



# **Seminar Paper of the Lecture Business Information Management**

-

## **“Web Browser: History, Concepts, Market”**

Feldmann, Tim (h1552931)

LV-Number: 4167

LV-Leiterin: ao.Univ.Prof. Dr. Rony G. Flatscher

Term: Summer 2020

Vienna, 25th March 2020

## **Declaration of Independence**

I hereby declare that:

1. I have written this seminar paper myself, independently and without the aid of unfair or unauthorized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced.
2. This seminar paper has not been previously presented as an examination paper in this or any other form in Austria or abroad.
3. This seminar paper is identical with the paper assessed by the examiner.

Date

Signature

# Abstract

Nowadays, web browsers are taken for granted - they are used by the broad public and are often only mentioned in connection with the internet. However, it is the browser that enables users to display the contents of the internet. In this seminar paper, the topic "Web Browser - History, Concepts, and Market" is discussed in detail and should help to provide an overview of this particular topic. In the theoretical part, I will revisit the history of web browsers, which started with the inventions of Tim Berners-Lee and then analyze the browser wars, which continue until the present day. Next, the concept of browsers, their functionality, and most important standards such as HTML5 and CSS3 are being reviewed and illustrated via practical examples. After discussing the primary organizations for common web standards, the fierce competition within the browser market is analyzed in more detail. Last but not least, to have a better understanding of the competition between market leaders, three benchmark tests were performed. The results of these tests were compared, and a conclusion was drawn.

## **Table of Contents**

<b>1. Introduction .....</b>	<b>- 6 -</b>
<b>2. History .....</b>	<b>- 7 -</b>
<b>2.1 Short Biography of Tim Berners-Lee.....</b>	<b>- 7 -</b>
<b>2.2 The First Steps.....</b>	<b>- 8 -</b>
<b>2.3 Milestones .....</b>	<b>- 10 -</b>
<b>2.4 Browser Wars.....</b>	<b>- 14 -</b>
<b>3. Concepts .....</b>	<b>- 16 -</b>
<b>3.1 Functionality .....</b>	<b>- 16 -</b>
<b>3.2 Standards .....</b>	<b>- 19 -</b>
<b>3.2.1 HTML5 Standards.....</b>	<b>- 20 -</b>
<b>3.2.2 CSS 3 Standards .....</b>	<b>- 21 -</b>
<b>3.2.3 JavaScript Standards .....</b>	<b>- 22 -</b>
<b>3.2.4 Best Practices .....</b>	<b>- 25 -</b>
<b>4. Market.....</b>	<b>- 27 -</b>
<b>4.1 General Overview .....</b>	<b>- 27 -</b>
<b>4.2 Benchmarking of the Different Browsers.....</b>	<b>- 28 -</b>
<b>4.2.1 ARES-6 .....</b>	<b>- 29 -</b>
<b>4.2.2 Basemark Web 3.0 .....</b>	<b>- 30 -</b>
<b>4.2.3 HTML5test.....</b>	<b>- 31 -</b>
<b>4.3 Summary .....</b>	<b>- 31 -</b>
<b>4.3.1 Brave .....</b>	<b>- 32 -</b>
<b>4.3.2 Google Chrome .....</b>	<b>- 32 -</b>
<b>4.3.3 Microsoft Edge .....</b>	<b>- 32 -</b>
<b>4.3.4 Mozilla Firefox .....</b>	<b>- 33 -</b>
<b>4.3.5 Safari .....</b>	<b>- 33 -</b>
<b>5. Conclusion.....</b>	<b>- 34 -</b>
<b>6. Bibliography .....</b>	<b>- 36 -</b>

## **List of Figures**

FIGURE 1 - TIM BERNERS-LEE (“THE BIRTH OF THE WEB   CERN,” N.D.) .....	- 7 -
FIGURE 2 - PROPOSAL OF THE WWW (“THE ORIGINAL PROPOSAL OF THE WWW, HTMLIZED”, N.D.).....	- 9 -
FIGURE 3 - FIRST WEB BROWSER (“CERN 2019 WORLDWIDEWEB REBUILD”, N.D.) ...	- 10 -
FIGURE 4- LINE MODE BROWSER (“LINE MODE BROWSER 2013”, N.D.).....	- 11 -
FIGURE 5 - MOSAIC BROWSER (GOBLE, 2019) .....	- 12 -
FIGURE 6 - INTERNET EXPLORER (GOBLE, 2019).....	- 13 -
FIGURE 7 - USAGE SHARE WEB BROWSERS (WIKIPEDIA CONTRIBUTORS, 2020) .....	- 14 -
FIGURE 8 - HTML5, CSS AND JAVAScript (“WHAT IS THE ROLE OF HTML, CSS AND JS IN FRONT-END DEVELOPMENT? - QUORA,” N.D.).....	- 16 -
FIGURE 9 - BROWSER COMPONENTS (GARSIEL & IRISH, 2011).....	- 17 -
FIGURE 10 - RENDERING MODULE (GARSIEL & IRISH, 2011).....	- 18 -
FIGURE 11 - MAIN FLOW – RENDERING ENGINES (GARSIEL & IRISH, 2011) .....	- 18 -
FIGURE 12 - HTML5 SEMANTIC TAGS (“HTML5 SEMANTIC TAGS   VIKING CODE SCHOOL,” N.D.).....	- 21 -
FIGURE 13 - EXAMPLE CSS ANIMATED BUTTOM (“TRYIT EDITOR V3.6,” N.D.) .....	- 22 -
FIGURE 14 - HTML5 (THE HTML 5 JAVAScript API INDEX, N.D.).....	- 22 -
FIGURE 15 - : EXAMPLE ASSIGN EVENTS USING THE HTML DOM (“TRYIT EDITOR V3.6,” N.D.).....	- 23 -
FIGURE 16 - EXAMPLE GEOLOCATION (“TRYIT EDITOR V3.6,” N.D.) .....	- 24 -
FIGURE 17 - EXAMPLE MOVING OBJECT IN CANVAS (“TRYIT EDITOR V3.6,” N.D.) .....	- 25 -
FIGURE 18 - MARKET SHARE FEBRUARY 2020 WORLDWIDE (STATCOUNTER, 2020) ...	- 27 -
FIGURE 19 - BROWSER MARKET SHARE WORLDWIDE FROM JAN 2009 -APR 2020 (“BROWSER MARKET SHARE WORLDWIDE,” N.D.) .....	- 28 -
FIGURE 20 - GRAPHIC ARES-6 .....	- 29 -
FIGURE 21 - GRAPHIC BASEMARK WEB 3.0 .....	- 30 -
FIGURE 22 - GRAPHIC HTML5TEST .....	- 31 -

# 1. Introduction

We use web browsers every day, whether it is at work or simply at home on our computers, tablets, smartphones, or other Internet-enabled devices. These programs allow us to graphically display websites and enter the infinite depths of the world wide web. According to the Duden dictionary, the term "Browser" is defined as "a program with which websites can be found, read and managed" and is derived from the English word "to browse" ("Browser," n.d.).

In this context we often hear the terms Internet and World Wide Web. These terms are often used interchangeably in their linguistic usage, however, technically there is a considerable difference between the two. The Internet is a global network of many individual computer networks and numerous services - such as e-mail, chat, file transfer or even Internet telephony - bring this infrastructure to life. One of the best-known services is the World Wide Web (WWW), which enables the transmission of web pages. To view it, users need a browser such as Internet Explorer, Mozilla Firefox, Safari or Opera. (Online, 2009)

Since this topic accompanies us on a daily basis and browsers play an important role in our daily lives, I have chosen this topic. In what follows, I describe in a more nuanced way how everything started and give a brief overview of the history of web browsers. Moreover, I will explain how a browser actually works and what concepts are being used. Towards the end of this paper, I will compare the market and finish off with a conclusion.

The goal of this seminar paper is to provide an insight into the subject matter and equip the reader with a better understanding of how web browsers as an actual everyday tool which we take for granted actually works.

## 2. History

The aim of this chapter is to review the history of Web Browser and to highlight the most important milestones within this context.

### 2.1 Short Biography of Tim Berners-Lee

One of the most important and influential computer scientists in connection with web browsers and the World Wide Web (WWW) is Tim Berners-Lee (shown in the picture below).



*Figure 1 - Tim Berners-Lee ("The birth of the Web | CERN," n.d.)*

Berners-Lee was born on the 8<sup>th</sup> of June 1955 in London, England. The inventor of the world wide web got his passion for computers from his parents because both of them were mathematicians, who collaborated to develop one of the world's first commercial computers controlled by stored programs. Following his graduation at Emanuel School, Berners\_Lee went to Queen's College, Oxford University, where he received a first-class degree in physics (Wikipedia contributors, 2020).

In 1989, while working at CERN, the European Particle Physics Laboratory in Geneva, Switzerland, Tim Berners-Lee proposed a global hypertext project, which later came to be known as the World Wide Web. Based on the earlier work, it was designed to allow people to work together by combining their knowledge in a web of hypertext documents (Mercier, n.d.). Berners-Lee wrote the first World Wide Web server, "httpd", and the first client, "WorldWideWeb" a what-you-see-is-what-you-get hypertext

browser/editor which ran in the NeXTStep environment. Berners-Lee started on this endeavor in October 1990, and the "WorldWideWeb" was made available as a program within CERN in December, and on the Internet at large in the summer of 1991 (Mercier, n.d.).

Furthermore, in 1994 Berners-Lee founded the World Wide Web Consortium (W3C), which is a committee for the standardization of techniques in the World Wide Web.

More specifically, the W3C develops technical specifications and guidelines in a mature, transparent process in order to achieve maximum consensus on the content of technical protocols, high technical and editorial quality and approval by the W3C and its supporters ("Facts About W3C," n.d.). Currently, Berners-Lee promotes open government data globally and spends time advocating for rights such as net neutrality, privacy and the openness of the Web.

## 2.2 The First Steps

As mentioned in the previous chapter, the first browser was created in December of 1990 by Tim Berners-Lee. Berners-Lee's discovery is also the starting point for this next chapter, in which I will cover the most important milestones leading up to the development of the first browser. What ultimately triggered the development of the Internet, can arguably be traced back to the Cold War, when the US Department of Defense was concerned that its messaging system could be destroyed by their enemies. In order to be fully functional – even when one node used to fail – the defense departmental introduced a computerized network. (Wikipedia contributors, 2020).

These plans led to the foundation of the Advanced Research Project Agency – short ARPA. With its inauguration, the ARPA carried out its first tests with packet-switched networks, which led to the development of Advanced Research Project Agency Network (short ARPAnet). The ARPAnet was a decentralized computer network, where computers were set up at different locations.

In 1968, the ARPAnet network was then opened up for non-military purposes, so that universities across the the USA were also able to connect to the ARPAnet.

In 1971, Ray Tomlinson – the inventor of the e-mail, sent out the world's first e-mail via an e-mail messaging system, which operated within the APRA net. In this email Ray



In March 1989 Tim Berners-Lee presented the proposal "Information Management: A Proposal", the basic idea of the WWW.

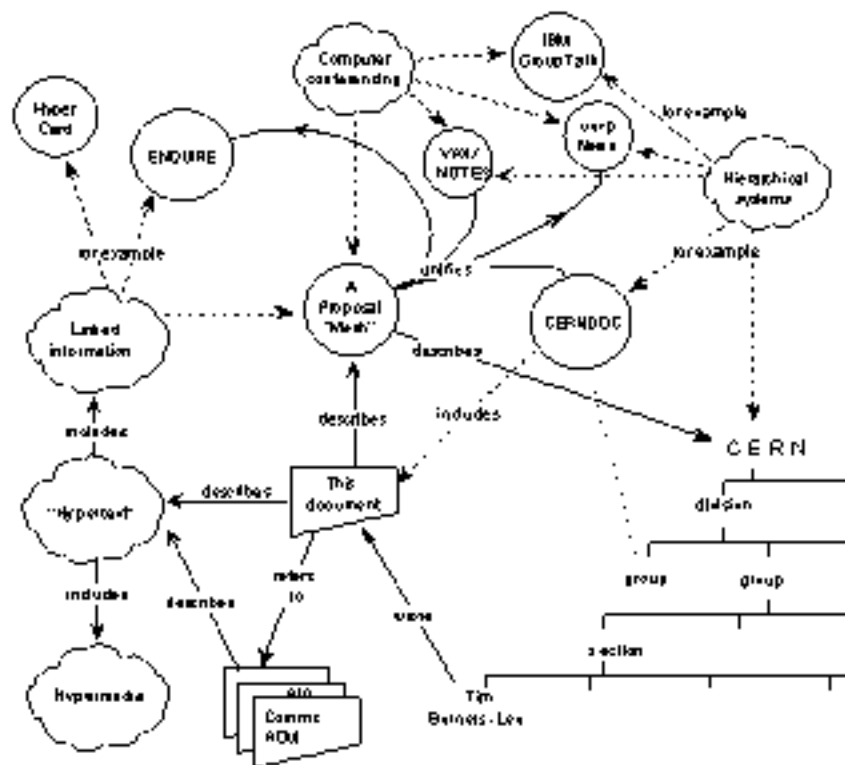


Figure 2 - Proposal of the WWW (“The original proposal of the WWW, HTMLized”, n.d.)

The main idea was to create a communication platform, which could be compatible to the already existing internet as a hyperlink system. Web documents that refer to external information sources should not simply mention the data of other websites, as it has been the case previously, but the data should be directly accessible via electronic links from the current document. Users should be able to jump from one web page to

other desired pages with a single click, regardless of the distance or geographical location of the computers. With the help of Robert Cailliau, the concept, which was based on client-server architecture, was pushed forward and presented to the public in October of 1990 as the "WorldWideWeb" (W3), also known as Nexus. Within a few weeks Berners-Lee programmed the first version of the WorldWideWeb - the first browser - on a NeXT computer (Moussaoui & Zeppenfeld, 2007, pp. 12–13). This was the beginning of the use of the internet for the public, which was made possible by these web browsers. Figure 3 below shows the first Web Browser.

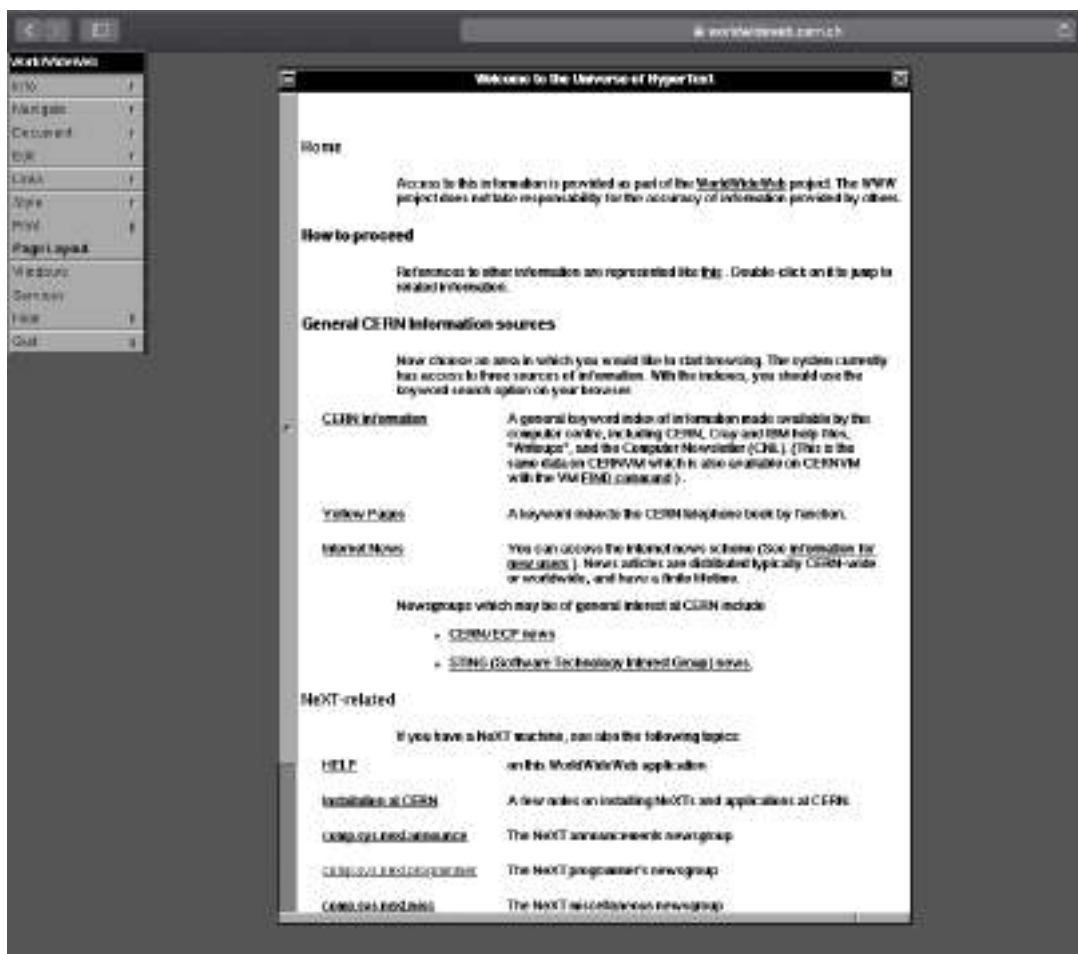


Figure 3 - First Web Browser ("CERN 2019 WorldWideWeb Rebuild", n.d.)

## 2.3 Milestones

In this chapter, I will follow up on the time after the release of the first web browser and complete the most important milestones from my point of view in the history of Web Browsers. The first two milestones in browser history - the ARPAnet (1968) and the WWW, the first web browser (1990) - were already described in the previous chapter.

The next milestone I want to mention is the second Web Browser called “The line-mode Browser”, which was launched in 1991. This was the first readily accessible browser for what we now know as the “world wide web”. The browser was also the first browser ever that has proven to work across various operating systems. Since it was operated via a simple command line interface, the browser could be used on many other computers and computing terminals throughout the Internet. (see “Line Mode Browser 2013”, n.d.)

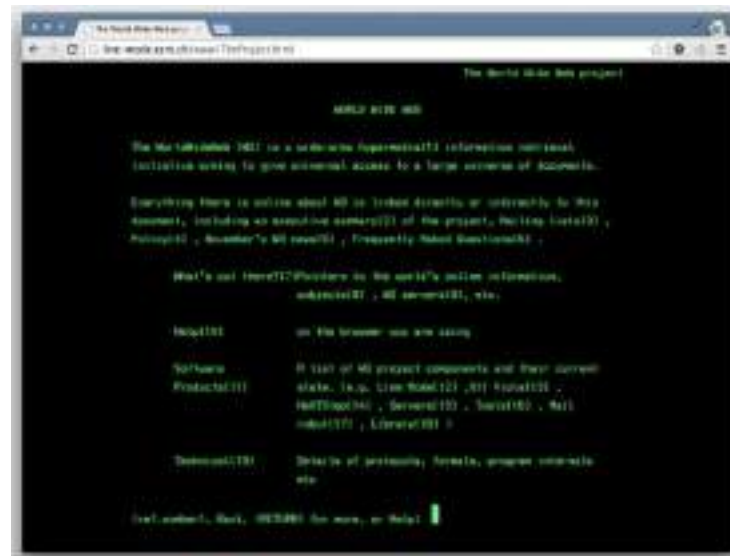


Figure 4- Line Mode Browser (“Line Mode Browser 2013”, n.d.)

In 1993, the National Center for Supercomputing Applications (NCSA) at the University of Illinois provided a pre-release version of its Mosaic browser for the Unix X Window System. The official launch happened in April of 1993. Mosaic quickly gained popularity and became the browser of choice due to its user-friendly graphical interface and easy installation process. Mosaic was able to display text and images together – a feature that no other browser was able to accomplish until then. Later that year, different versions of Mosaic – specifically for PC and Mac - became available. (“First pre-release of the Mosaic browser | timeline.web.cern.ch,” n.d.)



Figure 5 - Mosaic Browser (Goble, 2019)

As a greatly improved version of the Mosaic, the Netscape Navigator was developed next in 1994. Netscape Navigator supported the loading of a page directly at the request. As a major innovation, the Netscape browser brought, among other things, support for HTML frames, which were used to split browser windows into several sections, for which each individual section could load a separate HTML document. Until about 1996, Netscape Navigator was the leading web browser for Microsoft's 16-bit operating systems (up to Windows 95) and Apple Computer's Mac OS (Goble, 2019).

As previously mentioned, in October of 1994, Tim Berners-Lee founded the World Wide Web Consortium(W3C). This committee served for the standardization of techniques on the World Wide Web.

In 1996, Microsoft, which had not launched a browser before, finally released its first product - Internet Explorer (Fig. 6). This marked the beginning of the so-called "browser wars", in which Microsoft and Netscape competed for the web browser market.

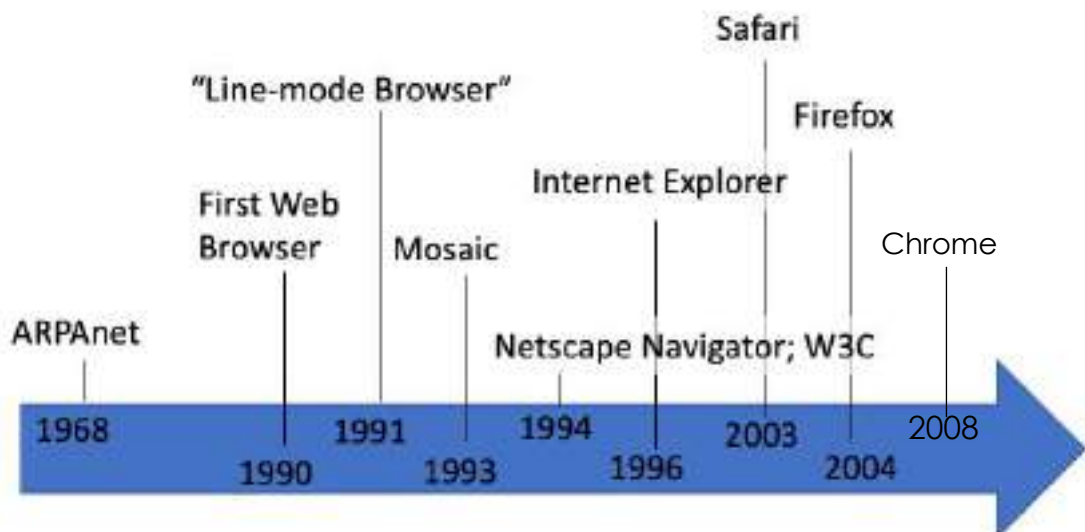


Figure 6 - Internet Explorer (Goble, 2019)

At the advent of the new millennium there are still some important milestones that I would like to mention briefly. In 2003, Apple released the Safari Browser, which was specifically designed for Macintosh computers instead of the Navigator. In 2004, the promising open source browser Firefox 1.0 was released. It was based on the Navigator from Netscape. In 2007, Mobile Safari was introduced as Apple's mobile web browser and dominates the iOS market ever since. (see McPeak, 2018)

In 2008, Google announced an in-house web browser called Chrome.

The timeline below shows the discussed milestones.



## 2.4 Browser Wars

To complete the history of web browsers, I would like to focus in this chapter on what is often referred to as the shadow side in the history of browsers. The browser war was a major rivalry between the companies Netscape and Microsoft, who at the time were dominating the web browser market. Hence, the stakes were high for all companies involved in the browser wars. A popular web browser could make a lot of money: while those companies who develop search engines would bid to be the standard tool used in the web browser, other companies with a web presence would bid to be listed in the standard set of bookmarks preinstalled with the browser. Because a web browser is a powerful gateway to a wealth of information, the company controlling that gateway could potentially have a great deal of influence over its users. In 1996, Netscape completely dominated the expanding browser market with a close to 90% market share. However, Microsoft began coincidentally shipping its browser, Internet Explorer, along with the Windows operating system, which was pre-installed on new computers. The strategy was a resounding success and Microsoft was later accused of violating the antitrust laws that are supposed to ensure unfettered and diverse competition across the market. In 2002, Internet Explorer led the browser market and was used by 95% of web users. (see Chatfield & Freytag, 2013)

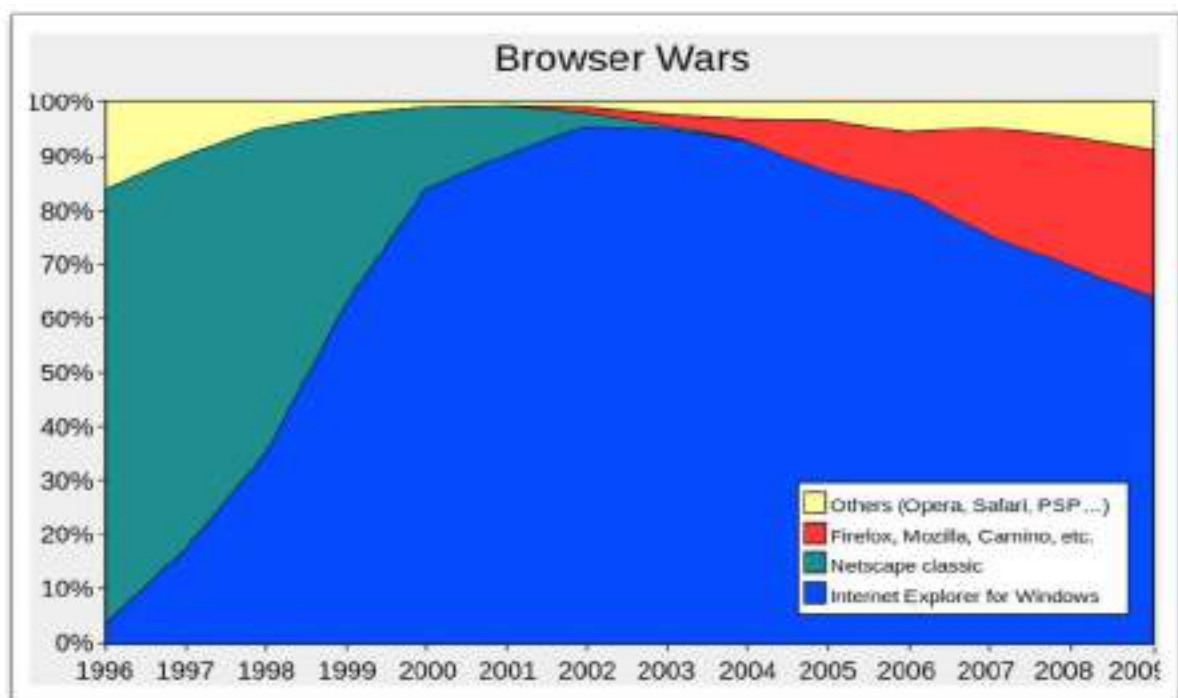


Figure 7 - Usage Share Web Browsers (Wikipedia contributors, 2020)

In the end, Microsoft won the first browser war because there was no real competition left. Netscape was sold to AOL in 1998 and slowly dissolved. Every user needed an operating system and a large part of them decided to choose Windows, which ultimately became the standard operating system. Users had the choice of using the faulty Internet Explorer for free or spending money on the Navigator. ("Browser Wars", Wikipedia contributors, 2020)

The next browser war, which started in 2004 and lasts until the Present day, can be seen as the second war among the top internet browsers. The primary browsers involved are the Internet Explorer, the winner from the first Browser War and Mozilla Firefox, which was a browser built on the Netscape open-source code, Google Chrome and Safari. Due to the many security holes in Internet Explorer, the demand for alternative browsers significantly increased. In 2004, Internet Explorer lost one percent of its market share to Mozilla products when serious security holes became known. After this, alternative browsers quickly found followers among those dissatisfied users who valued Internet Explorer thanks to its new and innovative functions. Since these alternative web browsers were also better at implementing common standards, web developers recommended these alternative browsers to dissatisfied users. Firefox eventually turned out to be the first browser succeeding in continuously taking away market share from Internet Explorer.

In 2008, Google announced an in-house web browser called Chrome. Optimization for performance triggered a race for the "fastest" browser, in which Safari, Opera and Firefox participated. While Microsoft's browser is losing more and more users, Google Chrome in particular is rapidly gaining new market shares. In addition, the battle for dominance on mobile devices is playing an important role today.



### 3. Concepts

In this chapter, the concept of web browsers is presented in a simplified way for a better understanding. Simply said, a browser helps to make the contents of the Internet visible on a user-friendly interface. For this purpose, hyperlinks are used, which allow the browser software to navigate to different pages. After the browser has called up this particular URI (= Uniform Resource Identifier; such as [www.google.com](http://www.google.com)), it reads a so-called HTML file on the target server. This is usually an HTML document, but can also be a PDF, image, or other type of content. With the help of these certain specifications, which are contained in the HTML (Hypertext Markup Language: is the standard markup language for Web pages) and CSS files (Cascading Style Sheets: describes how HTML elements are displayed on screen), the browser interprets all content and ensures that we can see and use the website on our screen. In addition, JavaScript enables interactive elements to be added that define what should happen.



Figure 8 - HTML5, CSS and JavaScript ("What is the role of HTML, CSS and JS in front-end development? - Quora," n.d.)

#### 3.1 Functionality

Most of the different browsers have a lot in common in terms of user interfaces. Common user interface elements include address bar to insert a URL, forward and backward buttons, bookmark options, refresh and stop buttons to refresh or stop loading current documents, and a home button that takes you to your home page. (see Garsiel & Irish, 2011)

Furthermore, a browser consists of the following components (See Fig. 9: Browser Components):



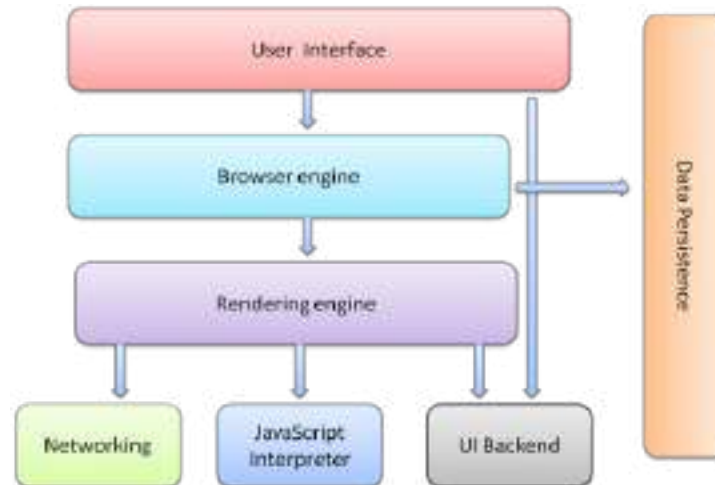


Figure 9 - Browser Components (Garsiel & Irish, 2011)

The **user interface** (UI) contains the address bar, back and forward buttons, a bookmark menu, etc. This includes every part of the browser display except the main window in which the requested page is displayed. The **browser engine** handles the actions between the user interface and the rendering module. The **rendering module** is responsible for displaying the requested content. For example, if the requested content is HTML, the module is responsible for parsing the HTML and CSS content and displaying the content on the screen. **Networking** is used for network calls such as HTTP requests. It has a platform independent interface and underlying implementations for each platform. **UI backend** is used to display basic widgets such as combo boxes and windows. It has a generic interface that is not platform specific. It uses the user interface methods of the operating system. The **JavaScript interpreter** is used for parsing (= analyzing, segmenting and coding machine-readable data) and executing the JavaScript code. The **data storage** is a persistence layer. The browser must store all kinds of data on the hard disk, for example cookies. (Garsiel & Irish, 2011)

The central element for the presentation of any content in the browser is the rendering engine. As already mentioned, it can display HTML and XML documents and images by default. However, other formats can also be displayed using a plug-in or browser extension, such as PDF documents who use a PDF Viewer plug-in. The primary use of the rendering engine however is to display HTML and images formatted with CSS. The rendering module starts with retrieving the contents of the requested document from the networking layer. The basic process of the rendering module looks like shown in the graphic below.

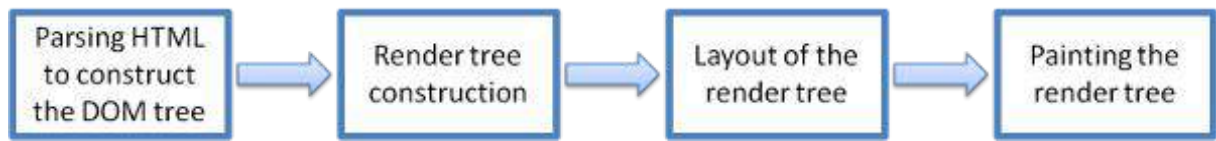


Figure 10 - Rendering Module (Garsiel & Irish, 2011)

The rendering engine will start parsing the HTML document and convert elements to DOM (Document Object Model) nodes in a tree called the "content tree". Parsing a document means translating it to a structure, which the code can then use. The result of parsing is usually a tree of nodes that represent the structure of the document. The engine will parse the style data, both in external CSS files and in style elements. Styling information together with visual instructions in the HTML will be used to create another tree: the render tree. This contains rectangles with visual attributes like color and dimensions. The rectangles have to be in the right order to be displayed properly on the screen. After the construction of the render tree, it goes through a "layout" process, which means giving each node the exact coordinates where it should appear on the screen. The next stage is painting—the render tree will be traversed and each node will be painted using the UI backend layer. For better user experience, the rendering engine will try to display contents on the screen as fast as possible. (Garsiel & Irish, 2011)

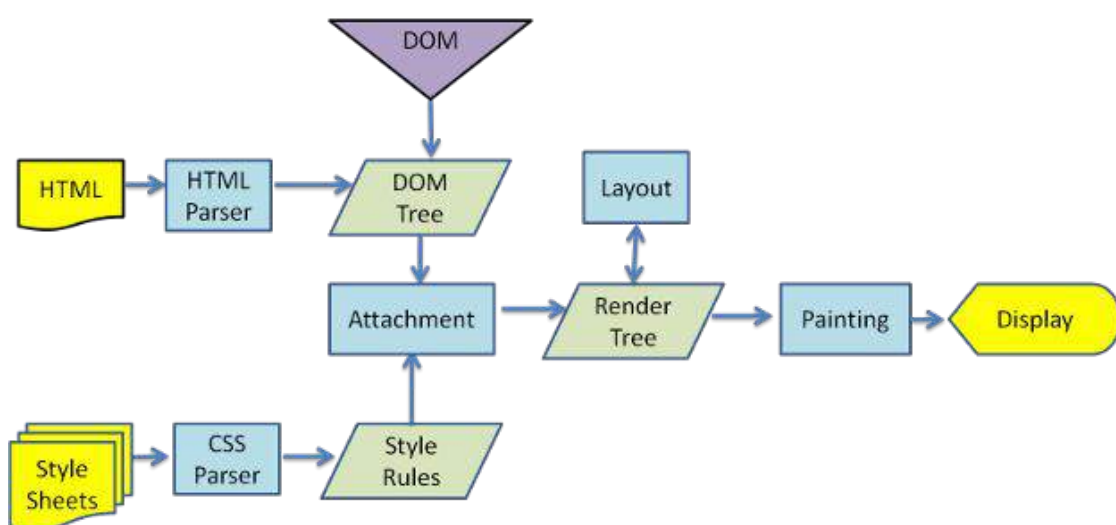


Figure 11 - Main Flow – Rendering Engines (Garsiel & Irish, 2011)

Besides the mentioned UI backend, which is responsible for the display, there are other modules. One of these modules handles the network requests using the common protocols, such as HTTP. A JavaScript interpreter parses and executes JavaScript code. A data memory is also provided as a persistent component, which stores, for example, cookies, the browsing history and the cache locally on the computer. (see "Was ist ein Browser?," 2020)

In this context I would like to mention the open source browser project Chromium with its rendering engine Blink, which aims to create a safer, faster and more stable way for all internet users to experience the web. Many modern browsers are based on the Chromium code, including Google Chrome, Microsoft Edge and Opera. ("Chromium - The Chromium Projects," n.d.) The reason for this is that the Blink rendering engine supports all modern standards, is fast, updates are pushed out rapidly, and bugs are patched quickly.

## 3.2 Standards

The way the browser interprets and displays HTML files is, as we already know, defined in the HTML and CSS specifications. These specifications are administered by different organizations. The best-known organization is the W3C (World Wide Web Consortium). For years, browsers have only followed a part of the specifications and developed their own extensions. This led to serious compatibility problems for web authors. To ensure high technical and editorial quality, and to earn endorsement by W3C and the broader community, W3C develops these technical specifications and guidelines through a process designed to maximize consensus about the content of a technical report, ("Standards - W3C," n.d.). Other important organizations for the development of web standards are the WHATWG, Internet Engineering Task Force (IETF) and the ECMA TC39 (Dickens, 2019). However, they all have a common goal and therefore most browsers comply with today's specifications and common standards. These Web standards are the technologies we use to build web sites. One of the main ideas of web standards is that the web should be free to contribute and use and should not be constrained by patents and licenses. Therefore, anyone can write the code to build a website, and anyone can contribute to the process of standards development, where the specifications are formed. In what follows, I will mainly focus on the W3C and their work with the web standards, as these have their

roots in the founder of the web browser. While the goal in the past was to achieve standard conformity for all browsers, the term web standards has expanded and today we understand it to mean a web design determined by best practices and universal principles, which on the one hand implements new developments even before their final definition, and on the other hand makes websites available to all users through accessibility and usability. In summary, web standards are primarily the formal standards and technical specifications that define and describe the rules and structures of HTML, CSS, SVG and JavaScript. Therefore, web browsers translate these languages to display the websites to its users.

### 3.2.1 HTML5 Standards

The HTML5 specification defines the 5th major revision of the core language of the World Wide Web, which is called Hypertext Markup Language and used for structuring and presenting content in the web. It was released in 2014 and introduced new features to support web application authors, new elements based on research on prevailing authoring practices, and paid special attention to defining clear conformance criteria for all developer to improve compatibility. (Hickson et al., 2014)

When it comes to the main changes that HTML5 has brought along, one particular change was the semantic markup. In the predecessors of HTML5 there was only a rough structuring by different `<div>` elements. Specified markups such as `<section>`, `<nav>`, `<article>`, `<aside>`, `<header>`, `<footer>`, or `<main>` allow you to define in HTML5 what kind of content each element is. (Wood, 2020)

Semantic markup of sections of a web page has one major advantage: it makes it easier for search engines to crawl an HTML document.

The next big change is the support of video and audio files. By supporting elements like `<video>` and `<audio>`, while `<canvas>` provides a defined space for JavaScript-created drawing and graphics, the need for plugins like Flash or Java became unnecessary. (Wood, 2020)

As a third and final point, I would like to go into more detail about the canvas `<element>`. The `<canvas>` element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, art, or other visual images on the fly. It is important to mention that the `<canvas>` element is only a container for graphics and JavaScript is needed to actually draw the graphics. Canvas

has several methods for drawing paths, boxes, circles, text, and adding images ("HTML Canvas," n.d.).

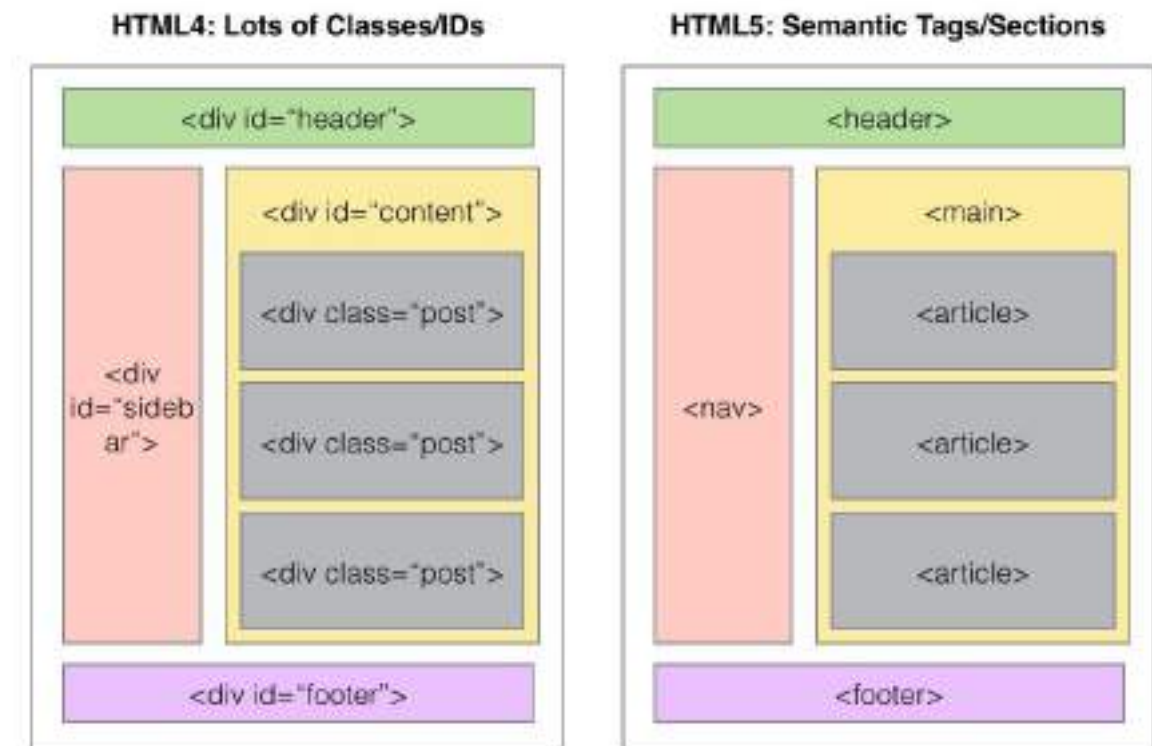


Figure 12 - HTML5 Semantic Tags ("HTML5 Semantic Tags | Viking Code School," n.d.)

### 3.2.2 CSS 3 Standards

Cascading Style Sheets (CSS) is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML. This separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. CSS is a common living standard and is constantly being developed by the W3C. ("Standards - W3C," n.d.)

The latest features are as follows: Users are able to give boxes a shadow via box-shadow and set multiple backgrounds. Furthermore, one also has the possibility to animate styles with CSS transitions. The typography has also been improved and authors have more control to achieve better typography. Authors can control text overflow with text-overflow and hyphenation, but they one also assigns shadows and control decorations even more precisely. Custom fonts can be downloaded and applied using the new `@font-face` at rule. ("HTML5," 2019)

Figure 13 shows an example of an animated button created with CSS.

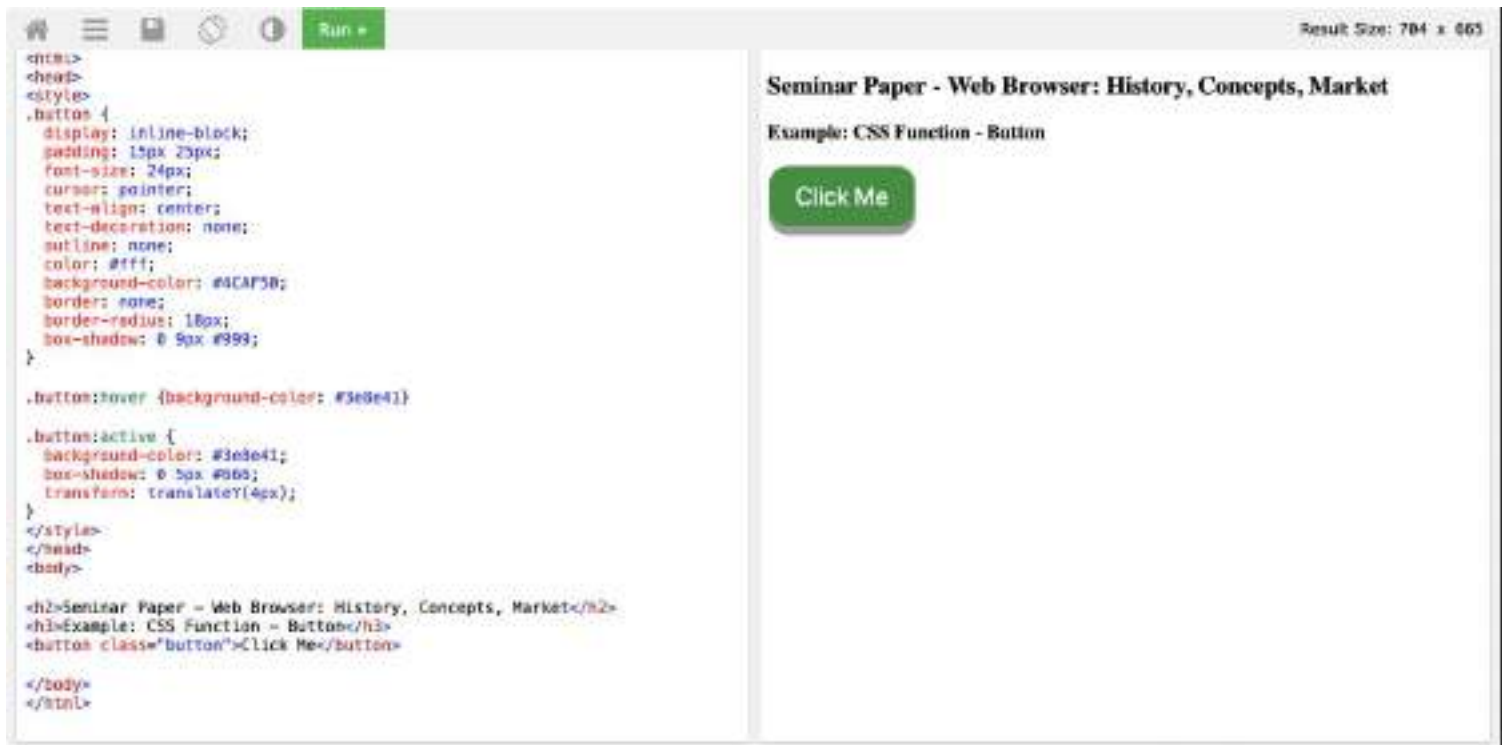


Figure 13 - Example CSS Animated Button ("Tryit Editor v3.6," n.d.)

### 3.2.3 JavaScript Standards

JavaScript constitutes the third part of nearly all modern webpages and is a scripting or programming language that allows users to implement complex features on web pages. JavaScript is used for displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, and more ("What is JavaScript?," 2020). Simply said without JavaScript a browser would just display static pages of information without user interactivity, animations, and reloading pages that try to fetch new information.



Figure 14 - HTML5 (The HTML 5 JavaScript API Index, n.d.)

In this context I would also like to mention the so-called Application Programming Interfaces (APIs), which offer additional possibilities to use the JavaScript code in

connection with HTML5. Figure 14 above gives an overview of the most common APIs. APIs are predefined sets of code building blocks that allow a developer to install programs that would otherwise be difficult or impossible to implement. In what follows, I will focus on some of the most important APIs related to HTML5.

The first API is WebRTC which is used for real-time communication via the browser. With WebRTC users can add real-time communication capabilities to their application, which work as an on top feature to open standards. WebRTC supports sending video, voice, and generic data between peers, and allows building powerful voice- and video-communication solutions. WebRTC consists of several interrelated APIs and protocols, which work together to achieve make this procedure possible. The technologies behind WebRTC are implemented as an open web standard and available as regular JavaScript APIs in all major browsers. ("Getting started with WebRTC," n.d.) This is especially important in today's world, as this API is under increasing demand in times of Covid-19. According to my research nearly all modern browsers support WebRTC at least in part.

The DOM API allows you to manipulate HTML and CSS, creating, removing and changing HTML, dynamically applying new styles to your page, etc. This can be the case, for example, whenever a pop-up window appears on a page or new content is displayed. Figure 15 highlights an example of the HTML DOM showing an event by using JavaScript.



Figure 15 - : Example Assign Events Using the HTML DOM ("Tryit Editor v3.6," n.d.)



Another API, the Geolocation API, enables us to get the physical location of a device and is a well-supported API implemented in more than 95% of the browsers ("What is JavaScript?," 2020). The example below returns the latitude and longitude of the user's position:

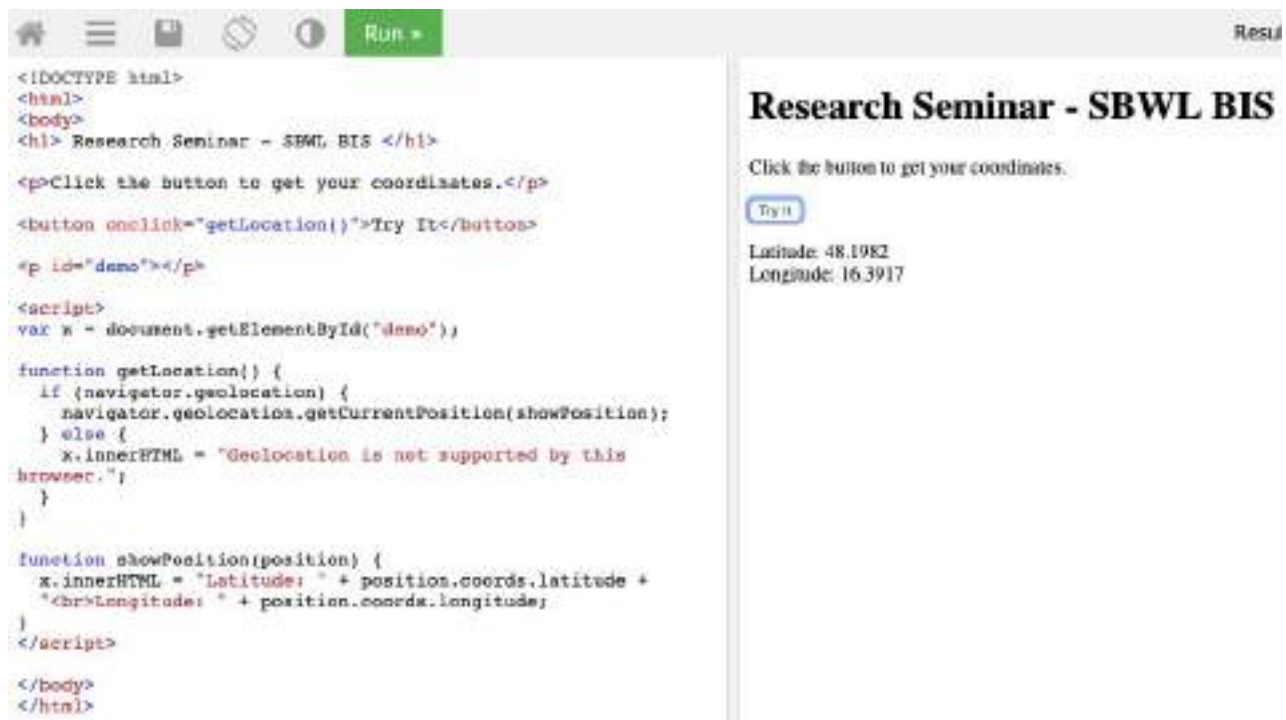


Figure 16 - Example Geolocation ("Tryit Editor v3.6," n.d.)

In this context, it is important to note that if a JavaScript application initiates a request for the location of any device before it can receive the data, this may also be an invasion of personal privacy. The website needs to request permission to receive data from the user. This is usually done with a popup, which is initiated via the DOM API as we have explained above. The browser then receives the data via the GPS of the device.

The HTML-canvas element creates a drawing area in which Javascript graphics, images and drawings can be created and animated. Since this was already mentioned in a previous chapter on the HTML5 standards, I will only show a short practical example (see Fig. 17). Figure 17 shows an animated square moving from the left to the right within the canvas.



```
function animate() {
    requestAnimationFrame(animate);

    x += speed;
    if(x <= 0 || x >= 475){
        speed = -speed;
    }
    // y = y + 10; so könnte das Objekt am Ende einer Zeile eine Zeile nach unten gehen
    draw();
}

function draw() {
    var canvas = document.getElementById('canvas');
    var context = canvas.getContext('2d');

    context.clearRect(0, 0, 500, 300);
    context.fillStyle = '#dfac20';
    context.fillRect(x, y, 20, 20);
    context.lineWidth = 3;
    context.strokeStyle = '#3983ab';
    context.strokeRect(x, y, 20, 20);
}
```

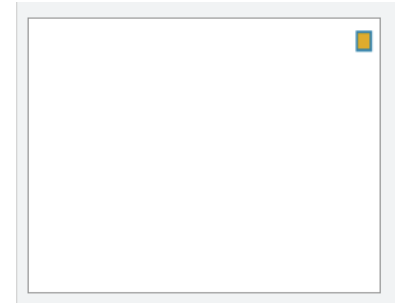


Figure 17 - Example Moving Object in Canvas ("Tryit Editor v3.6," n.d.)

### 3.2.4 Best Practices

During my research I came across some best practice guidelines that developers should follow to ensure the best use of web technologies. In web development, the main cause of uncertainty is that web developers can't anticipate what combination of technologies each user will use to view certain website. While some users browse via their smartphone, others use a Windows PC or even screen readers. Precisely because developers don't know what devices or technologies users are relying on to access different websites, developers need to make each site as flexible as possible. The following best practices should be used as guidelines:

- **Cross-browser compatibility** is the attempt to ensure that your website works on as many devices as possible. This includes using technologies that are supported by all browsers. Ensuring unilateral compatibility also involves a lot of testing to see whether something fails in certain browsers, and then to fix those failures. In this context **responsive web design** is also important. This is the practice of making your functionality and layouts flexible, so that they can automatically adapt to different browsers. **Performance** also plays a big role, since developers also have to ensure that their websites load and fetch information as fast as possible.

- A very important aspect is **privacy and security**. These two concepts are related, but obviously different from each other. Privacy refers to being private, and not spying on users or collecting more data than absolutely necessary. Security refers to making your site secure so that malicious users cannot steal the information from websites and/or users.
- **Accessibility** refers to making your websites usable by as many people as possible. Individuals with visual impairments, hearing impairments, cognitive disabilities or physical disabilities should be able to navigate different websites easily and people from different cultures, regardless of their age, country of origin, operating system, and speed of network connection should be able to access various websites . To account for different languages, it is also crucial to make websites available in different languages so that people from diverse cultures can easily navigate the web.

(see “The web and web standards,” 2020)

It must be said that these rules apply more to the developers of web pages, but are nevertheless closely linked to web browsers.

## 4. Market

As previously mentioned in chapter 2, there has always been a considerable and fierce level of competition amongst browser providers. In this chapter, I would like to revisit the browser wars, cover them in more depth and analyze the market in more detail to compare why certain browsers were able to take the lead and others fell behind.

### 4.1 General Overview

The current market shares of the various web browsers can be seen in Fig. 18. The current market leader is the Chrome Browser with its 2 latest versions, which together have a market share of just over 50%. However, looking back in time, this has not always been the case.

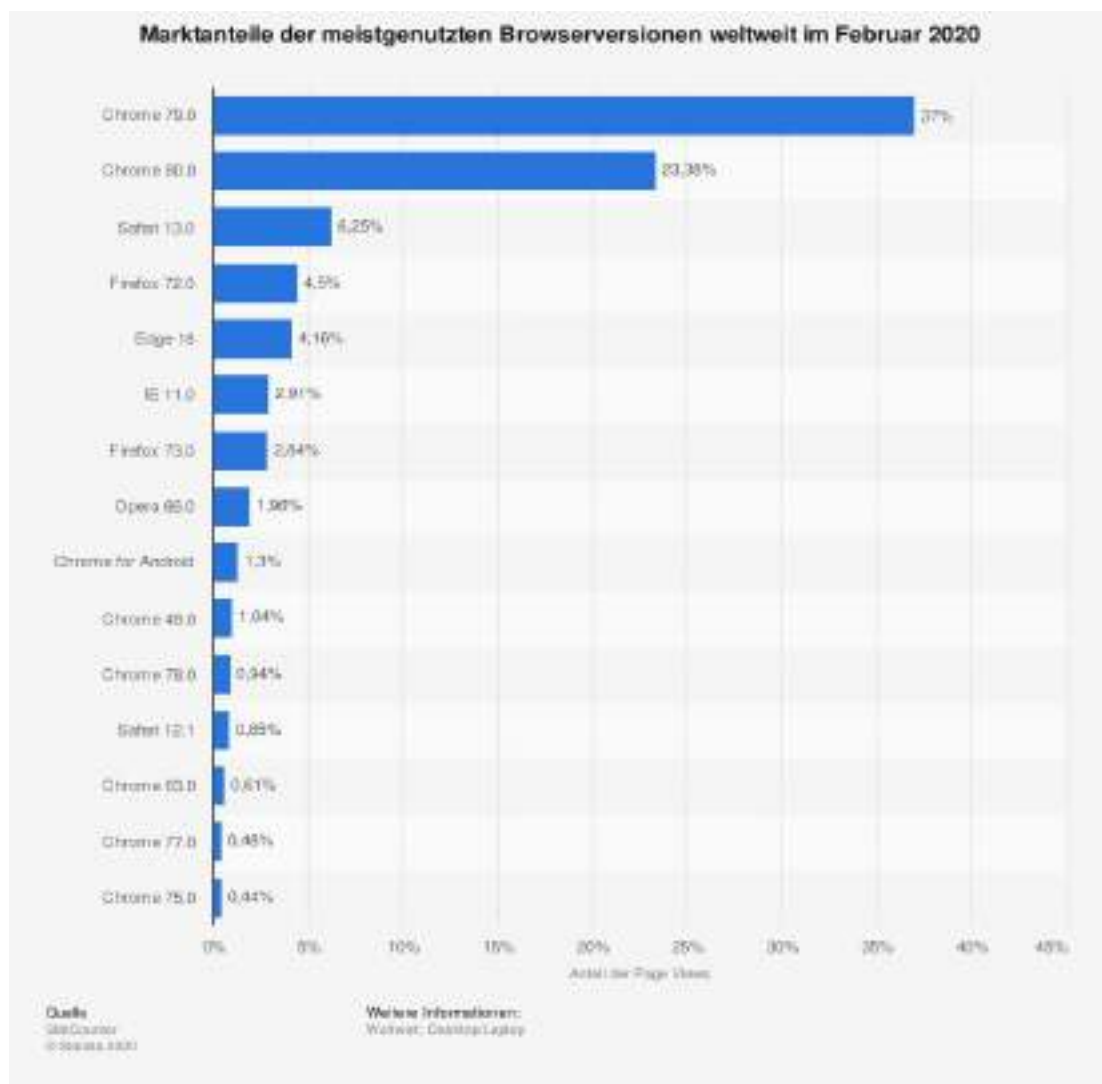


Figure 18 - Market Share February 2020 Worldwide (StatCounter, 2020)

Fig. 19 depicts the development of market shares across various browsers since 2009. While the foundation of web browsers has been described up until the year 2009 in chapter 2.4 “Browser Wars”, I will now exclusively focus on the historic development of market shares post-2009.

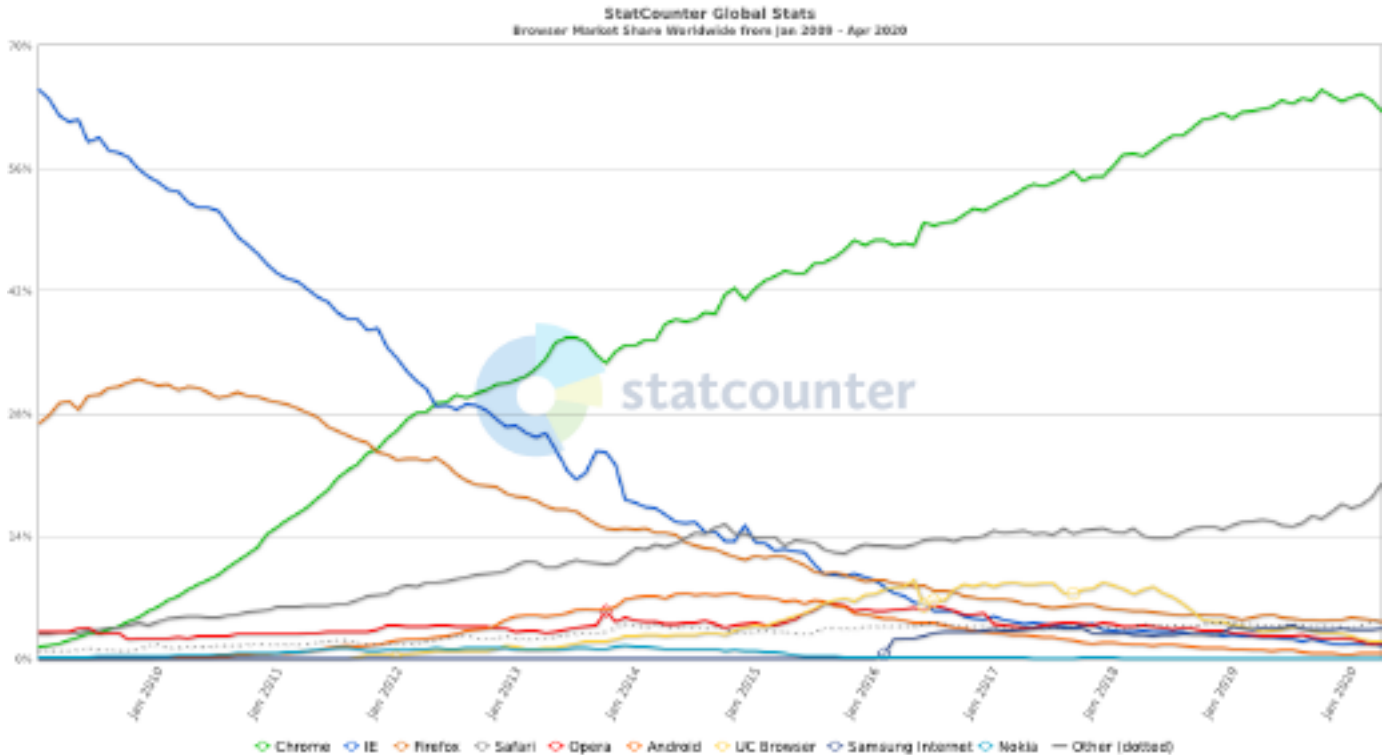


Figure 19 - Browser Market Share Worldwide from Jan 2009 -Apr 2020 (“Browser Market Share Worldwide,” n.d.)

This rise of Google Chrome as a market leader can be attributed to the outdated technology of Internet Explorer. As a result, IE lost its popularity amongst existing users at that time and Chrome was able to differentiate itself primarily through its superior speed. Google Chrome was able to extend this competitive advantage even further via constant developments in more recent years. However, Google has to be careful, because such monopoly positions are still of great importance for the browser war, since in EU antitrust law the abuse of power positions in the software sector is also monitored and, if necessary, sanctioned by the EU Competition Commissioner

## 4.2 Benchmarking of the Different Browsers

Given what I have presented on the various browsers so far, the competitive landscape begets the following questions: What features need to be offered to make a browser successful and given Chrome’s majority of the market share, is there really no

competition for Google Chrome? To evaluate these claims, I used different tools to benchmark the different browsers. It is important to note that all tests were performed on a MacBook Air (13-inch, Early 2015) with the macOS Catalina operating system, and may differ if testing is performed on other operating systems. For a closer comparison I have used the latest version of the following browsers, based on their respective market share: Google Chrome, Safari, Mozilla Firefox, Microsoft Edge. Due to its interesting features and as a newcomer in the industry, I also added the Brave Browser as an additional candidate for my testing.

For the benchmarks I used the tools HTML5test (<http://html5test.com>), Basemark Web 3.0 (<https://web.basemark.com>), and Ares-6 (<https://browserbench.org/ARES-6/>).

#### 4.2.1 ARES-6

The ARES-6 benchmark measures the execution time of JavaScript and the newest features of each browsers. Simply put, ARES-6 measures the speed of browsers, how fast they start and if they run smoothly. A lower loading time (measured in milliseconds) is indicative of superior performance.

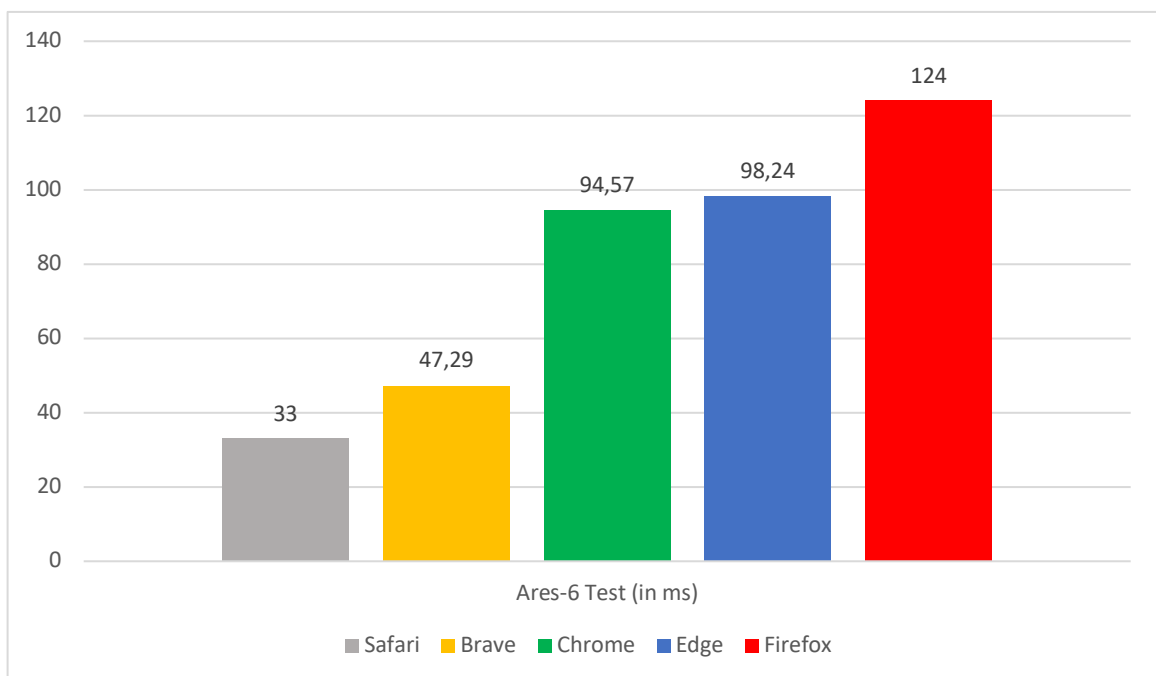


Figure 20 - Graphic Ares-6

As the chart above indicates, Brave and the Safari browser pulled away from the competition. While Safari took a total of 33 milliseconds in the Ares-6 test, Brave

came in second with a result of 47.29 milliseconds. Chrome, Firefox and Edge were significantly above the 90-millisecond mark.

#### 4.2.2 Basemark Web 3.0

Basemark Web 3.0 is based on various system and graphic tests, which use the web recommendations and features. To do this, the tool uses low-level JavaScript calculations (ECMAScript) and graphically intensive content such as WebGL 2.0, which mainly uses the GPU of the device. In addition, tests used, which were created with JavaScript frameworks such as jQuery.

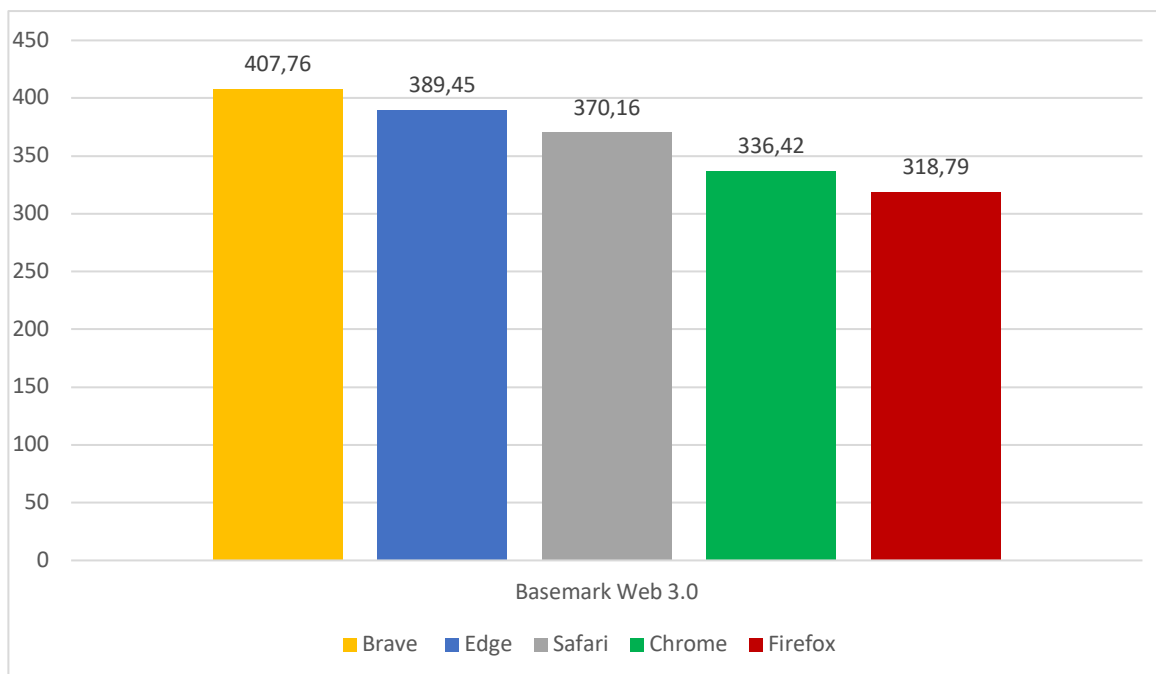


Figure 21 - Graphic Basemark Web 3.0

In the Basemark Web 3.0 test, the Brave Browser came out on top. The Brave browser scored 95.5% on HTML5 capabilities, 94.98% on Page Load and Responsiveness Capabilities, 75.97% on Resize Capabilities and performed rather poorly on CSS capabilities with 58.23%.

The Edge Browser scored better this time and achieved almost the same percentages as the Brave Browser. The Firefox and Chrome browsers performed the worst in the test.

### 4.2.3 HTML5test

The HTML5test measures how well your browser supports the HTML5 standard and its related specifications. The following categories are checked: parsing rules, elements, forms, web components, location and orientation, security, performance and many more (compare for more details <http://html5test.com>). The maximum score in this test is 555 points.

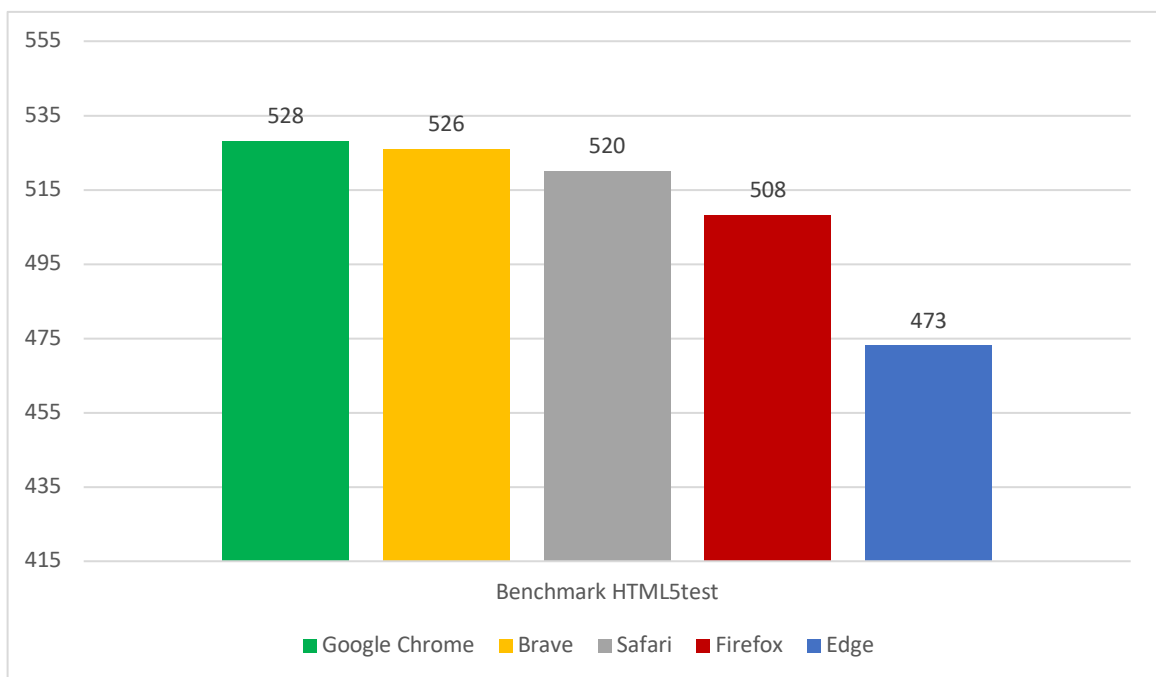


Figure 22 - Graphic HTML5test

As Graph 22 above reveals, almost all browsers support the same functions. The Google Chrome Browser is the winner, but it is closely followed by the underdog, the Brave Browser. The big loser, where most of the deductions were made for problems with the features because they could not be used or only partially used, was the Edge Browser.

## 4.3 Summary

Across all tests, the results were very different. None of the most used browsers could really succeed. The best results overall, however, were achieved by the underdog - the Brave Browser. The Basemark Web 3.0 test was the most meaningful benchmark

test because, as mentioned before, it includes various system and graphics tests based on the latest web standards and features. This is important because it is what users see that counts in the end. Nevertheless, it must be said that all tested browsers met today's standards. For the reason that there is no clear winner, I would like to give a short review of the tested browsers.

#### **4.3.1 Brave**

The Brave Browser could keep up with the most used browsers in all benchmark tests. Brave, like many other browsers, uses the Chromium rendering engine because it complies with web standards. I also used this browser for the benchmark because it is very security and privacy oriented. Brave can open tabs connected to the Tor network without the need to install the Tor Browser. This allows you to surf even more anonymously. Brave also uses the search engine DuckDuckGo, which does not collect any personal information. Brave also supports most chrome extensions in the Chrome Web Store. ("Secure, Fast & Private Web Browser with Adblocker," n.d.)

#### **4.3.2 Google Chrome**

Chrome has been very popular since its inauguration and has, so far, been able to expand its market share due to its high performance, stability, user-friendliness and innovative design. Although the Chrome Browser scored mid-range results in the benchmark tests carried out above, it was still the most used browser. By default, Google Chrome is very secure due to regular updates and has a spam blocker and a private mode. However, Chrome has a problem when it comes to tracking protection, as everything is closely linked to the Google Account. Finally, it has to be mentioned that Chrome is a compatible browser, and like its competitors, is also based on Chromium technology.

#### **4.3.3 Microsoft Edge**

After losing the browser war with Internet Explorer, which showed strong weaknesses, Microsoft is launching the Edge Browser in 2015. Throughout my testing, this browser ranked in the lower half of tested browsers, however, is still a new competitor on the



browser market. The Edge Browser also relies on Chromium. Therefore, it works quite fast and can handle chrome extensions. In addition, Edge is pre-installed on every Windows device, which can help Microsoft gain new market share. However, there are weaknesses in the area of privacy. The user has to choose between three settings: Simple, Balanced and Strict - but the description of these settings is not very clear. Unfortunately, third-party cookies are allowed by default.

#### **4.3.4 Mozilla Firefox**

After the lost browser war, Netscape released the code of its browser and founded the Mozilla Foundation. After a conceptual reorientation, Netscape Navigator thus became Mozilla Firefox. However, it did not really shine in the benchmark test, as the competition is very strong. Unlike the tested browsers, Firefox uses the Gecko rendering engine. Nevertheless, it still scores points for its privacy protection functions. Recent updates include better privacy protection with anti-tracker support, improved password synchronization between devices, and built-in alerts for security breaches. Firefox is also customizable, both in its appearance and in the choice of extensions and plugins you can use.

#### **4.3.5 Safari**

The Safari browser is standard on every Mac and iOS device. More recently, Safari has added fingerprint protection that prevents web trackers from identifying users by their system specifications. Safari is, similar to other browsers, up to date in supporting new HTML5 features and ranked solidly in the midfield across all the test.

## 5. Conclusion

First of all, I hope that this paper has given the reader a basic understanding of the subject by offering a simplified insight into a very complex subject. Given the huge amount of information, it proved to be challenging to review the literature, and browsers, despite their importance, are rather seen as a by-product of the internet. Due to the groundbreaking development of the Internet, there has been an avalanche of technical innovations in recent years. A pioneer in this context was Tim Berners-Lee. With Berners-Lee's first browser it was possible to discover the previously unknown depths of the Internet. Furthermore, Berners-Lee was the founder of the W3C, which was founded to create a common standard on which all developments are based. Even today, Berners-Lee is still working on creating common standards. In the early beginning of the history of browsers, there was still a great deal of competition amongst the various companies. This fierce competition and to attract users still characterizes the present-day industry and can – as I have alluded to in previous chapters – be referred to as browser wars. Ultimately, more users mean more profit. This plays an even greater role in today's world as technology becomes more affordable. Companies are therefore trying to implement more and more innovations to differentiate themselves and stand out in a crowded market space. The ongoing competition has led to great differences when it comes to browsers standards and features. Therefore, I consider the work of the W3C and similar organizations of uttermost importance as they have laid the groundwork for common standards in the industry.

In the chapter which covers the various concepts, I initially struggled to find my way. I found it very interesting to first work through the functionality of the browsers and then to understand the standards – especially the HTML5 standard. In this context I found the Best Practices interesting, because I liked the basic idea that everyone is equal. In the last chapter of my seminar paper I tried to review the market and offer an interesting, yet less technical and better-to-understand comparison of the top browsers. Here it was especially interesting that an underdog could perform better across most tests than the market leaders. While the results certainly allowed me to rank browsers hierarchically according to their performance, it has to be emphasized that one has to distinguish clearly between laptop, desktop, mobile, tablet when these tests are performed. Testing done on a different device might yield different results and

therefore could be vastly different from the results presented in this paper . My personal favorite is the browser Brave because it offers the best possibilities to stay anonymous on the net and still offers a great design with the latest technology.

In summary, I want to emphasize that it was very interesting to write this paper and to learn more about this complex topic. Finally, I would like to thank Mr. Flatscher for always encouraging us to work out solutions ourselves and to think critically.

## 6. Bibliography

Browser Market Share Worldwide. (n.d.). Retrieved May 1, 2020, from <https://gs.statcounter.com/browser-market-share#monthly-200901-202004>

Browser. (n.d.). Retrieved April 1, 2020, from <https://www.duden.de/node/25742/revision/25771>

CERN 2019 WorldWideWeb Rebuild. (n.d.). Retrieved April 14, 2020, from <https://worldwideweb.cern.ch>

Chatfield, T., & Freytag, C. (2013). 50 Schlüsselideen Digitale Kultur. Weinheim, Germany: Beltz Verlag

Chromium - The Chromium Projects. (n.d.). Retrieved May 15, 2020, from <https://www.chromium.org/Home>

Dickens, A. (2019, Januar 14). Web Standards: The What, The Why, And The How. Retrieve April 30, 2020, from <https://www.smashingmagazine.com/2019/01/web-standards-guide/>

Facts About W3C. (n.d.). Retrieved April 9, 2020, from <https://www.w3.org/Consortium/facts>

First pre-release of the Mosaic browser | timeline.web.cern.ch. (n.d.). Retrieved April 15, 2020, from <https://timeline.web.cern.ch/first-pre-release-mosaic-browser>

Garriel, T., & Irish, Paul. (2011, August 5). How Browsers Work: Behind the scenes of modern web browsers - HTML5 Rocks. Retrieved April 18, 2020, from <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>

Getting started with WebRTC. (n.d.). Retrieved May 5, 2020, from <https://webrtc.org/getting-started/overview>

Goble, G. (2019, December 28). Browser-History: Die 10 wichtigsten Meilensteine. Retrieved April 15, 2020, from [https://www.pcwelt.de/ratgeber/Browser-History\\_Die\\_10\\_wichtigsten\\_Meilensteine-Internet-7909354.html](https://www.pcwelt.de/ratgeber/Browser-History_Die_10_wichtigsten_Meilensteine-Internet-7909354.html)

Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, T., & Pfeiffer, S. (2014, October 28). HTML5. Retrieved May 2, 2020, from <https://www.w3.org/TR/2014/REC-html5-20141028/>

HTML Canvas. (n.d.). Retrieved May 4, 2020, from [https://www.w3schools.com/html/html5\\_canvas.asp](https://www.w3schools.com/html/html5_canvas.asp)

HTML5. (2019, March 23). Retrieved May 4, 2020, from <https://developer.mozilla.org/de/docs/Web/HTML/HTML5>

HTML5 Semantic Tags | Viking Code School. (n.d.). Retrieved May 3, 2020, from <https://www.vikingcodeschool.com/html5-and-css3/html5-semantic-tags>

JavaScript/Canvas/Animation – SELFHTML-Wiki. (n.d.). Retrieved May 5, 2020, from <https://wiki.selfhtml.org/wiki/JavaScript/Canvas/Animation>

Kroker, M. (2016, May 4). Die ersten Male im Internet: Von der ersten E-Mail 1971 bis zum ersten Instagram-Foto 2010 | Kroker's Look @ IT. Retrieved April 13, 2020, from <https://blog.wiwo.de/look-at-it/2016/05/04/die-ersten-male-im-internet-von-der-ersten-e-mail-1971-bis-zum-ersten-instagram-foto-2010/>

Line Mode Browser 2013. (n.d.). Retrieved April 15, 2020, from <http://line-mode.cern.ch>

McPeak, A. (2018, September 7). A Brief History of Web Browsers and How They Work. Retrieved April 15, 2020, from <https://crossbrowsertesting.com/blog/test-automation/history-of-web-browsers/>

Mercier, C. (n.d.). World Wide Web Consortium (W3C). Retrieved April 9, 2020, from <https://www.w3.org/People/Berners-Lee/>

Moussaoui, H. E., & Zeppenfeld, K. (2007). AJAX. New York, United States: Springer Publishing.

Online, F. (2009, March 13). Vor 20 Jahren: Internet versus World Wide Web. Retrieved April 13, 2020, from [https://www.focus.de/digital/internet/internetgeschichte/tid-13637/vor-20-jahren-internet-versus-world-wide-web\\_aid\\_379722.html](https://www.focus.de/digital/internet/internetgeschichte/tid-13637/vor-20-jahren-internet-versus-world-wide-web_aid_379722.html)

Secure, Fast & Private Web Browser with Adblocker. (n.d.). Retrieved May 2, 2020, from <https://brave.com>

Standards - W3C. (n.d.). Retrieved April 20, 2020, from <https://www.w3.org/standards/>

StatCounter. (18. März, 2020). Marktanteile der meistgenutzten Browserversionen weltweit im Februar 2020 [Graph]. Retrieved April 8, 2020, from <https://de.statista.com/statistik/daten/studie/158095/umfrage/meistgenutzte-browser-im-internet-weltweit/>

The birth of the Web | CERN. (n.d.). Retrieved April 8, 2020, from <https://home.cern/science/computing/birth-web>

Th HTML 5 JavaScript API Index. (n.d.). Retrieved May 5, 2020, from <http://html5index.org>

The original proposal of the WWW, HTMLized. (n.d.). Retrieved April 14, 2020, from <https://www.w3.org/History/1989/proposal.html>

The web and web standards. (2020, April 17). Retrieved May 1, 2020, from [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/The\\_web\\_and\\_web\\_standards](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards)

Tryit Editor v3.6. (n.d.). Retrieved May 4, 2020, from [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_buttons\\_animate3](https://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_animate3)

Was ist ein Browser? (2020, February 13). Retrieved April 20, 2020, from <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-ein-browser/>

What is JavaScript? (2020, April 28). Retrieved May 4, 2020, from [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

What is the role of HTML, CSS and JS in front-end development? - Quora. (n.d.). Retrieved May 4, 2020, from <https://www.quora.com/What-is-the-role-of-HTML-CSS-and-JS-in-front-end-development>

Wikipedia contributors. (2020, April 2). Tim Berners-Lee. Retrieved April 9, 2020, from [https://en.wikipedia.org/w/index.php?title=Tim\\_Berners-Lee&oldid=948684222](https://en.wikipedia.org/w/index.php?title=Tim_Berners-Lee&oldid=948684222)

Wikipedia contributors. (2020, April 5). Browser wars. Retrieved April 16, 2020, from [https://en.wikipedia.org/wiki/Browser\\_wars#/media/File:Layout\\_engine\\_usage\\_share-2009-01-07.svg](https://en.wikipedia.org/wiki/Browser_wars#/media/File:Layout_engine_usage_share-2009-01-07.svg)

Wikipedia contributors. (2020, April 9). Internet. Retrieved April 13, 2020, from <https://en.wikipedia.org/w/index.php?title=Internet&oldid=949988003>

Wikipedia contributors. (2020, March 30). Web browser. In Wikipedia, The Free Encyclopedia. Retrieved April 1, 2020, from [https://en.wikipedia.org/w/index.php?title=Web\\_browser&oldid=948219627](https://en.wikipedia.org/w/index.php?title=Web_browser&oldid=948219627)

Wood, A. (2020, January 15). HTML5 Basics For Everyone Tired Of Reading About Deprecated... Retrieved May 3, 2020, from [https://html.com/html5/#What\\_is\\_HTML5](https://html.com/html5/#What_is_HTML5)