<WA/>
2024

Browser Technologies

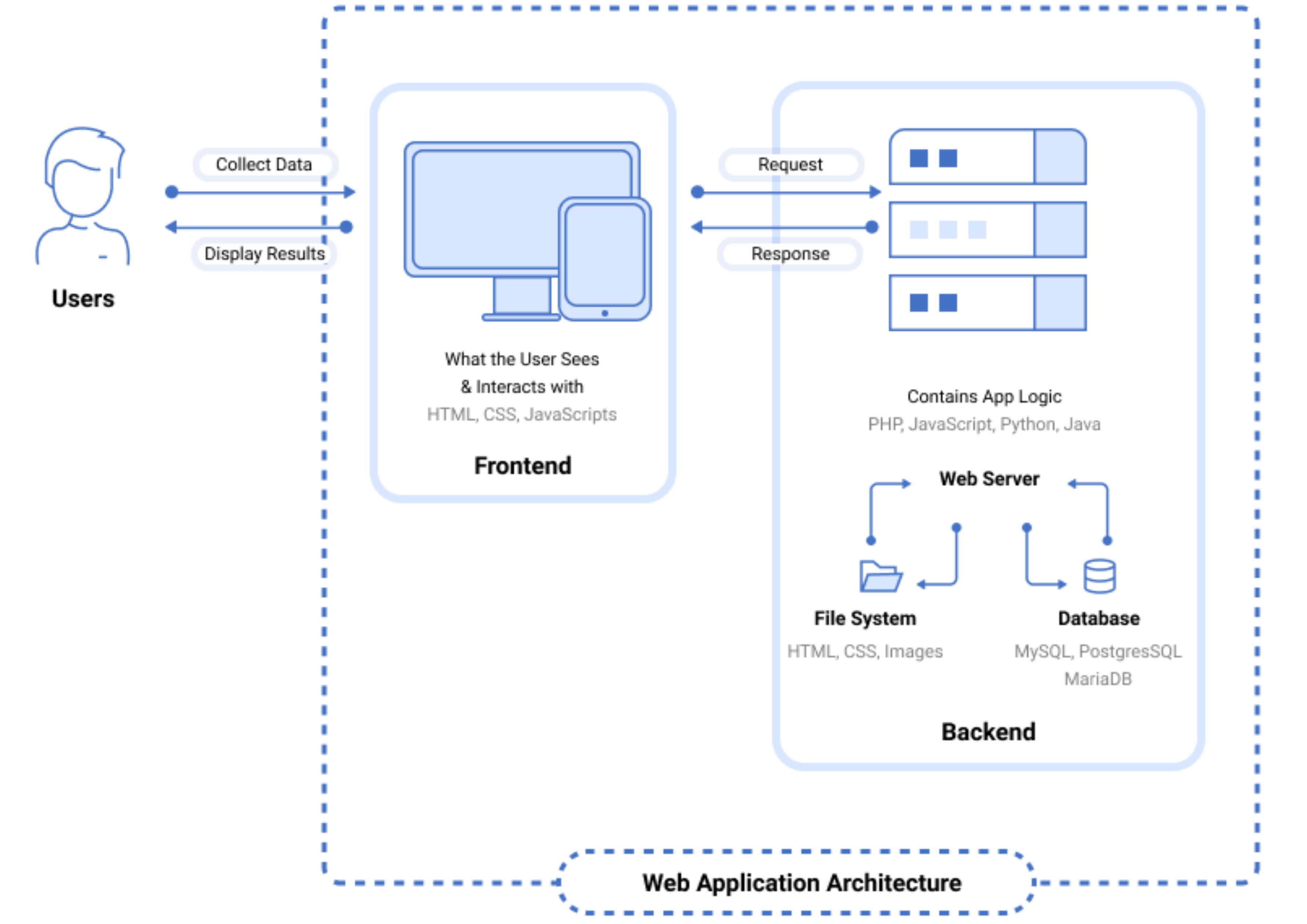
Layers and Languages

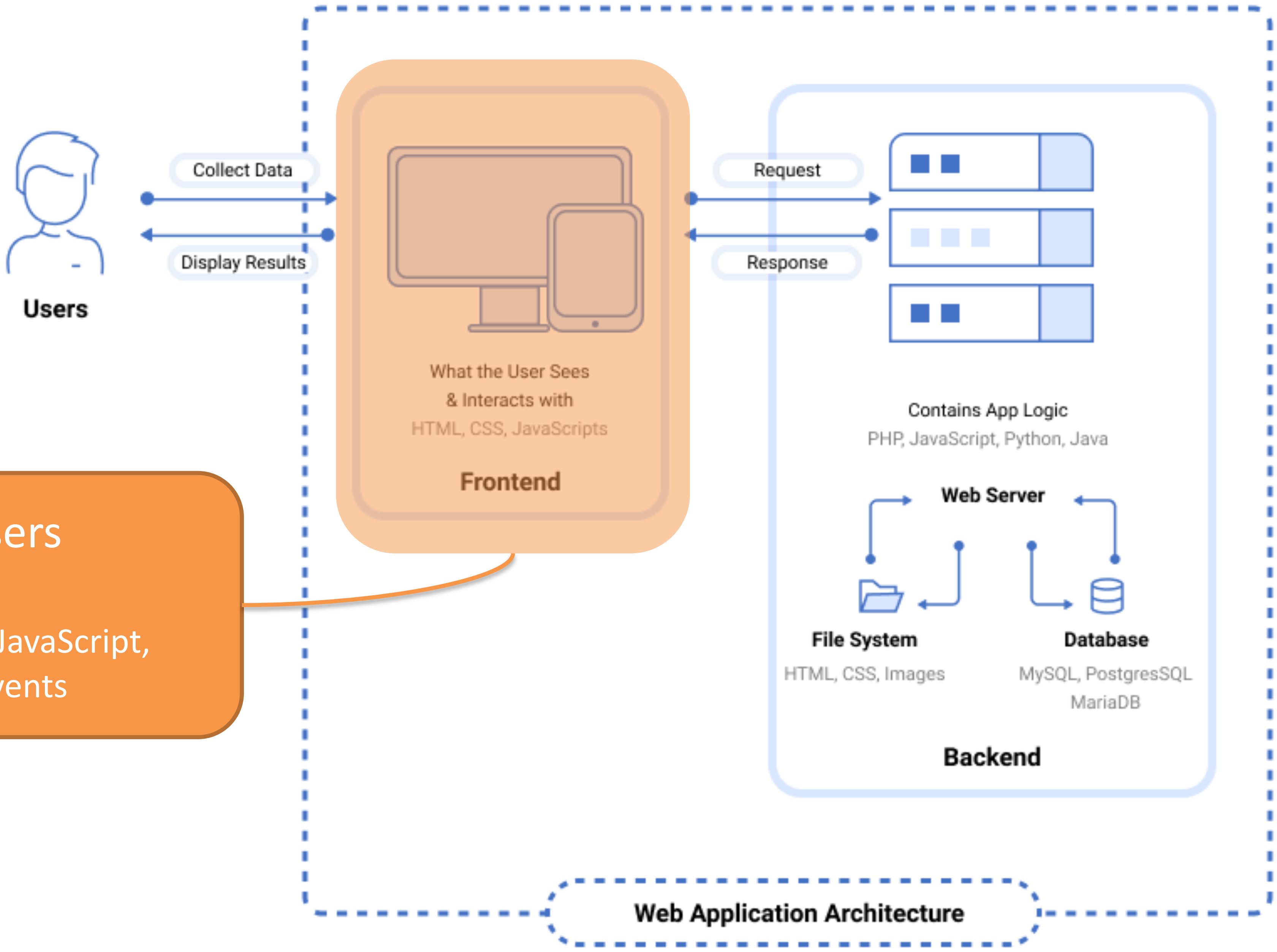
Fulvio Corno
Luigi De Russis
Enrico Masala

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*

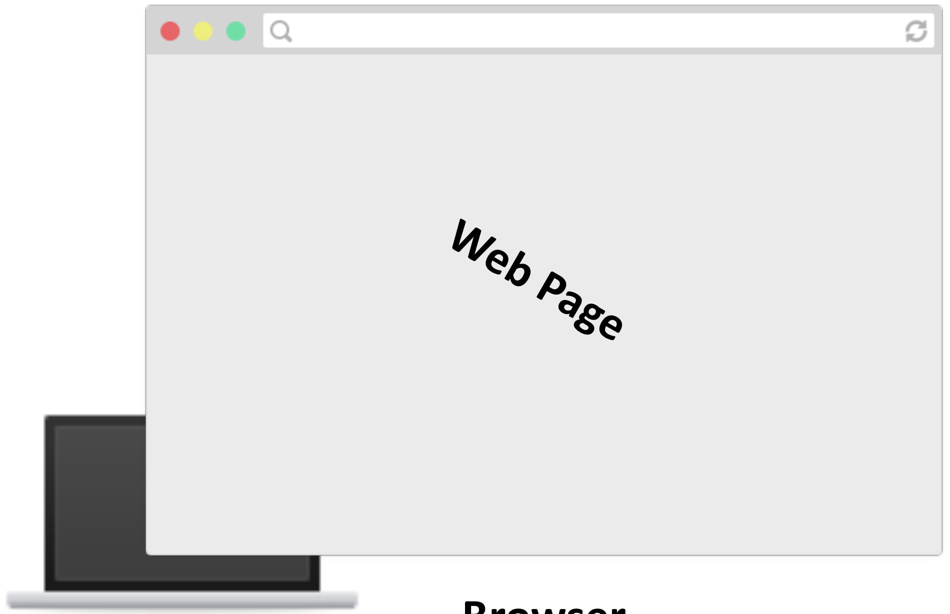

Users
Collect Data
Display Results
What the User Sees
& Interacts with
HTML, CSS, JavaScripts
Frontend
Request
Response
Contains App Logic
PHP, JavaScript, Python, Java
Web Server
File System
HTML, CSS, Images
Database
MySQL, PostgresSQL
MariaDB
Backend
Web Application Architecture



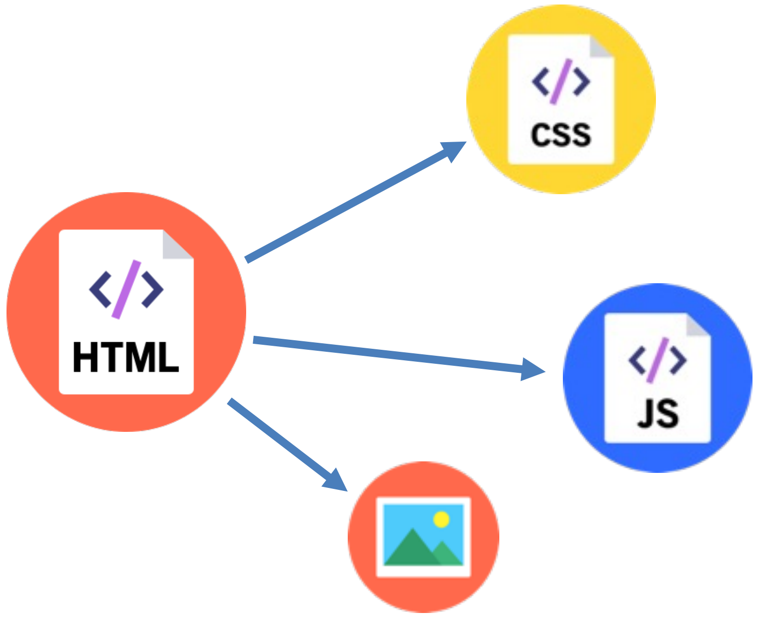
Browsers

HTML 5, CSS, JavaScript, DOM, Events

Browser



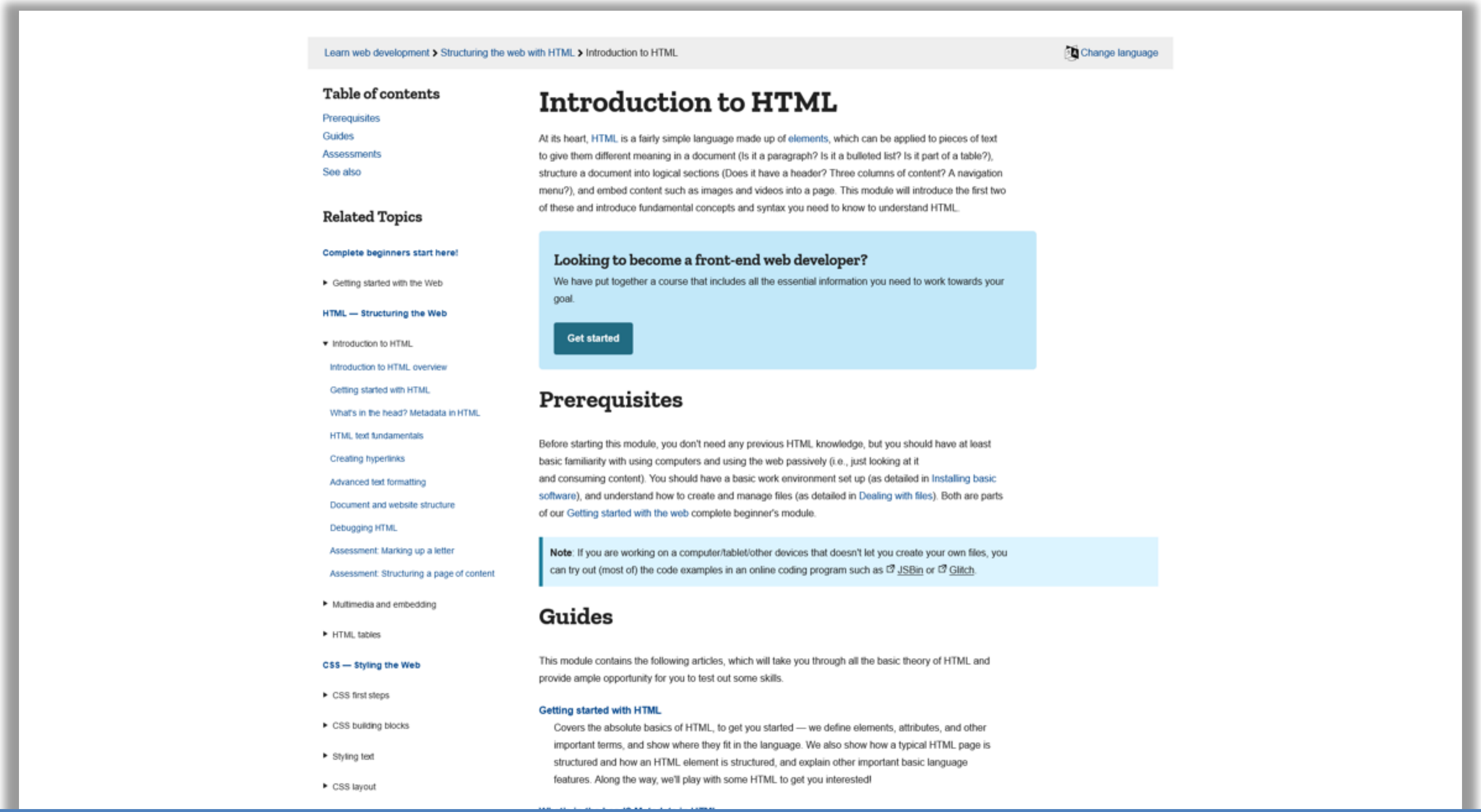
Browser



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



Learn web development > Structuring the web with HTML > Introduction to HTML

Change language

Table of contents

Related Topics

Complete beginners start here!

- Getting started with the Web

HTML — Structuring the Web

- Introduction to HTML
  - Introduction to HTML overview
  - Getting started with HTML
  - What's in the head? Metadata in HTML
  - HTML text fundamentals
  - Creating hyperlinks
  - Advanced text formatting
  - Document and website structure
  - Debugging HTML
  - Assessment: Marking up a letter
  - Assessment: Structuring a page of content
- Multimedia and embedding
- HTML tables

CSS — Styling the Web

- CSS first steps
- CSS building blocks
- Styling text
- CSS layout

Introduction to HTML

At its heart, HTML is a fairly simple language made up of elements, which can be applied to pieces of text to give them different meaning in a document (Is it a paragraph? Is it a bulleted list? Is it part of a table?), structure a document into logical sections (Does it have a header? Three columns of content? A navigation menu?), and embed content such as images and videos into a page. This module will introduce the first two of these and introduce fundamental concepts and syntax you need to know to understand HTML.

Looking to become a front-end web developer?

We have put together a course that includes all the essential information you need to work towards your goal.

Get started

Prerequisites

Before starting this module, you don't need any previous HTML knowledge, but you should have at least basic familiarity with using computers and using the web passively (i.e., just looking at it and consuming content). You should have a basic work environment set up (as detailed in Installing basic software), and understand how to create and manage files (as detailed in Dealing with files). Both are parts of our Getting started with the web complete beginner's module.

Note: If you are working on a computer/tablet/other devices that doesn't let you create your own files, you can try out (most of) the code examples in an online coding program such as JSBin or Glitch.

Guides

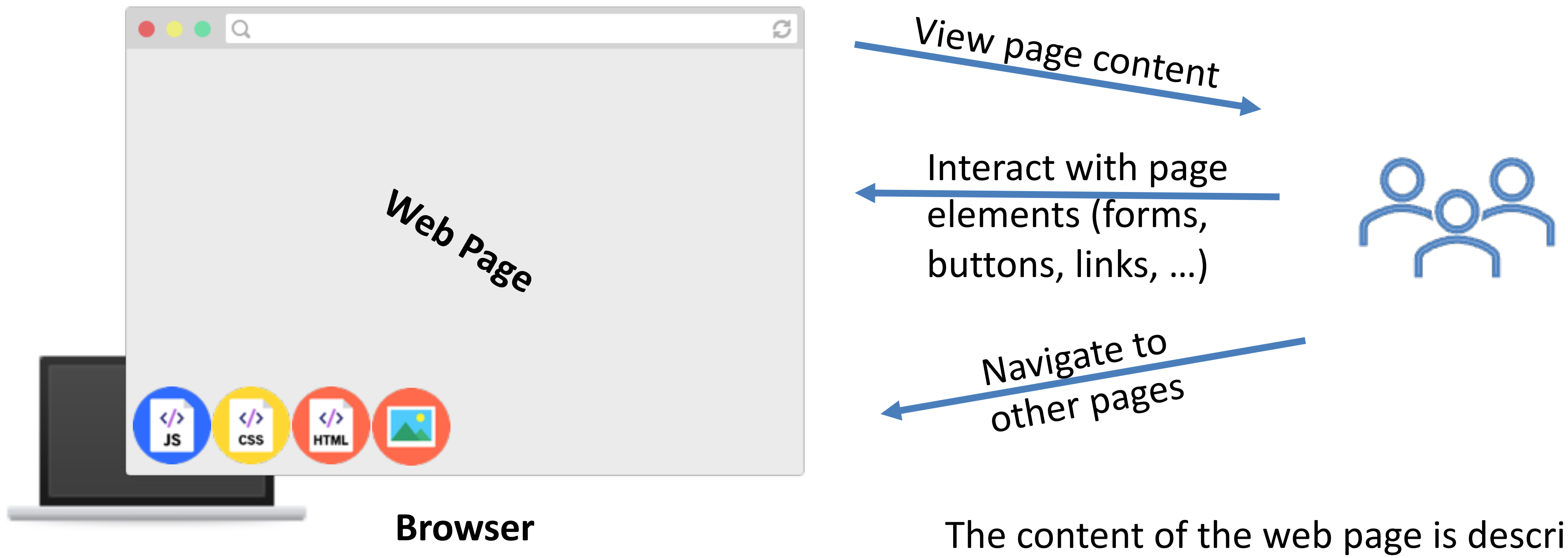
This module contains the following articles, which will take you through all the basic theory of HTML and provide ample opportunity for you to test out some skills.

Getting started with HTML

Covers the absolute basics of HTML, to get you started — we define elements, attributes, and other important terms, and show where they fit in the language. We also show how a typical HTML page is structured and how an HTML element is structured, and explain other important basic language features. Along the way, we'll play with some HTML to get you interested!

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

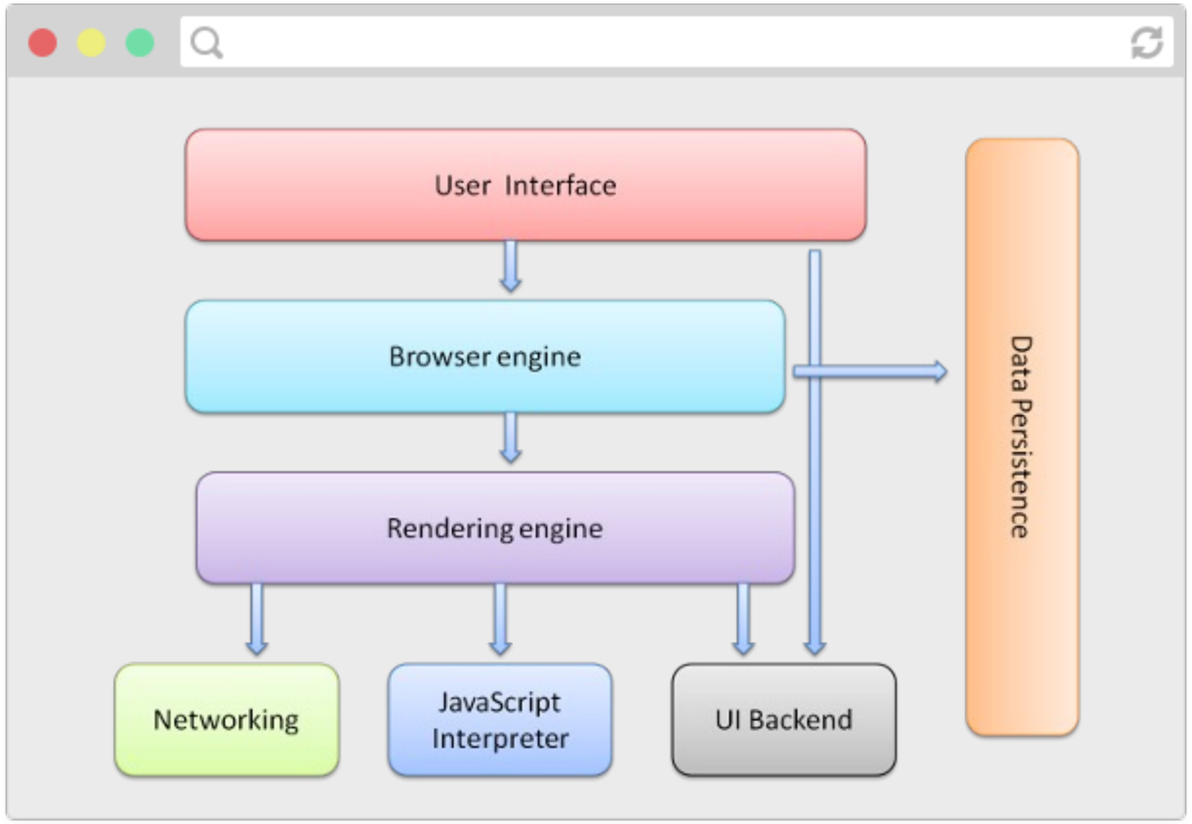
Browser



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

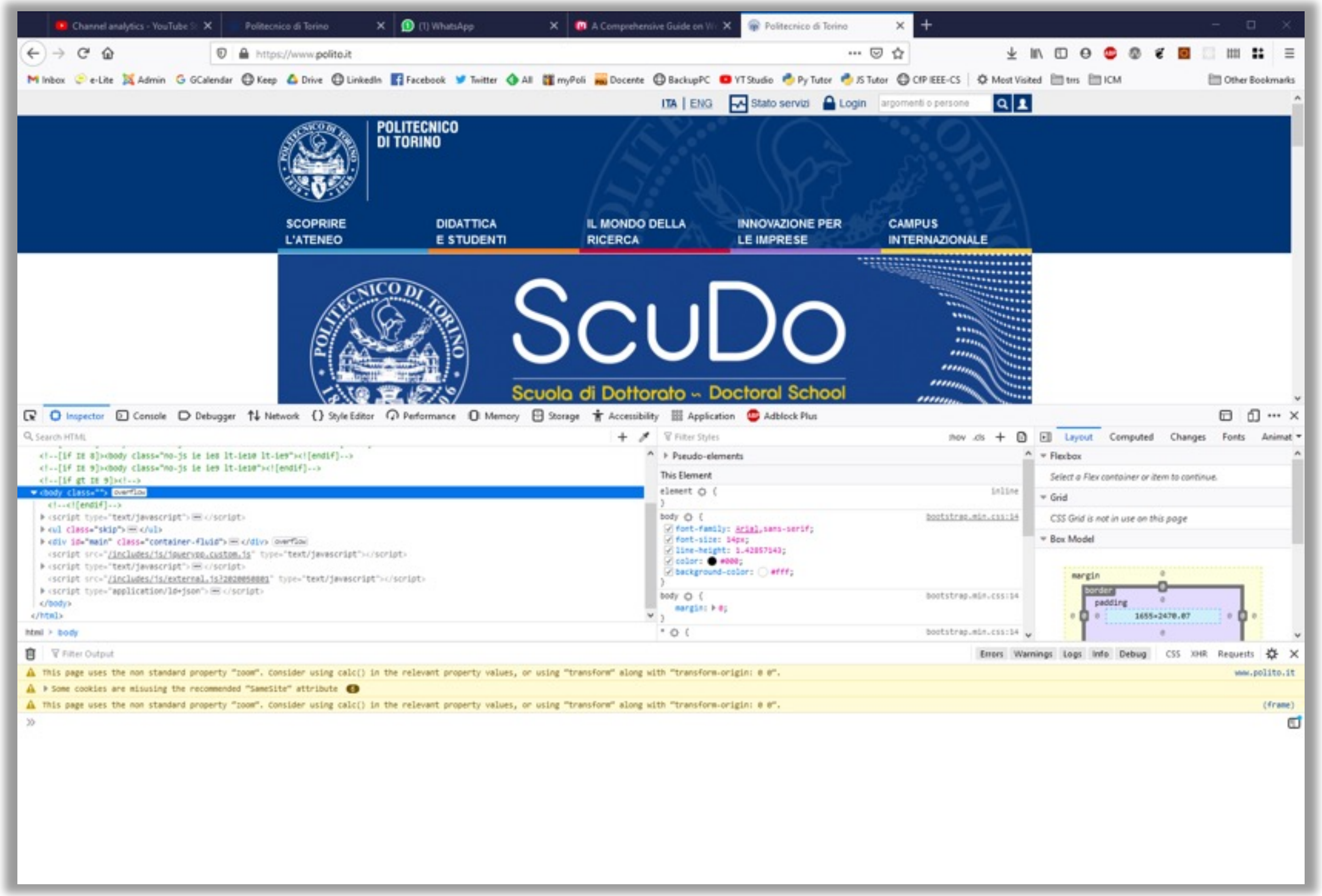
Clicking on a link brings the user to a **new page**.
Interacting with other elements may generate ***Events*** inside the browser.
Such Events are “captured” by JavaScript and may **update the page content**.

Conceptual Browser Architecture (from 10,000 feet)



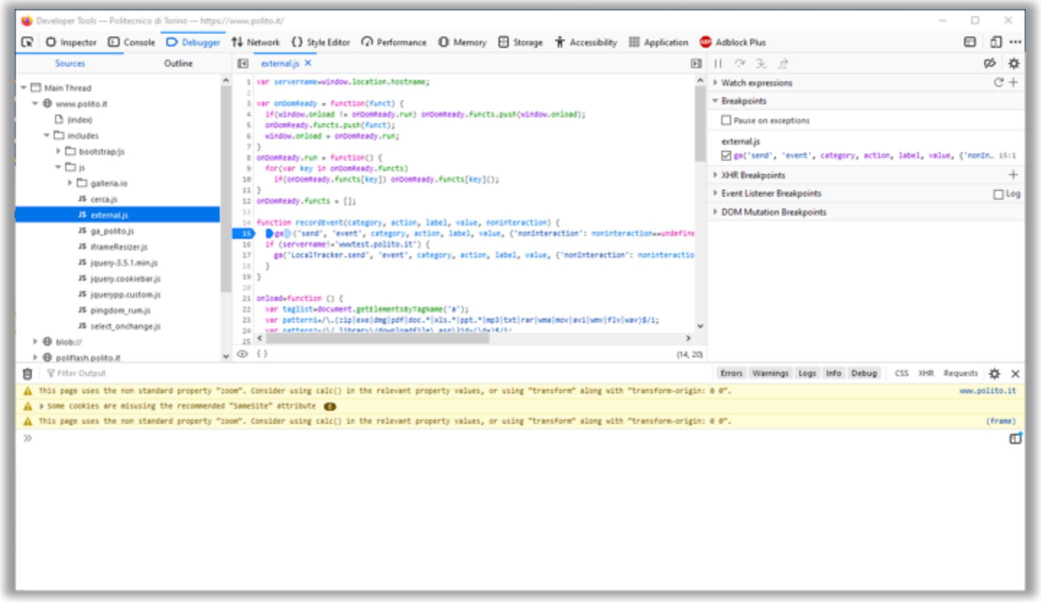
- **User Interface**: the address bar, back/forward button, bookmarking menu, etc. Every part of the browser display except the window where you see the requested page
- The **Browser Engine** marshals actions between the UI and the rendering engine
- **Rendering Engine:** responsible for displaying 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



```
<!--[if IE 8]><body class="no-js ie ie8 lt-ie10 lt-ie9"><![endif]-->
<!--[if IE 9]><body class="no-js ie ie9 lt-ie10"><![endif]-->
<!--[if gt IE 9]><!-->
<body class=""> overflow
  <!--<![endif]-->
  <script type="text/javascript"> … </script>
  <ul class="skip"> … </ul>
  <div id="main" class="container-fluid"> … </div> overflow
  <script src="/includes/js/jquerypp.custom.js" type="text/javascript"></script>
  <script type="text/javascript"> … </script>
  <script src="/includes/js/external.js?2020050801" type="text/javascript"></script>
  <script type="application/ld+json"> … </script>
</body>
</html>
```

```
This page uses the non standard property "zoom". Consider using calc() in the relevant property values, or using "transform" along with "transform-origin: 0 0".   www.polito.it
▸ Some cookies are misusing the recommended "SameSite" attribute
This page uses the non standard property "zoom". Consider using calc() in the relevant property values, or using "transform" along with "transform-origin: 0 0".   (frame)
```

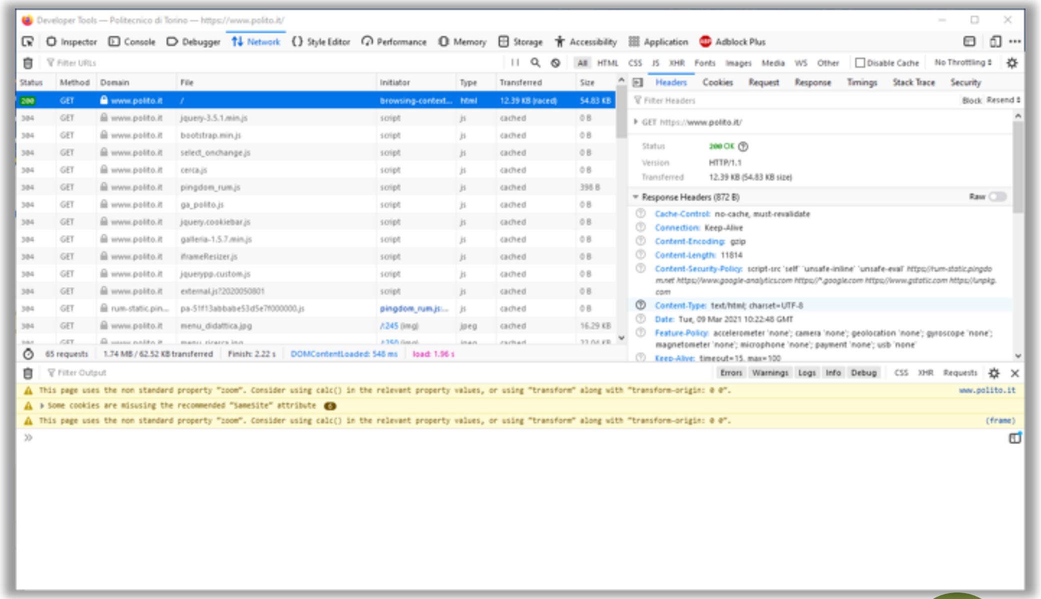


```
var servername=window.location.hostname;

var onDomReady = function(funct) {
  if(window.onload != onDomReady.run) onDomReady.functs.push(window.onload);
  onDomReady.functs.push(funct);
  window.onload = onDomReady.run;
}
onDomReady.run = function() {
  for(var key in onDomReady.functs)
    if(onDomReady.functs[key]) onDomReady.functs[key]();
}
onDomReady.functs = [];

function recordEvent(category, action, label, value, noninteraction) {
  ga('send', 'event', category, action, label, value, {'nonInteraction': noninteraction==undefine
  if (servername!='wwwtest.polito.it') {
    ga('LocalTracker.send', 'event', category, action, label, value, {'nonInteraction': noninteractio
  }
}

onload=function () {
  var taglist=document.getElementsByTagName('a');
  var pattern1=/\.(zip|exe|dmg|pdf|doc.*|xls.*|ppt.*|mp3|txt|rar|wma|mov|avi|wmv|flv|wav)$/i;
```



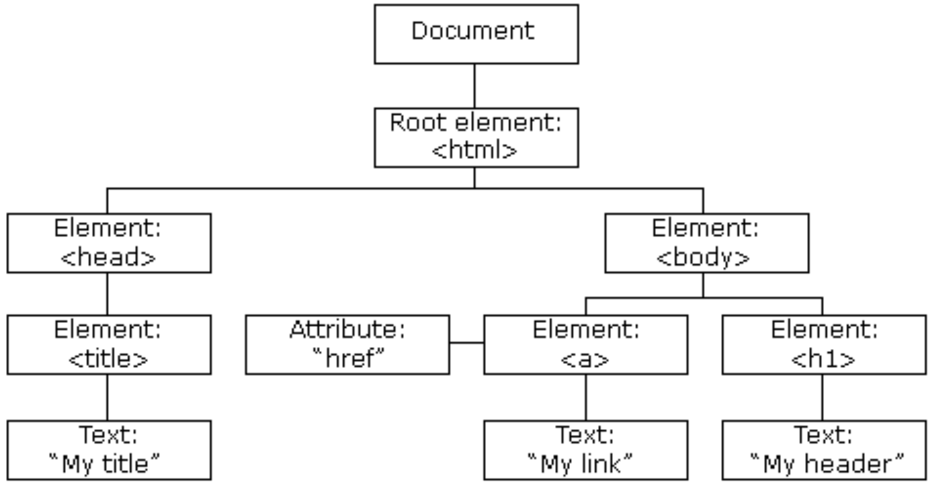
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	www.polito.it	/	browsing-context…	html	12.39 KB (raced)	54.83 KB
304	GET	www.polito.it	jquery-3.5.1.min.js	script	js	cached	0 B
304	GET	www.polito.it	bootstrap.min.js	script	js	cached	0 B
304	GET	www.polito.it	select_onchange.js	script	js	cached	0 B
304	GET	www.polito.it	cerca.js	script	js	cached	0 B
304	GET	www.polito.it	pingdom_rum.js	script	js	cached	398 B
304	GET	www.polito.it	ga_polito.js	script	js	cached	0 B
304	GET	www.polito.it	jquery.cookiebar.js	script	js	cached	0 B
304	GET	www.polito.it	galleria-1.5.7.min.js	script	js	cached	0 B
304	GET	www.polito.it	iframeResizer.js	script	js	cached	0 B
304	GET	www.polito.it	jquerypp.custom.js	script	js	cached	0 B
304	GET	www.polito.it	external.js?2020050801	script	js	cached	0 B
304	GET	rum-static.pin…	pa-51f13abbabe53d5e7f000000.js	pingdom_rum.js:…	js	cached	0 B
304	GET	www.polito.it	menu_didattica.jpg	/:245 (img)	jpeg	cached	16.29 KB

```
65 requests   1.74 MB / 62.52 KB transferred   Finish: 2.22 s   DOMContentLoaded: 548 ms   load: 1.96 s
```

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

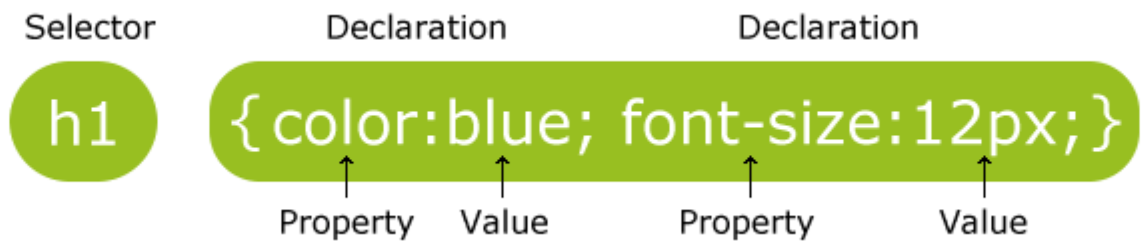


Cascading Style Sheets (CSS)

- Allow the definition of complex layouts
- 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

Cascading Style Sheets (CSS)

- A set of "*declarations*" applied to some "*selectors*"
  - Selectors identify portions of the DOM
  - Declarations set the value of some properties
  - Properties control everything
    - color, size, font, alignment, border, shadow, position, selection status, transitions, links, buttons, cursors, …



JavaScript

- JS Interpreter Embedded in the Browser
  - Executes within a strict “sandbox”
- JS Scripts 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 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