

# WWW - INTRO

Ilkka Kylmäniemi, Patrick Ausderau, Ulla Sederlöf



# CONTENT

- WWW
- How does web work
- Basic structure of HTML-documents
- Markup, elements and attributes

# WWW HISTORY

- At the turn of the 1980s and 1990s, under the leadership of Tim Berners-Lee, the original World Wide Web was developed at CERN. Its three leading ideas are
  - HTML – Hypertext Markup Language
  - HTTP – Hypertext Markup Protocol
  - Web servers and Web browsers
- The problem at the time was that the article databases used by researchers were really clumsy to use. So a nicer user interface was needed, where, for example, you didn't have to manually write the name of every computer you contacted, but you could navigate by just clicking on text links. The original motivation for the development of the WWW was therefore to make research databases easy to use.

# WWW AND INTERNET, I KNOW THESE.. DO I?

- Internet
  - A worldwide system of interconnected information networks that uses the Internet Protocol (TCP/IP) to connect information technology devices to each other.
- WWW
  - The World Wide Web, or WWW, is a distributed hypertext system operating on the Internet and accessible via the Internet. The WWW consists of web sites and applications connected to the Internet, as well as client devices (such as computers and phones) that have access to web content.

# WWW AND INTERNET, I KNOW THESE.. DO I?

- The Web is not the same thing as the Internet, it is a part of it (the Internet has also FTP, Mail etc.)
- [https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)
- WWW uses the Hypertext Transform Protocol (HTTP) or the Secure Hypertext Transform Protocol (HTTPS)
- A protocol is a set of rules that govern how systems communicate. For networking they govern how data is transferred from one system to another.

# SERVER, CLIENT, USER

- The main focus of web development is communication. It is basically two-party communication via the HTTP protocol:
  1. The Server - responsible for serving pages.
  2. The Client - requests pages from the Server, and displays them to the user. In most cases, the client is a web browser.
- The User - The user uses the Client in order to surf the web, watch online videos, fill contact form, listen to music, etc.
- Web programming uses the terms server-side programming and client-side programming. That is, server and browser programming.

# SERVER-SIDE PROGRAMMING

- Server-side programming, is the general name for the kinds of programs which are run on the Server.
  - Process user input.
  - Display pages.
  - Structure web applications.
  - Interact with permanent storage (SQL, files).
- Common programming languages used to create server applications :
  - PHP
  - Java
  - JavaScript
  - Python
  - C#
  - Etc.



# CLIENT-SIDE PROGRAMMING

- Client-side programming means the development of all applications that run in the browser, such as:
  - Web-pages
  - Interactive web applications
- Client-side programming languages:
  - HTML
  - CSS
  - JavaScript
  - (in the future via WebAssembly: C, Rust, Kotlin, Swift...)



# AN EXAMPLE OF A CLIENT-SERVER MODEL

- The User opens a web browser (the Client).
- The User browses to <https://oma.metropolia.fi>
- The Client (on the behalf of the User), sends a request to <http://oma.metropolia.fi> (the Server), for their home page.
- The Server then acknowledges the request, and replies the client with some meta-data (called headers), followed by the page's source.
- The Client then receives the page's source, and renders it into a human viewable website.

# UNIFORM RESOURCE LOCATOR (URL)

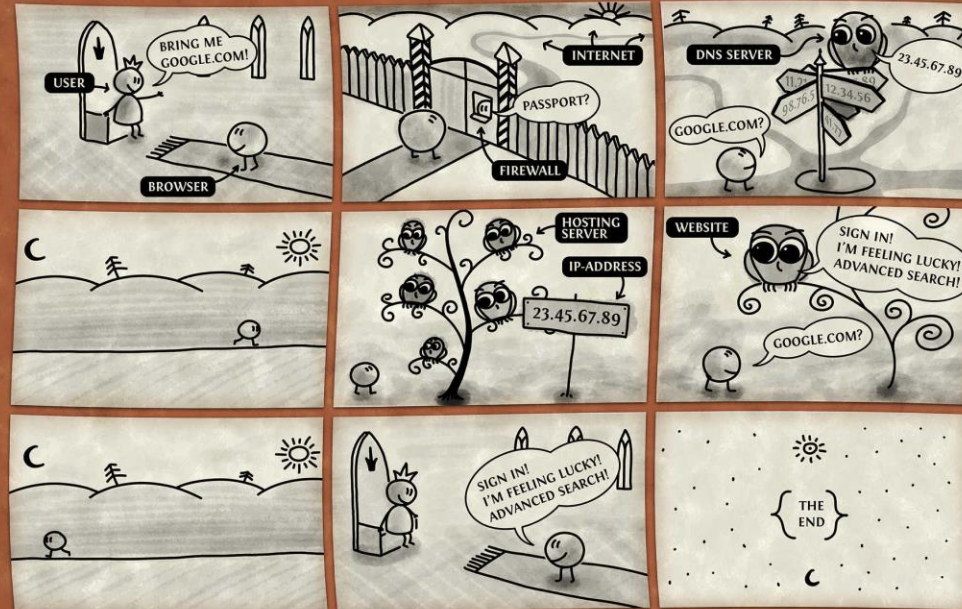
- Each URL follows the same format:

*scheme://username:password@domain:port/path?query\_string#anchor*

- Scheme = communication protocol. In practice http or https
- Username and password are not usually used for security reasons
- Domain = server address or IP address. E.g. metropolia.fi or 195.148.144.10
- Port = TCP port number. http = 80 and https = 443. In practice, always omitted
- Path = more precise path and/or file name for the requested resource/document
- Query\_string = can be used to pass values, for example to a script. E.g.  
<https://www.google.com/search?q=keyword>
- Anchor = allow to navigate to a specific part of a page.

# HOW INTERNET WORKS

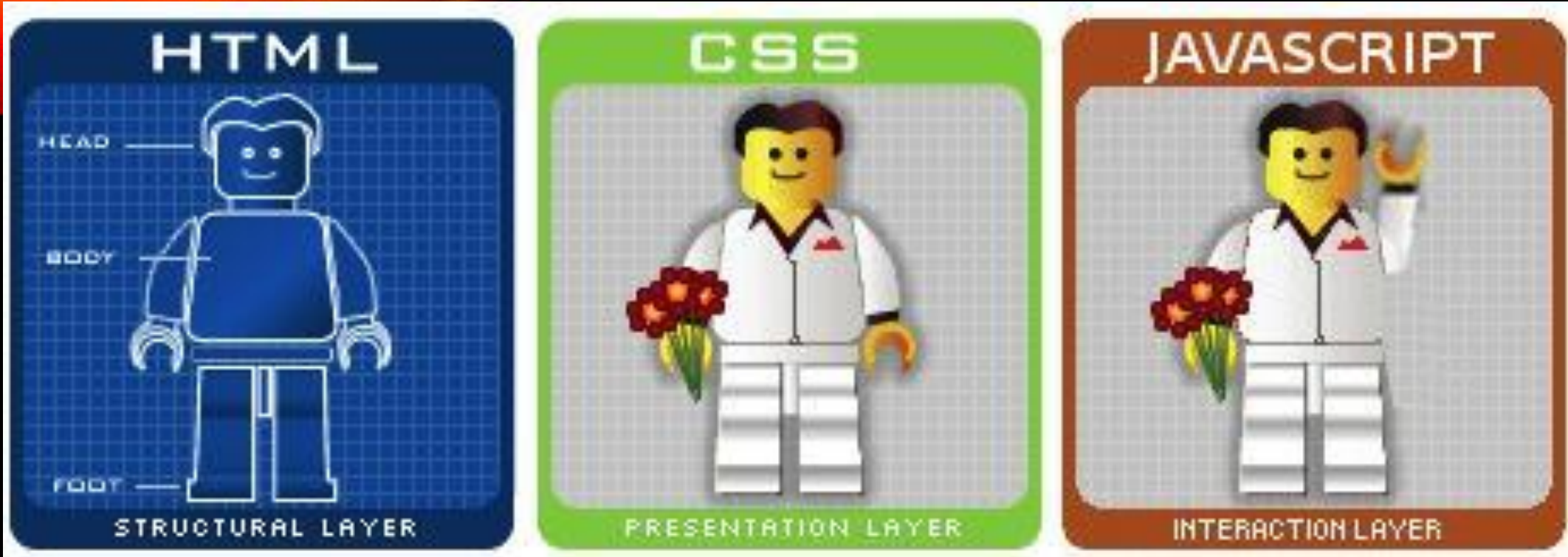
- [https://vlad.studio/wallpaper/how\\_internet\\_works](https://vlad.studio/wallpaper/how_internet_works)



# CLIENT-SIDE TECHNOLOGIES

- HTML
  - The structure of a web page
- CSS
  - The appearance of a web page
- JavaScript
  - The functionalities in a web page
- HTTP/HTTPS
  - Communicating with the server
- AJAX
  - Dynamic communication with the server





HTML + CSS + JAVASCRIPT

# HTML

- HTML = Hyper Text Markup Language
- HTML is a markup language, not a programming language
- A markup language is a set of markup tags. E.g. <p>
- The tags form elements with the content:
  - <p>Here is a text paragraph.</p>
  - <p> = opening tag
  - Here is a text paragraph. = content
  - </p> = closing tag
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages
- Filename extension \*.html or \*.htm
- It is the job of the Web browser to interpret these tags and render the text and other elements accordingly.

# HTML-ELEMENTS

- An HTML tag has '<' and '>' characters around the tag name. For example, the <header> tag is used to mark the header area.
- HTML tags are usually in pairs. Eg <p></p>
- The first is the opening tag and the latter is the closing tag.
- Together with the content, they form an HTML element: <p>Text</p>
- Some elements are 'empty'. That is, they do not have a closure tag. E.g. image element: 
- Elements are often nested:  
<p>Here is some <strong>bolded</strong> text</p>
  - In this case, it is important that the opening and closing tags are in the correct order



# HTML ATTRIBUTES

- HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes come in name/value pairs like: name="value"

`<a href="http://metropolia.fi"> Link to Metropolia's website</a>`

# WEB BROWSER

- Software that allows users to find, access, display and view web pages
- Popular browsers
  - Google Chrome
  - Mozilla Firefox
  - Apple Safari
  - Microsoft Edge
- Browsers convert HTML pages into human-readable content
- When a browser loads a web page, it processes the HTML, which may contain text, links, and references to images and other items like CSS and JavaScript files. The browser then renders these objects in the browser window after processing them.
- <https://web.dev/howbrowserswork/>

# DEVELOPER TOOLS

The image shows a web browser displaying the Metropolia website. The website has a yellow header with the text "Ukrainan tilanne ja vaikutukset Metropolian toimintaan". Below the header, there are language selection buttons for "FI" and "EN", and a search bar with the text "ETSI". The main content area features the Metropolia logo and a large image of a man pointing. A white overlay box contains the text "Tervetuloa uudet opiskelijat!" and "Syksyllä aloittavien opiskelijoiden ohjeet". Below this, there are three smaller images of people, each with a caption: "Jaa antailla", "Uudelle opiskelijalle", and "Täydennysko".

On the right side, the browser's developer tools are open, showing the "Elements" panel with the HTML structure of the page. The "Styles" panel shows the default styles for the body element, including font-family, font-size, line-height, color, text-align, and background-color. The "Console" panel shows several error messages, including "Failed to load resource: net::ERR\_BLOCKED\_BY\_CLIENT" and "DevTools failed to load source map: Could not load content for chrome-extension://gijghmmeiojklfjeocnamkbbjldom/brows-er-polyfill.js.map: System error: net::ERR\_FILE\_NOT\_FOUND".

# DEVELOPER TOOLS

- Modern web browsers include a powerful suite of developer tools
- With these tools you can
  - Inspect currently-loaded HTML and CSS
  - Debug JavaScript
  - Inspect the browser's network traffic
  - Etc.
- Open developer tools:
  - Context menu (right click) / Inspect element
  - Ctrl + shift + i or F12
  - MacOS:  $\mathcal{H}$  +  $\text{⌘}$  + *i*