Fundamentals of modern web development

Jonas Duri

E-Mail: jonas.duri@hs-duesseldorf.de

HSD | [https://hs-duesseldorf.de]
University Lecturer

E. Breuninger GmbH | [https://www.breuninger.com/de/]
Software Engineer

Boxine GmbH | [https://tonies.com]
Senior Frontend Developer

Stepstone Continental Europe | [https://stepstone.com]
Senior Frontend Developer

Ubisoft Blue Byte | [https://ubisoft.com]
Frontend Developer

Mapudo GmbH | [https://mapudo.com]
Frontend Developer

Digital Agency OYGO | [https://oygo.com/]
Frontend Developer

Freelance Webdesigner & Developer

German Army in Oldenburg

What is HTML

It stands for *Hypertext Markup Language*. **HTML** is the language used to create webpages. *Hypertext* refers to the hyperlinks that an HTML page may contain. *Markup language* refers to the way tags are used to define the page structure and elements within the page.

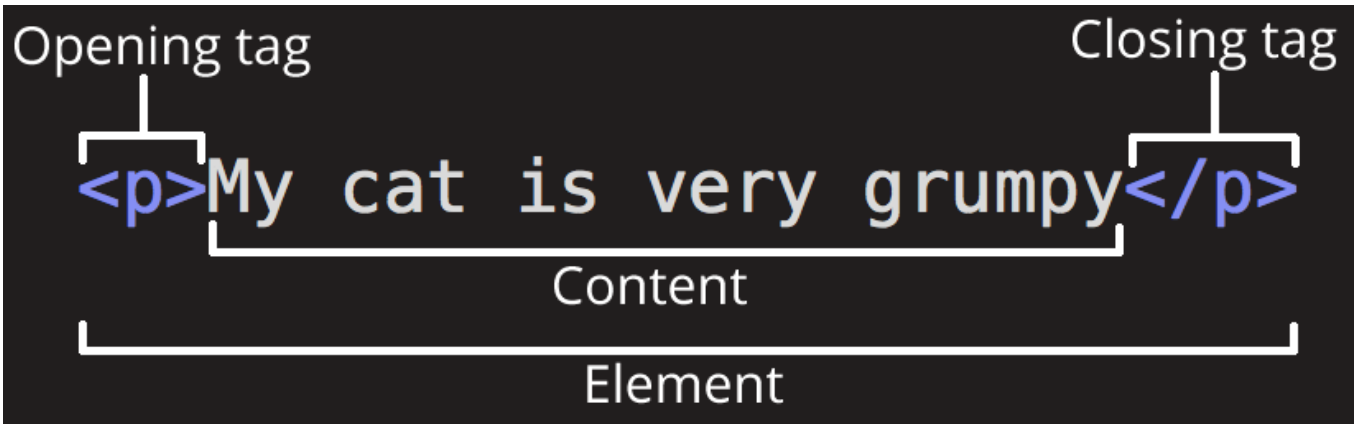
A HTML document is a simple text file with the extension `.html`.

```
<!doctype html>
<html>
  <head>
    <title>HSD</title>
  </head>
  <body>
    <p>This is an example of a paragraph in HTML.</p>
  </body>
</html>
```

Everything in HTML is a TAG

A HTML tag is the basic building block of the web.
Tags consist of an **opening** and a **closing** tag.

`<opening> ... </closing>`



Opening tag, content and closing tag create an **HTML Element**

Inside a tag can be **text**, more **tags** or both.

```
<p>Inside this "p" tag is just text</p>
```

```
<div>
  <p>Text inside a "p" tag, inside a "div" tag</p>
</div>
```

```
<div>
  <p>This is text inside a "p" tag.<strong>This is nested text</strong>.</p>
  This is text inside a "div" tag
</div>
```

Exception

Some tags are *self-closing*. When nesting content inside a tag doesn't make sense it can be self-closing
Example:

```
<div>
  <h1>This is a headline</h1>
  <img src="/link/to/image.jpg" alt="image description" />
</div>
```

Extra

You can write comments in HTML. These comments won't appear on the page and search engines won't index the comments.

```
<div>
  <p>Lorem ipsum</p> <!-- Replace with "real" content later -->
</div>
```

HTML Elements can have „attributes"

Every Element can have an abritrary number of „attributes“ attached to it. Some attributes are necessary for an element to work correctly.

```
1 <img src="/link/to/image.jpg" /> <!-- "src" is an attribtue -->
```

In order to show an image to a web page, the `img` tag needs to have a `src` attribute.

The most commonly used attributes are `class` and `id`. More on that later.

Semantic

HTML should be coded to represent the data that will be populated and not based on its default presentation styling. Presentation (how it should look), is the sole responsibility of CSS.

Some of the benefits from writing semantic markup:

- Search engines will consider its contents as important keywords to influence the page's search rankings (see SEO)
- Screen readers can use it as a signpost to help visually impaired users navigate a page

```
<!-- Headings -->
<h1>My main title</h1>
<h2>My top level heading</h2>
<h3>My subheading</h3>
<h4>My sub-subheading</h4>

<!-- paragraph -->
<p>This is a single paragraph</p>

<!-- lists -->
<ul><!-- unordered list -->
  <li>Apples</li>
  <li>Oranges</li>
  <li>Carrots</li>
</ul>

<ol><!-- ordered list -->
  <li>Get up</li>
  <li>Drink coffee</li>
  <li>Go to university</li>
</ol>

<!-- links -->
<a href="https://google.com">Visit Google</a>

<!-- navigations -->
<nav>
  <a href="/">Home</a>
```

```
  <a href="/about">About</a>
  <a href="/contact">contact</a>
</nav>

<!-- others -->
<article>...</article>
<aside>...</aside>
<details>...</details>
<figcaption>...</figcaption>
<figure>...</figure>
<footer>...</footer>
<header>...</header>
<main>...</main>
<mark>...</mark>
<section>...</section>
<summary>...</summary>
<time>...</time>
<em>...</em>
```

Non semantic elements

```
<div>...</div>
<span>...</span>
```

Anatomy of an HTML document

```
<!DOCTYPE html>
<!-- Tell the browser that this is an HTML web page -->
<html>
<!-- This element wraps all the content on the entire page -->
  <head>
    <!-- Add non-visual content here -->
    <title>Anatomy of an HTML document</title>
    <!-- The title of your browser tab -->
  </head>
  <body>
    <!-- This contains all the content that you want to show to web users when they visit your page -->
    <h1>Anatomy of an HTML document</h1>
    <p>Some content</p>
    <img src="//via.placeholder.com/350x150" alt="A placeholder image">
  </body>
</html>
```

Resources

Learn more about HTML

Learning

- Mozilla Developer Network - HTML Tutorial
- HTML Reference

Working with code

- https://code.visualstudio.com/
- Codepen - Cloud Editor

Debugging

- Validate HTML

What is CSS

CSS (Cascading Style Sheets) is the code you use to style your webpage.

A CSS stylesheet is a simple text file with the extension `.css`.

```
p {
  color: red;
}
h1 {
  color: green;
}
```

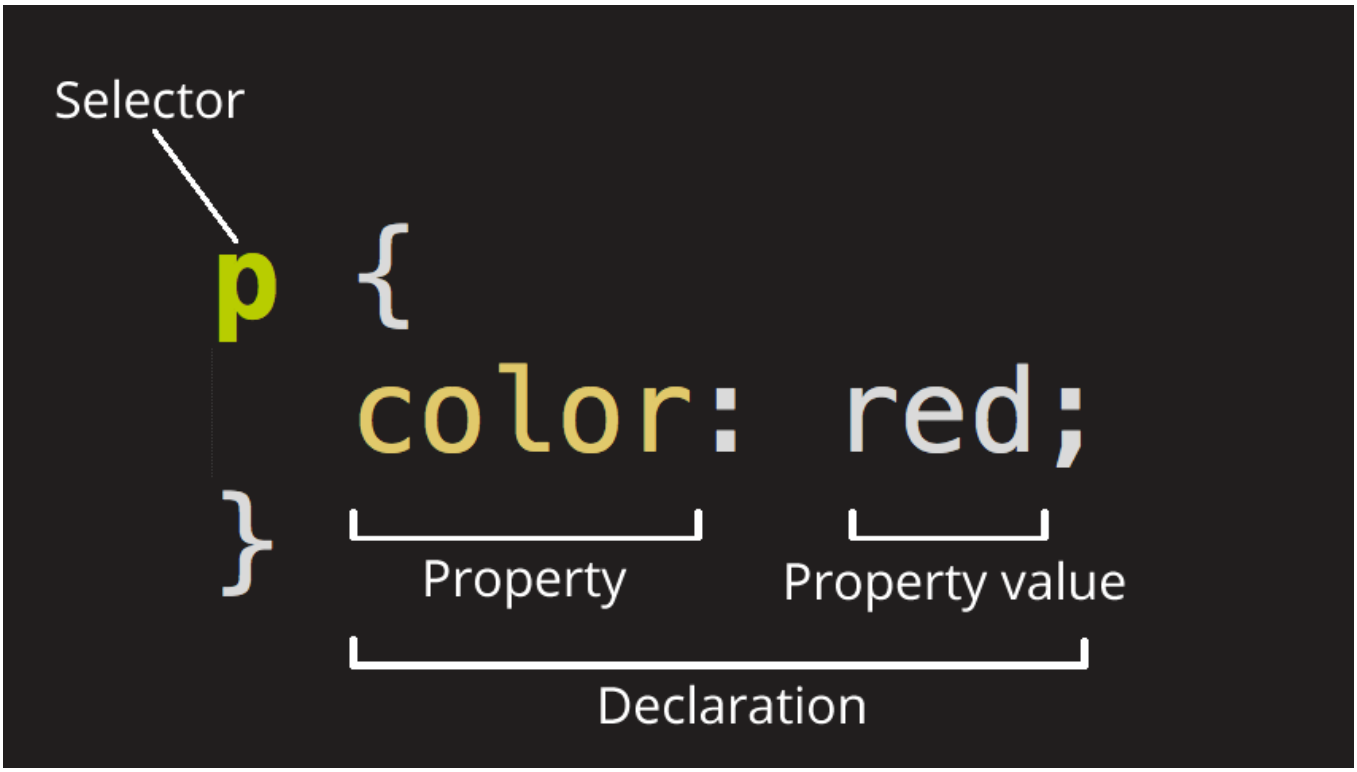
You can write CSS inside a `<style>` element in HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Anatomy of an HTML document</title>
    <style>
    p {
      color: red;
    }
    h1 {
      color: green;
    }
    </style>
  </head>
  <body>
    <h1>Anatomy of an HTML document</h1><!-- this will be "green" -->
    <p>Some content</p><!-- this will be "red" -->
  </body>
</html>
```

You can also write CSS directly inside an HTML element. This method is called **inline-styles**.

```
<p style="color: red;">Some content</p>
```

Structure of a css rule


Selector
p {
color: red;
}
Property
Property value
Declaration

Using external stylesheets

While using a `<style>` element in HTML directly is possible, it is recommended to use external stylesheets.

You can link to an external CSS file with a `<link>` element.

styles.css

```
p {
  color: red;
}
h1 {
  color: green;
}
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Anatomy of an HTML document</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Anatomy of an HTML document</h1><!-- this will be "green" -->
    <p>Some content</p><!-- this will be "red" -->
  </body>
</html>
```

Styling text with css

The CSS properties used to style text generally fall into two categories.

Font styles: Properties that affect the font that is applied to the text, affecting what font is applied, how big it is, whether it is bold, italic, etc.

Text layout styles: Properties that affect the spacing and other layout features of the text, allowing manipulation of, for example, the space between lines and letters, and how the text is aligned within the content box.

Resources

Learn more about CSS

Learning

- Mozilla Developer Network - CSS Tutorial
- CSS Reference

Working with CSS

- Google Chrome

Debugging

- Learn to work with Chrome Devtools