THE ODIN PROJECT

What is the internet?

The internet is a worldwide network of linked computers which can communicate with each other in a specific format.

What are packets?

Packets are the information of the internet, packets are sent from the sender to the receiver in a specific format that contains all the information needed to communicate.

Differences between a web page, a web server, a web browser, and a search engine?

• A web page is the page you look at in your browser.

• A web server is the computer that has the code for how the webpage should look and behave.

• A web browser is a program on the computer that displays web pages

• A search engine is a web server that searches all other web servers for information, a user can then pull up the web page on their web browser and search for a site.

What is a client?

A client is a computer or device that connects to the internet and can request information from a server.

What is a server?

A server is a computer that receives requests (as packets) from the client and responds with the appropriate data (information)

What are DNS servers?

DNS (Domain name system) are the phone book of the internet. They translate domain names into IP addresses.

What happens when you run a search in google (explain in your own words)?

My search will go to Google’s server, this will look through other web servers, and then it will respond with the appropriate data searched.

Network (group of computers that can communicate with each other)

The group of computers have to agree on how the use the connection and how messages will be passed, on the internet these rules are called PROTOCOLS

The way that internet works is breaking the content into small pieces what we call packets. Packets contain information and instructions from where is coming and where is going.

Each computer on the internet has unique numeric address call IP addresses.

Understanding how DNS works?

www.example.com. the last dot represents the root of the internet name space

after this the browser and the operating system will first determine if they know already the IP address you are looking for, this could be stored in the DNS cache, after this the operating system is configured to ask the resolving name server for ip addresses it does not know

• The resolving name server takes the information of the IP address and starts looking for information

• The root name server takes a look at the root of the page we are looking for which is .com and save it in its cache

• TLDs (Top level domain) name server this looks at the domain name and it will save in its cache the domain name and the information processed by the root server in this case(.com)

• With the help of the domain name registrar the tld will know what ANS to go for.

• Authoritative name servers (ANS) will then find the IP address for ex 192.103.1.20 and will give it back to the Resolving name server

• Then the RNS will give the information to the Operating System which then gives the information to the web browser.

We can install ubuntu using a virtual machine or a virtual box

- Installing ubuntu dual-boot in a windows machine is very good for resume information ( This we will need to go through it)

The black screen in the known as the command prompt or the terminal in Mac is known as the command line interface (CLI)

Coomands:

Pwd(present working directory) —> will show the present working directory <username>

Whoami—> will give the <username> of the system

ls —> will give a list of documents, folders, files of what is in the current directory

touch —> is used to create a file or files ex: touch ex.html ex.js ex.css

rm —> use for removing

pwd -P —> If we want to see the actual physical path, with all of the symlinks resolved, we can use the -P flag.

ls —> listing for different directories is possible by following the ls with the path ls /usr/local/lib/node\_modules

ls -a —> will list all files including hidden files (hidden files that are preceded with a .(dot)

ls -l —> for long form listing will show more detailed info of the files such as: file mode, number of links, owner name, group name, number of bytes, abbv month, day of month file was last modified, hour of last modified, and the path name

ls -lh: will show us the long listing form plus human readable sizes.

ls -lhS: will sort the same list by file size from big to small.

ls -lt —> will sort by last modified time

ls -lr —> will apply reverse sorting for ls result alphabetically

ls -lSr —> will apply reverse sorting but base on the file size.

ls -ln —> this is the link command, Links allow us to create an association from one file or directory to another. A link allows us to “point” one file or directory to another. Links can be either hard or symbolic.

Hard links: Hard links create an identical copy of the linked file on disk, that gets updated automatically as the source file gets updated. That means if the content of the source is changed, so will the target file.

Symbolic links: ln -s —> We have seen that the default type of link that gets created when using ln is that of a hard link. Unfortunately hard links do not work for directories. To create a link to a directory, we can use the -s flag to create a symbolic link. This can also be used for linking to files as well, not just directories.

cd —> change directories. Ex cd ~/Documents will take us to the documents directory, /Users/<username>/Downloads

cd .. —> will navigate up (to the parent directory)in the previous case will be /Users/<username>/

cd —> cd by itself will take us to the home directory

mkdir —> is used to create a directory (make directory)

mkdir -p —> will create intermediate directories ex mkdir -p a/b/c

Mkdir -v a —> The mkdir command supports a verbose mode flag, -v. Adding the -v flag will print the results of mkdir to the console. This can provide useful reporting when writing scripts that will execute a lot of commands for you

cp —> use for copying files the 1st argument is the file we want to copy and the 2nd argument the target file.

When trying to copy directories we will get an error, to avoid this we can use the -R flag to recursively copy the directory’s content to the target directory

f —> -f command is use to force actions

i —> -i command is use to confirm overwriting of a file

rm —> remove command used to delete files and folders and it supports the same flags as cp command

mv--> the moving files command use to move files, it supports the same flag as cp command and it acts like cp rm together. mv a.txt b.txt —> result will be b.txt (a.txt will be copied and move to b.txt)

Git and Github

Git is a very popular version control system.

GitHub is a service that allows you to upload your code using Git and to manage your code with a nice web interface. GitHub and Git are not the same thing or even the same company.

git config --global user.name “UserName

git config --global user.email “email@gmail.com

git config —-get user.name —> this should return your username

Git init —> git init repo-directory —> this will initialized a repository from existing code

The GIT structure:

Working Directory —> Staging Area —> REPOSITORY

Files that are local —> files that are added —> files that are committed to repo

Git add -A will add all the files from local to staging area

Git add example.css will add that specific files

Git reset —> this command will remove the files added to the staging area

ex: git reset example.css or git reset -A

Git commit —> this command will commit to the repository all the files added previously

ex: git commit -m “changes to example.css to include blue bg”. —> the -m stands for message

Git log command will give us information about history of actions in git

Git clone command is to bring the current code for that repository to your local —> git clone <repo url> <where to be cloned>

Or git clone ../another-local-repo . —> the dot at the end means cloned into my CWD (Current working directory)

Git remote -v this command will give us information of the remote repository

Recommended processing of git actions:

1. Git diff —> check the differences on the repo

2. Git status —> checks the status of the cwd

3. Git add —> to add all -A or specific files ex.css

4. Git commit -m —> to commit the changes made

5. Git pull origin master —> this will pull the lates changes in master to be up to date with any changes the team made

6. Git push origin master —> this will push the lates changes you made to the master repo

Creating a branch

1. Git branch BranchName

2. Git checkout BranchName

The command git branch will list all the branches

Pushing changes to a branch

Git push -u origin BranchName

From branch to Master

Git checkout master —> this is the way we change from BranchName to master

Git pull origin master —> will pull al the latest changes from master

Git branch —merged —> this will show a list a branches that are merged

Git merge BranchName —> will merge the branch

Git push origin master —> will push the changes implemented on BranchName to master

After pushing to master - Deleter the BranchName

Git branch —merged —> this is to confirm that the branch was merged successfully

Git branch -d BranchName —> this will deleter the branch from local

Git branch -a —> will show us information on the remote repository and if BranchName still showing there then we need to the delete BranchName from remote repo

Git push origin —delete BranchName —> this will delete the BranchName from our remote repo

FRONT END DEVELOPMENT:

What is front end development?

It’s a mix of programming and layout that powers the visuals and interactions of the web.

 If a website were a house, front-end web development would be \_\_\_\_\_\_?

The pretty exterior that gives the house character, or the host that invites guests in and makes them feel at home.

What’s your favorite part about programming/coding?

That there are almost always multiple ways to solve a single problem, or achieve any specific functionality. On the outside I think programming can seem like a very prescriptive, direct, binary kind of job. But there’s actually a lot of creativity and ingenuity involved.

HTML, meaning “HyperText Markup Language”, is a document format used for defining the semantic structure of a single web page. One could say that HTML is what the internet is made of: All the websites that we are looking at every day are all defined (described) as HTML.

If you look at this document you’ll notice the recurring pattern of “tags” that start with <something> and then are closed with </something>. E.g. the entire document starts with an opening <html> tag, and ends with a closing </html> tag.

HTML entirely consists of these tags that have a certain meaning, can be nested, and contain content.

Here’s what the HTML tags used in this example mean:

• <html>...</html> - the HTML document as a whole

• <head>...</head> - the header of the document, containing meta information (i.e. information *about* the document, not *part of* the document itself)

• <title>...</title> - an example of one bit of meta information, the title of the page as displayed in your browser history, and the browser window title (or tab)

• <body>...</body> - the body of the document itself, i.e. the whole of its content

• <h1>...</h1>, <h2>...</h2> - a headline level 1, and level 2, containing the headline’s text. HTML defines heading levels 1-6, which should be enough to define the structure even of large documents.

• <p>...</p> - a single paragraph, containing the paragraph’s text.

• <ul>...</ul> - an unordered list (i.e. a list that uses bullet points, as opposed to, e.g., a numbered list)

• <li>...</li> - a single list item, must be contained in either a <ul> or <ol> tag