Web Application Development

INFWAD01-D

INFWAD21-D

HOGESCHOOL ROTTERDAM

CMI - Informatica

Web Application Development

Course content	Learning outcomes
Introduction to Web Technologies (HTML, CSS, Client-Side interaction)	Understand the Web and get a grasp of basic Web Technologies
Frontend with Typescript & React	Master React with TypeScript and create a stateful User Interface
Backend with C# & Entity Framework	Design and implement a RESTful API using .NET Core
Integrating frontend and backend to get a full-stack web application	Building full-stack applications

Course structure

- Grading:
  - Theory exam consisting of a multiple choice exam (50%)
  - Practical project (50%)
- More in the course manual
- Team code: Use **jlj8p76** to join the INFWAD team if you’re added already

1.1 Overview of Web & Web Technologies

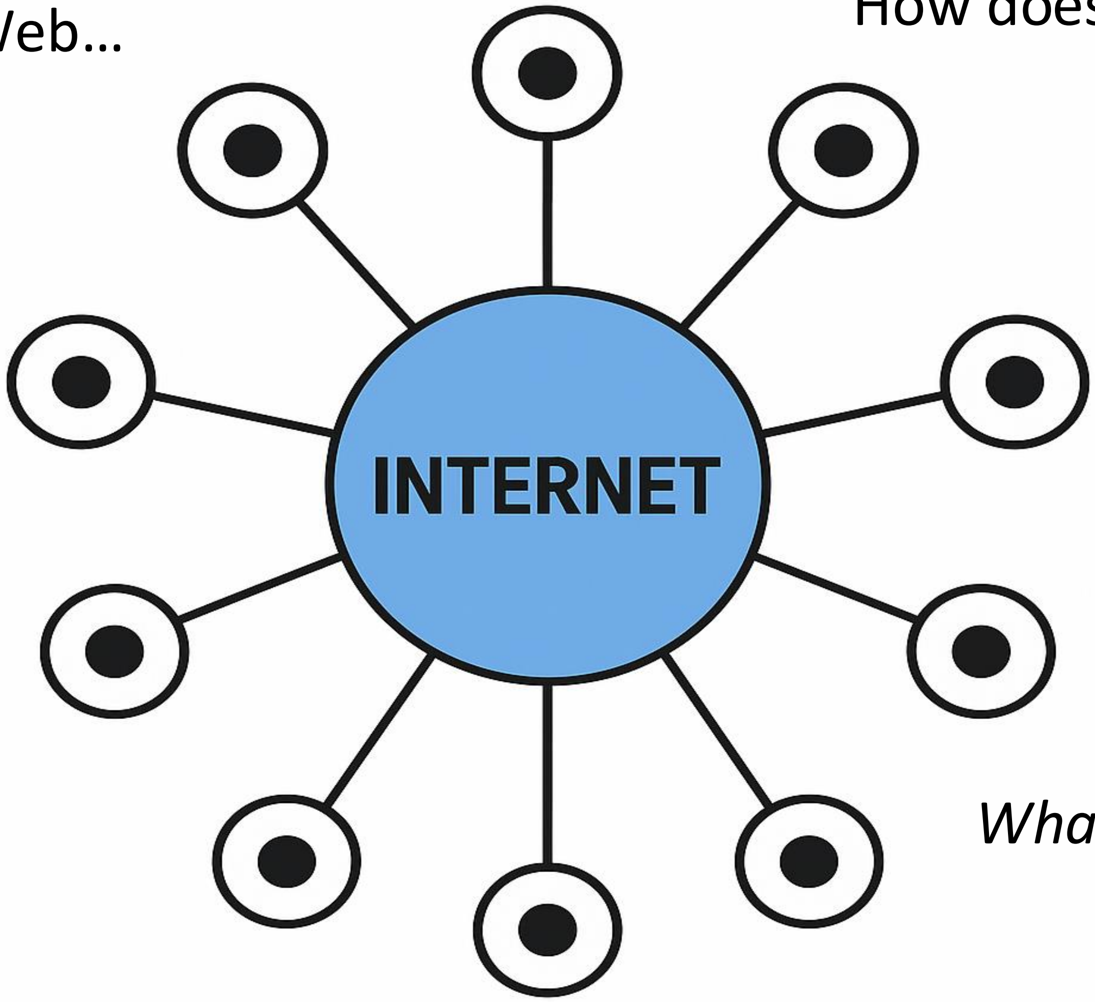
- Introduction to the Web
- How does the Web work
- What is the HTTP and how it works
- Web Development

The Web in our lives

How does the internet work

Before we go into the Web…

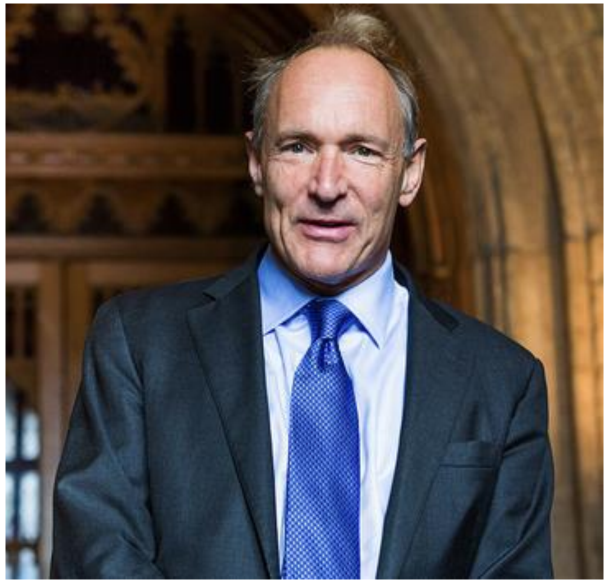
How does the internet roughly work?



What happened in the early 90s?

What is the World Wide Web (WWW)

System of interlinked hypertext documents and resources accessible over the internet with certain rules/protocols

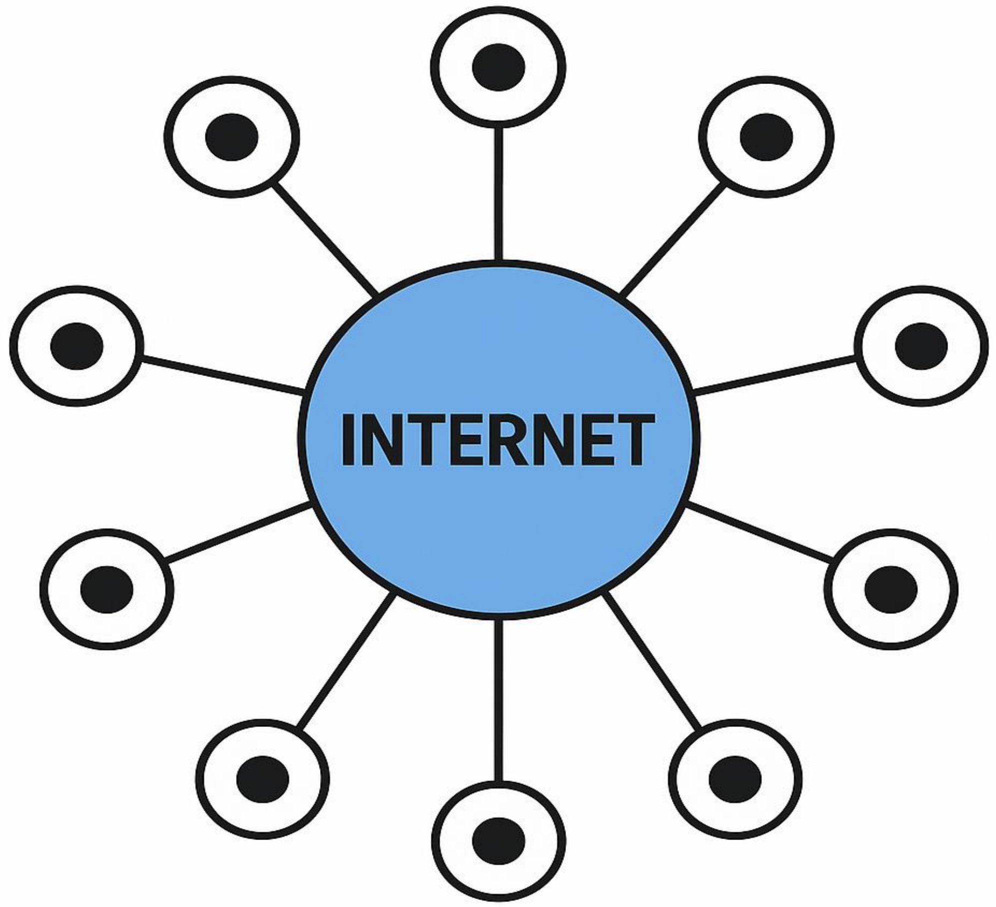

Tim Berners Lee [1]

This machine is a server, DO NOT POWER IT DOWN!!

[1] https://nl.wikipedia.org/wiki/Tim_Berners-Lee

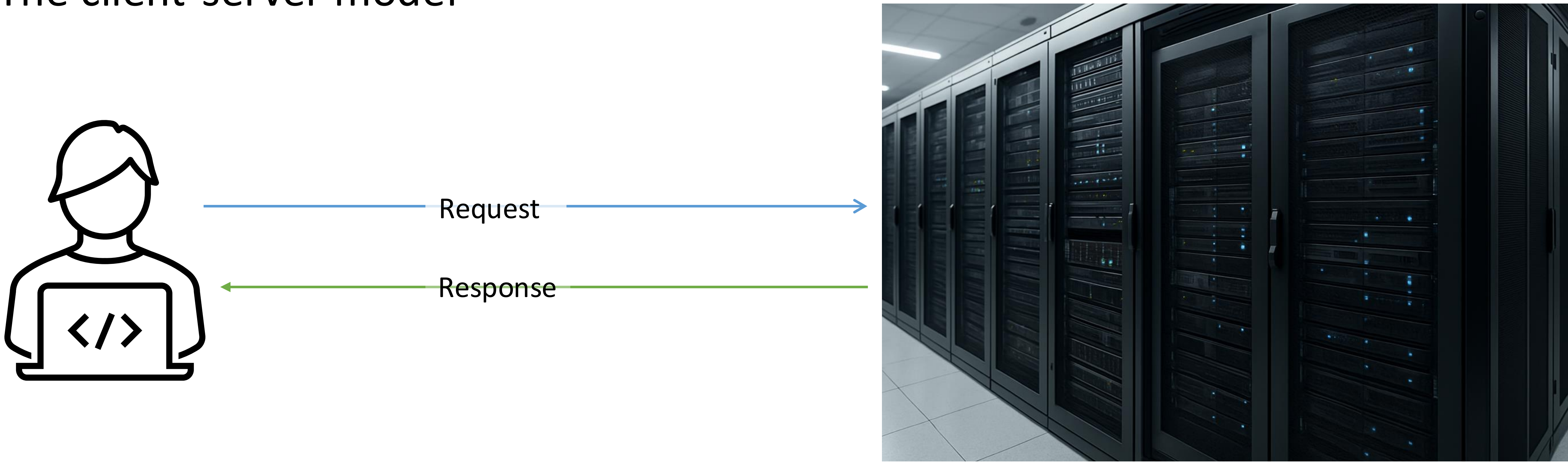
How does the Web work

How does the Web roughly work?



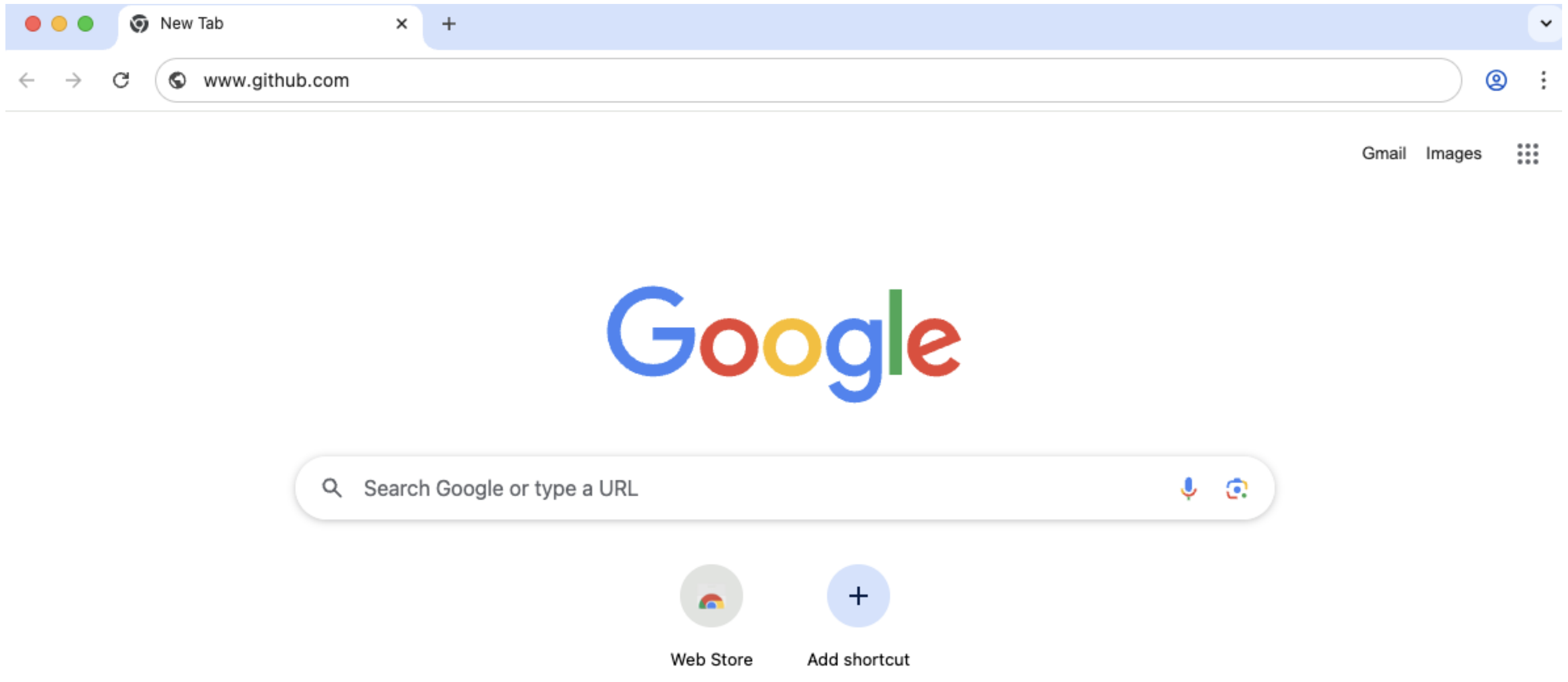
How does the Web work

The client-server model



Browser

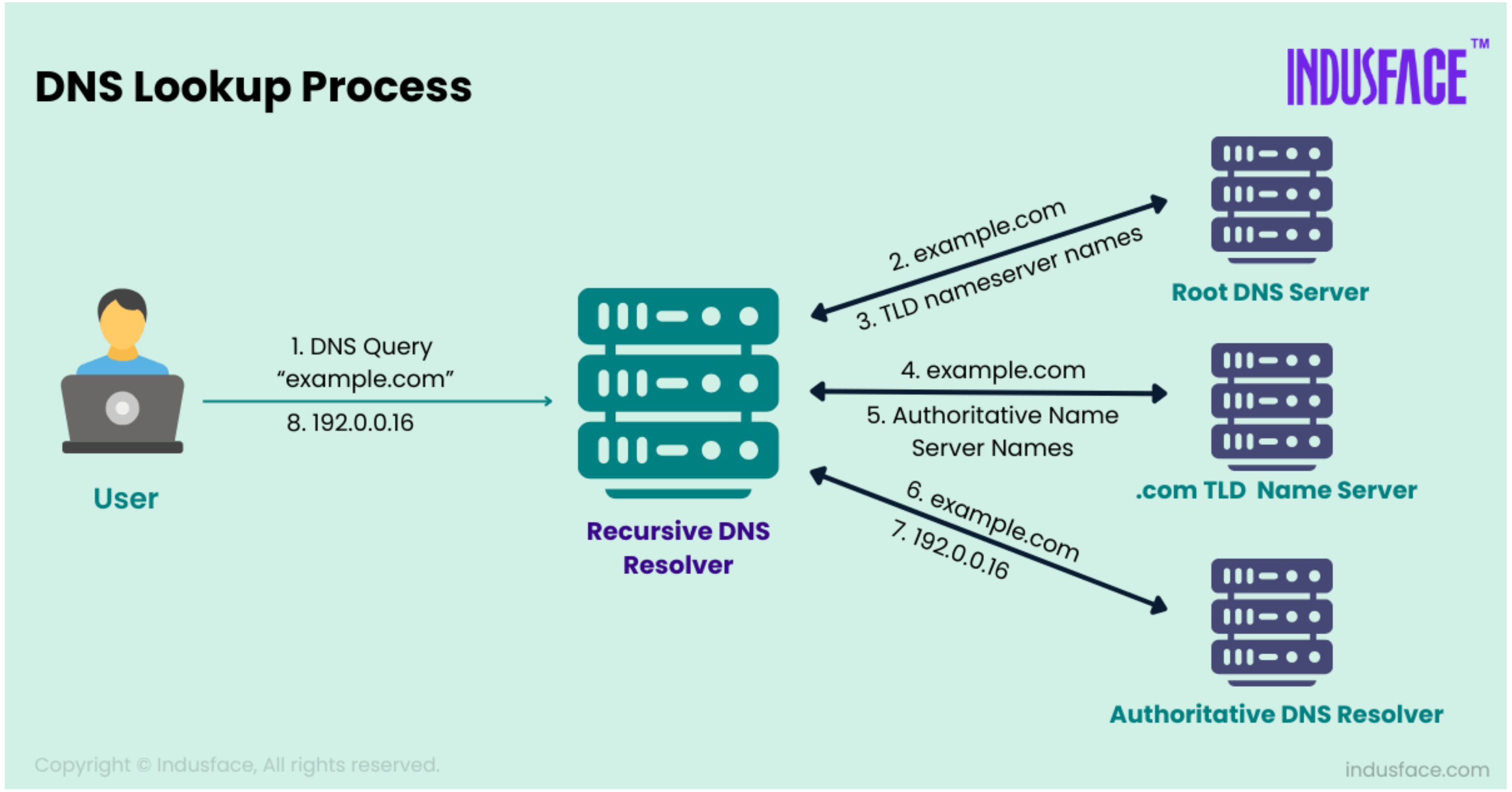
What happens when we search something in our browsers?



DNS Lookup!

Domain Name Server (DNS)

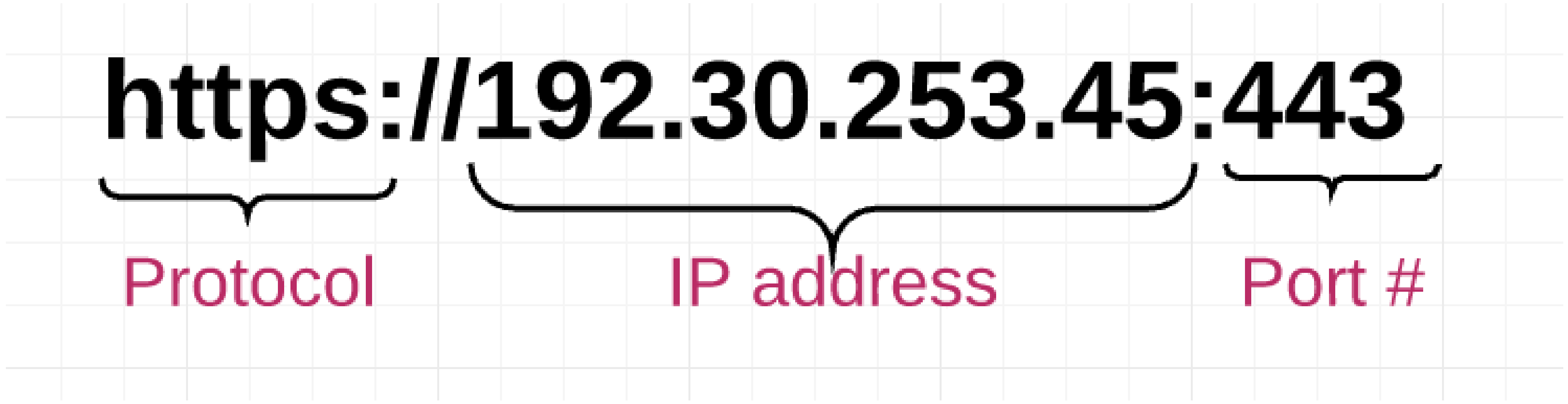
- Address book for websites
- Provides the unique location (IP) of the server



Try it out via the [DNS Lookup]()

HTTP Request

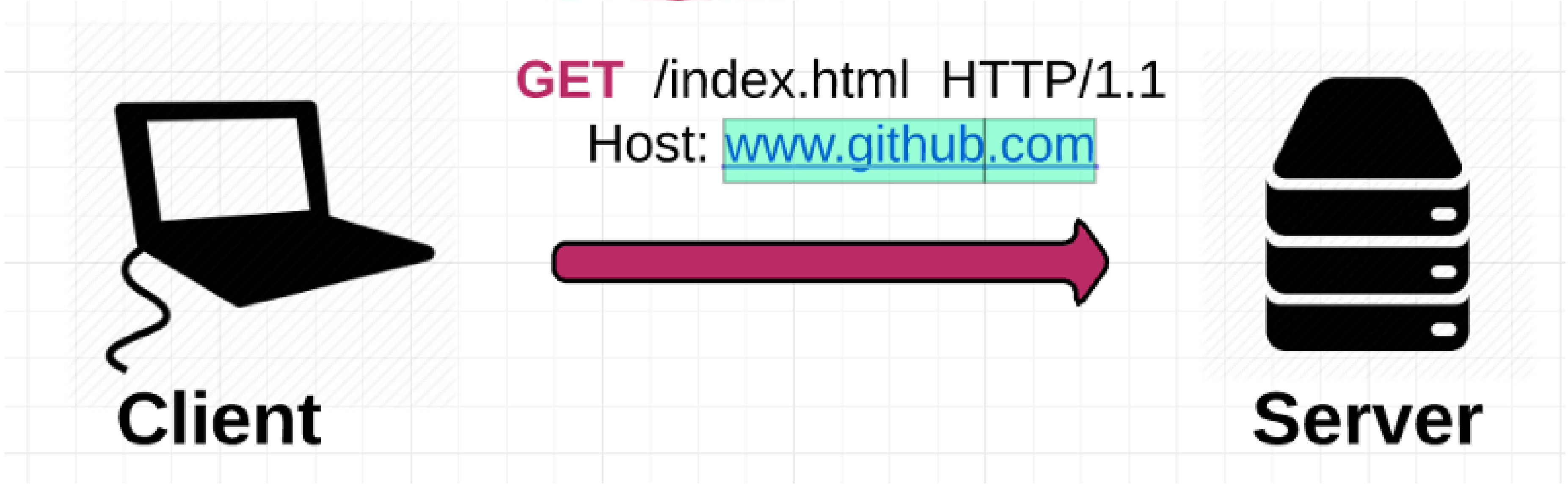
2. IP Address retrieved from DNS

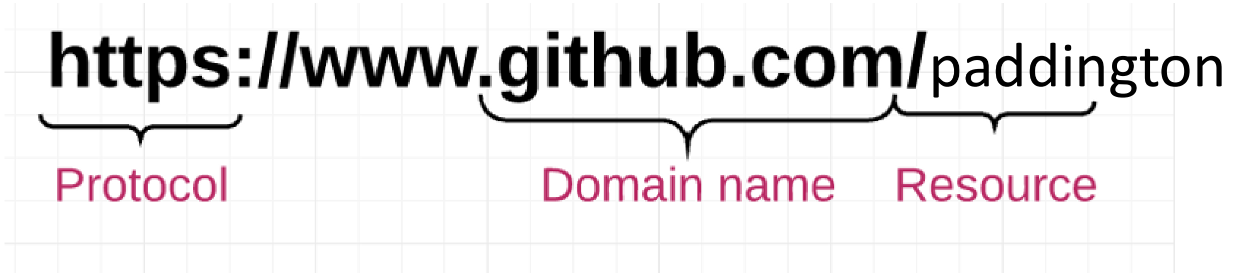


Are we ready to send an HTTP request?

First, a TCP connection is established between the client and server – will explain more later

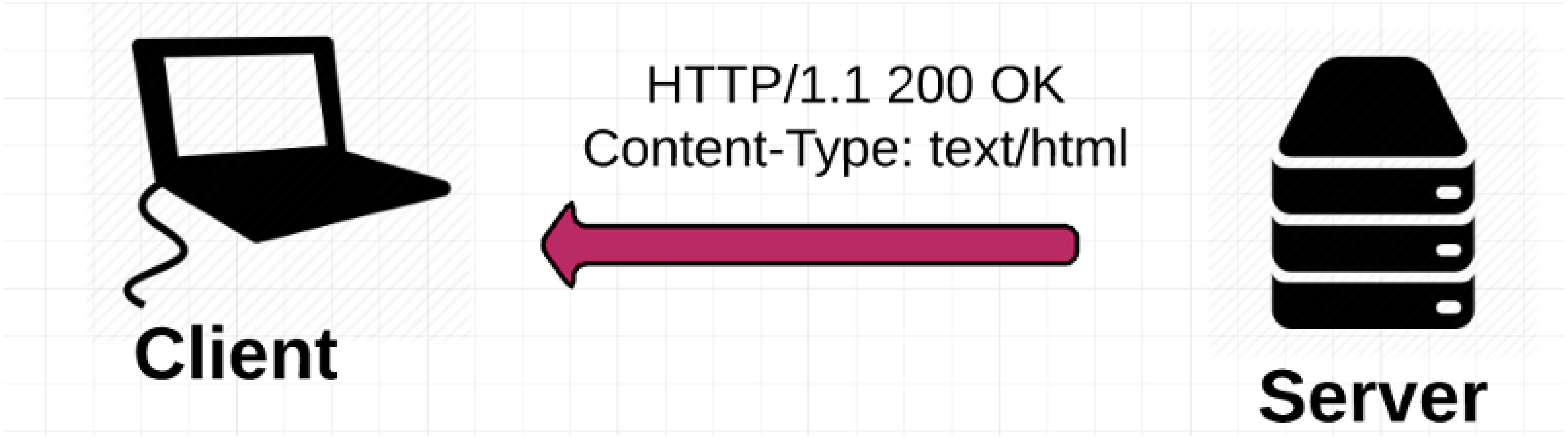
HTTP Request





HTTP Response

4. Server recieves request, starts looking for the HTML page.



HTTP Response

- Example HTTP Response:

HTTP

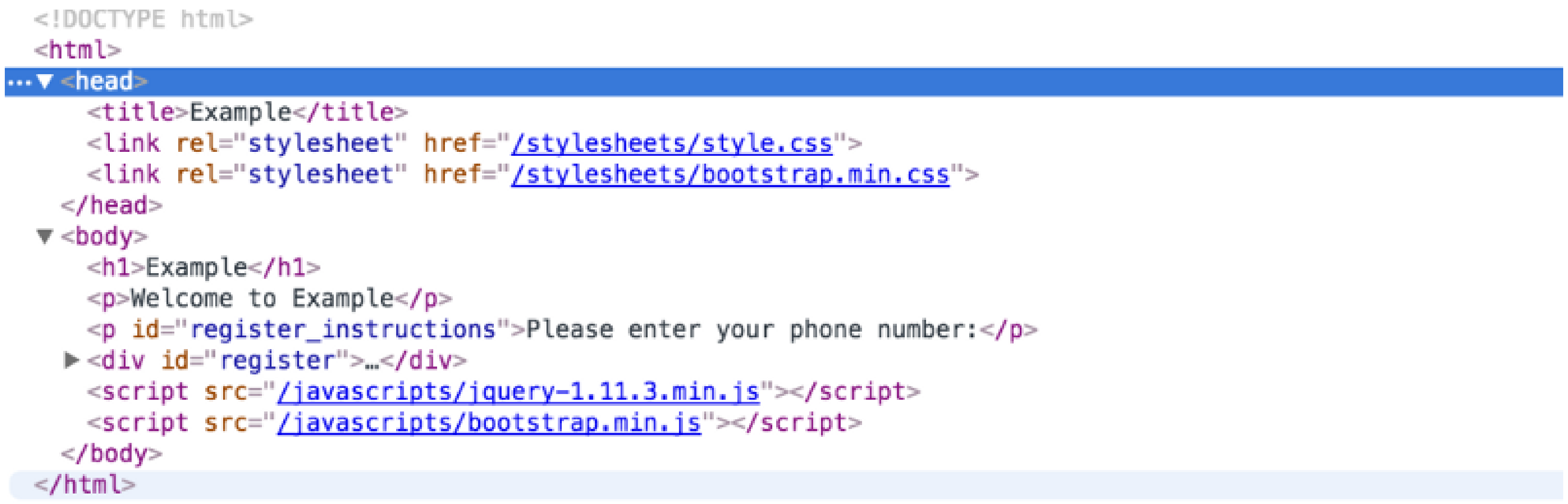
```
HTTP/2 200

date: Tue, 11 Feb 2025 11:13:30 GMT
expires: Tue, 11 Feb 2025 11:40:01 GMT
server: Google frontend
last-modified: Tue, 11 Feb 2025 00:49:32 GMT
etag: "65f26b7f6463e2347f4e5a7a2adcee54"
content-length: 45227
content-type: text/html

<!doctype html> ... (the 45227 bytes of the requested web page HTML)
```

HTTP Response body

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



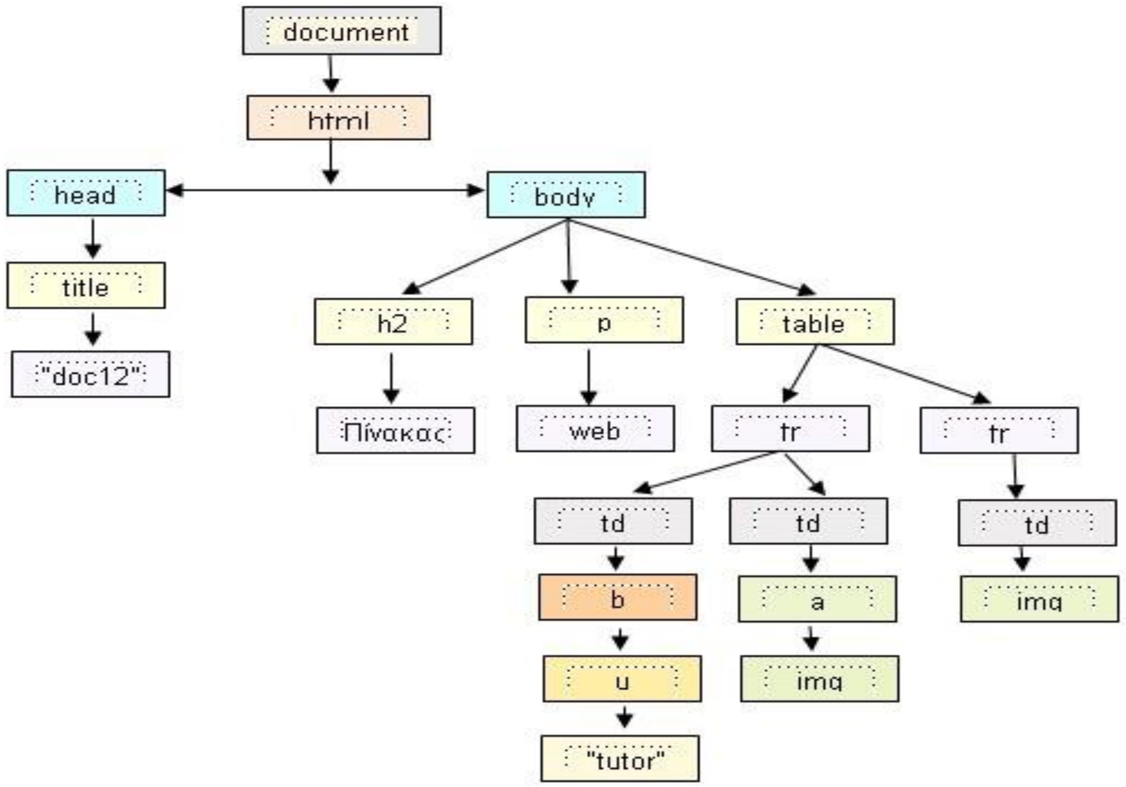
```
<!DOCTYPE html>
<html>
···▼ <head>
    <title>Example</title>
    <link rel="stylesheet" href="/stylesheets/style.css">
    <link rel="stylesheet" href="/stylesheets/bootstrap.min.css">
  </head>
 ▼ <body>
    <h1>Example</h1>
    <p>Welcome to Example</p>
    <p id="register_instructions">Please 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>
```

How does the browser process the HTML response body?

Feel free to check the inspector and network tabs of your browser console for assets of other websites

HTTP Response body (DOM)

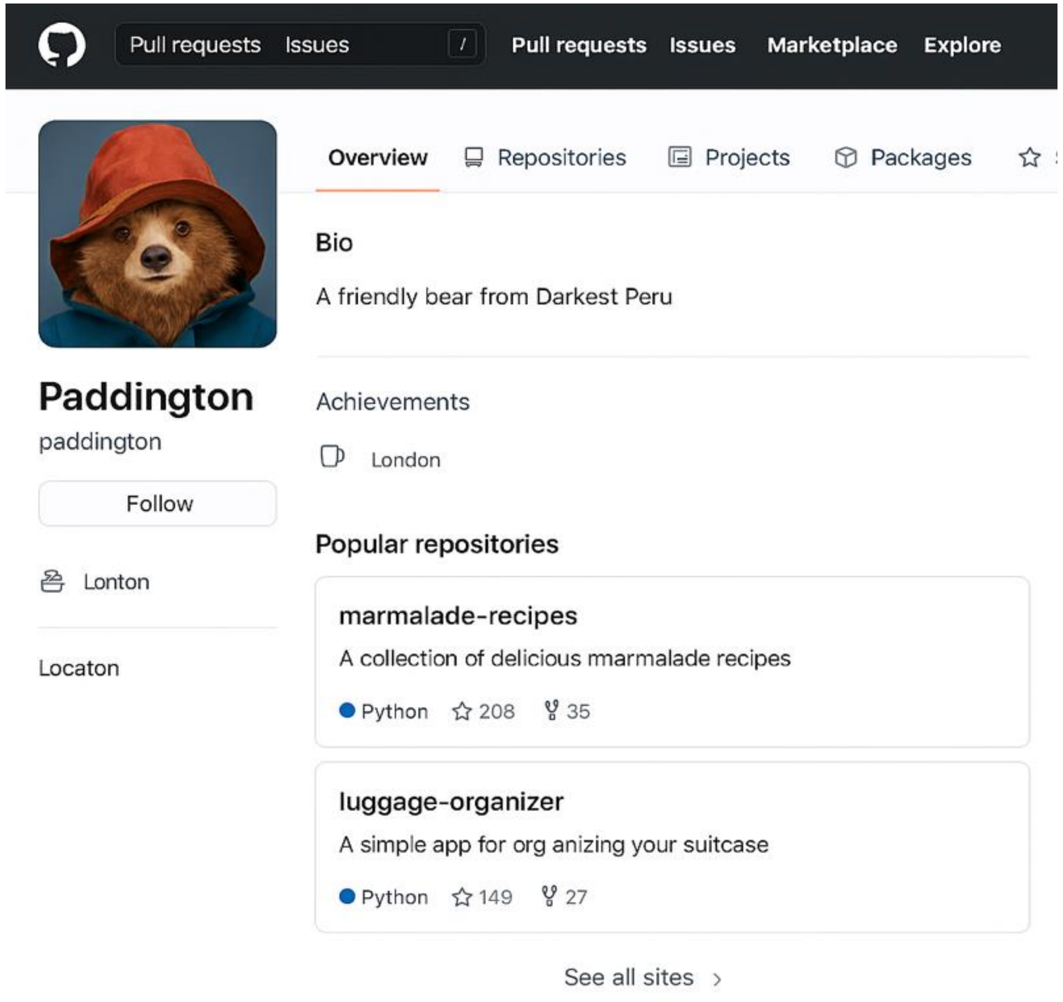
6. The browser (using the HTML parsing algorithm) parses the HTML page into a tree structure, namely the Document Object Model (DOM)



Everything aside, what do we actually see?

Feel free to check the inspector tab of your browser console to see the DOM

What we see


Pull requests Issues /
Pull requests Issues Marketplace Explore

Overview Repositories Projects Packages

Paddington
paddington
Follow

Lonton

Locaton

Bio

A friendly bear from Darkest Peru

Achievements

London

Popular repositories

marmalade-recipes
A collection of delicious marmalade recipes
● Python ☆ 208 35

luggage-organizer
A simple app for org anizing your suitcase
● Python ☆ 149 27

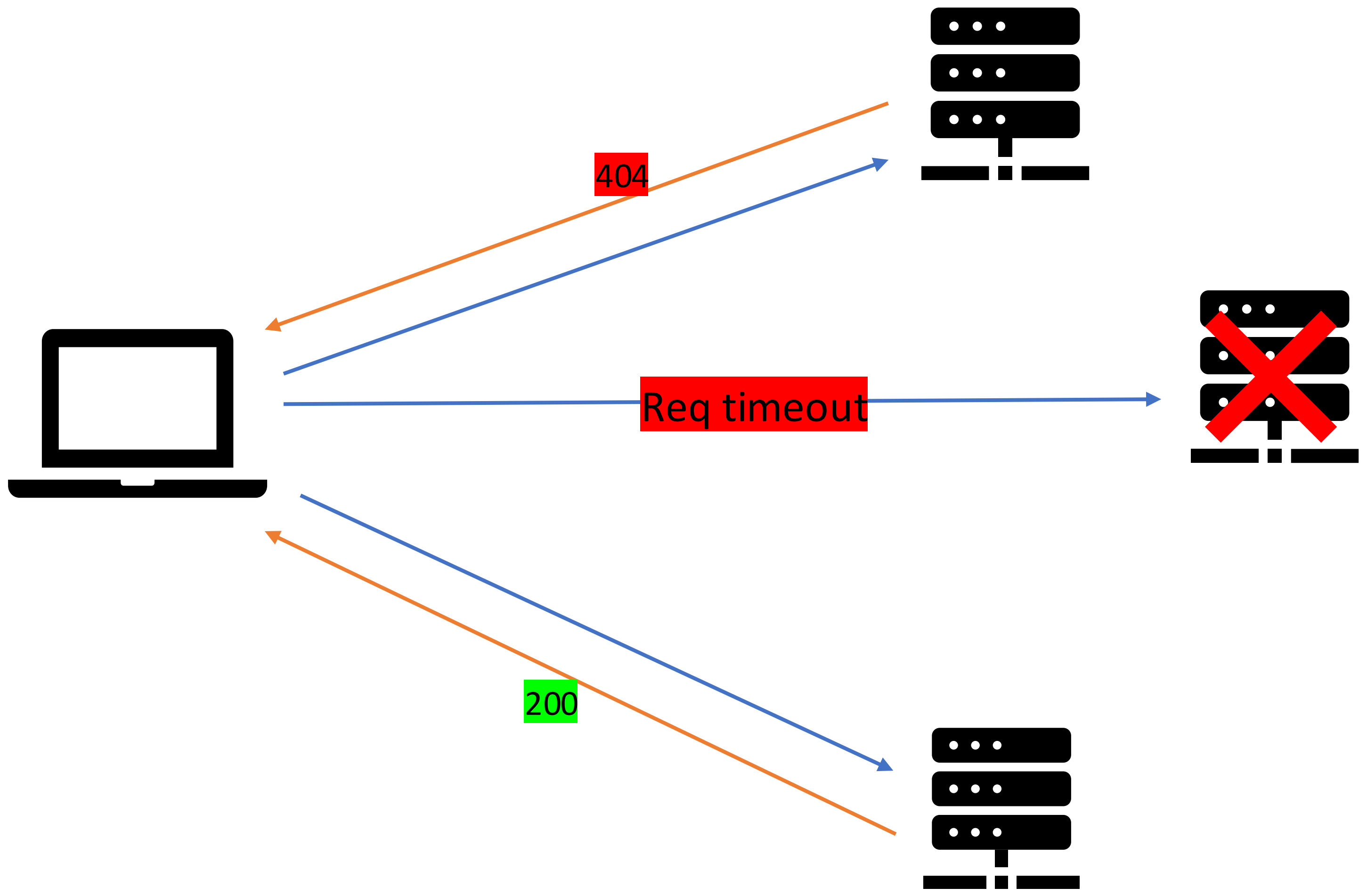
See all sites ›

HTTP

We saw that for each assest in an HTML page, the browser makes a new http call, so …

Do we make HTTP requests one at a time?

Asynchronous HTTP Requests

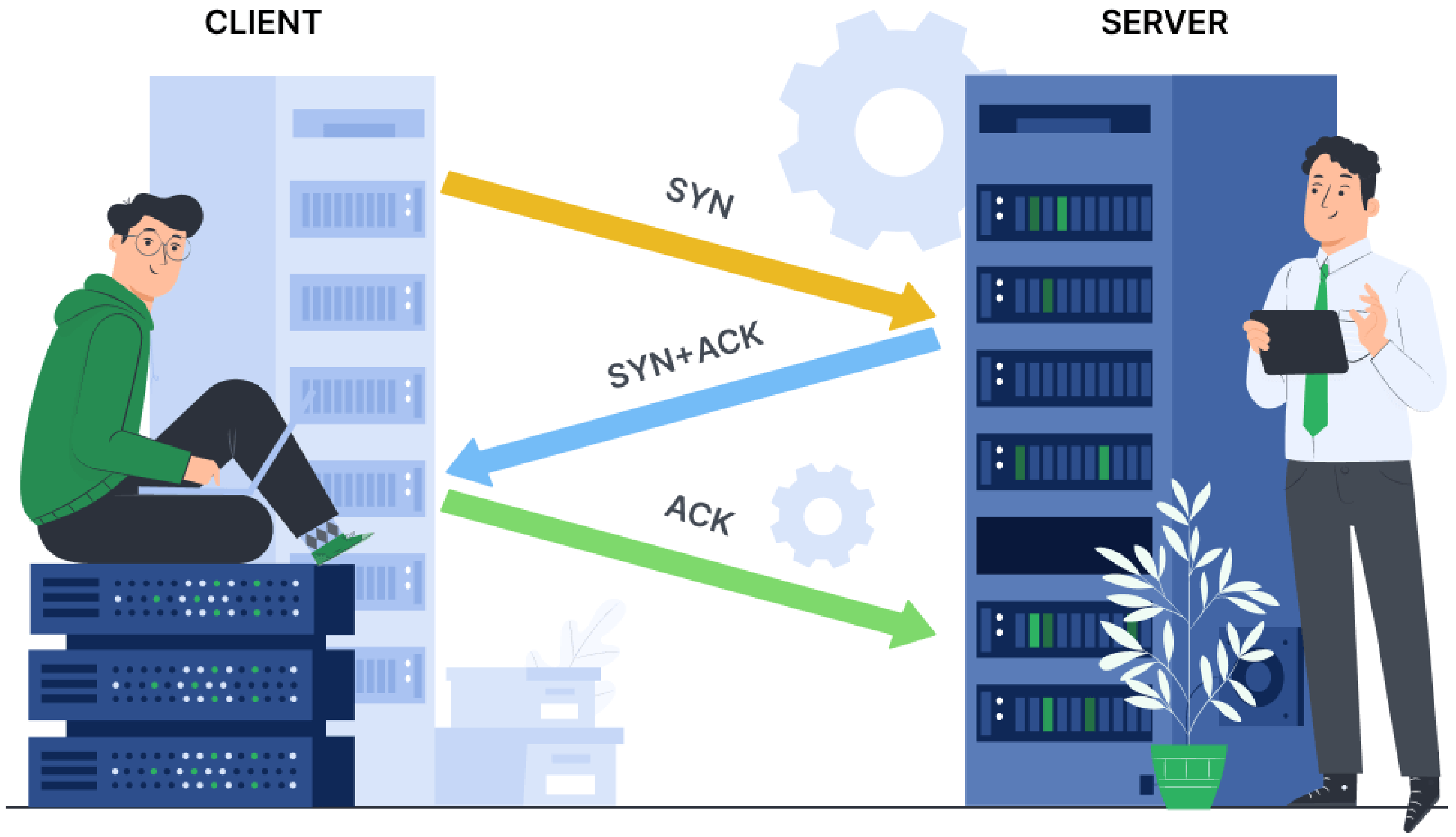


HTTP – A bit deeper

Which protocol does HTTP use (to establish a connection)?

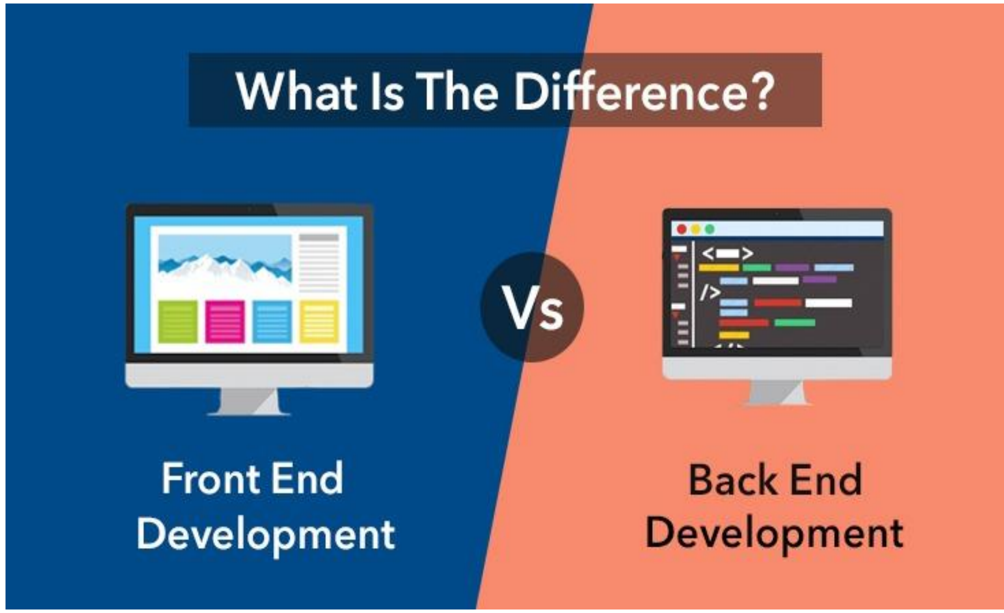
Transport Control Protocol

Three way handshake



Web Development

- Web Application Development is the development of applications that run on the web utilizing various tools and techniques.
- Web development is classified mainly into two parts:
  - Frontend development
  - Backend development

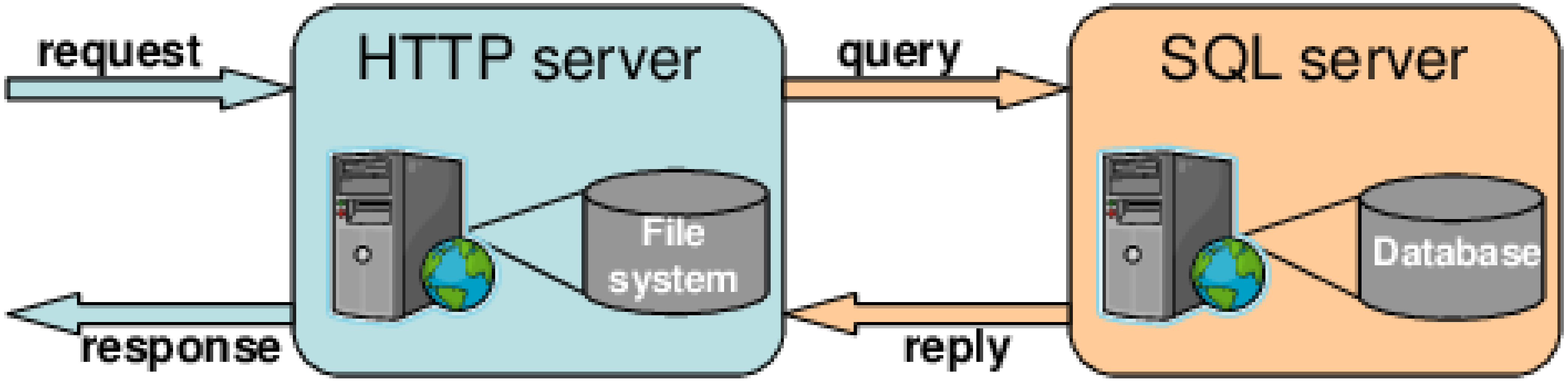


Web Development: Frontend

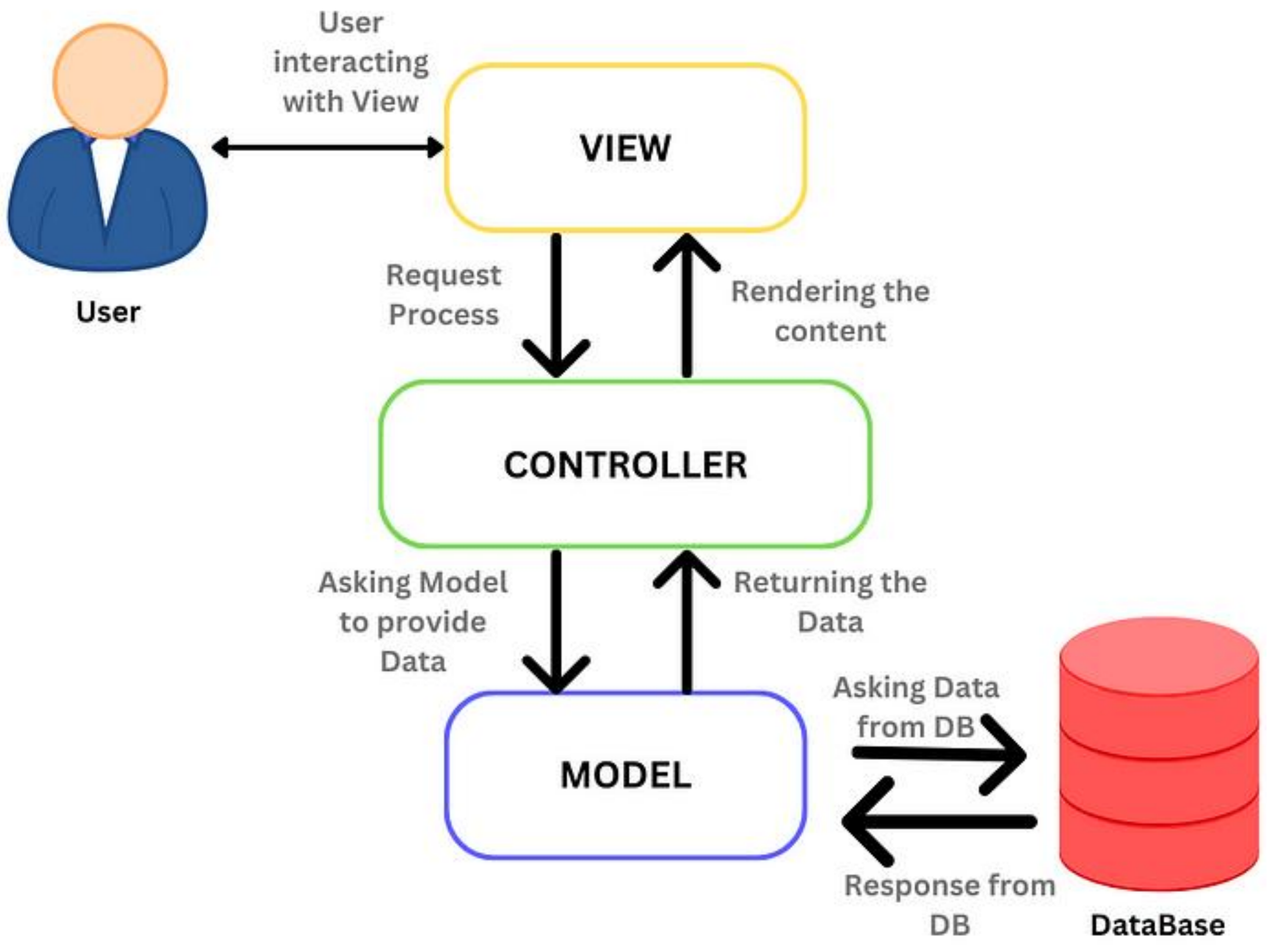
- Focuses on user interface (UI) and interaction, renders information retrieved from the backend side
- Also referred as client-side development

Web Development: Backend

- Focuses mainly on data processing and makes sure that everything is alright behind the scene (security, authentication, authorization, etc.)
- Also reffered as server-side development



An Architecture: Model View Controller (MVC)



What we learned

1. Internet & WWW
2. What and how the web works
3. HTTP requests & TCP connection
4. Web development: Frontend & Backend
5. MVC

Additional Reading Resources

- TCP/IP:
  - https://www.geeksforgeeks.org/tcp-ip-model/
  - https://stormwall.network/resources/terms/general/tcp-handshake
  - https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Web_standards/How_the_web_works
  - https://developer.mozilla.org/en-US/docs/Learn_web_development/Howto/Web_mechanics/How_does_the_Internet_work
- HTTP:
  - https://igoro.com/archive/what-really-happens-when-you-navigate-to-a-url/
  - https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods
- Web:
  - Video (4 min): https://www.youtube.com/watch?v=J8hzJxb0rpc
- MVC:
  - https://developer.mozilla.org/en-US/docs/Glossary/MVC

References

1. https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Web_standards/How_the_web_works
2. https://www.geeksforgeeks.org/html/what-is-http/
3. https://www.swhosting.com/en/blog/how-dns-works
4. https://home.cern/science/computing/birth-web/short-history-web
5. https://developer.mozilla.org/en-US/docs/Glossary/MVC