Front-End Web Development

Unit 1: Getting Started with Front-End

1. Getting Started

In This Unit

Title
Introductions
Course Overview
Assessment Overview
What will / will not be covered
How does the Internet / Web work
Setting up development environment and Hello world

Introductions

- About me
- Who are you?
- Why are you here?
- What do you hope to gain from this course?

UCD DUBLIN PROFESSIONAL ACADEMY

Course Outline

1. **Getting Started**
2. HTML - Structuring the Web
3. CSS - Styling the Web
4. JavaScript - Dynamic client-side scripting
5. CSS - Making Layouts
6. Introduction to Websites/Web Applications
7. CSS - Advanced
8. JavaScript - Modifying the Document Object Model (DOM)
9. Dynamic HTML
10. Web Forms - Working with user data
11. JavaScript - Advanced
12. Building a Web Application with JavaScript
13. Introduction to CSS Frameworks – Bootstrap
14. Building a Web Application with Svelte
15. SEO, Web security, Performance
16. Walkthrough project

UCD PROFESSIONAL ACADEMY

Learning Outcomes

- Competently write HTML and CSS code
- Create web page layouts according to requirements
- Using styles
- Add interactivity to a web page with JavaScript
- Access and display third-party data on the web page
- Leverage Bootstrap and Static Site Generator

Assessment

- Final Project - 100% of the grade

  - Design and Build functioning Website using HTML5, CSS (including Bootstrap), JavaScript (browser only)

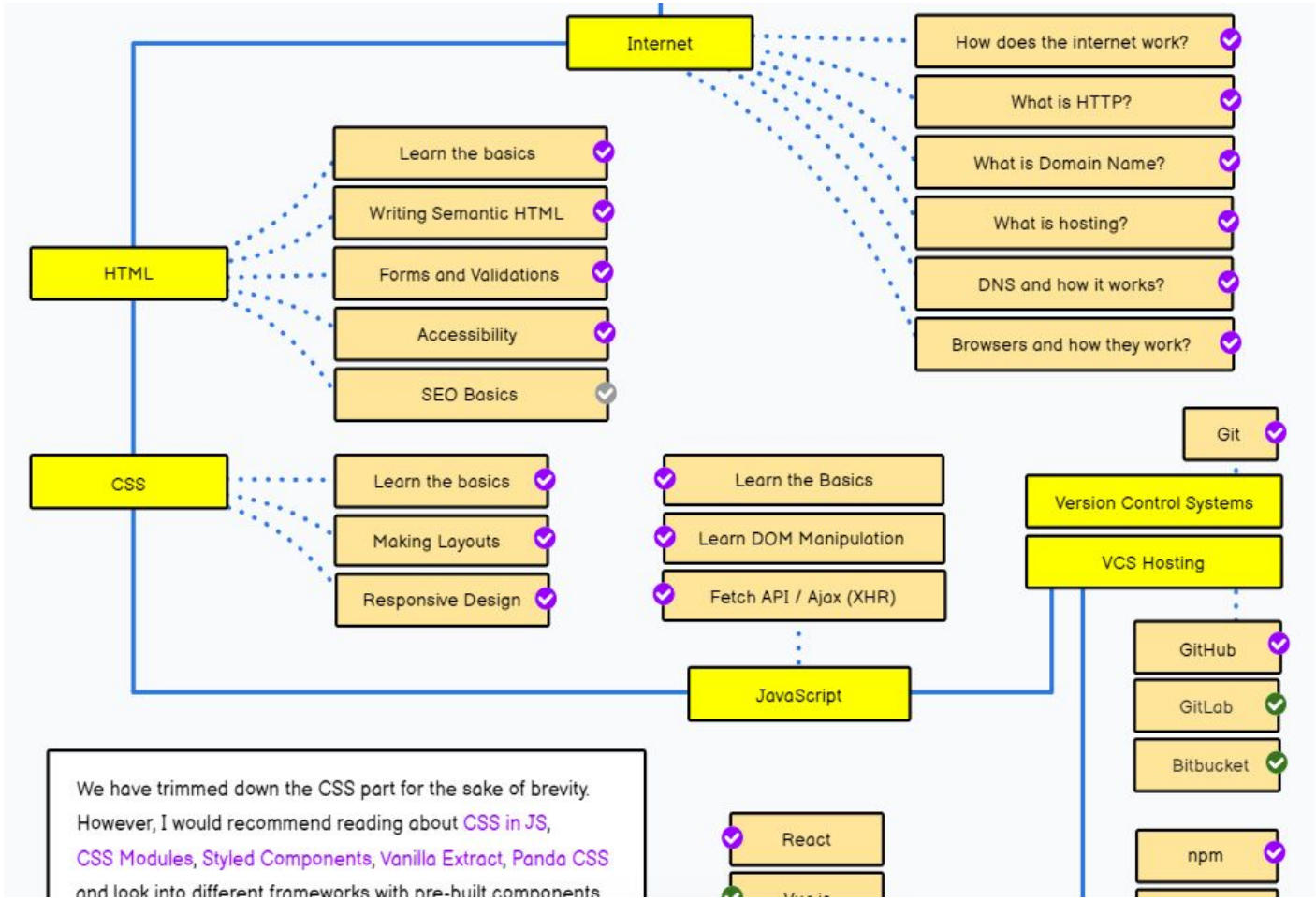
✔ Code will be managed in GitHub
✔ Website will be deployed to GitHub Pages
✔ All code to follow best practice and be documented

- Details and How-To-Guide are available on the course page under the section called Assessments

What will be covered?

Internet, HTML, CSS, JavaScript, Git, GitHub/Pages, Astro, Svelte

https://roadmap.sh/frontend



What will not be covered?

- Web Servers, Backend Development, Databases
- Serverside JavaScript, e.g. NodeJS, Express
- Testing
- Build tools, Gulp, WebPack, etc...
- Web Assembly
- Web API's

Course Approach

- Each Unit (3 hour workshop) is broken into 3 topics each lasting 1 hour each

Description	Timing
Lecture/Demo	20 mins
Exercise (breakout room)	35 mins
Short break	5 mins

- Student will be expected to have reviewed course slides/notes before the class
- Lecture time will be spent on reviewing main slides and walking through a detailed demo(s)
- Students will then complete an exercise in a breakout room. Lecturer will visit each breakroom to assist and answer any questions
- These timings are estimates and can be adjusted as required
- Students will be expected to complete exercises before the next unit/class

Break

Back in 5 minutes

How does the internet/web work?

The web is more a social creation than a technical one. I designed it for a social effect — to help people work together — and not as a technical toy. The ultimate goal of the Web is to support and improve our weblike existence in the world.

from “Weaving the Web” by Tim Berners-Lee

What is the Internet?

- A vast network of interconnected computers and devices that allows for communication and data exchange
- A global network that spans the globe and connects billions of people
- An essential part of modern life, used for various purposes, including communication, information access, and business

Timeline of the Internet

1965

Lawrence Roberts and Thomas Merril communicate with one another using computers connected via a low-speed dial-up telephone line in Massachusetts and California, creating the first Wide Area Network and laying the groundwork for the internet.

1969

ARPAnet, the first version of the internet, is created and used to link computers at UCLA and Stanford.

1974

Robert Kahn and Vinto Cerf publish "A protocol for Packet Network Communication" laying the groundwork for TCP/IP and much larger, interconnected computer networks. The word "internet"is first used.

1986

NSFNET is founded, creating the backbone and providing the investment needed to create the internet as we know it today.

1989

The first commercial ISP, the world launches.

1989

Tim Berners-Lee invents the world wide web, which opens the door for the internet to go mainstream.

1993

The first web browser available to the public, Mosaic launches.

1994

Netscape navigator is released in stores.

2000

Some 300 million people around the world are officially online.

2004

Facebook launches and the Web 2.0 begins to grow rapidly.

2007

The iPhone is released, giving rise to the mobile revolution.

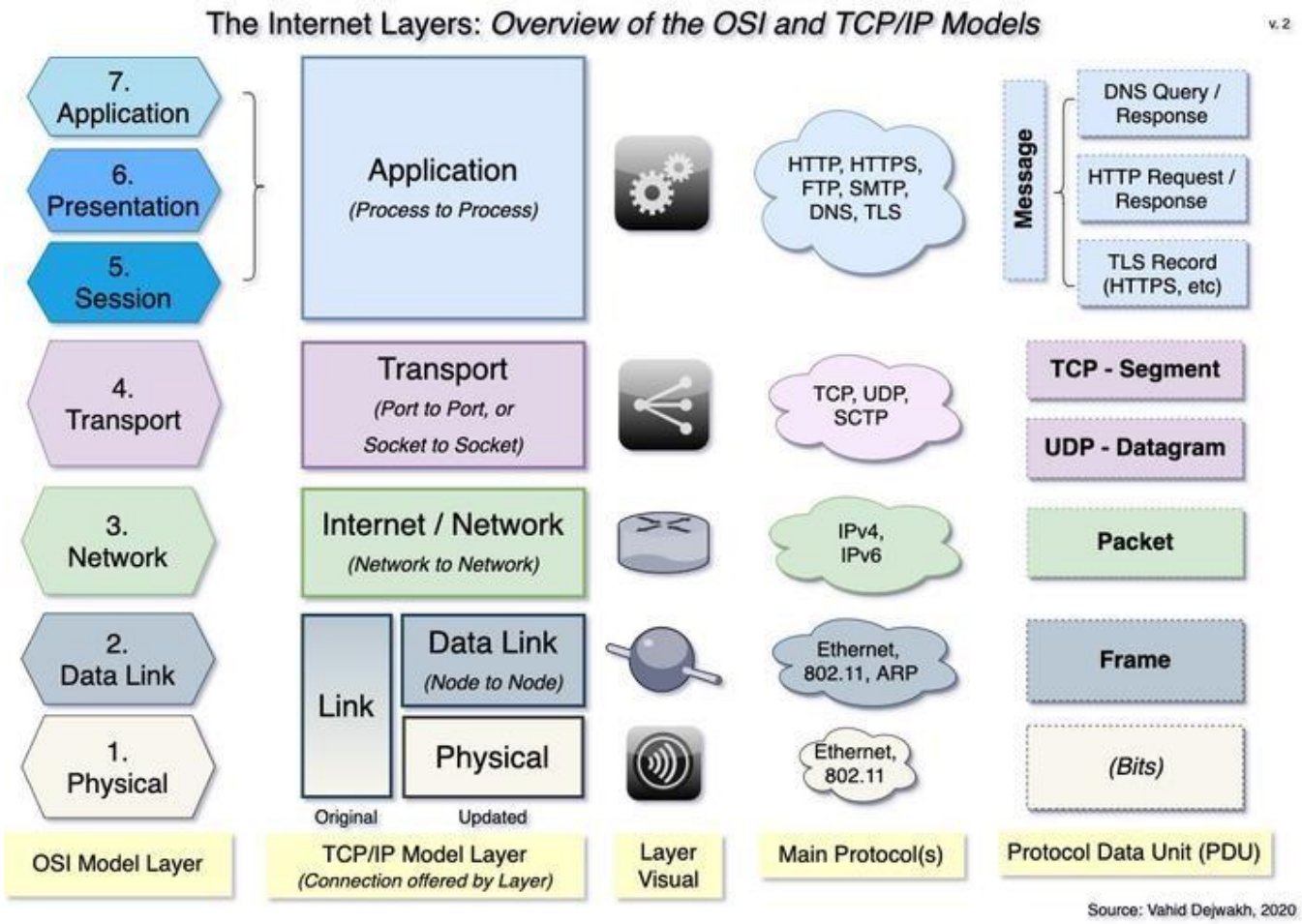
2010

1.966 billion people are online worldwide.

2022

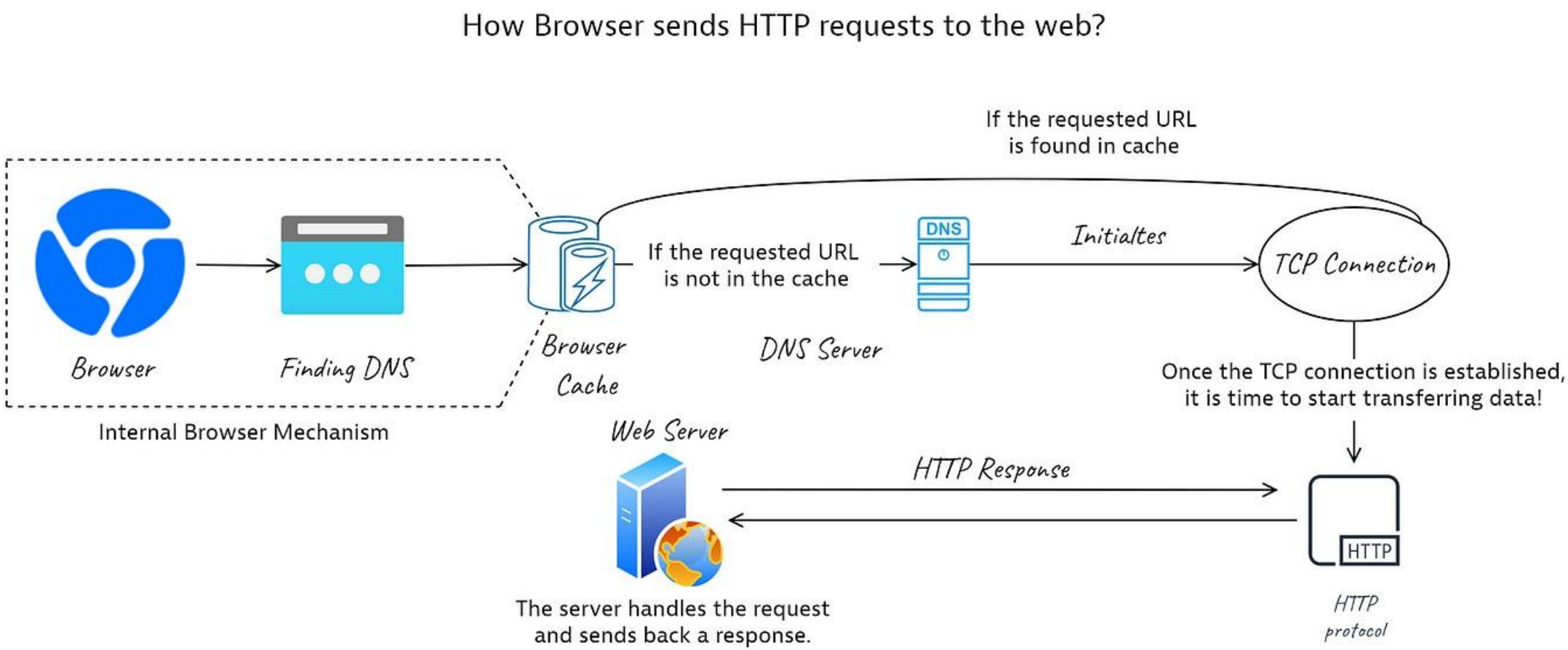
5 billion people are online

Internet Layers



The Web (World-Wide-Web/WWW) is an application built on top of the Internet

Browsers & how they work



What is HTTP

Hyper**t**ext **T**ransfer **P**rotocol (HTTP) is an application-level protocol for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.

Aspects of HTTP

HTTP is simple

HTTP is generally designed to be simple and human-readable, even with the added complexity introduced in HTTP/2 by encapsulating HTTP messages into frames. HTTP messages can be read and understood by humans, providing easier testing for developers, and reduced complexity for newcomers. Can be viewed in the browser developer tools.

HTTP is extensible

Introduced in HTTP/1.0, HTTP headers make this protocol easy to extend and experiment with. New functionality can even be introduced by a simple agreement between a client and a server about a new header's semantics.

Aspects of HTTP

- **HTTP is stateless, but not sessionless**

  HTTP is stateless: there is no link between two requests being successively carried out on the same connection.

  This immediately has the prospect of being problematic for users attempting to interact with certain pages coherently, for example, using e-commerce shopping baskets. But while the core of HTTP itself is stateless, HTTP cookies allow the use of stateful sessions. Using header extensibility, HTTP Cookies are added to the workflow, allowing session creation on each HTTP request to share the same context/state

- **HTTP and connections**

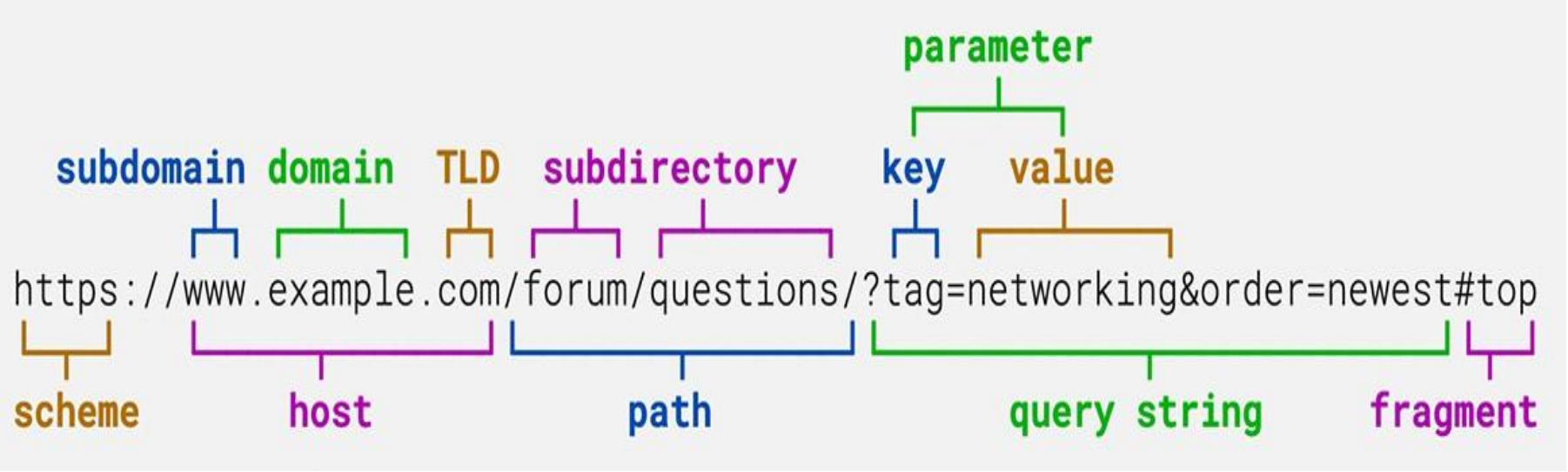
  A connection is controlled at the transport layer, and therefore fundamentally out of scope for HTTP. HTTP doesn't require the underlying transport protocol to be connection-based; it only requires it to be reliable, or not lose messages (at minimum, presenting an error in such cases). HTTP relies on the TCP standard, which is connection-based.

  The default behavior of HTTP/1.0 is to open a separate TCP connection for each HTTP request/response pair. HTTP/2 multiplexing messages over a single connection, helping keep the connection warm and more efficient.

Identifying resources on the Web

The target of an HTTP request is called a "resource", whose nature isn't defined further; it can be a document, a photo, or anything else.

Each resource is identified by a Uniform Resource Identifier (URI) used throughout HTTP for identifying resources

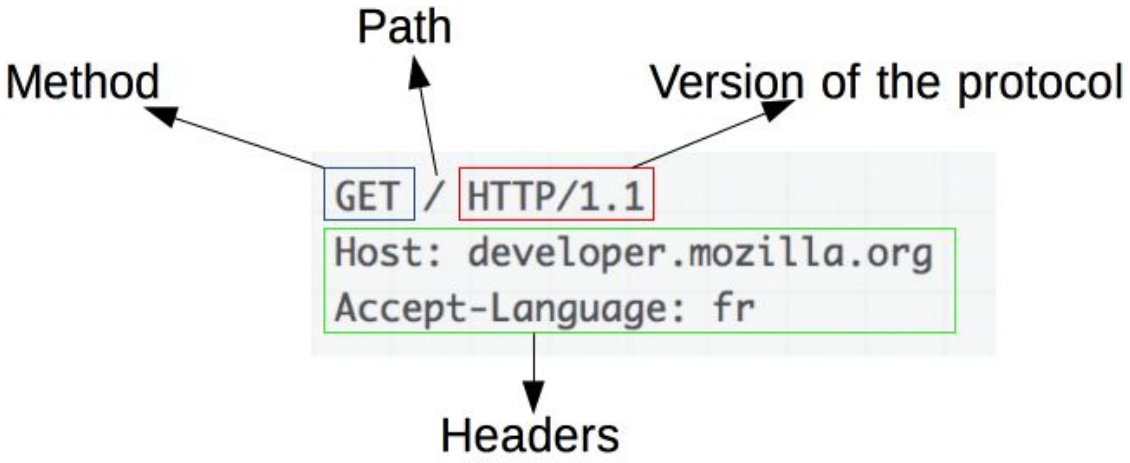


Common Schemes (Protocols)

Scheme	Description	Example
data	Data URLs	...
file	Host-specific file names	file:///C:/Users/jbyrne/.../demos/simple-layout/index.html
ftp	File Transfer Protocol	ftp://example.org/resource.txt
http/https	Hypertext transfer protocol (Secure)	https://developer.mozilla.org/en-US/docs/Learn
mailto	Electronic mail address	mailto:help@supercyberhelpdesk.info
ssh	Secure shell	
tel	telephone	tel:+1-816-555-1212
urn	Uniform Resource Names	urn:isbn:9780141036144
view-source	Source code of the resource	view-source:https://developer.mozilla.org/
ws/wss	WebSocket connections (Secure)	

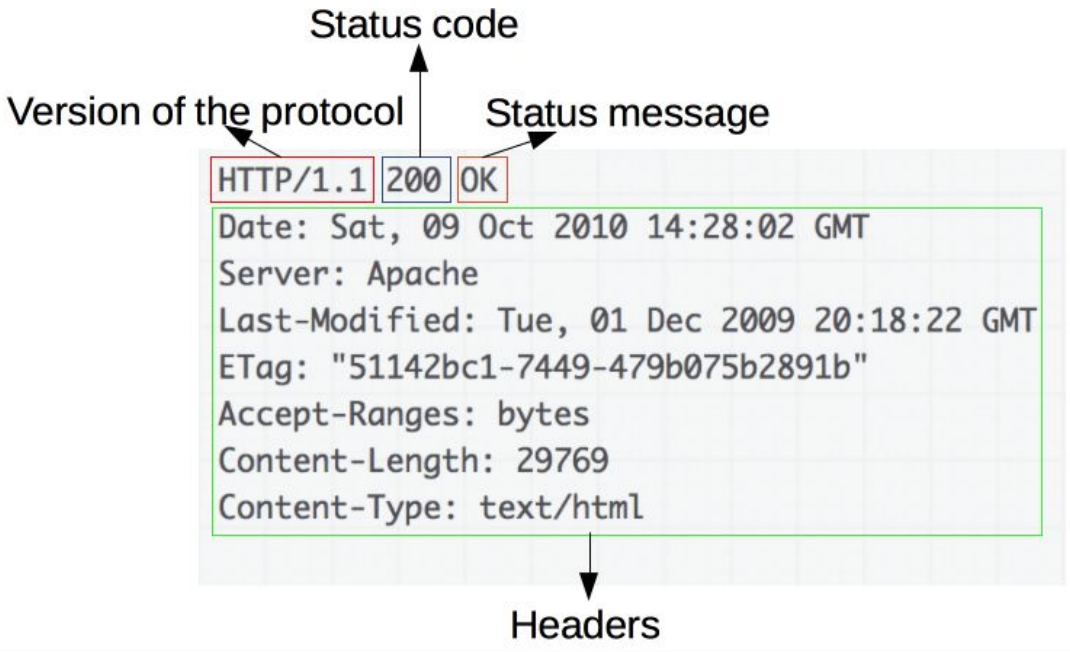
HTTP Messages - Request

- A HTTP **Method** usually a verb like `GET`, `POST`, `PUT`, `DELETE`
- The path of the resource to fetch; the URL of the resource stripped from elements that are obvious from the context, for example without the protocol, the domain, or the TCP port (here, 80)
- The version of the HTTP protocol
- Optional headers that convey additional information for the servers.
- A body, for some methods like POST, similar to those in responses, which contain the resource sent



HTTP Messages - Response

- The version of the HTTP protocol they follow
- A status code, indicating if the request was successful or not, and why
- A status message, a non-authoritative short description of the status code HTTP headers, like those for requests
- Optionally, a body containing the fetched resource



What is Domain Name Service (phonebook of the Internet)?

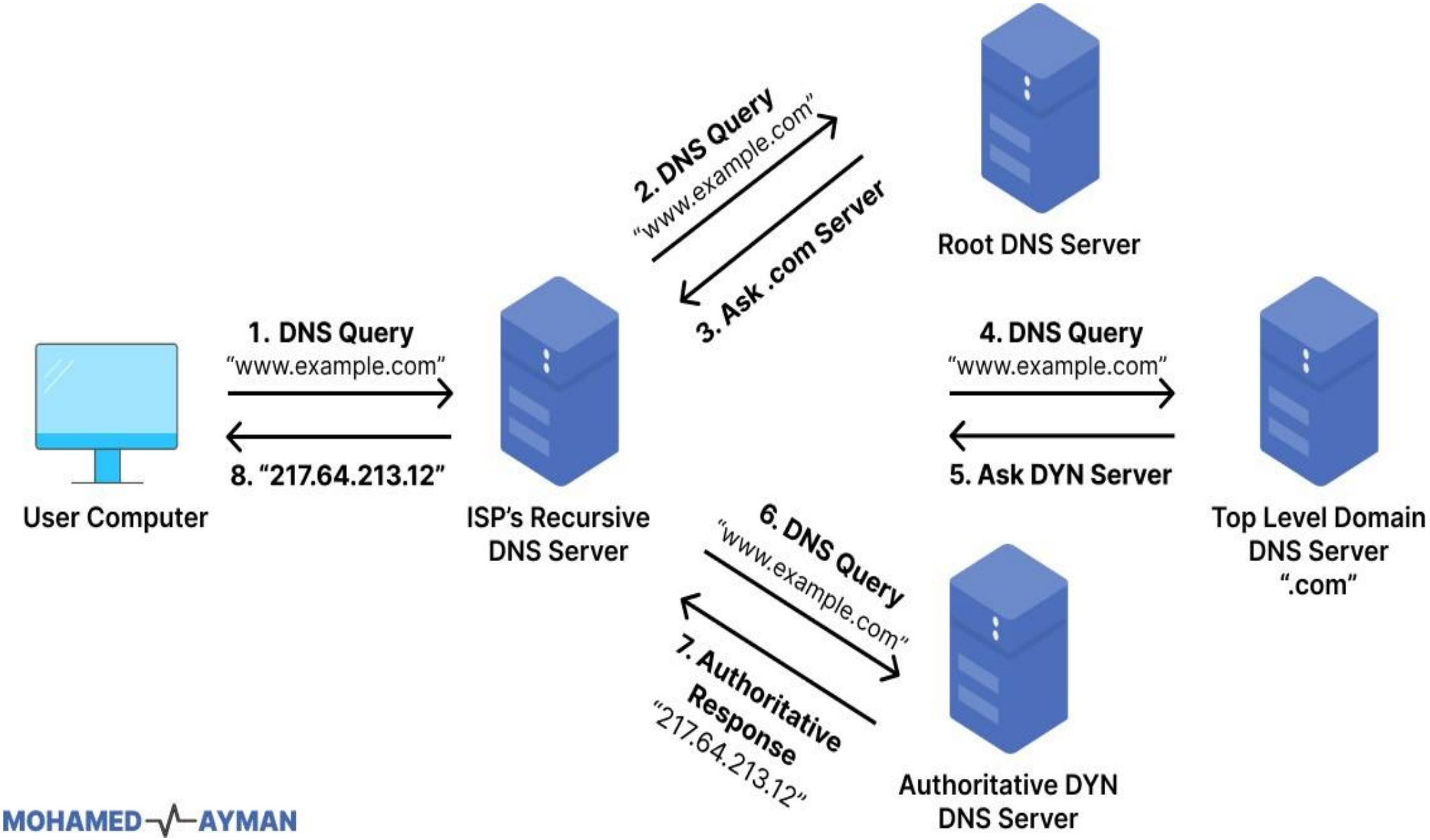
Humans access information online through domain names, like irishtimes.com or mozilla.org

DNS translates domain names to IP addresses so browsers can load Internet resources

Each device connected to the Internet has a unique IP address which other machines use to find the device

DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6)

UCD DUBLIN PROFESSIONAL ACADEMY



MOHAMED AYMAN

What is hosting?

Web hosting is a service that provides storage for the files that make up your website and the software, physical hardware, and network infrastructure that makes your website available to others on the internet.

Web hosting service providers offer a variety of hosting options, ranging from expensive to inexpensive:

- The amount of storage space and computing capacity allocated specifically for your site
- The degree to which your site shares computing resources with other sites or is isolated from the impact of other sites sharing the same resources
- The additional capabilities and services offered (e.g., number of email inboxes, blogging capabilities, etc.)
- The degree of control and flexibility you have (e.g., which operating system (OS) and/or content management system (CMS) you can use, support for special web applications, etc.)
- The extent to which you manage your web site or have the service provider manage it for you
- Types:
  - Shared, e.g. https://blacknight.com/
  - Virtual private server (VPS) hosting or cloud-based VPS, e.g. https://aws.amazon.com/
  - Dedicated, e.g. run a server machine inhouse, requires a public IP address

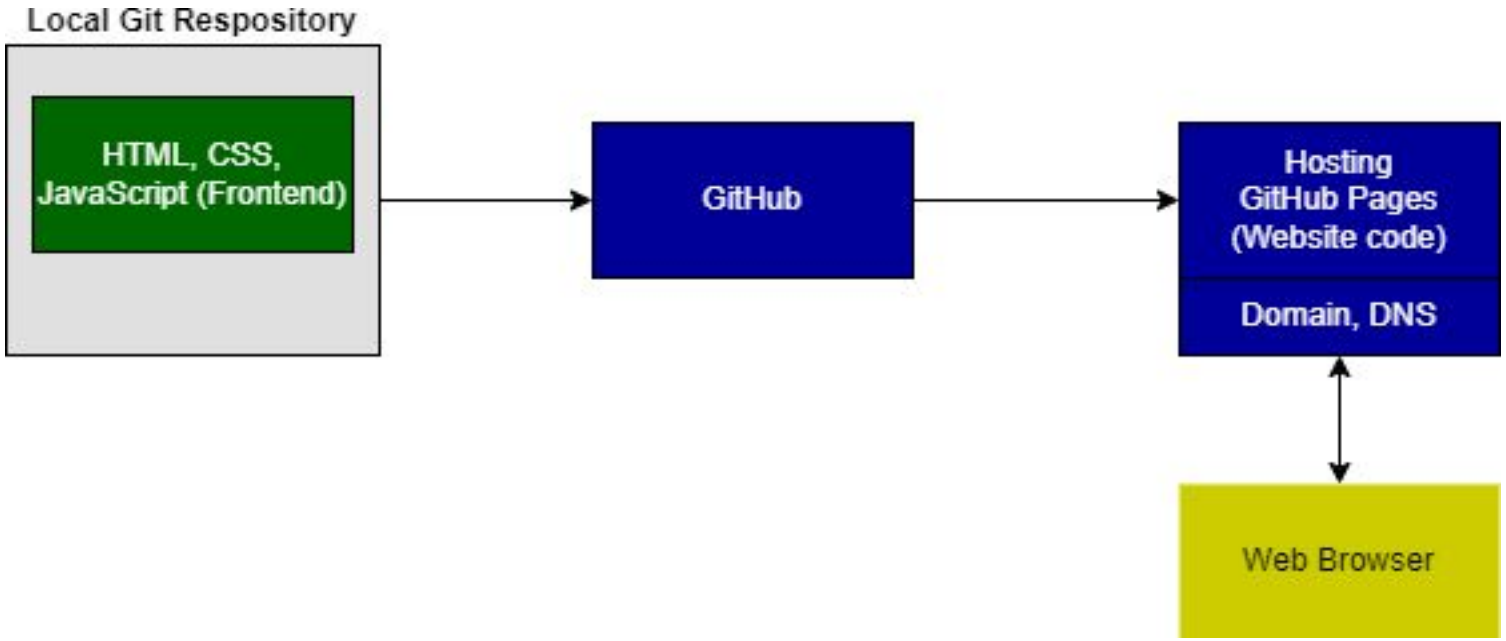
Activity

- Open the Chrome browser
- Right-click on the page and click on the "inspect" option
- Click on the "Network" tab
- Load any webpage

Break

Back in 5 minutes

Development Environment



3 Step Process (iterative)

1. Develop and test locally
2. Commit and push to GitHub
3. Deploy to GitHub Pages

Software to install (laptop/desktop)

Software	Download/Install	Information
Google Chrome*	Download & Install	Developer Tools Documentation
Visual Code	Download & Install, Extensions (optional) - GitLens, CSS Formatter, Tabnine, Babel JavaScript, LiveServer, Markdown All in One	Setting up VS Code
Git	Download & Install	Introduction to Git & GitHub
GitHub	Signup to GitHub	

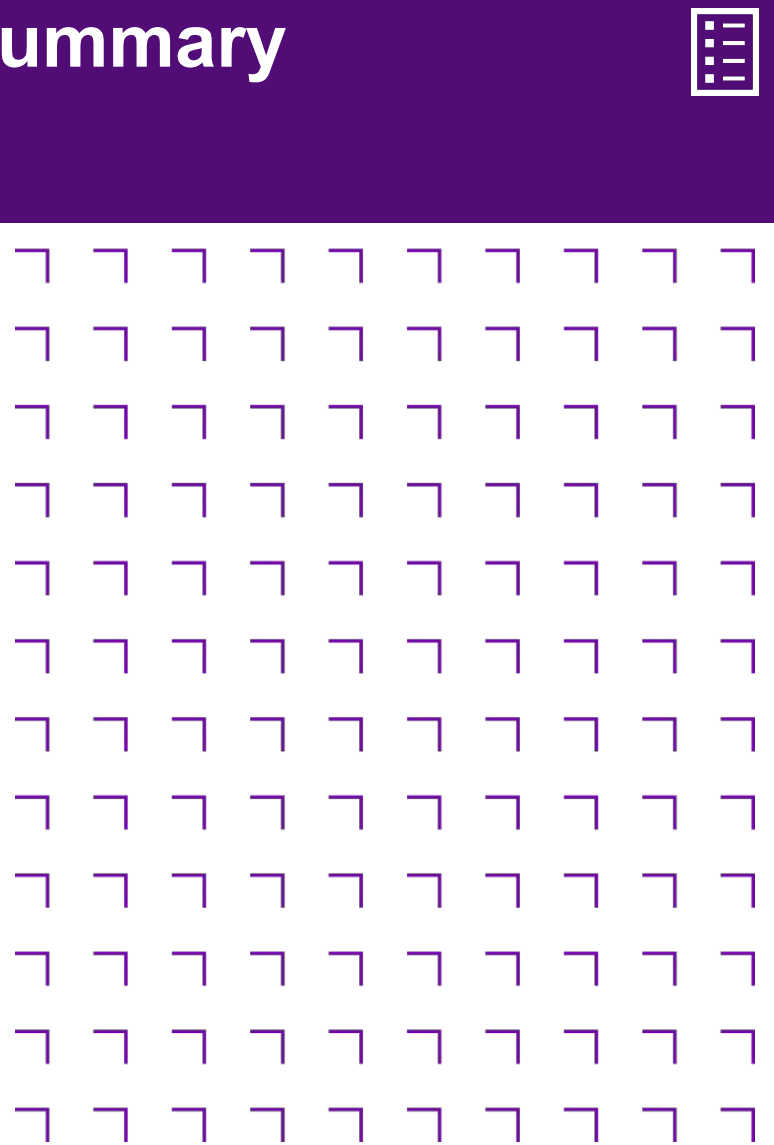
*preferred browser as it will be the one used in class

Demo/Exercise

- Create a new folder
- Create a new file called `hello-world.html` and open using Visual Code
- Paste the following html code into the file
- Save the changes and then view the file in Chrome by double-clicking it in file explorer

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>
      <a href="https://en.wikipedia.org/wiki/%22Hello,_World!%22_program">information on "Hello, World! program</a>
    </p>
  </body>
</html>
```

Summary



- Course overview and Introductions
- How the Web works
- Setting up Development Environment and Hello World example
- Complete the remaining exercises before next class

Assessment

- Final Project - 100% of the grade

  - Design and Build functioning Website using HTML5, CSS (including Bootstrap), JavaScript (browser only)

✔ Code will be managed in GitHub
✔ Website will be deployed to GitHub Pages
✔ All code to follow best practice and be documented

- Details and How-To-Guide are available on the course page under the section called Assessments