~/Desktop/GitStuff/SeansGreatRepository$ git commit -m "Add a new file to the repository, to make it even more awesome"
[main e8d978c] Add a new file to the repository, to make it even more awesome
 1 file changed, 1 insertion(+)
 create mode 100644 myNewFile.txt
```

Let's push our changes up!

- ◊ Always do a pull before you push!
 - ◊ Someone else may be working, too.
 - ◊ It avoids massive headaches in the future, so make the good habit.
- ◊ Now, let's git push and send our new file up with what we put in it.

```
kali㉿kali:~/Desktop/GitStuff/SeansGreatRepository$ git pull
Username for 'https://github.com': sean-wagner
Password for 'https://sean-wagner@github.com':
Already up to date.
kali㉿kali:~/Desktop/GitStuff/SeansGreatRepository$ git push
Username for 'https://github.com': sean-wagner
Password for 'https://sean-wagner@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 375 bytes | 375.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sean-wagner/SeansGreatRepository.git
 2630444 .. e8d978c  main → main
```

Let's double check it worked.

	sean-wagner	Add a new file to the repository, to make it even more awesome	e8d978c	4 minutes ago	 2 commits
	LICENSE	Initial commit		18 minutes ago	
	README.md	Initial commit		18 minutes ago	
	myNewFile.txt	Add a new file to the repository, to make it even more awesome		4 minutes ago	

In Class Exercise

- ❖ Use your github account and do the GitHub Hello World Exercise
 - ❖ <https://guides.github.com/activities/hello-world/>
- ❖ This will:
 - ❖ Allow you to create and use a repository
 - ❖ Start and manage a new branch
 - ❖ Make changes to a file and push them to GitHub as commits
 - ❖ Open and merge a pull request
- ❖ Bonus resource:
 - ❖ GitHub primer from Dr. Matt Hale from University of Nebraska Omaha