Web.1

▼ What are computers connected to the web called?

Clients and servers

▼ What are clients?

the typical web user's internet-connected devices and web-accessing software available on those devices (usually a web browser like Firefox or Chrome)

▼ What are servers?

computers that store webpages, sites, or apps

▼ When a client device wants to access a webpage, what happens?

a copy of the webpage is downloaded from the server onto the client machine to be displayed in the user's web browser

▼ What is TCP/IP?

Transmission Control Protocol and Internet Protocol are communication protocols that define **how data should travel** across the internet.

▼ What is DNS?

Domain Name Servers are like an address book for websites. When you type a web address in your browser, the browser looks at the DNS to find the website's real address before it can retrieve the website.

▼ What is HTTP?

Hypertext Transfer Protocol is an application protocol that defines a **language** for clients and servers to speak to each other.

▼ When you type a web address into your browser?

1. The browser goes to the DNS server, and finds the real address of the server that the website lives on.

2. The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client. This message, and all other data sent between the client and the server, is sent across your internet connection using TCP/IP.

3. If the server approves the client's request, the server sends the client a "200 OK" message, and then starts sending the website's files to the browser as a series of small chunks called data packets

4. The browser **assembles** the small chunks into a complete web page and displays it to yo

▼ What is the order in which component files are parsed?

HTML then CSS then JavaScript.

▼ What happens when HTML is parsed?

a DOM tree structure is generated from it

▼ What happens when CSS is parsed?

styles are applied to the appropriate parts of the DOM tree and the visual representation of the page is painted to the screen

▼ What is an IP address?

a unique location on the web

63.245.215.20

▼ When two computers need to communicate, you have to link them how?

either physically (usually with an Ethernet cable) or wirelessly (for example with WiFi or Bluetooth systems)

▼ Each computer on a network is connected to a special tiny computer called what?

a router

▼ What does a router do?

it makes sure that a message sent from a given computer arrives at the right destination computer

▼ By connecting computers to routers, then routers to routers, we are able to scale...

infinitely

▼ How to connect to other networks?

The telephone infrastructure already connects your house with anyone in the world.

To connect our network to the telephone infrastructure, we need a special piece of equipment called a modem.

▼ What is a modem?

A modem turns the information from our network into information manageable by the telephone infrastructure

▼ How can we send the messages from our network to the network we want to reach?

we will connect our network to an Internet Service Provider (ISP)

▼ What is an ISP?

An ISP is a company that manages some special routers that are all linked together and can also access other ISPs' routers.

So the message from our network is carried through the network of ISP networks to the destination network.

▼ What does the Internet consist of?

an infrastructure of networks

▼ What computers can send messages intelligible to web browsers?

web servers

▼ The Internet is an infrastructure, whereas the Web is a what?

a service built on top of the infrastructure

▼ What are Intranets?

private networks that are restricted to members of a particular organization

▼ What are Extranets?

Extranets are very similar to Intranets, except they open all or part of a private network to allow sharing and collaboration with other organizations.

They are typically used to safely and securely share information with clients and stakeholders who work closely with a business.

▼ What is a web page?

A document which can be displayed in a web browser such as Firefox, Google Chrome, Opera, Microsoft Internet Explorer or Edge, or Apple's Safari.

▼ What is a website?

A collection of web pages which are grouped together and usually connected together in various ways.

▼ What is a web server?

A computer that hosts a website on the Internet.

▼ What is a search engine?

A web service that helps you find other web pages, such as Google, Bing, Yahoo, or DuckDuckGo.

▼ What is a TLD or Top-Level Domain?

TLDs tell users the general purpose of the service behind the domain name.

The most generic TLDs (.com, .org, .net).

▼ What is the Secondary Level Domain (SLD)?

The label located right before the TLD

e.g. google in google.com

▼ What is a subdomain?

For any domain you control (e.g. mozilla.org), you can create "subdomains" with different content located at each, like developer.mozilla.org, iot.mozilla.org, or wiki.developer.mozilla.org

▼ How does a DNS request work?

1. Type the website in your browser's location bar.

2. Your browser asks your computer if it already recognizes the IP address identified by this domain name (**using a local DNS cache**). If it does, the name is translated to the IP address and the browser negotiates contents with the web server.

3. If your computer does not know which IP is behind the website's name, it goes on to ask a DNS server, whose job is precisely to tell your computer which IP address matches each registered domain name.

4. Now that the computer knows the requested IP address, your browser can negotiate contents with the web server.

▼ How to prevent XSS attacks?

- **Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.

- **Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.

- **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

- **Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.