🌐

Web Fundamentals Notes

Decode URL Parts

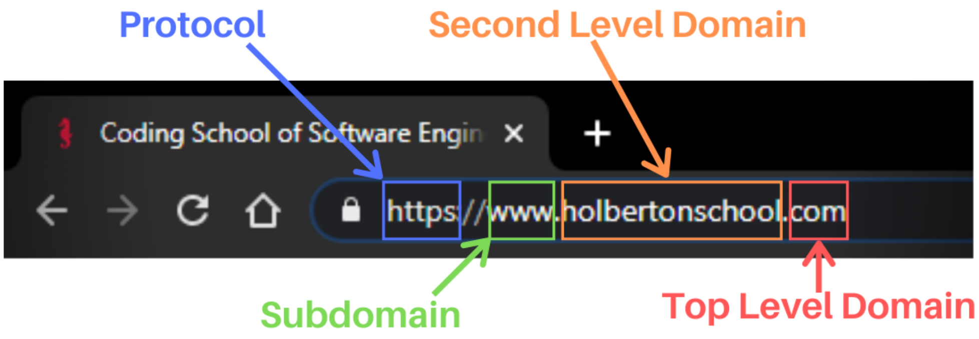
When a user enters an URL in the browser, how does the browser fetch the desired result ?

In order to get into the matter and discover what is happening, it is important to understand the ***client-server model***.

Client-Server-Model - This model is a core framework of network computing that is organized between clients which make data requests, such as a web browser, and servers that store this data and manage most processes to serve it to clients, like a database or applications.

So, now, if we type an url say https://www.holbertonschool.com, in the search bar, what actually happens?

Starting the search


Protocol
Second Level Domain
Coding School of Software Engin
https://www.holbertonschool.com
Subdomain
Top Level Domain

URL stands for Uniform Resource Locator and as its name says, it contains the location of the resources you want to access, in other words, is an address that points to the place you want to visit. Looking at the image above, the first part you see is '*https*' that indicates the browser which protocol (set of rules) should be used to establish the communication on the network. It can be FTP, HTTP, HTTPS, etc.

The second part is the **hostname** reached through Domain Name System (DNS), which is composed of:

- **Subdomain**: also known as third-level domain, is a subdivision of the domain name that is used to organize the web content into main categories, and in a URL indicates a particular page of the website the browser should serve. In the example, ***'www'*** is the subdomain, this is given by default and is often used to land the home page of a website.
- **Second-level Domain**: is the name of the website, it is unique for the company or person that registers it, helping people to know what brand's site they are visiting. In the example, 'holbertonschool' is the name of the website for Holberton School.
- **Top-level domain**: is the highest level in the DNS and it specifies the type of entity under which the website is registered on the internet, some of the most common TLDs are 'org', 'edu', 'com', etc. In the example, 'com' is used to identify the website as a commercial entity.

Locating our destiny website - The DNS query

Well, once you hit Enter, the browser brakes down the URL in pieces as shown above, and looks for the hostname in its cache. If the browser does not find the hostname in there, then it asks outside. And the DNS comes into action, it takes the human-readable hostname and translates it to its corresponding computer-friendly IP Address to serve the requested content.

The IP address is a suite of four numbers each one in a range from 0 to 255, separated by dots. Probably this makes you wonder if it is possible to use an IP address directly instead of a hostname, and the answer is yes you can do it, but the reason why this is not common is that as humans we remember better words than numbers.

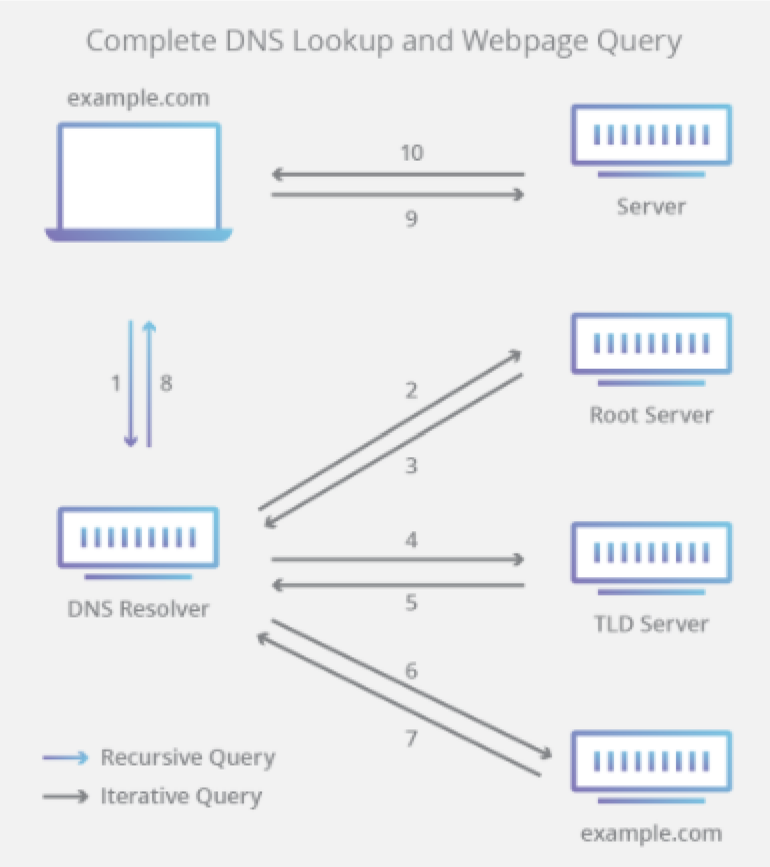
Continuing the DNS process, it is also known as a DNS request, where your computer is the DNS client demanding information to a **DNS server**. You can think of this process as a bus trip and the first stop, that can be the only one or not, is your Internet Service Provider (ISP) DNS server to check if it can respond properly to the request. This DNS server will search into its local cache and if it handles the request, the DNS process ends here returning the IP address for Holberton School's website allowing the browser to serve it up to you. However, if the resolver cannot handle the request, then a recursive query will start until the IP address is found.

Remember the URL components, specifically of the hostname? There is a server that stores information for each one of them.

So, following the bus trip, the second stop is a DNS root nameserver. The root domain does not have a formal name in the DNS hierarchy and its label is an empty string, which is represented by an implicit dot (.) that is at the end of every URL. The resolver requests the DNS root nameserver and it responds with the IP address of a TLD DNS Server, in this case 'com', that stores the information for its domains.

And the bus gets to the third stop, where the resolver requests to the .com TLD and the TLD DNS server responds with the address of the domain’s nameserver, holbertonschool.com. Lastly, the bus leaves to the domain’s nameserver where the ISP DNS server makes a request and the IP address for 'hobertonschool.com' is then returned.

After all of this trip, the bus returns to the starting point, then the resolver responds to the browser with the IP address of the domain requested at the beginning. Besides this, the resolver will store the collected information into its local cache and it will not have to repeat the entire process if the same query comes in the future, it can now provide the Holberton School's IP address directly.

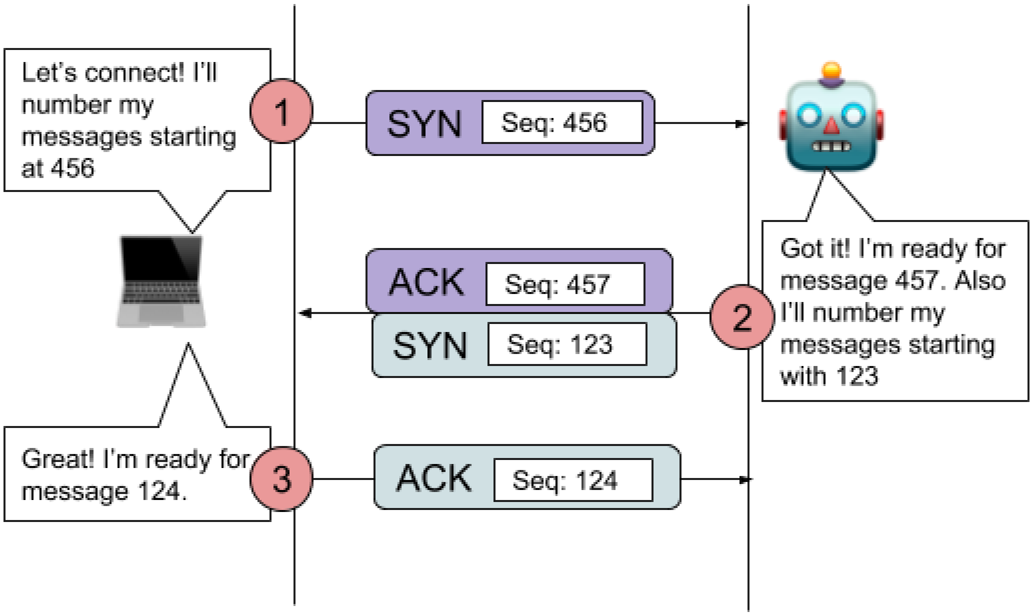


Getting to our destination - The TCP/IP connection

Having the Holberton School's IP address it's time to initiate the connection with it, this is done by following the ***HTTPS*** protocol you indicated in the URL. This protocol is the secure version of the HyperText Transfer Protocol (HTTP) and it is part of a larger protocol suite called Transmission Control Protocol / Internet Protocol(TCP/IP).

In TCP/IP, the connection is built between two hosts (the client and the server) using a process called **TCP 3-way handshake**. The process can be described in brief:

1. A client computer sends a **SYN message** means, whether the second computer is open for a new connection or not.
2. Then another computer, if open for a new connection, sends an **acknowledge message** with SYN message as well.
3. After this, the first computer receives the message and acknowledges it by sending an **ACK message**, that is received by the second computer and the TCP socket connection is established.


Let's connect! I'll number my messages starting at 456
1
SYN
Seq: 456
Got it! I'm ready for message 457. Also I'll number my messages starting with 123
2
ACK
Seq: 457
SYN
Seq: 123
Great! I'm ready for message 124.
3
ACK
Seq: 124

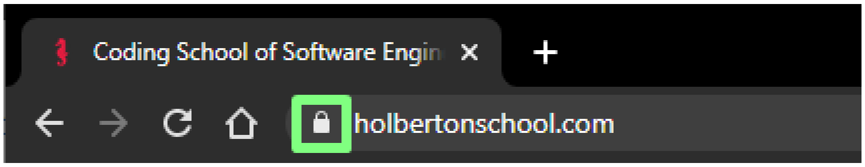
Connection Denied? - The firewall

When you use your network it would be good to consider looking at your firewall as well. The firewall could be part of your computer system or it could be a network. **Firewalls are designed to restrict incoming and outgoing traffic** based on a set of security rules defined by the user. This way, a list of IP addresses is verified before establishing any connection, if the IP address you are trying to reach is on the list, then the connection is blocked and it will not be an exchange of data over the network.

Security and Encryption - HTTPS/SSL

As mentioned before *HTTPS* stands for *HyperText Transfer Protocol* Secure, and is a secure version of the HTTP. This protocol defines different types of requests and responses served between clients and servers over a network. It is the main way to transfer data between a browser and a website. The requests include GET, POST, PUT, among others. The HTTPS requests and responses are secure by a standard security protocol called Secure Sockets Layer (SSL) which encrypts the data being exchanged between your browser and Holberton School's website server using a Public Key Infrastructure (PKI).

In the PKI, two cryptographic keys are used: a public key and a private key. These pair of keys is asymmetrical which ensures only the owners of the matching keys can un-encrypt the sent and/or received information. For being able to get this done, Holberton School's website manager had to issue an SSL certificate from a trusted certificate authority and that is the reason why you can see a lock icon on the search bar of the browser.



Web content is received

Finally, once the TCP connection, HTTP connection, and HTTPS connection are all established, your browser will receive the data from the web server located at the address you entered. If that’s a webpage, that data will be in HTML (hypertext markup language) format, a data format specific to the web.

Then, if you click on any hyperlinks on the page, enter any information to send to the server (like if you fill out a sign-up form or enter payment information), this information is sent and received via HTTP.

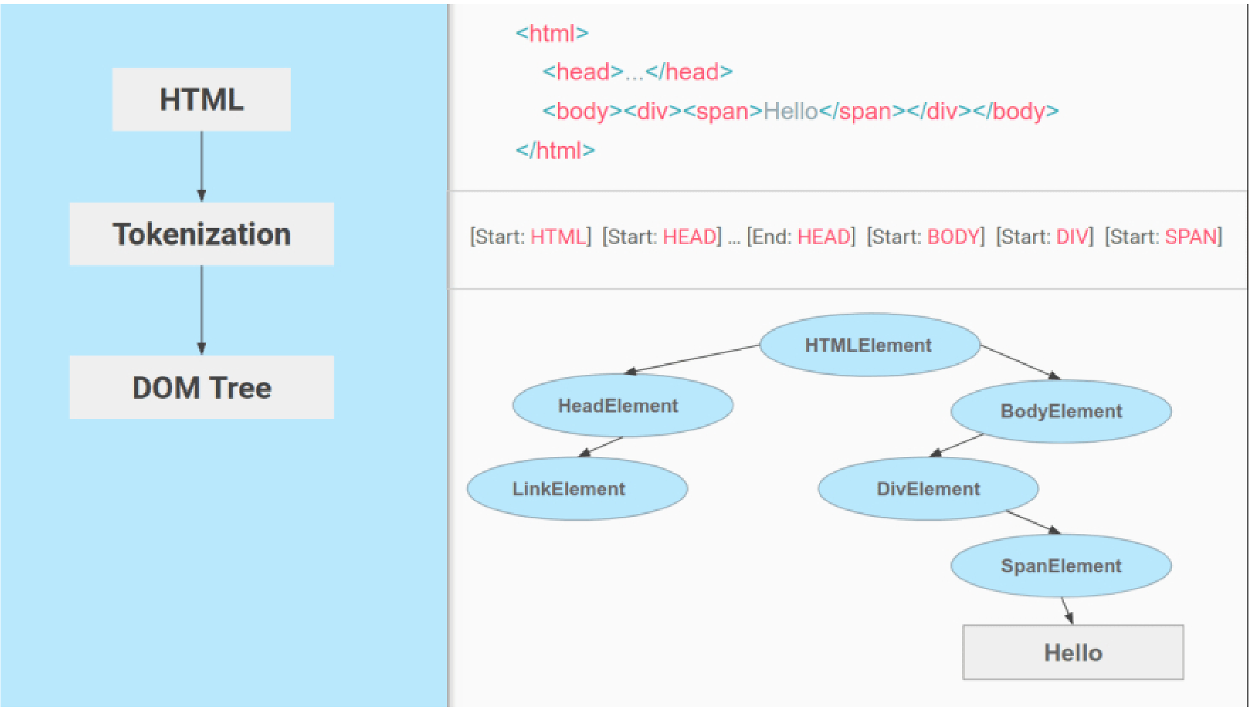
On many websites, HTML documents also refer to specific resources outside of themselves to complete the page that you see when the webpage loads. One good example of this is the stylesheet, or CSS. This is a way to store all the information about colors, fonts, background images, etc. that might apply to all webpages on a particular website separately from the data for the specific webpages. This can help ensure uniformity across an entire website and it keeps HTML documents lean, since that data doesn’t have to be added to every HTML page on a website.

This will be a separate file stored on the same web server, and will require a separate HTTP request from your browser.

HTML Parsing

So we have HTML content at the beginning which goes through a process called tokenization, tokenization is a common process in almost every programming language where code is split into several tokens which are easier to understand while parsing. This is where the HTML's parser understands which is the start and which is the end of the tag, which tag it is and what is inside the tag.

Now we know, html tag starts at the top and then the head tag starts before the html ends so we can figure out that the head is inside html and create a tree out of it. Thus we then get something called a parse tree which eventually becomes a DOM tree as shown in the image below:



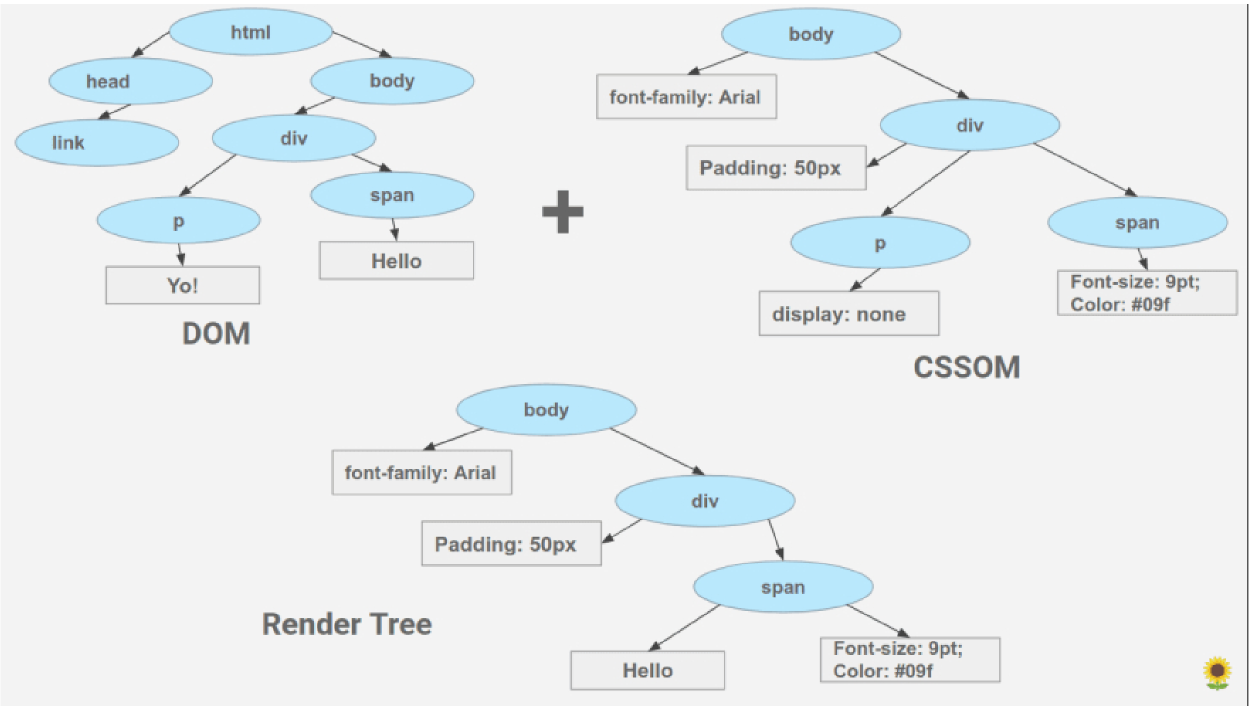
```
<html>
  <head>...</head>
  <body><div><span>Hello</span></div></body>
</html>
```

```
[Start: HTML]  [Start: HEAD] ... [End: HEAD]  [Start: BODY]  [Start: DIV]  [Start: SPAN]
```

DOM tree is what we access when we do document.getElementById or document.querySelector in JavaScript.

Just like HTML, CSS goes through a similar process where we have the CSS text and then the tokenization of CSS to eventually create something called a CSSOM or CSS Object Model.

This is what a CSS Object Model looks like:



Rendering of Web Page

For rendering, a DOM and CSSOM are merged to form something called a Render Tree.
Render Tree has the information required to mark and paint elements on the screen.

Also while forming a Render Tree, elements like \<head>, \<link>, \<script>, and elements with 'display: none' in CSS are ignored since they are not rendered on the screen.

Note that the elements with 'opacity:0' or 'visibility: none' are included in the render tree, even though they are not painted on the screen they do take their positions and render as an empty space and thus are required for calculations.
So now we have a render tree with all the information that is needed to create a visual page. Now, the renderer will use this information to create a Layout and then a Paint

Layout -
The layout is where the elements are marked on the screen. The layout includes all the calculations and mathematics behind an element's position so it takes all the properties related to the position (height, width, position, top left right bottom, etc) from The Render Tree and places the elements on the screen.

Paint -
After Layout, a Paint happens. Paint takes properties like color, background-color, border-color, box-shadow, etc. to paint the screen with colors.

Client-side scripting

Code can also be sent in the HTML file to be executed by your browser.

JavaScript, which enables scripts to be sent to your computer and run in your browser to create effects or enable you to interact with elements on a webpage. Responsive webpages also change according to your browser window size, which in turn usually corresponds to your screen size (desktop, tablet, or phone).

These are applications that run in your browser on your computer and so they're called web applications.

Server-side scripting

The other possibility for greater dynamism in a webpage is server-side scripting. This is where scripting languages like PHP for example, are run on the web server that produce different output depending on input from the web browser (or by extension the user).

In this case, when you enter a URL in your browser, you might be "running" script remotely on a server.

Load-balancer

As mentioned previously, websites are stored on servers. You already know that once the connection is established, both your browser (client) and Holberton School's website (server) are able to send/receive data between each other over HTTPS protocol. Until here, everything works perfectly, but what if multiple requests are coming to the server and it cannot handle this load?

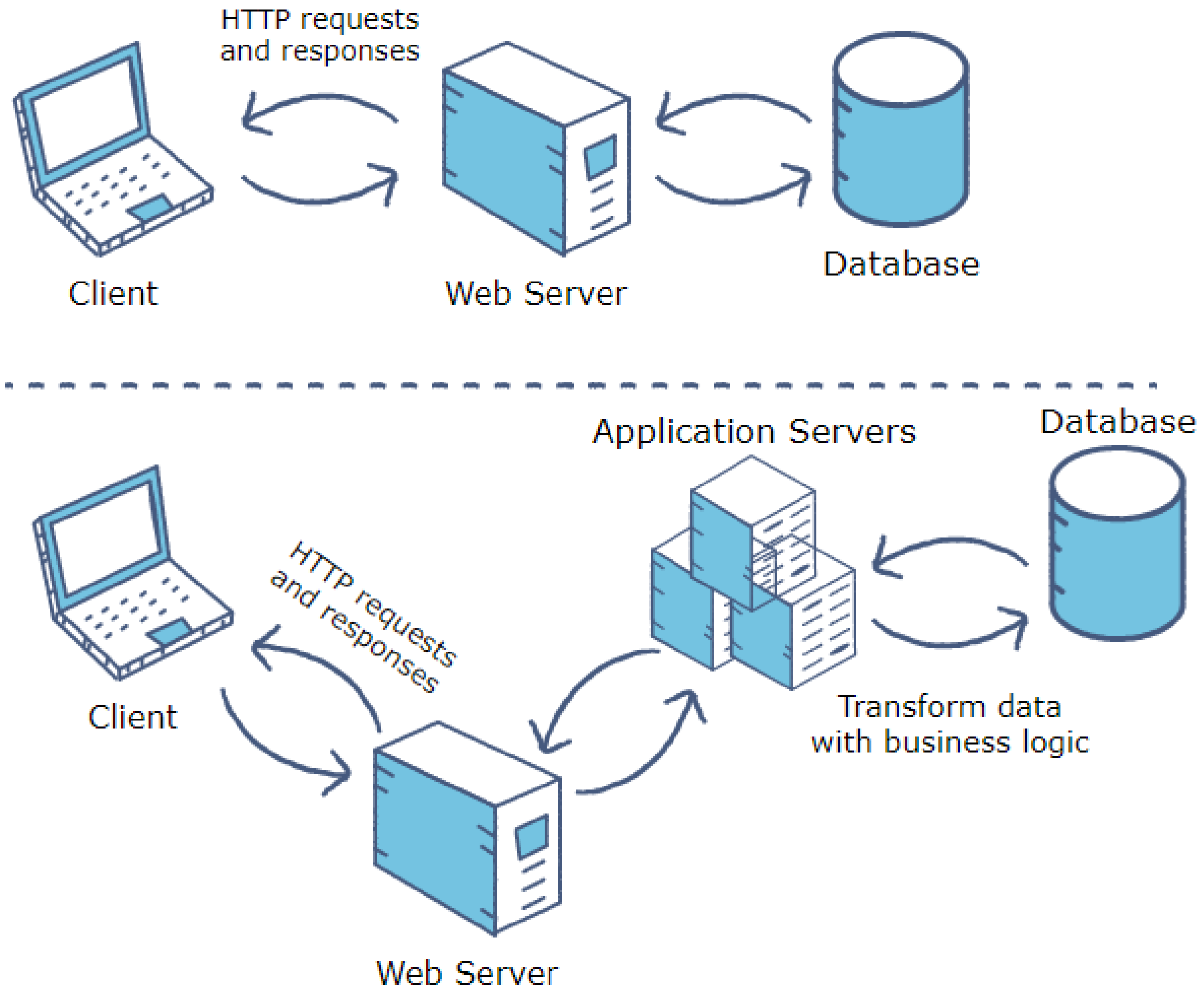
Well, it would be a **Single Point of Failure (SPOF)**, because it would only need a failure or attack on the server to cause the whole site crash. This scenario has led to the implementation of multiple servers increasing the scalability of the websites, therefore, improving its performance. With that, the use of load-balancers is absolutely necessary. A load-balancer is a software program that distributes network requests between the different servers associated with it, based on a load-balancing algorithm like the most common, the round-robin, that distributes the incoming load alternating between all the servers evenly and consequentially.

Where is the website stored? - Web server & Application Server

After being accepted by the firewall the request gets to the load-balancer that distributes the incoming load to the web servers. A web server only handle HTTP/HTTPS requests and serves static content, like simple HTML pages, plain text files or images. Nginx or Apache are examples of web servers.

Holberton School's website, like any other website, could not exist without its web servers. However, having static content only is not the best option to generate interaction with the user. Imagine yourself entering https://www.holbertonschool.com with the purpose of applying to their software engineering program and not being able to sign up or log in with your own username or password to ask and receive information and track each stage of the process, it would not be useful.

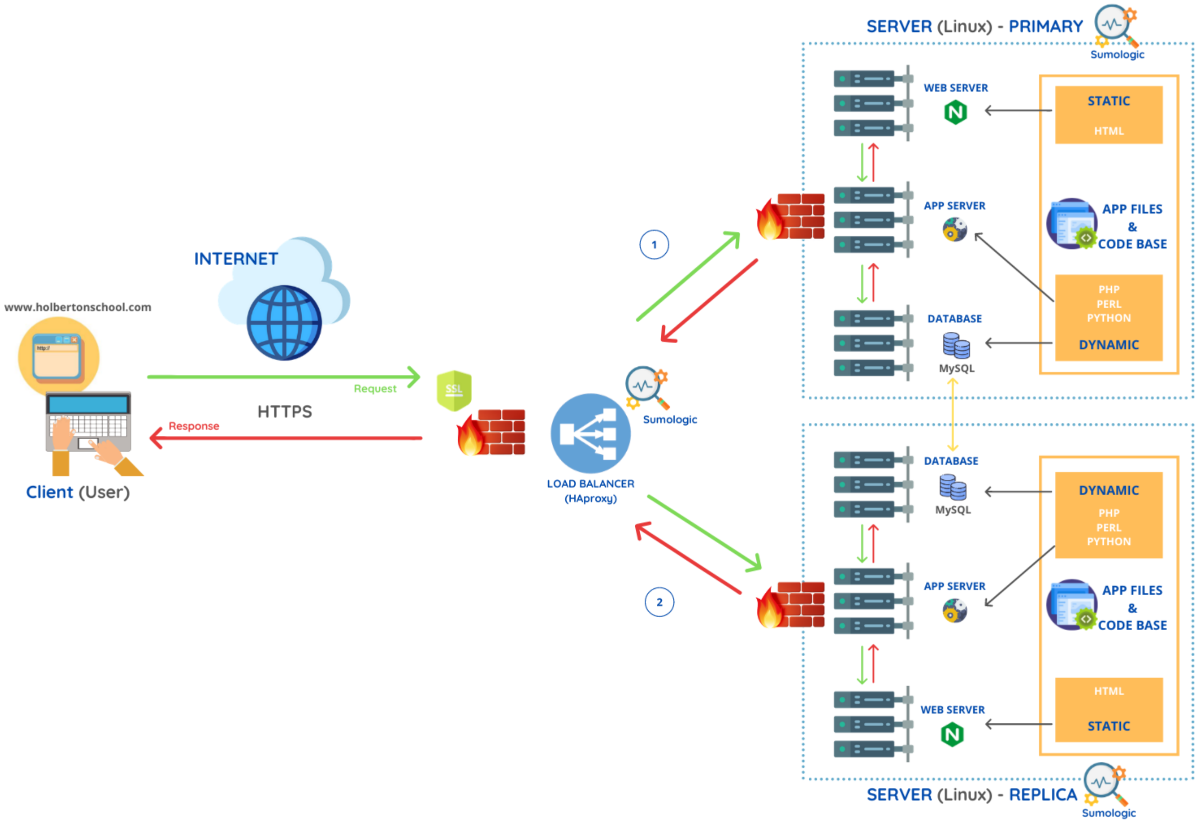
So, this is made possible by having one or more application servers. An **application server** generates dynamic content, by operating applications, communicating with databases, and managing your user information, among other things. All of this, providing their business logic to application programs through several protocols.



How is the data managed? - The Database

The last step in your web journey is the database, used by https://www.holbertonschool.com to add, update, edit, or remove any new information from its website. A **database** is a structured collection of data held inside of a computer or a server and internally organized easy to access the data via its rows, columns, or tables. The actions performed into the database are managed by a Database Management System (DBMS) which is the program that allows the interaction with the database to define, manipulate, retrieve, and manage the data according to the request.

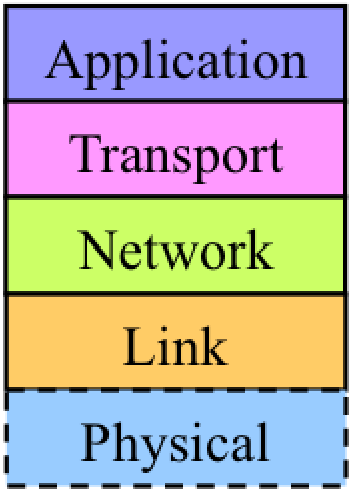
When you are in https://www.holbertonschool.com and you want to enroll, for example, you will be redirected to an application server where you will store your user information in its database.



Internet Protocol Stack

What is a protocol stack?

All the protocols that are part of the Internet standards are organised into a 4/5-layer **protocol stack**. Each *layer* in the protocol stack correspond to a level of abstraction. The Internet protocol stack looks like this:



At the bottom is the **physical layer**, which sometimes are omitted. It represent different types of physical media that can be used for communication: open air, copper cables, optic fibres, water etc.

The next layer is the **link layer**, which represent different types of access network technologies. Examples include Ethernet, different types of WiFi, 4G, Bluetooth etc. These are technologies that can be used to transport messages over the Internet.

The **network layer** is the heart of the Internet protocol stack. It basically hosts two different versions of the IP protocol, and the ICMP control message protocol. The basic service provided at this level is *best-effort global delivery of datagrams*.

As the communication service provided by IP in the network layer is host-to-host only and unreliability, the **transport layer** provide communication services that add message integrity checks, process-to-process communication and the option to get reliable communication. The two basic services in the transport layer are TCP and UDP, but there are also other lesser known protocols at this level.

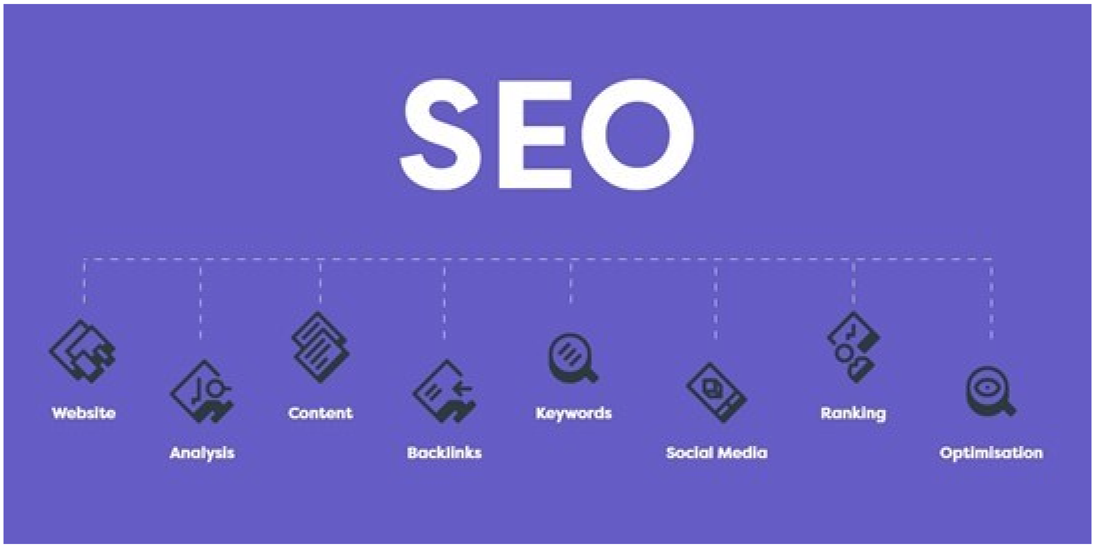
Finally, at the top of the stack are all the applications that exist out there. Applications mostly communicate with other processes at other hosts through one of the communication services provided by the transport layer but there are also a few applications that bypass the transport layer.

SEO

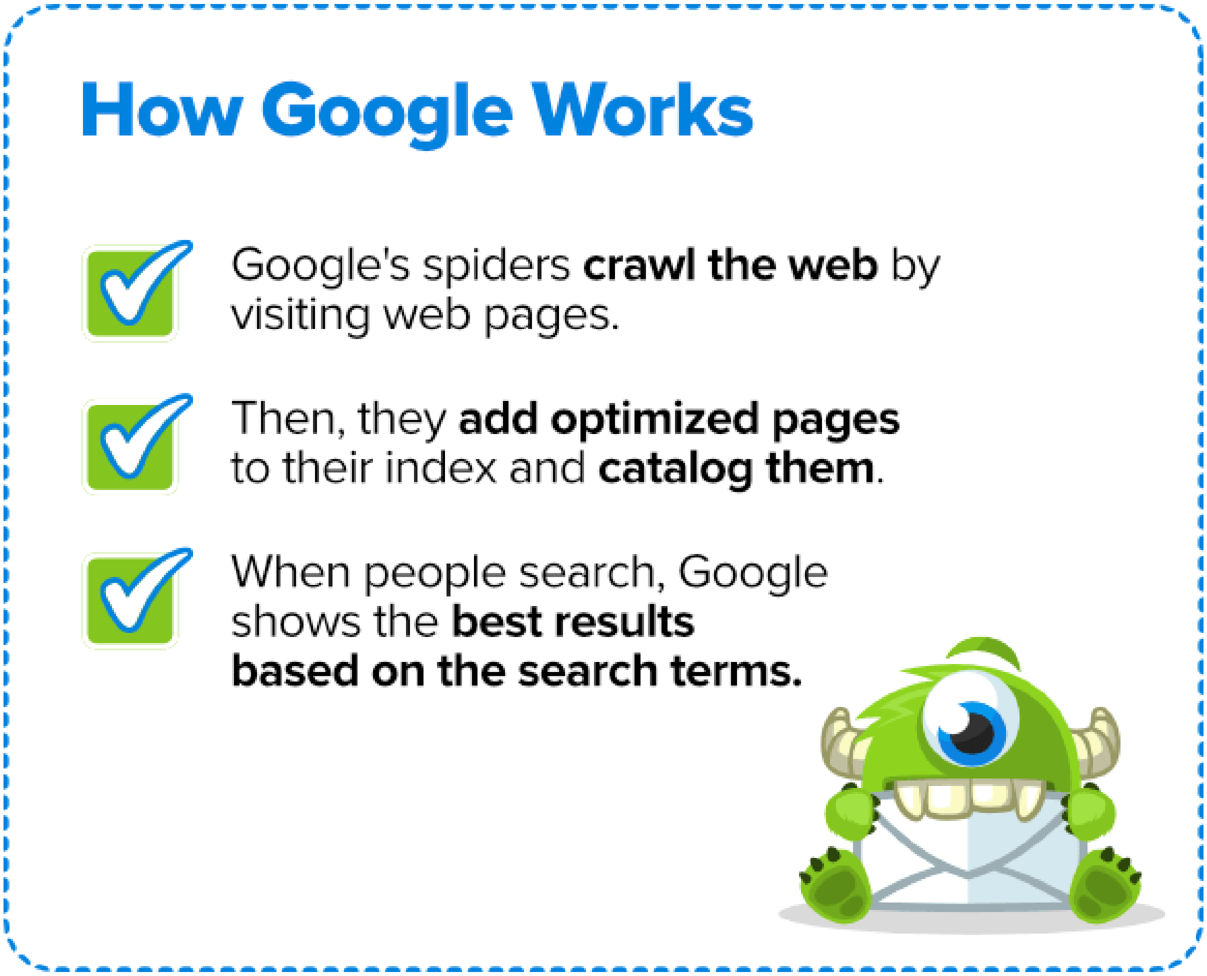
What is ranking in SEO?

In SEO, ranking refers to your content's position on the search engine results pages (SERPs). A #1 ranking means that when people search for a particular term, your web page is the first result

SEO Factors:



What Does Google Look for in SEO?



How Google Works

☑ Google's spiders **crawl the web** by visiting web pages.

☑ Then, they **add optimized pages** to their index and **catalog them**.

☑ When people search, Google shows the **best results based on the search terms.**

First, Google’s search bots (pieces of automated software called “spiders” or “crawlers”) crawl the web. Sounds creepy, we know. But all that means is they visit web pages.

Second, they add correctly optimized and crawlable pages to Google’s index and catalog them.

Third, when people search Google, it shows what it thinks are the most relevant results based on the search terms they enter (out of the trillions of pages in Google’s index).

Google has various deeply complex algorithms to decide which content gets displayed and in which order.

Once your page lands on the SERP, you have to rely on your page titles and meta descriptions to get searchers to click **your** link and visit **your** site.

How Do Google Search Rankings Work?

When people want to find information, they type or say words related to what they're looking for. Those are called keywords, and we'll look at those in the content optimization section of this guide.

But keeping your website ranking on Google isn't just about making the most out of competitive keywords. It's also about the quality of information.

According to Google's own search quality ratings, when it indexes the main content of each page, it checks factors like:

- Purpose of the page
- Content quality and amount
- Website info and info about the content creator
- Website reputation and content creator reputation
- User interaction with the page (time on page, bounce rates, and so on)
- Expertise, authority, and trustworthiness (E-A-T)

What Is E-A-T and Why Does It Matter?

For more information on this:

10 Crucial SEO Ranking Factors You Need to Know

Discover the essential SEO ranking factors to boost your Google rankings. This guide will unlock your website's potential and outrank your competition.

https://optinmonster.com/seo-ranking-factors/