

---

# HTML5

---

## ***Seminar Paper***



**Date:** June 1, 2021

**Author, ID:** Mustafa Rafie, 01612425

**University:** University of Business and Economics Vienna

**Subject:** Business Information Systems Seminar

**Course number:** 4152

**Instructor:** ao.Univ.Prof. Dr. Rony G. Flatscher

**Semester:** SS 2021



## **Declaration of Authorship**

I assure:

- to have individually written, to not have used any other sources or tools than referenced and to not have used any other unauthorized tools for the writing of this seminar paper.
- to never have submitted this seminar paper topic to an advisor neither in this, nor in any foreign country.
- that this seminar paper matches the seminar paper reviewed by the advisor.

Date: 1st June 2021

Signature: Mustafa Rafie

# Table of Contents

<b>1. Introduction .....</b>	<b>5</b>
1.1 About this Paper .....	5
<b>2. History.....</b>	<b>6</b>
2.1 The Birth of the Web .....	6
2.2 The History of Browser Wars.....	7
2.2.1 The First Browser War (1995-2001) .....	7
2.2.2 The Second Browser War (2004-2017) .....	8
2.3 W3C and WHATWG .....	9
2.3.1 The W3C.....	9
2.3.2 History Timeline of HTML.....	9
2.3.3 The W3C vs. WHATWG Conflict .....	11
<b>3. HTML5 Elements .....</b>	<b>13</b>
3.1 DOCTYPE .....	14
3.2 New Elements in HTML5.....	15
3.2.1 Basic Information on HTML Elements .....	15
3.2.2 Semantic in HTML Documents .....	17
3.2.3 Structural Elements .....	19
3.2.4 Text Elements.....	20
3.2.5 Form Elements.....	22
3.2.6 Embedded content .....	27
3.2.7 The new <dialog> Element of HTML 5.2 .....	30
<b>4. APIs .....</b>	<b>30</b>
4.1 What are APIs? .....	30
<b>5. Changes from HTML5 to 5.2 .....</b>	<b>32</b>
5.1 Changes in HTML 5.1.....	33
5.2 Changes in HTML 5.2.....	33
5.3 HTML – Living Standard .....	34
<b>6. Use of HTML5 Today .....</b>	<b>34</b>
6.1 Browser support for HTML5 .....	34
6.2 General Criticism of HTML5 .....	35
<b>7. Conclusion .....</b>	<b>36</b>
<b>Bibliography .....</b>	<b>37</b>

## List of Figures

1. Figure: WWW- 3 Specification by Tim Berners-Lee.....	6
2. Figure: History Timeline from HTML to HTML 5.2 (W3C) .....	13
3. Figure: DOCTYPE Syntax of HTML 4.01 [76] .....	14
4. Figure: HTML Element Structure [77] .....	16
5. Figure: Example of a HTML Attribute .....	17
6. Figure: Browser output HTML Attribute .....	17
7. Figure: Non-Semantic vs Semantic Page Structure [78] .....	18
8. Figure: Example for <div> element.....	19
9. Figure: Example for Structural Element.....	20
10. Figure: Example of a <time> element: .....	21
11. Figure: Output of the <time> element .....	21
12. Figure: Output <datalist> Example .....	23
13. Figure: <datalist> Example.....	23
14. Figure: <progress> Example .....	24
15. Figure: Output <progress> Example.....	24
16. Figure: <meter> Example.....	24
17. Figure: Output <meter> Example .....	25
18. Figure: <input> Tag Example .....	27
19. Figure: Output <input> Tag .....	27
20. Figure: <audio> Example.....	29
21. Figure: Output <audio> .....	29
22. Figure: <video> Example.....	29
23. Figure: <dialog> Example.....	30
24. Figure: Output <dialog> Element .....	30
25. Figure: Browser output - Geolocation Request.....	32
26. Figure <details> Example .....	33
27. Figure: Output <details> .....	33
28. Figure: Browser Comparison.....	35

**List of Tables**

Table 1: New Structural Elements in HTML5 [34] .....19

Table 2: New HTML5 Text Elements .....21

Table 3: HTML5 - New Form Elements .....23

Table 4: HTML5 new <input> Attributes .....26

Table 5: New <input> Types [43].....27

Table 6: New HTML5 embedded content .....28

Table 7: New HTML5 APIs.....32

# 1. Introduction

## 1.1 About this Paper

This seminar paper deals with HTML5, also called as HyperText Markup Language, which is the current standard markup language for web programming. With the help of HTML5, everybody is able to create web pages or web applications with hyperlinks, images and other content. It is basically the building block of the web and is used to structure the content of documents of a web page or a web application. As more and more people began to use the Internet and mobile devices over the last few years, dynamic content has become increasingly important. Therefore, the Web Hypertext Application Technology Working Group (WHATWG) has decided to release the first public draft of the fifth generation of HTML with new features and capabilities in 2008. [1]

The seminar paper is divided into five main parts and covers the following topics. The first parts will discuss the whole history of HTML including the invention of the World Wide Web and the browser wars especially the conflict between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The second part will explore the new elements in HTML5. You can find out how you can implement these new elements into your own website. In addition, you will get insights on what differences there are to the previous HTML standards and a short overview of elements that no longer exist.

The third part gives an overview of the new APIs and will show the differences between the previous models and the new ones. You can find out how you can implement these new APIs into your own website and what advantages they have.

The fourth part deals with the HTML5 timeline. The W3C is constantly working on improvements and further development of HTML5. In this part, you get an overview of the changes in the last versions of HTML5.

The last part deals with general questions such as how HTML5 is used in the world and what the criticisms of HTML5 are and if there will be a new version soon.

The target of this paper is that it should be shown that the text-based markup language is really a blessing for the internet world. Or is it simply overhyped?

## 2. History

### 2.1 The Birth of the Web

Everything we are able to do on the internet these days such as searching on Google, being in touch with friends and family through social media platforms, reading online newspapers and so on is only possible through Tim Berners-Lee, who invented the World Wide Web in 1989. Tim Berners-Lee had studied physics at the University of Oxford. He is a British researcher and computer scientist who worked as a software developer at CERN, the European Particle Physics Laboratory. Tim and his colleagues had big problems with exchanging information and research findings between Switzerland and other countries due to different network infrastructures. It was not possible to surf the Internet like today because there were no hyperlinks and developers could only exchange data via FTP servers. Digital data could only be exchanged with effort. In order to solve this problem, he invented a software called "Enquire", which was meant for communicating by sharing information electronically between several people in a simple, fast and structured way and link them efficiently with each other. This markup language was based on the standard of Standard Generalized Markup Language (SGML). Based on this software, Tim had a brilliant idea and submitted an innovative project to CERN. Finally, the World Wide Web was born and the first website went online on December 20, 1990. [2] In the same year, he wrote the 3 specifications that were crucial for the system:

<b>HTTP</b>	<ul style="list-style-type: none"><li>• <b>HTTP protocol (Hypertext Transfer Protocol)</b></li><li>• For communication between web server and web client.</li></ul>
<b>HTML</b>	<ul style="list-style-type: none"><li>• <b>HTML (Hypertext Markup Language)</b></li><li>• Markup language for web documents.</li></ul>
<b>URI</b>	<ul style="list-style-type: none"><li>• <b>URI (Universal Resource Identifier)</b></li><li>• Addressing of any files and data sources on the Internet.</li></ul>

1. Figure: WWW- 3 Specification by Tim Berners-Lee

Besides the 3 above-mentioned main elements of the World Wide Web, Tim Berners-Lee also developed the first web server (httpd) and the first web browser (WorldWideWeb, later: Nexus) for the operating system NeXTStep. [3]

The content could now be displayed with a user interface that could be operated very simple by a mouse. His first website can still be accessed today: <http://info.cern.ch>.

Today, he is director of the World Wide Web Consortium (W3C), a web standards organization founded in 1994. At the same time, he is director of the Web Foundation, which has been working for a free and open web for everybody since 2009. The foundation's mission is to promote the Web as a public good and a fundamental right. [4]

## **2.2 The History of Browser Wars**

Tim Berners-Lee saw a great demand for his technology outside CERN as well and decided, instead of patenting the software and securing the rights to the technology, that he will share his innovation with the public. Therefore, CERN published the software code to the world on April 30, 1993. [5] Since Tim's browser was only compatible on certain devices, it did not take long until the first companies recognized the Web's potential and used it as the basis for their businesses.

### **2.2.1 The First Browser War (1995-2001)**

In the 90's it was not like today that you have a different choice of browsers and can just download them on the Internet and use them for free. At that time, Netscape Navigator was the most widely used browser. The Netscape Navigator was founded by Marc Andreessen on April 4, 1994. The basis of this Navigator was the browser "NCSA Mosaic", which was the first browser that could also display graphic elements on web pages. The Netscape Navigator spread very quickly due to its speed, new features, improved usability, and lack of competition. Netscape became the market leader and had a market share of 80 percent in 1995. [6] Microsoft, who initially underestimated the browser market, became aware of the potential and decided to release the first version of its browser called Internet Explorer in August 1995. This led to a heated war between Netscape and Microsoft. Since Microsoft was financially better positioned than Netscape, they invested a lot of money in the development and marketing of their browser. In addition, Microsoft distributed the Internet Explorer for free, while users had to pay for the Netscape Navigator. Furthermore, Microsoft made another ingenious move and integrated their own browser into their own



operating system Microsoft Windows and has managed to get more and more people to use their browser. Since Microsoft dominated the operating system market at that time, every windows computer had the Internet explorer installed and could easily be used. [7] Users didn't have to waste time installing a browser anymore and people who owned a computer for the first time, they have not had the opportunity to try a comparable browser. As a result, Internet Explorer's market share increased rapidly up to 90% and Microsoft's browser dominated the market by the end of the 1990s. However, Microsoft was accused because of bundling their products and exploiting their monopoly power through a US antitrust case and had to pay very high fines. Microsoft lost the court case but won the war and managed to push the Netscape Navigator out of the market. [8] This resulted that the Netscape's browser had become almost irrelevant, the source code of the latest version was released under the name "Mozilla" as open source to the public in March 1998. Based on this foundation, the Firefox browser was developed with the slogan "internet for people and not for profit" under the codename "Phoenix" in order to provide Internet users a different choice in 2002. [9]

### **2.2.2 The Second Browser War (2004-2017)**

Due to their monopoly status and no competition, Microsoft had hardly worked on the further development of their browser. There were hardly any new updates and new features. It didn't take long before new companies started to develop new browsers and gain market share. [10] This is how it turned out. The successor of the Netscape Navigator, Firefox came with new features such as a search bar, enabled multi-tab browsing, a pop-up blocker, a phishing filter, spell checking, support web standards, full support for PNG and so on in 2004. The Internet explorer, which was the most widely used at the time, struggled with security vulnerabilities and fell victim to spyware and malware. On the other side Firefox was much more secure, and had many new features, therefore they managed to generate 60 million downloads in 9 months. Of course, Microsoft also tried to come up with new features, but it was far too late, as most people were already happy with using Firefox. Other browsers also became more relevant, such as Opera and Apple with its own Safari browser. Finally, Google also launched its browser Google Chrome in 2008. Google was already a very successful search engine at that time. Chrome managed to become the most used browser due its new features, speed, security and clarity since 2012. Even

today, Chrome is considered one of the most secure browsers and therefore is still the most widely used. [11] [12]

## **2.3 W3C and WHATWG**

### **2.3.1 The W3C**

W3C stands for World Wide Web Consortium and was founded by the World Wide Web inventor Tim Berners-Lee on October 1, 1994. Since Tim Berners-Lee has realized through the disputes of the different browsers that his invention went in a wrong direction and it did not follow his vision, he has decided to found this organization to define technical specifications and guidelines for the World Wide Web. [13] The mission of the W3C includes the development of protocols and guidelines for the long-term growth of the World Wide Web. The W3C follows the principles “Web for All” and “Web on everything”. It enables communication, commerce, and knowledge exchange between people, and these features should become available to everyone, regardless of hardware, software, language, or culture. Furthermore, the W3C is especially committed to people with disabilities and constantly tries to find solutions and makes the web accessible. It also deals with topics such as web interaction and web trust especially about security and privacy on the web. [14]

The W3C is not an intergovernmental recognized body and therefore it cannot define ISO standards. Nevertheless, standards that are developed and defined by the W3C are qualified to build the foundation for ISO standards like in the case of XML. Therefore, the organization calls its standards “W3C Recommendations”. Moreover, there are lots of well-known technologies that have been standardized by the W3C include: HTML, XML, CSS, and SVG. Every web designer is confronted with these recommendations on a daily basis. [15]

However, not everyone has always been satisfied with W3C's work and new conflicts arose again in the history timeline of HTML.

### **2.3.2 History Timeline of HTML**

HTML made very rapid progress in the first few years, especially the first specifications were defined very quickly one after the other. It didn't take long before web designers had several specifications with which they could create their web pages. As already mentioned in the history of the WWW, HTML was published

officially for the first time in 1992. At that time, it was only designed for texts and it was only possible to display bold and italic texts. A few years later, the next specification, HTML 2.0, was published with new features such as forms in 1995. With HTML 3.2 features like tables, applets, text flow around images, superscripts and subscripts were added in 1997. A few months later the same year, the next specification HTML 4.0, was published. This W3C recommendation supported a lot of new features such as more multimedia options, scripting languages, style sheets and documents that are more accessible for users with disabilities. In 1999 the next specification HTML 4.01 was published in which a few things from the predecessor were reworked. Until HTML 4.01, all specifications were based on SGML. [16]

At that time, the Extensible Markup Language (XML) was becoming increasingly popular among web developers and more and more companies decided to use XML as the basis for their file formats. As a result, W3C decided to reformulate HTML in XML and published the first version of XHTML in 2000. At first, there weren't many differences. XHTML 1.0 was actually the same as HTML 4.01, but as a web developer you had to pay attention to the syntax rules of XML. XHTML 1.0 was a strict and better way of writing HTML. A big disadvantage of HTML was that no error messages were displayed. The majority of the web developers did not pay attention to the capitalization, tags were not closed properly, and a just a bunch of developers specified attributes with quotation marks. Everyone simply coded without structure and rules. The main thing was that it was displayed in the web browser. As a result, many developers had difficulties reading HTML codes and finding solutions to the errors. XHTML was introduced to remove such errors. With the new specification and by reformulating HTML into XHTML, things had changed. [17] [18]

In HTML 4.01 and also XHTML 1.0, web developers had to classify the documents with the three different Document Type Definitions (DTD) to prevent syntactical errors and make sure that the document will be parsed the same way by different browsers [19]:

- Strict: it contains all HTML attributes and elements, but doesn't include presentational or deprecated elements.
- Transitional: it contains all HTML attributes and elements, including presentational and deprecated elements.
- Frameset: same as transitional, but allows the use of frameset content.

Web developers had to follow rules, such as all tags must be closed, tags must always be properly nested, tag names and attributes must always be in lower case, and so on. This generates a cleaner and understandable code. XHTML 1.0 was perceived by almost everyone as a largely good thing and provided a solid foundation for building web pages. Shortly thereafter, XHTML 1.1 was released with very minor differences. With XHTML 1.1, related and associated elements and attributes were grouped into so-called modules. For example, the forms module contains all forms of elements. [20]

The W3C continued to focus only on the further development of XHTML and decided to redevelop the XHTML 1.1 and develop a completely new specification. XHTML 2 was an ambitious attempt to correct some of the birth defects of HTML. However, although the specification represents a continuous evolution of the HTML 4, XHTML 1.0 and XHTML 1.1 standards, it was not backward compatible with those standards. A major problem was that the new XHTML 2.0 specification was defined very strictly according to the XML framework and many web pages that are built on XHTML 1.1 could not be converted to XHTML 2.0. This step would have made it impossible to display 99.9% of all real websites, as it is estimated today. The W3C had also planned with the new specification XHTML 2.0 to abolish many known HTML elements and added new ones which should change the structure of the documents. [21] However, the W3C did not consider that this radical change would not please all participants. Especially browser vendors criticized the drafts and were not very happy about the further development of the web.

At a W3C workshop, the web browser vendors Mozilla and Opera submitted a proposal that dealt with the further development of HTML 4 in 2004. The W3C rejected the proposal and argued that the standard should continue to be based on XML. From this moment on, the browser vendors decided to take the further development of HTML into their own hands and founded the Web Hypertext Application Technology Working Group (WHATWG). This is how HTML5 became a competitor to XHTML 2.0 and provoked a conflict. [22]

### **2.3.3 The W3C vs. WHATWG Conflict**

The Web Hypertext Application Technology Working Group (WHATWG) is a working group which tries to evolve the web through new technologies which should make it easier for web developers to create internet applications.

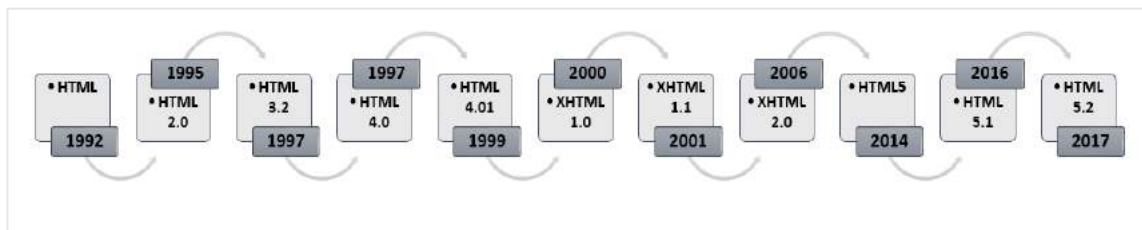
According to some browser vendors, the slow development of web standards, following the rules of XML and lack interest of HTML by the W3C was the cause for the foundation of the organization. Therefore, Mozilla, Opera and Apple teamed up together and founded the WHATWG in 2004. The organisation started working on its own specification based on HTML 4.0 and XHTML 1.0. It did not take long before the first specification was published under the name Web Applications 1.0 and later HTML5 which should be a rival version to XHTML 2.0. [23]

Of course, the different specifications from the two organisations caused conflicts in the history of HTML. The conflict between these two had played an essential role in the development history of HTML5. At that time, HTML5 was already at least partially supported by some browsers and XHTML 2.0 support does not exist in any browser. The W3C had no other choice and also decided to form a group to work on the further development of HTML5. However, they copied the existing specification from the WHATWG and developed it independently. There were two working groups within the W3C, which pursued different goals. One working group worked on XHTML 2.0 and the other one on HTML5. It did not take long and the W3C announced in 2009 that they will stop working on XHTML 2.0 and will only concentrate on the further development of HTML5. HTML5 had won the battle. So both organizations started to work on their specifications in 2009.

The two organizations had worked together on the further development of HTML5 until 2011. However, it then came to disputes again because they pursued different goals. The two organisations worked differently on their specifications of the web. The WHATWG works according to the motto "Living Standard". They do not have a specific version number and are constantly working on the further development of HTML. This means that new features are constantly being added and some removed. On the other hand, the W3C works with version numbers. This means that they develop a new recommendation and define it as a new version, for example HTML 5.1 and so on. The procedure for a recommendation to be published by the W3C takes a lot of time. The idea must go through several phases. It starts with an editor draft, then it continues with a working draft, then there is a last call working draft, then it comes to a candidate recommendation, afterwards it comes to a proposed recommendation and finally it is published as a W3C recommendation. This is why the development of the HTML5 recommendation by the W3C took years to be published. The first working draft was made in 2008 and it took until 2014 before the W3C recommendation was finally published.

On 28 October 2014, HTML5 was published as a W3C Recommendation. After that, the new improvements went a bit faster and new versions came very quickly. Two years later came the first update with the next W3C Recommendation HTML 5.1 on November 1st 2016. Finally, on December 14th 2017, the latest version HTML 5.2 was released with a lot of new features and new APIs. [24]

As the two organisations did not want to continue to divide the web, they decided to make peace. In addition, the two organizations have also signed an agreement to work together on the development of a single version of the HTML and DOM specifications in 2019. It was decided that the WHATWG maintains the HTML and DOM Living Standards, and the W3C adopts certain review drafts from the WHATWG and publishes them as W3C Recommendations. May 28th 2019 was a very important day in the history of the web because from that day on there has been only one HTML standard. [25] [26]



2. Figure: History Timeline from HTML to HTML 5.2 (W3C)

### 3. HTML5 Elements

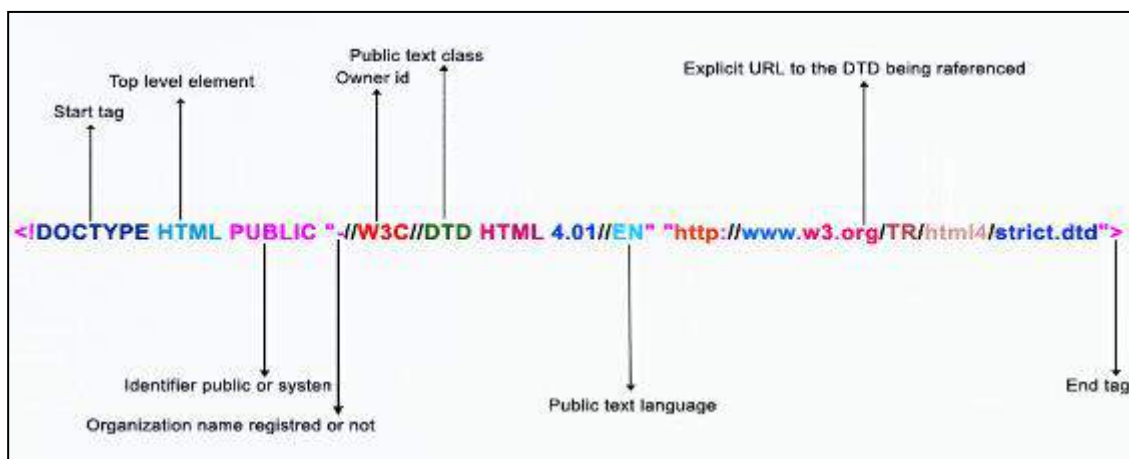
HTML5 has made a huge impact on web development and brought new possibilities and opportunities. It simplifies surfing as we know it today. Because now the focus is more on interaction, for example in the form of contact forms, social media comments or online games. In addition, external plug-ins were still necessary in HTML4.01. HTML5 now has new tags, for example, specified mark-up options, multimedia elements or various programming interfaces. It also makes it easier to structure web pages compared to earlier versions. The following chapters will give an overview of HTML5 and what exactly has changed from the predecessors HTML 4.01 and XHTML 1.0. [27]

### 3.1 DOCTYPE

An HTML document describes the structure, content and formatting of a digital document. HTML documents are the foundation of the World Wide Web and are displayed by web browsers. Each HTML document is stored in a separate file that can be edited with a text editor.

You are probably wondering what version of HTML a website use. The easiest way to find out is to look at the code of the website in your browser. The document type is specified in the first line of every HTML file. This tells a web browser according to which set of rules the document has been created. These rules are defined with the DTD (Document Type Definition) and tells the browser what type of document it is. The browser needs this information to be able to display the document correctly. In HTML5, the syntax of Doctypes has been extremely simplified compared to its predecessors. As already briefly described in chapter 2.3.2, it was necessary to specify Doctypes for HTML 4.01 and XHTML 1.0. Web developers had to classify the documents with the three different Document Type Definitions (DTD) to prevent syntactical errors and make sure that the document will be parsed the same way by different browsers. These Doctypes declarations were very long and complicated. Since HTML4 and also previous versions are based on SGML, the correct DTDs (Document Type Definition) must be specified in the Doctype. SGML stands for Standard Generalised Markup Language and refers to a now outdated predecessor of HTML. [28] Since the introduction of HTML5, Doctypes are no longer classified as SGML because of compatibility problems. HTML5 is downward compatible, every browser can handle it and documents stay valid. [29] [30]

#### Example of a DOCTYPE (HTML 4.01 - strict) before HTML5:



3. Figure: DOCTYPE Syntax of HTML 4.01 [76]

As you can see from the illustration, the syntax of the Doctype was very long and complicated and has to be referenced to a DTD. Therefore, developers always had to specify the Doctype with the following elements:

- *"DOCTYPE HTML PUBLIC"* means that the document refers to the public available HTML DTD.
- *"-//W3C//DTD HTML 4.01//EN"* identifies the W3C as the authoritative publisher of the document type "HTML" used in version 4.01 in the document.
- *"EN"* is the language specification (English) and does not refer to the content of the document, but to the notation of HTML elements and attributes.
- The URL *"http://www.w3.org/TR/html4/strict.dtd"* allows the user agent (browser) to load the DTD and the rules and elements it contains when they are needed.
- *"Strict"* precludes the document's rendering from using elements and attributes that the W3C considers being the responsibility of stylesheets.

With the introduction of HTML5, the syntax of Doctypes became extremely simple, short and easy to remember. You also have no longer to pay attention to capitalization. The Doctype declaration in HTML5 is defined as:

- **<!DOCTYPE HTML>**

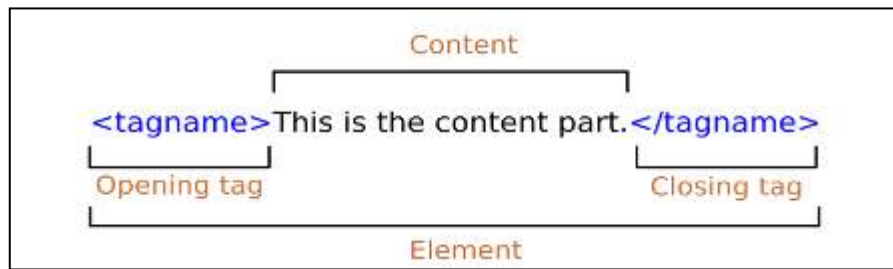
Many people confuse Doctypes with HTML elements. The Doctype is not an HTML element, it only gives the browser information about the type of document. [31]

## **3.2 New Elements in HTML5**

### **3.2.1 Basic Information on HTML Elements**

Before you create a website, you should know what an HTML element is. HTML elements are the building blocks from which a website or web application is created and structured. Almost every HTML element consists of an opening (<>) and a closing tag (</>). The HTML elements and HTML tags are often confusing. The tags are used to open and close the object, while the element contains both tags and their content. The following illustration shows how HTML elements are structured.





4. Figure: HTML Element Structure [77]

#### Examples of elements in HTML5:

- `<h1> Seminar Paper </h1>` // Start tag – Content - End tag
- `<p> History of HTML </p>` // > this tag is used for paragraph

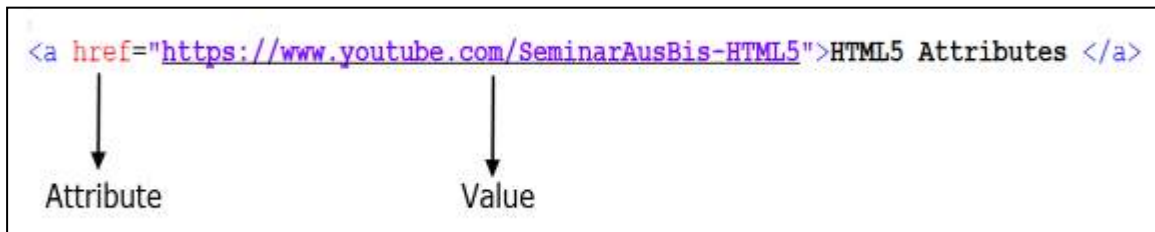
HTML elements define the content located between the "opening" and the "closing" tag. As can be seen in the illustration above, the closing tag is distinguished from the opening tag by a slash after the opening angle bracket. [32] The browser does not display HTML tags when displaying the web page. It only uses them to display the content correctly. As already mentioned, the HTML element structure with an opening and a closing tag does not apply to all HTML elements. The exceptions are the so-called self-closing tags. These are HTML elements that do not transport any content, but this does not mean that these elements cannot contain any information. In contrast to XHTML, the use of the slash in HTML5 is optional.

#### Examples of self-closing tags in HTML5:

- `<br>` - this tag forces a line break at the set position.
- `<hr>` - this tag creates a horizontal line to separate thematically different topics
- `<img>` - this tag inserts an image on the place where it is set.

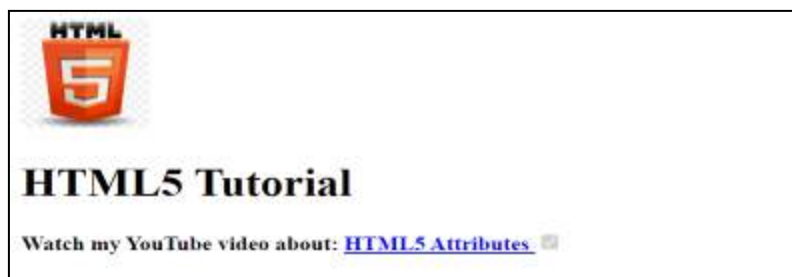
Furthermore, there are also HTML attributes that are used to provide additional information about the HTML elements. The attribute has a name that is placed before the equal sign (=) and the value. HTML5 removes the requirement to enclose the value of the attribute in inverted commas if the value is a string without spaces and not an enumeration. For example, we would like to refer a

visitor from our website to a YouTube video. We can do this very easily with the help of HTML attributes. [33]



5. Figure: Example of a HTML Attribute

In our case we use the `<a>` tag which defines a hyperlink and with the href attribute we specify the link from a YouTube video. If we save the HTML file and run it in the browser, our page will look like the picture below. Now, if visitors would click on the hyperlink "HTML5 Attributes", they would automatically be directed to the YouTube video.



6. Figure: Browser output HTML Attribute

Furthermore, HTML tags are not case sensitive, but it is recommended by W3C to write the tags in lower case. `<h1>` for example is the same as `<H1>`. In XHTML, it was mandatory to pay attention to the upper and lower case. Developers had to write the tags in the lower case. In HTML5, it is up to the developer.

### 3.2.2 Semantic in HTML Documents

One of the most important features of HTML5 is the ability to put content into a semantic context. The demands of HTML5 are to be expressive and to allow the developers to express very clearly how the document is structured. This is the reason why there are new semantic tags in HTML5 that should serve this purpose. The web as Tim Berners-Lee had invented was rather to provide people with information. It didn't really have much to do with the look of the website. But over the years, more and more people use the web and designed layouts became

more and more important. The only way to structure an HTML document or define layout areas under HTML 4.01 or XHTML 1.0 was to use `<div></div>` blocks. The `<div>` is a block element that can be used to combine other elements such as paragraphs, lists and graphics into a common area (div = division = area). In other words: div is a container or a box, in which several other boxes can be stored, for example, in order to be able to design them together later using CSS. A `<div>` element does not have any semantics. One way to integrate semantics into this element is to specify an id or class attribute. With a large number of div elements, it is difficult to keep an overview, especially with closing tags. Therefore, it is recommended to use the new HTML5 elements and structure your website without `<div>` tags. This way, you save unnecessary classes and id selectors.

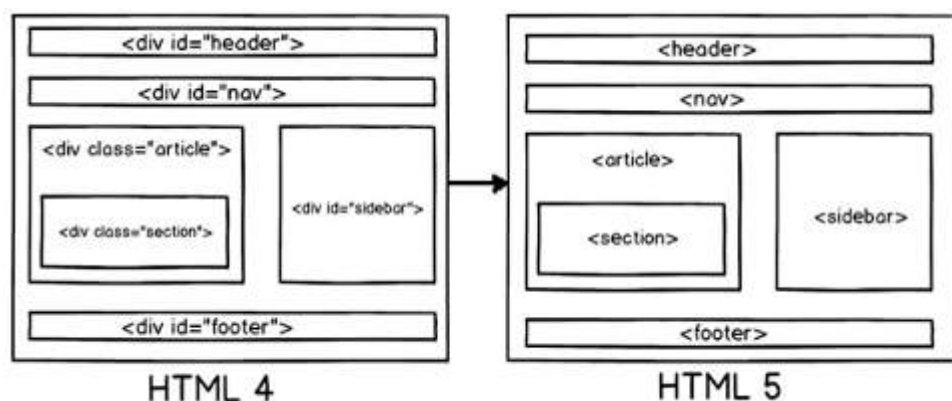
A semantic element clearly describes its meaning to both the browser and the developer. Another important aspect of semantically correct HTML code is that search engines read the source code and determine the importance of content based on the HTML tags used. This has a strong impact on the ranking of a website and increases the chances of a better search result. Moreover, semantically correct HTML code plays a major role in the accessibility of websites. Screen readers are able to read the source code and read it out to visually impaired people. If the HTML code is incorrectly marked up, it makes it unnecessarily difficult for disabled people to access the website. [34]

Examples of non-semantic elements:

- `<div>` and `<span>` - tells nothing about its content.

Examples of semantic elements:

- `<header>` , `<footer>` and `<section>` - clearly defines its content.



7. Figure: Non-Semantic vs Semantic Page Structure [78]

### 3.2.3 Structural Elements

The time when web developers had to define many `<div>` tags with different CSS classes is now over. Now there are new HTML5 elements which with developers can define areas and bring structure to a page. Semantics play a central role in this context. In HTML5, new semantic elements have appeared that can be used to structure, group and mark up text content. The table below shows an overview of the new structural elements in HTML5.

Elements	Description
<b><code>&lt;header&gt;</code></b>	Header of a page or section, this includes for example logo, search bar, welcome text
<b><code>&lt;nav&gt;</code></b>	Main navigation on a website.
<b><code>&lt;section&gt;</code></b>	Specific area in a document with a separate header and footer - similar to <code>&lt;div&gt;</code>
<b><code>&lt;article&gt;</code></b>	Defines a section or element as independent content within an HTML5 document. (blog post, article)
<b><code>&lt;aside&gt;</code></b>	Defines a sidebar. It can be useful to present more information to the reader and thus encourage them to stay on the page.
<b><code>&lt;footer&gt;</code></b>	Footer area of a website, or section. It includes information such as copyrights.

Table 1: New Structural Elements in HTML5 [35]

Of course, the whole thing can be better illustrated by an example so that you can see the difference better. On the illustrations below, you can see a page structured with the `<div>` element and a page structured with the new semantic HTML5 element.

#### HTML `<div>` Based Layout (non-semantic way):

```
<div id= "header"><h1> HTML5 Tutorials (BIS) </h1 > </div >
<div id= "navigation">
  <ul >
    <li ><a href= "page1.html"> Home </a> </li >
    <li ><a href= "page2.html"> Videos </a> </li >
  </ul >
</div >
```

8. Figure: Example for `<div>` element

## Using the HTML5 Structural Elements (semantic way):

```
<header><h1> HTML5 Tutorials (BIS) </h1 > </header>
<nav>
  <ul >
    <li ><a href= "page1.html"> Home </a> </li >
    <li ><a href= "page2.html"> Videos </a> </li >
  </ul >
</nav>
```

9. Figure: Example for Structural Element

If you compare the two code blocks with each other, you will quickly realise that the semantic code block is much easier to read. Now this is a small example where I haven't used many of these elements, but imagine you want to create a website. It has several hundreds of lines of code then it quickly becomes confusing with `<div>` block. As mentioned above, semantic elements are easier to understand.

As accessibility plays an important role in our era, semantic elements make the context of a web page much easier to understand by search engines and screen readers and thus improve usability. Furthermore, many web developers have also declared different `<div>` code blocks according to their preferences. For example, there were many ways to declare a header:

- `<div id="header">`
- `<div class="head">`
- `<div class="header">`
- or simply `<div>`.

By introducing unified semantic elements, it is now much easier for people to read, understand and use code blocks. [36]

### 3.2.4 Text Elements

In HTML5, new text elements have also been added to improve the semantics and structure of a web page. They are intended for the flow content, which is mostly used in the body of an HTML document.

Elements	Description
<b>&lt;time&gt;</b>	The <time> element indicates the time in 24-hour format and/or the exact date.
<b>&lt;bdi&gt;</b>	The <bdi> element (bidirectional isolation) isolates an excerpt of text that should appear in a different direction from the text surrounding it.
<b>&lt;wbr&gt;</b>	The <wbr> element (word break) marks places within a word where a browser may break the text. Used for better readability.
<b>&lt;ruby&gt;</b>	The <ruby> element is used to integrate ruby annotations for special characters with the indication of the pronunciation into the HTML code. Mostly used for Asian characters
<b>&lt;mark&gt;</b>	The <mark> element defines a text that should be highlighted.
<b>&lt;data&gt;</b>	The HTML <data> element links a given content to a machine-readable translation. If the content is time or date related, the HTML <time> element must be used.

Table 2: New HTML5 Text Elements

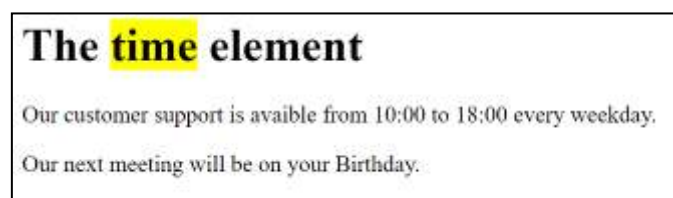
The table above lists the latest HTML5 text elements and their descriptions. Especially the element <time> is very important for many developers. This element allows times and dates to be presented in a machine-readable format and makes it readable for computers and search engines. This can be useful for user agents that provide any type of scheduling for the user's calendar. Like all other HTML elements, this element also supports attributes too. The task of this attribute is to translate the time into a machine-readable format. In order to make it a little more understandable, see below the code block and the output.

```

5 <h1>The <mark>time</mark> element</h1>
6
7 <p>Our customer support is available from <time>10:00</time> to <time>18:00</time> every weekday.</p>
8
9 <p>Our next meeting will be on your <time datetime="2021-10-01 12:00">Birthday</time>.</p>

```

10. Figure: Example of a <time> element:



11. Figure: Output of the <time> element

This is a good illustration of what is meant by the attribute `datetime`. Even if there is no exact time in the output, the computer still knows exactly what time and date of the "Birthday" is. Furthermore, I also used the `<mark>` element to show how it is used. With the `<mark>` element words can be highlighted. In this case I used it for the word "time". [37]

### 3.2.5 Form Elements

In this day and age, no matter what we do on the Internet, we very rarely get away with signing up to a website, or logging into a web application, and filling out contact forms. The interaction between users and the website or web applications is currently becoming stronger than ever through social media platforms, online shopping, and streaming services. Forms already existed before HTML5. However, they were not very user-friendly and were very limited in their functionality, and were only used to type in and query data. HTML4 provides only general form elements, such as text input and password fields, checkboxes or dropdown lists. If you expect a date to be entered in a text field, for example, you are usually forced to verify on the server side whether the user input is a valid date at all. The situation is similar for URLs, numeric values or other inputs. Besides, nowadays we live in a fast digital world. We usually use our smartphones a lot to surf the internet and use web applications. Therefore, with the introduction of HTML5, new form elements were introduced to make it easier to create forms and make use of them. Compared to HTML4 and XHTML1.0, HTML5 can display extended form elements. For example, the progress of a task can be displayed (`<progress>`) or completion suggestions can be given (`<datalist>`). The list below shows all new form elements of HTML5. [38]

Elements	Description
<b>&lt;datalist&gt;</b>	The <code>&lt;datalist&gt;</code> element is used to define an input field with predefined options for an <code>&lt;input&gt;</code> element – like a dropdown list
<b>&lt;output&gt;</b>	The <code>&lt;output&gt;</code> element is used to show the result of calculations by a script
<b>&lt;progress&gt;</b>	The <code>&lt;progress&gt;</code> element is used to visualize the progress of a task. For example a Progress bar. For example Download
<b>&lt;meter&gt;</b>	The <code>&lt;meter&gt;</code> element is used to visualize a value in a defined range with graphical display for displaying measured values. For example, the number of comments on a query

<b>&lt;keygen&gt;</b>	The <keygen> element generates a private and public security key. <b><i>This element is not supported any more since HTML 5.2.</i></b>
-----------------------	--

Table 3: HTML5 - New Form Elements

The <datalist> element makes it possible to display a selection with options under a text field. The user has the possibility to enter a text or to select an entry from the data list (also called combo box). Let's take a closer look with an example. [39]

12. Figure: Output <datalist> Example

This example describes how you can support the input in a text field by suggestions from a list. The user can accept one of these four suggestions such as "Business Economics", but also enter any other text. Let's take a closer look at the code.

```

12 <form action="">
13   Subjects: <input type="text" list="Subjects">
14   <datalist id="Subjects">
15     <option>Business Economics</option>
16     <option>Business Informatics</option>
17     <option>Macroeconomics</option>
18     <option>Mathematical Economics </option>
19   </datalist>
20 </form>

```

13. Figure: <datalist> Example

The HTML <datalist> tag specifies set of options for the <input> element. It is basically used in conjunction with the <input> element to display some kind of selection list from several given values. The <option> elements within the <datalist> element define the individual list entries. The list attribute of the <input> tag must match the identifier (ID) of the <datalist> element.



Furthermore, another form element called `<progress>` was released as part of HTML5. The HTML `<progress>` element is used to visualize the progress of a task. Progress bars can be found on many blogs in many different variations: as a progress bar for actions, as a graphical countdown to some event, or as visible progress when filling out a form.

A progress bar is very easy to create in HTML5. Let's take a quick look at an example. I will now create a progress bar that can be used as feedback when a user is filling out a long form on a quizz.

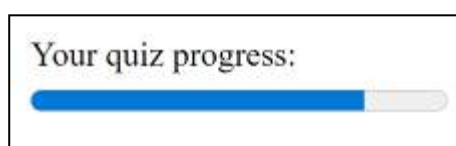
```
22 <progress max="100" value="80">Progress on a quiz: 80%</progress>  
23
```

14. Figure: `<progress>` Example

As you can see the Progress bar is very easy to define. You only have to pay attention to the two attributes that belong to this element.

- "max" indicates the totality of steps or tasks that are necessary to reach the goal. This can be, for example, the number of question in a survey. The only important thing is that the number has to be greater than 0.
- "value" indicates how many of these steps, tasks or percentages have already been achieved. This number must be less than or equal to max.

In addition to these two mandatory details, you should not forget the labeling, which is strongly advised for accessibility reasons. The output from our example above would look like this in a browser. [40]



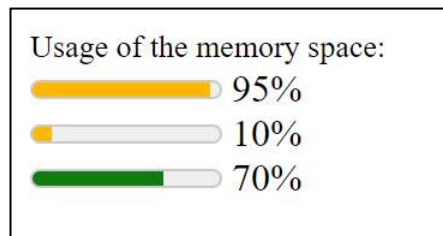
15. Figure: Output `<progress>` Example

Often people like to confuse the `<progress>` element with the `<meter>` element, because both visualize progress indicators. However, there is a clear difference here between these two introduced HTML5 elements. The `<meter>` element stands for measured values between a minimum and a maximum. The element can show the user for example the usage of the memory space.

```
<meter value="75" min="0" max="100" high="90" low="10" optimum="60" >100 of 128 GB used</meter>
```

16. Figure: `<meter>` Example

In contrast to the `<progress>` element, the `<meter>` element has many more attributes than “value” and “max” that you can use. Here you can divide the areas of the progress bar with attributes like “low”, “high” and “optimum”. Depending on the position of the optimum, the color of the bar will change. See Output below. [41]



17. Figure: Output `<meter>` Example

As mentioned before, web applications are becoming more and more popular and people tend to use their smartphones and tablets for surfing the internet. Therefore, websites must be presented in a correspondingly user-friendly way. In XHTML and HTML4, the `<input>` element accepted just three different values for the type attribute. These type attributes were “text”, “hidden”, and “password”. Furthermore, new input types have been added too. These are email, phone number, URL, search box, slider, numeric input, date and time selection, and color selection. This diverse selection of input types allows you to create better forms that can accept more content types, with built-in validation to make sure they are correct. In HTML5 a lot of importance was put on new `<input>` types and attributes. Therefore, a lot of new attributes and types were added. In the two following lists below you can find all new attributes and types. [42] [43]

Attributes	Description
<b>autocomplete</b>	Specifies for enabling or disabling autocomplete in <code>&lt;input&gt;</code> element.
<b>autofocus</b>	The autofocus attribute indicates an input field that should get the focus when the page is loaded.
<b>form</b>	The form attribute from HTML5 connects input fields to one or more forms.
<b>formaction</b>	Specifies the URL of the file which should be called when the form is submitted.
<b>formenctype</b>	Specifies how the form data should be encoded.

<b>formmethod</b>	Defines which HTTP method should be used.
<b>formnovalidate</b>	Defines that the input element should not be validated.
<b>formtarget</b>	Specifies where to display the file specified in the "formaction".
<b>height and width</b>	Specifies the height of the <input> element. Specifies the width of the <input> element.
<b>list</b>	The list attribute for input fields refers to a list of given values.
<b>min, max and step</b>	Min, max and step are controls for entering numbers. They define the minimum / maximum (e.g. from -10 to 10) and the step size (e.g. 1 or 0.1).
<b>pattern (regexp)</b>	The pattern attribute of an input tag can restrict input by regular expressions to certain input patterns.
<b>placeholder</b>	This attribute can be used to display a placeholder.
<b>required</b>	Indicates that an input field must be filled before the form is submitted
<b>multiple</b>	Indicates that a user can enter multiple values.

Table 4: HTML5 new <input> Attributes

Elements	Description
<b>email</b>	A field to enter your own e-mail address or a list of e-mail addresses.
<b>number</b>	Used for numeric inputs
<b>url</b>	To enter a URL.
<b>tel</b>	To enter a phone number.
<b>color</b>	To select a color via a corresponding control element
<b>range</b>	Used for numeric inputs, in contrast to number, the actual value is not important.
<b>search</b>	Used to create a search field
<b>date</b>	To enter a date without time zone.
<b>datetime</b>	To enter the date and time based on the UTC time zone.
<b>time</b>	To enter a time with hour, minute, seconds and fractional seconds, but without time zone.
<b>datetime-local</b>	To enter date and time without time zone.
<b>week</b>	To enter a date consisting of a year and a week number, but without time zone.

<b>month</b>	To enter a date with year and month, but without time zone.
--------------	---

Table 5: New `<input>` Types [44]

Let's take a look at the new `<input>` tags with an example.

```

7 <p>Please enter your E-mail: <input type="email"></p>
8 <p>Range: <input type="range" min="-10" max="10" />
9 <p>Number: <input type="number" step="1" />
10 <p>Select a color: <input type="color">
11 <p>Search: <input type="search">
12 <p>URL: <input type="url" name="website" placeholder="https://www.google.at" pattern="https://.*" required>
13 <input type="submit" value="Send">

```

18. Figure: `<input>` Tag Example

19. Figure: Output `<input>` Tag

### 3.2.6 Embedded content

Without HTML ("Hypertext Markup Language"), the Internet would be different. The markup language structures texts, images or links and thus forms the basis of the World Wide Web. However, the Web has changed over time. Multimedia has become an important factor for web designers. Audio and video content as well as mobile displayability of content are becoming more and more important. Before HTML5, plug-ins such as Adobe Flashplayer or Microsoft Silverlight had to be used to integrate audio and video content. Plug-ins are software modules that have a specific functionality, such as playing audio or video content. Plug-ins requires a lot of power and consumes too much of battery life on mobile devices and is considered unsafe. Therefore, HTML5 has introduced many new features for embedded content which make it possible to include and control media content

without plug-ins. In the following list you can see new HTML5 tags that you can use to embed multimedia files.

Elements	Description
<b>&lt;audio&gt;</b>	With the <audio> element, audio files can be played on web pages.
<b>&lt;video&gt;</b>	With the <video> element, video files can be played on web pages.
<b>&lt;track&gt;</b>	You can use <track> to indicate external media tracks such as subtitles or translations.
<b>&lt;source&gt;</b>	With <source> you can specify alternative media resources for files.
<b>&lt;canvas&gt;</b>	Container for interactive graphics, in which you embed animations and other visual elements that you have created with JavaScript.
<b>&lt;embed&gt;</b>	Embedding external content by using plug-ins.
<b>&lt;svg&gt;</b>	Embedding an SVG graphic (two-dimensional vector graphic).
<b>&lt;mathml&gt;</b>	Direct embedding of mathematical formulas
<b>&lt;figure&gt;</b>	The element <figure> is a container for images, graphics and other figures (e.g. tables) that can be used instead of the generic div tag.
<b>&lt;figcaption&gt;</b>	The element <figcaption> is the container for a caption or legend of the figures.

Table 6: New HTML5 embedded content

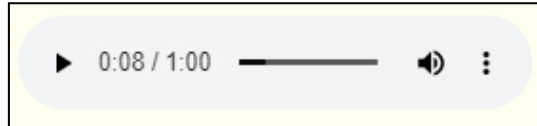
Especially the embedding of audios and videos have made a lot of things easier. Let's take a look at them in more detail. The <audio> element can be used to play audio files like music, podcasts or sounds without plugins or add-ons on web pages. The source of the audio file can be specified with the "src" attribute in the start tag of the "audio" element or with "source" elements between the start and end tags of the "audio" element. The advantage is that multiple sources can be specified this way. In addition, free text can also be used within the "audio" element, which is displayed if the browser cannot play any of the audio files. There are also some attributes that can be used such as the "autoplay" attribute to start an audio file automatically, the "controls" attribute to display the browser's own player with its controls or the "loop" attribute to specify that the audio file should be played over and over again. [45]

```

6 <audio controls>
7   <source src="/BillieJean.m4a" type="audio/x-aac" />
8   <source src="/BillieJean.mp3" type="audio/mpeg" />
9 </audio>

```

20. Figure: <audio> Example



21. Figure: Output <audio>

In the early days of HTML5, browsers supported different video formats: Firefox plays only OGV, Chrome plays OGV, WebM, and MP4, Safari and Internet Explorer 9 natively play only MP4 video. The video tag simplified the handling of the different video formats and codecs. Between the opening and closing video tag, alternative formats of the video can be used. The type of the video - e.g. type="video/mp4" or type="video/ogg" - should always be listed, because otherwise the browser would have to load all videos to determine which video it can play. Today, all modern browsers play the MP4 format and a single video tag is basically enough. Again, as with the "audio" element, the source can be specified with the "src" attribute in the start tag or with the "source" element between the start and end tags of the "video" element. Also for the "video" element, a free text can be specified between the start and end tag to be displayed when the video file cannot be played. [46]

```

3 <video controls height="200" width="350">
4   <source src="Michael Jackson.mp4" type="video/mp4" >
5   <source src="Michael Jackson.webm" type="video/webm;codecs="vp8, vorbis" >
6   <source src="Michael Jackson.ogv" type="video/ogg" >
7   <p>Your Browser do not support this video format!</p>
8 </video>

```

22. Figure: <video> Example

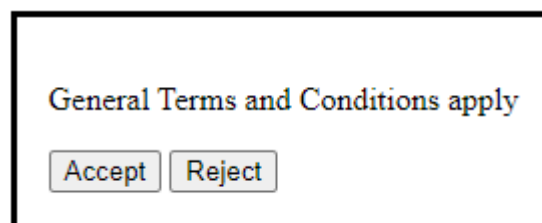
Besides audio and video elements, the canvas element was also introduced. It basically provides a 2D bitmap drawing area of freely definable size. JavaScript can be used to draw in it and create graphics on-the-fly. This makes it easy to generate diagrams dynamically, for example, or to produce mini-games with it [47] .

### 3.2.7 The new <dialog> Element of HTML 5.2

The new HTML 5.2 standard brings a new element called <dialog>. This can be used to create modal dialog boxes to ask a question, request an input, or offer multiple options. This element can be used very simple and with the help of JavaScript you can also make this element more interactive. Furthermore, the "open" attribute can be used to indicate that the dialog box is active and the user can interact with it. [48] To make it a little more understandable, let's look at the example below.

```
5 <dialog open="">
6   <p>General Terms and Conditions apply</p>
7   <form method="dialog">
8     <button type="submit" value="ja">Accept</button>
9     <button type="submit" value="nein">Reject</button>
10  </form>
11 </dialog>
```

23. Figure: <dialog> Example



24. Figure: Output <dialog> Element

Since this element was newly implemented, you have to be a bit careful by using this element since it is still not supported in all browsers.

## 4. APIs

### 4.1 What are APIs?

API stands for "Application Programming Interface". An API is basically an interface that connects software and hardware components, such as applications or user interfaces and allows them to communicate and exchange data with each other. The transfer of data and commands is structured according to a pre-defined syntax. [49] The API defines the form in which the information and data are received from the application module and back again. The API does not take

care of the actual program flow within the module. In contrast to a binary interface, the program connection in the API takes place purely at the source code level. [50] However, in the context of this paper, we are dealing with APIs that are used in many applications and in the web environment in form of Web APIs. With the introduction of HTML5, many new APIs were published by W3C and the WHATWG to make the life of programmers easier by allowing them to easily connect different applications with each other. In HTML5, there are a number of JavaScript objects that can be integrated into HTML5 documents as APIs. [51] The table below lists the most commonly used HTML5 APIs.

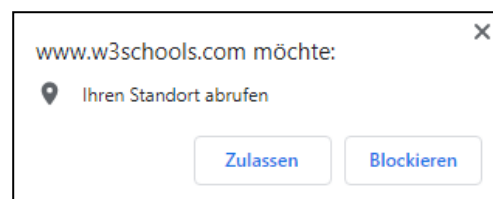
HTML5 APIs	Description
<b>Media API</b>	The "Media API" is the browser's implementation of JavaScript methods on HTML video and audio elements. It uses different methods in order to play or pause videos or audio elements. [52]
<b>TextTrack</b>	With the "TextTrack" you can include timed subtitles and captions for video and audio elements. [53]
<b>History API</b>	It provides useful methods and settings that allow to control the back button, forward button and add or remove URLs in the browser history. [54]
<b>Drag and Drop API</b>	This allows you to move objects within an HTML5 document or between two web pages. [55]
<b>Web Messaging or Cross-document messaging</b>	Allows messages to be sent and received between two documents in a client. If you want to exchange information between multiple browser windows of a web app or between frames, you should use this API. [56]
<b>Canvas API</b>	Used for 2D illustration with the new canvas element. [57]
<b>Web Storage API</b>	This API allows saving data in the browser cache for later use by web applications. [58]
<b>Geolocation API</b>	This API provides the user's geographic position and makes it available for web applications. [59]
<b>Offline-Applications API</b>	This API makes web applications available on HTML5 pages even when there is no connection to the Internet. This is relevant for mobile Internet use via smartphone. For this purpose, an application with all required content is loaded into the application cache. [60]
<b>Web Workers API</b>	This API enables the execution of scripts in the background to increase the performance of web applications, for example. [61]
<b>Web Socket API</b>	This API allows data to be exchanged between client and server in real time. Examples: Chat applications, streaming or online games. [62]



<b>File API</b>	This API allows access to files uploaded by forms. Using this API, files can be uploaded via drag-and-drop and a preview can be displayed. [63]
<b>Server-Sent Events (SSE) APIs</b>	This API allows a web page to receive updates from a server. [64]

Table 7: New HTML5 APIs

Especially the “Geolocation API” is nowadays relevant for numerous web applications. As already described in the table above, this API defines how to get the exact position of a user. Since we use our smartphones almost exclusively to surf the web, this API can be used to obtain very accurate information about the user behaviour through the Global Positioning System (GPS) of the smartphone. Therefore, it is important to think about data protection, since we are dealing with sensitive data. For this reason, users must be asked explicitly for permission of their location because of data privacy. [65]



25. Figure: Browser output - Geolocation Request

The example shows how a permission request for this API from a website looks like.

Of course, there are more APIs than the popular ones such as the Geolocation, Web storage, Drag and Drop and those mentioned in the table above. There are many other APIs that you can find out about and integrate them into your web pages at the following site <https://platform.html5.org/>.

## 5. Changes from HTML5 to 5.2

It took a good 15 years from HTML4 to HTML5. This time it went a bit faster with the development of HTML5. Already two years after the release of HTML5 further new HTML5 specification was published by the W3C.

## 5.1 Changes in HTML 5.1

Although the change in this new specification was not big, nevertheless some new features and elements were added and some were removed to improve the structuring of the web. HTML 5.1 should simplify the life of web developers and offers interesting innovations. A great feature of the specification is the new picture element that allows to include different sized graphics for different sized screens, such as monitors, smartphones and tablets. This element should not replace the <img> element, but extend it. [66] In addition, the source tag within the picture element lists alternative image formats for browsers that do not support the images preferred format. This makes it easy for phones and tablets in portrait mode to get a meaningful slice of the image for slideshows. [67] As more and more users use smartphones and tablets to surf the web, web developers obviously have to work in a very space-saving way due to the size of the displays. Therefore the two new elements <details> and <summary> have been released, which allow to hide content and show it only when needed. These can be created very easily without the use of additional JavaScript. [68]

```
5 <details>
6   <summary>HTML 5.1 New Features</summary>
7   <p>picture element</p>
8   <p>hide element</p>
9 </details>
```

26. Figure <details> Example

▼ HTML 5.1 New Features

picture element

hide element

27. Figure: Output  
<details>

It was also specified that new proposed features must work in at least two browsers, otherwise they would be removed. Therefore some elements like <menu> and <menuitem> and many more which can be read on the W3C web page have been removed. A complete list of changes can be found on the official [HTML5.1](#) changes page.

## 5.2 Changes in HTML 5.2

The latest and most recent specification from the W3C is the HTML5.2 which was released on December 14, 2017. HTML5.2 similar to its predecessor, new features have been added and some removed. Possible bugs were fixed to make the web more effective and user-friendly. One of the new features is the <dialog> element which can be used to create simple dialog boxes to communicate with

the user, which I have already covered in the chapter 3.2.7. Also, a new Payment Request API was released to simplify payments on the Internet. The API is intended to copy data already entered by a user in previous order process and then automatically transfer this information in a new process. The advantage of this over normal autofill variants is that no fields are accidentally filled with incorrect information and it simplifies the navigation of the website. [69] Furthermore, the apple-touch-icon element was introduced for Apple users which allows to define icons in different sizes for Apple devices. Besides these new features, however, some elements have also been removed, such as the <keygen> element. Also like its predecessor, a complete list of changes can be found on the official [HTML5.2](#) changes page.

### **5.3 HTML – Living Standard**

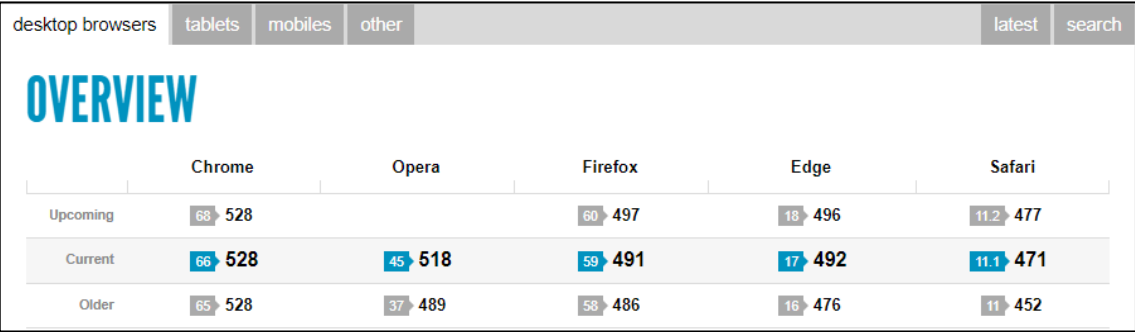
Many people wonder why the W3C has not released a new HTML5 specification for several years. The W3C had even worked on the further development of HTML5.3 and wanted to make new improvements and add new elements to make the web even better. However, on the HTML5.3 page it is written that *"This document is retired and MUST NOT be used for further technical work."* As already briefly mentioned in chapter 2.3.3, there was an agreement between the WHATWG and the W3C in 2019 to create a single common HTML and DOM specification. Furthermore, it was decided in this agreement that the WHATWG will be responsible for the maintenance of the HTML and DOM standards. [25] Therefore all current and new features and elements of HTML can be found on the WHATWG webpage. The HTML specifications of the WHATWG, to which all major browser vendors now belong, are updated regularly. There are no fixed version numbers and no release intervals. In contrast to W3C, which has tried to publish new recommendations annually, the WHATWG allows constant adaptations and extensions. [70]

## **6. Use of HTML5 Today**

### **6.1 Browser support for HTML5**

If you browse the Internet and visit websites or web applications with videos, music, and animations, you need a browser that can handle HTML5. Currently,

there are many popular browsers like Google Chrome, Firefox, Microsoft Edge, Opera and Apple Safari and many more that can handle HTML5 very well and support almost all elements and features. But also mobile and tablet browsers like Samsung Internet and Safari are getting better and dynamic and use especially HTML5 semantic and form elements to structure their pages in a more user-friendly way. Furthermore, HTML5 is also very well represented in gaming consoles such as Xbox One and Playstation, but also in smart TVs and e-readers. There are many good websites where you can easily check how many HTML5 elements current browsers support and compare them with each other. For example, you can make a test on the website <http://html5test.com/> or <https://caniuse.com/> to look which browsers support how many HTML5 elements.



28. Figure: Browser Comparison

I did this test for desktop browsers and compared them with each other and could see that HTML5 has established itself very well among web developers and websites. [71] Especially the HTML5 video player and APIs like Geolocation, Drag and Drop and so on are used by companies to offer users a very nice structured website and web application.

## 6.2 General Criticism of HTML5

HTML5 is a very long standing project and has had to deal with a lot of criticism in the past. Especially the different specification on the side of the W3C and the WHATWG caused a lot of confusion among web developers. Web developers had to wait for years until there were finally new updates and changes from the W3C. However, the whole thing works better today, through the WHATWG are always constantly made new updates and improvements. Nevertheless, there was the criticism that the WHATWG specification is mostly created by the Google employee Ian Hickson and a small number of web browser manufacturers and

that it does not work according to a democratic principle like the W3C. Therefore, many criticize that Ian Hickson alone has editing rights to the HTML5 draft and is responsible for many decisions affecting the specification, which may not be beneficial to every single web developer in the world. [72] However, there are some web developers who are of the opinion that HTML5 has some disadvantages. They argue that HTML is very complex and insecure. Moreover, you can't create dynamic websites and applications with HTML alone and that you have to create and program each page of your website individually and cannot edit it as one file. [73]

## **7. Conclusion**

HTML5, in contrast to its predecessors, has changed the Internet world in a very positive way. In addition to the features of previous HTML versions, HTML5 also includes a description of the structure of how elements must be nested. HTML5 also enables a set of JavaScript objects that can be integrated into HTML5 documents as APIs. While HTML 4.01 provides only rough structures of div elements, HTML5 provides new semantic elements. As a result, the structure of the code is no longer confusing and easy to understand and less prone to errors. In contrast to HTML4.01, external plug-ins are no longer necessary. HTML5 now has new tags for multimedia elements or various programming interfaces. With HTML5, the integration of audio and video files is also possible without an additional plug-in. Nowadays, interaction with the user is also very important. Therefore, new elements have been added to simplify the implementation of forms. But also new elements like dialog, summary and details that make the web more interactive and user friendly. Especially APIs like Geolocation, Drag and Drop and Webstorage make HTML5 very unique and enhance the user experience. Another advantage of HTML5 is the ability to create web applications that can better display content on smartphones and tablets. In addition, HTML5 is backward compatible, so that older browser versions also understand it. In conclusion, HTML5 is more than just a markup language, it is constantly evolving and justifies the Hype.

## Bibliography

- [1] "WHATWG," [Online]. Available: <https://html.spec.whatwg.org/#history-2>. [Accessed 10 April 2021].
- [2] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/World\\_Wide\\_Web](https://en.wikipedia.org/wiki/World_Wide_Web). [Accessed 10 April 2021].
- [3] "World Wide Web Foundation," [Online]. Available: <https://webfoundation.org/about/vision/history-of-the-web/>. [Accessed 10 April 2021].
- [4] "W3," [Online]. Available: <https://www.w3.org/People/Berners-Lee/Longer.html>. [Accessed 10 April 2021].
- [5] "Home.cern," [Online]. Available: <https://home.cern/science/computing/birth-web>. [Accessed 11 April 2021].
- [6] „Wikipedia,“ [Online]. Available: [https://en.wikipedia.org/wiki/Netscape\\_Navigator](https://en.wikipedia.org/wiki/Netscape_Navigator).
- [7] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Browser\\_wars](https://en.wikipedia.org/wiki/Browser_wars). [Accessed 11 April 2021].
- [8] A. Beattie, "Investopedia," 30 March 2020. [Online]. Available: <https://www.investopedia.com/ask/answers/08/microsoft-antitrust.asp>. [Accessed 12 April 2021].
- [9] "Wikiwand," [Online]. Available: [https://www.wikiwand.com/en/Browser\\_wars#/First\\_Browser\\_War\\_\(1995%E2%80%932001\)](https://www.wikiwand.com/en/Browser_wars#/First_Browser_War_(1995%E2%80%932001)). [Accessed 12 April 2021].
- [10] R. Raghavan, "Acodez," 21 August 2019. [Online]. Available: <https://acodez.in/browser-wars/>. [Accessed 13 April 2021].
- [11] "Wikiwand," [Online]. Available: [https://www.wikiwand.com/en/Browser\\_wars#/Second\\_Browser\\_War\\_\(2004%E2%80%932017\)](https://www.wikiwand.com/en/Browser_wars#/Second_Browser_War_(2004%E2%80%932017)). [Accessed 13 April 2021].
- [12] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Browser\\_wars#Second\\_Browser\\_War\\_\(2004%E2%80%932017\)](https://en.wikipedia.org/wiki/Browser_wars#Second_Browser_War_(2004%E2%80%932017)). [Accessed 13 April 2021].
- [13] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/World\\_Wide\\_Web\\_Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium). [Accessed 13 April 2021].
- [14] "W3C," [Online]. Available: <https://www.w3.org/Consortium/mission.html#principles>. [Accessed 13 April 2021].
- [15] "W3," [Online]. Available: <https://www.w3.org/standards/>. [Accessed 13 April 2021].
- [16] A. W. Longman, "W3," 1998. [Online]. Available: <https://www.w3.org/People/Raggett/book4/ch02.html>. [Accessed 19 April 2021].

- [17] "Landofcode.com," [Online]. Available: <http://www.landofcode.com/xhtml-tutorials/xhtml-history.php>. [Accessed 19 April 2021].
- [18] R. Shannon, "HTML Source," 21 August 2012. [Online]. Available: <https://www.yourhtmlsource.com/accessibility/xhtmlexplained.html>. [Accessed 29 April 2021].
- [19] „Landofcode.com,“ [Online]. Available: <http://www.landofcode.com/xhtml-tutorials/xhtml-dtd.php>.
- [20] "Tutorialspoint," [Online]. Available: [https://www.tutorialspoint.com/xhtml/what\\_is\\_xhtml.htm](https://www.tutorialspoint.com/xhtml/what_is_xhtml.htm). [Accessed 18 April 2021].
- [21] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/XHTML#:~:text=XHTML%201.0%20became%20a%20World,adaptation%20of%20the%20HTML5%20specification..> [Accessed 20 April 2021].
- [22] "WHATWG," [Online]. Available: <https://html.spec.whatwg.org/#history-2>. [Accessed 20 April 2021].
- [23] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/WHATWG>. [Accessed 17 April 2021].
- [24] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/HTML5>. [Accessed 20 April 2021].
- [25] J. Jaffe, "W3," 28 May 2019. [Online]. Available: <https://www.w3.org/blog/2019/05/w3c-and-whatwg-to-work-together-to-advance-the-open-web-platform/>. [Accessed 16 April 2021].
- [26] WHATWG, "WHATWG," [Online]. Available: <https://html.spec.whatwg.org/multipage/introduction.html#history-2>. [Accessed 16 April 2021].
- [27] D. Robinson, 10 May 2020. [Online]. Available: <https://www.topnotchdesigns.com/reasons-html5-future/>. [Accessed 3 May 2021].
- [28] „Seobility,“ [Online]. Available: [https://www.seobility.net/en/wiki/HTML\\_Doctype](https://www.seobility.net/en/wiki/HTML_Doctype). [Zugriff am 4 May 2021].
- [29] "Quackit," [Online]. Available: [https://www.quackit.com/html/html\\_4/tags/doctype\\_html\\_public.cfm](https://www.quackit.com/html/html_4/tags/doctype_html_public.cfm). [Accessed 18 April 2021].
- [30] "W3," [Online]. Available: <https://www.w3.org/QA/2002/04/valid-dtd-list.html>. [Accessed 16 April 2021].
- [31] "W3," [Online]. Available: [https://www.w3.org/wiki/Doctypes\\_and\\_markup\\_styles](https://www.w3.org/wiki/Doctypes_and_markup_styles). [Accessed 1 May 2021].
- [32] "W3C," [Online]. Available: <https://dev.w3.org/html5/html-author/#elements>. [Accessed 9 May 2021].

- [33] "W3Schools," [Online]. Available: [https://www.w3schools.com/html/html\\_attributes.asp](https://www.w3schools.com/html/html_attributes.asp). [Accessed 4 May 2021].
- [34] V. Bidaux, "webflow Blog," 16 December 2016. [Online]. Available: <https://webflow.com/blog/html5-semantic-elements-and-webflow-the-essential-guide#:~:text=So%2C%20in%20web%20design%2C%20a,%3E%20are%20non%2Dsemantic%20elements.&text=But%20%2C%20%3Ctable,They%20clearly%20define%20their%20content..> [Accessed 9 May 2021].
- [35] "W3Schools," [Online]. Available: [https://www.w3schools.com/html/html5\\_semantic\\_elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp). [Accessed 15 May 2021].
- [36] "TutorialRepublic," [Online]. Available: <https://www.tutorialrepublic.com/html-tutorial/html-layout.php>. [Accessed 15 May 2021].
- [37] "W3School," [Online]. Available: [http://w3schools-fa.ir/html/html5\\_new\\_elements.html](http://w3schools-fa.ir/html/html5_new_elements.html). [Accessed 15 May 2021].
- [38] J. K. Pete LePage, "HTML5Rocks," 19 November 2013. [Online]. Available: <https://www.html5rocks.com/en/tutorials/forms/html5forms/>. [Accessed 4 May 2021].
- [39] M. contributors, "Mozilla Developer Network," 30 April 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist>. [Accessed 16 May 2021].
- [40] M. contributors, "Mozilla Developer Network," 27 March 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress>. [Accessed 16 May 2021].
- [41] M. contributors, "Mozilla Developer Network," 30 April 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meter>. [Accessed 17 May 2021].
- [42] "Tutorialspoint," [Online]. Available: [https://www.tutorialspoint.com/html/html\\_input\\_tag.htm](https://www.tutorialspoint.com/html/html_input_tag.htm). [Accessed 16 May 2021].
- [43] W. Craig, "WebFX," [Online]. Available: <https://www.webfx.com/blog/web-design/new-html5-form-input-types/>. [Accessed 19 May 2021].
- [44] F. Ahmed, 21 January 2021. [Online]. Available: <https://www.c-sharpcorner.com/article/list-of-new-input-types-in-html-5/>. [Accessed 8 May 2021].
- [45] „W3Schools.com,“ [Online]. Available: [https://www.w3schools.com/html/html5\\_audio.asp](https://www.w3schools.com/html/html5_audio.asp).
- [46] "W3Schools," [Online]. Available: [https://www.w3schools.com/html/html\\_media.asp](https://www.w3schools.com/html/html_media.asp). [Accessed 23 May 2021].
- [47] „W3Schools,“ [Online]. Available: [https://www.w3schools.com/html/html5\\_canvas.asp](https://www.w3schools.com/html/html5_canvas.asp).
- [48] „Alligator.io,“ [Online]. Available: <https://alligator.io/html/dialog-element>.



- [49] Alex, „TechPlanet,“ 28 August 2020. [Online]. Available: <https://techplanet.today/post/what-is-an-api>.
- [50] „Wikipedia,“ [Online]. Available: <https://en.wikipedia.org/wiki/API>.
- [51] I. Ivanov, „codecoda,“ 15 October 2020. [Online]. Available: <https://codecoda.com/en/blog/entry/a-guide-to-understanding-html-apis>.
- [52] M. contributors, „Mozilla Developer Network,“ 30 April 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side\\_web\\_APIs/Video\\_and\\_audio\\_APIs](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Video_and_audio_APIs).
- [53] M. contributors, „Mozilla Developer Network,“ 1 April 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/TextTrack>.
- [54] M. contributors, „Mozilla Developer Network,“ 16 May 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/History\\_API](https://developer.mozilla.org/en-US/docs/Web/API/History_API).
- [55] „W3Schools,“ [Online]. Available: [https://www.w3schools.com/html/html5\\_draganddrop.asp](https://www.w3schools.com/html/html5_draganddrop.asp).
- [56] „Tizen Docs,“ [Online]. Available: <https://docs.tizen.org/application/web/guides/w3c/communication/web-messaging/>.
- [57] M. contributors, „Mozilla Developer Network,“ 11 February 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial).
- [58] M. contributors, „Mozilla Developer Network,“ 27 May 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Storage\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API).
- [59] M. contributors, „Mozilla Developer Network,“ 28 May 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API).
- [60] I. H. Anne van Kesteren, „W3C,“ 30 May 2008. [Online]. Available: <https://www.w3.org/TR/offline-webapps/>.
- [61] M. contributors, „Mozilla Developer Network,“ 7 May 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Workers\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API).
- [62] M. contributors, „Mozilla Developer Network,“ 25 May 2021. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API).
- [63] M. contributors, „Mozilla Developer Network,“ 12 March 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/File>.
- [64] „W3Schools,“ [Online]. Available: [https://www.w3schools.com/html/html5\\_serversentevents.asp](https://www.w3schools.com/html/html5_serversentevents.asp).
- [65] „W3Schools,“ [Online]. Available: [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp).
- [66] „W3Schools,“ [Online]. Available: [https://www.w3schools.com/tags/tag\\_picture.asp](https://www.w3schools.com/tags/tag_picture.asp).

- [67] „W3Schools,“ [Online]. Available:  
[https://www.w3schools.com/html/html\\_images\\_picture.asp#:~:text=The%20HTML%20element%20gives,current%20view%20and%20For%20device..](https://www.w3schools.com/html/html_images_picture.asp#:~:text=The%20HTML%20element%20gives,current%20view%20and%20For%20device..)
- [68] A. Monus, „HONGKIAT,“ 9 April 2018. [Online]. Available:  
<https://www.hongkiat.com/blog/html-5-1-new-features/>.
- [69] M. contributors, „Mozilla Developer Network,“ 18 December 2020. [Online]. Available:  
[https://developer.mozilla.org/en-US/docs/Web/API/Payment\\_Request\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Payment_Request_API).
- [70] WHATWG, „HTML Living Standard,“ 31 May 2021. [Online]. Available:  
<https://html.spec.whatwg.org/>.
- [71] „HTML5test,“ [Online]. Available: <https://html5test.com/results/desktop.html>. [Zugriff am 26 05 2021].
- [72] „Wikipedia,“ 23 July 2020. [Online]. Available: <https://de.zxc.wiki/wiki/HTML5#Kritik>.
- [73] S. Besra, „Techtually,“ 28 September 2020. [Online]. Available:  
<https://www.techtually.com/advantages-and-disadvantages-of-html/>.
- [74] „W3,“ 1 November 2016. [Online]. Available: <https://www.w3.org/TR/2016/REC-html51-20161101/changes.html>.
- [75] „Ecomspark,“ 25 July 2015. [Online]. Available: <https://www.ecomspark.com/html-doctypes-a-complete-guide/>. [Accessed 18 April 2021].
- [76] GK, „Ecomspark,“ 25 July 20115. [Online]. Available:  
<https://www.ecomspark.com/html-doctypes-a-complete-guide/>. [Accessed 1 May 2021].
- [77] „Bluekatanasoft,“ [Online]. Available: <https://www.bluekatanasoft.com/web-development/html/html-elements-and-tags/>. [Accessed 2 May 2021].
- [78] G. Keshri, „Wordpress,“ 29 April 2014. [Online]. Available:  
<https://gautamkeshri.wordpress.com/2014/04/29/how-is-the-page-structure-of-html-5-different-from-html-4-or-previous-html/>. [Accessed 13 May 2021].