<WA/>
2025

Browser Technologies

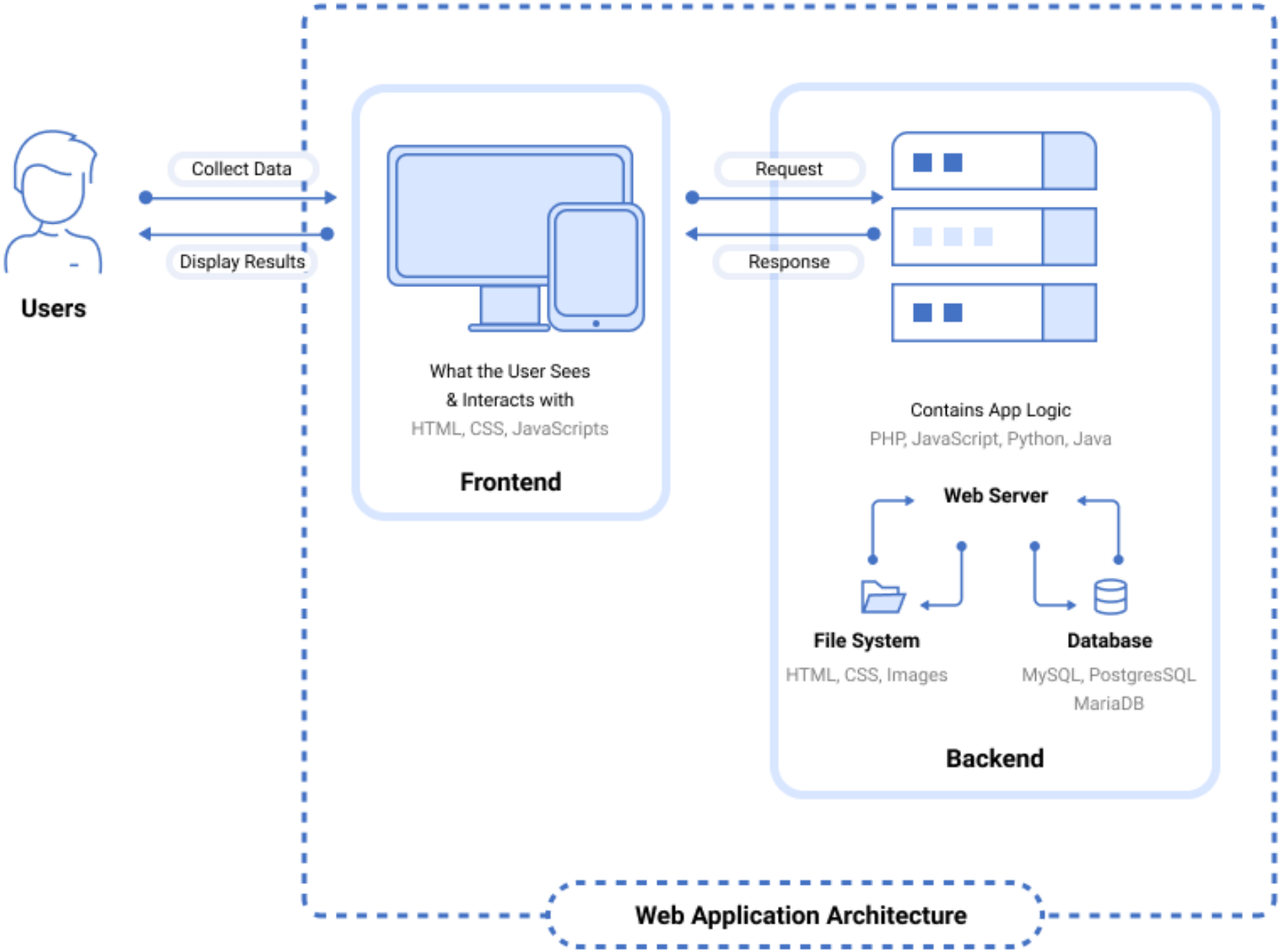
Layers and Languages

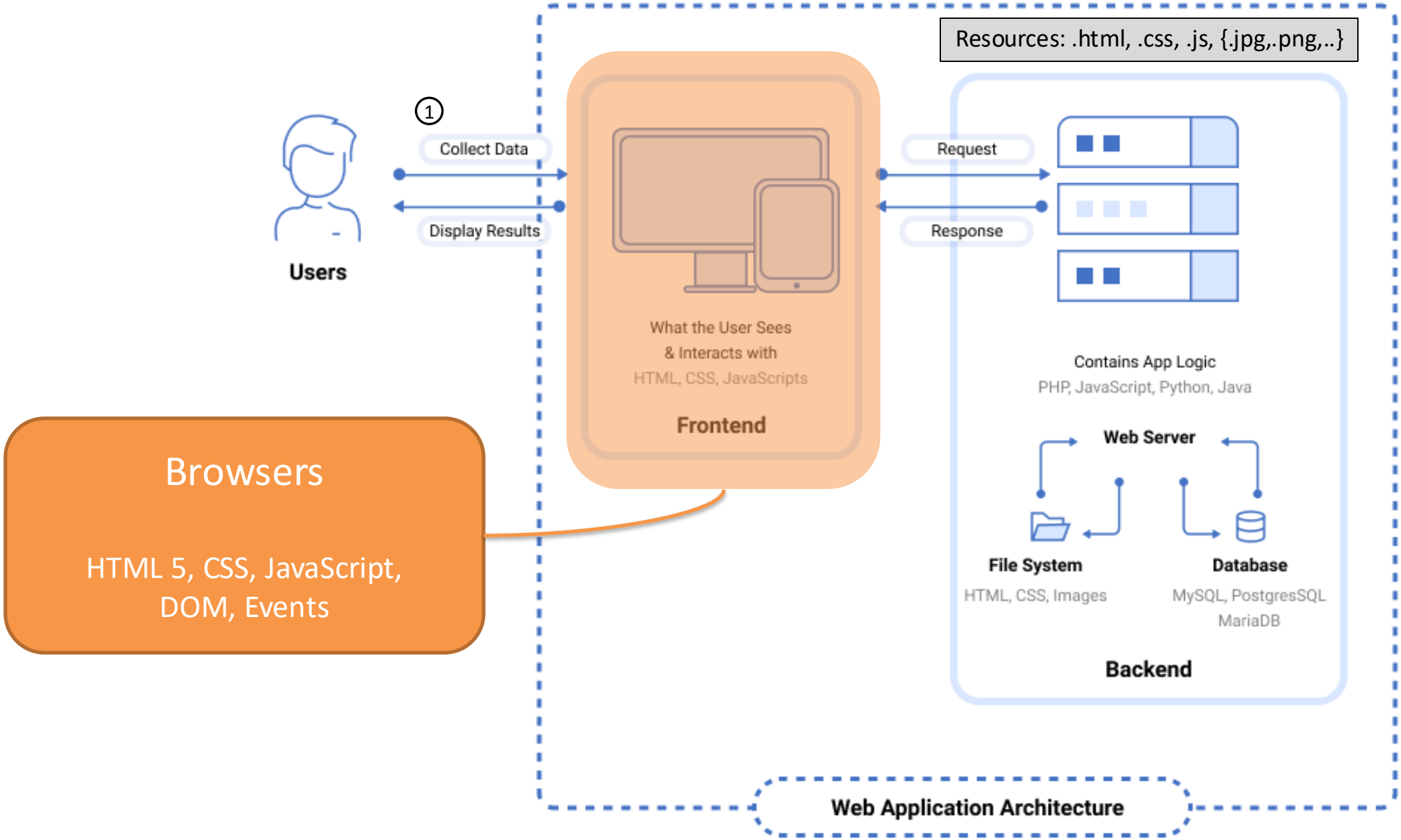
Fulvio Corno
Luigi De Russis
Enrico Masala

Goal

- Learn the basics of how a browser works
- Know the interaction and communication across components

- *NOTE: All the topics mentioned here will be presented in more details in the next lectures*

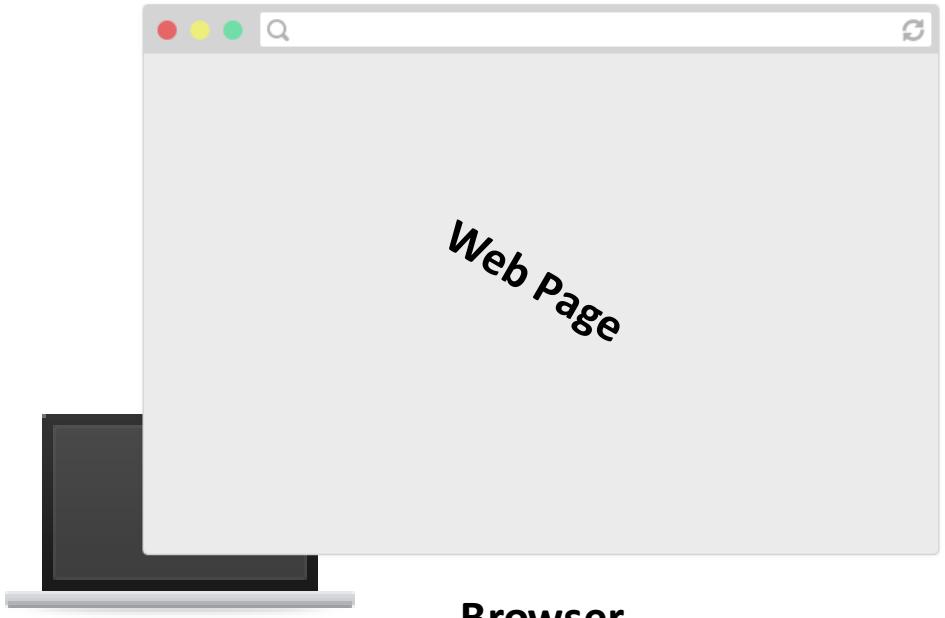

Users
Collect Data
Display Results
What the User Sees
& Interacts with
HTML, CSS, JavaScripts
Frontend
Request
Response
Contains App Logic
PHP, JavaScript, Python, Java
Web Server
File System
HTML, CSS, Images
Database
MySQL, PostgresSQL
MariaDB
Backend
Web Application Architecture



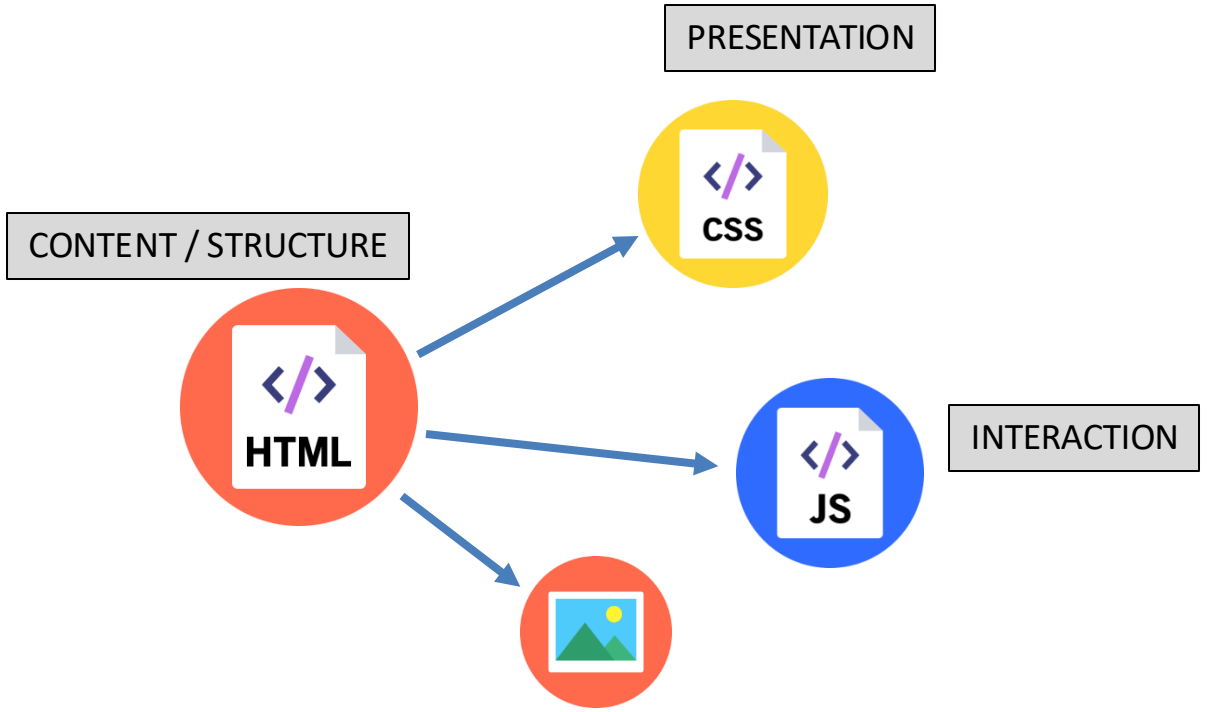
Browsers

HTML 5, CSS, JavaScript,
DOM, Events

Browser



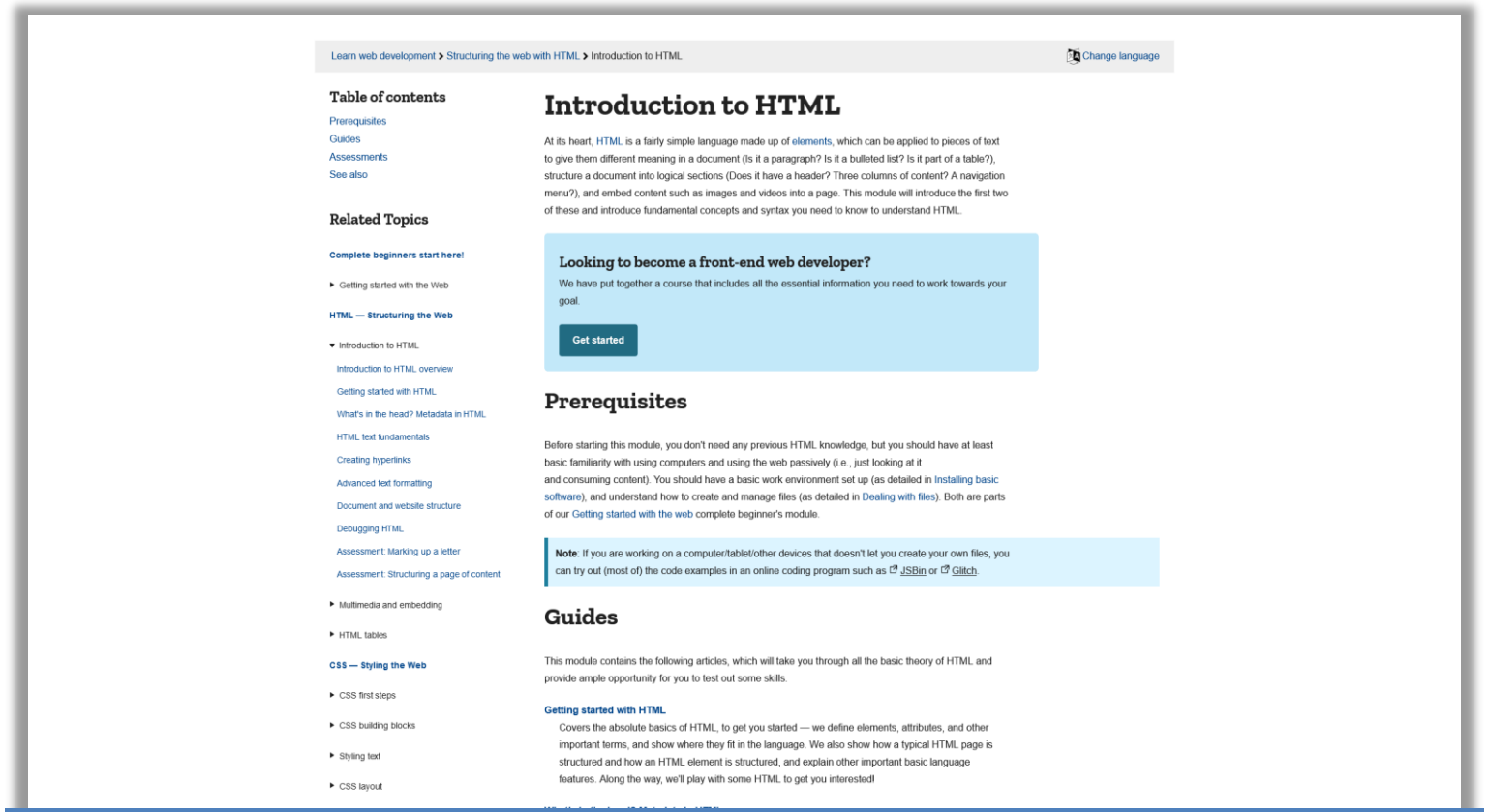
Browser



The HTML file might link to other **resources** (images, videos, …)
as well as **JavaScript** and **CSS** files,
which the browser then also loads

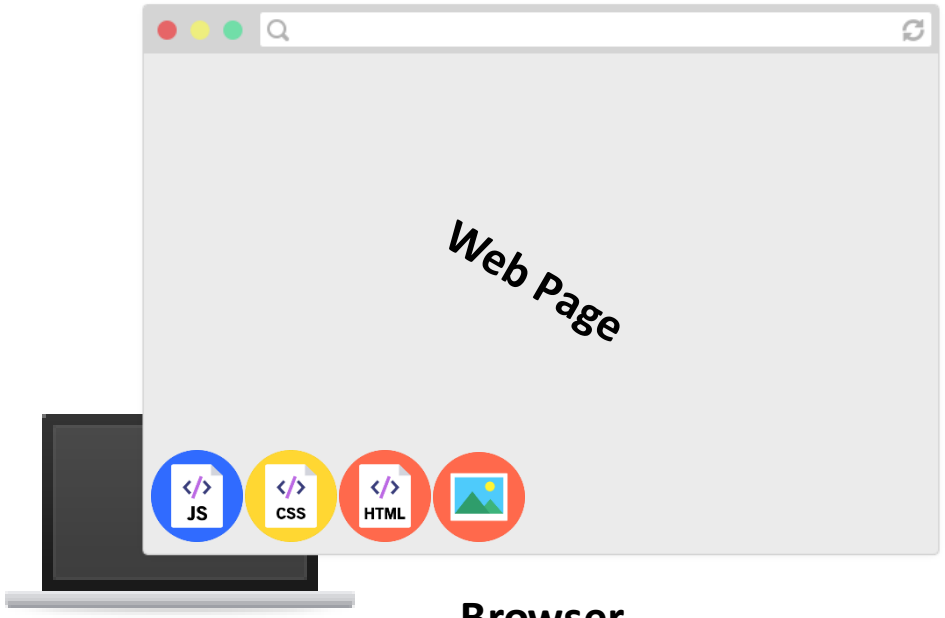
These are stored or generated by a **server**

Quick Introduction to HTML

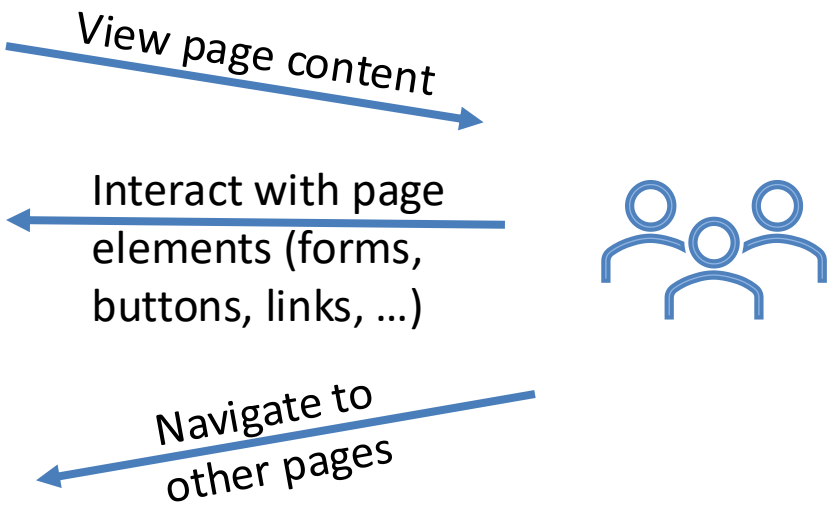


https://developer.mozilla.org/docs/Learn/HTML/Introduction_to_HTML

Browser



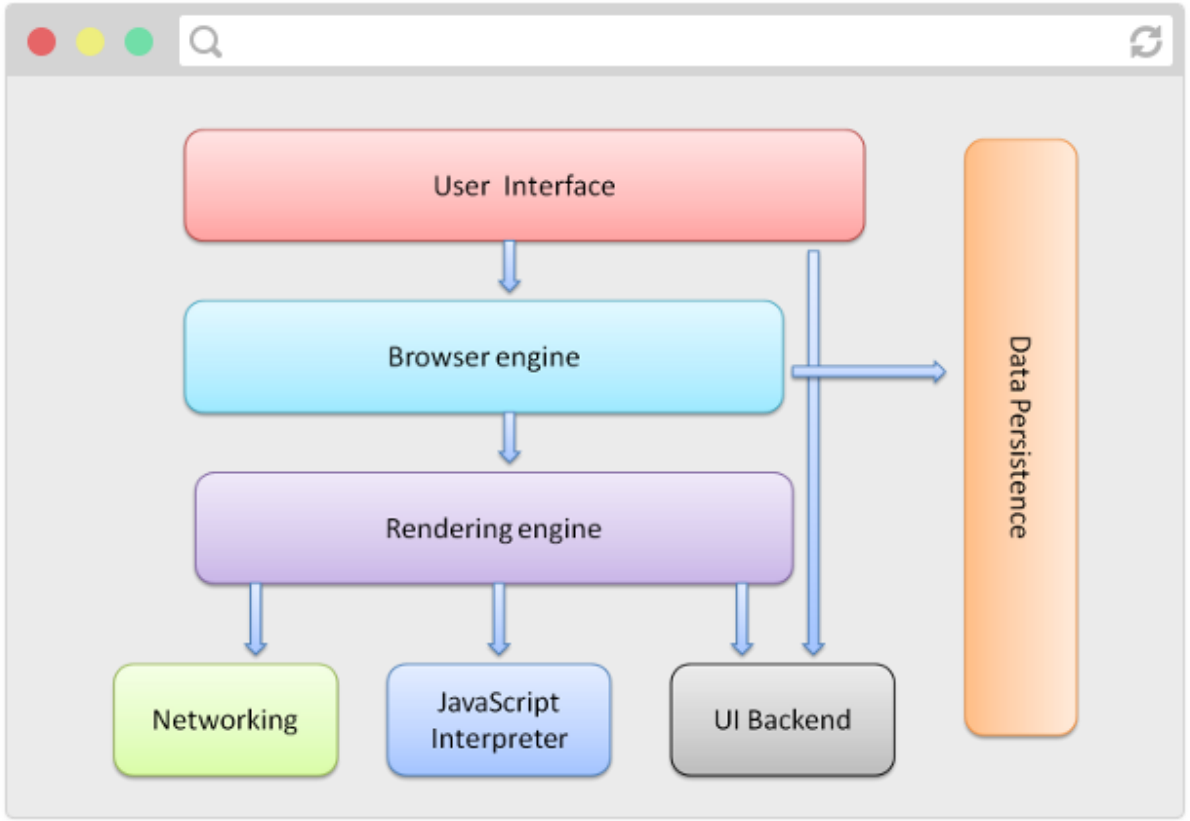
Browser



The content of the web page is described by HTML+CSS.

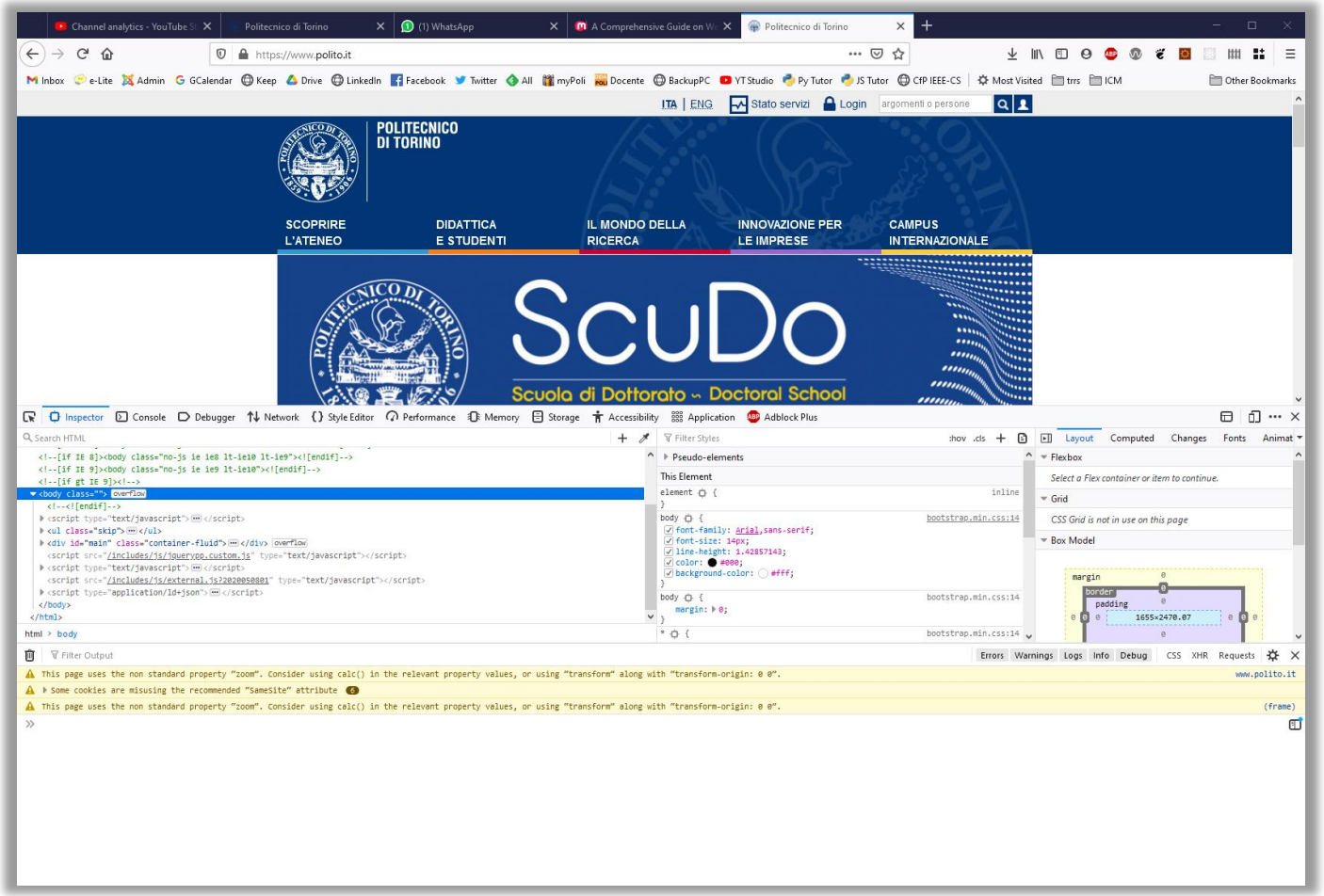
Clicking on a link brings the user to a **new page**.
Interacting with other elements may generate ***Events*** inside the browser.
Such Events are “captured” by JavaScript and may **update the page content**.

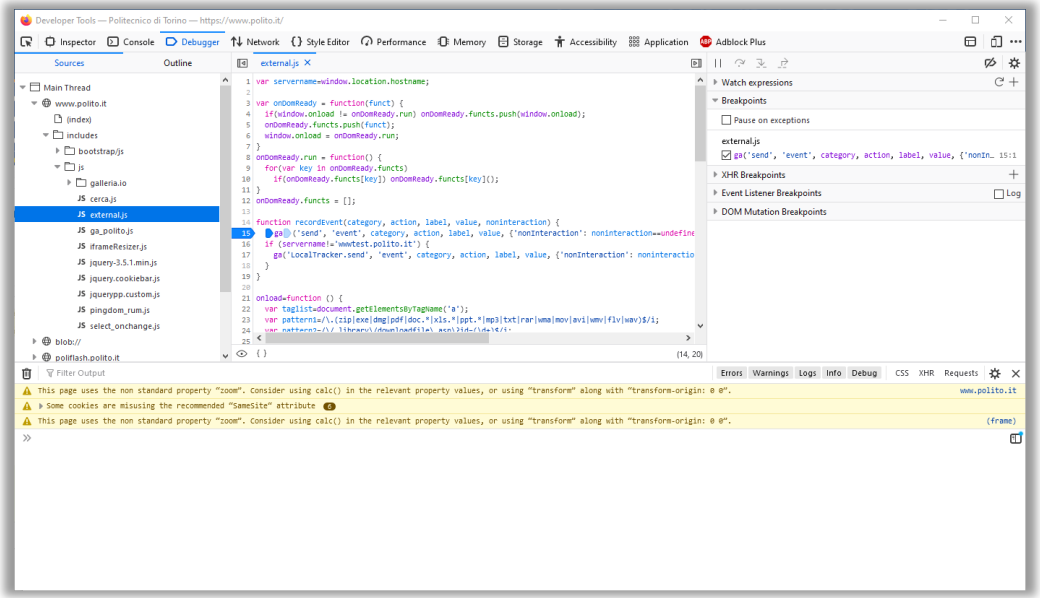
Conceptual Browser Architecture (from 10,000 feet)

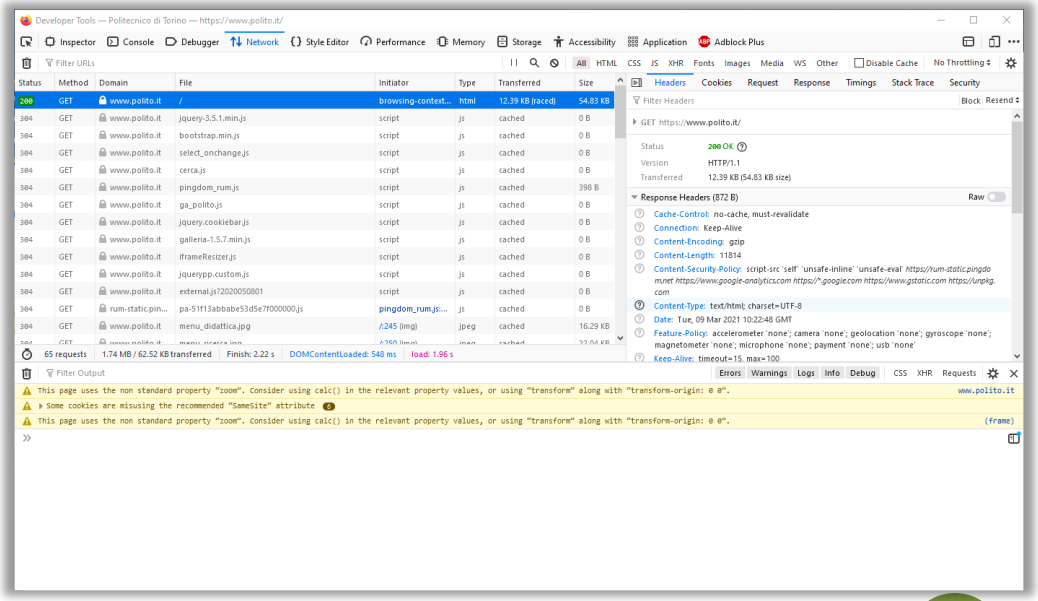


- **User Interface**: the address bar, back/forward button, bookmarking menu, etc. Every part of the browser display except the window where you see the requested page
- The **Browser Engine** marshals actions between the UI and the rendering engine
- **Rendering Engine:** responsible for displaying the requested content. For example, if the requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen
- **Networking**: for network calls such as HTTP requests, using different implementations for different platform behind a platform-independent interface
- **UI Backend**: used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. Underneath it uses operating system user interface methods
- **JavaScript Interpreter:** used to parse and execute JavaScript code
- **Data Persistence:** a persistence layer. The browser may need to save all sorts of data locally, such as cookies. Browsers also support storage mechanisms such as LocalStorage, IndexedDB, WebSQL and FileSystem

Browser Development tools



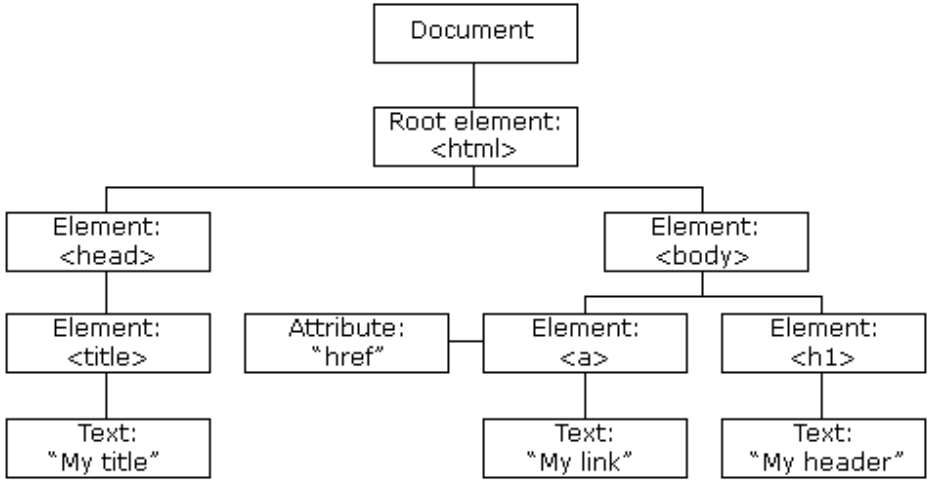




Document Object Model (DOM)

"The W3C **Document Object Model** (DOM) is a *platform and language-neutral interface* that allows programs and scripts to dynamically *access* and *update* the content, structure, and style of a document."

- Standard **data structure** for representing the web page content
- Allows to get, change, add, or delete HTML elements
- Supported by all browsers
- **JavaScript programs can read and modify the DOM**
- Abstracts and standardizes APIs to
  - Browser
  - HTML



TAG vs ELEMENT
HTMLElement interface (API)

$0

Cascading Style Sheets (CSS)

- Allow the definition of complex layouts and ways of presenting info
- Adapt web pages to
  - different resolutions
  - different devices (e.g., smartphones)
  - different preferences (e.g., color schemes)
  - to different media (e.g., text vs. video)
  - in a standard way
- Works with a set of declarations about the rules to apply

JavaScript

- JS Interpreter Embedded in the Browser
  - Executes within a strict “sandbox”
- JS Scripts are loaded by the HTML page
  - `<script src="/js/myscript.js" type="text/javascript"></script>`
- JS Scripts have read-write access to
  - Browser API
  - HTML DOM (including style and form data)
  - User events and actions

References

- How Browsers Work: Behind the scenes of modern web browsers - https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/
- Inside look at modern web browser
  - Part 1: https://developers.google.com/web/updates/2018/09/inside-browser-part1
  - Part 2: https://developers.google.com/web/updates/2018/09/inside-browser-part2
  - Part 3: https://developers.google.com/web/updates/2018/09/inside-browser-part3
  - Part 4: https://developers.google.com/web/updates/2018/09/inside-browser-part4

License

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"
- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- https://creativecommons.org/licenses/by-nc-sa/4.0/