The Odin Project

Assignments

**Written by: Franco Andrés Ramos**

Foundations Course

How Does the Web Work, lesson 7:

What is a web server?

A server is the place of the internet where requests from clients go

What is a network?

A network is an interconnected number of computers that work together and can be either private or public

What is the internet?

The internet is the infrastructure in which the Web is built, an immense number of routers directing packages from servers to clients back and forth, displaying what the client needs

What is an IP address?

An IP address is the particular number that’s assigned to every computer using the internet, and also applies to websites, in these cases, the IP address is simplified and converted to a domain, like Google.com and every single other website in existence

What is a router?

A router is one part of the internet, it’s a mini computer that signals requests from the client to a server, this signal encounters much more routers in its path

What is an ISP?

Abbreviature for “Internet Service Provider” like Tigo or Entel where I live

What are packets and how are they used to transfer data?

Packets are the answer to requests from clients all around the world. These packets are information, such as YouTube videos, broken down in pieces so the page/website can load faster and more efficiently

What is a client?

A client is a computer in one place on the world making requests to servers through browsers and with the help of ISPs

What is a server?

A server is the physical PC someplace in the world that receives requests from clients through ISPs and if the server approves said request, the page is broken in small packets and rebuilt with **HTML** instructions with **DOM** trees, **CSS <links>** which design the page with a **CSSOM** structure, compiles and executes **JavaScript’s <script>,** all of this to display a page or a website

What is a web page?

A web page is the response from the server, these conform web sites as a whole, much like a book, a page is one part of a site, and a site (being a book in my metaphor), can be accessed through the server, which would be the library.

What is a web browser?

A web browser is **a software that retrieves and displays web pages**, these being organized and located with a **DNS (Domain Name System),** the browser is the infrastructure, while the Search Engine is the service.

What is a search engine?

A search engine is a service, provided by many (such as Google, DuckDuckGo, Mozilla, etc.) which allows a client to communicate with servers all around the world, this runs thanks to a Web Browser, which is the infrastructure its built upon.

What is a DNS request?

A Domain Name System (DNS) request it’s the web browser request to retrieve a web page or site from the web server by requesting an IP number for a domain, and lets users go through pages using hyperlinks.

Which browser are you currently using?

Google Chrome, thanks for asking, never felt comfortable with Firefox or worse, Edge, but that’s to be expected.

In your own words, explain what happens when you run a search on google.com.

The Search Engine uses the Web Browser to send a request to the Web Server thought the Domain Name System, then the web server approves this request and sends the webpage back in small packages to be rebuilt using HTML, CSS and JavaScript (which are the more basic web building tools, I might add)

Installations, lesson 8:

Step 1. Download VM and Xubuntu

1.1 step: Download the latest version of Virtual Box for Windows Hosts

1.2 step: Download Xubuntu

Step 2. Install VM and set up Xubuntu

2.1 step: Install Virtual Box

2.2 step: Prepare VirtualBox for Xubuntu

Step 2.2.1: Unattended Guest OS Install Setup

Step 2.2.2: Modifying Hardware config

Step 2.2.3: Virtual Hard Disk

Step 2.2.4: Begin the Unattended Installation

Step 3: Setting Correct sudo Permissions

Step 3.1: Navigate to Users and Group

Step 3.2: Manage Groups

Step 3.2.1: Add Yourself to sudo

Step 3.3: Reboot Your VM

Step 3.4: Test Your Newly Gained sudo Privileges

Step 4: Understand Your New VM

Step 5: Safely Shutting Down Your VM

Option 1 - Shutting Down from Inside the VM with UI

Option 2 - Shutting Down from Inside the VM with the Terminal

Option 3 - Shutting Down from Outside the VM

2: Google Chrome Installation

**Step 1: Download Google Chrome**

#### Step 2: Install Google Chrome

#### Step 3: Delete the Installer File

#### Step 4: Using Google Chrome

Done

**Shell Novice Knowledge Checks:**

**Introducing the Shell**

What is a command shell and why would I use one?

A command shell is the **CLI (Command-Line Interface)** as opposed to the normal **GUI (Graphical User Interface)** which is the normal, everyday interface we use. The main advantages of a CLI over a GUI is that it allows automation of tasks, which is exactly what I want for my business. Furthermore, knowing the commands is important for learning to code, and everyday life. Also bragging rights.

-------------------------------------------------------------------------------------------------------------------

**List of commands so far (important):**

**$ ls = List Files, shows files in the current directory**

**$ cd = Change Directory, changes your current directory to any folder**

**$ cd (dir. name) = Goes to specified directory without writing everything**

**$ cd .. = Goes up a folder on your current directory**

**$ cd - = Goes to the previous stated folder**

**$ cd ~ = shortens your current directory to ~ so it saves you some time typing**

**$ pwd = Print (current) Working Directory, aka lists the directory you’re in**

**$ man = Manual, on the most literal sense, shows the meaning of a command**

**$ start = Open a program, a docx, pdf, etc. File with the directory you type**

**$ mkdir = make directory, aka create a new folder on the current directory**

**$ nano\* (name).txt = opens a simple text editor that works with plain data:**

**\*Nano is associated with Linux’s CLI, and can be used in Windows thru Git Bash**

**\*If you see a ^ symbol on the bottom, it means “ctrl”**

**$ cat = opens and writes the .txt file on the CLI. Comes from concatenate.**

**$ rm = remove, as its name, removes a file from a directory.**

**$ mv = moves a file to a new directory “mv dir/mew.txt newdir/mewtwo.txt” \***

**\*Giving it a new name changes the txt name, in a different or in the same dir**

**\*Be careful giving it the same name as another file, as the new one will overwrite (aka delete) the old one**

**$ ../ = Goes up a level in the directory hierarchy**

**$ ../.. = Goes up TWO levels in the directory hierarchy**

**$ cp = copy, copies the file or directory (with its contents) to another stated directory**

**$ open = opens a file**

**$ wc = Word Count, counts the number of words, lines and characters, returning this count from left to right\***

**\*Use the -l option to see only the number of lines**

**\*Use the -m option to see only the number of characters**

**\*Use the -w option to see only the number of words**

**$ > = Tells the output of a command to be written on a file. Usually a text.txt file, it’s incredibly useful to save given information for later. For example:**

**$wc -w \*.pdb > words.txt # (creating a text file with the output)**

**$ >> = Instead of writing the output on a file, doing this 2 or more times in a row with the same textfile.txt name WRITES THE OUTPUT TWICE OR MORE TIMES IN THE SAME TEXTFILE.txt, this is extremely useful.**

**$echo hello >> Texto.txt (x3 times)**

**Texto.txt: hello**

**hello**

**hello**

**$ less = Use this command with a text.txt file, shows a small screenful of said .txt**

**$ sort = Sorts the numbers on a .txt file on alphanumerical order, with the -n option it sorts them on a numerical order.**

**$ head -n (input a number) (.txt) = shows the first numbers on a txt file according to the input number.**

**$ tail -n (input a number) (.txt) = shows the last numbers of a txt file according to the input number.**

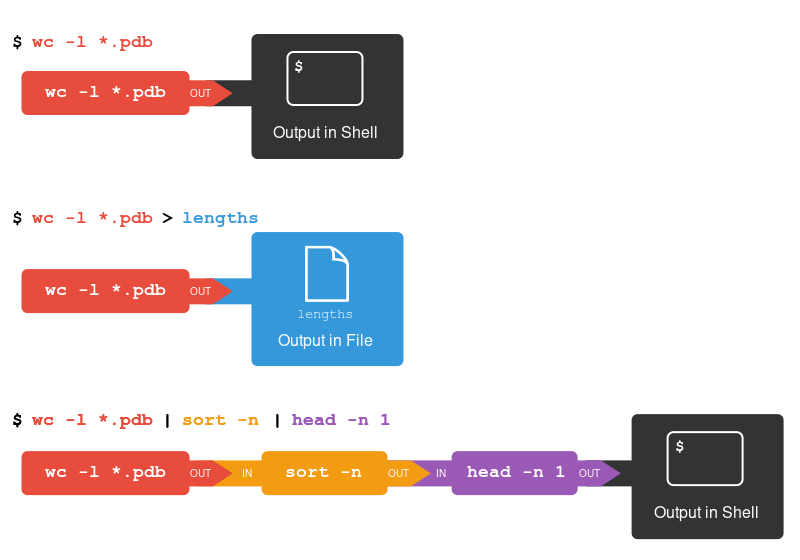
**The “|” Command.**

**This is one of the most important and useful commands you’ll have to learn, so bear with me.**

**The | command is called a “pipe”, it’s used to take the output from one command and use it as an input for another \*, and you’re not limited to a single pipe, you can use as many as you want. Let’s see it in action:**

**$ sort -n numbers.txt | head n -7**

**This opens the “numbers” text file, sorts them numerically, and sends the sorted numbers to the head command, which shows the first (6) numbers sorted.**

**Illustrative image:**

**-------------------------------------------------------------------------------------------------------------------**

**$ cut = cuts certain sections of each line in the file and expects them to be separated by columns by a tab character. Goes with the -d option to specify a certain character as our delimiter and the -f (x) option to extract x “field” (column) (x=number)**

**$ uniq = filters matching lines in a the columns of a file, used alongside the -c option, which stands for count, counts the number of times a word, character or line is repeated on the file.**

**$ . = this command is incredibly important but native only to Bash. It allows quickly execution of commands inside .txt files. This allows a fuckton of things, really.**

List of options to use alongside ls

--help : Shows instructions on how and when to use these commands  
-a : Include hidden files and directories (those starting with a dot).

-d : Show directories themselves, rather than their contents.

-F : Append an indicator character to each listed entry to indicate its type. For example, a "/" character denotes directories. Helpful showing the files on your current directory

-h : Print file sizes in human-readable format (e.g., 1K, 234M, 2G).

-I : Print the inode number of each file.

-l : Display detailed information, including file permissions, ownership, size, and modification time.

-r : Display the list in reverse order.

-R : Recursively list subdirectories and their contents.

-S : Sort files by size, largest first.

-t : Sort files by modification time, newest first.

-u : Sort files by access time, most recently accessed first.

-FR : Recursively lists the new directories

List of options to use alongside mkdir

-p : stands for “parent”, creates folders to access to the directory you’re creating, useful to create multiple nested directories in one go.

List of options to use alongside mv

-I : stands for “—interactive”, will ask for confirmation when moving a file to another directory

. : putting this at the end of the new text allows you to move a file, such as text.txt into the current directory\*:

\*IMPORTANT NOTE: To efficiently move from up a folder to the next folder, use the ../ command, like this: “mv ../texto.txt.”

List of options to use alongside rm

-i : stands for interactive, gives you a y/n option if you are sure you want to delete this file or directory, this decision can’t be reversed.

-r : stands for recursive, it allows deletion of directories, be CAREFUL using it

List of options to use alongside cp:

Wildcards: Wildcards are very important, they allow mass copying of files with shared characteristics, let’s see how:

\*: \* stands for zero or more other characters, this is extremely useful to highlight the shared component each file has on their names. For example:

recipes\_for\_dog\_food.txt

Vulkan\_lives.txt

Horses\_horsing\_around.webm

recipes\_for\_cat\_food.txt

recipes\_for\_pet\_amebas.txt

/Recipes\_only\_nothing\_else\_officer

if you want to copy ONLY the recipe files, you use the \* wildcard like this:

**$cp recipes\*.txt recipes\_only\_nothing\_else\_officer**

This allows for quickly mass copying from one folder to another, this could be used like \*.pdf to select all .pdf files.

?: The ? wildcard is similar to \*, except it’s used to represent a SINGLE character, in that case, we would write something like this:

**$cp recipes\_for\_????????.txt recipes\_only\_nothing\_else\_officer** This does copy recipes to the folder, but only **recipes\_for\_dog\_food.txt** and **recipes\_for\_cat\_food.txt** since they share the same text length. It won’t copy **recipes\_for\_pet\_amebas.txt**, you’d need to add more ?s. Be specific

List of options to use alongside sort:

-n : Stands for “numerical”, sorts the order of numbers on the .txt file to numerical order instead of alphanumerical. To add, use this along with the command **$head** or **$tails** to show the first or last numbers respectively on a .txt file.

List of options to use alongside wc:

-l : shows lines on a txt file

-m : shows characters on a txt file

-w : shows words on a txt file

-------------------------------------------------------------------------------------------------------------------

How can I move around in my computer?

To move around your computer, you can use the cd command and type as follows:

$**cd ­/home/(‘username’)/Desktop/(‘folder name’)**

This allows you to access any folder given the direction from the root, which is indicated by the /home/ specification. You can also use cd .. to go up a folder on your current directory.

How can I see what files and directories I have?

By using the ls command, you can see the current files on your computer depending on your current directory. Your default directory is always set on something like ‘/Users/Frank/’ or ‘/home/fronk’. To move around, use cd .. to go up a folder, to go back to a previously used directory, use cd -, and to simplify, your default directory can be written and summarized with ~ instead.

How can I specify the location of a file or directory in my computer?

To specify the location of a file or directory, write the direction in which it’s located, for example, if you want to set your current directory to your downloads folder, and access its contents from the CLI, type “cd C:\Users\(Frank)\Downloads”

-------------------------------------------------------------------------------------------------------------------

***EXTRA QUESTION NOT INCLUDED IN THE PROGRAM.***

What’s the difference between an absolute path and a relative path?

Absolute paths are written in code from the root directory to an exact directory you wish to use, they are written as such /home/fronk/Desktop/Unix\_Shell

Relative paths change every time you use the cd (change directory) command, describing the path starting from the current location rather than the root directory, for example ./documents/file.txt

-------------------------------------------------------------------------------------------------------------------

How can I create, copy, and delete files and directories?

You can create directories using **mkdir**, and a directory hierarchy by using -p

You can create files (.txt so far) with Nano. Type **nano** textname.txt and start editing

You can copy directories with all its contents using the **-r** option: **“$cp -r thesis thesis\_backup”**

You can copy files with the **cp** command: **“$cp textname.txt thesis/textname.txt”**, note that you need to be on the directory with the file on it. \*

\*IMPORTANT:

You can delete directories using the rm command, type rm and the directory name alongside it

You can delete files with the same method above, note that you need to be on the directory with the file on it. \*

**\*Use the -i option to be sure if you really are deleting the correct file**

How can I edit files?

So far, you can edit text files (not pdf files!) by using the nano command, nano is a simple terminal-based text editor, just like this:

**$ nano textfile.txt**

Use Ctrl+O to change and save the name, Ctrl+X to exit nano.

How can I combine existing commands to do new things?

To combine different commands, there’s a useful neat thing called pipes and filters.

A **pipe** is the “|” command that, as previously stated, uses the output of a command for the input of a next command, check the pipe page for a further explanation.

A **filter** is a command like **$wc** or **$sort** that transforms a stream of input into a stream of output, in simpler terms, they convert information based on the instructions you give them.

A beautiful line I love from the sw-carpentry.github page states the best way to use pipes and filters:

“The **key** is that any program that reads lines of text from standard input and writes lines of text to standard output **can be combined with every other program that behaves this way as well**. **You can and should write your programs this way** so that you and other people can put those programs into pipes to multiply their power.”

Let’s deconstruct this, the example below is taken from the sw-carpentry page

This is the animals.csv file:

-------------------------------------------------------------------------------------------------------------------

2012-11-05,deer,5

2012-11-05,rabbit,22

2012-11-05,raccoon,7

2012-11-06,rabbit,19

2012-11-06,deer,2

2012-11-06,fox,4

2012-11-07,rabbit,16

2012-11-07,bear,1

-------------------------------------------------------------------------------------------------------------------

Then, we use the following pipeline:

**$ cat animals.csv | head -n 5 | tail -n 3 | sort -r > final.txt**

We’ll go step by step:

**1: cat animals.csv** reads the .csv file

**2: head -n 5** then takes the first 5 lines (\*)

**3: tail -n 3** then takes the last 3 lines from the head output

**4: sort -r** then sorts them alphanumerically in the reverse order

**5: > final.txt** puts the final output on a .txt file

(\* **-n** stands for number of lines)

The result looks like this:

2012-11-06,rabbit,19

2012-11-06,deer,2

2012-11-05,raccoon,7

*With enough knowledge and practice, the possibilities are endless.*

To further ado, let’s look at the **$cut** command, which is very important to specify what we want to extract from a file.

**$ cut -d , -f 2 animals.csv**

Step by step:

**1: cut -d** , specifies that we want to use “,” as our delimiter character. (\*)

**2: -f 2** specifies we want to take and display the second **field** (column)

\* this makes it so it’s delimited like this:

2012-11-05 , **deer** , 5

2012-11-05 , **rabbit**  , 22

2012-11-05 , **raccoon** , 7

2012-11-06 , **rabbit**  , 19

2012-11-06 , **deer** , 2

2012-11-06 , **fox** , 4

2012-11-07 , **rabbit** , 16

2012-11-07 , **bear** , 1

The result looks like this:

**deer**

**rabbit**

**raccoon**

**rabbit**

**deer**

**fox**

**rabbit**

**bear**

**Tab Autocompletion**

**Programmers are lazy, and so is the author (who’s not a programmer yet, but he’s lazy nonetheless)**

**That’s why this tip is so important to learn, trust me it’ll save you pain. When pressing “TAB” it has autocomplete on the text you’re writing if there’s only one option. Say, something like this:**

**You’re currently on the Desktop/ folder**

**And you want to access to /TOP/basics/Javascript/tricks**

**Start typing $cd T**

**And press the TAB button.**

To remember.  
Another important command CLI is “.” which allows summary execution of commands  
This command is only native to Bash.

Knowledge checks

What is the command line?

A Command Line Interface (CLI) is a terminal based interface that runs commands from either the user or various programs. As opposed to a Graphical User Interface (GUI), it uses plain text.

How do you open the command line on your computer?

On Linux based OS, open it with **ctrl+alt+T** on windows, I prefer to use CMder

How can you navigate to a particular directory?

By using cd and choosing the directory you’d like to be in. Use tab to make the process much shorter

Where will cd on its own navigate you to?

Back to the first directory your started in.

Where will cd .. navigate you to?

Up a directory.

How do you display the name of the directory you are currently in?

By typing pwd (Print Working Directory)

How do you display the contents of the directory you are currently in?

By typing ls. Fun fact, typing ls -R shows the content on every carpet in the directory you’re in, so it may take a while to load.

How do you create a new directory?

By typing mkdir (directory name)/

How do you create a new file?

A file, like a .txt document can be created typing nano on the current directory you want the file to be in. Keep in mind you can create multiple types of documents with the correct .end

How do you destroy a directory or file?

By typing -rm, this stands for remove, and removes any **file.** If you want to delete directories instead, use the option -r, which will permanently delete any directory.

How do you rename a directory or file?

Two ways (maybe more) by using nano if it’s a txt file and renaming it with ctrl+c. The more easier way is to move it and changing its name in the process.

What is Git?

Git, simply put, is a DVCS (Distributed Version Control System) that began in 2005 as the main way for the Linux team to save, take snapshots and storage versions of their work on a decentralized and more secure way.

So, here comes the knowledge check for Git.

What kind of program is Git?

Git is the most popular DVCS in the world, more than 90% of the world’s projects work with Git, and it comes with Git bash, the CLI used for storing projects and more.

What are the differences between Git and a text editor in terms of what they save and their record keeping?

Git saves the changes done to the project vertically, on a tree architecture that saves files on different states, these are modified, staged and committed.

Modified.- Modified means you worked on the project and modified files but didn’t commit them to your database yet.

Stagged.- Staged means you marked a modified file in its current version and is ready to be committed to the next snapshot.

Committed.- Committed means the data is safely stored in your local database

Does Git work at a local or remote level?

Both. Since it’s a decentralized, distributed system, every team member has a copy (aka snapshot) of each stage of development, and can request the copy from one another in case of data loss or corruption.

Does GitHub work at a local or remote level?

Also both, since you need a SSH key to upload your works to it’s platform, save snapshots and synchronize your work with your GitHub portfolio.

Why is Git useful for developers?

It’s a DVCS, it’s pretty much essential, allowing snapshots to be kept secure and to save multiple stages of development.

Why are Git and GitHub useful for a team of developers?

For a team of developers, those are both essential and useful, Git saves headaches when it comes to data handling and different stages of development, Github provides a platform for developers to communicate and work together much more efficiently.

GIT COMMIT AND GIT PUSH, TWO BIG DIFFERENCES.

First, I had problems differentiating one from the other. But in this small paragraph you’ll see the difference explained in a very simplified manner.

Git commit saves your work files in your LOCAL git repository, I cannot stress how important the local part is, by previously staging them and then committing them, you’re making sure to save your files on your local repository, and then saving them on the remote repository (which is Github)

Git push (origin)(main) saves the previously committed files on the Github repository, origin and main are used to specify the route in which these files are located.

Origin is the default name for the remote directory from which you copied the file, aka the Github repository

Main refers to the default branch name in git, this branch is expected to be the primary branch of development, where the latest code version is expected to reside.

-------------------------------------------------------------------------------------------------------------------

Git Cheat sheet:

This is a reference list of the most commonly used Git commands. (You might consider bookmarking this handy page.) Try to familiarize yourself with the commands so that you can eventually remember them all:

Commands related to a remote repository:

* git clone git@github.com:USER-NAME/REPOSITORY-NAME.git
* git push or git push origin main

Commands related to the workflow:

* git add .
* git commit -m "A message describing what you have done to make this snapshot different"

Commands related to checking status or log history

* git status
* git log

The basic Git syntax is program | action | destination.

For example;

* git add . is read as git | add | . , where the period represents everything in the current directory;
* git commit -m "message" is read as git | commit -m | "message"; and
* git status is read as git | status | (no destination).

-------------------------------------------------------------------------------------------------------------------

Tip: To open the inbuilt terminal on VScode, type ctrl + `

Knowledge Checks (Git Edition!)

How do you create a new repository on GitHub?

By clicking the “New Repository” button on Github’s top right corner

How do you copy a repository onto your local machine from GitHub?

First click on code, then local, click on SSH and copy the URL, now use the command git clone, followed by the copied URL and you should have a successful copy of the repository in your machine, granted if you connected both first.

What is the default name of your remote connection?

The default name is Origin, you can change this later

Explain what origin is in git push origin main.

Origin represents your computer, in git push origin main, it specifies your main folder as the one exporting the branch to your Github’s repository.

Explain what main is in git push origin main.

Main is the chosen folder for exporting the edited files to the Github repository

Explain the two-stage system that Git uses to save files.

Simply put, there are two stages before committing a file to the Github repository.

1st stage is modified, when a file on the project branch has been modified.

2nd stage is staged, when a file is staged, it’s ready to be committed to the Github repository.

How do you check the status of your current repository?

By entering the command git status which shows the status of the files on the current folder.

How do you add files to the staging area in git?

By typing the git add command, you prepare your files to be committed on your local repository with the git commit command

How do you commit the files in the staging area and add a descriptive message?

With the same command you use to commit files from the staging area, insert a message, take this for example:

git commit -m "Your descriptive commit message"

This should allow you to add a descriptive message while committing the file, and check it afterwards with git log. Remember to use this to keep track of your meaningful commits.

How do you push your changes to your repository on GitHub?

With git push.

Kinda self-explanatory.

How do you look at the history of your previous commits?

With git log, you can also see the history of past name changes done.

HTML and CSS Basics.

Knowledge Check:

What do HTML and CSS stand for?

Hypertext Markup Language and Cascade Style Sheets

Between HTML and CSS, which would you use for putting paragraphs of text on a webpage?

HTML, as it’s used to build the core components (which include text) on the webpage

Between HTML and CSS, which would you use for changing the font and background color of a button?

CSS, as it’s used to design and build the “style” of a webpage

What is the difference between HTML, CSS and JavaScript?

HTML.- Used to build the infrastructure and the bases of a webpage

CSS.- Used to stylize and give the website a design according to its needs

JavaScript.- Used to add interaction to the website with embedded music, videos, and more

What is an HTML tag?

An HTML tag is used to mark the first elements of an HTML page, presented using angle brackets <> (alt+60, alt+62)

What are the three parts of an HTML element?

The three parts are:

* The opening tag <p>
* The content inside it <p>callaoelhocicomamawebo</p>
* The closing tag </p>

HTML elements

<!DOCTYPE html>

<”lang=en”>

<Head>

<meta charset=”UTF-8”>

<title>inserteuntitulobienmamonaquí<title>

<h3>contenidosfw<h3>

-------------------------------------------------------------------------------------------------------------------

**Neat little trick in VScode.**

**After already specifying that the file is an .html extension, type ! and press enter.**

**This will show you an already made HTML boilerplate, from which you can just type and fill the blanks.**

-------------------------------------------------------------------------------------------------------------------

What is the purpose of the doctype declaration?

It’s to determine the type of HTML version the web browser should use to render the webpage. The latest version at the time of writing this document is HTML 5

What is the HTML element?

The HTML element defines the start of the whole webpage, it is marked with <> and ends the webpage on </html>, also comes with the lang= element, to provide the language for the browser to read on the page

What is the purpose of the head element?

It’s the part where we put rendering information and meta information in our webpage to be shown correctly. Do not put any element that displays content on this element.

By meta information, we mean <meta charset="UTF-8">, which accomplishes the encoding task. Without this, we wouldn’t be able to use special characters, charset stands for Character Set, and UTF-8 is the most widely used encoding scheme.

What is the purpose of the body element?

The body element is the central key part of building webpages. It’s where content will be displayed, be it texts, images, links, sound, and so on.

-------------------------------------------------------------------------------------------------------------------

Next up.- Text tags list for HTML

<p></p>: This is the paragraph element, without these, the web browser would display everything written inside as a long single line

<h1></h1>: Headings, these mark the headlines or the start of a reading article, keep in mind, these can go up to <h6>

<strong></strong>: strong puts the text in **bold**, this is done to mark important text parts of a website that need extra attention from the client.

<em></em>: em puts the text in *italic*, done to mark certain words to the client.

<!-- These two signs are used to leave developer comments, as in, if you write something inside these symbols, it won’t display on the final page -->

<a href=”**website.com**”> </a>: used to put a link to another website on a text, useful when you make multiple pages

<img src=”**/smtn.png**”>: used to insert an image to the website, useful for self-explanatory reasons.

* Btw, **Links and Images**

If the page or image you're referring to is already in your Neocities web directory, you can call it by its filename: for example, index.html or cat.png.

Otherwise, use full URLs like this: <http://neocities.org/img/heartcat.png>

<ul>: adds a list to your website, needs one of the two below:

<li>: makes a column using circles.

<lo>: makes a numbered column using, well… numbers!

Knowledge check

How do you create a paragraph in HTML?

By using the <p> tag. It signalizes paragraphs and should be used accordingly. Remember to close the tag with /

How do you create a heading in HTML?

To create a heading, use the <h1-6> tag, from one thru six, indicating the font size, 1 is very big and 6 is very small.

How many different levels of headings are there and what is the difference between them?

Already explained above hahahahaha I’m so fucking cool going the extra mile

What element should you use to make text bold and important?

Use the <strong> tag, this puts the bold in **bold,** remember to close it with a /

What element should you use to make text italicized to add emphasis to it?

Use the <em> tag, this puts the italic in *italic* as always.

What relationship does an element have with any nested elements within it?

It contains information for the website to display, the nested element is the parent and the nested elements inside it are its children, yeah it’s a weird comparison but bear with me, it makes it much simple to understand later on.

What relationship do two elements have if they are at the same level of nesting?

They become the siblings, for example, multiple <p> tags at the same parent aka the <body> element.

How do you create HTML comments?

By typing <!-- insert comment here --> you can get a comment inside the code that won’t show up

What HTML element is used to create an unordered list?

It uses the <li> tag

What HTML element is used to create an ordered list?

It uses the <lo> tag

What HTML element is used to create list items within both unordered and ordered lists?

Now, this is the important part, you need the <ul> tag BEFORE you even type the previous two <li><lo> tags.

Two kinds of links.- There are two types of link, relative and absolute links

Relative.- relative links refer to links inside our own website, these don’t include the domain name, since its another page inside the website, it assume we use the same domain. It’s called a relative link due to the next page being relative to its original counterpart.

Absolute.- absolute links are the opposite, these contain links to other websites, as such, these include the domain protocol.

HOW TO PUT IMAGES ON YOUR WEBSITE.

This is the real meat of the learning, HOW TO PUT IMAGES ON YOUR WEBSITE, well, <> there’s the perfect tool for that. And it works in two ways! Let’s see ‘em.

1.- <img src=”./images/dog.jpg”> .- Shows an image from a local pc folder. Be sure to write the direction correctly with ./

2.- <img src=”<http://neocities.org/img/heartcat.png>”> .- Shows an image from a specific link on the internet.

Knowledge check:

What element is used to create a link?

<a href=””> is used to create a link to another page in your own webpage

What is an attribute?

An attribute is basically a command. It has both a name and a value.

What attribute tells links where to go to?

The value inside href is where we want our link to go.

What is the difference between an absolute and relative link?

Absolute links send you to another website, relative links send you to another page inside the website

Which element is used to display an image?

<img src=””> is used to display an image that’s either saved in your pc or taken from another website

What two attributes do images always need to have?

An alt=”” attribute and a direction to where they’re located

How do you access a parent directory in a file path?

With the ../”carpet-name” command.

“../” tells us to go a folder up

“carpet-name” tells us to go to such carpet

And then you type the image name and it’s format

What are the four main image formats that you can use for images on the web?

JPG, PNG, SVG and GIF