

## Eine Einführung in die Terminologie

Anders als im Bereich der Musik wird im Bereich der Programmierung sehr viel mit freien Lizenzen gearbeitet, das heißt, Informationen und (fast) alle Problemlösungen kann man im Internet finden und unter der Voraussetzung nutzen, dass man die Anleitungen bzw. Erläuterungen versteht, wie andere das Problem gelöst haben. Die bekannteste Fundgrube für Problemlösungen der Programmierung ist die Website **Stackoverflow** (<https://stackoverflow.com>), die jeweils hilfreichste Lösung ist an einem grünen Check-Symbol ersichtlich.

Eine Webseite besteht aus Dateien, die in einer **Markup Language** (ML) bzw. maschinenlesbaren Sprache geschrieben werden. Die ML gliedert Texte und andere Daten wie z.B. Bilder. Ein Browser ist ein Programm, das die Hypertext ML lesen und darstellen kann, ML-Dateien erkennt ein Browser an der Extension html (\*.html).

Darüber hinaus gehören zu einer Website **Cascading Style Sheets** (CSS). CSS enthalten Layout-Anweisungen und sagen dem Browser, wie er die in Markup Language notierten Elemente anzeigen soll (z.B. ob in rot, grün oder blau). CSS-Anweisungen lassen sich in der ML (inline), im Head-Tag einer Datei oder in einer externen Datei notieren. Zu empfehlen ist das Notieren der CSS-Anweisungen in einer externen Datei, weil sich durch die Trennung von Logik und Layout Webseiten gut warten und global verändern lassen.

Für das Auswerten von Benutzereingaben, zum Verändern, Nachladen oder Erzeugen von Inhalten sowie für Animationen wird **JavaScript** (JS) verwendet.

### HTML (=ML)

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport"
      content="width=device-width,
      initial-scale=1.0"
    />
    <title>Titel</title>
  </head>
  <body>
    <div id="container">
      <h1>Hallo Welt</h1>
      <p class="redText">Webprogrammierung</p>
    </div>
  </body>
</html>
```

### Styles (CSS)

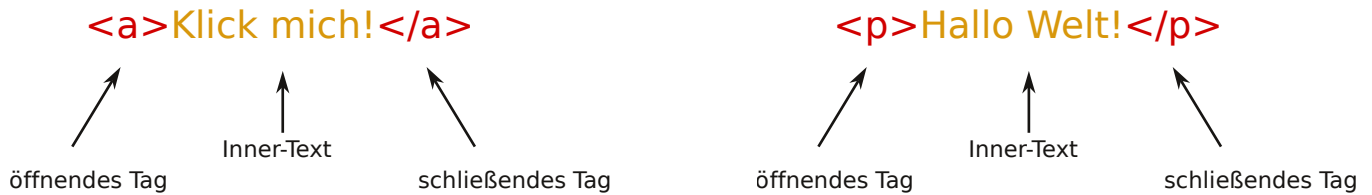
```
body {
  margin: 0;
  padding: 0;
  background-color: lightgray;
}
#container {
  width: 70vw;
  height: 50vh;
  margin: 100px auto;
  padding: 20px;
}
h1{
  font-style: italic;
}
.redText {
  color: darkred;
}
```

## Hilfreiche Seiten

HTML	<a href="https://wiki.selfhtml.org/wiki/HTML/Tutorials">https://wiki.selfhtml.org/wiki/HTML/Tutorials</a>
CSS	<a href="http://www.css4you.de/">http://www.css4you.de/</a> und <a href="https://wiki.selfhtml.org/wiki/CSS">https://wiki.selfhtml.org/wiki/CSS</a>
Flexbox	<a href="https://css-tricks.com/snippets/css/a-guide-to-flexbox/">https://css-tricks.com/snippets/css/a-guide-to-flexbox/</a> und <a href="https://blog.kulturbause.de/2013/07/einfuehrung-in-das-flexbox-modell-von-css/">https://blog.kulturbause.de/2013/07/einfuehrung-in-das-flexbox-modell-von-css/</a>
Javascript	<a href="https://wiki.selfhtml.org/wiki/JavaScript">https://wiki.selfhtml.org/wiki/JavaScript</a> <a href="https://molily.de/js/">https://molily.de/js/</a> Online-Tester: <a href="https://js.do/">https://js.do/</a>

## Was ist ein HTML-Element?

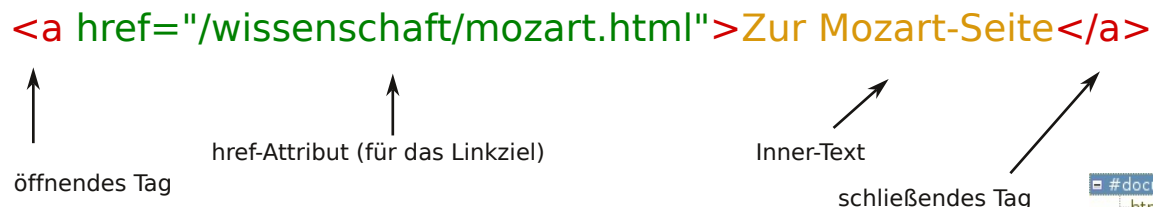
Ein HTML-Element besteht bis auf wenige Ausnahmen aus einem öffnenden und einem schließenden Tag sowie einem Text, der im Inneren steht, z.B.:



Wichtige Elemente sind **Inline-Elemente** wie `<a>` (Anker bzw. Link) und `<span>` (unsichtbare Hülle für CSS-Anweisungen), **Blocklevel-Elemente** wie `<h1>` (Überschriften), `<p>` (Paragraph für Texteingaben) und `<div>` (unsichtbare Hülle für CSS-Anweisungen) sowie **Flow-Elemente**, die nur in Verbindung mit bestimmten anderen Elementen verwendet werden dürfen, z.B. ein `<li>` (Listenelement) nur in einer `<ul>` oder `<ol>` (Liste), eine `<tr>` (Tabellenreihe) nur in einer `<table>` (Tabelle) und eine `<td>` (Tabellenzelle) nur in einer `<tr>` einer `<table>`.

## Was sind Attribute?

Das oben abgebildete Anker- bzw. Link-Element würde im HTML noch nicht viel bewirken, weil die Zieleingabe fehlt, wohin der Link führen soll. Angaben wie Pfade (für Links und Bilder), aber auch für ID's und Klassen (um die Elemente über CSS referenzieren zu können), werden in sogenannten Attributen notiert. Das `<a>`-Element von oben könnte wie folgt mit einem Attribut versehen werden:



## Das Document Object Model (DOM)

Das Document Object Model ist eine Programmierschnittstelle, welche HTML-Seiten (und auch XML-Dokumente) als eine Baumstruktur aus Objekten beschreibt. Das Wurzel-Element ist das `<html>`-Tag, jeder Knoten des Baums steht wie oben beschrieben für ein untergeordnetes Element bzw. ein Objekt in der Seite (wie z.B. einen Textabsatz `<p>`, Überschriften von `<h1>` bis `<h6>`, Container- `<div>`, Audio- `<audio>` oder Video-Elemente `<video>`). Die wesentliche Aufgabe des DOMs ist es, Webseiten mit Scripts oder Programmiersprachen zu verbinden. Die Repräsentation eines Dokuments als Baum erlaubt es, mit einer Script- bzw. Programmiersprache einzelne Elemente dieses Baumes anzusprechen und über entsprechenden Code zu verändern. Über die Tags der Elemente selbst sowie über Klassen und IDs lassen sich beispielsweise Elemente im DOM identifizieren und können anschließend – in der Regel mit JavaScript – manipuliert bzw. verändert werden.

