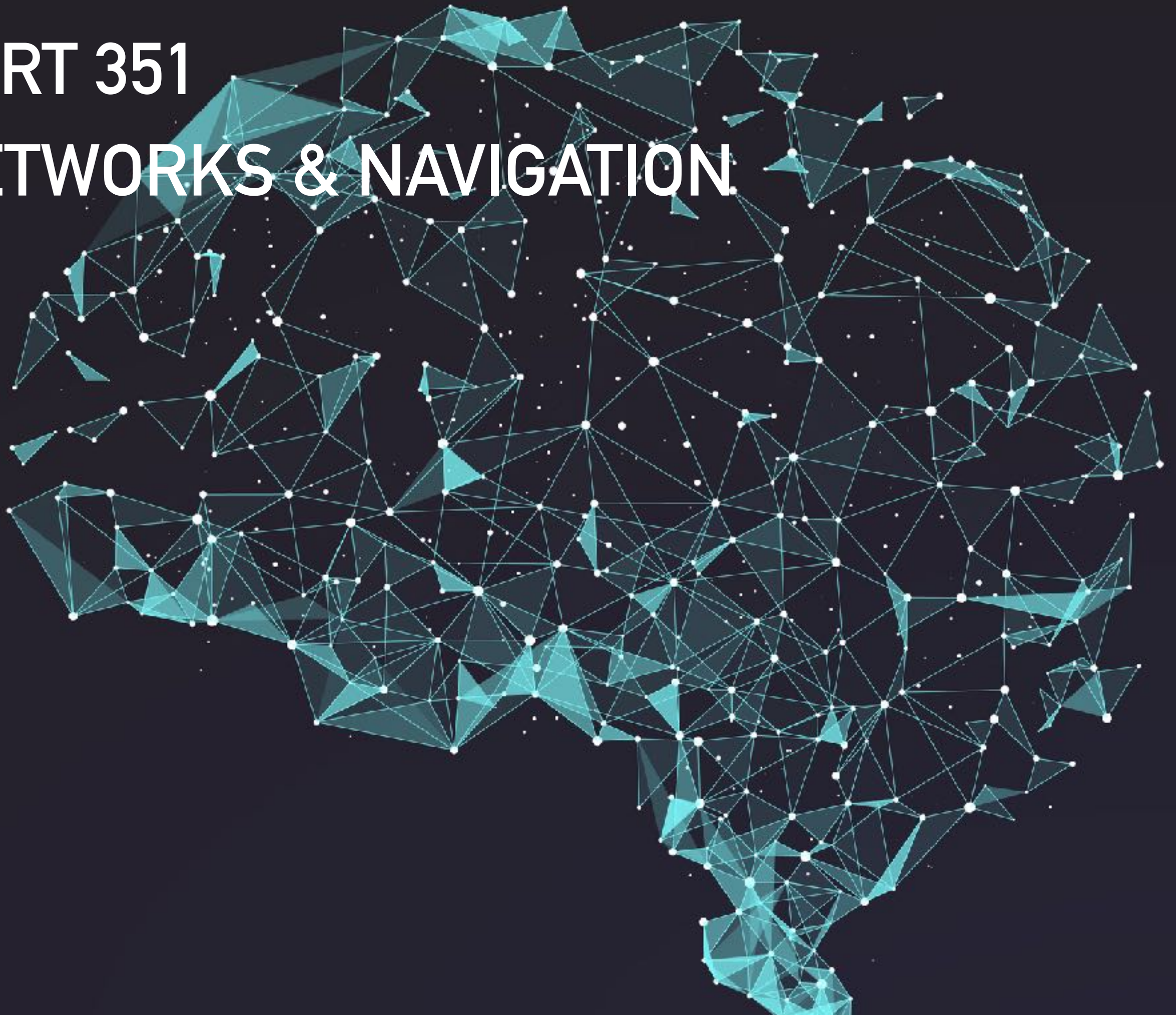


CART 351

NETWORKS & NAVIGATION



AGENDA

INTRODUCTIONS

COURSE OUTLINE & SYLLABUS

HOW THE WEB WORKS

TOOLS

AGENDA

INTRODUCTIONS

COURSE OUTLINE & SYLLABUS

HOW THE WEB WORKS

TOOLS

HOW THE WEB WORKS

TERMINOLOGY

CLIENTS

- ▶ An **application** (i.e. a Browser) such as Chrome or Firefox that runs on a device (computer, phone etc...) & is connected to the Internet
- ▶ Primary function: to **take** user input, **translate** this input into a request, & **send** this request to a remote computer called a **web server**
- ▶ Similarly, the client will **receive** a response from the web server, **translate** the response and **display the result** to the user.
- ▶ A client is not ONLY the application itself, rather it is ALSO the device that this application resides on (computer, tablet, phone...)
- ▶ Every client device **has a unique address called an IP address** that other computers can use to identify it.

SERVERS

- ▶ A machine that is connected to the internet and also has an IP address.
- ▶ A server **waits for requests** from other machines (i.e. a client) & **responds** to these requests.
- ▶ Unlike a client, which also has an IP address, the server has specific server software installed on it, **enabling it to know how to respond** to incoming requests from the client.
- ▶ The primary function of a **web server** is to store, process and deliver web pages to clients.
- ▶ There are many types of servers, web servers, database servers, file servers, application servers, and more...

IP ADDRESS

- ▶ **The Internet Protocol Address:**

- ▶ This is a numerical identifier for a device (computer, server, printer, router, etc.) on a **TCP/IP** network.
- ▶ **Every device** connected to the internet has an IP address that it uses to **identify** and **communicate** with other connected devices.
- ▶ IP addresses have four sets of numbers separated by decimal points (e.g. 245.22.19.26). This is called the **logical address**.
- ▶ In order to locate a device in the network, the logical IP address is converted to a **physical address** by the TCP/IP protocol software.
- ▶ This physical address (i.e. MAC address) is built into your hardware.

ISP

- ▶ **The Internet Service Provider:**

- ▶ This is the middle man between the client and server(s).
- ▶ For the typical homeowner, the ISP is usually a “cable company”
 - Videotron, Bell, etc...
- ▶ When a browser receives a request from a user to go to www.concordia.ca, it doesn't know where or how to look for www.concordia.ca.
- ▶ It is the ISP's task to do a **DNS (Domain Name System) lookup** to find out what the IP address is of the site you're trying to visit.
- ▶ Essentially, the browser will send the request to the ISP...

DNS & DOMAIN NAME

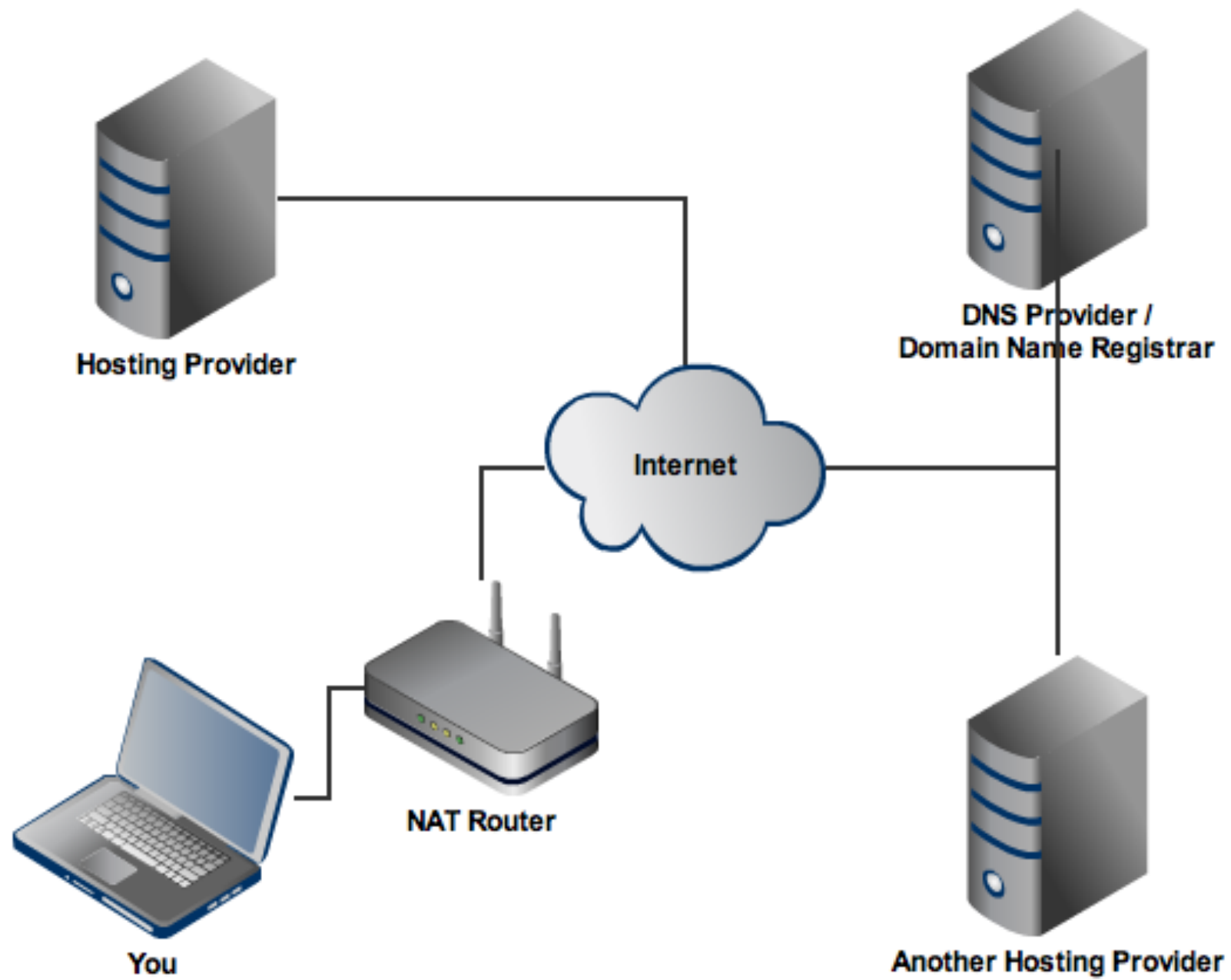
▶ Domain Name System:

- ▶ **A distributed database** which keeps track of all domain names of connected devices and their corresponding IP addresses on the Internet. (don't worry about the details of a distributed database)
- ▶ The DNS exists so users can enter www.concordia.ca instead of an IP address.

▶ Domain Name:

- ▶ Used to identify one or more IP addresses.
- ▶ Users use the domain name (ie. www.concordia.ca) to access a distinct website on the internet.
- ▶ When a user types the domain name into the browser, the DNS uses it to look up the corresponding IP address for that given website.

OVERVIEW



PORT NUMBER

- ▶ A 16-bit integer that **identifies a specific port** on a server and is always associated with an IP address.
- ▶ This integer is used to identify a **specific process** on a server that network requests can be forwarded to.
- ▶ Think of it as a mailbox in an apartment building...

HOST

- ▶ **Very simply:**
 - ▶ **a device connected to a network: a client, server or any other type of device.**
 - ▶ **Each host has a unique IP address.**
- ▶ **For a website like www.google.com, a host could be the web server that serves the pages for the website.**
- ▶ **DO NOT confuse the terms host and server:**
 - ▶ **Servers are a type of host—they are a specific machine.**
 - ▶ **A host can also refer to an entire organization that provides a hosting service to maintain multiple web servers.**

In that sense, you can run a server from a host .

HTTP & URL

- ▶ **Hyper-text Transfer Protocol:**

The **protocol** that web browsers and web servers use to communicate with each other over the Internet. The HTTP protocol runs on-top of the TCP/IP protocol.

- ▶ **Uniform Resource Locators:**

URLs identify a particular web resource.

A simple example is <http://www.concordia.ca/academics/undergraduate.html>

- ▶ **The URL specifies:**

- ▶ **the protocol:** http
- ▶ **Host name:** concordia.ca
- ▶ **A file/path name:** /academics/undergraduate.html.

- ▶ **A user can obtain the web resource identified by this URL via HTTP from a network host whose domain name is concordia.ca.**

MORE ON URLS

- ▶ A URL is also a **hypertext link**: the threads that interconnect the web.
- ▶ Hypertext was first popularized by Ted Nelson, used by Douglas Engelbart, and implemented by Tim Berners-Lee.
- ▶ A link is a portion of text or graphic that transfers the user to the associated web address when clicked on.
- ▶ When the user clicks on a link it is recorded in the browser's history file,
- ▶ When your cursor passes over a link, most browsers display the address of the linked page is shown on the bottom border of the window. .

URL FORMATS

- ▶ The key advantage of a URL is its **universality**: the address is the same no matter where in the world it is used.
- ▶ There are several different standard URL formats which use distinct protocols, listed below are a few:
 - ▶ **Web**: `http://www.concordia.ca/academics/undergraduate.html`
 - ▶ **FTP**: `ftp://neoftp.sci.gsfc.nasa.gov/rgb/`
 - ▶ **Email**: `Mailto:computation.lab@concordia.ca`
 - ▶ **Usenet**: `news://user:password@host:port/newsgroup`
 - ▶ **Telnet**: `telnet://user:password@host:port/`
 - ▶ **SSH**: `ssh://user:password@host:port/`
 - ▶ **File**: `file:///D/Web Site/recipes/chickensoup.html`

TCP/IP

- ▶ The Transmission Control Protocol/Internet Protocol.
 - ▶ The most widely used communications protocol.
 - ▶ A “protocol” is simply a standard set of rules for doing something.
 - ▶ TCP/IP is used as a **standard** for transmitting data over networks.



DATA TRANSMISSION OVERVIEW

- ▶ When a request is made for data, that data is broken up into many tiny chunks called packets.
- ▶ Each packet is tagged with:
 - ▶ a unique header (includes sender/receiver port info)
 - ▶ An IP header which includes the sender/receiver IP addresses.
- ▶ The packet is then transmitted through an ethernet, WiFi or Cellular network and is allowed to travel on any route and take as many hops (has to pass through another device) as it needs to get to the final destination.
- ▶ Once the packets reach the destination, they are reassembled again and delivered as one piece.

TCP/IP DEFINED (NUTSHELL)

- ▶ So how do all the packets know how to get to their destination without getting lost?
- ▶ TCP/IP:
 - ▶ **IP stands for Internet Protocol**: its task is to send and route packets to other connected devices using the IP headers (i.e. the IP addresses) on each packet
 - ▶ **Transmission Control Protocol (TCP)**: is responsible for :
 - ▶ breaking the message or file into smaller packets,
 - ▶ routing packets to the correct application on the destination computer
 - ▶ resending the packets if they get lost on the way,
 - ▶ reassembling the packets in the correct order once they've reached the other end.

**** it is beyond the scope of this class to go into the details of how TCP/IP is implemented.**

HOW THE WEB WORKS

HOW DOES IT WORK?

LET'S TRY ACCESS [HTTPS://WWW.GITHUB.COM](https://www.github.com)

RESOLUTION OF USER REQUEST I

▶ USER ACTION:

Type URL into a WEB BROWSER : i.e. <https://www.github.com>

▶ BROWSER ACTION:

The browser **parses** the information contained in the URL.

- ▶ the protocol (“https”),
- ▶ the host(“github.com”)
- ▶ and the resource (“/”).

In this example, there is nothing after the “.com” to indicate a specific resource, so the browser knows to retrieve just the main (index) page.

RESOLUTION OF USER REQUEST II

▶ BROWSER ACTION -> ISP:

The browser communicates with the ISP to do a DNS lookup of the IP address for the web server that hosts www.github.com

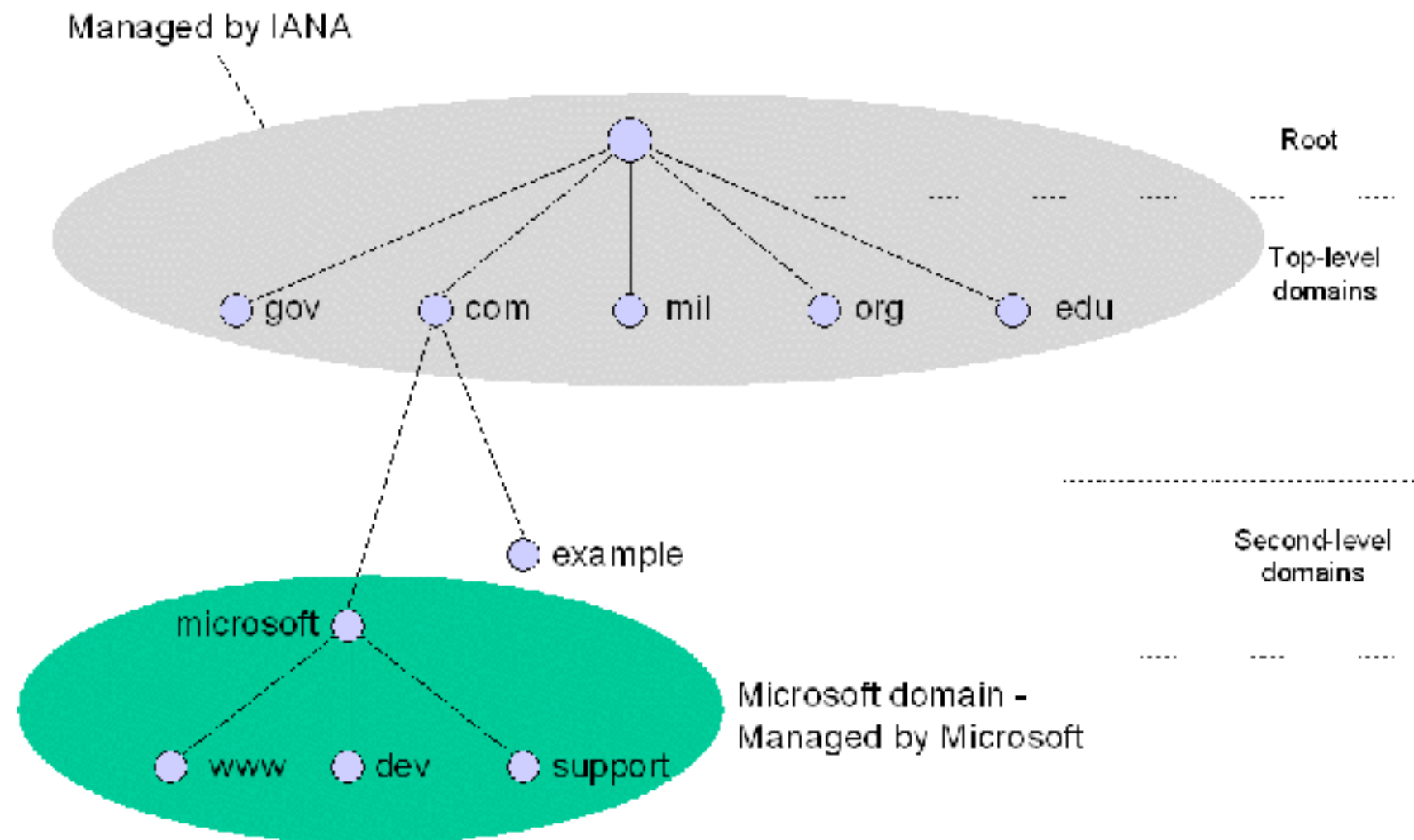
▶ ISP -> DNS ACTION I -> ROOT NAME SERVER:

- ▶ The DNS service will first contact the **Root Name Server** to find the correct server which resolves **.com** addresses.
- ▶ The **Root Name Server** looks at <https://www.github.com> & responds with the IP address of a name server for the **.com** top-level domain.
- ▶ This address is **sent back** to your DNS service.

▶ DNS ACTION II -> NAME SERVER:

The DNS service makes a request to the **.com** name server and asks it for the address of <https://www.github.com>. Once resolved, the ISP can send back the address.

NAME SERVER HIERARCHY (SIMPLIFIED)



source: <https://technet.microsoft.com/en-us/library/bb962069.aspx>

RESOLUTION OF USER REQUEST III

► ISP ACTION -> BROWSER:

Once the ISP receives the IP address of the destination server, it sends it to your web browser:

- The protocol: https://
- IP address 192.30.253.112
- Port number:443

Note: there are tools available to find out info about domain names <http://whois.domaintools.com/github.com> -> I get the ip address of 192.30.253.112

- The browser takes the IP address and the given port number from the URL (the HTTP protocol defaults to port 80 and HTTPS defaults to port 443) and opens a TCP socket connection. At this point, your web browser and web server are finally connected.

RESOLUTION OF USER REQUEST IV

► CLIENT ACTION -> WEB SERVER:

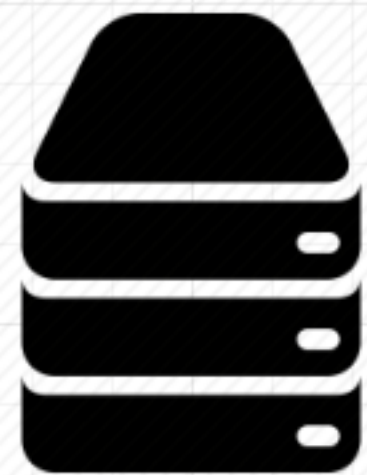
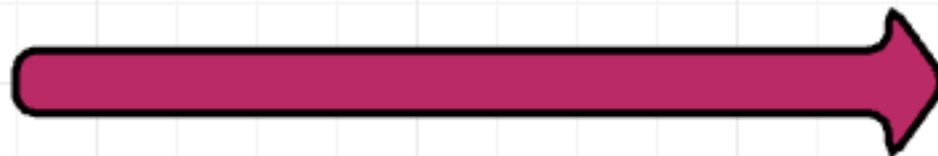
The web browser sends an HTTP request to the web server for the main HTML web page of www.github.com



Client

GET /index.html HTTP/1.1

Host: www.github.com



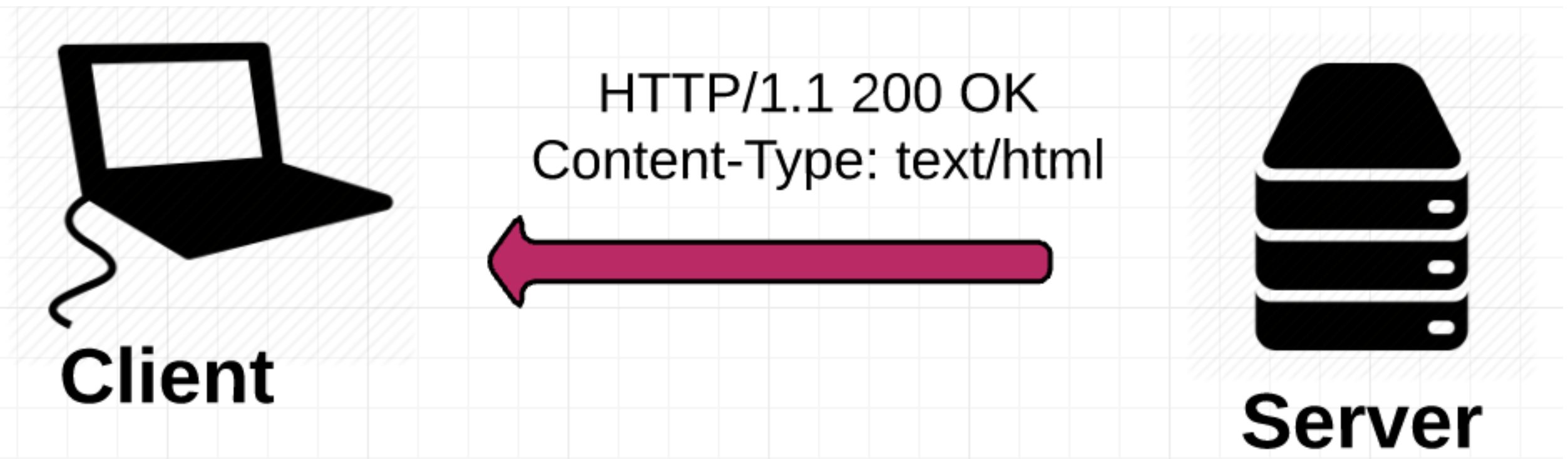
Server

RESOLUTION OF USER REQUEST V

▶ WEB SERVER ACTION -> CLIENT:

The web server receives the request and looks for that HTML page.

- ▶ If the page exists, the web server prepares the response and sends it back to your browser.
- ▶ If the server cannot find the requested page, it will send an HTTP 404 error message, which stands for “Page Not Found”.



RESOLUTION OF USER REQUEST VI

► BROWSER ACTION:

The web browser takes the HTML page it receives **and then parses through** it doing a full scan looking for other assets that are listed, such as images, CSS files, JavaScript files, etc...

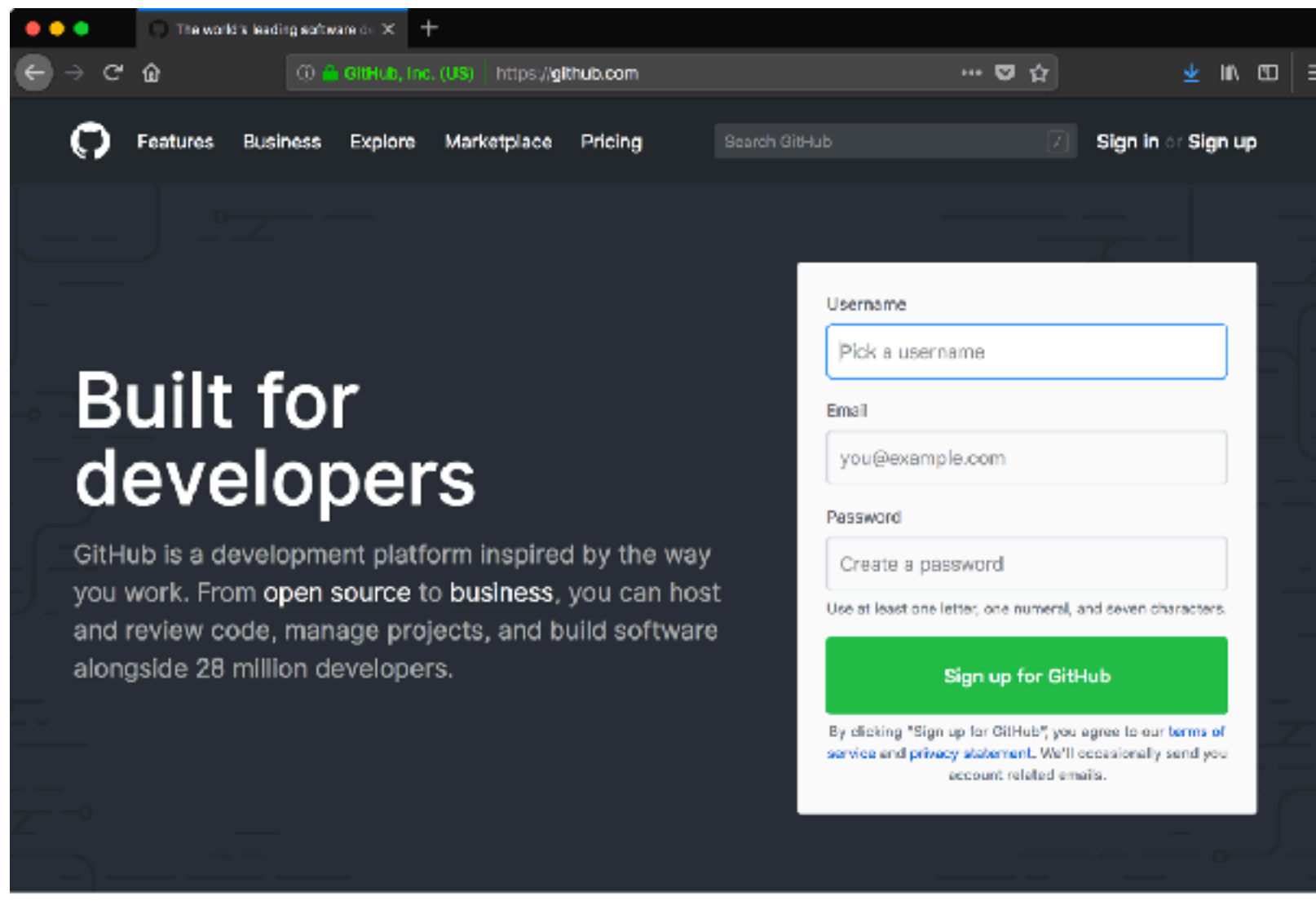
For each asset listed, the browser **repeats the entire process above, making additional HTTP requests to the server for each resource.**

```
<!DOCTYPE html>
<html>
...▼ <head>
  <title>Example</title>
  <link rel="stylesheet" href="/stylesheets/style.css">
  <link rel="stylesheet" href="/stylesheets/bootstrap.min.css">
</head>
▼ <body>
  <h1>Example</h1>
  <p>Welcome to Example</p>
  <p id="register_instructions">Please enter your phone number:</p>
  ► <div id="register">...</div>
  <script src="/javascripts/jquery-1.11.3.min.js"></script>
  <script src="/javascripts/bootstrap.min.js"></script>
</body>
</html>
```


RESOLUTION OF USER REQUEST VII

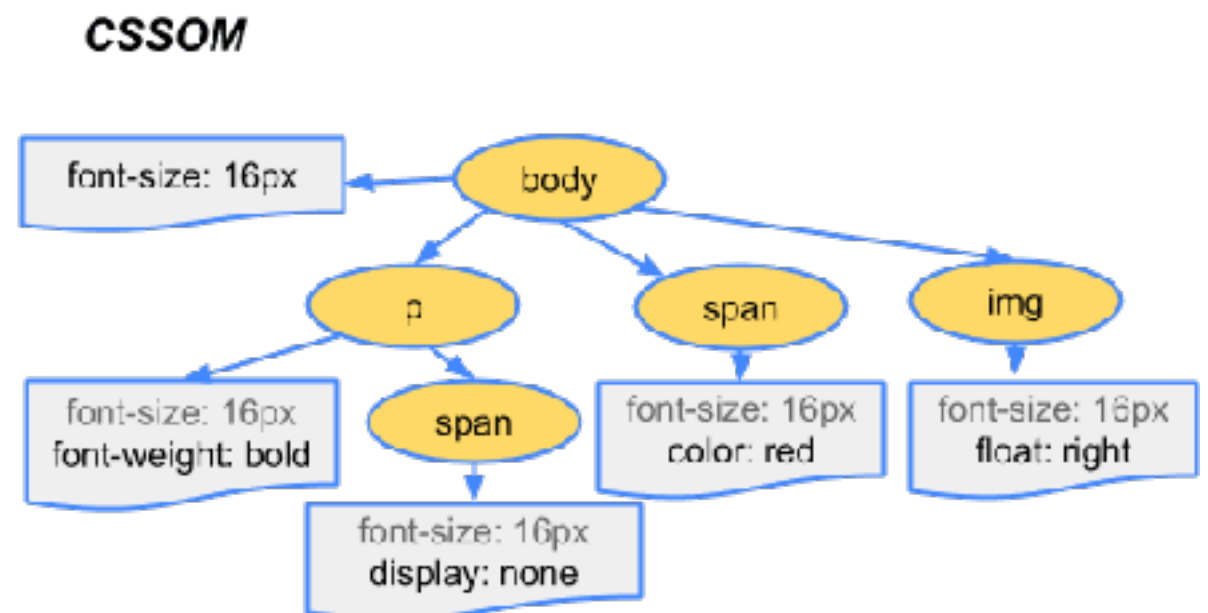
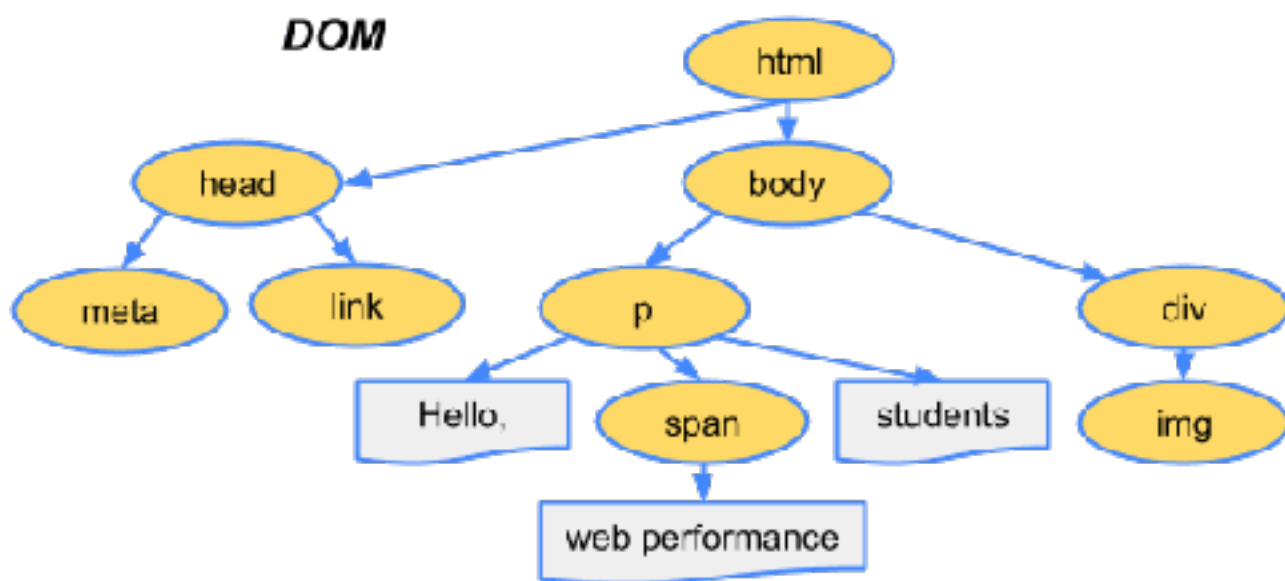
► BROWSER ACTION:

Once the browser has finished loading all other assets that were listed in the HTML page, the page will finally be loaded in the browser window and the connection will be closed.



THE BROWSER RENDERING ENGINE I

- ▶ Every browser **has a rendering engine** responsible for displaying the content. The rendering engine **receives the content of the resources** (images, text, css files...) in small chunks
 - ▶ An HTML parsing algorithm tells the browser **how to parse** the resources.
 - ▶ Once parsed, a **DOM (Document Object Model)** tree is generated.
 - ▶ While the DOM tree is being built, the style sheets are also parsed to understand how to style each node and a **CSSOM (CSS Object Model)** tree is generated.



AGENDA

INTRODUCTIONS

COURSE OUTLINE & SYLLABUS

HOW THE WEB WORKS

TOOLS

TOOLS TO BE USED IN CART 351

- ▶ GITHUB (FOR VERSION CONTROL)
- ▶ ATOM OPEN SOURCE TEXT EDITOR + ATOM LIVE SERVER + OTHER PACKAGES
== IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) FOR DEVELOPMENT
- ▶ WEB BROWSERS I.E. CHROME AS CLIENTS & DEBUGGING
- ▶ HYBRID.CONCORDIA.CA FOR HOSTING & SERVER PROGRAMMING
- ▶ HTML, CSS, JAVASCRIPT + LIBRARIES, PHP & SQL LITE FOR MARKUP & CODING & STORING
- ▶ API'S FOR *ACCESSING SPECIAL SERVICES*