

HTML and the web

Learning Objectives

- understanding client/server communication
 - writing HTML code
 - knowing about the importance of semantic HTML
-

How the web works

The world wide web is a network of computers that can exchange information with each other. There are many different protocols that define the rules on how machines communicate. Browsers use HTTP (Hypertext Transfer Protocol) to communicate with web servers.

- The URL (Uniform Resource Locator) is the unique address of a resource on the web contains a human readable domain name, that needs to be resolved to the technical IP (Internet Protocol) address of the web server via a DNS (Domain Name Server)
- The browser sends a **GET** (that's an HTTP method) **request** to load a HTML (Hyper Text Markup Language) document from a web server
- The web server sends a **response** containing the document
- Often the HTML code contains references to additional resources (CSS (Cascading Style Sheet) files, images, etc.), which the browser then also requests from the server
- The browser **renders** the received content to the screen and makes it interactive
- Browsers might also request additional data from servers later via subsequent **GET** or **POST** requests



HTML basics

HTML (Hyper Text Markup Language) is used to express text in a structured way. HTML tags indicate what kind of element is displayed on the website. For example, a headline is written like this:

```
<h1>I am a headline!</h1>
```

The content considered as headline is wrapped within an **opening tag** and a **closing tag**. The whole thing is called an **element**.

Elements are nested into each other to create structure and hierarchy.

```
<h1>I am a <em>headline!</em></h1>
```

Some elements can't contain any other elements and therefore don't have a closing tag. They are self-closing and called *empty elements*.

```
<hr />  
or  
<br />
```

HTML tag attributes

Some elements require some more information in order to work properly. This information is specified via attributes, for example:

- the source of an image

```

```

- the destination of an anchor element

```
<a href="https://example.com"> click me </a>
```

- the type of an input element

```
<input type="date" />
```

💡 The [MDN web docs](#) contain detailed information about elements and corresponding attributes.

Layout of an HTML file

Every HTML document starts with a [doctype](#) followed by the `<html>` element, which consists of two main parts:

- The `<head>` contains important meta information for the browser like
 - the charset (utf-8)
 - the favicon displayed in the tab
 - the title of the website
 - CSS and JavaScript files needed for the website
- The `<body>` contains the visible content of the website structured by html elements

```
<!DOCTYPE html>  
<html>  
  <head>  
    ... meta information, additional links to CSS / JavaScript files ...  
  </head>  
  <body>  
    ... elements displayed on the web page ...
```

```
</body>
</html>
```

List of common HTML elements

element	meaning
<code><head></head></code>	only once per website, includes meta data and linked files
<code><body></body></code>	only once per website, includes the html website content
<code><h1></h1></code>	only once per website, a level one heading
<code><h2></h2></code>	a level two heading
<code><p></p></code>	a paragraph
<code><a></code>	an anchor (link)
<code></code>	an image (self-closing / empty)
<code><form></form></code>	a form element
<code><input></code>	an input field (self-closing / empty)
<code><button></button></code>	a clickable element equipped with some kind of functionality

💡 A comprehensive [list of all html elements can be found at the MDN web docs](#).

Structuring a Website

Developers have two main tools to express a meaningful structure in a website:

1. Using semantic HTML elements
2. Nesting / grouping of HTML elements

Semantic HTML

Semantic HTML elements not only divide the content of the web page into distinct parts, but also describe the function or purpose of the elements. This has two major benefits:

- The HTML becomes more understandable for other developers
- Accessibility tools and search engines can interpret the website

Therefore, one should use semantic HTML elements whenever possible.

List of Semantic HTML elements

element	meaning
<code><main></main></code>	only once per website, includes the main content of the page

element	meaning
<code><section></code> <code></section></code>	a generic standalone section of a document
<code></code> / <code></code> <code></code>	a list of elements with the same structure, only has <code></code> elements as direct children
<code><nav></nav></code>	a navigation bar
<code><aside></code> <code></aside></code>	element representing a portion of a document whose content is only indirectly related to the main content
<code><article></code> <code></article></code>	representing a self-containing part of the website, which is intended to be independently distributable or reusable
<code><header></code> <code></header></code>	representing introductory content, typically a group of introductory or navigational aids
<code><footer></code> <code></footer></code>	typically contains information about the author of the section, copyright data or links to related documents



You can find a comprehensive [list of semantic html elements in the MDN web docs](#).

Nesting HTML elements

Nesting groups elements together in a meaningful way. The element containing the other elements is called the **parent element**, which contains one or more **child elements**.

The following cases are typical examples of nested elements:

- ```

 first item
 second item
 third item

```
- ```
<article>
  <h2>Some headline</h2>
  <p>I am a paragraph...</p>
  <a href="https://www.github.com">a link to another website</a>
</article>
```
- ```
<button>

 submit
</button>
```

Below is a sketch of how semantic elements can be nested in a web page.



## Emmet

Visual Studio Code has a useful tool called Emmet which lets you autocomplete a lot of code by just typing certain snippets and pressing the `Tab` key afterwards. Try these snippets inside an HTML file and see what happens:

- `!`
- `.highlight`
- `button#red`
- `ul>li.card\*10`



You can learn about more Emmet commands in [this cheat-sheet](#)

---

## Resources

- [MDN: Introduction to HTML](#)
- [MDN: Getting started with HTML](#)
- [MDN: HTML Elements](#)
- [MDN: Semantic elements: Glossary](#)
- [MDN: HTML attributes](#)